Multirate Filter Design - An Introduction

Jerry E. Purcell Ph.D.
President
Momentum Data Systems

The Multirate Filter Design method is used for FIR filters that have very narrow transition bands, or narrow passbands or wide passbands. These FIR filters are in general not practical to design or implement as ordinary time invariant FIR filters due to the extremely long filter lengths. The following filter types can be designed using the Multirate Filter Design Technique:

- *narrow lowpass filters*
- *narrow bandpass filters*
- *narrow highpass filters*
- *wide highpass filters*
- *wide lowpass pass filters*
- *narrow bandstop filters*

Multirate Signal Processing for Filter Design: Multirate Signal Processing consists of using different sample rates within a system to achieve computational efficiencies that are impossible to obtain with a system that operates on a single fixed sample rate.

As an example, consider the following lowpass filter:

Sampling frequency	50 kHz
Passband cutoff frequency	800 Hz
Stopband cutoff frequency	1 kHz
Maximum passband attenuation	0.1 dB
Minimum stopband attenuation	60.0dB

The filter implemented as a standard Parks-McClellan algorithm design requires 681 taps or 681 multiplies and add combinations. However, if the sampling rate was changed to 2500 Hz, the filter would require only 35 multiply and add combinations. This leads to the concept of changing the sampling rates downward (decimation) to a lower sampling rate; filtering the signal and then changing the sampling rate upward (interpolation) to the original sampling rate.

Reducing the sampling rate requires an anti-aliasing filter prior to the decimation to a lower sampling rate. Increasing the sampling rate requires an anti-imaging filter after the interpolation. The two filters are specified using the original lowpass filter specification. To achieve any gain in computational efficiency, the two filters must run at the reduced sampling rates. This paper will show how this efficiency gain can be achieved.

Returning to our example, a computational gain in efficiency of 25 to 1 can be achieved. As the passband edge approaches zero for a fixed sampling frequency, the computational efficiency approaches infinity.

DECIMATION

A reduction in the sampling rate by factor M is achieved by discarding every M-1 samples or equivalently keeping every Mth sample. While discarding M-1 of every M input samples reduces the original sample rate by a factor of M, it also causes input frequencies above one-half the decimated sample rate to be aliased into the frequency band from DC to the decimated Nyquist frequency. To mitigate this effect, the input signal must be lowpass filtered to remove frequency components from portions of the output spectrum which are required to be alias free in subsequent signal processing steps. A benefit of the decimation process is that the lowpass filter may be designed to operate at the decimated sample rate, rather than the faster input sample rate by using an FIR filter structure, and by noting that the output samples associated with the M-1 discarded sample need not be computed.

Let x(m) be the input signal, h(k), 0 <= k <= K be the coefficients of a given lowpass filter and z(m) be the output signal before decimating by factor M, then:

$$z(m) = \sum_{k=0}^{K} h(k)x(m-k) \qquad \text{(EQ 1)}$$

Now, let the output signal after the decimator be y(r) = z(rM) where the sampling rate is reduced by a factor M. Then, y(r) = z(rM) if the output signal is decimated by factor M.

$$y(r) = \sum_{k=0}^{K} h(k)x(rM-k) \qquad \text{(EQ 2)}$$

Looking carefully at this equation one can see that the filter is in effect using the downsampled signal. Thus the operations of downsampling and the lowpass filter have been embedded in such a way that the lowpass filter is operating at the reduced data rate and the average number of computations to generate one output sample is reduced by factor M.

Each output sample requires K multiply/accumulate cycles but only 1 out of M samples needs to be calculated. If the downsampling of a signal is applied after the anti-aliasing lowpass filter, the number of multiply/accumulates is K*M for every Minput samples. By embedding the downsampling into the lowpass filter, the number of multiply/accumulate cycles is reduced to just K for every M input samples. The average number of multiply/accumulates per input sample is K/M.

INTERPOLATION

An increase in sample rate (interpolation) by a factor of L is achieved by inserting L-1 uniformly spaced, zero value samples between each input sample. While adding L-1 new samples between each input sample increases the sample rate by a factor of L, it also introduces images of the input spectrum into the interpolated output spectrum at frequencies between the original Nyquist frequency and the higher interpolated Nyquist frequency. To mitigate this effect, the interpolated signal must be lowpass filtered to remove any image frequencies which will disturb subsequent signal processing steps. A benefit of the interpolation process is that the lowpass filter may be designed to operate at the input sample rate, rather than the faster output sample rate by using an FIR filter structure and by noting that the input associated with the L-1 inserted values have zero value.

Let x(n) be the original input sequence, v(n) the sequence with L-1 zeros inserted, y(n) the output sequence of the lowpass filter and let h(0), ..., h(k-1) be the coefficients of the lowpass filter, then:

$$y(n) = \sum_{k=0}^{K} h(k)v(n-k) \qquad \text{(EQ 3)}$$

However, v(n-k) = 0 unless n-k is a multiple of L, the interpolation factor. This is because L-1 zeros were inserted in the sequence x(n) to get v(n).

Again let x(n) be the input signals, and h(k) be the filter coefficients. Then the output signal y(r) has a simple formula:

$$y(r) = \sum_{n=0}^{K/L} h(r-Ln)x(n) \qquad \text{(EQ 4)}$$

For a single input sample, L output samples are created. If the anti-imaging filter is not embedded in the interpolation process, then the number of multiply/accumulate cycles for L output samples is L*K. However, taking advantage of the fact that L-1 zeros were inserted into the output stream, the anti-imaging filter has only K/L nonzero values. Hence, the number of multiply/accumulate cycles where the interpolation process is embedded into the anti-imaging filter is just K for L output samples. The average of multiply/accumulate cycles per output is K/L.

POLYPHASE FILTERS

Interpolator and decimator polyphase filters are used to implement multirate filters. The general polyphase filter approach using a combination of both upsampling and downsampling in the same filter is not used in multirate filter design.

Interpolator Only Polyphase Filters

The computational efficiency of the Interpolator filter structure can also be achieved by reducing the large FIR filter length of length K into a set of smaller filters. These smaller filters will have a length N = K/L, where K is selected to be a multiple of L. Because the interpolation process inserts L - 1 zeros between successive values of x(n), only N out of the K input values stored in the FIR filter at any one time are nonzero. At one time instant, these nonzero values coincide and are multiplied by the filter coefficients h(0), h(L), h(2L), ... , h(K - L). In the following instant, the nonzero values of the input sequence coincide and are multiplied by the filter coefficients h(1), h(L + 1), h(2L + 1), ..., h(K - L + 1), and so on. This observation leads us to define a set of smaller filters called polyphase filters, with unit sample responses:

$$p_k(n) = h(k + nL) \quad k = 0,1,...,L-1, \quad n = 0,1,...,N-1 \qquad \text{(EQ 5)}$$

where N = K/L is an integer.

The polyphase filter can also be viewed as a set of L subfilters connected to a common delay line. Ideally, the k'th subfilter will generate a forward time shift of (k/L)F_{in}, for k = 0, 1, 2, ... , L - 1, relative to the zeroth subfilter. Therefore, if the zeroth filter generates zero delay, the frequency response of the k'th subfilter is:

$$p_k(\omega) = e^{j\omega\frac{k}{L}} \qquad \text{(EQ 6)}$$

Decimator Only Polyphase Filters

By transposing the interpolator structure we obtain a commutator structure for a decimator that is based on the parallel bank of polyphase filters. The unit sample responses of the polyphase filter are now defined as:

$$p_k(n) = h(k + nM) \quad k = 0,1,...,M-1, \quad n = 0,1,...,N-1 \qquad \text{(EQ 7)}$$

where N = K/M is an integer when K is selected to be a multiple of M. The commutator rotates in a counter-clockwise direction starting with filter $p_0(n)$.

MULTIRATE FILTER DESIGNS

Multirate Filter Designs use the basic properties of decimation and interpolation in the implementation of the filter.

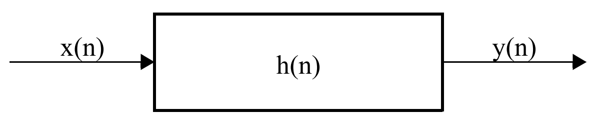
Use of Decimation and Interpolation: All multirate filter designs use the basic method of decimation to implement the desired filter and then use interpolation to restore the sampling rate back to the original rate. Using a decimator followed by an interpolator - both of which are implemented using filters running at the low data rate, can lead to significant reductions in the computational requirements of a filter compared to the direct method of convolving the filter coefficients with the incoming signal.

Modulation: Filter designs other than lowpass filters use the concept of modulation in their implementation. For example, a bandpass filter is implemented by modulating the signal to baseband, lowpass filtering the baseband signal and then modulating the baseband signal back to the center frequency of the bandpass filter

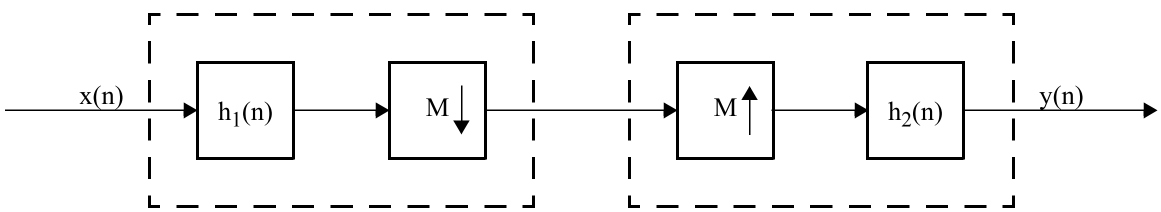
NARROW LOWPASS FILTER

A narrow lowpass filter is defined as a lowpass filter with a narrow passband. To design a multirate narrow lowpass FIR filter, the time invariant classic FIR filter is replaced with a lowpass antialiasing filter and decimator followed by an interpolator and lowpass anti-imaging filter. The decimator and interpolator make the equivalent multirate system a time varying linear phase filter.

This process can be represented in block diagram form where (model 1)



is replaced by (model 2)



In this system, the decimation factor is always is equal to the interpolation factor. Therefore, for the input and output of the model concerned, the sampling rate is NOT changed, but it is changed within the model. A so-designed lowpass filter is linear phase, but periodically time-varying.

The lowpass filter consists of two polyphase filters - one for the decimator and one for the interpolator. Each polyphase filter runs at the reduced sample rate of F_s / M where M is the decimation (interpolation) factor and F_s is the sampling rate of the original filter. The efficiency gain of this model is M/2. If M = 1 or 2, there is no efficiency gain and a regular FIR filter should be used. If M is a product of several factors, a multi-stage design and further reduction on computations are possible.

Filter response is calculated as follows: Let N be the number of stages, and $H_i(\omega)$ (i = 0, ..., N-1) the filter response for each stage, and D_i the stage factor. Let $X_i(\omega)$ be the input signal of decimation stage i (i = 0 to N-1), and $X_{i+1}(\omega)$ the output of decimation stage i and then the input for decimation stage i+1. This leads to the following formula:

$$X_{i+1}(\omega) = \sum_{n=0}^{D_i - 1} H_i\left(\frac{\omega - 2\pi n}{D_i}\right) X_i\left(\frac{\omega - 2\pi n}{D_i}\right) \quad \text{(EQ 8)}$$

Let $Y_{i+1}(\omega)$ be the input of interpolation stage i, and $Y_i(\omega)$ the output of stage i, then we have the formula for the output:

$$Y_i(\omega) = H_i(\omega) Y_{i+1}(\omega D_i) \quad \text{(EQ 9)}$$

Since $X_N(\omega) = Y_N(\omega)$, i.e. the output of the final decimation is the input to the first interpolation (note that indexing for interpolation stage is reversed in order). Let P_i be the product of D_0 to D_{i-1}, we have

$$Y_0(\omega) = \left(\prod_{i=0}^{N-1} H_i(\omega P_i)\right) X_N(\omega P_N) \quad \text{(EQ 10)}$$

If the factor of the last decimation stage = 2, we can combine the last decimation and first interpolation stage. Let M = N - 1, then we have:

$$Y_0(\omega) = \left(\prod_{i=0}^{M-1} H_i(\omega P_i)\right) H_M(\omega P_M) X_M(\omega P_M) \quad \text{(EQ 11)}$$

In either case, we have a relation $Y_0(\omega) = H(\omega)\, X_0(\omega)$. The composite filter response $H(\omega)$ is time-varying, for a given $X_0(\omega)$, we can calculate $H(\omega)$ as the plot. In particular, pick the input as the impulse response

$$X_0(\omega) = e^{-jd\omega} \quad \text{(EQ 12)}$$

where d is the impulse delay. The value d can be set in the plot control dialog box.

Note that two lowpass filters are required and since each lowpass filter is a multirate lowpass filter, the original lowpass filter must have a narrow passband to achieve any computational gain. Let B be the width of the passband and F_s the sampling frequency. To achieve gain in computational efficiency, the following must hold:

$$B < \frac{F_s}{4}$$

Narrow Lowpass Filter Implementation

Let S be the number of stages, and D_i be the decimator for stage i (i = 1, 2, ..., 8).

Case S = 1.

If $D_1 = 2$, then the filter should be implemented in the regular way. Let x(n) be the input sample, h(m) (m = 0, 1, ... L_1-1) be the FIR filter with length L_1, then output is

$$y(k) = \sum_{i=0}^{L_1 - 1} h(i)x(k-i) \qquad \text{(EQ 13)}$$

For every input, we get an output.

If $D_1 > 2$, then the following implementation achieves the best computational gain. First, calculate the output $x_1(n)$ for stage 1:

$$x_1(n) = \sum_{i=0}^{L_1 - 1} h(i)x(nD_1 - i) \qquad \text{(EQ 14)}$$

$x_1(n)$ is calculated using L_1 input samples and D_1 values of $x_1(n)$ are computed before proceeding to the interpolation part.

The interpolation part is calculated as follows: for $k = nD_1 + p$, $(0 <= p < D_1)$, and $M_1 = L_1 / D_1$ (assume that L_1 is a multiple of D_1),

$$y(k) = y(nD_1 + p) = D_1 \sum_{i=0}^{M_1 - 1} h(iD_1 + p)x_1(n-i) \qquad \text{(EQ 15)}$$

For each input sample $x_1(n)$, there are D_1 outputs (p = 0, 1, ..., D_1-1). For each p = 0, 1, ... D_1-1, we call the coefficients: h(i, p) = h(iD_1 + p) (i = 0, 1, ..., M_1-1) a poly-phase filter, and there are D_1 polyphase filters. The D_1 outputs are the results of D_1 poly-phase filters with the same input sample $x_1(n)$. So for each input sample $x_1(n)$, D_1 outputs are obtained.

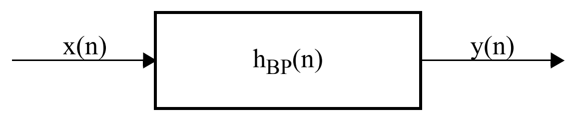
Thus, for every D_1 input samples of x(k), there is one output $x_1(n)$ for stage 1, and D_1 outputs for the interpolation stage.

There are two models for single-stage implementation, and the situation is same for multi-stage implementation. If the decimator of the last stage is 2, the last stage should be implemented as a regular filter.

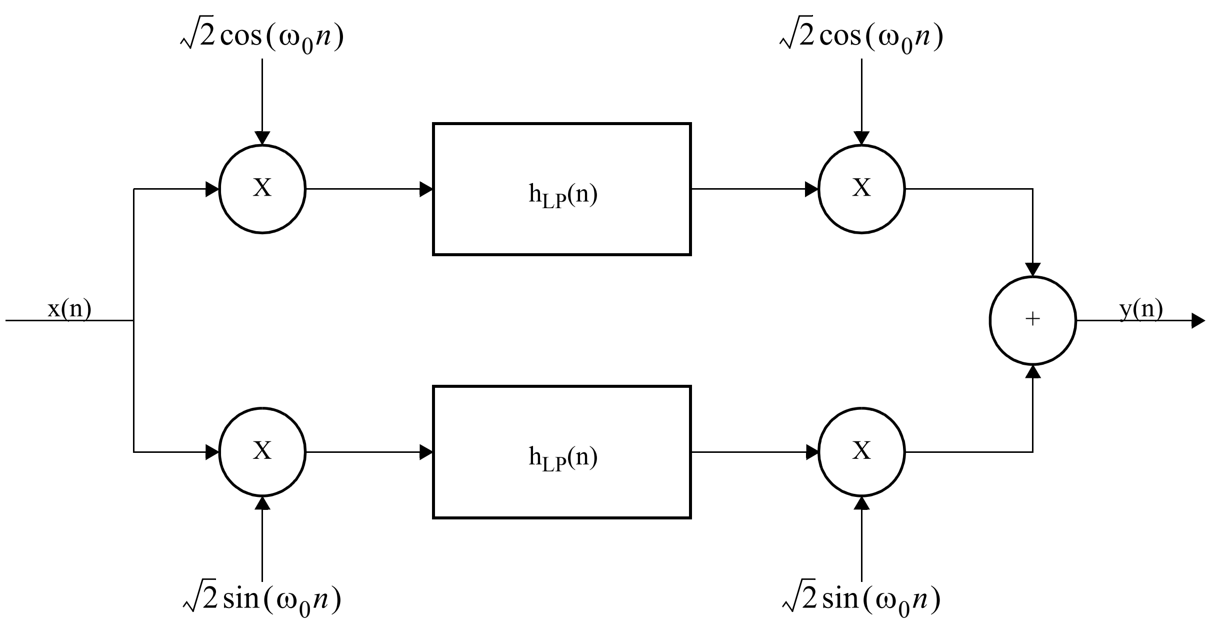
To implement a multi-stage filter, all of the decimator stages are processed in order and then the interpolator stages are processed in order. It also should be noted that the number samples processed on single call must be a multiple of product of the decimators.

Narrow Bandpass Filter

A narrow bandpass filter is defined as a bandpass filter with a narrow passband. The narrow bandpass filter is implemented using modulation techniques. Thus the following bandpass filter:



is replaced by the following equivalent functional blocks:



The desired passband of the bandpass filter is modulated so that the center of the passband denoted by ω_0 is translated to the origin. This, of course, requires both a real and imaginary part. Thus, there are two lowpass filters - one for the real part of the modulated signal and one for the imaginary part of the modulated signal. Each of the lowpass filters can be implemented as a multirate lowpass filter. After processing the two signals through the multirate lowpass filters, the filtered signals are demodulated and combined to form the resulting output signal. The demodulation of the signals and combination of the results translates the frequency spectrum back to the original position.

Thus a bandpass filter has been implemented by two multirate lowpass filters. ***This technique of modulating a signal to baseband, lowpass filtering and demodulation of the filtered signal is one of two fundamental techniques for constructing multirate filters.***

This design in general does not give a constant group delay, but it is possible to adjust the filter lengths for multi-stage lowpass filters such that the bandpass filter implemented as a result of cosine modulation has a constant group delay.

The formula for the composite filter of narrow bandpass filter is as follows:

$$Y_0(\omega) = (H(\omega - \omega_0) + H(\omega + \omega_0))X_0(\omega) \qquad \text{(EQ 16)}$$

where H(ω) is calculated as narrow lowpass filter.

Note that two lowpass filters are required and since each lowpass filter is a multirate lowpass filter, the bandpass filter must have a narrow bandwidth to achieve any computational gain. Let B be the width of the passband and F_s the sampling frequency. To achieve gain in computational efficiency, the following must hold:

$B < \frac{F_s}{4}$

Since the modulation and demodulation consists of multiplying the signals by $\cos\omega_0 nT$ or $\sin\omega_0 nT$, the choice of ω_0 can significantly affect the efficiency of the filter design.

In general, one should choose ω_0 to avoid the actual calculation of a sine or cosine but instead rely on some type of direct table lookup of the sine and cosine values based on the current value of n.

Narrow Highpass Filters

The modulation technique described for the narrow bandpass filter can also be applied to a narrow highpass design. In this case,

$$\omega_0 = 2\pi\frac{F_s}{2} \qquad \text{(EQ 17)}$$

and

$$\omega_0 T = 2\pi\frac{F_s}{2} \bullet \frac{1}{F_s} = \pi \qquad \text{(EQ 18)}$$

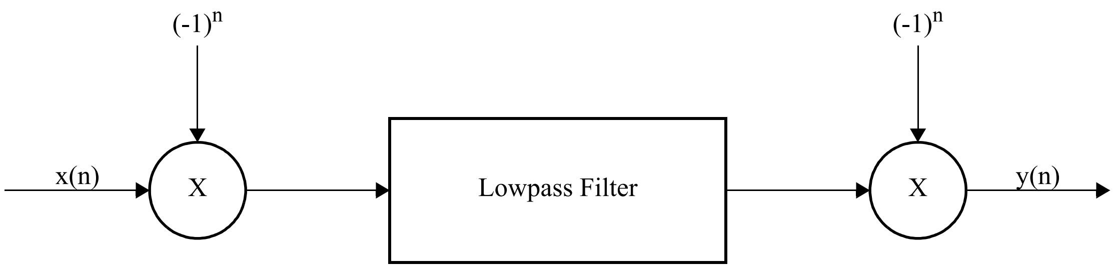
Hence the modulation multipliers become

$$\sin\omega_0 nT = \sin n\pi \quad \text{(EQ 19)}$$

$$\cos\omega_0 nT = \cos n\pi \quad \text{(EQ 20)}$$

Note that sin (nπ) = 0 for all n since n is an integer. Thus the imaginary branch in the bandpass design is eliminated. Note: cos (nπ) = $(-1)^n$ which is just alternating +1,-1. So for the narrow highpass filter, the modulation and demodulation is reduced to multiplying by 1 and -1 alternatively on the real branch of a bandpass design.

This means that a time invariant highpass filter is replaced by the following equivalent functional block:



The formula for narrow highpass filter is:

$$Y_0(\omega) = H(\pi - \omega)X_0(\omega) \quad \text{(EQ 21)}$$

where H(ω) is the response of narrow lowpass filter.

Again, the lowpass filter is a multirate filter. For any gain in computational efficiency the passband width B of the wide lowpass filter must satisfy the following requirement:

$$B > \frac{F_s}{4}$$

Narrow Highpass Filter Implementation

The filter is implemented in a similar way to the case of narrow lowpass. Every other input sample sign is changed before the input to first stage, and every other output sample sign must be changed.

In term of transfer function, if H(ω) is the transfer function for lowpass filter, then the transfer function for highpass filter is H(π−ω). Instead of changing the sign of input and output samples, it is possible to change the sign of filter coefficients, and implement the filter (after the coefficients are modified) exactly the same as lowpass filter. The modification of coefficients of filter can be made as follows:

$$w(i) = ch(i)$$

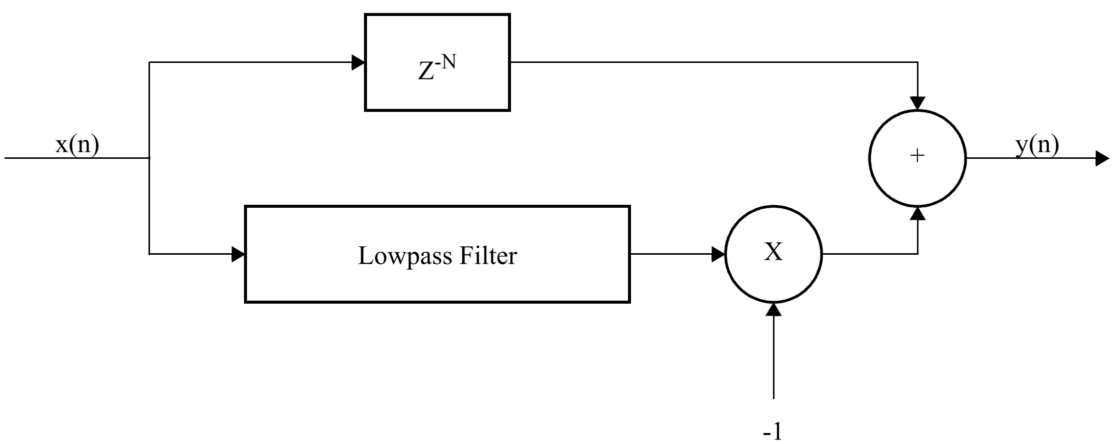
where h(i) is the original coefficient, and w(i) is the modified coefficient, and the value c is 1.0 or -1.0. c = -1.0 only in following case: j = u * i is odd, where u is determined as follows: for stage 1, u = 1; for stage 2, u = D_1 (the first decimator), for stage n, the value u is the product of all decimators prior to the current stage. If u is even, then j is always even, and c is always 1.0, then the filter coefficients are not changed. Every other coefficient of the filter for stage 1 always has the sign change.

Wide Highpass Filters

Wide highpass filters can be implemented using difference techniques. Let $H_{NLP}(z)$ be the transfer function of a narrow lowpass filter. Then the transfer function of a wideband highpass filter is $H_{WHP}(z) = 1-H_{NLP}(z)$.

To implement a wide highpass filter, the output of a narrow lowpass filter is subtracted from the delayed original signal.

Therefore, the wide highpass filter is implemented as follows:



The delay z^{-N} is very important. N is selected to be exactly one half of the filter length of the composite lowpass filter. If the delay is not implemented to be exactly half the filter length, the frequency response will be affected and the desired highpass filter will not be achieved. The formula for the delay in multi-stage design is given in the section on Multi-Stage Filter Design.

This concept of forming differences of transfer functions to obtain desired transfer functions is the second fundamental technique for constructing multirate filters.

The response of wide highpass filter is calculated as follows:

$$Y_0(\omega) = (e^{-jN\omega} - H(\omega))X_0(\omega) \quad \text{(EQ 22)}$$

where H(ω) is the response of narrow lowpass filter, N is the group delay of narrow lowpass filter H(ω). It is possible to have a half-sample delay for filter H(ω), and in this case we can not have a pure delay term. Filter lengths should be adjusted to avoid the half-sample delay.

Again, the lowpass filter is a multirate filter. For any gain in computational efficiency the passband width B of the wide lowpass filter must satisfy the following requirement:

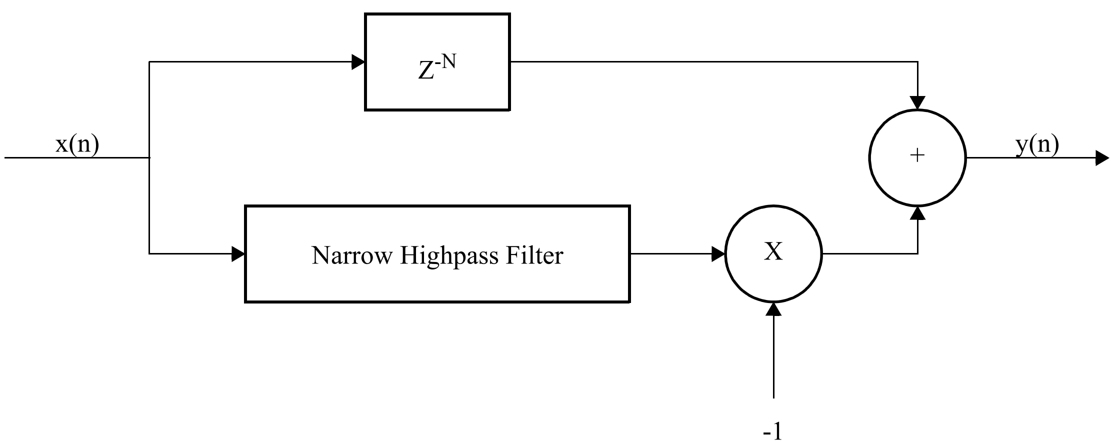
$$B > \frac{F_s}{4}$$

Wide Highpass Filter Implementation

The wide highpass filter is implemented by subtracting the result of a narrow lowpass filter from a delayed input. Since the narrow lowpass filter has a delay, the input samples must be saved in a delay buffer, the size of buffer is least equal to the delay of narrow lowpass filter. The integer delay is necessary for this structure. The narrow lowpass filter is designed and implemented as a multi-rate filter.

Wide Lowpass Filters

Let $H_{NHP}(z)$ be the transfer function of a narrow highpass filter. Then the transfer function of a wide lowpass filter is $H_{WLP}(z) = 1 - H_{NHP}(z)$. Thus to implement a wideband lowpass filter, the output of a narrow highpass filter is subtracted from the delayed original signal. This is shown in the following diagram:



N is the delay of NHP. If the narrow highpass filter is implemented as the cosine modulation of narrow lowpass filter, then the N is the delay of the NLP.

The response for wide lowpass filter is calculated as follows:

$$Y_0(\omega) = (e^{-jN(\pi-\omega)} - H(\pi-\omega))X_0(\omega) \qquad \text{(EQ 23)}$$

where H(ω) is the narrow lowpass filter response, and N is the group delay of narrow lowpass filter H(ω). It is possible to have a half-sample delay for filter H(ω), and in this case we can not have a pure delay term. The system will adjust the filter length to avoid the half-sample delay problem.

Note that the lowpass filter is a multirate lowpass filter. For any gain in computational efficiency the passband width B of the wide lowpass filter must satisfy the following requirement:

$$B > \frac{F_s}{4}$$

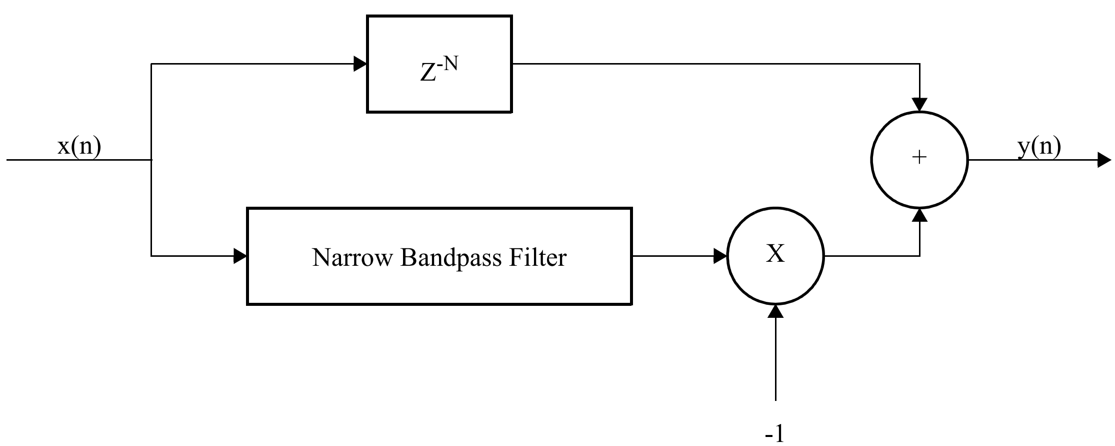
Wide Lowpass Filter Implementation

The filter is implemented by subtracting the result of a narrow highpass filter from a delayed input. A delay buffer equal to the delay of the narrow highpass filter is required to store the delayed input samples.

Narrow Bandstop Filters

Let $H_{NBP}(z)$ be the transfer function of a narrow bandpass filter and $H_{NBS}(z)$ the transfer function of the narrow bandstop filter. If $H_{NBP}(z)$ has a constant group delay N, then the transfer function of a narrow bandstop filter is $H_{NBS}(z) = 1 - H_{NBP}(z)$. This filter is called a narrow bandstop due to the narrow stopband region. Note, this terminology is different from the other filters where the adjective narrow or wide refers to the passband region.

To implement a narrow bandstop filter, the output of a narrow bandpass filter is subtracted from the original delayed signal. This is shown in the following diagram:



In order for this model to work, the narrow bandpass filter must have a constant group delay. In general, a bandpass filter implemented using cosine modulation of a lowpass filter does not have a constant group delay. It is possible to adjust the filter lengths for multistage lowpass filters such that the bandpass filter has a constant group delay. This is the group delay of the lowpass filter. Filter lengths should be adjusted so that the group delay is constant.

The response of the narrow bandstop filter is calculated as follows:

$$Y_0(\omega) = (e^{-jN\omega} - H(\omega + \omega_0) - H(\omega - \omega_0))X_0(\omega) \quad \text{(EQ 24)}$$

where H(ω) is the response of narrow lowpass filter, and N is the group delay. Again the half-sample delay is not allowed in the system.

Multi-Stage Filter Design

We will consider the narrow lowpass filter only. Other filter types are converted into a narrow lowpass filter and implemented in a structure previously described. Let

$$M = \frac{\text{sample rate} \times 0.5}{\text{stopband frequency}} \quad \text{(EQ 25)}$$

M is the maximum decimator for the narrow lowpass filter. Any integer > M used as a decimator will cause signal aliasing in passband region.

If we choose a factor D <= M with three factors: D_1, D_2, and D_3 such that

$$D = D_1 * D_2 * D_3 \quad \text{(EQ 26)}$$

We can have a 3-stage filter design, with one filter for each stage. The specifications for each stage filter depend on the original narrow lowpass filter and the stage factors D_1, D_2 and D_3. Let L_1, L_2, and L_3 be the filter lengths for each stage respectively, then the total computations per D samples can be formulated as follows

$$2 * (L_3 + L_2 * D_3 + L_1 * D_3 * D_2) \quad \text{(EQ 27)}$$

and the group delay is

$$N = (L_1 - 1) + D_1 * (L_2 - 1) + D_1 * D_2 * (L_3 - 1) \quad \text{(EQ 28)}$$

However, if the last stage factor is D3 = 2, and since there is no gain in computational efficiency for the model 2, then model 1 is used for this stage. In this case, the group delay is

$$N = (L_1 - 1) + D_1 * (L_2 - 1) + D_1 * D_2 * (L_3 - 1) / 2 \quad \text{(EQ 29)}$$

For most optimal designs, the last stage factor is 2. If L_3 is an odd number, N is always an integer. If there is a half sample delay (N is not an integer), problems will occur for wide lowpass and highpass filters where an integer group delay is required. This system will adjust the filter length in order to avoid the half-sample delay in these two filters.

The choice of decimator D and number of stages and factors for each stage is not a single-objective optimization problem. This system will give a set of 'best decimators' for the user to select.

In conclusion, it is possible to design very efficient FIR filters using multirate design methods. The only real disadvantage is the complexity of the implementation and design unless an automated design program is available for use. These filters inherently have long delays and are not suitable for applications where such long delays are inappropriate. However, the computational gain using this approach can be significant compared to standard FIR filter design methods.