Filter banks for software defined radio *(Lead: fred harris, 16 pages)*

6.1 Introduction (harris, ½ p.)

6.2 Cascade Polyphase Analysis and Synthesis Channelizers (Xiaofei, Venosa, harris, 5 p.)

- Filter Design for Perfect Reconstruction
- Multiple Arbitrary Bandwidth, Arbitrary Center Frequency Channelizers
- Pre and Post Signal Conditioning Appended to Polyphase Channelizers

6.3 Narrow Band Filters Embedded in Cascade Channelizers (Venosa, harris, Xiaofei, 5 p.)

- Cascade Down-Sample and Up-Sample Filters
- Outer Down-Sample Filter, Inner Filter, Outer Up-Sample Filter
- Multple Cascade Down-Sample and Up-Sample Filters

6.4 Broadband Filters Embedded in Cascade Channelizers (harris, Xiaofei, Venosa, 5 p.)

- Variable Broadband Filters with Binary Mask
- Variable Broadband Filters with Binary Mask And End Filter Tuning
- Variable Broadband Filters with Interleaved Narrow and Broad Sub-Channels

6.5 Concluding Remarks, (harris, ½ p.)

Contact Information
fred harris
+ (619) 594-6162
College of Engineering
San Diego State University
San Diego, CA 92182-1309
fred.harris@sdsu.edu

Xiaofei Chen
+ (619) 944-3545
7644 Seattle Drive,
La Mesa, CA 91941
chenxiaofei_sdsu@yahoo.com

Elettra Venosa
+ (610) 390-3942
12108 Caminito Campana,
San Diego, CA. 92128
evenosa@spacemicro.com

6.0 Filter Banks for Software Defined Radios.

6.1 Introduction

In this chapter we examine filter banks. There are two types of filter banks; banks that assemble or synthesize composite signals and banks that disassemble or analyze composite signals. The synthesis channelizer forms a composite broadband output signal from a set of narrowband baseband input signals. The analysis channelizer reverses the process and forms a set of narrowband baseband output signals from a composite broadband input signals. The two types of filter banks are each other's duals. The filter banks are remarkable in their capability, flexibility, and efficiency in the tasks they perform. Central to their use is their ability to change sample rate while changing bandwidth and move signals between different spectral regions using aliasing and separate aliases by phase coherent sums. We are about to develop our skills and work our way to using polyphase channelizers as a flexible variable multiple bandwidth channelizers. To do so it is useful to first examine and learn how a polyphase filter uses resampling to implement an efficient single bandwidth filter. The entry point to this process is the design and implementation of a filter when there is a large ratio of sample rate to bandwidth. Figure 6.1 presents an example of such a filter.

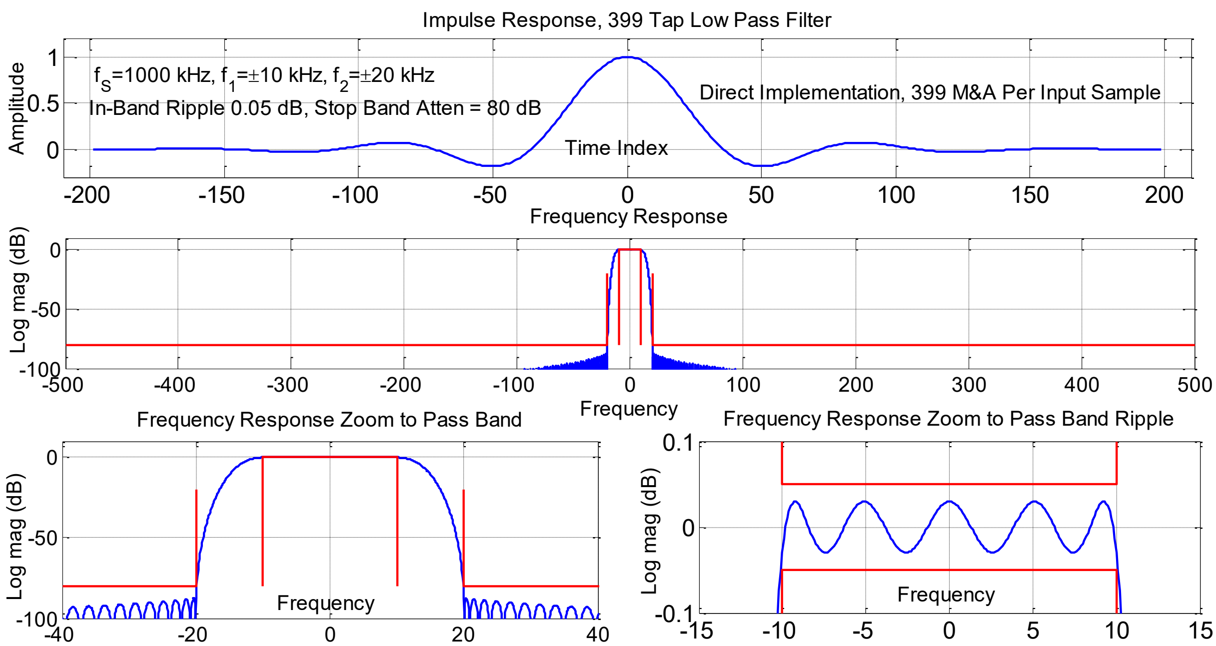

Figure 6.1. Time and Frequency Response of 399 Tap FIR Filter with Large Ratio of Sample Rate to Bandwidth

When we have a large ratio of sample rate to bandwidth, the filter has a large number of coefficients and a large number of arithmetic operations are required to implement it. We now examine a number of option that implement these filters with reduced workload. Since the problem is caused by the high sample rate relative to the filter bandwidth, the obvious solution is to reduce the sample rate. We can do this with an M-path polyphase filter that reduces the sample rate as part of the filtering process. There are scenarios in which we have to preserve the sample rate for system considerations. We acknowledge the need to preserve sample rate by considering the efficient filtering problem to have two parts; the first is to perform the filtering with a small workload and the second is to preserve the sample rate. We solve the two problems in two filters; the first reduces the sample rate while reducing the bandwidth and the second increases the sample rate while preserving the bandwidth. We have been asked the question "Why would two filters be better than one filter?" The answer is because there are two problems here and we treat them as such. The block diagram of the polyphase down sampler and the polyphase up sampler is shown in Figure 6.2. In this example the prototype filter is partitioned into a 20 path polyphase filter with 20 coefficients per path. The input and output sample rates of the filter are 1000 kHz and 50 kHz respectively, and the two sided bandwidth of the filter, down to its -80 dB stopband level, is 40 kHz. A second 20-path filter with different weights is designed to use the 10 kHz excess sample rate as its transition bandwidth when up-sampling the 50 kHz sample rate back to the 1000 kHz sample rate. Note the cascade filters require 20 operations per input sample and 20 operations per output sample for a total of 40 operations per input-output sample pair. This is a 10 to one reduction in workload to implement this filter. Figure 6.3 shows the time and frequency response of the cascade filter. The significant aspects of the spectral responses are essentially identical to that seen in the direct implementation. The obvious difference in the two implementations is the time delay of the impulse response. The delay is seen to be approximately twice the original interval, 380 samples rather than 199 samples. This extra delay is of course the consequence of passing the signal through two filters.

We can reduce this delay as well as reduce the workload by abandoning the original filter and replacing it with an equivalent filter with the same specifications as the original but designed for a reduced sample rate. A modified form of figure 6.2 can be seen in Figure 6.4 where we show that the input and output polyphase filters simply perform sample rate changes for a reduced length inner filter which performs the filtering task at a reduced input sample rate. In this example the sample rate is reduced 10-to-1 to 100 kHz by the input filter. Its output is processed by the inner filter that performs the reduced workload filtering and passes along the bandwidth limited samples to the output filter. This filter performs a 1-to-10 sample rate increase as an interpolation process which forms output samples matched to the original input rate.

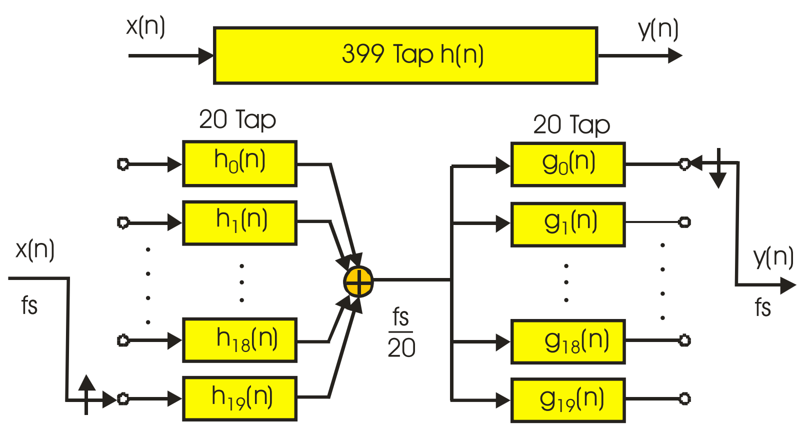

Figure 6.2 Implementing a Narrow Bandwidth Filter as a Cascade of Polyphase Down and Up Sampling Filters

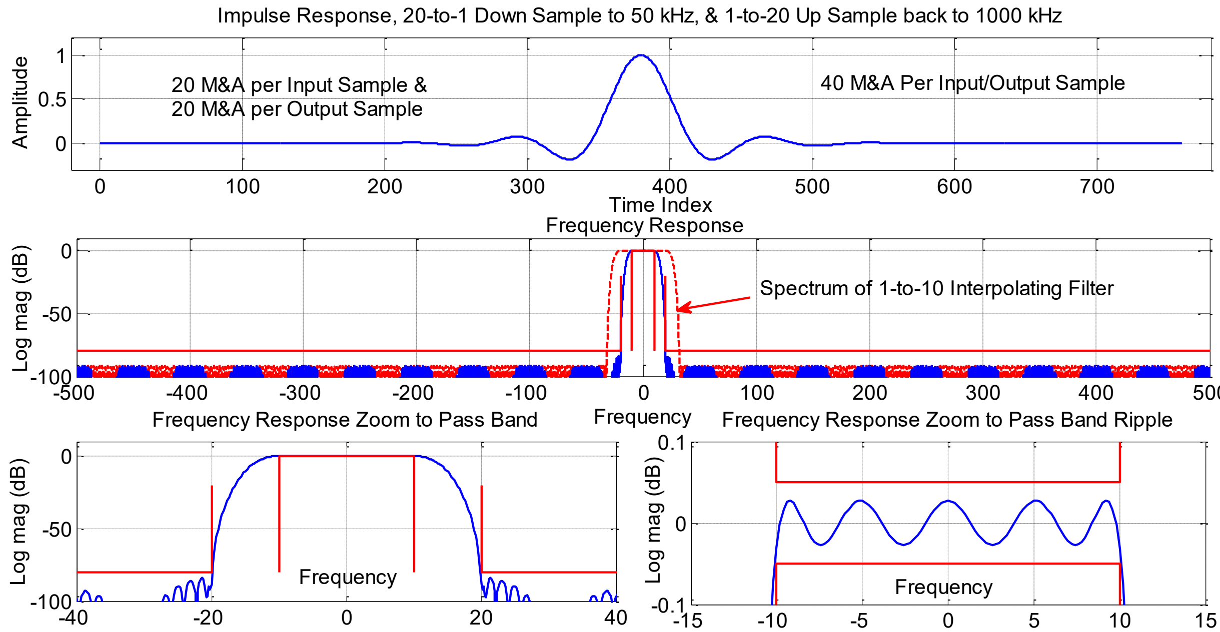

Figure 6.3. Time and Frequency Response of Cascade 20-to-1 Down-Sampling and 1-to-20 Up-Sampling M-Path Filters

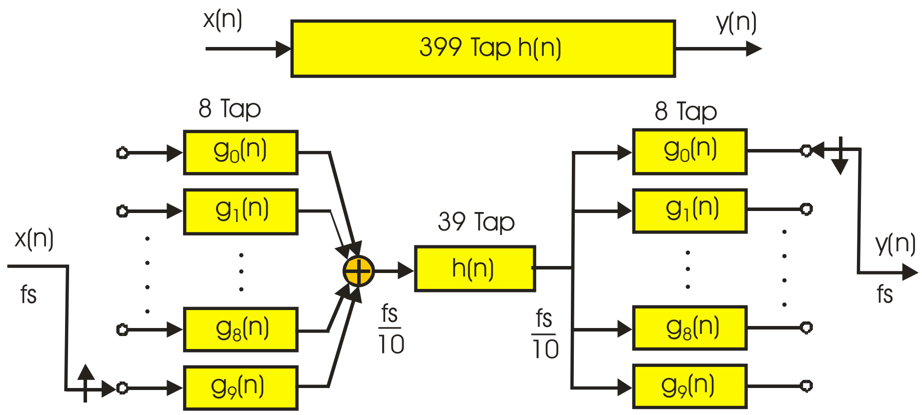

Figure 6.4 Implementing a Narrow Bandwidth Filter as a Cascade of Input 10-to-1 Polyphase Down Sampling Filter, an Inner Filter, and an Output 1-to-10 Polyphase Up Sampling Filter

Figure 6.5 shows the time and frequency response of the three cascade filters. The first pleasant surprise is the net reduction in workload to implement the narrowband filter. The workload here is 20 operations as opposed to the original 400 and the previous reduction to 40. The time delay has also been reduced. The delay here is 259 samples, an increase from the original 199 but reduced considerably from the 360 of the previous reduced workload. We note the reduced delay even though we have traversed three filters in this cascade.

The lesson to be learned in this section is that when implementing a digital FIR filter, workload reduction, on the order of a magnitude, can be had if we can reduce the sample rate while reducing the bandwidth. We have a mechanism, the interpolator, to return to the original input sample rate or to any other desired sample rate commensurate with the bandwidth reduction. It would seem that a requirement to access this

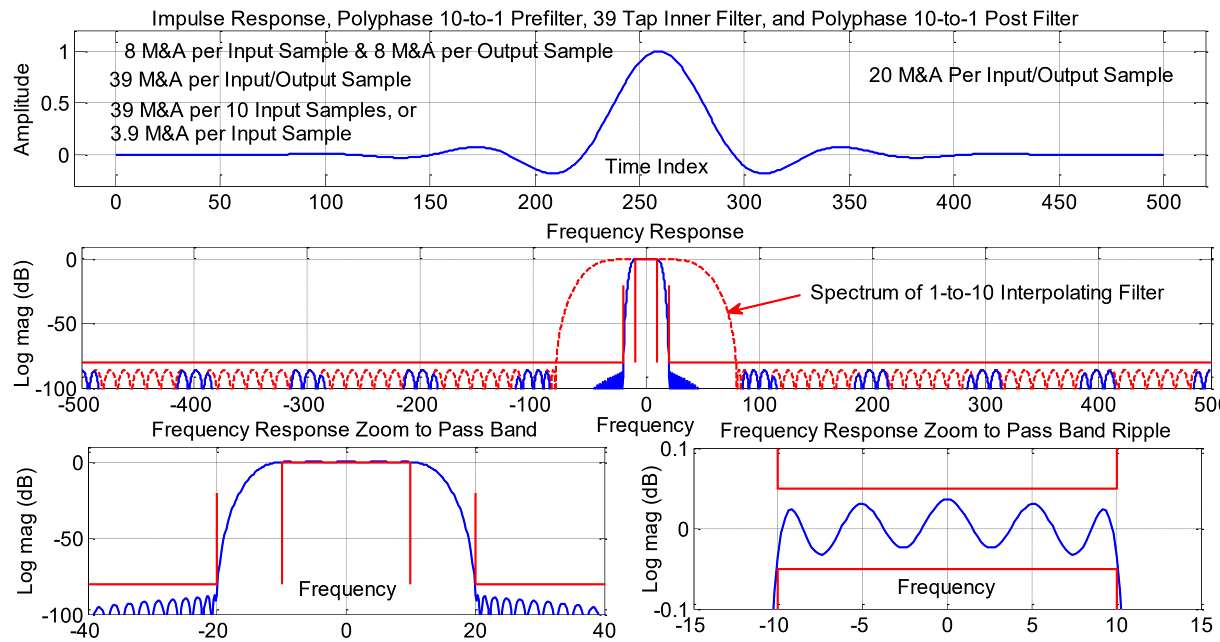

Figure 6.5. Time and Frequency Response of Cascade 10-to-1 Down-Sampling, Inner Filter, and 1-to-10 Up-Sampling M-Path Filters

significant workload reduction is a sample rate reduction as part of the bandwidth reduction, a condition assured when there is large ratio of sample rate to bandwidth. Then it would seem that this option is not available when this condition is not met, such as when the sample rate to bandwidth ratio is small, such 1.5 or 2.2. The surprise is that the option is still valid for this later case. We can use the analysis channelizer to partition the input bandwidth into narrow bandwidth segments for which there is large ratio of sample rate to bandwidth. Thus the computational savings can then be had for wide bandwidth signals partitioned temporarily into narrow bandwidth signals which are then reassembled by the synthesis channelizer.

6.2 Filter Banks

Figure 6.6 illustrates the tasks performed by the two types of channelizers. We start with a set of narrow bandwidth baseband signal sequences each sampled at a common low sample rate slightly higher than the signal's two sided bandwidth. For instance, the signals may have a two sided bandwidth of 16 MHz at a 20 MHz sample rate. Having the sample rate exceed the two sided bandwidth by 20 to 25 percent is an important consideration in the design of channel filters in the sampled data signal domain. The multiple baseband sequences, say M of them, are presented to the synthesis channelizer. The channelizer interpolates each sequence to raise the sample rate by a factor of M, the rate necessary to satisfy the Nyquist criterion for the wider bandwidth of its composite output signal. With access to the higher sample rate the synthesizer can up convert each signal to its assigned center frequency and sums their up-converted components to form the composite output signal.

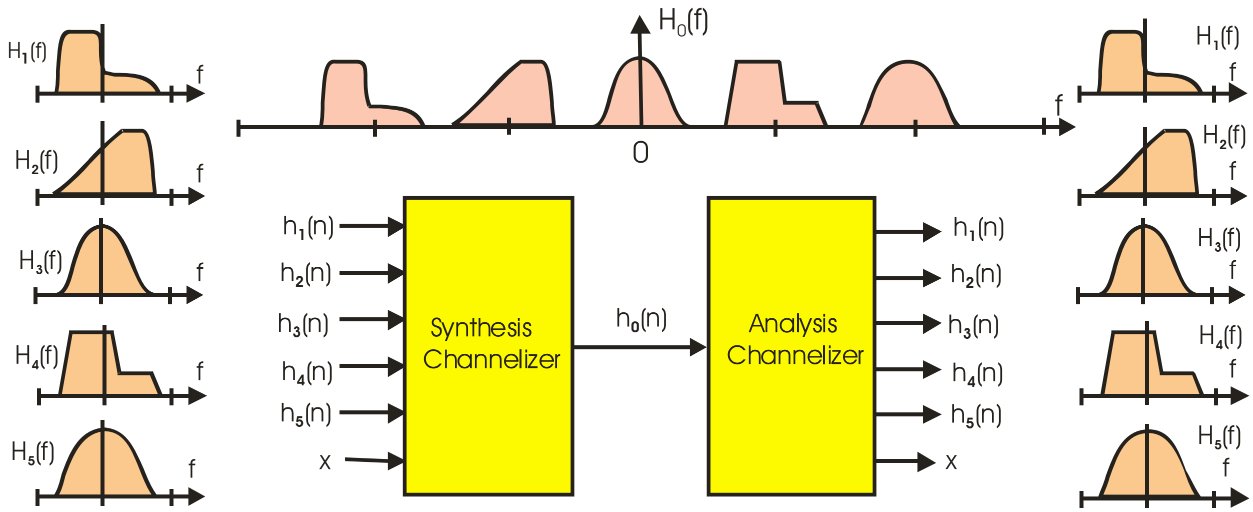

Figure 6.6 Input and Output Spectra for Synthesis and Analysis Channelizers

As a specific example, suppose we form a composite output signal containing 5 of the baseband signals sampled at 20 MHz and arrange for the channel centers to be multiples of 20 MHz. When the channel spacing equals the channel sample rate the channelizer is known as a maximally decimated filter bank. The bandwidth of the composite signal, spanning five 20 MHz bands is approximately 100 MHz. Keeping in mind the need to have a sample rate slightly higher than the signal bandwidth we design the synthesizer for an output sample rate of 120 MHz, the sample rate equivalent to having six 20 MHz channels. We design the synthesis channelizer to accept six 20 MHz input signals with one channel being a null channel. We use the null channel to raise the sample rate to 120 MHz and reserve the extra bandwidth of the null channel for the transition bandwidth of the analog filters following the conversion from the sampled data representation to the continuous domain. In a similar fashion, we collect and present the composite signal at the input to the analysis channelizer at a sample rate that supports null channels for the benefit of the analog anti-aliasing filter. We could collect the data at 160 MHz sample rate and decompose the received composite signal with an eight channel analysis channelizer, keeping five channels and discarding three channels. For the illustration of Figure 6.6, the analysis channelizer operates at the 120 MHz sample rate and forms six channelized outputs, five of which contain signals, and one to be discarded null channel.

Two examples of spectral responses for maximally decimated filter bank are shown in figure 6.7. In both examples the output sample rate is 20 MHz, same as the channel spacing, the filter passband bandwidth is 16 MHz with transition bandwidths of 2 MHz and 4 MHz respectively. We see in the upper subplot there is no overlap of adjacent channel filter responses and we see in the lower subplot that each channel response overlaps their two adjacent channels. When channel filtered and down sampled to 20 MHz, the transition bandwidth extending beyond the 10 MHz folding frequency folds or aliases into its own transition band. While there are many scenarios where this folding is acceptable we are interested in a variation of the channelizer in which the transition bands that extend into the adjacent spectral interval do not fold due to resampling. We avoid the band edge folding by raising the sample rate beyond the two sided bandwidth that includes the transition bandwidth out to their stopband edges. Raising the output sample rate so it no longer performs M-to-1 or 1-to-M resampling changes the filter architecture slightly and reclassifies the filter bank to be a non-maximally decimated filter bank.

The appendix to this chapter contains MATLAB script for the 6-channel maximally decimated synthesis and analysis channelizers. We have included this script so the reader can compare the architecture change between the maximally decimated filter bank and the non-maximally decimated filter bank.

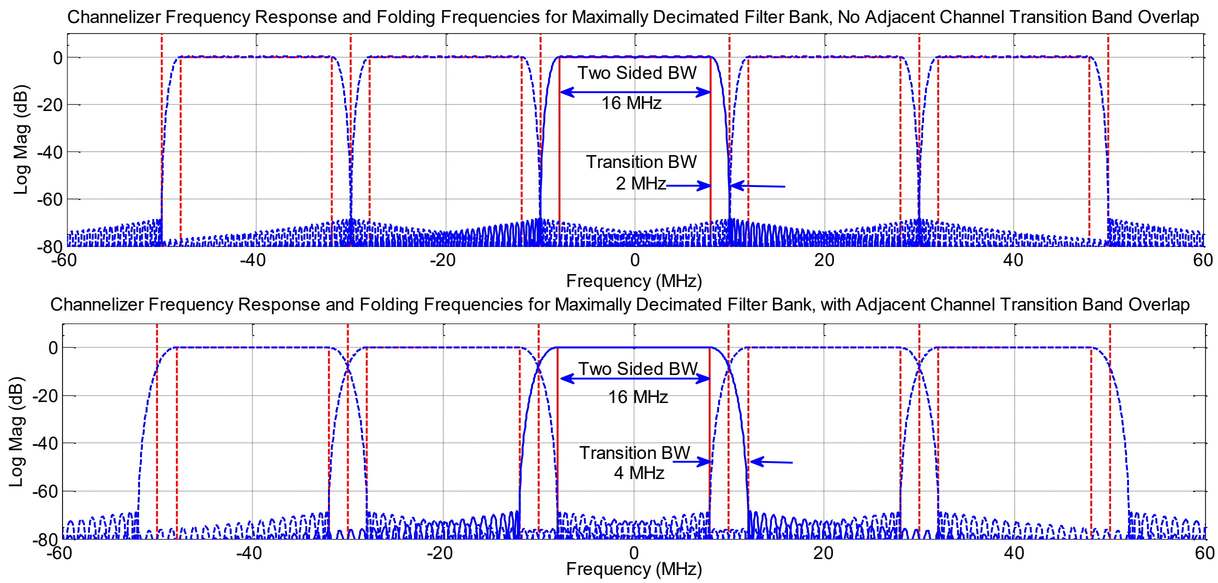

Figure 6.7 Spectra of five Occupied Channels in a Six Channel Maximally Decimated Filter Bank with No Overlap between Adjacent Channels and with 2-MHz Overlap and Aliasing with Adjacent Channels

6.3 Cascade Polyphase Analysis and Synthesis Channelizers

Channelizers can be stand-alone processes and in fact often are. For instance a synthesis channelizer can form a broadband signal from multiple baseband signals as a modulator embedded in a transmitter of a communication system. Its signal is delivered through a channel to a receiver that contains an analysis channelizer that extracts the multiple narrow bandwidth components preceding the detection process. Similarly, an analysis channelizer is the core of source coding processes such as the MP3 audio compression algorithm. The components of the analysis process are sorted, ranked, quantized and compressed with minor losses by rules related to a psycho-acoustic model and perceptual limitations of the human hearing process. The compressed data is collected by the synthesis channelizer that distributes tagged components to their appropriate ports of a filter bank containing selected bandwidths and center frequencies matching those of the analysis filter bank. In the examples just cited, while the analysis and synthesis channelizers are in cascade, they reside at the two ends of a communication link. The channelizers we now study operate as a cascade coupled pair with the pair residing at both ends of the communication link.

6.3.1 Non-Maximally Decimated M-Path Polyphase Analysis and Synthesis Channelizers

The maximally decimated M-path analysis filter bank accepts M input samples prior to computing its output vector from its M output ports. Figure 6.8 shows the essential components of an M-path analysis and an M-path synthesis channelizer. Here we can clearly see their dual structures. The analysis filter bank is formed by an input M-port commutator that delivers M input samples to the length M input data buffer, an M-path polyphase filter and an M-point IFFT that outputs successive samples from the M-output channels. The synthesis filter is formed by an M-point IFFT which accepts M samples of an input time series, an M-path polyphase filter, and a length M output data buffer accessed by the M-port output commutator that delivers M output time samples. The filter bank performs an M-to-1 down sampling, forming an M-point output vector at the rate fs/M which matches the channel frequency spacing of fs/M.

We can raise the channelizer output sample rate by altering the filtering process to accept fewer than M-samples prior to computing the M-point output vector. For instance, in the 6-path channelizer we introduced earlier as our ongoing example, we can form a 6-point vector output for every 5 input samples, which would result in an output sample rate of 24 MHz (120/5) as opposed to the original 20 MHz (120/6) sample rate. Another option is to obtain a 6-point vector output for every 4 input samples, or every 3 input samples which would result in output rates of 30 MHz (120/4) or 40 MHz (120/3) respectively. The down sampling embedded in the polyphase filter is responsible for the spectral aliasing. The aliasing causes all multiples of the output sample rate to alias to baseband. We will process the aliased signal components to separate the aliases. If the sample rate is less than the channel's the two sided bandwidth the filter's transition bandwidth aliases within the band. These aliasing terms can also be removed, as they are in an MP3 channelizer, but we elect to modify the channelizer to avoid the transition bandwidth folding by increasing the sample rate of the separate channels.

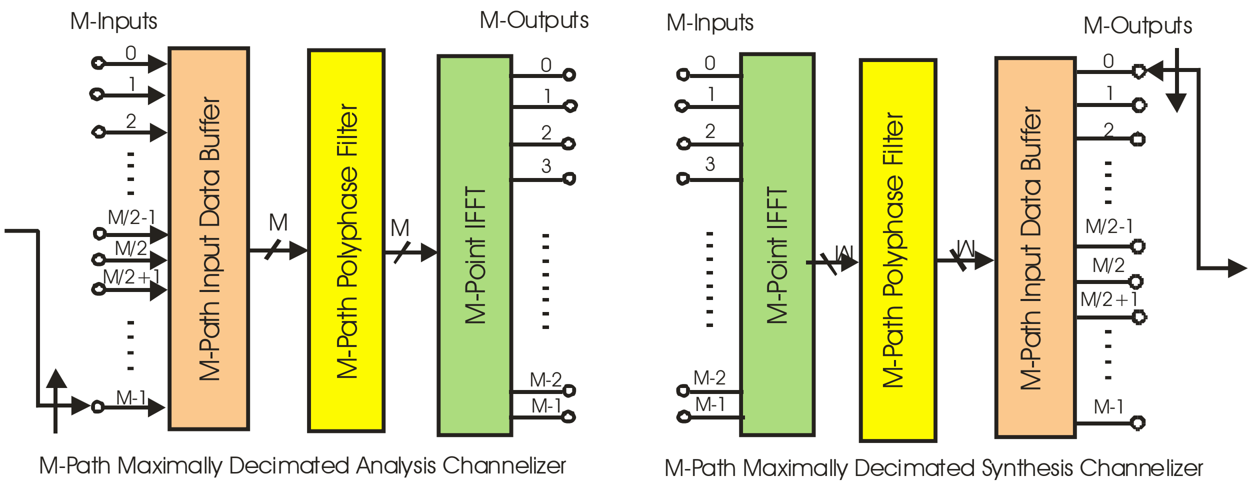

Figure 6.8 Essential Components of M-Path Maximally Decimated Analysis and Synthesis Filter Banks

When the output sample rate matches the channel spacing, all channels are aliased to baseband which was a wonderful attribute of the maximally decimated filter bank. With an increased output sample rate the channel centers no longer alias to baseband, but they do alias some known frequency and we know to which frequency each channel's center aliases. The channels still alias as a result of the down sampling, they just alias to some known offset frequency. Successive samples from each channel port are spinning due to the frequency offset and we can complete the conversion to baseband by simply de-spinning the successive samples with the conjugate complex phasor. For instance, had we elected to do 4-to-1 down sampling in our 6-channel analysis channelizer to obtain a 30 MHz output sample rate, we would know that the signal centered at 20 MHz has aliased to -10 MHz with a 30 MHz sample rate and is thus spinning $-2\pi/3$ radians per sample, while the signal centered at 40 MHz has aliased to +10 MHz and is thus spinning at $+2\pi/3$ radians/sample. A simple state machine can track the successive angle corrections to be applied to each offset alias output channel to de-spin its signal back to baseband.

We have another option which proves the worth of understanding properties of linear systems. We know that time shift in the time domain is responsible for frequency dependent phase shift in the frequency domain. Thus rather than apply a phase rotation to the output ports of each channel output, we can obtain the same phase correction as appropriate end-around rotations of a circular buffer containing the time samples to be processed by the IFFT. The state machine we described earlier will now control the end around rotation of the circular buffer. Isn't that grand?

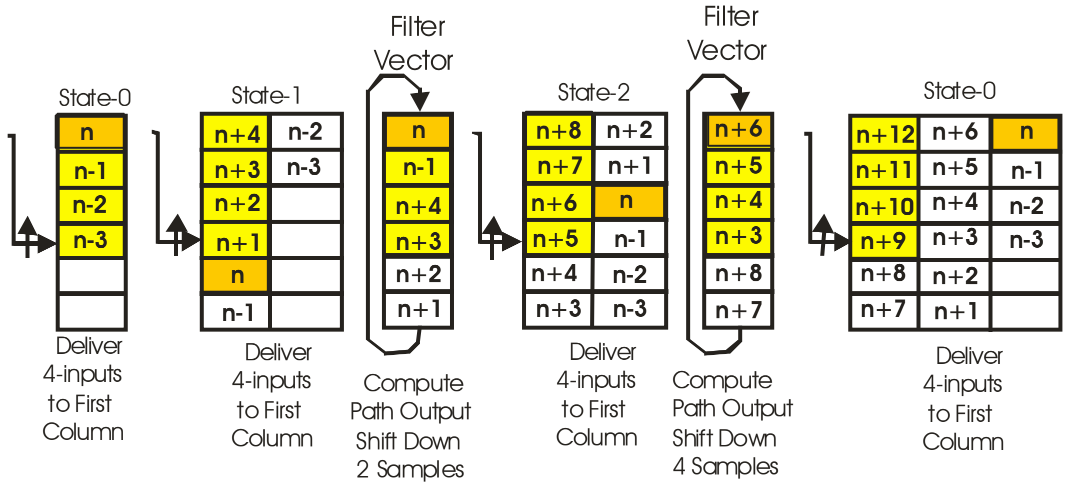

Figure 6.9. Deliver 4-Input samples to Serpentine Data Array. Monitor Origin, Top of Array on Successive Shifts. Compute Path Row Outputs, Perform Circular Shift to Return Origin Row to Top of Array

Figure 6.9 illustrates how the state machine tracks a specific input sample through the serpentine shifts on successive inputs of 4 input samples presented to a six path filter. We deliver the first 4 input samples, declare the sample at the top of array our temporary origin and the data state, state 0. On the next shift of 4 input samples, the tagged sample is moved down to the 4-th row of the buffer. We label this as the next state, state 1, compute the path outputs of the 6-path filter and then shift the output vector down 2-samples to roll the entry with the tagged sample back to the top of the array. When the next 4 samples are input to the array, the serpentine shift of data in the array slides our tagged sample to row 2 in the next column. We label this next state, state 2, compute the path outputs of the 6-path filter and then shift the output vector down 4 samples to roll the entry with the tagged sample back to the top of the array. When the next 4 input samples are input to the array, the serpentine shift slides the tagged sample to the first row in the next column. This where we started. We recognize this as state zero and declare the current sample at the top of the first column as our new temporary origin.

The amount of down sampling of particular interest to us in this section is M/2 input samples per output sample. For ease of implementation, we select M to be an even integer. For the M/2 down sample, the output sample rate becomes fs/(M/2) or 2 fs/M. Thus the output sample rate is twice fs/M, the channel spacing in the M-path channelizer. For the example we have been using in this section, we will perform 3-to-1 down sampling in our 6-path channelizer which means the output sample rate for each channel is 40 MHz with channel spacing of 20 MHz for which adjacent channel filter responses cross at ±10 MHz. The M/2 down sample in an M-path filter cause the even multiples of the channels spacing center frequencies to alias to baseband while the odd multiples alias to the half sample rate. For our on-going example the 3-to-1 down sample of the 120 MH sample rate sets the channel output rate to 40 MHz. The even multiple of 20 MHz, which are ± 40 MHz aliases to baseband while the odd multiples, ± 20 and 60 MHz alias to the half sample rate. The circular buffer that shifts the origin of the filter output vector is controlled by a two state state-machine. The machine performs a 3-sample end around shift on alternate output vectors to flip the signs of alternate samples formed in the odd indexed bins of the IFFT. The result of this circular shift is that all six channels now reside at baseband. The appendix to this chapter presents the MATLAB script for a 6-to-1 and for a 3-to-1 down sampled 6-path analysis channelizer. It is instructive to see the difference in the script files. The primary difference is the 3-sample end around shift on alternate outputs controlled by a binary flag that alternates value on successive 3-input sample vectors. The appendix also presents the state machine for the 1-to-6 and 1-to-3 up sampled 6-path synthesis channelizer. Here again the primary difference between the two is the binary flag that alternates values on successive 3-output sample vectors. There is also a slight change in addressing for the inner products that form the 6-path output vector which merge the upper and lower half to form the 3-output samples formed by the 6-path filter. The modification to the M-path channelizers when they are non-maximally decimated is shown in Figure 6.10. Compare this to the versions shown in Figures 6.8. The primary difference is seen in the sum at the output of the synthesis channelizer the merges the first and second halves of the output data buffer. To emphasize and preserve the duality of the two channelizers this summation is echoed at the input to the analysis filter. The derivation of the dual flow diagram is found in reference 2. While the script written to reflect the complete duality of the two channelizers will work, the MATLAB script in the appendix matches the signal flow of the synthesis channelizer but uses an equivalent but more efficient implementation of the analysis channelizer.

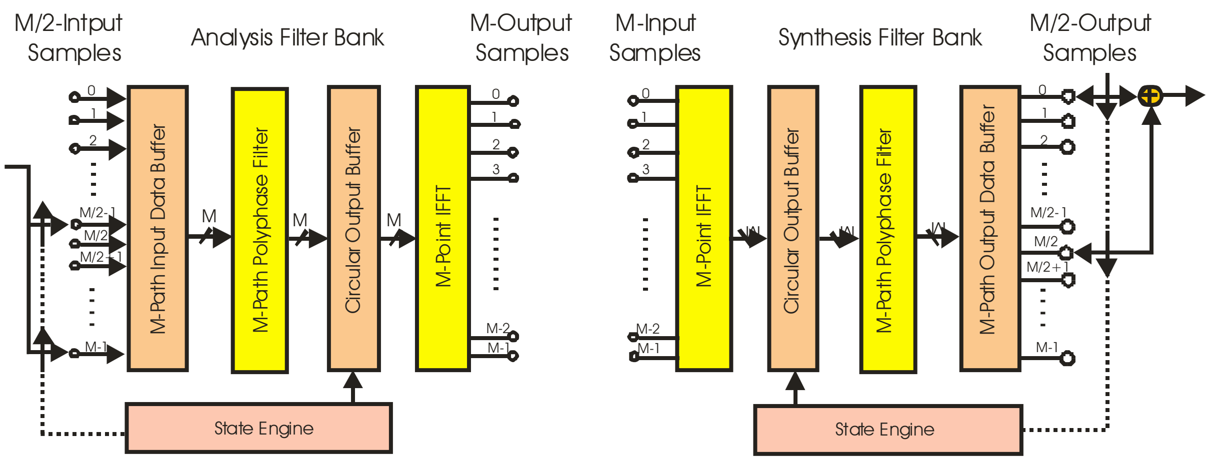

Figure 6.10 Essential Components of M-Path Non-Maximally Decimated Analysis and Synthesis Filter Banks

6.3.2 Filter Design for Perfect Reconstruction

We will shortly examine an extremely versatile channelizer structure constructed from a cascade of non-maximally decimated analysis and synthesis channelizers. In order to make use of this configuration, the channel filters must be designed to realize a Nyquist frequency response. Nyquist filter frequency response for channelizers require the adjacent channel cross over point to be 0.5 (or -6 dB) and have an odd symmetric transition bandwidth about the crossover frequency. Our first response to meet this requirement is to design the channelizer filter to be a SQRT Nyquist filter. This seems reasonable since we pass through the same filter twice, once in the analysis filter and then again in the synthesis filter. We do this all the time in communication systems, deliver Nyquist pulses to the receiver with half the shaping occurring at the modulator and half at the demodulator. The problem here is that the cosine tapered SQRT Nyquist filter is a terrible filter. Hard to believe, but it is true! The cosine taper is not sufficiently smooth to obtain the spectral side lobe levels or in-band ripple levels we require for our channelizers. We know how to design SQRT Nyquist filters with other tapers that will support the desired side lobe and in band ripple levels. See references 3 and 4. We have used the SQRT Nyquist filters designed with these alternate techniques to implement the cascade filter banks and they significantly improve the channel frequency response in side lobe levels and in-band ripple level. The problem with the improved SQRT Nyquist filters is that when we merged adjacent channels to obtain perfect reconstruction, the composite spectrum exhibited significant ripple levels at the channel crossover frequencies.

It is not necessary to form a Nyquist filter as a cascade of two SQRT Nyquist filters. The reason we use the two filters in a communication application is we use the second filter as a matched filter to suppress the receiver's additive white Gaussian noise. We don't have to suppress additive noise in our cascade channelizers since they are not separated by a noisy channel. Thus we are free to make one filter of the cascade filter pair a Nyquist filter and design the other filter to have a wider passband that does not distort the passband or transition band of the Nyquist filter. We normally place the Nyquist filter in the analysis filter bank because there are times we want to observe the channelized signals between the analysis and synthesis banks.

Figure 6.11 shows the frequency response limits for the Nyquist filter design that will reside in the analysis filter bank and for the wider reconstruction filter that will reside in the synthesis filter bank. The top subplot shows the Nyquist spectrum at baseband and at the 20 MHz offset center frequencies either side of baseband. The adjacent channels crossover at 10 MHz offsets at amplitude 0.5 or -6 dB. The transition bandwidth of the Nyquist filter should not extend more than half way to the folding frequency of its band because we have to leave a reasonable span of transition bandwidth for the following synthesis filter. Reducing the transition bandwidth of the Nyquist filter lengthens the filter which increases the workload and delay through the filter. A reduced transition bandwidth may offer benefit when examining the signal content at the analysis channelizer output. The center subplot shows the spectral response of the baseband analysis filter with a nominal 6 dB bandwidth of 20 MHz with 40 MHz sample rate. The bottom subplot shows the spectral replicates of the analysis filter output at baseband and at 40 MHz offsets. The synthesis channelizer is designed with a pass band that spans the two sided passband and transition bandwidth of the Nyquist spectrum and rejects the edges of the spectral replicates centered at 40 MHz offsets. In practice the stopband edge of the synthesis filter is permitted to have a slight overlap with the edge of the analysis filter spectral replica.

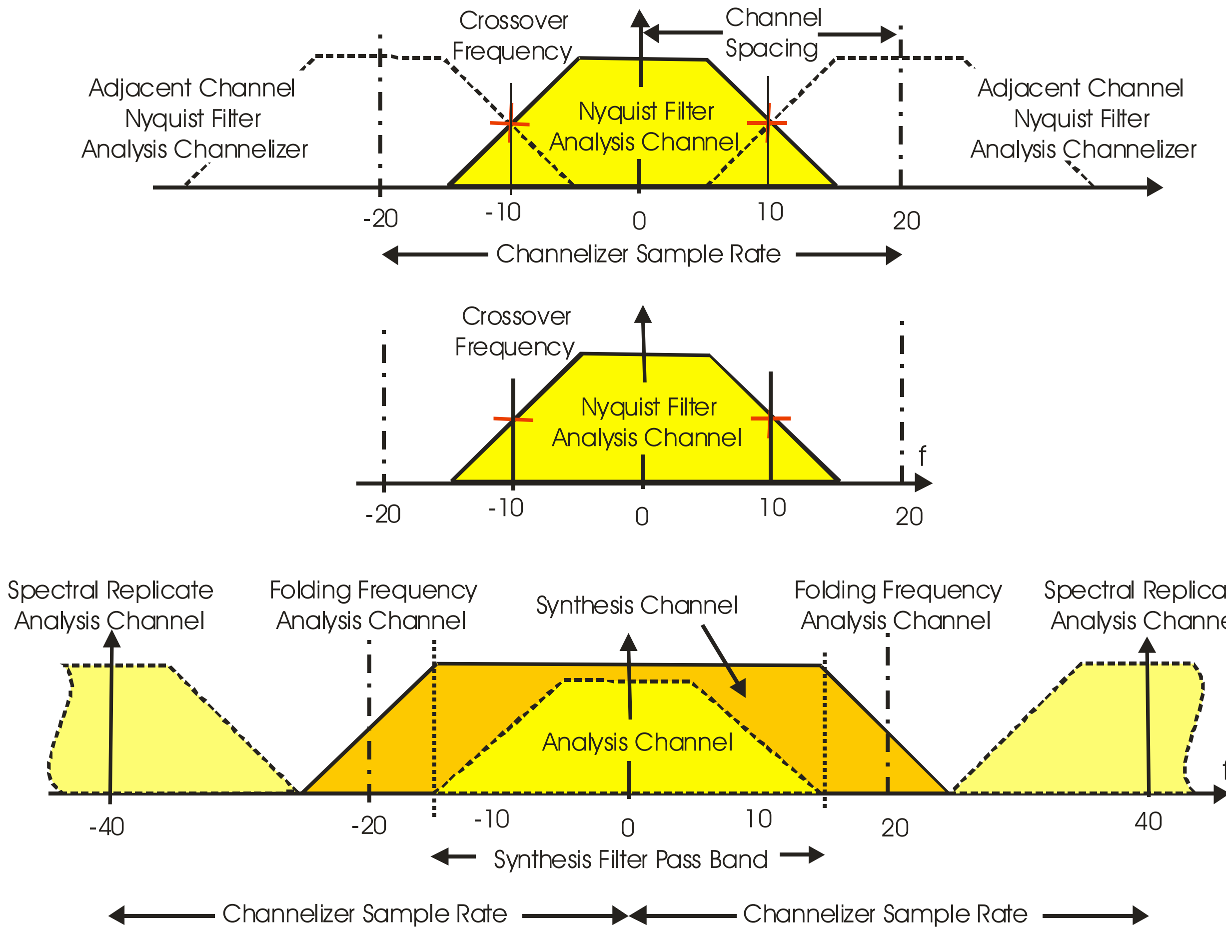

Figure 6.11 Frequency Response, Nyquist Filter in Analysis Channelizer, Analysis Channelizer Output Channel, and Wider Synthesizer Channelizer Reconstruction Filter

6.3.2.1 Nyquist Filter Design

Now the question is, how do we design the Nyquist filter? Remarkably simple: we use a windowed sinc function which has the form shown in (6.1). In the sinc, the distance between zero crossings is the reciprocal bandwidth and the distance between samples is the reciprocal sample rate. Using Nz as the number of zero crossings offset from the origin, we can use the argument, -Nz:1/M:Nz, the argument of the sinc, instructs the sinc script to form samples spanning the interval –Nz to +Nz zeros crossings in increment step sizes of 1/M. Counting the matching left and right samples and the center valued sample the filter length obtained by this indexing is (2·Nz·M)+1. We want an odd number of samples but as an implementation considera-tion, we want it to be one less than M times an integer. We obtain this number by removing one sample from each side of the index by starting at –Nz+1/M and ending at +Nz-1/M and obtain a filter length (2·Nz·M)-1 where 2·Nz are integers, i.e. 6, 7, 8, …., which are the number of taps per path of the M-path filter. We select an Nz, as multiples of 0.5, and set the length for the Kaiser window to the length of the sinc and scale the product for unity passband gain.

```
hh=sinc(-Nz:1/M:+Nz).*kaiser(2*Nz*M+1,beta)';
hh=sinc(-Nz+1/M:1/M:Nz-1/M).*kaiser(2*Nz*M-1,beta)';          (6.1)
hh=hh/M;
```

The parameter beta in the Kaiser window argument is the window's time bandwidth product. As this parameter increases, the spectral main lobe width increases and the spectral side lobe levels decrease. The spectral main lobe affects the transition bandwidth of the windowed Nyquist filter. After applying the window, we examine the spectrum and adjust the value of beta till the edge of its transition bandwidth meets the stopband spectral mask and examine the level of stopband side lobe. If the side lobe levels are above the required stopband level we have to increase beta. An in beta increases the window's main lobe width which in turn increases the widowed filter's transition bandwidth. We respond by increasing the filter length by incrementing Nz. We stop when the stop band attenuation is below the targeted design level when the transition band edge is touching the stopband spectral mask. This process is illustrated in Figure 6.12. When the filter length and window parameter have been determined, we append a zero to the filter so the total number of weights is an integer multiple of M the number of paths.

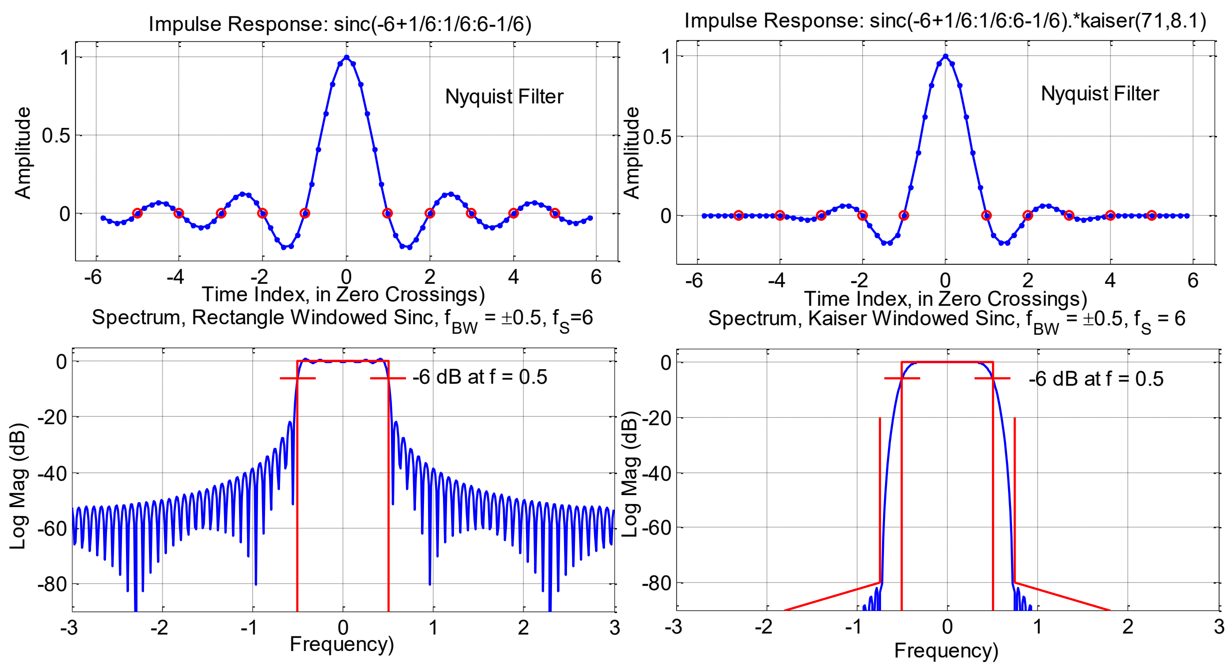

Figure 6.12 Rectangle Windowed and Kaiser Windowed Sinc Filter and Their Spectra with Spectral Masks

Using normalized frequencies for the passband edge, stopband edge and sample rate of 0.5, 0.75 and 3, we ran through this design process for increasing filter length to obtain the relationship between Nz, beta, and attenuation that is shown in table 6.1. If our design criterion for the analysis filter required 90 dB or better side lobe attenuation we would select a 12 taps per path, 71 tap filter with Kaiser window parameter slightly below 9.4

Table 6.1. Stopband Attenuation as Function of Filter Length with Required Beta for that Attenuation

Ntaps/path	6	7	8	9	10	11	12	13
Filter Size	35	41	47	53	59	65	71	77
Atten (dB)	47.1	54.9	60.0	68.8	75.0	84.8	93.2	103.4
Nz	3.0	3.5	4.0	4.5	5.0	5.5	6.0	6.5
Beta	4.0	5.0	5.8	6.6	7.5	8.4	9.4	10.4

The script we used to design the Nyquist filter is shown in (6.2). The scaling coefficient 6 sets the filter's passband gain to 1 but is removed in the reshape command because we want the path gains to be 1. The reshape command appends a leading 0 to bring the number of coefficients to 72, a multiple of 6, and partitions the prototype impulse response into a 6-path filter with 12 coefficients per path containing coefficients with indices h(r+6n), r = 0, 1, 2, 3, 4, 5 for the row index.

```
hh=sinc(-6+1/6:1/6:6-1/6).*kaiser(71,9.2)'; % Nyquist Filter          (6.2)
hh=hh/6;                                    % Scaling
hh2=reshape([0 6*hh],6,12);                 % 6-Path Partition
```

The coefficients of the polyphase partition are shown in Table 6.2 where the rows are labeled 0 through 5. Interesting to note that the 0-th row contains all zeros except for the 1 in column 6. You would expect this in an M-path Nyquist filter, we have seen a similar relationship in a true half band filter where a 2-path polyphase partition has the top row all zeros with a single centered value of 0.5 (or 1 if scaled). The remaining rows 1 through 5 are mirror images of their rows 5 through 1 with row 3 being its own mirror image.

Table 6.2. Coefficients of 6-Path Filter with 12 Coefficients per Row

	0	1	2	3	4	5	6	7	8	9	10	11
0	0	0.0000	-0.0000	0.0000	-0.0000	0.0000	1.0000	-0.0000	0.0000	-0.0000	0.0000	-0.0000
1	-0.0000	0.0008	-0.0050	0.0189	-0.0559	0.1747	0.9516	-0.1145	0.0395	-0.0126	0.0029	-0.0003
2	-0.0001	0.0019	-0.0111	0.0397	-0.1152	0.3906	0.8153	-0.1643	0.0573	-0.0176	0.0037	-0.0003
3	-0.0002	0.0032	-0.0162	0.0552	-0.1585	0.6166	0.6166	-0.1585	0.0552	-0.0162	0.0032	-0.0002
4	-0.0003	0.0037	-0.0176	0.0573	-0.1643	0.8153	0.3906	-0.1152	0.0397	-0.0111	0.0019	-0.0001
5	-0.0003	0.0029	-0.0126	0.0395	-0.1145	0.9516	0.1747	-0.0559	0.0189	-0.0050	0.0008	-0.0000

We now address the filter designed for the synthesis channelizer. The prototype for the synthesis filter is designed using a variation of the Remez algorithm. The variation uses a modified weight vector that forms an L_∞, Chebyshev, or equal ripple, passband approximation to unity gain passband and a 1/f or -6dB/octave decay rate for its stopband ripple. The MATLAB script code for the variation is available from the authors. The MATLAB script for the synthesis filter of a 6-path channelizer with output sample rate of 120 MHz is shown in (6.3).

```
gg=remez(70,[0 15 25 60]/60,{'myfrf',[1 1 0 0]},[1 1]); % Synthesis Filter     (6.3)
gg2=[reshape([0 gg],6,12);                              % 6-path partition
```

We can now examine the spectral response of the Nyquist filter and the reconstruction filter used in the analysis and synthesis filter banks. The top subplot of Figure 6.13 shows the time response of the Nyquist filter with each 6-th sample marked by a red circle to tag the zeros of the Nyquist time series. These zeros effect the polyphase partition by establishing a row consisting of zeros and a single centered 1. The center subplot shows the frequency response of the filter with the spectral masks indicating the -6 dB passband between ±10 MHz and the -90 dB stopband at ±15 MHz. The three bottom subplots show spectral detail important to the filter response. The bottom left most subplot shows the cross-over response at 10 MHz of adjacent channel responses. As expected, the crossover level is -6.02 dB. The bottom center subplot shows the pass band ripple pass in the ±5 MHz frequency span. This span width is the result of the odd symmetric transition band about the -6 dB 10 MHz Frequency. The level of the ripple, 0.0002 dB, is important because it contributes to the reconstruction error level when we merge adjacent channel spectra. The bottom right most subplot shows the transition bandwidth of the Nyquist filter and the spectral stopband mask at 15 MHz.

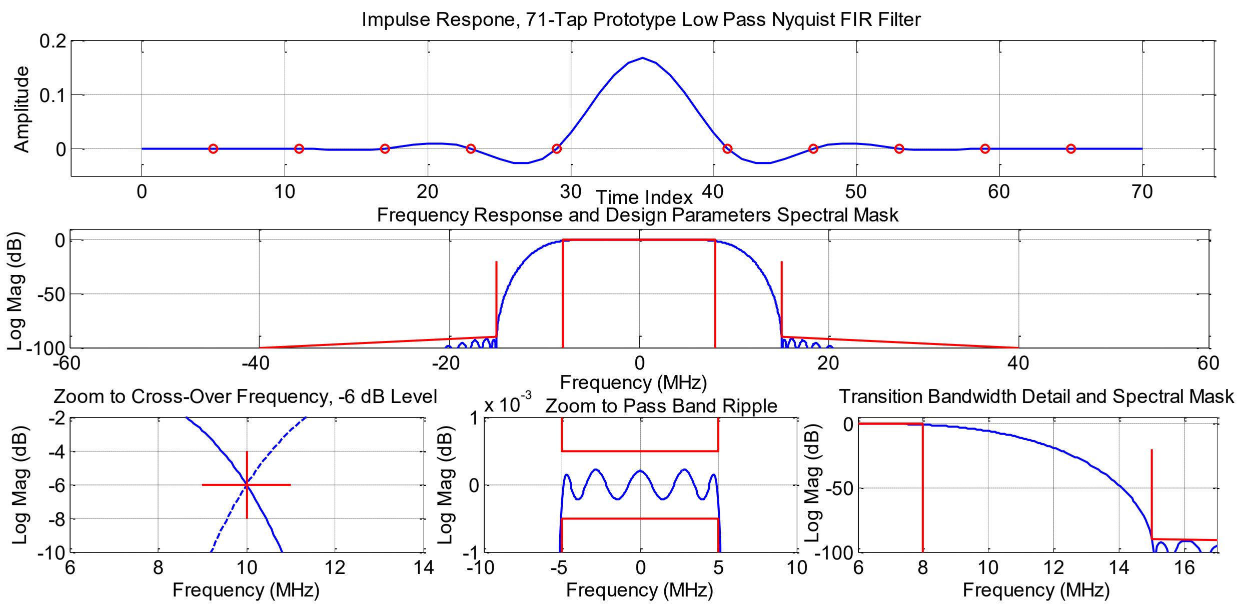

Figure 6.13 Impulse Response and Frequency Response of Nyquist Filter of Analysis Filter Bank Along with Details of Crossover levels, In-Band Ripple Levels and Transition Bandwidth

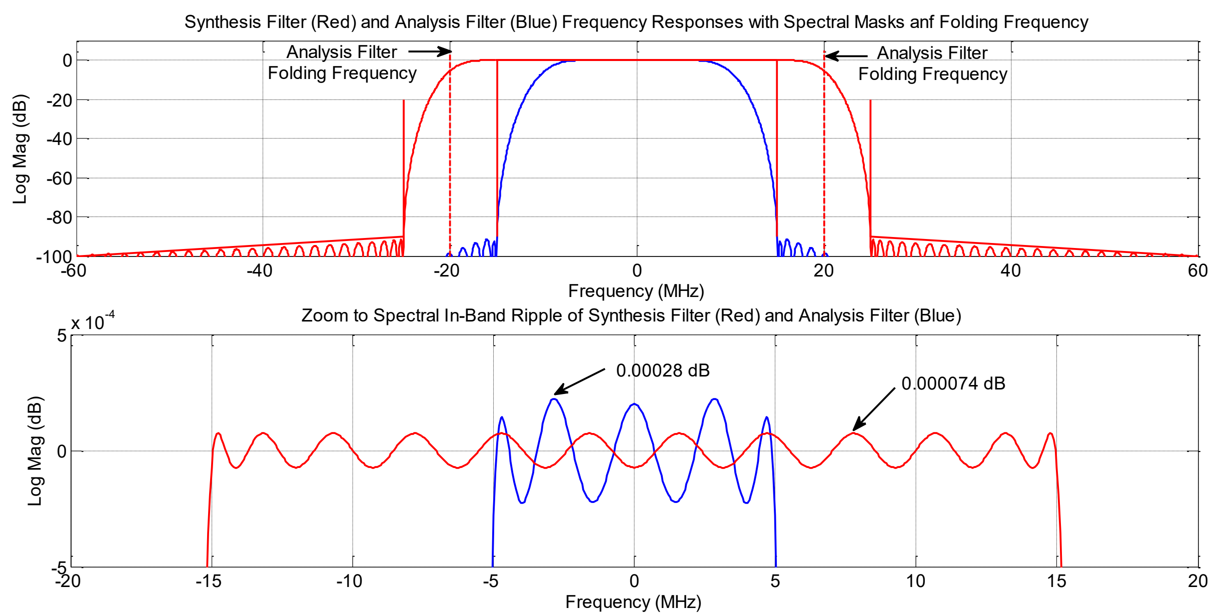

Figure 6.14 Frequency Response of Reconstruction Filter in the Synthesis Filter Bank Bracketing Frequency Response of Analysis Filter with Spectral Masks, Folding Frequencies, and Zoom to Passband Ripple

Figure 6.14 shows the frequency response of the reconstruction filter with the spectral masks of its passband and stopband at 15 and 25 MHz respectively. It is shown as an overlay on the spectral response of the Nyquist filter to illustrate how well the passband fits over the full sided bandwidth of the Nyquist filter. The bottom subplot is a zoom to the in-band ripple levels of the reconstruction filter and of the Nyquist filter. Not that both filters have the 10 MHz transition bandwidth and the same 90 dB stopband attenuation level but differ in their passband ripple levels. This is because the Nyquist filter symmetry require the passband ripple and the stop band ripple to have the same values (relative to their target levels of unity and zero). The reconstruction filter response is not so restricted and the penalty weights in the Remez algorithm allow us to set different levels of in band and out of band ripple. Remember the combination of the two ripples become the reconstruction error as the input signal passes through both analysis and synthesis filters.

6.4.1 Cascade Channelizers

We are now prepared to join the analysis and the synthesis filter banks. Figure 6.15 shows how to connect the analysis and synthesis filter banks for two demonstrations. To review the process, the analysis filter bank partitions the input spectrum into multiple contiguous overlapped narrow band channels that are down sampled and down converted to baseband. The synthesis channelizer accepts multiple input narrow band baseband channels that are up sampled and up converted to contiguous overlapped channels. We can form a selectable bandwidth filter by presenting a subset of the base band channelized time series from the analysis filter bank to the corresponding input terminals of the synthesis filter bank. The output of the synthesis channelizer is a super channel formed by seamlessly merging the multiple narrowband input channels. This is the option illustrated on the right side of Figure 6.10.

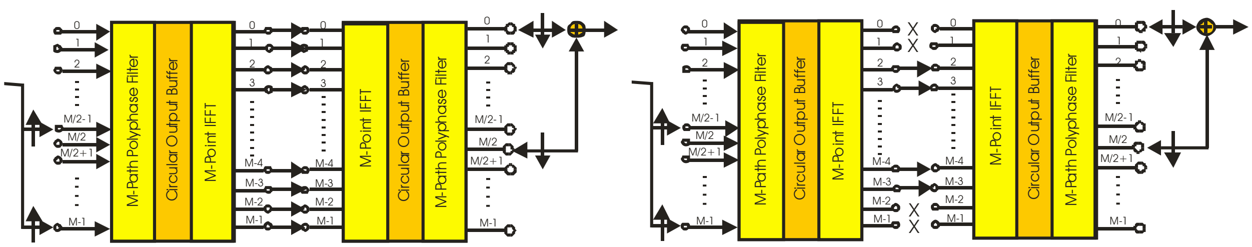

Figure 6.15 Cascade Non-Maximally Decimated Analysis and Synthesis Filter Banks. Pair on Left, Fully Connected for Verification Test, Pair on Right Partially Connected to Merge Channels for Wider Super Channel

Before we illustrate the disassembly of an input signal into multiple narrowband signals and then reassemble the multiple narrowband signal back to its original form we should verify the proper operation of the cascade filter banks. An interesting and telling test of the system is its impulse response. This is a valid test in spite of the fact that a multirate system does not have a transfer function and in fact has multiple impulse responses. We still have to answer the question, “Why are we breaking an input signal into many narrow band segments and then reassembling them? We will answer that question in a moment. Let’s first conduct the impulse response test for the fully connected channelizer, the one corresponding to the left segment of Figure 6.10. Here every spectral segment formed by the analysis filter is presented to the synthesis filter which, if things work as planned, will assemble a signal occupying the full spectral span of the input signal. This means that an impulse at the input will cause an impulse to be formed at the output with a delay associated with the two causal channelizer filters. Anything else, other than the impulse, which appears at the output will be artifacts reflecting imperfect reconstruction. The questions will be, “How well have we done? What sizes are the artifacts?” The MATLAB script that performed the impulse response test on the cascade filter bank is listed in the appendix.

Figure 6.16 presents the result of the impulse response probe of the fully connected cascade non-maximally decimated analysis and synthesis filter bank. The top subplot shows the expected response of the cascade 71-tap 6-path filter banks. Note we are not seeing the impulse response of the filter which would be the 6-times oversampled Nyquist pulse we embedded in the channelizer, bur rather are seeing the impulse response of perfect reconstruction filter bank, a bank the reproduces the input at its output. The pulse has been reconstructed after a 71 sample delay. The center subplot is a view of the time domain artifacts obtains by zooming in with high magnification to the low level components. We see the largest artifact at position

35 of amplitude 1.7 10^{-5}, an artifact 5 orders of magnitude below the desired signal. The bottom subplot is the spectrum of the unit impulse, which if there were no artifacts would be a constant 0 dB over the spectral interval. We see instead there is a periodic ripple pattern with peak amplitude 1.8 10^{-4} dB which is approximately 20.7 parts per million. That's worth repeating, the reconstruction error is 0.18 thousandth of a dB, not perfect, but not bad! In fact it is overkill but it is a consequence of designing the Nyquist filter for greater than 90 dB stop band attenuation, -90 dB being a deviation from zero of 3.16 10^{-5} or 31.6 parts per million. We would have larger reconstruction errors by allowing the reconstruction filter to have larger in-band ripple. Bear in mind that the 6-path filter has 6 impulse responses. Testing all 6 responses, we found the worst case periodic ripple pattern had a peak amplitude of 2.6 10^{-4} dB, or 30.3 maximum ppm reconstruction error.

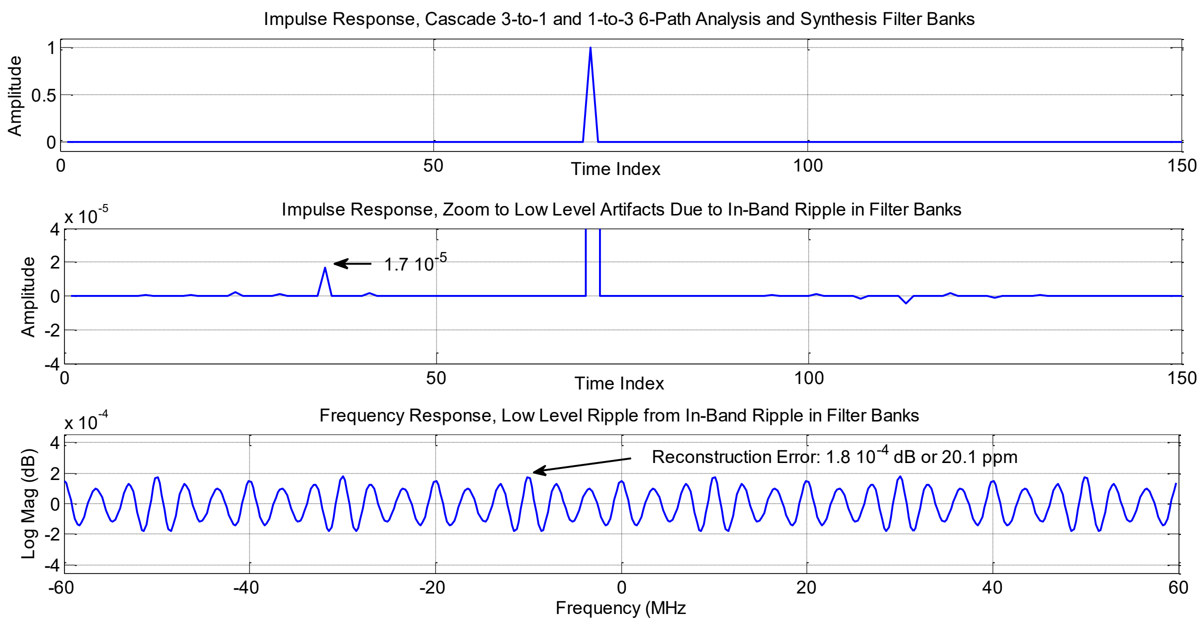

Figure 6.16 Top Subplot, Impulse Response of Cascade 71-Tap 6-Path Analysis and Synthesis Filter Banks. Center Subplot, Zoom to Low Level Time Domain Artifacts, and Bottom Subplot, Zoom to Spectrum Ripple

Figure 6.17 shows the frequency response of the six channels formed by the 6-path, 3-to-1 down sampled analysis filter bank. When we performed the impulse response of the cascade filter bank, the time series output from the 6-output ports of the analysis channelizer were the impulse response of its 6 channels. We simply collected the time series from the output ports of the analysis channelizer and transformed their impulse responses to obtain the separate aliased and down sampled frequency responses. Note, as designed, all their responses have -6 dB frequencies at ±10 MHz, output sample rate of 20 MHz, and 90 dB stop band attenuation.

Figure 6.18 presents the result of the impulse probe of three merged channels in the cascade non-maximally decimated analysis and synthesis filter bank. This demonstrates the ability of the cascade to form super channel output filters from a selected subset of analysis filter outputs. The top subplot shows the impulse response of the super channel formed by merging three narrow channels in the cascade 71-tap 6-path filter banks. Notice that selecting 3 channels to be merged out of the 6 available channels builds a half band filter. We verified this by placing red o markers on alternate impulse response, zero valued, samples. The center subplot shows the spectra of the three adjacent sub channels merged by the synthesis filter bank to form the

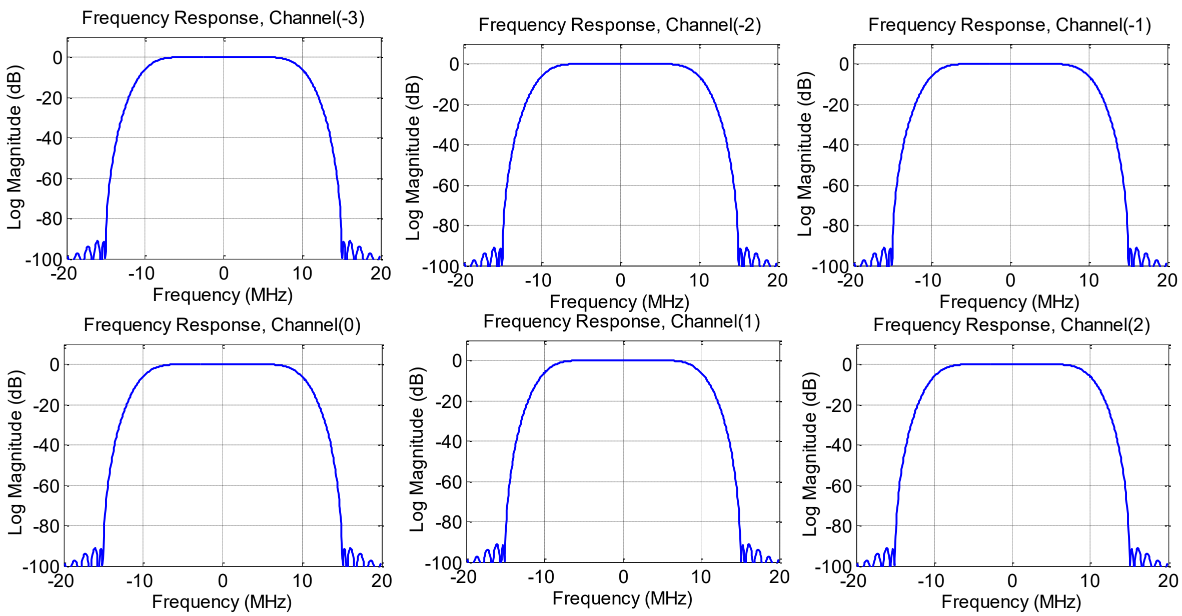

Figure 6.17 Frequency Response of All Six Channelizer Output Ports of 3-to-1 Down-Sampled 6-Path Analysis Channelizer

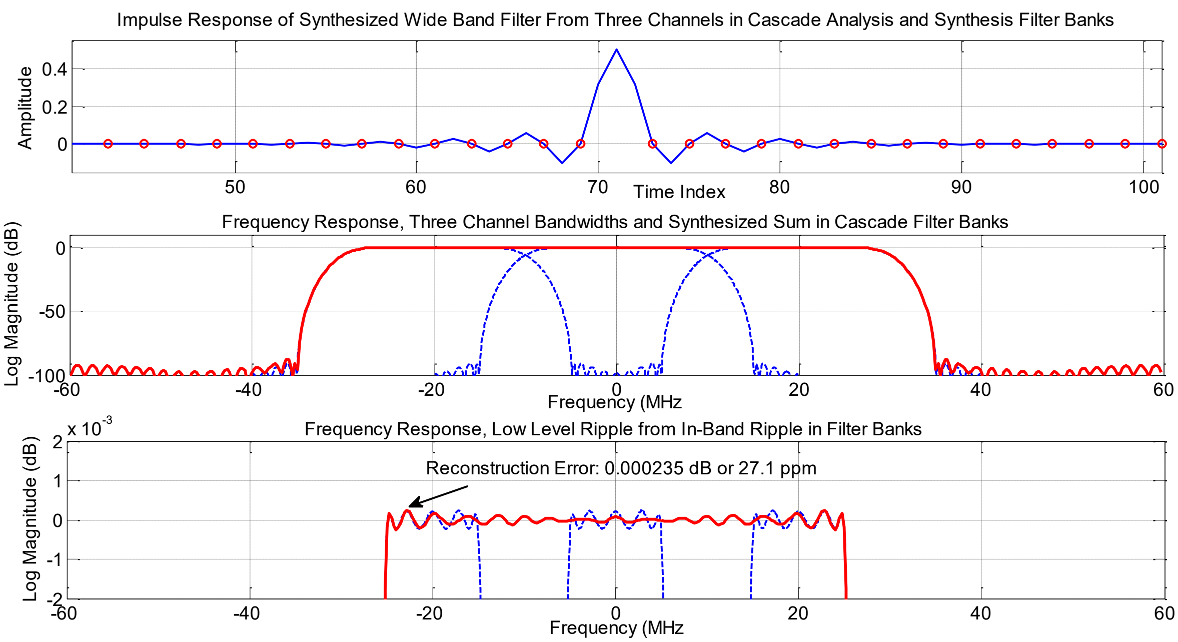

Figure 6.18 Top Subplot, Impulse Response of 3-Merged Channels of Cascade 71-Tap 6-Path Analysis and Synthesis Filter Banks, Center Subplot, Spectra of 3 Adjacent Offset Channels and their Merged Sum Super Channel from Synthesis channelizer, and Bottom Subplot, Zoom to In-Band Spectral Ripple Levels.

super channel as well as the channel response of the merged channel. The bottom subplot is a zoom to the spectra of the in-band ripple of the three sub channels and of the merged composite channel. It is interesting to see that in the interval between the ripple levels of the adjacent channels the (blue) filter responses fall quite dramatically but their sum smoothly fills in the (red) interval to the nominal 0 dB level. The in-band ripple level are seen to be comparable to the separate channel ripple levels, specifically 2.4 10^{-4} dB.

6.4.2 Cascade Channelizers for Variable Bandwidth Filters

In the previous section we examined a cascade of six path analysis and synthesis channelizers operating as a 3-to-1 and 1-to-3 resampling filter bank. We had selected a small number of stages to illustrate the frequency and time domain properties of the prototype filters and how they interact in the cascade. We now examine a 30-to-1 and 1-to-30 resampling sixty channel channelizer to better illustrate the flexibility and versatility offered by have access to more degrees of freedom. In particular we designed a 60 channel channelizer. Using the designs similar to those in (6.2) and (6.3) and shown in (6.4) and (6.5) for the Nyquist analysis filter and reconstruction synthesis filter respectively. As done for the six path filter, the analysis filter was designed for 80 dB stopband attenuation. Since the Nyquist filter has equal passband and stopband ripple, its passband is absurdly small, $2.8\cdot10^{-4}$ dB. Using the Remez algorithm penalty weights we were able to design the synthesis filter with the same 80 dB attenuation and with a more reasonable 0.05 dB passband ripple. We were able to achieve the two design ripple levels with a shorter length filter, 699 taps for the synthesis as opposed to the 719 taps for the analysis. Designing the synthesis filter to relaxed passband ripple specifications resulted in the reduced length filter which in turn reduced the computational workload of the filter as well as the group delay through the filter.

```
hh=sinc((-6+1/60:1/60:6-1/60)).*kaiser(719,9.2)';                    (6.4)
hh=hh/60;
hh2=reshape(60*[0 hh],60,12);

gg=remez(598,[0 15 25 600]/600,{'myfrf',[1 1 0 0]},[1 80]);          (6.5)
gg2=reshape([0 gg],60,10);
```

We will not show the result of the impulse response test for the cascade of the 60-path analysis and synthesis filter banks, but be assured we did conduct the test to validate proper operation of our script. What we will show you is the result of the reduced bandwidth design presented in the right side of Figure 6.15. In this design we coupled 11 of the output ports from the analysis filter bank to their corresponding ports at the input to the synthesis filter bank to form a filter with two sided bandwidth 11/60 of the input sample rate. Figure 6.19 shows the impulse response and the frequency response of the synthesized super channel formed from the 11 selected filter bank channels. Also shown is the in-band ripple of the supper channel which has inherited the 0.05 dB design ripple level of the synthesis channelizer bank. Note in particular the equal level ripple of the synthesized super channel has two distinct components. The ripple aligned with the channel pass band has a different period than the ripple spanning the interval between the channel pass bands. We have an option to select the 11 channels from any offset spectral interval of the analysis filter band and deliver them from the offset frequency span to the baseband span of the synthesis filter. Had we elected this option we would have had a free spectral translation in the filter bank. We have a few more clever options available to us but we want to call attention here to the computational economy of implementing the filter as a super filter formed by narrow band sub channels.

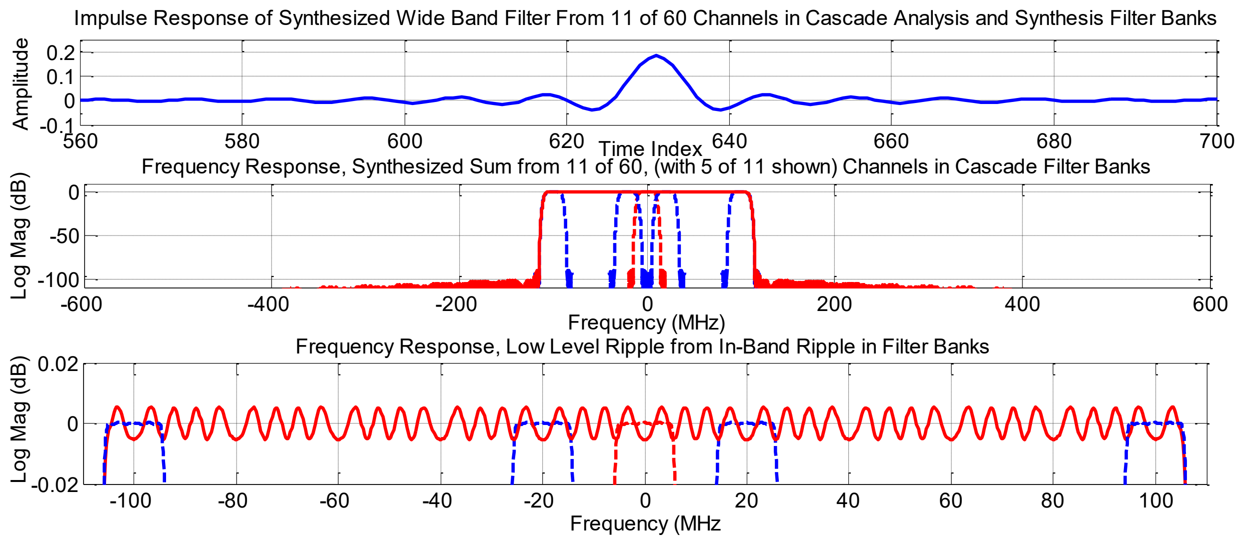

Figure 6.19. Impulse Response of 11-Sub-Channel Synthesized Super Channel, Frequency Response of Synthesized Channel and Analysis Channel Sub-Channels, and Zoomed Detail of Passband Ripple.

Suppose we are tasked to implement the super filter we just synthesized by merging 11-sub-channels in the 60-path cascade analysis synthesis engine. The filter that meets the same specifications of the 11-merged analysis filter bands with transition bandwidth, passband ripple, and stopband attenuation level of the synthesis filter, not surprisingly requires 601 taps. The MATLAB script that designed the equivalent filter is shown in (6.6).

```
qq=remez(600,[0 106 115 600]/600,{'myfrf',[1 1 0 0]},[1 20]);                (6.6)
```

Now we might ask why would we implement the 601 tap filter with a 719 tap analysis and 599 tap synthesis filter on the input and output of two 60 point IFFTs? Good question! Table 6.3 itemizes the workload of the four processing blocks. What we have to do is amortize the workload per processing cycle over the 30 input-output samples processed per cycle. Here we see the input channelizer, with its 720 taps distributed over its 60 input ports and exercised every 30 input samples requires 24 multiplies per input and similarly the output channelizer, with its 600 taps distributed over its 60 output ports and exercised every 30 output samples requires 20 multiplies per output. We are up to 44 multiplies per input; what remains is the workload for two 60 point IFFTs. The 60 point IFFT is implemented as a Good-Thomas, or Prime Factor, algorithm with factors, 3, 4, and 5. When the short factor IFFTs are performed by a set of un-nested Winograd transforms the workload for the 60 point IFFT is 200 real multiples for complex input samples. The nested version of the same algorithm would require 188 real multiplies. The workload for the pair of IFFTs is 13.3 multiplies per input. Thus the total workload for the cascade polyphaser implementation of the 601 tap filter is 57.3 multiplies per input. This is workload is less than 10% of the workload for the direct implementation. So this is the reason we might consider the cascade polyphaser filters to perform a filtering task. The cascade filters are Green, they offer an order of magnitude reduction in workload to perform the filtering task! Figure 6.20 presents the frequency responses of the two implementations. The reduced workload of the channelizer implementation does have a processing cost. The cost is to be seen in Figure 6.19; the delay to the center tap of the channelizer impulse response is seen to be 630 samples, and the corresponding delay of the 601 tap filter would be 300 samples. The cascade channelizer has the signal propagating through both the input and output channelizer filters so we would expect the additional delay in the cascade implementation.

Table 6.3 Computational Workload to Implement a 601 Tap Filter in 60-Path Channelizer

Processing Task	Workload per 30 Inputs	Workload per Input
720 Tap Analysis Filter	720 Multiplies	24 Multiplies
60 Point IFFT	200 Multiplies	6.67 Multiplies
60-Point IFFT	200 Multiplies	6.67 Multiplies
600 Tap Synthesis Filter	600 Multiplies	20 Multiplies
Total Workload	1720 Multiplies	57.4 Multiplies

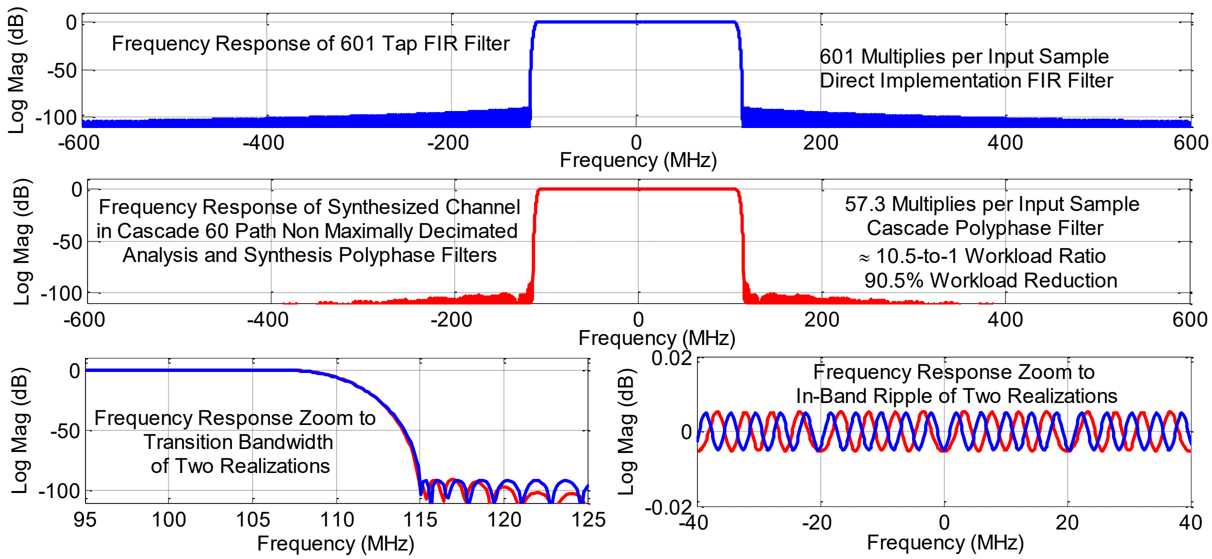

Figure 6.20. Frequency Response of 601 Tap Direct Implementation of Low Pass Filter and Synthesized Impulse Response from 11-Sub-Channels in Input 60-Path, 30-to-1 Down Sample and Output 60-Path 1-to-30 Up Sample Filter bank with Comparisons of Transition Bandwidths and In-Band Ripple Levels.

One of the clever things we can do with the cascade channelizer is change sample rate while forming super channels from the narrow bandwidth sub-channels. We may have cause to do this when the bandwidth of the synthesized channel is significantly narrower than the sample rate of the channelizer. This is true for the channel we just synthesized from 11 sub channels. There the bandwidth of the synthesized channel was ± 106 MHz and the sample rate was maintained at 1200 MHz. We have the option to reduce the sample rate, say to 400 MHz by reducing the size of the output IFFT and M-path synthesis channelizer from 60-paths to 20 paths. The block diagram of the synthesized and down sampled cascade is shown on right side of Figure 6.21. The left side of the same figure presents the synthesized and up sampled cascade, the dual of version of the task we are now examining. We discuss use of the dual resampling channelizers in the next section.

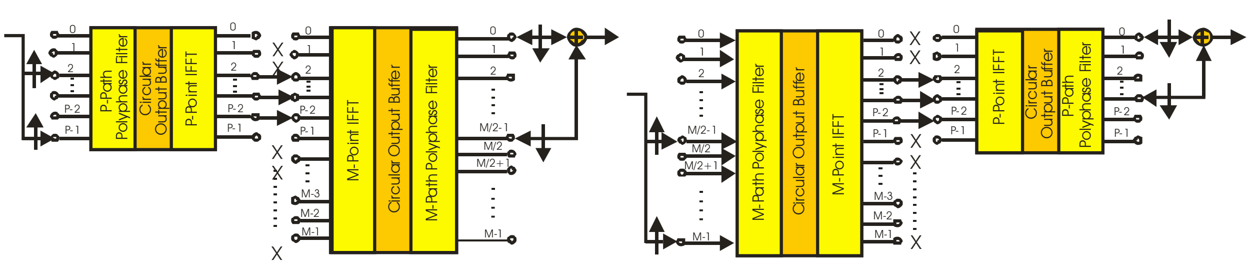

Figure 6.21 Cascade Non-Maximally Decimated P-Path Analysis, M-Path Synthesis Up Sampling Filter Bank and M-Path Analysis, P-Path Down Sampling Filter Bank.

Table 6.4 itemizes the workload of the two processing blocks in the now different size analysis and synthesis channelizers. We still amortize the workload per processing cycle over the 30 input-samples presented per work cycle. The input channelizer, with its 720 taps, the output channelizer with its 200 taps, and the two IFFTs, the 60 and 20 point IFFTs requiring 200 and 40 multiplies respectively require a total of 1160 multiplies per 30 input samples. Thus the total workload for the cascade polyphase implementation and 3-to-1 down sampled version of the 601 tap filter is 1160/30 or 38.7 multiplies per input. We now compare this workload to the 3-to-1 down sampled 3-path polyphase partition of the 601 tap direct implementation which has a workload of 200 multiplies per input sample. The resampled channelizer workload of 38.7 multiplies is 19.35% of the workload for the resampled direct implementation which is still a significant workload ratio. Figure 6.22 presents the frequency responses of the two down sampled implementations. Interestingly, while both versions of the filters meet the stop band specifications, due to stop band aliasing, they are seen to have rates of stopband roll off. The impulse responses of the two filters have delays of 211 and 100 samples, which of course is the same delay interval but 1/3 of the of the clock samples at clocks with 3-times longer periods.

Table 6.4 Computational Workload to Implement a 601 Tap Filter in 60-Path Channelizer
With Embedded 3-to-1 Down Sample from 1200 MHz to 400 MHz

Processing Task	Workload per 30 Inputs	Workload per Input
720 Tap Analysis Filter	720 Multiplies	24 Multiplies
60 Point IFFT	200 Multiplies	6.67 Multiplies
20-Point IFFT	40 Multiplies	1.33 Multiplies
200 Tap Synthesis Filter	200 Multiplies	6.67 Multiplies
Total Workload	1160 Multiplies	38.7 Multiplies

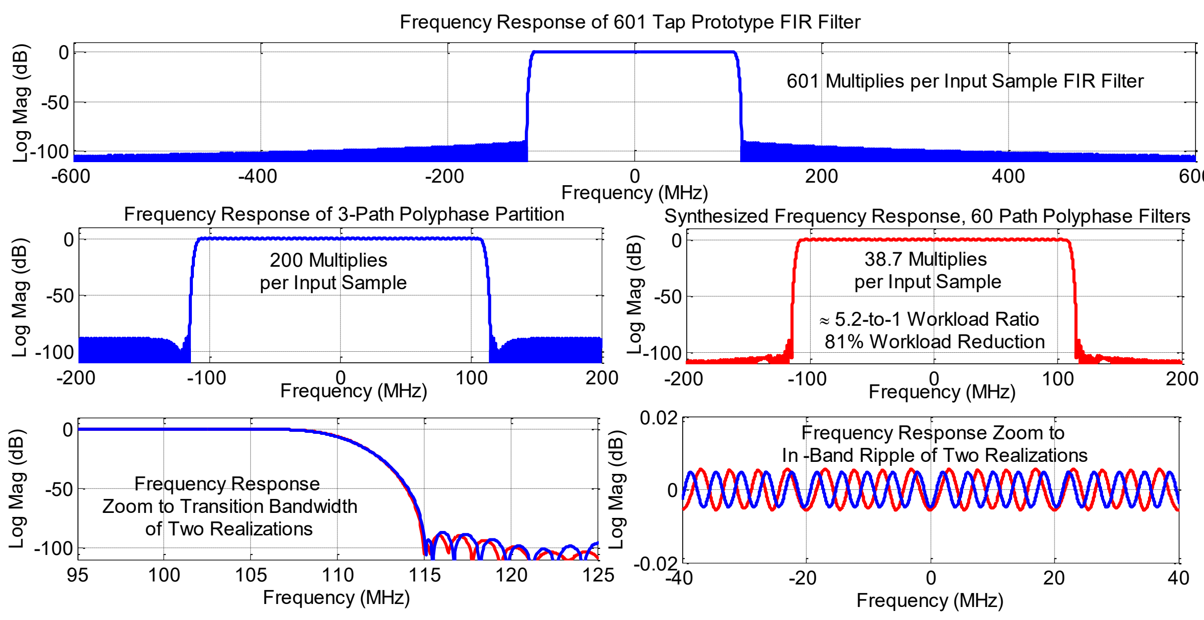

Figure 6.22. Frequency Response of 601 Tap Direct Implementation of Low Pass Filter and of 3-to-1 Down Sampled Direct Filter and 3-to-1 Down Sampled Synthesized Impulse Response from 11-Sub-Channels in Input 60-Path, 30-to-1 Down Sample and Output 20-Path 1-to-10 Up Sample Filter bank with Comparisons of Transition Bandwidths and In-Band Ripple Levels.

6.4.3 Cascade Channelizers for Multiple Simultaneous Variable Bandwidth Filters

In this section we apply the resampling M-path channelizers to assemble and disassemble composite waveforms containing multiple arbitrary bandwidth signal components. We start with the assembly process, developing and implementing a number of useful variants and then demonstrate their dual processes in the disassembly process. We use our 60 channel synthesis channelizer as the framework for the demonstrations. Our 60-path synthesis channelizer forms a composite output signal containing the components from 60 sub-channels separated by 20 MHz center frequencies. Input signal samples presented to the k-th port of the synthesis channelizer are up-sampled by a factor of 30 and translated to the k-th center frequency of the composite output signal. The sample rate of each input sequence is 40 MHz which is twice the spacing between adjacent channels. We want to be clear now what we mean by input signal bandwidth and it would be useful to examine the spectra presented in Figure 6.11. The signal bandwidth to the channelizer includes its passband width plus the width of both transition bandwidths to their stopband edge. Since the input sample rate to each port is 40 MHz, the un-aliased input signal bandwidths must be below 40 MHz, and to accommodate the synthesizer filter's transition bandwidth we restrict the input bandwidth to be 30 MHz.

Figure 6.23 presents the spectra of 4 different QPSK modulated waveforms we want to present to the synthesis channelizer. In the upper left, the spectrum of our first signal input signal is seen to be a shaped baseband modulation signal with a 10 MHz symbol rate exhibiting a 15 MHz bandwidth, sampled at 4-samples per symbol to obtain the desired 40 MHz sample rate. We simply present these samples to a channelizer input port which we will do shortly. In the upper right, the spectrum of our second input signal is seen to be a shaped baseband modulation signal with a 20 MHz symbol rate exhibiting a 30 MHz bandwidth, sampled at 2-samples per symbol to also obtain the desired 40 MHz sample rate. Here too, we simply present these samples to a channelizer input port.

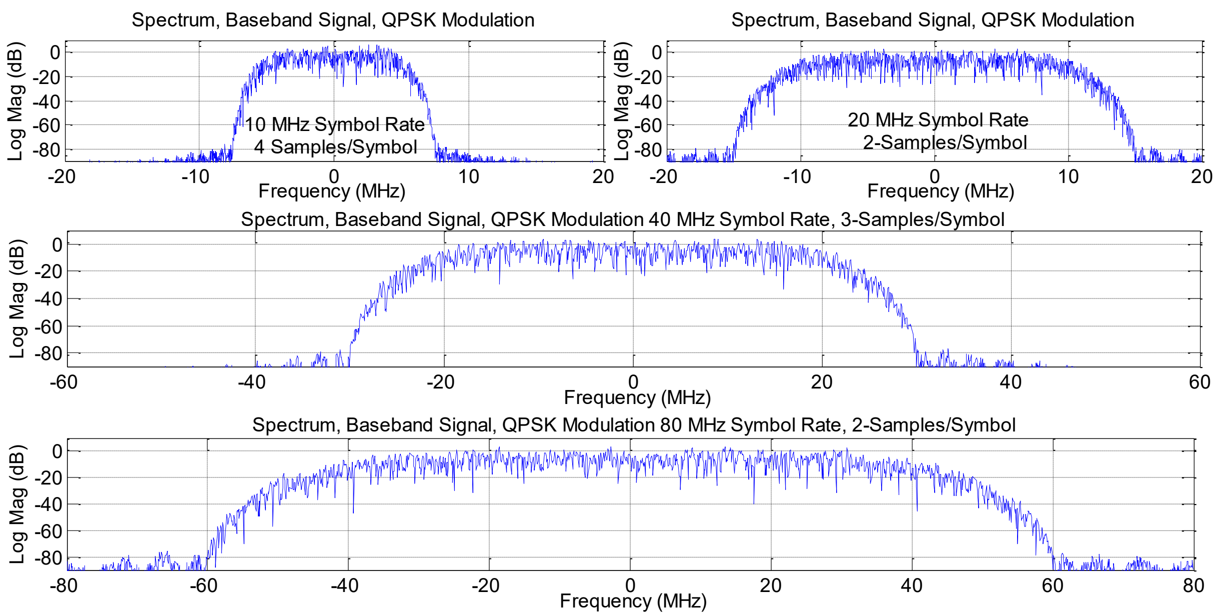

Figure 6.23. Spectra of Four Baseband Modulated Signals, with Symbol Rates of 10, 20, 40 and 80 MHz Sampled at 40, 40, 10, and 160 MHz Respectively. Signals to be Processed and Presented to 60 Path Synthesis Channelizer

In the center row of Figure 6.23, the spectrum of our third input signal is seen to be a shaped baseband modulated signal with a 40 MHz symbol rate exhibiting a 60 MHz bandwidth sampled at 120 MHz. This signal requires processing in a 6-path, 3-to-1 down sampling analysis channelizer. This will form 20 MHz bandwidth output channels at a 40 MHz output sample rate with channels separated by 20 MHz centers.

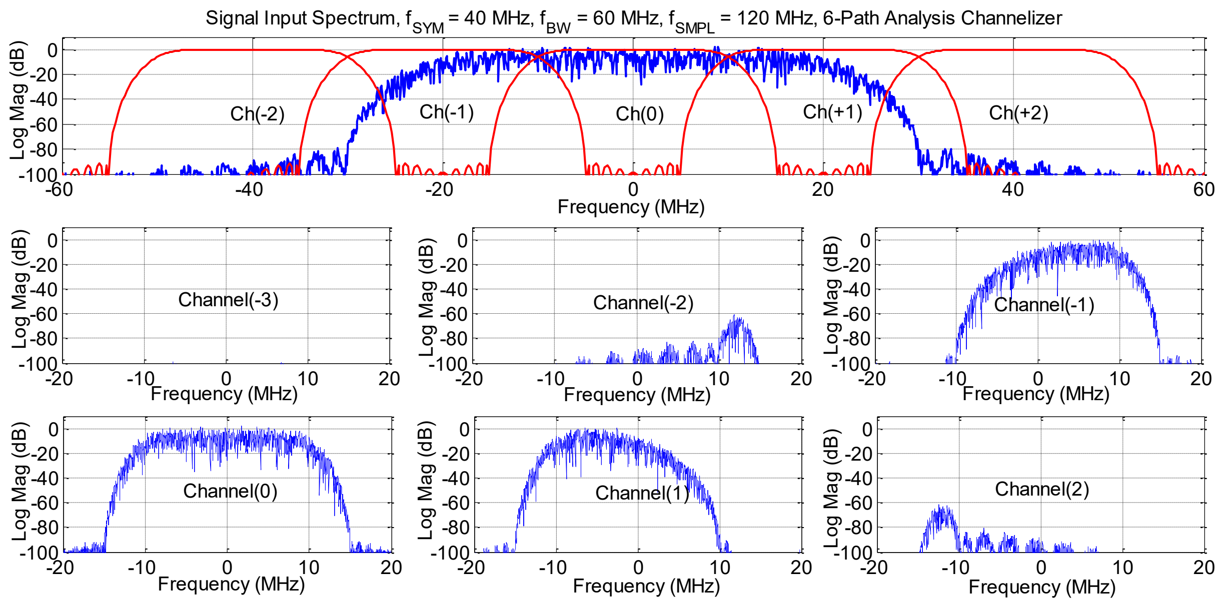

Figure 6.24. Spectra of Input Signal and Channel Frequency Responses of 6-Path Analysis Channelizer and Filtered, Base Banded and Resampled Spectra from Six Channel Filters Illustrating the Spectral Decomposition Performed by 6-Path Analysis Filter Bank

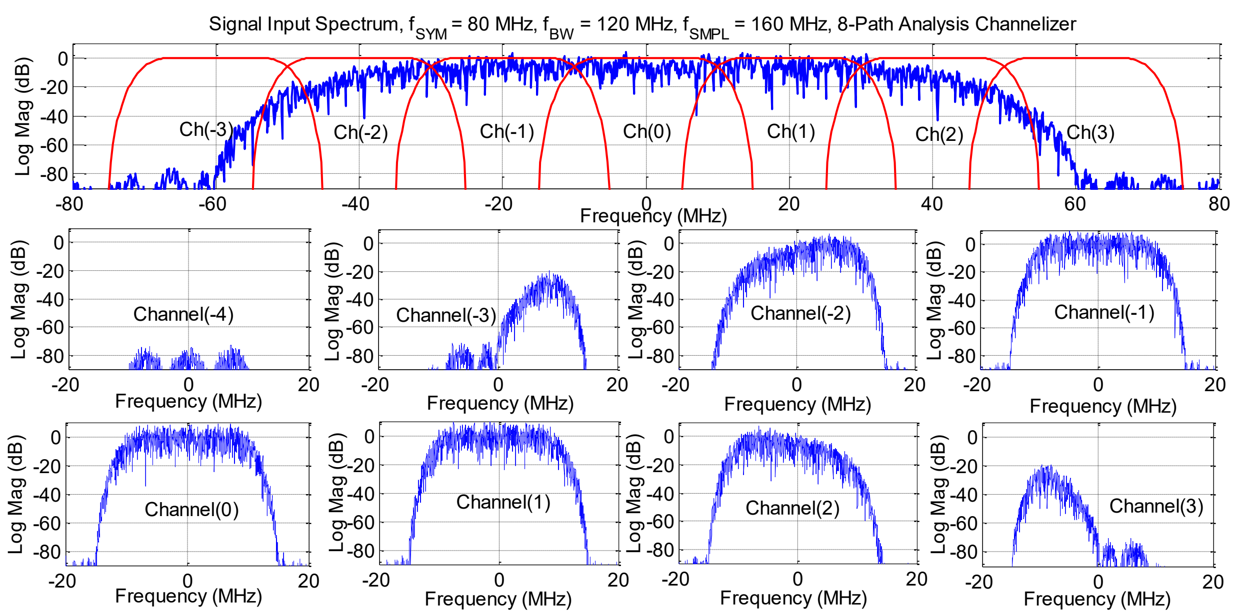

Figure 6.25. Spectra of Input Signal and Channel Frequency Responses of 8-Path Analysis Channelizer and Filtered, Base Banded and Resampled Spectra from Eight Channel Filters Illustrating the Spectral Decomposition Performed by 8-Path Analysis Filter Bank

The spectra of the six path analysis filter bank channel bandwidths and the spectra of the time series from each channel are shown in Figure 6.24. Note that five of the six channels contain spectral components from the analyzed input signal. The time series from the five channels of the analysis filter bank will be presented to five ports of the synthesis filter bank in which the will be reassembled in the five frequency offset channels of the synthesizer.

In the bottom row of Figure 6.23, the spectrum of our fourth input signal is seen to be a shaped baseband modulated signal with an 80 MHz symbol rate exhibiting a 120 MHz bandwidth sampled at 160 MHz. This signal requires processing in an n-path, 4-to-1 down sampling analysis channelizer. This too will form 20 MHz bandwidth output channels at a 40 MHz output sample rate with channels separated by 20 MHz centers. The spectra of the 8-path analysis filter bank channel bandwidths and the spectra of the time series from each channel are shown in Figure 6.25. Note that seven of the eight channels contain spectral components from the analyzed input signal. The time series from the seven channels of the analysis filter bank will be presented to seven ports of the synthesis filter bank in which the will be reassembled in the selected seven frequency offset channels of the synthesizer. Figure 26 shows the spectra formed by the 60-path synthesis channelizer at its output sample rate of 120 MHz along with the spectral segments delivered to it at their 40 MHz sample rates. The narrow bandwidth signals were presented to the channelizer without pre -processing, while wide bandwidth signals were pre-processed to form channelized segments by the 6-path and 8-path analysis filters. The spectrum of the channelizer channel is overlaid in red to show the relative sizes of the synthesized supper channels and the channel bandwidth which assembled them.

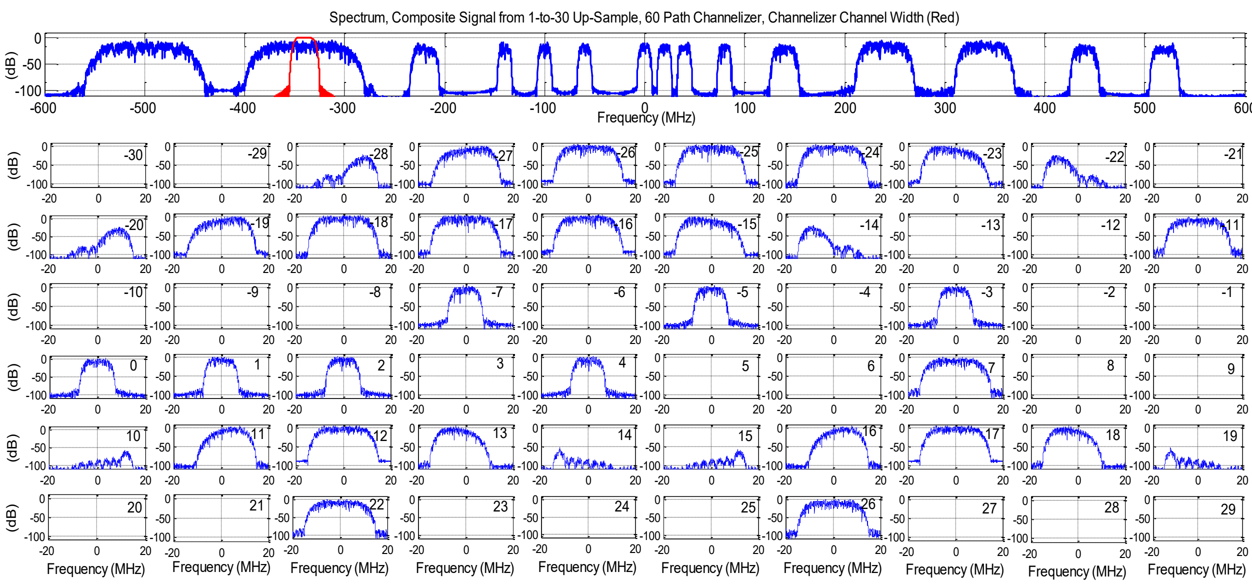

Figure 6.25. Spectra of Input Signal and Channel Frequency Responses of 8-Path Analysis Channelizer and Filtered, Base Banded and Resampled Spectra from Eight Channel Filters Illustrating the Spectral Decomposition Performed by 8-Path Analysis Filter Bank

6.4.4 Cascade Channelizers Enhancements for Increased Flexibility

The cascade M-path analysis and synthesis filter banks, while impressive by themselves, can offer increased capabilities through the aid of minor signal processing tasks residing at the input or output of the bank as well as between the analysis and synthesis channelizers. We will discuss some of these enhancements. Figure 6.26 shows a synthesis filter bank being accessed through a range of signal conditioning processing blocks connecting the external signal source to the channelizer input ports. The first processing block we encounter starting at the top of the input chain is the analysis filter block which channelizes wide band input signals into narrow channels that match the synthesizer's channel bandwidth and spacing. We have already covered this option!

The next option is interesting because it removes one of the synthesis channelizers constraints. The channelizer aliases baseband signal spectra to center frequencies which are multiples of fs/M. This usually corresponds to multiples of the input sample rate, but with of filters performing M/2 to-1 resampling, the center frequencies are multiples of half the input sample rate. We can heterodyne the input signal off of baseband with the complex heterodyne operating at the input sample rate. The amount of frequency shift we apply here is always less than half the channel spacing. If we want a larger shift we would simply use the next input port as the reference frequency. The amount we shift the input signal off of DC is the amount the spectrum is offset relative to the frequency bin center selected as the input port of the M-point IFFT. We call this the worm-hole effect. We perform a small frequency offset at baseband at a low input sample rate and the spectrum nominally at the k-th center frequency operating at the high output sample rate experience the same frequency offset. The dual graph benefits from the same coupled offset property. If a signal center frequency is offset from one of the Nyquist zone center frequencies that aliases to baseband when we alias the center frequency to baseband in an analysis channelizer, that same offset frequency becomes the offset from baseband's DC. A final heterodyne down conversion at the low output sample rate will complete the base-banding down conversion.

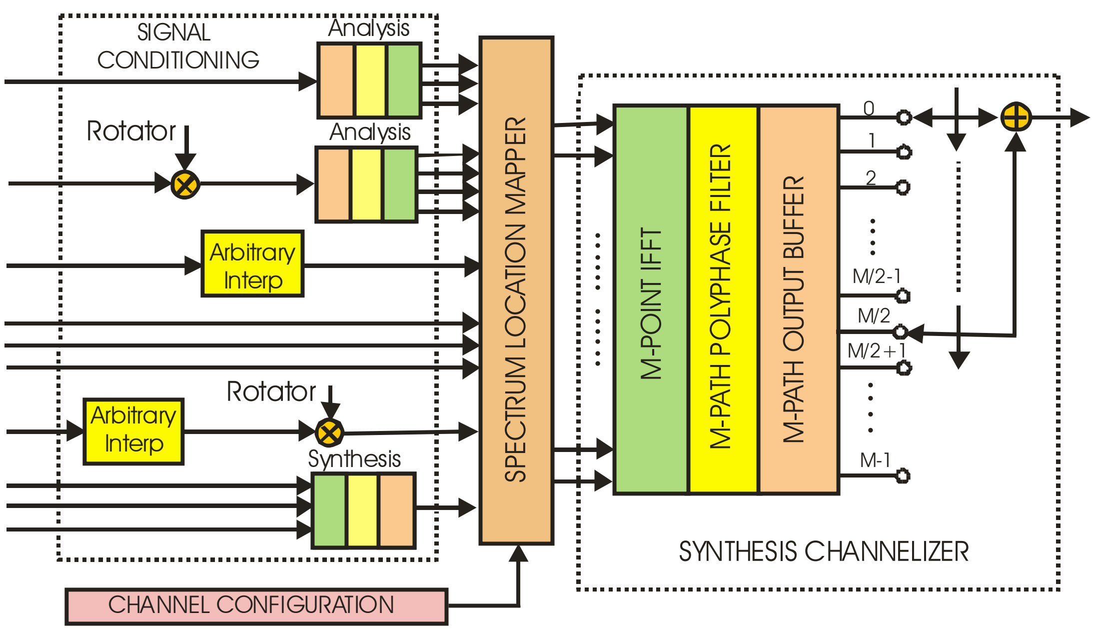

Figure 6.26. Signal Conditioning options at Input to Cascade M-Path Channelizers.

The third signal conditioning option is an arbitrary interpolator which converts the sample rate of the input signal to the required input sample rate of the synthesis channelizer. As an example of the use of this option follows. Suppose we are presented with an input signal with a 10.24 MHz symbol rate generated formed at 4-samples per symbol which gives us a 40.96 MHz sample rate. The channelizer wants a 40.0 MHz sample rate so we use the arbitrary interpolator to change the 40.96 to 40.0. A matching interpolator at the dual analysis channelizer would perform the reverse interpolation, converting the 40.0 MHz output sample rate back to the 40.96 MHz sample rate.

References

[1]. fred harris, “Multirate Signal; Processing for Communication Systems”, Prentice Hall, 2004

[2]. fred harris, Chris Dick, Xiaofei Chen & Elettra Venosa, “Wideband 160 Channel Polyphase Filter bank Cable TV Channelizer”, IET Signal Processing, Special Issue on Multirate Signal Processing, Vol. 5, Issue 3, June 2011, pp. 325-332.

[3]. fred harris, Chris Dick, & Sridhar Seshigiri, “An Improved Square-Root Nyquist Shaping Filter”, SDR Conference-2005, Anaheim, CA, 15-17 November 2005.

[4]. fred harris & Chris Dick, “An Alternate Design technique for Square Root Nyquist Shaping Filters”, SDR WinnCom-2015 Conference, San Diego, CA, 15-17 March 2015.

[5]. fred harris, Elettra Venosa, & Xiaofei Chen, “Polyphase Analysis Filter Bank Down-Converts Unequal Channel Bandwidths with Arbitrary Center Frequencies, Design I”, Software Defined Radio Conference (SDR’10), Washington DC, 30 Nov. – 3 Dec. 2010.

[6]. fred harris, Elettra Venosa, & Xiaofei Chen, “Polyphase Synthesis Filter Bank Up-Converts Unequal Channel Bandwidths with Arbitrary Center Frequencies, Design I”, Software Defined Radio Conference (SDR’10), Washington DC, 30 Nov. – 3 Dec. 2010.