ALGORITHMS FOR ARBITRARY RESAMPLING FILTERS

fred harris (San Diego Sate University, San Diego, CA, fred.harris@sdsu.edu)
Bruce Fette, (General Dynamics DS, Scottsdale, Az,bruce.fette@gdds.com)

1.ABSTRACT

A common task in software defined radio receivers is that of changing the synchronous sample rate of the input modulation process to a fixed asynchronous sample rate required by the output D-to-A converter. Fixing the sample rate at the output D-to-A converter permits a wide range of modulation bandwidths to be serviced by a single fixed bandwidth signal conditioning analog filter. A similar comment is appropriate for signal conditioning at the input to A-to-D converters. The process that changes the bandwidth and sample rate of a digital signal is known as decimation or down sampling and as interpolation or up sampling. It is a fairly simple matter to digitally change the sample rate of a digital signal by a rational ratio. Arbitrary ratio resampling requires an approximation that forms spectral artifacts that are well understood and controllable to arbitrarily small levels by appropriate design considerations. This paper reviews the relationship between filter complexity and performance requirements and then presents an efficient and flexible filter structure to realize arbitrary resampling ratios.

2. INTRODUCTION

Webster's Second Collegiate dictionary lists, in its third entry, a math definition of *interpolate* as: "to estimate a missing functional value by taking a weighted average of known functional values at neighboring points". Not bad! That certainly describes the processing performed in a multirate filter. Interpolation is an old skill that many of us learned in a different era before the advent of calculators and key strokes replaced tables of transcendental functions such as log(x) and the various trigonometry functions. Take for example the NBS Applied Mathematics Series, AMS-55 "Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables" by Abramowitz and Stegan. This publication contains tables listing functional values of sin(θ) for values of θ equal to …40.0, 40.1, 40.2,….,etcetera. Interpolation is required to determine the value of sin(θ) for the value of θ equal to 40.137. Interpolation was such an important tool in numerical analysis that three pages in the introduction of the Handbook are devoted to the interpolation process. Interpolation, an old art, continues in the modern era as an important tool used in modern communication systems. We now discuss where interpolation is used in these systems and then present efficient implementations of the process.

The most common application of interpolators in a modern communication system is the 1-to-4 up sampling operation to which waveform samples are subjected prior to being presented to the digital to analog converter. The increase in output sample rate by a factor of four obtained by the 1-to-4 interpolator increases the spacing between the spectral copies of the periodic spectrum processed by the DAC and subsequent analog smoothing filters. This increased separation allows us to employ a low order analog filter to complete the required filtering action, the spectral suppression of the residual spectral terms seen at the output of the DAC. Remember when the advertisement for CD player proudly proclaimed "4-Times Oversampled"? The Signal processing chain for the CD is shown in figure 1.

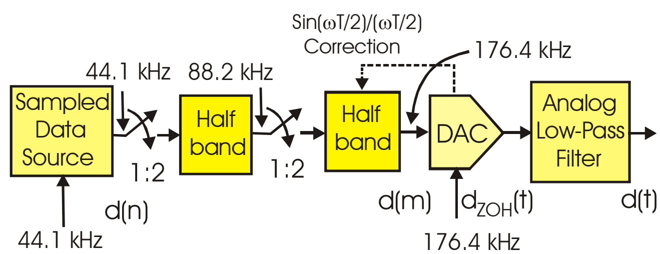

Figure 1 Four-to-One Up Sampling Chain In CD Player

The length of a digital filter is determined by its specification in accord with the relationship shown in figure 2 and in equations 1 and 2. The relationship shown in equation 2 is a good first order approximation for FIR filters designed by windowing or by the Remez algorithm with equal pass band and stop band ripple.

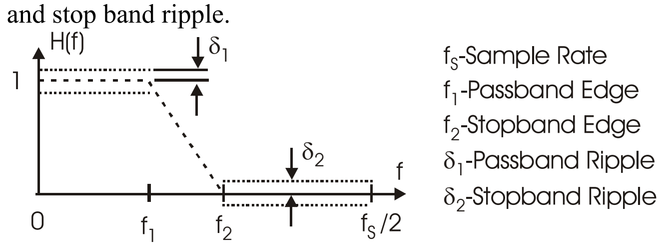

Figure 2 Parameters of Sampled Data Low-Pass Filter

$$N = function(f_S, f_1, f_2, \delta_1, \delta_2) \qquad (1)$$

$$N \cong \frac{f_S}{\Delta f}\,\frac{Atten(dB)-8}{14} \qquad (2)$$

A cascade of two half-band filters is more efficient than a single 1-to-4 up sampler because the second half-band requires significantly fewer coefficients than the first filter. This is because of its wider transition bandwidth commensurate with the increased separation between spectral copies as a result of the up sampling by the first filter. True half-band filters are used in the interpolator to take advantage of the fact that alternate samples in the half-band impulse response are exactly zero hence half the coefficients in the impulse response do not contribute to the processing workload. The first half-band filter is designed to operate at 2fs or 88.2 kHz with a transition bandwidth of 4.1 kHz, and present 100 dB attenuation to the next spectral copy centered at 44.1 kHz. Equation 2 estimated the required filter length to be 141 taps with the actual filter length found to be 133 taps with 66 non-zero weights. The second filter is designed to operate at 4fs or 176.4 kHz with a much wider transition band of 42.2 kHz, and present the same 100 dB attenuation to the next spectral copy centered at 88.2 kHz. Equation 2 predicted the required filter length to be 27 taps with the actual designed filter length required 25 taps with 10 non-zero weights.

The signal flow structure of the two-stage interpolator is shown in figure 3. For each input sample the first stage forms two output samples from its two-port commutator. The workload for the two output samples is 66 multiplies. The second filter also forms two output samples for each input sample delivered to it. For each input sample to the first filter, two samples are passed to the second, which cycles through its output port twice to deliver a total of 4 outputs per input. The second filter requires 12 multiplies per pair of outputs. The total workload for the 4-output samples is 66 + 12 + 12 or 90 which when amortized over the 4 output samples is 22.5 multiplies per output sample. If the designer can take advantage of the even symmetry of the filter weights, the number of multiplies can be reduced by a factor of 2. The impulse response and frequency response of the two filters in the filter chain are shown in figure 4.

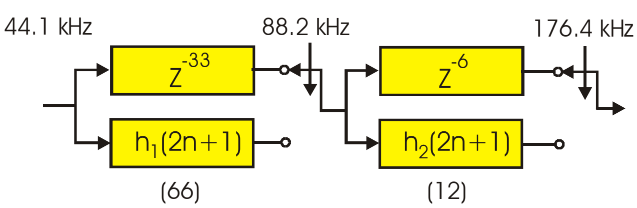

Figure 3 Two Stage 1-to-4 Up Sampler Filter for CD Signals

An up sample by 4 interpolator is also used in the shaping filter path of most digital modulators. In this application, a two stage filter is composed of a 1-to-2 SQRT-Nyquist shaping filter followed by a 1-to-2 interpolating filter that collectively form shaped data samples at 4-samples per input symbol. This structure is shown in figure 5. Here the SQRT-Nyquist shaping filter has an excess bandwidth of 25% and requires its length to extend over 20 symbols to obtain spectral side lobes 60 dB below its main lobe level.

The 1-to-2 interpolator is a half band filter of length 21 with 10 non-zero weights. The cascade forms 4-output samples for each input sample with the shaping filter performing 40 multiplies per input sample and the interpolator performing 10 multiplies per pair of output samples for a total workload of (40 + 10 + 10) or 60 multiplies. Amortizing the 60 multiplies over the 4 output samples we find the workload is 15 multiplies per output sample. Impulse response of the shaping filter and of the interpolating filter along with their respective spectra are shown in figure 6.

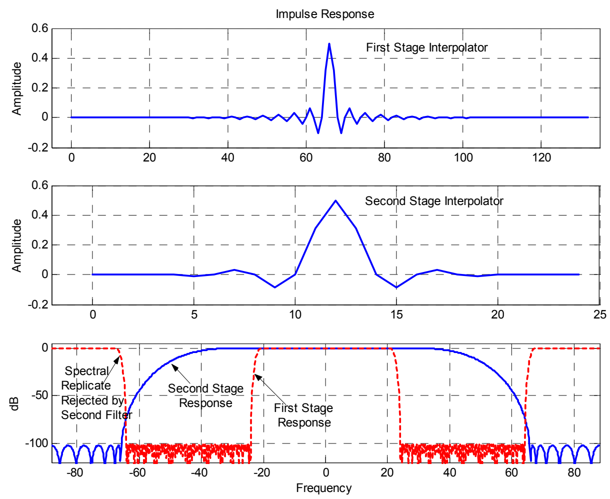

Figure 4 Impulse and Frequency Response Half-Band Interpolators

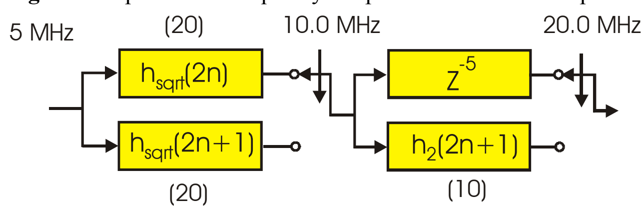

Figure 5 Modulator Shaping Filter at 4-Samples Per Symbol

3. RATIONAL RATIO RESAMPLING

Rational ratio resampling interpolating filters find use in both modulators and demodulators. In modulators they are used to raise the samples rate of data samples from the shaping filter operating at 4-samples per symbol up to the desired fixed output sample rate. In some flexible systems, the input symbol rate is arbitrary, user selected from a specified range such as 10 kHz to 10 MHz, while the output sample rate is fixed by the analog-smoothing filter chosen to satisfy the sample rate of 4-samples per symbol at the maximum symbol rate. Another application requiring an interpolating up sampler is the set of modulators in which the spectral translation to the first intermediate frequency is performed in the DSP domain using digital multipliers as opposed to in the analog domain using matched balanced mixers. Block diagrams presenting these applications are shown in figures 7 and 8. When the up sampling ratio is restricted to 1-to-M, for an arbitrary M, and if the platform is an ASIC or an FPGA, the up sampling is often performed with a variant of the cascade integrator comb filter (CIC) known as the Hogenauer Filter. The up sampling factor "M" of the CIC has an upper limit bounded by the register width of the internal accumulators and a lower limit of 16 or 20 a restriction required to have the pass-band band width span the 4-times oversampled input spectrum. We leave the CIC now and continue with the rational then arbitrary ratio resampling options.

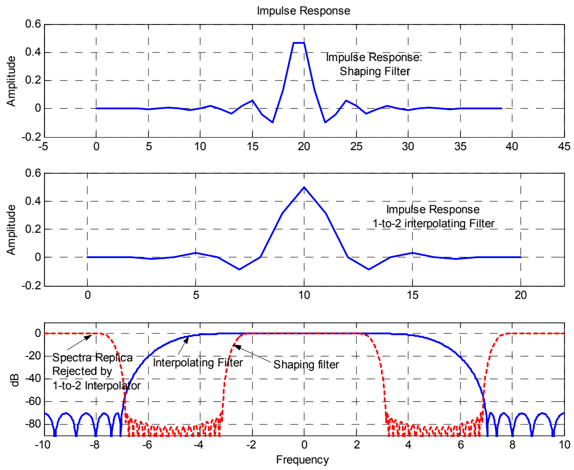

Figure 6 Impulse Responses of Shaping Filter and 1-to-2 Interpolator with their Frequency Responses

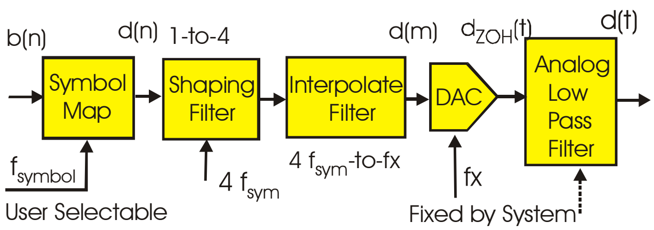

Figure 7 Interpolator Following Shaping Filter to Change User Defined $4f_{sym}$ to Fixed System Clock f_X

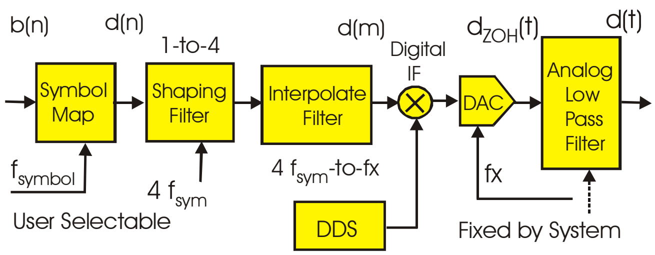

Figure 8 Interpolator Following Shaping Filter to Increase Sample Rate for Digital IF Up Conversion

In the first order approach to up sampling we first zero pack the input data 1-to-M to raise the sample rate and thus access the M-fold multiple Nyquist zones. We then filter the zero packed data with a filter designed to suppress all the undesired spectral replicates. Knowing there is no need to process the inserted zeros we recast the filter into an M-path polyphase filter and then invoke the noble identity to pull the up sampler through the filter so that it resides at the filter output rather than at the filter input. Doing so allows us to perform the filtering at the low input rate rather than at the higher output rate. In the modified form, the up sampling is accomplished as the output commutator cycles through the M-paths of the polyphase partition. We are about to examine the polyphase partition but before doing so we should determine the workload of each polyphase path of the M-path partition. We estimate the workload with the aid of figure 9, which shows the periodic spectrum we would obtain by zero packing a shaped spectrum that had been initially up sampled to 4-samples per symbol. Here the shaped spectrum is assumed to have an excess bandwidth of 0.5 so that the normalized two sided bandwidth is 1.50 with spectral replicates located at integer multiples of 4. The overlaid spectrum indicates the pass band and the transition bandwidth required of the interpolation filter. These specifications are cast explicitly listed in table 1. Remember this M-path filter is changing the normalized sample rate from 4 to 4 M.

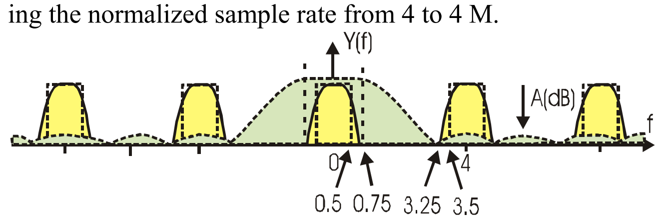

Figure 9. Periodic Input Spectrum and Filter Response of 1-to-M Interpolator Filter

Parameters	Values
Pass Band Ripple	0.1 dB
Stop Band Attenuation	60 dB
Pass Band Frequencies	0-to-0.75
Stop Band Frequencies	3.25-to-2 M
Sample Frequency	4 M

Table 1. Filtering Specifications for M-Path Polyphase Filter

An estimate of the number of filter taps required to implement the 1-to-M interpolating filter is obtained from standard estimate such as presented in equation 2. A variant of this equation, valid for 0.1 dB in-band ripple, is shown in equation 3, which when applied leads to the estimate shown in equation 4. When we partition the filter into M paths as shown in equation 5, we obtain the ratio N/M, which is the number of coefficients per path. Here we see that the number of taps is a small number, either 5, a conservative choice or 4, an option accessible by widening the transition bandwidth to overlap the replica spectra. We can also ask the Remez algorithm to form multiple stop bands spanning the locations of the spectral replicates with don't care intervals between the stop bands and thus allow the algorithm to redistribute stop band zeros to frequency intervals that need the suppression. After all, why hold down a frequency span known not to contain energy by virtue of the previous interpolating filter attached to the shaping filter?

$$N \cong \frac{f_S}{\Delta f} K(\delta_1,\delta_2) \cong \frac{f_S}{\Delta f}\frac{Atten(dB)}{22} \tag{3}$$

$$N \cong \left(\frac{4M}{3.25-0.75}\right)\frac{60}{22} \approx 4.4M \tag{4}$$

$$\frac{N}{M} = 4.4 \tag{5}$$

In either case, the interpolating M-path filter has a small number of taps per path. We still have to determine M, the number of paths. Figure 10 presents the periodic spectra and a detailed zoom we would see if we had zero packed the input signal along with the frequency response of the M-stage interpolating filter for the specific case M=32. Figure 11 presents the spectra observed at the output of the 1-to-32 interpolating filter. The structure of the M-path polyphase filter is shown in figure 12.

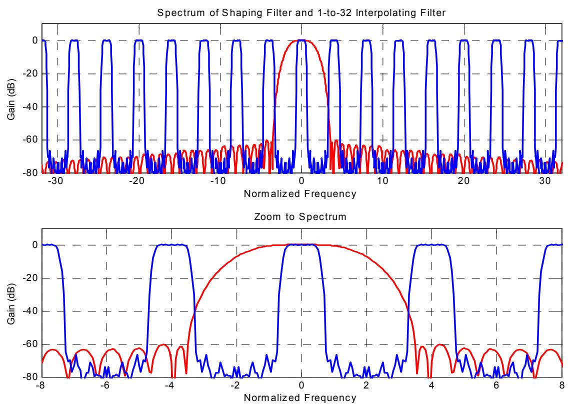

Figure 10. Spectrum of Interpolator and Periodic Spectrum of Zero-Packed Shaping Filter

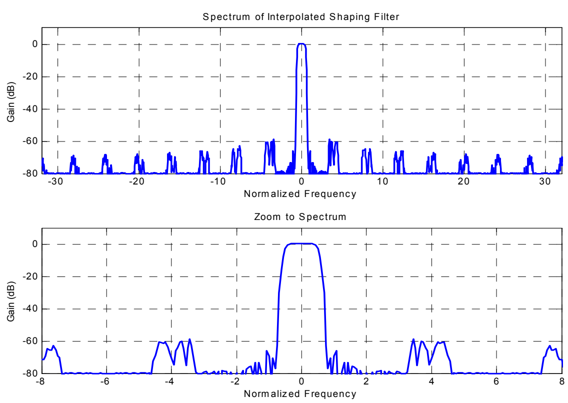

Figure 11. Spectrum of 1-to-32 Interpolated Shaping Filter

We note the polyphase arms contribute their outputs one at a time as the commutator points to successive output ports. We also note that the separate filters all contain the same input data and only differ by their unique coefficient sets. We can replace the M-path version of the polyphase filter with a single stage filter with M-coefficient sets that are sequentially presented to the filter to compute successive outputs. This structure is shown in figure 13. What we have accomplished here is first move the input commutator that performed the input zero packing to the output of the filter where it selected successive filter outputs and then again moved the commutator to coefficient memory bank where it selected successive coefficient sets rather than filter outputs.

While on the topic of selecting filter coefficient sets from a memory bank, we realize that we have the option to skip filter sets and thus perform simultaneous down sampling along with the up sampling. This option is suggested in figure 14. The up M, down Q, rational ratio resampling proceeds as follows: an input sample arrives and we compute M/Q output samples by incrementing through weight sets in memory in stride of length Q.

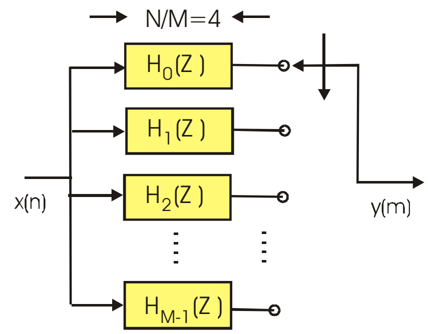

Figure 12 Polyphase Partition of M-Path Resampling Filter

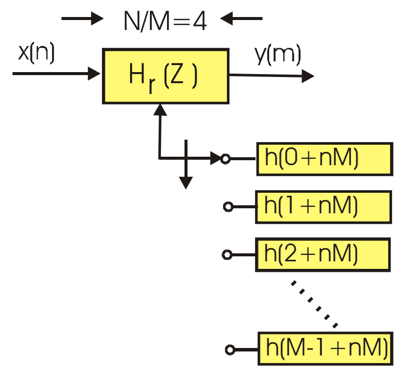

Figure 13. Efficient 1-to-M Polyphase Interpolator

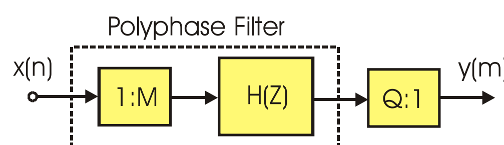

Figure 14 Rational Ratio Resampling by following 1-to-M Up Sampler with Q-to-1 Down Sampler

4. ARBITRARY RESAMPLING INTERPOLATORS

The increment by Q addressing described in the last section is performed modulo M, the number of available stages. When the modulo operation is invoked, the address wraps to or past the top of the address stack. The wrap indicates that we have crossed the boundary that defines the interval between input samples and that a new input sample must be delivered to the data register. Rational ratio resampling can be converted to arbitrary ratio resampling by allowing Q to be a non-integer and then bringing into play a mechanism to accommodate the fractional part of the address shift. The fractional part of the desired address increment can be visualized with the aid of figure 15.

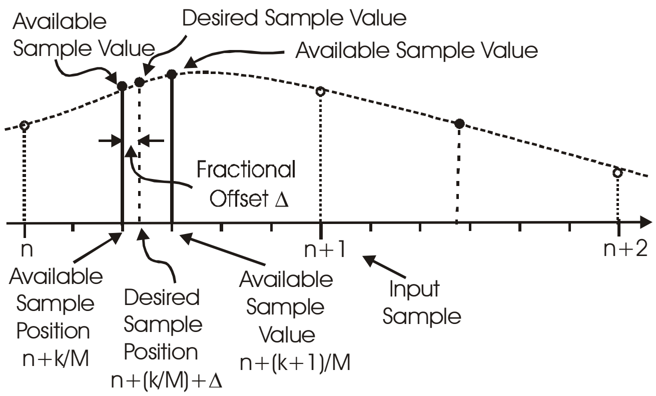

Figure 15 Interpolating to a Position Between Available Output Points in an M-Path Interpolator

A simple option for arbitrary resampling with a polyphase interpolator is to assign the sample value obtained from the nearest neighbor sample location. This nearest neighbor replacement is shown in figure 16. In practice, nearest neighbor can be replaced by the neighbor to the immediate left. This is equivalent to truncating the desired offset value rather than rounding the desired offset value.

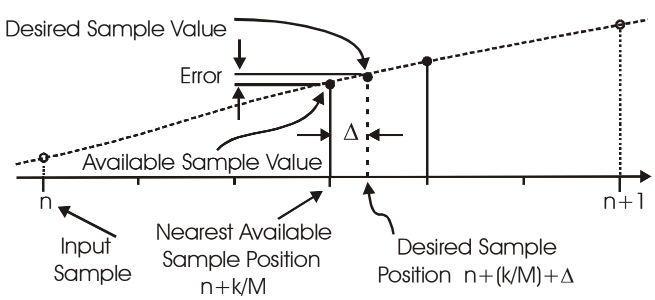

Figure 16 Replacing Desired Sample Value with Nearest Neighbor Sample Value

The sample value errors formed over successive output samples are modeled as timing jitter errors. The local slope and the time difference between adjacent interpolated samples bound the amplitude of these errors. This bound can be made arbitrarily small by increasing the sample rate and reducing the interval between adjacent interpolated values. When the amplitude errors are smaller than the errors due to amplitude quantization of the sample values, the timing jitter errors do not degrade the quality of the interpolated samples. A simple way to estimate the size of the timing jitter errors is to describe the errors due to nearest neighbor replacement as an equivalent linear process and then estimate the errors in that process. We do this by imagining that we interpolate the input series to the maximum output sample rate of M-time sthe input rate to form an analog signal with a perfect DAC or zero order hold, and then resample this virtual analog waveform at the desired time locations. This model is shown in figure 17.

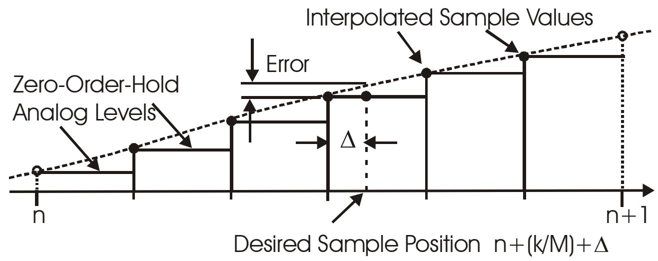

Figure 17 Timing Jitter: Sampling of Virtual DAC Output for Interpolated Sample Values

Our upper bound to the errors due to the timing jitter is found by examining the spectrum of the signal obtained from maximally up sampled virtual DAC. In this analysis we model the spectrum of the original sampled signal as uniform and of unity bandwidth, which is a reasonable approximation to a Nyquist shaped spectrum in a communication system. A signal originally formed with a sample rate that is 4-times the input bandwidth is now up sampled to a new sample rate N times its Nyquist rate with an N/4 stage interpolator. The important parameter here is the output sample rate N. The spectrum of the up-sampled signal at the input and output of the DAC processed signal is shown in figure 18. The frequency response of the DAC is the standard sin(x)/x, shown in eq-6, with zeros located at multiples of the output sample rate. These zeros suppress the spectral copies centered at multiples of the sample rate but leave a residual spectrum in the neighborhood of the spectral zero.

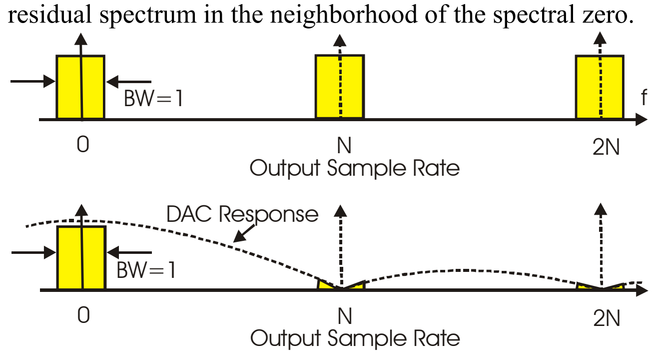

Figure 18 Spectrum of Up-Sampled Signal at Input and Output of Virtual DAC

The DAC spectral response shown in equation 6, and its first derivative is shown in equation 7. Evaluating equation 7 at the first zero crossing of the spectrum, at f = 1/T, we obtain the results shown in equation 8.

$$H(f) = \frac{\sin(2\pi f \frac{T}{2})}{(2\pi f \frac{T}{2})} \tag{6}$$

$$\frac{d}{df}H(f) = \frac{(2\pi f \frac{T}{2})(2\pi \frac{T}{2})\cos(2\pi f \frac{T}{2}) - (2\pi \frac{T}{2})\sin(2\pi f \frac{T}{2})}{(2\pi f \frac{T}{2})^2} \tag{7}$$

$$\left.\frac{d}{df}H(f)\right|_{f=\frac{1}{T}} = -T = -\frac{1}{f_S} \qquad (8)$$

The Taylor Series expansion of the DAC's spectral response at the first zero crossing , f_S or 1/T, is shown in equation 9.

$$H(\Delta f) = -\frac{1}{f_S}\Delta f \qquad (9)$$

Substituting the sample rate indicated in the normalized frequency axis presented in figure 24, we obtain the local Taylor series shown in equation 10.

$$H(\Delta f) = -\frac{1}{N}\Delta f \qquad (10)$$

A zoom to the spectral response of the DAC in the neighborhood of the first spectral zero is shown in figure 19.

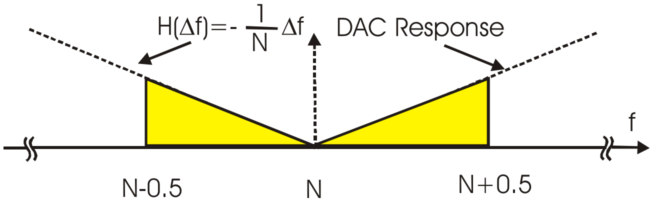

Figure 19 Frequency Response at DACs First Spectral Null

The maximum amplitude of the residual spectrum centered about the first spectral null, obtained by substituting 1/2 for Δf, is seen in equation 11.

$$|H(\Delta f)_{MAX}| = \left.\frac{1}{N}\Delta f\right|_{\Delta f=1/2} = \frac{1}{2N} \qquad (11)$$

The smallest resolvable signal level of a b-bit quantizer is 2^{-b}. If the residual spectral levels at the output of the DAC are below this level, the errors attributed to the timing jitter of the nearest neighbor interpolator is below the quantization noise level of quantized signal samples. To assure this condition, the maximum spectral level of the residual spectrum must satisfy the condition shown in equation 12.

$$\frac{1}{2N} < \frac{1}{2^b} \text{ or } N > 2^{(b-1)} \qquad (12)$$

The virtual analog signal described by the up sampling condition satisfying equation 12 can now be sampled at any output rate that satisfies the Nyquist criterion for the input bandwidth. The aliasing terms that fold into the primary Nyquist zone are the multiple residual spectra residing at the successive spectral nulls of the DAC interpolator. The amplitudes of the aliasing terms inherit the alternating signs and 1/M gain terms of the DACs sin(x)/x. The alternating signs of the successive alias terms limit the composite spectral growth to unity. Thus the collective aliases of all the residual spectra from the DACs spectral zero crossings to the Nyquist interval of the new sample rate do not rise above the highest level of 1/(2N).

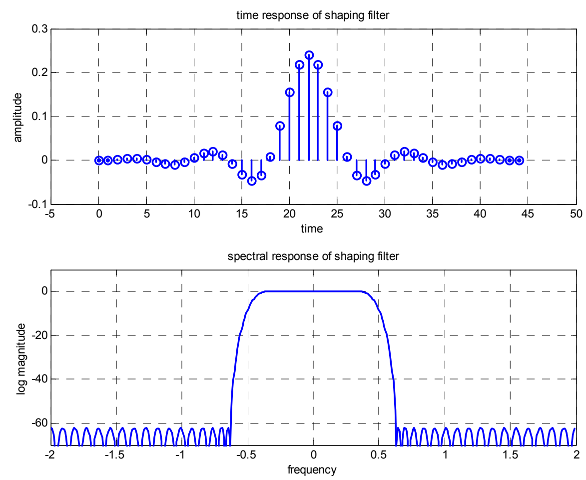

Figure 20 Shaping Filter: Time and Frequency Response Four Times Oversampled

By way of example, consider interpolating a time series represented by 8 bit samples. For this case, if we operate the interpolator to obtain a maximum output sample rate 128 times the signal bandwidth, the noise spectrum due to nearest neighbor interpolation will be below the noise caused by the signal quantization process. If the input signal is originally oversampled by a factor of 4, the interpolator must make up the additional factor of 128/4 or 32. Thus a 32-stage interpolator can resample an 8-bit input signal with jitter related noise levels below the -48 dB dynamic range noise level of the 8-bit quantized data. Figure 20 presents the time and spectral response of a 45-tap filter response that is initially oversampled by a factor of 4. Note the spectrum is scaled for two sided 6-dB bandwidth of "1" for which the sample rate of "4" is presented on an axis of ±2.

Figure 21 presents the time and frequency response obtained by applying the interpolation process to resample the time response of figure 20. The sample rate change shown here is 32/6.4, a sample rate change of 5. This sample rate change is obtained by the stepping through the 32 output commutator port indices by the integer part of an accumulator that is incremented in steps of 6.4. The number of output samples formed by the interpolation process is seen to be 245 points. The spectrum obtained by the interpolation process clearly shows the 4-spectral regions at multiples of the input rate that have been suppressed by the sin(x)/x response of the virtual DAC employed by the nearest neighbor interpolator. The level of suppression is 50 dB, 2-dB more than the maximum 48 dB level estimated in eq-12. This apparent excess attenuation is due to the fact that the spectral amplitude at the 0.5 band edge is less than unity due to its transition roll off. The reduced spectral occupancy lowers the Δf in eq-11 and results in a reduced level estimate for the maximum amplitude spectral residue.

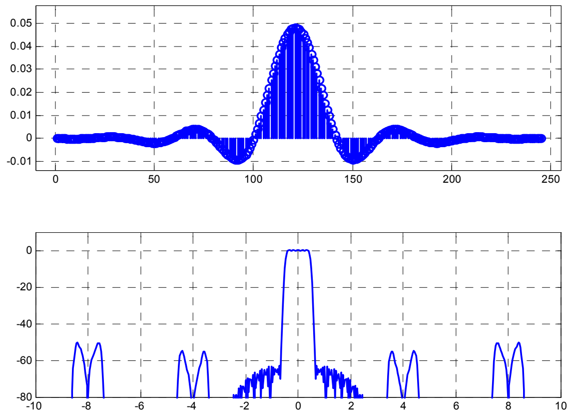

Figure 21. Time and Frequency Response of 32/6.4 Nearest Neighbor Interpolator

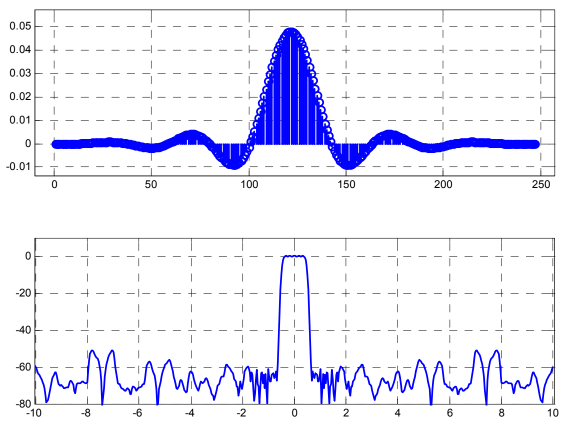

Figure 22. Time and Frequency Response of 32/6.37 Nearest Neighbor Interpolator

For comparison figure 22 shows the time and spectral response when the same 32-stage interpolator is incremented in steps of 6.37 to obtain a sample rate change of 32/6.37 or 5.0235. The slighter higher sample rate presents 246 instead of the 245 output samples of the previous example. We note here that the aliases no longer fold to four common frequencies and now appear over the entire span of the new Nyquist interval of width 4* 5.0235 or 20.094.

5. CONCLUSIONS

We have shown that nearest neighbor interpolation in a polyphase filter bank containing a sufficient number of paths can control the level of spectral artifacts due to the timing jitter caused by the nearest neighbor timing approximation. Since many signals presented to an interpolator have already been up sampled by a factor of 4, the required amount of additional up sampling, or the number of polyphase paths is surprising small, only 32 paths to obtain 48 dB suppression. The advantage of having the input spectra initially oversampled by 4 is that the length of each polyphase interpolator path is only 4, or occasionally 5 taps. Thus the arbitrary interpolator only requires a 4-tap or a 5-tap filter with an access to 32 weight sets. We note that the number of weight sets increases by a factor of 2 for every additional 6-dB of spectral suppression. We have implemented systems with 256 sets of weights. Remember, weight sets only use memory, not processing power. In pursuit of significantly more spectral suppression, say 100 dB, we can exchange memory for computation in the following manner. We achieve 100 dB suppression levels with a polyphase filter containing 48 paths and then use this filter to interpolate to both nearest neighbors, the floor integer offset to the left and the ceiling integer offset to the right and then linearly interpolate between these interpolated samples with the fractional part of the accumulator address. This requires us to form two interpolated samples per desired output sample.

6. REFERENCES

fred harris, "A Fresh View of Digital Signal Processing for Software Defined Radios, Part I and Part II", International Telemetry Conference (ITC), San Diego, CA, 21-24 October 2002.

fred harris and Chris Dick, "Performing Simultaneous Arbitrary Spectral Translation and Sample Rate Change in Polyphase Interpolating and Decimating Filters in Transmitters and Receivers", 2002-Software Defined Radio Technical Conference, San Diego, CA, 11-12 November 2002.

fred harris and Michael Rice, "Multirate Digital Filters for Symbol Timing Synchronization in Software Defined Radios", IEEE Journal on Selected Areas in Communications, Vol. 19, pp. 2346-2357, Dec. 2001.

Michael Rice and fred harris, "Polyphase Filter Banks for Symbol Synchronization in Sampled Data Receivers", MILCOM-2002, Anaheim, CA, 7-10 October 2002

Ronald Crochiere and Lawrence Rabiner, " Multirate Signal Processing", Prentice-Hall Inc. Englewood Cliff, NJ, 1983.

Norbert Fliege, "Multirate Digital Signal Processing: Multirate Systems, Filter Banks, Wavelets", John Wiley & Sons, Ltd, West Sussex, 1994.

Gordana Jovanovic-Dolecek, "Multirate Systems: Design and Applications", Idea Group, London, 2002.

P. P. Vaidyanathan, "Multirate Systems and Filter Banks", Prentice-Hall Inc. Englewood Cliff, NJ, 1993.

Sanjit Mitra and James Kaiser, "Handbook for Digital Signal Processing", John Wiley & Sons, NY, 1993

fred harris, "Multirate Signal Processing for Communication Systems: *Current Practice and Next Generation Techniques*", Prentice Hall, 2004.