12: Polyphase Filters

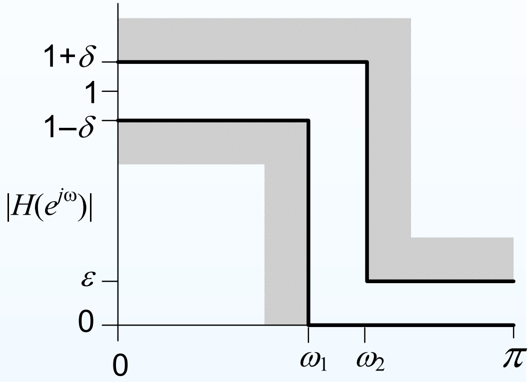
Heavy Lowpass filtering

Filter Specification:

Sample Rate: 20 kHz

Passband edge: 100 Hz ($\omega_1 = 0.03$)

Stopband edge: 300 Hz ($\omega_2 = 0.09$)



Heavy Lowpass filtering

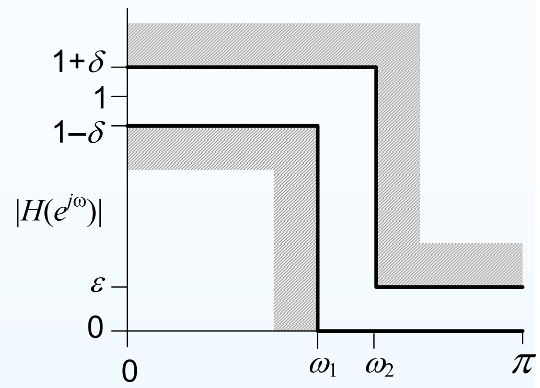
Filter Specification:

Sample Rate: 20 kHz

Passband edge: 100 Hz ($\omega_1 = 0.03$)

Stopband edge: 300 Hz ($\omega_2 = 0.09$)

Passband ripple: ± 0.05 dB ($\delta = 0.006$)



Heavy Lowpass filtering

Filter Specification:

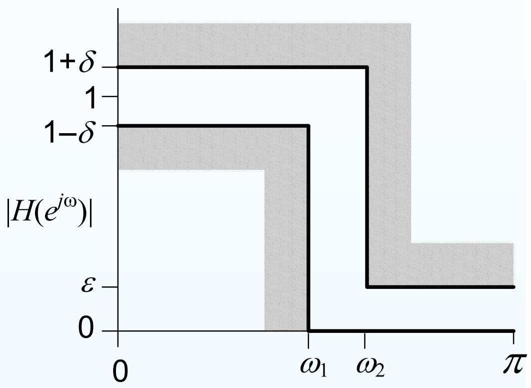
Sample Rate: 20 kHz

Passband edge: 100 Hz ($\omega_1 = 0.03$)

Stopband edge: 300 Hz ($\omega_2 = 0.09$)

Passband ripple: ± 0.05 dB ($\delta = 0.006$)

Stopband Gain: -80 dB ($\epsilon = 0.0001$)



Heavy Lowpass filtering

Filter Specification:

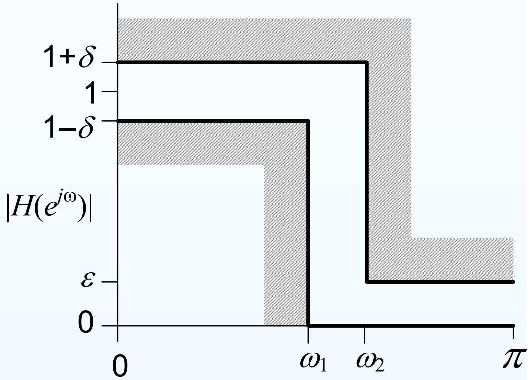
Sample Rate: 20 kHz

Passband edge: 100 Hz ($\omega_1 = 0.03$)

Stopband edge: 300 Hz ($\omega_2 = 0.09$)

Passband ripple: ± 0.05 dB ($\delta = 0.006$)

Stopband Gain: -80 dB ($\epsilon = 0.0001$)

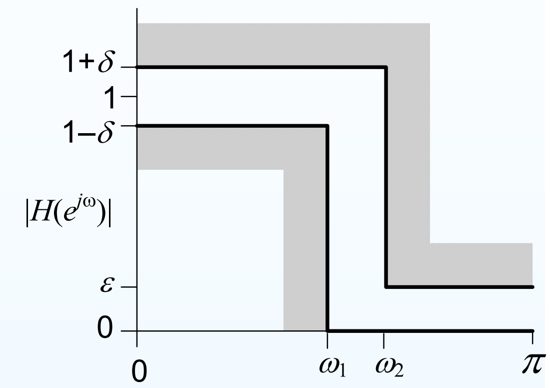


This is an extreme filter because the cutoff frequency is only 1% of the Nyquist frequency.

Heavy Lowpass filtering

Filter Specification:

Sample Rate: 20 kHz
Passband edge: 100 Hz ($\omega_1 = 0.03$)
Stopband edge: 300 Hz ($\omega_2 = 0.09$)
Passband ripple: ± 0.05 dB ($\delta = 0.006$)
Stopband Gain: -80 dB ($\epsilon = 0.0001$)



This is an extreme filter because the cutoff frequency is only 1% of the Nyquist frequency.

Symmetric FIR Filter:

Heavy Lowpass filtering

Filter Specification:

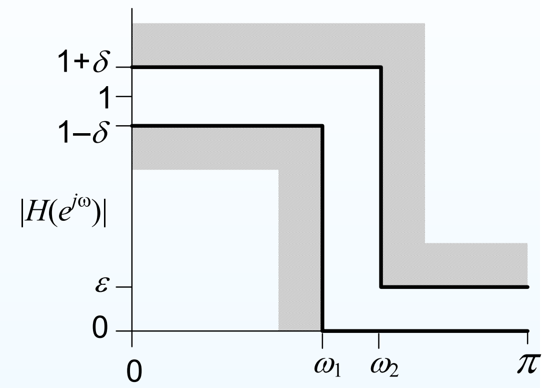
Sample Rate: 20 kHz

Passband edge: 100 Hz ($\omega_1 = 0.03$)

Stopband edge: 300 Hz ($\omega_2 = 0.09$)

Passband ripple: ± 0.05 dB ($\delta = 0.006$)

Stopband Gain: -80 dB ($\epsilon = 0.0001$)

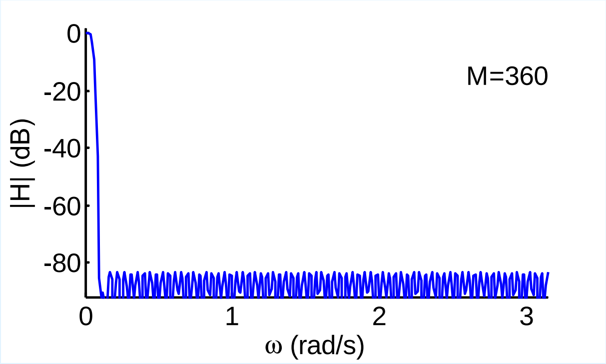


This is an extreme filter because the cutoff frequency is only 1% of the Nyquist frequency.

Symmetric FIR Filter:

Design with Remez-exchange algorithm

Order = 360



Heavy Lowpass filtering

Filter Specification:

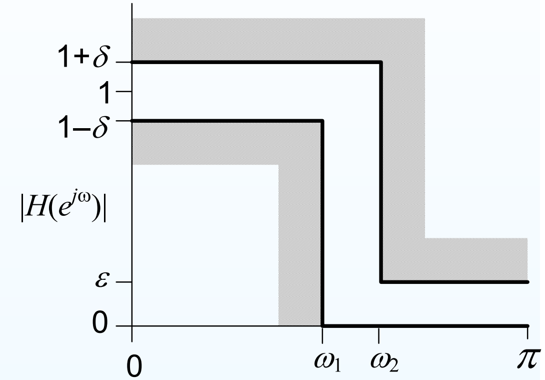
Sample Rate: 20 kHz

Passband edge: 100 Hz ($\omega_1 = 0.03$)

Stopband edge: 300 Hz ($\omega_2 = 0.09$)

Passband ripple: ± 0.05 dB ($\delta = 0.006$)

Stopband Gain: -80 dB ($\epsilon = 0.0001$)

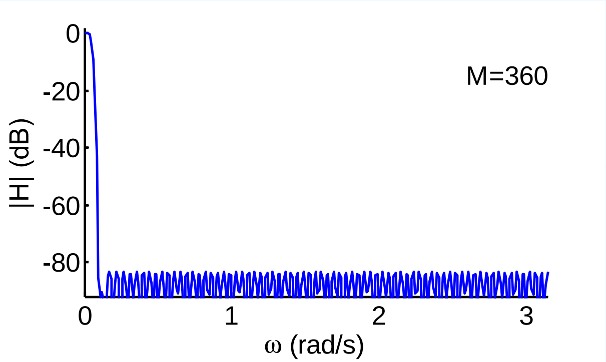


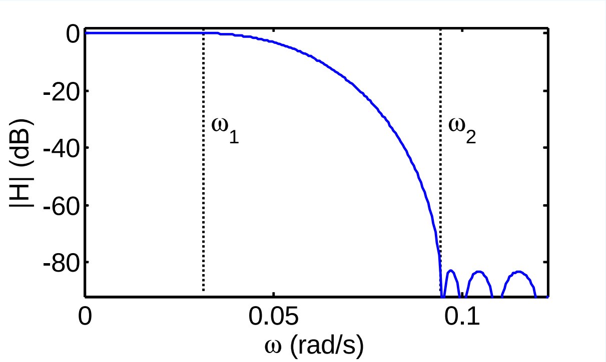
This is an extreme filter because the cutoff frequency is only 1% of the Nyquist frequency.

Symmetric FIR Filter:

Design with Remez-exchange algorithm

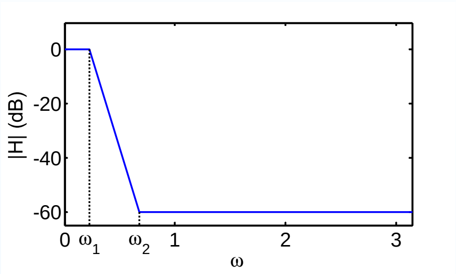
Order = 360





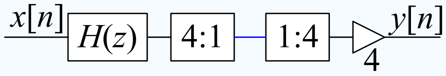
Maximum Decimation Frequency

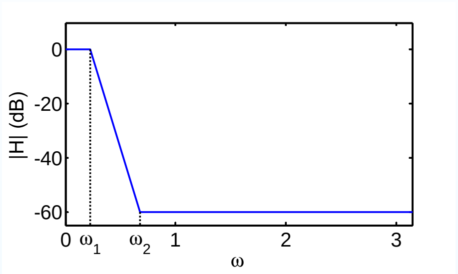
If a filter passband occupies only a small fraction of $[0, \pi]$, we can downsample then upsample without losing information.



Maximum Decimation Frequency

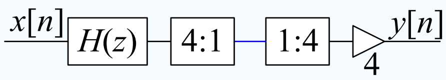
If a filter passband occupies only a small fraction of $[0, \pi]$, we can downsample then upsample without losing information.

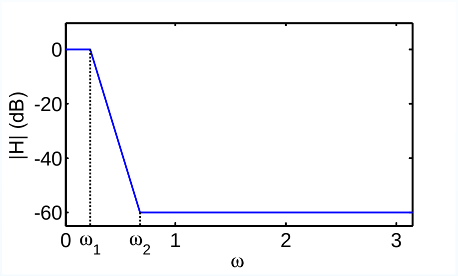




Maximum Decimation Frequency

If a filter passband occupies only a small fraction of $[0, \pi]$, we can downsample then upsample without losing information.

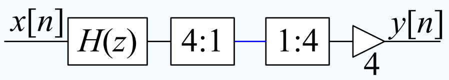


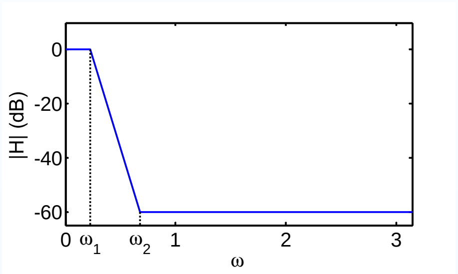


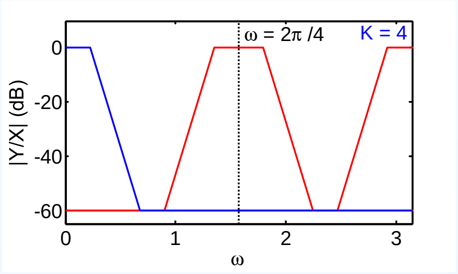
Downsample: aliased components at offsets of $\frac{2\pi}{K}$ are almost zero because of $H(z)$

Maximum Decimation Frequency

If a filter passband occupies only a small fraction of $[0, \pi]$, we can downsample then upsample without losing information.





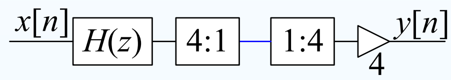


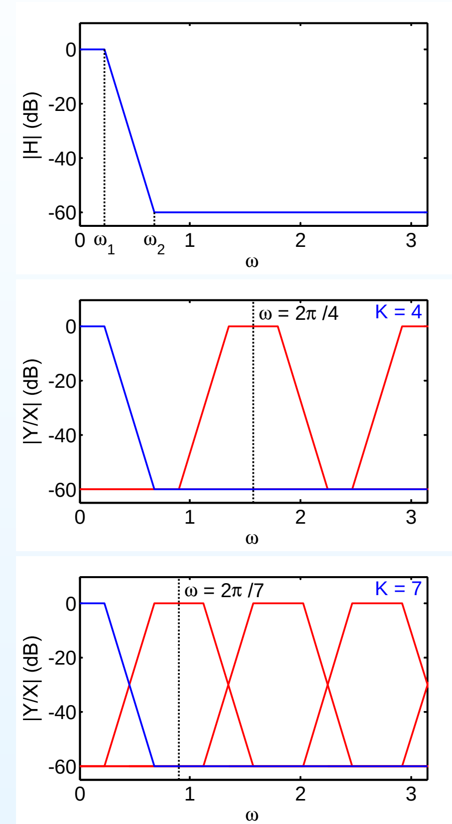
Downsample: aliased components at offsets of $\frac{2\pi}{K}$ are almost zero because of $H(z)$

Upsample: Images spaced at $\frac{2\pi}{K}$ can be removed using another low pass filter

Maximum Decimation Frequency

If a filter passband occupies only a small fraction of $[0, \pi]$, we can downsample then upsample without losing information.





Downsample: aliased components at offsets of $\frac{2\pi}{K}$ are almost zero because of $H(z)$

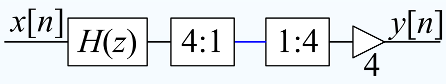
Upsample: Images spaced at $\frac{2\pi}{K}$ can be removed using another low pass filter

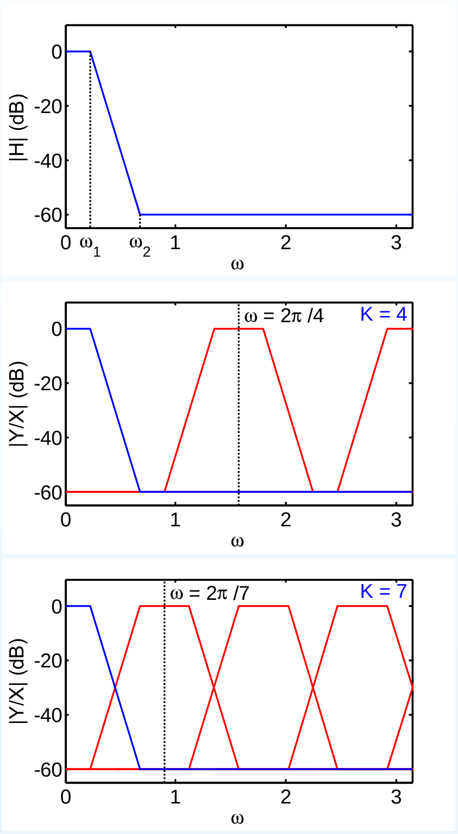
To avoid aliasing in the passband, we need

$$\frac{2\pi}{K} - \omega_2 \ge \omega_1 \quad \Rightarrow \quad K \le \frac{2\pi}{\omega_1 + \omega_2}$$

Maximum Decimation Frequency

If a filter passband occupies only a small fraction of $[0, \pi]$, we can downsample then upsample without losing information.





Downsample: aliased components at offsets of $\frac{2\pi}{K}$ are almost zero because of $H(z)$

Upsample: Images spaced at $\frac{2\pi}{K}$ can be removed using another low pass filter

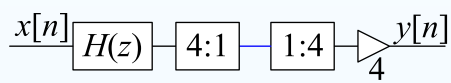
To avoid aliasing in the passband, we need

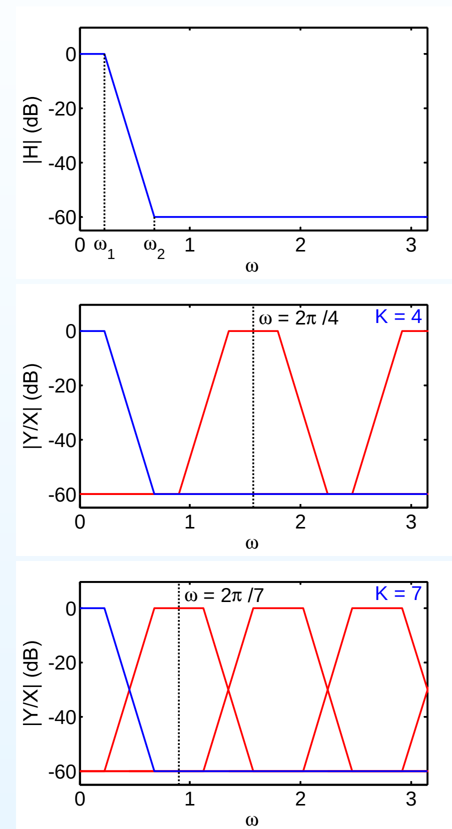
$$\frac{2\pi}{K} - \omega_2 \geq \omega_1 \quad \Rightarrow \quad K \leq \frac{2\pi}{\omega_1 + \omega_2}$$

Centre of transition band must be $\leq$ intermediate Nyquist freq, $\frac{\pi}{K}$

Maximum Decimation Frequency

If a filter passband occupies only a small fraction of $[0, \pi]$, we can downsample then upsample without losing information.





Downsample: aliased components at offsets of $\frac{2\pi}{K}$ are almost zero because of $H(z)$

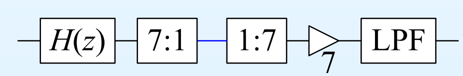
Upsample: Images spaced at $\frac{2\pi}{K}$ can be removed using another low pass filter

To avoid aliasing in the passband, we need

$$\frac{2\pi}{K} - \omega_2 \geq \omega_1 \quad \Rightarrow \quad K \leq \frac{2\pi}{\omega_1+\omega_2}$$

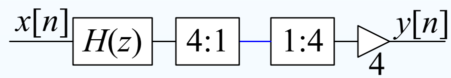
Centre of transition band must be $\leq$ intermediate Nyquist freq, $\frac{\pi}{K}$

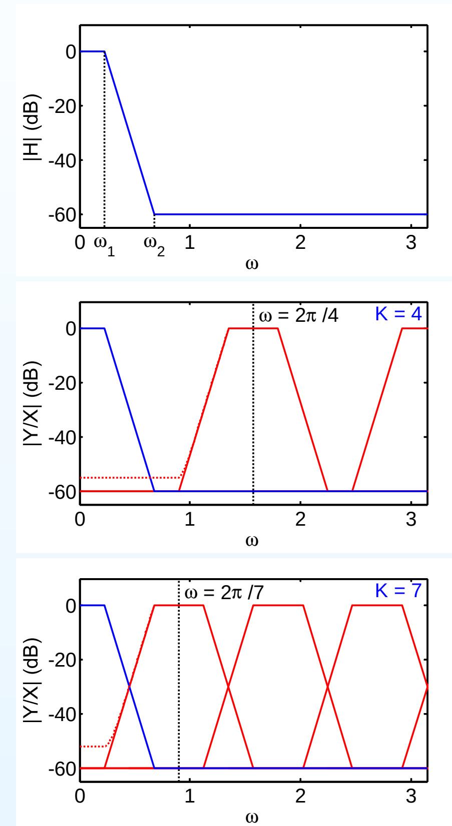
We must add a lowpass filter to remove the images:



Maximum Decimation Frequency

If a filter passband occupies only a small fraction of $[0, \pi]$, we can downsample then upsample without losing information.





Downsample: aliased components at offsets of $\frac{2\pi}{K}$ are almost zero because of $H(z)$

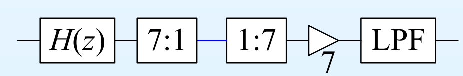
Upsample: Images spaced at $\frac{2\pi}{K}$ can be removed using another low pass filter

To avoid aliasing in the passband, we need

$$\frac{2\pi}{K} - \omega_2 \geq \omega_1 \quad \Rightarrow \quad K \leq \frac{2\pi}{\omega_1 + \omega_2}$$

Centre of transition band must be $\leq$ intermediate Nyquist freq, $\frac{\pi}{K}$

We must add a lowpass filter to remove the images:



Passband noise = noise floor at output of $H(z)$ plus $10 \log_{10} (K - 1)$ dB.

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R - 1$.

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R-1$. For convenience, assume $M+1$ is a multiple of K (else zero-pad $h[n]$).

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R-1$. For convenience, assume $M+1$ is a multiple of K (else zero-pad $h[n]$).

Example: $M = 399$, $K = 50$

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R - 1$. For convenience, assume $M + 1$ is a multiple of K (else zero-pad $h[n]$).

Example: $M = 399$, $K = 50 \Rightarrow R = \frac{M+1}{K} = 8$

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R-1$. For convenience, assume $M+1$ is a multiple of K (else zero-pad $h[n]$).

Example: $M = 399$, $K = 50 \Rightarrow R = \frac{M+1}{K} = 8$

$$H(z) = \sum_{m=0}^{M} h[m]z^{-m}$$

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R-1$. For convenience, assume $M+1$ is a multiple of K (else zero-pad $h[n]$).

Example: $M = 399$, $K = 50 \Rightarrow R = \frac{M+1}{K} = 8$

$$H(z) = \sum_{m=0}^{M} h[m]z^{-m}$$

$$= \sum_{m=0}^{K-1} h[m]z^{-m} + \sum_{m=0}^{K-1} h[m+K]z^{-(m+K)} + \cdots \qquad [R \text{ terms}]$$

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R-1$. For convenience, assume $M+1$ is a multiple of K (else zero-pad $h[n]$).

Example: $M = 399$, $K = 50 \Rightarrow R = \frac{M+1}{K} = 8$

$$H(z) = \sum_{m=0}^{M} h[m]z^{-m}$$

$$= \sum_{m=0}^{K-1} h[m]z^{-m} + \sum_{m=0}^{K-1} h[m+K]z^{-(m+K)} + \cdots \qquad [R \text{ terms}]$$

$$= \sum_{r=0}^{R-1} \sum_{m=0}^{K-1} h[m+Kr]z^{-m-Kr}$$

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R-1$. For convenience, assume $M+1$ is a multiple of K (else zero-pad $h[n]$).

Example: $M = 399$, $K = 50 \Rightarrow R = \frac{M+1}{K} = 8$

$$H(z) = \sum_{m=0}^{M} h[m] z^{-m}$$

$$= \sum_{m=0}^{K-1} h[m] z^{-m} + \sum_{m=0}^{K-1} h[m+K] z^{-(m+K)} + \cdots \qquad [R \text{ terms}]$$

$$= \sum_{r=0}^{R-1} \sum_{m=0}^{K-1} h[m+Kr] z^{-m-Kr}$$

$$= \sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r] z^{-Kr}$$

where $h_m[r] = h[m+Kr]$

$$= \sum_{m=0}^{K-1} z^{-m} H_m\left(z^K\right)$$

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R-1$. For convenience, assume $M+1$ is a multiple of K (else zero-pad $h[n]$).

Example: $M = 399$, $K = 50 \Rightarrow R = \frac{M+1}{K} = 8$

$$H(z) = \sum_{m=0}^{M} h[m] z^{-m}$$

$$= \sum_{m=0}^{K-1} h[m] z^{-m} + \sum_{m=0}^{K-1} h[m+K] z^{-(m+K)} + \cdots \quad [R \text{ terms}]$$

$$= \sum_{r=0}^{R-1} \sum_{m=0}^{K-1} h[m+Kr] z^{-m-Kr}$$

$$= \sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r] z^{-Kr}$$

where $h_m[r] = h[m+Kr]$

$$= \sum_{m=0}^{K-1} z^{-m} H_m\left(z^K\right)$$

Example: $M = 399$, $K = 50$, $R = 8$

$$h_3[r] = [h[3],\ h[53],\ \cdots,\ h[303],\ h[353]]$$

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R-1$. For convenience, assume $M+1$ is a multiple of K (else zero-pad $h[n]$).

Example: $M = 399$, $K = 50 \Rightarrow R = \frac{M+1}{K} = 8$

$$H(z) = \sum_{m=0}^{M} h[m]z^{-m}$$

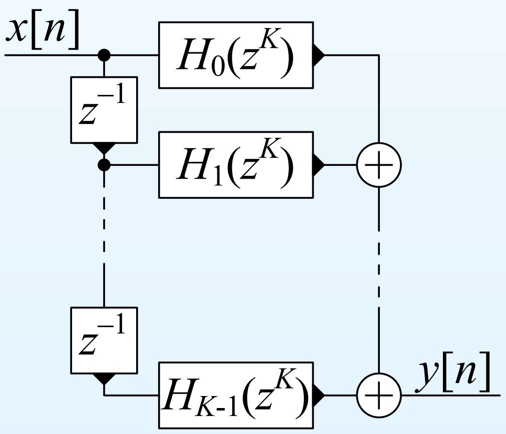
$$= \sum_{m=0}^{K-1} h[m]z^{-m} + \sum_{m=0}^{K-1} h[m+K]z^{-(m+K)} + \cdots \quad [R \text{ terms}]$$

$$= \sum_{r=0}^{R-1} \sum_{m=0}^{K-1} h[m+Kr]z^{-m-Kr}$$

$$= \sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r]z^{-Kr}$$

where $h_m[r] = h[m+Kr]$

$$= \sum_{m=0}^{K-1} z^{-m} H_m\left(z^K\right)$$



Example: $M = 399$, $K = 50$, $R = 8$

$h_3[r] = [h[3],\ h[53],\ \cdots,\ h[303],\ h[353]]$

Polyphase decomposition

For our filter: original Nyquist frequency = 10 kHz and transition band centre is at 200 Hz so we can use $K = 50$.

We will split $H(z)$ into K filters each of order $R-1$. For convenience, assume $M+1$ is a multiple of K (else zero-pad $h[n]$).

Example: $M = 399$, $K = 50 \Rightarrow R = \frac{M+1}{K} = 8$

$$H(z) = \sum_{m=0}^{M} h[m]z^{-m}$$

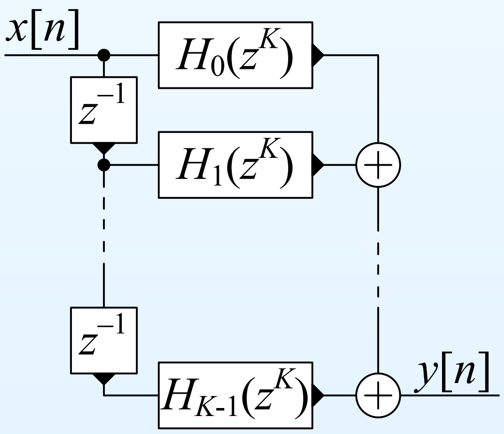
$$= \sum_{m=0}^{K-1} h[m]z^{-m} + \sum_{m=0}^{K-1} h[m+K]z^{-(m+K)} + \cdots \quad [R \text{ terms}]$$

$$= \sum_{r=0}^{R-1} \sum_{m=0}^{K-1} h[m+Kr]z^{-m-Kr}$$

$$= \sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r]z^{-Kr}$$

where $h_m[r] = h[m+Kr]$

$$= \sum_{m=0}^{K-1} z^{-m} H_m\left(z^K\right)$$



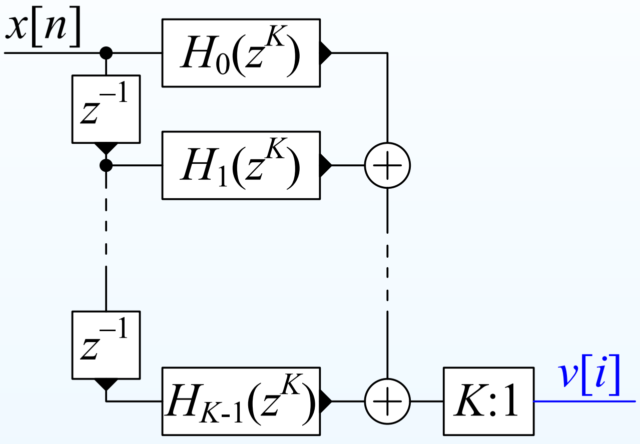
Example: $M = 399$, $K = 50$, $R = 8$

$h_3[r] = [h[3],\ h[53],\ \cdots,\ h[303],\ h[353]]$

This is a polyphase implementation of the filter $H(z)$

Downsampled Polyphase Filter

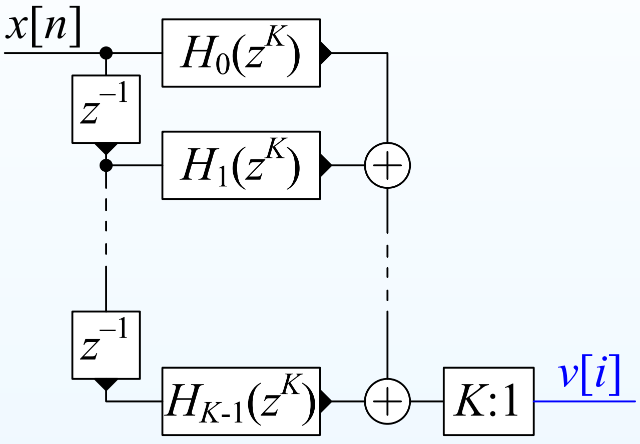
$H(z)$ is low pass so we downsample its output by K without aliasing.



Downsampled Polyphase Filter

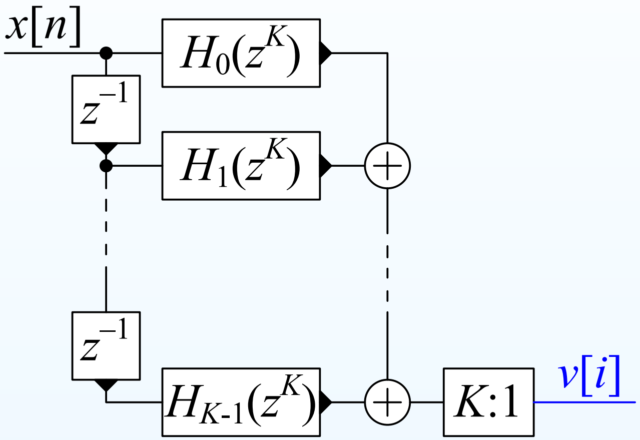
$H(z)$ is low pass so we downsample its output by K without aliasing.

The number of multiplications per input sample is $M + 1 = 400$.



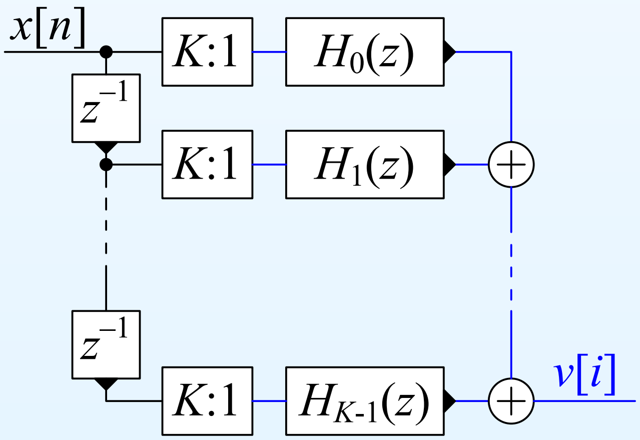
Downsampled Polyphase Filter

$H(z)$ is low pass so we downsample its output by K without aliasing.



The number of multiplications per input sample is $M + 1 = 400$.

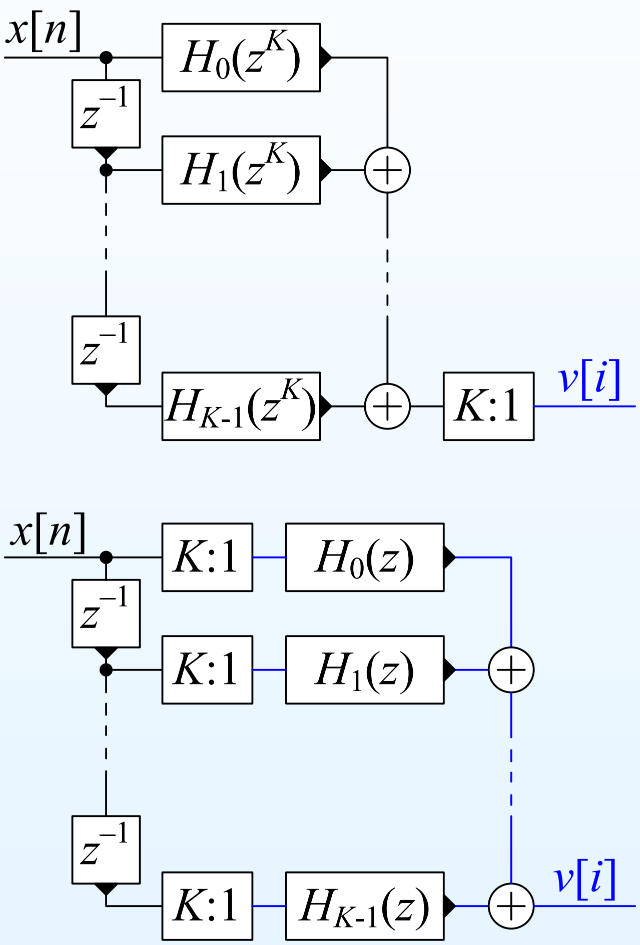
Using the Noble identities, we can move the resampling back through the adders and filters. $H_m(z^K)$ turns into $H_m(z)$ at a lower sample rate.



Downsampled Polyphase Filter

$H(z)$ is low pass so we downsample its output by K without aliasing.

The number of multiplications per input sample is $M + 1 = 400$.



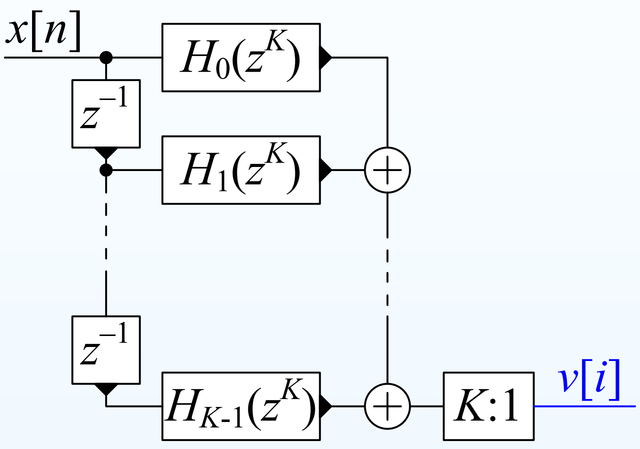
Using the Noble identities, we can move the resampling back through the adders and filters. $H_m(z^K)$ turns into $H_m(z)$ at a lower sample rate.

We still perform 400 multiplications but now only once for every K input samples.

Downsampled Polyphase Filter

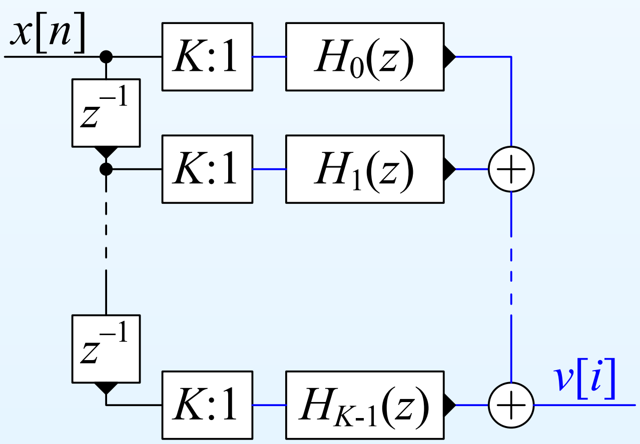
$H(z)$ is low pass so we downsample its output by K without aliasing.

The number of multiplications per input sample is $M+1=400$.



Using the Noble identities, we can move the resampling back through the adders and filters. $H_m(z^K)$ turns into $H_m(z)$ at a lower sample rate.

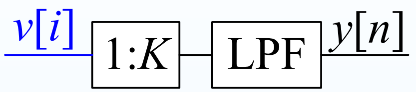
We still perform 400 multiplications but now only once for every K input samples.



Multiplications per input sample = 8 (down by a factor of 50 ☺) but $v[n]$ has the wrong sample rate (☹).

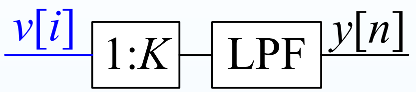
Polyphase Upsampler

To restore sample rate: upsample and then lowpass filter to remove images



Polyphase Upsampler

To restore sample rate: upsample and then lowpass filter to remove images



We can use the same lowpass filter, $H(z)$, in polyphase form:

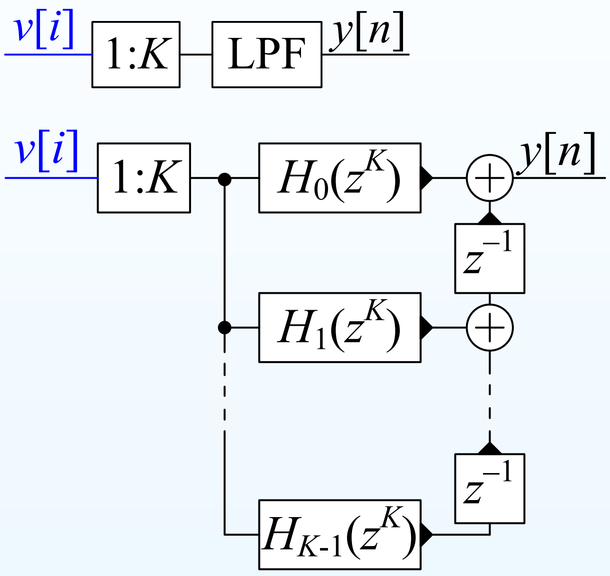
$$\sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r] z^{-Kr}$$

Polyphase Upsampler

To restore sample rate: upsample and then lowpass filter to remove images

We can use the same lowpass filter, $H(z)$, in polyphase form:

$$\sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r] z^{-Kr}$$



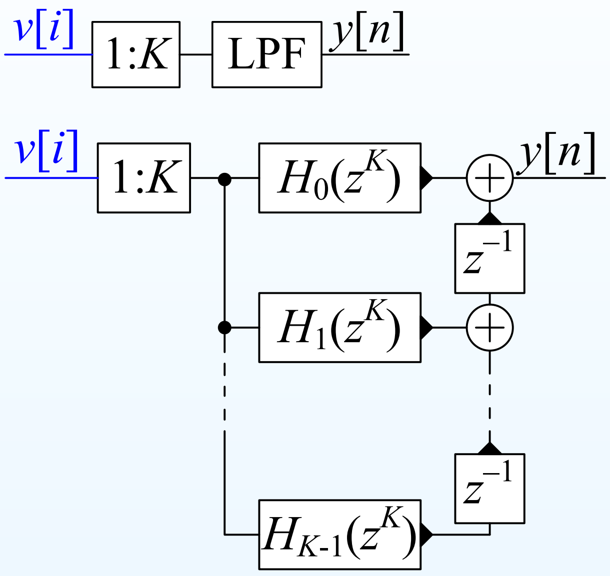
Polyphase Upsampler

To restore sample rate: upsample and then lowpass filter to remove images

We can use the same lowpass filter, $H(z)$, in polyphase form:

$$\sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r] z^{-Kr}$$

This time we put the delay z^{-m} after the filters.



Polyphase Upsampler

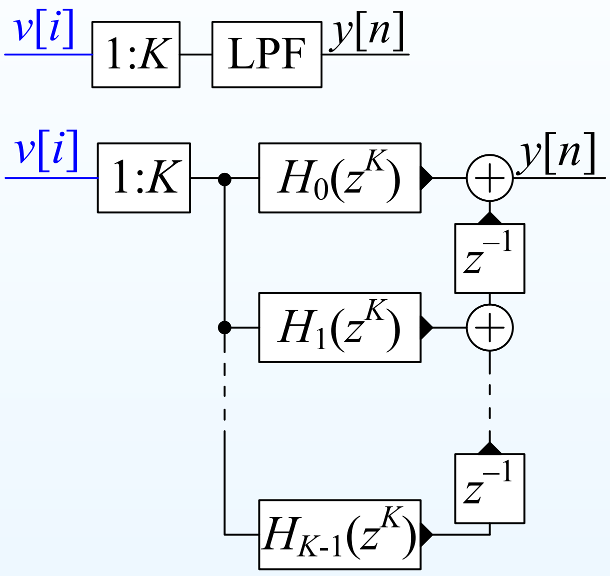
To restore sample rate: upsample and then lowpass filter to remove images

We can use the same lowpass filter, $H(z)$, in polyphase form:

$$\sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r] z^{-Kr}$$

This time we put the delay z^{-m} after the filters.

Multiplications per output sample = 400



Polyphase Upsampler

To restore sample rate: upsample and then lowpass filter to remove images

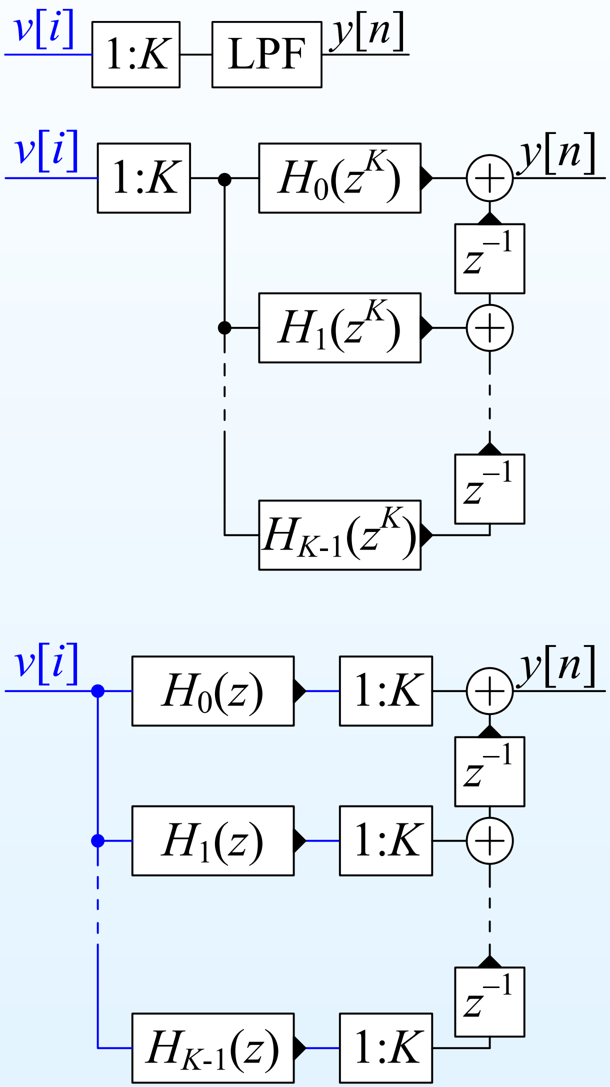
We can use the same lowpass filter, $H(z)$, in polyphase form:

$$\sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r] z^{-Kr}$$

This time we put the delay z^{-m} after the filters.

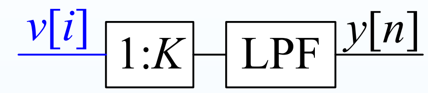
Multiplications per output sample = 400

Using the Noble identities, we can move the resampling forwards through the filters. $H_m(z^K)$ turns into $H_m(z)$ at a lower sample rate.



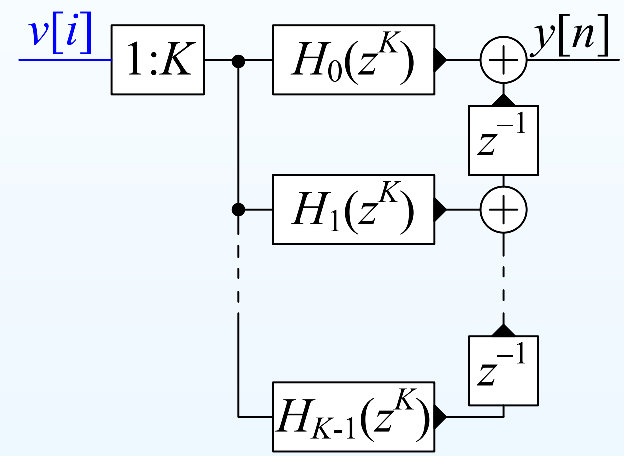
Polyphase Upsampler

To restore sample rate: upsample and then lowpass filter to remove images



We can use the same lowpass filter, $H(z)$, in polyphase form:

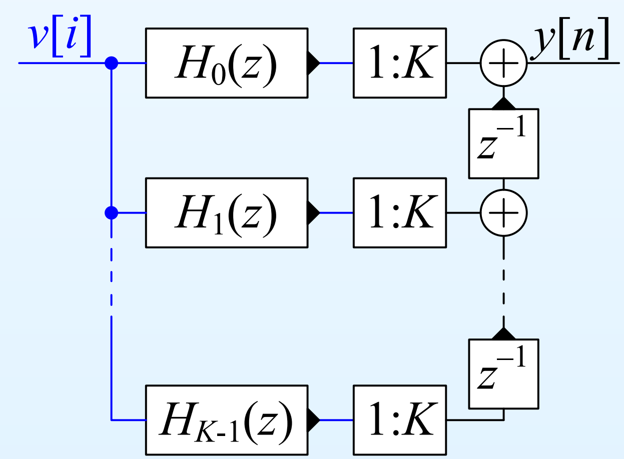
$$\sum_{m=0}^{K-1} z^{-m} \sum_{r=0}^{R-1} h_m[r] z^{-Kr}$$



This time we put the delay z^{-m} after the filters.

Multiplications per output sample = 400

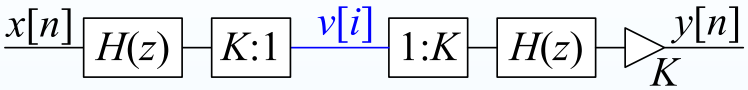
Using the Noble identities, we can move the resampling forwards through the filters. $H_m(z^K)$ turns into $H_m(z)$ at a lower sample rate.



Multiplications per output sample = 8 (down by a factor of 50 ☺).

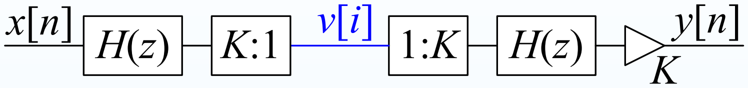
Complete Filter

The overall system implements:



Complete Filter

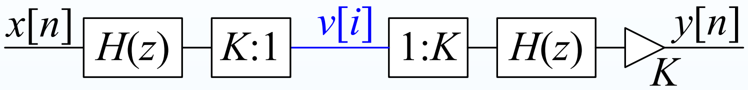
The overall system implements:



Need an extra gain of K to compensate for the downsampling energy loss.

Complete Filter

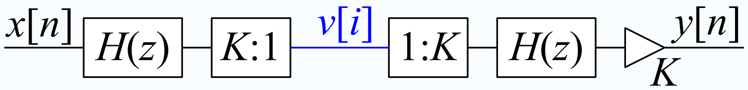
The overall system implements:



Need an extra gain of K to compensate for the downsampling energy loss.

Filtering at downsampled rate requires 16 multiplications per input sample (8 for each filter). Reduced by $\frac{K}{2}$ from the original 400.

Complete Filter

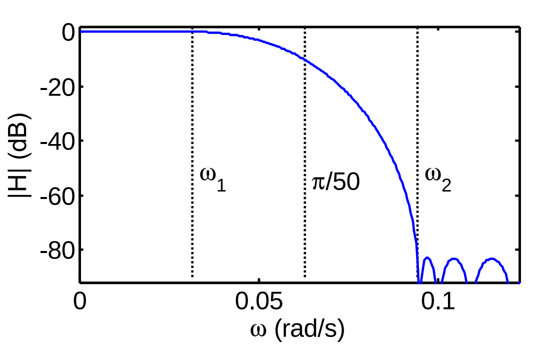


The overall system implements:

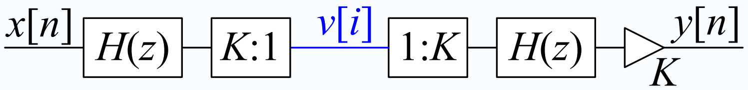
Need an extra gain of K to compensate for the downsampling energy loss.

Filtering at downsampled rate requires 16 multiplications per input sample (8 for each filter). Reduced by $\frac{K}{2}$ from the original 400.

$H(e^{j\omega})$ reaches -10 dB at the downsampler Nyquist frequency of $\frac{\pi}{K}$.



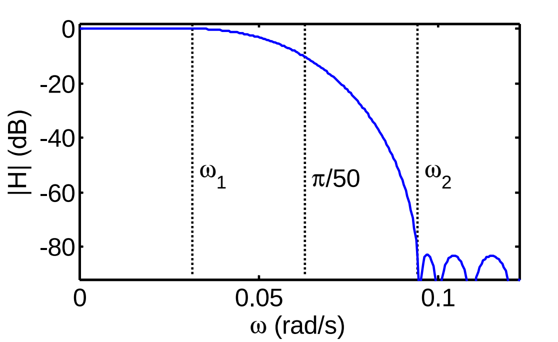
Complete Filter



The overall system implements:
Need an extra gain of K to compensate for the downsampling energy loss.

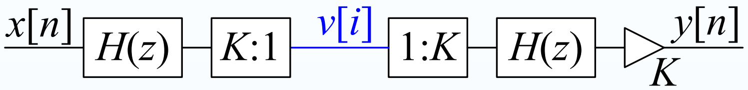
Filtering at downsampled rate requires 16 multiplications per input sample (8 for each filter). Reduced by $\frac{K}{2}$ from the original 400.

$H(e^{j\omega})$ reaches -10 dB at the downsampler Nyquist frequency of $\frac{\pi}{K}$.
Spectral components $> \frac{\pi}{K}$ will be aliased down in frequency in $V(e^{j\omega})$.



Complete Filter

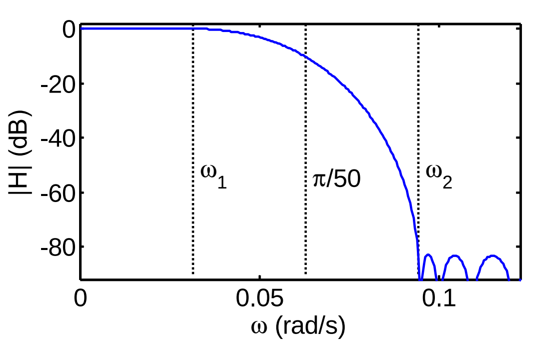
The overall system implements:



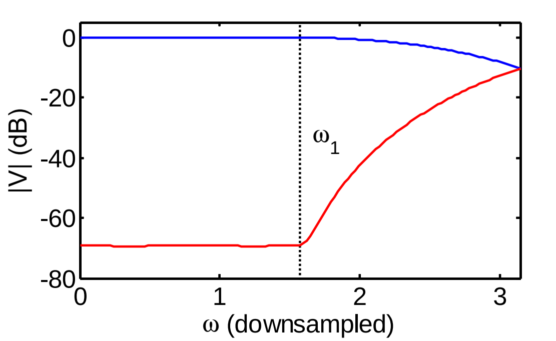
Need an extra gain of K to compensate for the downsampling energy loss.

Filtering at downsampled rate requires 16 multiplications per input sample (8 for each filter). Reduced by $\frac{K}{2}$ from the original 400.

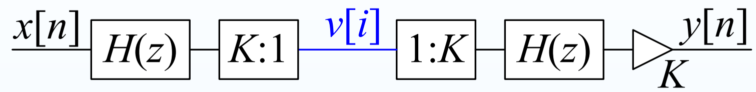
$H(e^{j\omega})$ reaches -10 dB at the downsampler Nyquist frequency of $\frac{\pi}{K}$. Spectral components $> \frac{\pi}{K}$ will be aliased down in frequency in $V(e^{j\omega})$.



For $V(e^{j\omega})$, passband gain (blue curve) follows the same curve as $X(e^{j\omega})$.



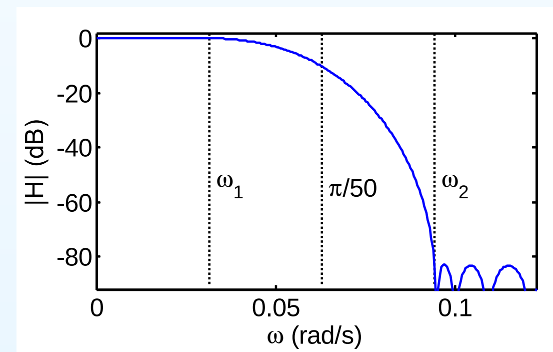
Complete Filter



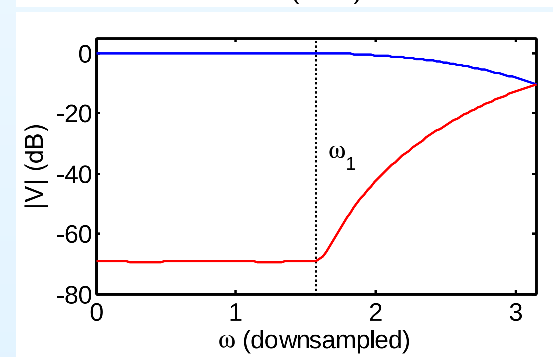
The overall system implements:
Need an extra gain of K to compensate for the downsampling energy loss.

Filtering at downsampled rate requires 16 multiplications per input sample (8 for each filter). Reduced by $\frac{K}{2}$ from the original 400.

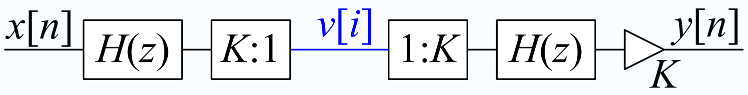
$H(e^{j\omega})$ reaches -10 dB at the downsampler Nyquist frequency of $\frac{\pi}{K}$. Spectral components $> \frac{\pi}{K}$ will be aliased down in frequency in $V(e^{j\omega})$.



For $V(e^{j\omega})$, passband gain (blue curve) follows the same curve as $X(e^{j\omega})$.
Noise arises from K aliased spectral intervals.



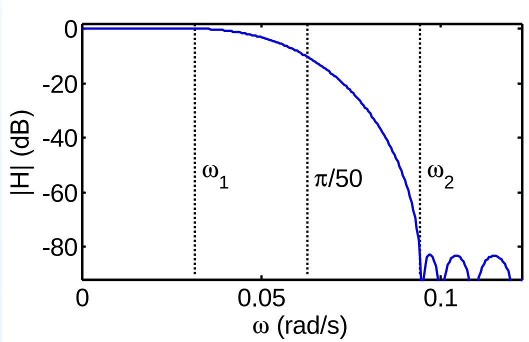
Complete Filter



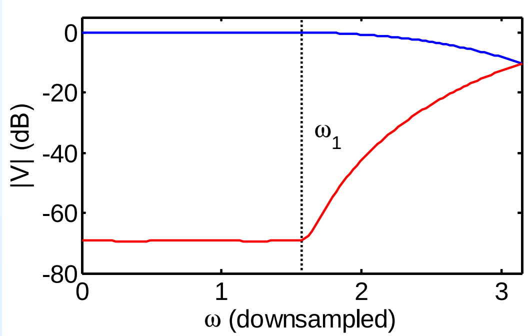
The overall system implements:
Need an extra gain of K to compensate for the downsampling energy loss.

Filtering at downsampled rate requires 16 multiplications per input sample (8 for each filter). Reduced by $\frac{K}{2}$ from the original 400.

$H(e^{j\omega})$ reaches -10 dB at the downsampler Nyquist frequency of $\frac{\pi}{K}$.
Spectral components $> \frac{\pi}{K}$ will be aliased down in frequency in $V(e^{j\omega})$.



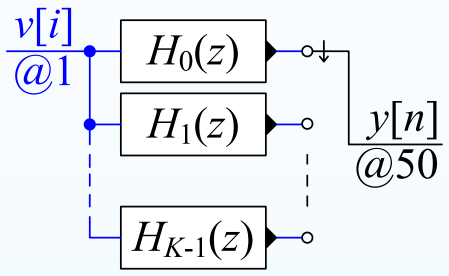
For $V(e^{j\omega})$, passband gain (blue curve) follows the same curve as $X(e^{j\omega})$.
Noise arises from K aliased spectral intervals.
Unit white noise in $X(e^{j\omega})$ gives passband noise floor at -69 dB (red curve) even though stop band ripple is below -83 dB (due to $K-1$ aliased stopband copies).



Upsampler Implementation

We can represent the upsampler compactly using a commutator. Sample $y[n]$ comes from $H_k(z)$ where $k = n \mod K$.

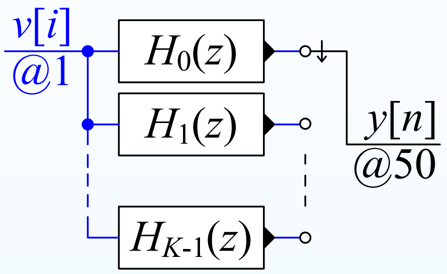
["$@f$" indicates the sample rate]



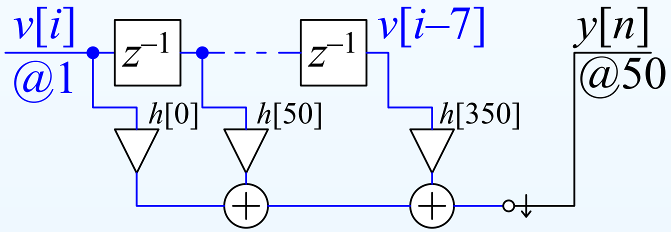
Upsampler Implementation

We can represent the upsampler compactly using a commutator. Sample $y[n]$ comes from $H_k(z)$ where $k = n \mod K$.

["$@f$" indicates the sample rate]



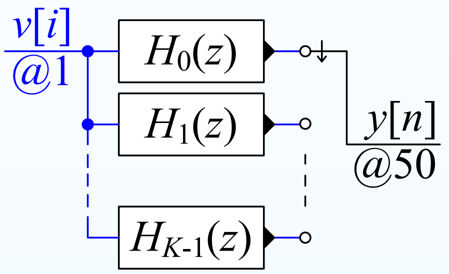
$H_0(z)$ comprises a sequence of 7 delays, 7 adders and 8 gains.



Upsampler Implementation

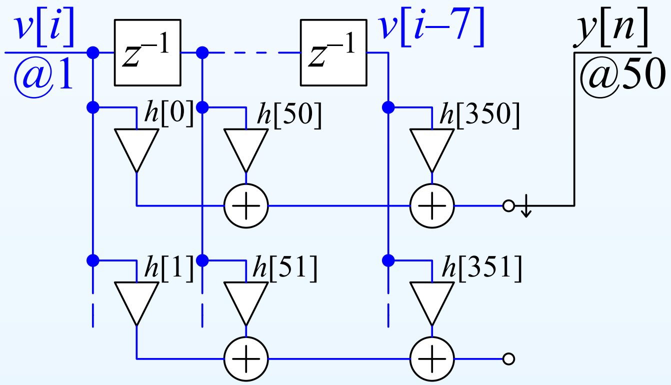
We can represent the upsampler compactly using a commutator. Sample $y[n]$ comes from $H_k(z)$ where $k = n \mod K$.

["$@f$" indicates the sample rate]



$H_0(z)$ comprises a sequence of 7 delays, 7 adders and 8 gains.

We can share the delays between all 50 filters.



Upsampler Implementation

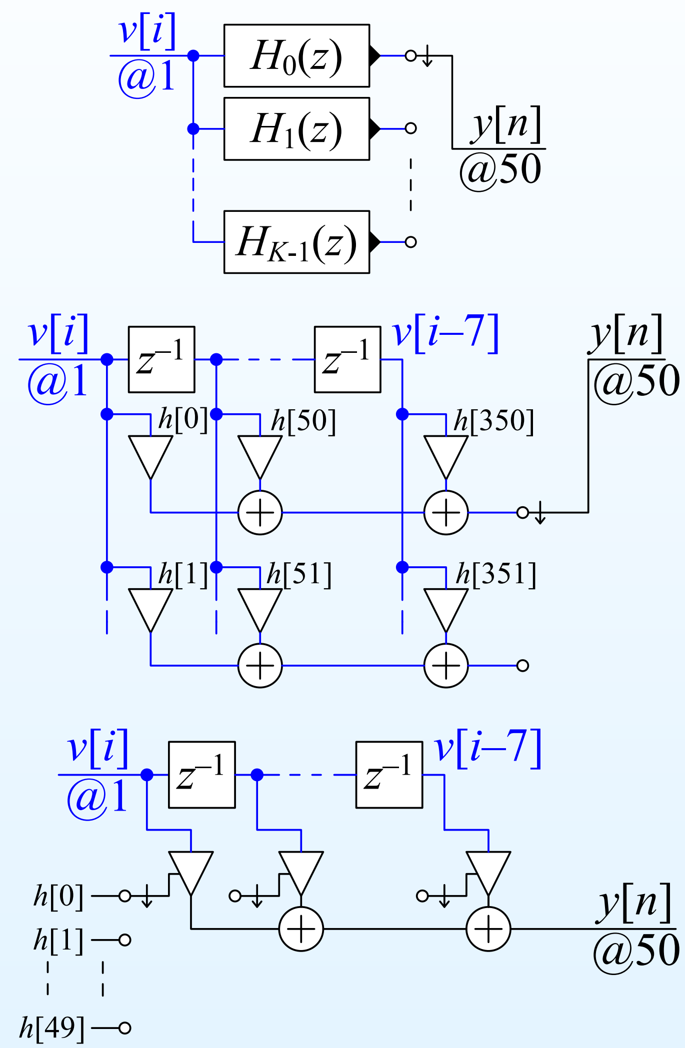
We can represent the upsampler compactly using a commutator. Sample $y[n]$ comes from $H_k(z)$ where $k = n \mod K$.

["$@f$" indicates the sample rate]

$H_0(z)$ comprises a sequence of 7 delays, 7 adders and 8 gains.

We can share the delays between all 50 filters.

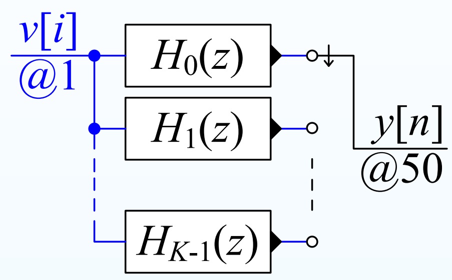
We can also share the gains and adders between all 50 filters and use commutators to switch the coefficients.



Upsampler Implementation

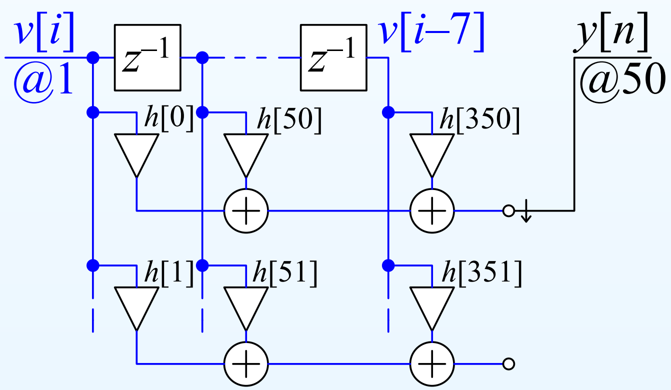
We can represent the upsampler compactly using a commutator. Sample $y[n]$ comes from $H_k(z)$ where $k = n \mod K$.

["$@f$" indicates the sample rate]

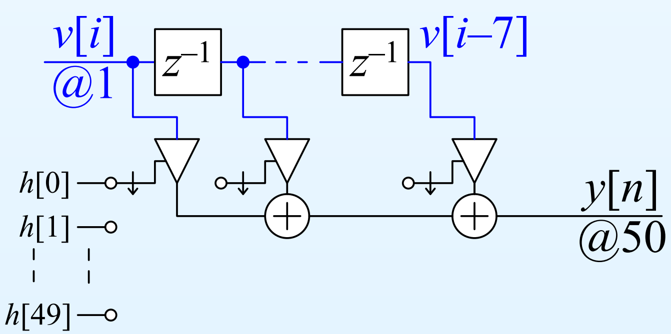


$H_0(z)$ comprises a sequence of 7 delays, 7 adders and 8 gains.

We can share the delays between all 50 filters.



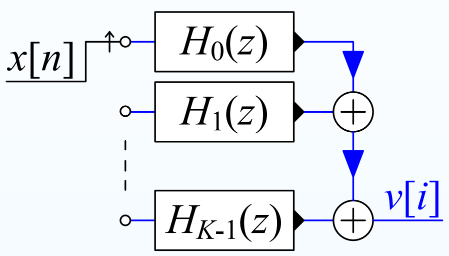
We can also share the gains and adders between all 50 filters and use commutators to switch the coefficients.



We now need 7 delays, 7 adders and 8 gains for the entire filter.

Downsampler Implementation

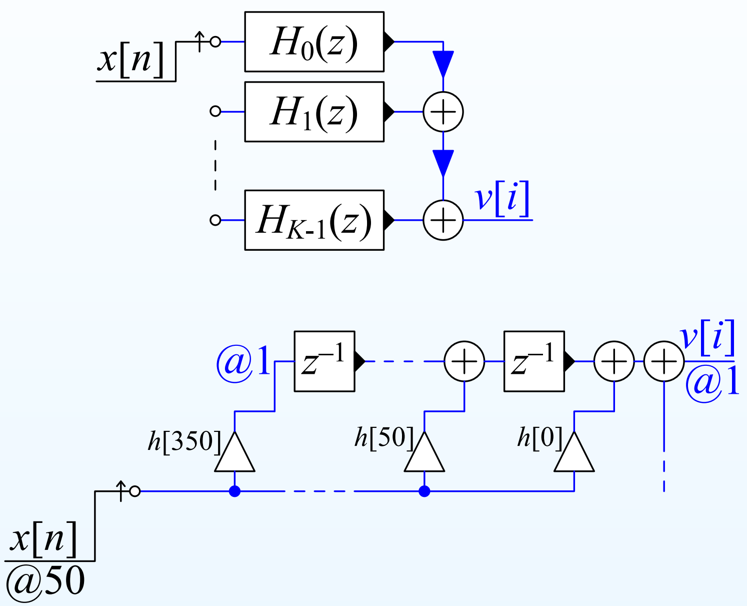
We can again use a commutator.
The outputs from all 50 filters are added together to form $v[i]$.



Downsampler Implementation

We can again use a commutator. The outputs from all 50 filters are added together to form $v[i]$.

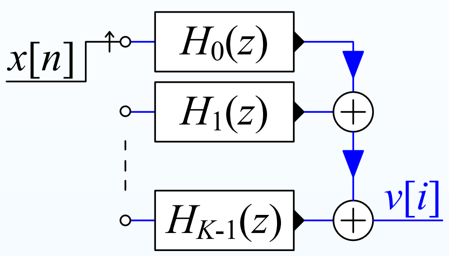
We use the transposed form of $H_m(z)$ because this will allow us to share components.

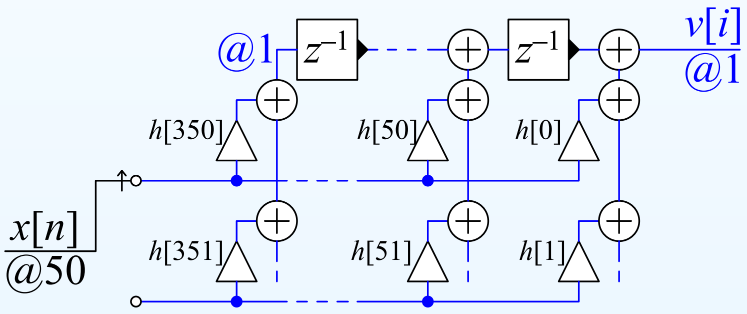


Downsampler Implementation

We can again use a commutator. The outputs from all 50 filters are added together to form $v[i]$.

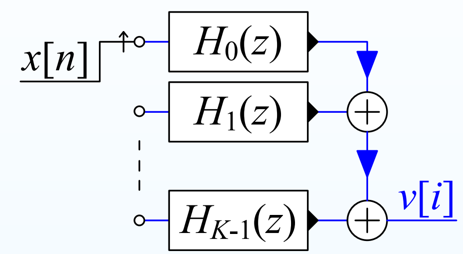
We use the transposed form of $H_m(z)$ because this will allow us to share components.



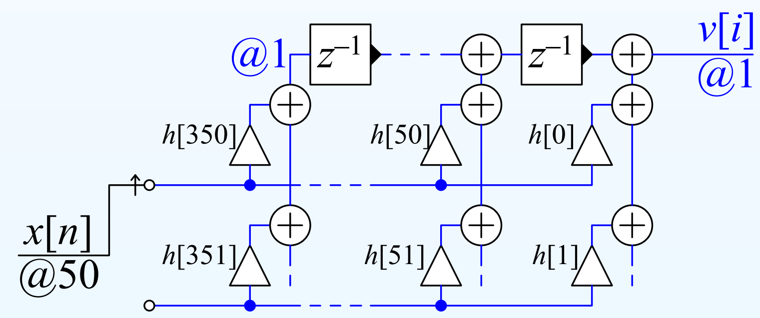


Downsampler Implementation

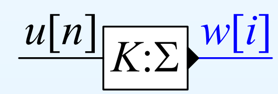
We can again use a commutator. The outputs from all 50 filters are added together to form $v[i]$.



We use the transposed form of $H_m(z)$ because this will allow us to share components.



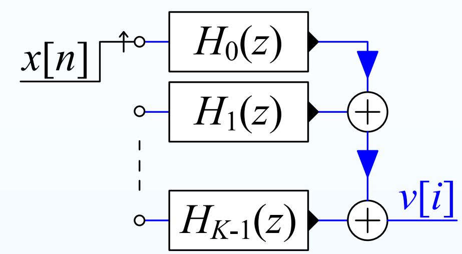
We can sum the outputs of the gain elements using an accumulator which sums blocks of K samples.



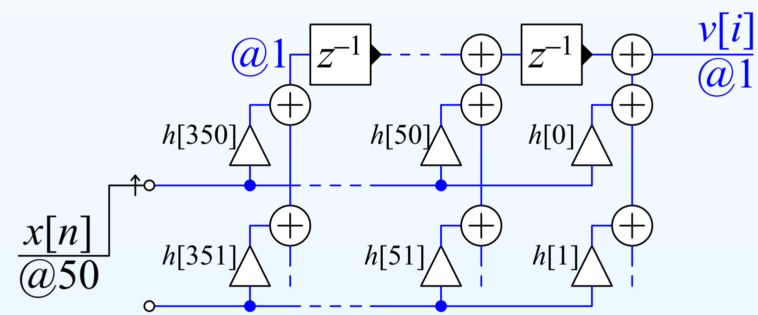
$$w[i] = \sum_{r=0}^{K-1} u[Ki - r]$$

Downsampler Implementation

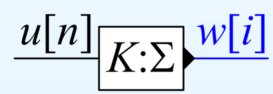
We can again use a commutator. The outputs from all 50 filters are added together to form $v[i]$.



We use the transposed form of $H_m(z)$ because this will allow us to share components.

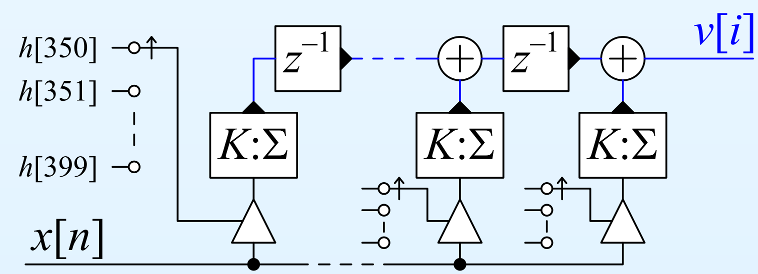


We can sum the outputs of the gain elements using an accumulator which sums blocks of K samples.



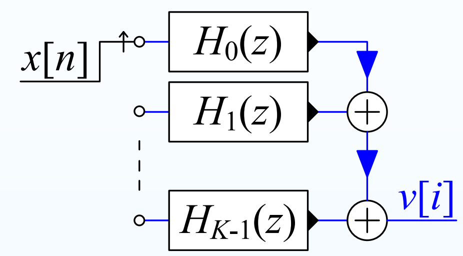
$w[i] = \sum_{r=0}^{K-1} u[Ki - r]$

Now we can share all the components and use commutators to switch the gain coefficients.

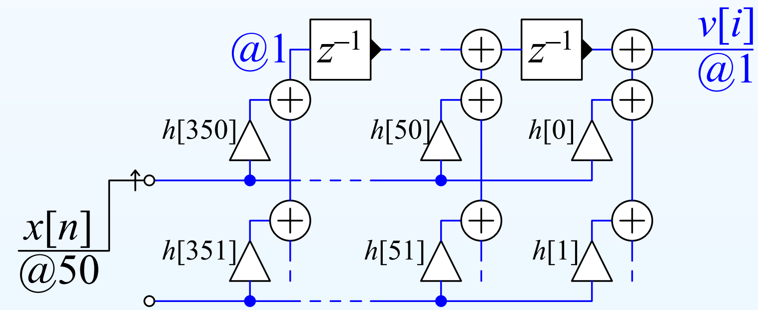


Downsampler Implementation

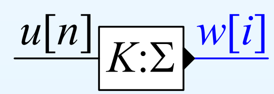
We can again use a commutator. The outputs from all 50 filters are added together to form $v[i]$.



We use the transposed form of $H_m(z)$ because this will allow us to share components.

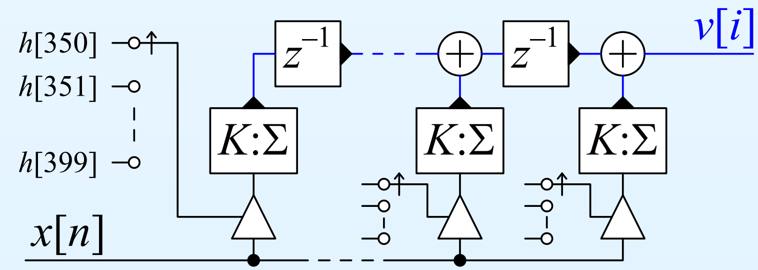


We can sum the outputs of the gain elements using an accumulator which sums blocks of K samples.



$w[i] = \sum_{r=0}^{K-1} u[Ki - r]$

Now we can share all the components and use commutators to switch the gain coefficients.



We need 7 delays, 7 adders, 8 gains and 8 accumulators in total.

Summary

- Filtering should be performed at the lowest possible sample rate
  - reduce filter computation by K
  - actual saving is only $\frac{K}{2}$ because you need a second filter
  - downsampled Nyquist frequency $\geq \max\left(\omega_{\text{passband}}\right) + \frac{\Delta\omega}{2}$

Summary

- Filtering should be performed at the lowest possible sample rate
  - reduce filter computation by K
  - actual saving is only $\frac{K}{2}$ because you need a second filter
  - downsampled Nyquist frequency $\geq \max\left(\omega_{\text{passband}}\right) + \frac{\Delta\omega}{2}$
- Polyphase decomposition: split $H(z)$ as $\sum_{m=0}^{K-1} z^{-m} H_m(z^K)$
  - each $H_m(z^K)$ can operate on subsampled data
  - combine the filtering and down/up sampling

Summary

- Filtering should be performed at the lowest possible sample rate
  - reduce filter computation by K
  - actual saving is only $\frac{K}{2}$ because you need a second filter
  - downsampled Nyquist frequency $\geq \max\left(\omega_{\text{passband}}\right) + \frac{\Delta\omega}{2}$
- Polyphase decomposition: split $H(z)$ as $\sum_{m=0}^{K-1} z^{-m} H_m(z^K)$
  - each $H_m(z^K)$ can operate on subsampled data
  - combine the filtering and down/up sampling
- Noise floor is higher because it arises from K spectral intervals that are aliased together by the downsampling.

Summary

- Filtering should be performed at the lowest possible sample rate
  - reduce filter computation by K
  - actual saving is only $\frac{K}{2}$ because you need a second filter
  - downsampled Nyquist frequency $\geq \max\left(\omega_{\text{passband}}\right) + \frac{\Delta\omega}{2}$
- Polyphase decomposition: split $H(z)$ as $\sum_{m=0}^{K-1} z^{-m} H_m(z^K)$
  - each $H_m(z^K)$ can operate on subsampled data
  - combine the filtering and down/up sampling
- Noise floor is higher because it arises from K spectral intervals that are aliased together by the downsampling.
- Share components between the K filters
  - multiplier gain coefficients switch at the original sampling rate
  - need a new component: accumulator/downsampler $(K : \Sigma)$

Summary

- Filtering should be performed at the lowest possible sample rate
  - reduce filter computation by K
  - actual saving is only $\frac{K}{2}$ because you need a second filter
  - downsampled Nyquist frequency $\geq \max\left(\omega_{\text{passband}}\right) + \frac{\Delta\omega}{2}$
- Polyphase decomposition: split $H(z)$ as $\sum_{m=0}^{K-1} z^{-m} H_m(z^K)$
  - each $H_m(z^K)$ can operate on subsampled data
  - combine the filtering and down/up sampling
- Noise floor is higher because it arises from K spectral intervals that are aliased together by the downsampling.
- Share components between the K filters
  - multiplier gain coefficients switch at the original sampling rate
  - need a new component: accumulator/downsampler $(K : \Sigma)$

For further details see Harris 5.