CARMA Memorandum Series #12

Digital FIR Filtering Options for CARMA Digitizers

Kevin P. Rauch
University of Maryland
September 29, 2003

ABSTRACT

Design possibilities are examined for the implementation of narrowband CARMA spectral-line modes using digital filtering techniques. The model designs are based on an analog pre-filter followed by one or more stages of digital filtering and decimation that narrow and shape the input analog bandpass to create the desired observing window. The pre-filter is the same one used to construct the narrowest analog-based spectral-line mode. It is found that for a 62 MHz pre-filter, 31 MHz, 8 MHz and 2 MHz observing bands exhibiting < 0.5 dB of passband ripple (peak-to-peak, analog pre-filter excluded), 30 dB of stopband rejection, and a relative transition width of $\sim$1/64 (per edge) require $\approx 50\%$ of the digital logic available in one COBRA digitizer FPGA. It is therefore possible to implement the corresponding spectral modes using COBRA digitizer cards without resorting to narrowband analog filters. Requantization after digital processing is found to reduce efficiency by 3-6%, depending on the number of bits retained between decimation stages. Assuming as-is reuse of COBRA hardware, the 31 MHz mode is feasible only if digital frequency modulation is used. Spectral bandwidths < 2 MHz are also easily supported in designs based on frequency modulation (with small additional reductions in efficiency).

Change Record

Revision	Date	Author	Sections/Pages Affected
	Remarks		
1.0	2003-Jun-13	Kevin Rauch	
	Initial release.		
1.1	2003-Sep-29	Kevin Rauch	mainly Sec. 2,3
	Added detailed FIR processing pipeline and simulation results (including two figures and two tables).		

1. CARMA First-Light Digitizer Cards

As part of the proposed CARMA first-light correlator plan (Beasley, Woody, and Hawkins 2002; Beasley, Hawkins, Rauch, and Woody 2003), reuse of both COBRA correlator cards and COBRA digitizer cards is envisioned. A COBRA digitizer card contains two digitizer modules, each of which samples one analog telescope signal at a rate of 1 GHz (with 2 bits of precision), and thereby supports observing windows up to 500 MHz wide. The first-light plan includes both this 500 MHz wideband observing mode and a series of spectral-line modes ranging in bandwidth from 250 MHz to 2 MHz. A detailed analysis of correlator card performance for the proposed spectral-line modes can be found in Rauch (2003).

The downconverters place the 500 MHz wideband input at 500 MHz to 1 GHz in absolute frequency to avoid electronics issues at low frequencies. To prevent aliasing at a 1 GHz sample rate, analog filters are used to remove signal power below 500 MHz and above 1 GHz before feeding the signal to the digitizer modules. The act of sampling then neatly aliases the signal down to baseband (0-500 MHz). One method of implementing the CARMA spectral-line modes is to replace the 500 MHz filter with a narrower analog filter centered somewhere in the 500-1000 MHz region; so long as the nominal filter band edges are located at integer multiples of its passband width, the sampled, band-limited signal will still neatly alias down to its corresponding baseband location. This is the method by which the 250 MHz and 125 MHz spectral-line modes will be implemented. However, it becomes progressively more difficult (i.e., expensive) to extend this approach to the narrowband modes—accommodating their finer channelization implies relative transition widths sharper than practical analog filters can achieve. In addition, analog filters consume precious board space inside the spectral downconverter, hence it is desirable to minimize their number.

To avoid the issues associated with narrowband analog filtering, a digital filtering approach can be used. In this scheme, the digitized analog signal is processed by an FPGA (a programmable logic device) on the digitizer card to remove unwanted frequency components before decimating the signal to a Nyquist-limited sample rate (i.e., twice the final spectral bandwidth). The rate-reduced signal can then be transmitted to the correlator cards for cross-correlation analysis. This is the method being proposed for the 2 MHz, 8 MHz, and 31 MHz spectral-line modes. The purpose of this memo is to determine the feasibility of this option assuming the reuse of COBRA digitizer hardware. The results can be easily generalized to estimate logic requirements under other circumstances, such as for the next-generation CARMA correlator.

2. Digital Filtering and Decimation

2.1. Linear Digital Filters

Digital filtering can be performed in either the time or frequency domain. However, for a lag correlator receiving a continuous stream of samples in real-time, only the time domain approach is feasible for the sample rates of interest in radio astronomy. The general form of a linear digital filter is

$$y_n = \sum_{i=-\infty}^{\infty} a_i x_{n-i} + \sum_{i=1}^{\infty} b_i y_{n-i},$$

where x_i are the input samples, y_i are the filtered output samples, and the constants a_i and b_i are the filter coefficients. Through suitable choice of coefficients, low-pass, high-pass, and band-pass filters, as well as integrators, differentiators, etc., can be produced. For real-time work, the filter must be *causal*—that is, the output y_n must not depend on future samples x_k, where $k > n$; this implies that $a_i = 0$ for $i < 0$. In addition, for practical filters only a finite number of coefficients can be non-zero. Hence in practical real-time work

$$y_n = \sum_{i=0}^{M} a_i x_{n-i} + \sum_{i=1}^{N} b_i y_{n-i},$$

where M and N define the *orders* of the filter. Filters for which $N = 0$ are called finite impulse response (FIR) filters because if $x_n = 0$ for all $n > n_0$, then $y_n = 0$ for all $n > n_0 + M$. When $N > 0$ the filter is an infinite impulse response (IIR) filter since (in general) the recursive feedback from previous output samples y_k is able to keep $y_n \neq 0$ as $n \to \infty$ even if $x_n = 0$ for all $n > n_0$.

Although IIR filters generally require fewer coefficients than FIR filters to achieve a specified response, FIR filters offer two important advantages: first, they can be made to have exactly linear phase, meaning that the group delay between input and output is independent of the frequency content of the signal; second, they are always numerically stable, as well as less susceptible to degradation due to finite-precision arithmetic—an especially important consideration for implementation in programmable logic devices, where minimizing arithmetic precision also minimizes the logic requirements of the resulting circuit. Therefore, only linear-phase FIR filters will be considered here. Note that the linear phase constraint implies symmetry within the coefficients a_i; there are only $M - \lfloor M/2 \rfloor$ unique values for an order-M filter.

2.2. Decimation Methods

In the present case, the purpose of the digital filtering is to facilitate reduction of the sample rate from 1 GHz to twice the bandwidth of the corresponding spectral-line mode (the Nyquist-limited rate) with acceptably low aliasing of out-of-band power into the desired observing window, and with acceptably flat response within the window itself. These criteria translate directly into tolerances for passband ripple and stopband rejection of the digital filter. In addition, the transition between passband and stopband should be sharp enough to preserve the integrity of the frequency channels at each edge of the passband.

It is important to keep in mind that the signal processing objective is decimation, not filtering per se. In other words, the most effective design process is not to create a generic digital filter whose output happens to be suitable for decimation; rather, the specialized design methods applicable only to decimators drive the filtering approach. For the narrowband CARMA spectral-line modes in particular, the task is not only decimation, but decimation by a large factor (256-to-1 for the 2 MHz spectral mode and 64-to-1 for the 8 MHz mode). Highly efficient implementations of L-to-1 decimators, where $L \gg 1$, can be created using a multirate, multistage technique (e.g., Crochiere and Rabiner 1983). In this approach, the anti-aliasing filter and decimation is implemented as a cascade of several independent decimator stages. If there are I stages, where stage i implements an L_i-to-1 decimator and $L = \prod_{i=1}^{I} L_i$, then the cascade of I stages together constitute an L-to-1 decimator. To understand the advantage of the multistage approach, note that the order of anti-aliasing (FIR) filter needed to obtain an *absolute* transition width of Δf scales as $M \propto f_s/\Delta f$, where f_s is the sampling frequency at the input of the filter. Since in the early stages of a multistage stage implementation the passband of interest occupies only a small fraction of the Nyquist interval (assuming $L \gg 1$), *and* since significant decimation will be done by succeeding stages, the early anti-aliasing filters need only prevent aliasing into the narrow, final passband—there is no need to curtail aliasing into *all* frequencies below $f_s/(2L_i)$, as would be the case for a generic L_i-to-1 decimator. Hence the early anti-aliasing filters can have a much larger transition width $\Delta f_i \gg \Delta f_0$, where Δf_0 is the transition width required by the final spectral window; as a consequence, the early filters are quite small and simple. Although the transition width of the filter for stage I must still be Δf_0, by this time the sample frequency f_s has been reduced by a factor of $L/L_I \gg 1$, and thus the final filter's order can be reduced by this same amount. In this way, the cascade in total can require many fewer coefficients than a single-stage decimator would, assuming $L \gg 1$ (and that L can be factored into suitably small cofactors L_i). As an added bonus, these smaller filters are less subject to round-off errors than a single large filter would be, and their coefficients can be quantized to fewer bits.

Consider now the implementation of a single L-to-1 decimation stage consisting of an anti-aliasing FIR filter followed by rate compression by a factor of L (in which every L-th output sample is retained and the rest discarded). Arranged in this form, the FIR filter must operate at the input sample rate—only to have all but a fraction $1/L \ll 1$ of the outputs thrown away by the rate compressor; this is very inefficient. Using the *polyphase* computational structure, however, it is possible to integrate the rate compression into the filtering process so that calculations are performed at the final sample rate instead of the much higher input data rate. The polyphase structure consists of separating the M coefficients a_i of the original anti-aliasing filter into L parallel sub-filters, each of length $P = M/L$. The P coefficients c_i^k of the k-th sub-filter (where k runs from 0 to $L-1$) are the subset of original coefficients a_i for which $i \bmod L = k$—i.e., $\{c^0\} = \{a_0, a_L, a_{2L}, ...\}$, $\{c^1\} = \{a_1, a_{L+1}, a_{2L+1}, ...\}$, and so on. Conceptually, L consecutive input samples are distributed one at a time to the individual sub-filters, each of which therefore operates at only $1/L$ of the input rate (i.e., at the output rate). The final output of the L-to-1 decimator is formed by adding together the L sub-filter outputs. Use of the polyphase structure is particularly important for implementation in a programmable logic device since it relaxes the timing requirements of the filtering circuit by a factor of L.

An alternative approach to achieving large decimation ratios is through the use of so-called *interpolated* FIR filters (e.g., Dick 1998). Recall that an L-from-1 interpolator increases the sampling rate of an input data stream by inserting $L-1$ zeros between each of the original sample points, and then applying an anti-imaging filter to remove the $L-1$ images of the original spectral band created by the zero insertion; hence it is the inverse of an L-to-1 decimator. An interpolated FIR (IFIR) filter operates similarly to an L-from-1 interpolator except that the initial upsampling is applied to the filter coefficients instead of the input data. The insertion of $L-1$ zeros between successive coefficients of an FIR filter (of order M, say) has two effects. First, the order of the filter is increased by a factor of L—and hence (as one would intuitively expect) the original filter response is *compressed* toward zero frequency by a factor of L. In addition, however, the interpolated filter response contains $L-1$ equally-spaced images of this compressed response (extending to the sampling frequency). Hence IFIR filters must be followed by a second, anti-imaging filter to remove these response images. The net result is a narrowband filter whose effective order is ML, yet whose computational requirement is equivalent to an order M filter (not including the cost of the anti-imaging filter, which can be comparable).

IFIR filters can be combined with the polyphase structure to produce effective large-ratio decimators. This relies on use of the "Noble Identities" (e.g., Vaidyanathan 1993), which specify the result of commuting an IFIR filter with a rate compressor (or expander). In particular, the result of IFIR filtering (whose interpolation factor is L) followed by L-to-1 rate compression is equivalent to L-to-1 rate compression followed by *FIR* filtering using the original, uninterpolated FIR filter from which the IFIR filter was derived. Note that in IFIR decimation the anti-imaging filter (which does not participate in the Noble Identities) must be applied *before* the IFIR filter; but since linear filters commute, the order of application is irrelevant and does not change the output. This does imply, however, that the polyphase structure for the L-to-1 rate compression gets applied to the anti-imaging filter; this is highly beneficial since it is this filter which is fed the original, high sample rate input. Finally, note that if a total decimation ratio of $M > L$ is required, where M/L is an integer, then a second polyphase structure will apply to the final (now uninterpolated) FIR filter as well, since only a factor L of rate compression can be commuted through the IFIR filter and integrated into the initial anti-imaging filter.

At a conceptual and operational level, an IFIR decimator is very similar to a multirate, multistage decimator in which all decimation stages except the last have been combined into a single stage—namely, the polyphase structure for the anti-imaging filter. Using the Noble Identities, it is easy to see that the final (now uninterpolated) FIR filter in an IFIR decimator will generally be identical to the last stage of the equivalent multirate, multistage decimator. In practice, then, the choice of which to use involves comparing the trade-offs in using a single, moderately complex anti-imaging decimation stage (in the IFIR decimator) instead of several small, simple decimation stages (in the multirate decimator). For large decimation ratios, however (or more precisely, when that portion of the rate compression incorporated into the anti-imaging polyphase structure is large), the same reasoning used initially to justify a multistage approach can be used to conclude that a single anti-imaging stage will be the more expensive choice. On the other hand, a programmable logic implementation with numerous stages can produce an explosion in the bit widths between successive stages, unless repeated truncation or rounding of intermediate signals is performed. For this reason it is difficult to dismiss the IFIR approach a priori. One can even use a hybrid approach in which one or two extremely simple stages are used to significantly "pre-decimate" the input signal, whose decimation is completed using an IFIR approach at the point where the individual stages of a multistage approach begin to increase in complexity. In terms of CARMA spectral-line modes, though—and their relatively modest requirements for stopband attenuation in particular—sufficient use of "pre-decimation" can be made that employing an IFIR stage for the remainder offers no advantage. All decimators described in the following section are full multistage designs.

2.3. Frequency Modulation

Decimation can be implemented using low-pass, high-pass, or band-pass filters, depending on the location of the desired passband within the Nyquist interval. A multistage design, however, is generally only practical if low-pass filters can be used—at least for all stages before the last. This is due to the fact that the passband at each stage must reside at a location where it will not alias onto itself during decimation. If there are three 2-to-1 decimation stages, for example, then the original passband must not overlap any of the frequencies {1/8, 1/4, 3/8, 1/2, 5/8, 3/4, 7/8} (in units of the Nyquist frequency); and so on. For large decimation ratios, therefore, the only sensible choice is to place the passband adjacent to either zero frequency or the Nyquist frequency (in which case it will alias to zero frequency after the first decimation stage). As an added bonus, low-pass 2-to-1 decimation filters (i.e., half-band filters) are twice as efficient as equivalent band-pass filters since (by symmetry) half of their coefficients are zero.

To allow decimators employing only low-pass filters to be used with passbands centered somewhere inside the original Nyquist interval, digital frequency modulation must be used. In this technique, the original (real) input signal is modulated by a complex exponential to produce a complex signal whose passband can be centered at any desired frequency. Assuming that the forward Fourier transform (from time t to frequency f) is defined using a positive exponent:

$$H(f) = \int_{-\infty}^{\infty} h(t) e^{2\pi i f t} \, dt,$$

zero frequency can be shifted to frequency f_0 by multiplying the input samples $\{x_k\}$ by $e^{-i\pi f_0 k}$, where f_0 is in units of the Nyquist frequency. The result is particularly simple for $f_0 = \pm 1/2$, in which case the frequency modulated samples are $\{x_0, \mp i x_1, -x_2, \pm i x_3, x_4, ...\}$; in particular, no trigonometric calculations are required, and the bit width of quantized samples is not increased by the modulation process—both important considerations for FPGA implementations. (There is one exception to this for n-bit signed integer samples: negating the minimum representable value, -2^n overflows the maximum representable value, $2^n - 1$, which increases the bit width by one.) However, as the input samples are now complex instead of real, the amount of data to be filtered is doubled. But there is another significant advantage to frequency modulation: it allows the passband to be *centered* on zero frequency, effectively halving the sampling frequency and absolute transition width of the final, expensive half-band filter compared to a passband with one edge aligned with DC. This fully compensates for the need to filter two sample streams (real and imaginary parts) instead of one. For CARMA, an analog filter centered at 750 MHz will alias to 250 MHz (half Nyquist) and support the use of frequency modulation. As explained in the next section, a design based on digital frequency modulation is the most practical solution for the CARMA first-light system.

3. Decimators for CARMA Spectral Modes

3.1. Design Considerations

All of the CARMA spectral line modes are based on analog filtering (which takes place inside the spectral downconverters) followed by decimation (in digitizer FPGAs) of the quantized input samples from the original 1 GHz sampling frequency down to the Nyquist-limited sampling rate of the corresponding spectral mode. How the decimation is performed varies according to the bandwidth of the mode. As a limiting case, the 500 MHz "wideband" mode can (and should) be regarded as the spectral mode with the widest bandwidth, for which no decimation is needed. For the 250 MHz and 125 MHz modes, decimation is limited to simple rate compression; that is, L-to-1 decimation (where $L = 2$ for 250 MHz and $L = 4$ for 125 MHz) is implemented by keeping only every L-th input sample and discarding the rest. Thus for these modes, passband fidelity is entirely determined by the analog filter's response. In principle this approach can be extended to the narrower spectral modes as well; however, there are several practical difficulties. One is cost: each additional mode implies an extra filter in each spectral downconverter (120 being required for the 15-station, 8-band first-light correlator). More seriously, the downconverter design must accommodate the extra filters, increasing both board size and overall complexity. Adding or modifying bands after manufacture also becomes difficult or impossible.

Implementing spectral modes using digital filtering techniques avoids all of these concerns, but naturally has limitations of its own. The most important of these is the data throughput required of the digital filter—the wider the spectral bandwidth, the more demanding the filter will be in terms of clock frequency and/or logic requirements. To determine the digital filtering capabilities of the Altera FLEX 10K100E FPGAs used in the COBRA digitizers, Altera's FIR Filter Compiler was used to generate a number of digital filter components. Tolerances suitable for implementing CARMA spectral modes were chosen: maximum peak-to-peak passband ripple of 1 dB, minimum stopband rejection of 25 dB, and transition width from passband to stopband of 1/64 the passband width. (All first-light spectral modes contain ≤ 64 frequency channels, hence filter transition widths of at most 1/64 the passband are desirable to limit out-of-band aliasing to the outermost channels only. Note however that subsequent Hann windowing will draw some of this noise into adjacent bins.) The prototype filters were then individually synthesized (placed and routed) to determine their maximum operating frequency (fmax) and total logic usage. Reported fmax values were in the range of 100-130 MHz, and the filters consumed $\approx 50\%$ of the logic elements in a FLEX 10K100E.

In multistage decimation the final stage is normally the most demanding—it is the one responsible for creating the final, sharp passband edges. The simulated filters described above are the ones required for this stage. The frequency at which the final filter must operate depends not only on the final bandwidth, but also on how the filter is implemented in digital logic (parallel or serial) and the number of input streams that must be processed (one if the input is real, two if it is complex). A serial implementation processes one input bit per clock cycle, and hence requires n cycles to produce a single output when fed n-bit input samples; the consecutive application of 1-bit processing minimizes the filter's logic usage, at the expense of overall throughput. A fully parallel implementation, by contrast, is able to produce one filtered output per clock cycle, at the expense of increased logic usage. The 50% logic usage quoted above is for a fully parallel, two channel (4-bit input each), half-band filter of order 127 (65 non-zero coefficients, quantized to 9 bits), suitable for use in a decimator based on frequency modulation. In this design, the filter operates at the output frequency (twice the spectral bandwidth). Equivalent designs based on band-pass filtering of an unmodulated signal have similar logic requirements (the equivalent decimator uses a fully serial, single channel, 4-bit input band-pass filter with 256 coefficients). In the band-pass design the filter operates at quadruple the output frequency (8 times the final bandwidth), and hence is inferior to the frequency modulation approach in that regard.

Another consideration for digitally-created spectral modes is efficiency loss, defined in terms of the expected cross-correlation signal-to-noise ratio (SNR). Since the digitally filtered output needs to be requantized into 2-bit samples for transmission to the correlator cards, the final cross-correlation SNR is necessarily less than the nominal value of 0.872 that results from the initial 2-bit (1 GHz) digitizer sampling followed by a deleted-inner-products correlation scheme (e.g., Hagen & Farley 1973), assuming weakly correlated input ($\rho \ll 1$). Since the analog signal is always digitized at 1 GHz, however, the input to the narrowband modes is highly oversampled, and hence most of the quantization noise produced by the initial 2-bit sampling can be removed during decimation. In this way the final SNR can remain relatively close to 0.87 instead of $0.87^2 = 0.76$ as expected otherwise, although some additional loss in SNR is unavoidable. The decrement depends on the decimation ratio and the details of the filters used in the decimator (both coefficients and bits of precision); for the proposed CARMA design (see § 3.2) it is a few percent.

Applying these considerations to CARMA leads to the following conclusions. First, it is easier to minimize the loss in cross-correlation SNR if the narrowest available analog filter is used to "pre-filter" the input signal; in this case most of the original Nyquist interval will contain only small amounts of quantization noise, easing requirements on the early decimation stages. Second, the relatively broad transition width of this analog filter—estimated at 1/16 its passband width—implies that it is not feasible to use a decimator based solely on low-pass filters unless frequency modulation is also employed. Simply put, the region near digital DC is too corrupted by aliasing from the wings of the analog filter to be included in the passband of any high resolution spectral modes. Hence the practical choices for CARMA are (i) center the narrowest analog filter at 750 MHz (the middle of the downconverter window) and use digital frequency modulation to recenter the passband at zero frequency for further processing, or (ii) align one edge of said analog filter at either 500 MHz or 1 GHz, and use a band-pass filter in the final decimation stage to avoid zero frequency when carving out the final passband. The 2 MHz and 8 MHz modes can be implemented using either option. The 31 MHz mode can be implemented only with option (i), since in option (ii) the band-pass filter would need to operate at 256 MHz—well beyond the capabilities of a FLEX 10KE device; using a parallel filter would avoid this issue, but would not fit in the 10K100E FPGAs used by the digitizers. Hence frequency modulation is the better alternative.

In neither option can spectral modes with bandwidths larger than 31 MHz use digital filtering to create sharp passband edges. Four analog filters are therefore required: 500 MHz, 250 MHz, 125 MHz, and 62 MHz. The three widest analog filters produce their final passbands directly, and involve no digital processing other than simple rate compression; their nominal edges can be aligned to either 500 MHz or 1 GHz in the downconverter window. To meaningfully support the frequency modulation option, the 62 MHz filter *must* be centered at 750 MHz. This demands that frequency modulation also be applied in the creation of the 62 MHz observing mode, since otherwise the passband would alias onto itself when subsampled. This in turn means that the signal must be low-pass filtered, and requantized to 2-bits (with some loss in SNR). Since only simple digital filters can be used here (due to the large throughput required), the passband edges for the 62 MHz mode will be no sharper than that of the analog filter. The 31 MHz, 8 MHz, and 2 MHz modes are each formed from the center of the 62 MHz analog passband after modulation to zero frequency and application of a multistage decimator, as described previously. Detailed filter specifications are given in the follow section.

3.2. Proposed CARMA Design

Based on the preceding considerations, I recommend that the 62 MHz, 31 MHz, 8 MHz, and 2 MHz spectral-line modes be implemented using the following processing pipeline:

1. Encode Re-encode the 2-bit sign/magnitude input samples as two's-complement signed integers for further processing. Note that the proper numerical values (weights) for a 2-bit quantization scheme are {-3, -1, +1, +3}, which nominally requires three bits to represent. This can be reduced back to two bits using the linear transformation $x_k \to (x_k - 1)/2$ on the samples $\{x_k\}$, which maps the original values to {-2, -1, 0, 1}. This transformation amounts to adding a large DC component to the input spectrum, which does not affect the passband (still centered on 250 MHz at this stage) and is completely removed by subsequent processing.

2. Modulate Center the positive frequency analog passband on zero frequency by multiplying the encoded samples $\{x_k\}$ by $e^{\pi ik/2}$, producing a complex sequence $\{z_k\}$=$\{x_0, ix_1, -x_2, -ix_3, \ldots\}$. The negative frequency passband becomes centered on the Nyquist frequency. Note that half of the samples (real or imaginary parts thereof) are zero after modulation.

3. Decimate Apply one or more FIR filter/decimation stages to reduce the input sampling rate to the Nyquist-limited rate of the corresponding spectral mode. Five distinct filters are used in all; their coefficients and the increase in bit width they produce (output vs. input) are listed in Table 1. The complete filtering sequence for each band is:

$$
\begin{array}{rllllll}
\text{62 MHz:} & F1(8); & & & & & \\
\text{31 MHz:} & F1(8), & F4, & HB; & & & \\
\text{8 MHz:} & F1(16), & F4, & F4, & HB; & & \\
\text{2 MHz:} & F1(16), & F2, & F2, & F4, & F4, & HB.
\end{array}
$$

The filters $F1(8)$ and $F1(16)$ are, respectively, 8-to-1 and 16-to-1 comb-like decimators whose coefficients have been optimized to strongly suppress the negative frequency passband (> 50 dB rejection); they also exactly null the spectral component (originally at DC) added during encoding. The $F2$ and $F4$ filters are simple 2-to-1 half-band decimators (cf. Crochiere and Rabiner, § 5.5.2). Filter HB is the primary half-band filter, responsible for creating the final, sharp edge for the 31 MHz, 8 MHz, and 2 MHz modes. Its coefficients are based on the windowing method, scaled and truncated to eight bit integers (the maximum, central coefficient requires nine bits); the window function was non-standard and hand-tweaked to place the response ripples at favorable locations relative to the estimated channel boundaries. Note that there is no decimation associated with filter HB.

4. Demodulate Center the passband at half the Nyquist frequency by multiplying the decimated samples $\{z_k\}$ by $e^{-\pi ik/2}$. The passband now fills the entire positive frequency range, from zero to Nyquist. Negative frequencies contain no signal after this stage, only residual noise left over from the previous step. Note that the real part of the demodulated sample sequence consists of even-k real parts interleaved with odd-k imaginary parts from the original sequence $\{z_k\}$ (with appropriate sign inversions).

5. Reconstruct Recover the output sequence $\{x_k\}$ from the demodulated sequence $\{z_k\}$ by taking the real part, $x_k = (z_k + \bar{z}_k)/2$. In the frequency domain this is equivalent to adding the spectrum to its reflection about zero frequency—hence the residual noise mentioned in the previous step is unavoidably added to the desired output signal during reconstruction.

6. Requantize Filtering increases the bit widths of the samples (cf. Table 1). This final processing stage requantizes the reconstructed samples back to two bits.

Note that more than one stage may be integrated into a single component in the VHDL implementation. In particular, creation of the 31 MHz mode requires that *only* those samples actually needed during reconstruction be calculated by filter HB (otherwise, either operating frequency or logic usage must be doubled, neither of which is feasible for the 31 MHz mode). This amounts to computing only the *even-numbered* filter outputs for the real part of the sample stream, and only the *odd-numbered* filter outputs for the imaginary part of the stream. This can be accomplished using the polyphase representation of HB, in which the even- and odd-indexed coefficients are separated into sub-filters P_0, and P_1, respectively. The (purely) even-numbered outputs $\{y_0, y_2, y_4, ...\}$ of an input sequence $\{x_0, x_1, x_2, ...\}$ can then be computed by filtering the sub-sequence $\{x_0, x_2, x_4, ...\}$ through P_0, the sub-sequence $\{0, x_1, x_3, ...\}$ through P_1, and then adding the two sub-filter outputs together. Similarly, the (purely) odd-numbered outputs are computed by filtering the sub-sequences $\{x_1, x_3, x_5, ...\}$ and $\{x_0, x_2, x_4, ...\}$ with P_0 and P_1, respectively, and then adding the results. Hence in this case the P_0 filter will process two input sub-streams and (assuming a fully parallel implementation) operate at the final sampling frequency—62 MHz for the 31 MHz band, which easily satisfies timing limits. For a half-band filter (such as HB), the sub-filter P_1 is trivial as it contains only one non-zero coefficient.

Net frequency response for the 62 MHz and 8 MHz modes are shown in Figures 1 and 2, respectively; response for the 31 MHz and 2 MHz modes is nearly identical to that for the 8 MHz band, all three being dominated by the response of filter HB. The 62 MHz mode exhibits 0.6 dB of fall-off at the band edges due to digital processing (by comparison, the analog filter response is estimated to be -10 dB at this point). The channel-averaged passband response for the remaining modes is quite flat, with < 0.3 dB peak-to-peak ripple due to digital processing. In addition, out-of-band rejection is > 30 dB, except in the outermost channels. The large outer-channel aliasing is an inherent limitation (‘feature’) of half-band filters, whose response at half Nyquist is fixed by symmetry at -6 dB, independent of the number of coefficients used. The zero and

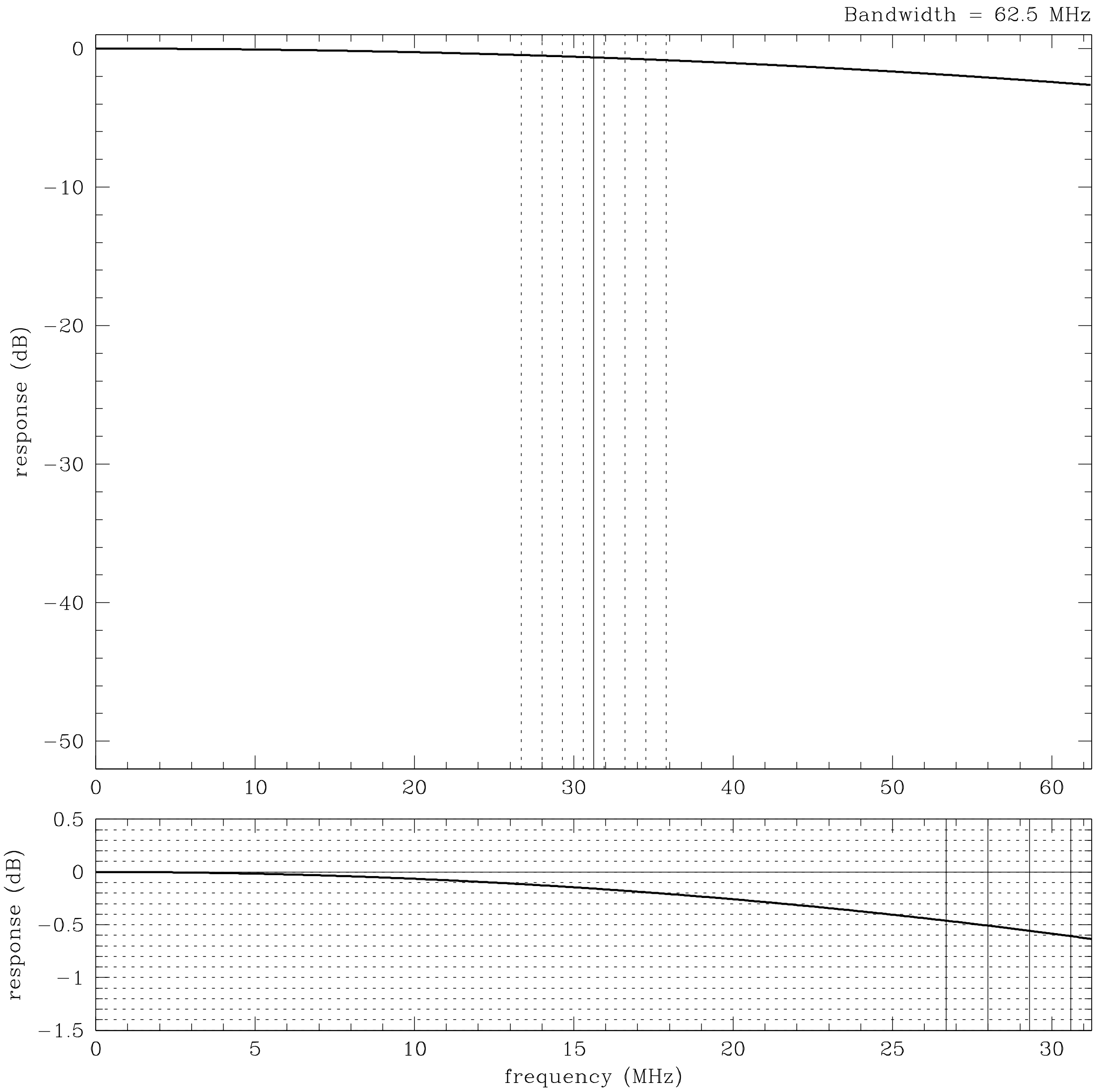

Fig. 1.— Net frequency response and stopband rejection of the proposed 62 MHz observing mode. The response shown is for the digital filtering component only; the analog filter produces an additional roll-off in response across the band, and is also solely responsible for creating the passband edges for this mode. Note that the passband is centered on zero frequency and that filter response is symmetric about this point. Estimated channel boundaries for the outermost channels are indicated by vertical lines.

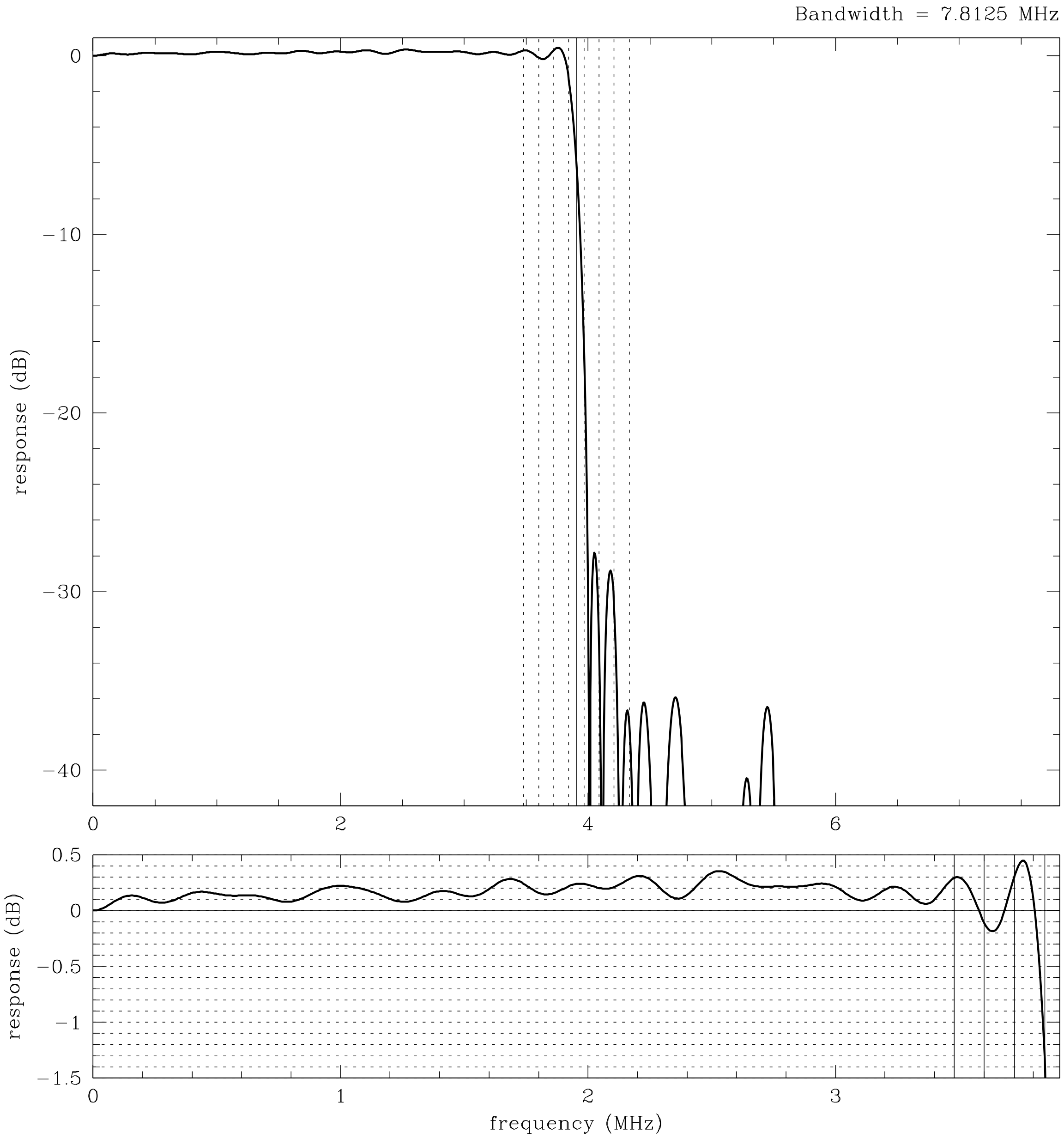

Fig. 2.— Net frequency response and stopband rejection of the proposed 8 MHz observing mode. The response shown is for the digital filtering component only; the analog filter produces an additional, small roll-off in response across the band ($\sim$ 0.5 dB). Note that the passband is centered on zero frequency and that filter response is symmetric about this point. Estimated channel boundaries for the outermost channels are indicated by vertical lines. Responses for the 31 MHz and 2 MHz modes are similar (all are dominated by the final half-band filter).

Nyquist frequency channels are therefore usable only for determining continuum baselines, assuming there are no lines in the adjacent out-of-band channel.

Note that filtering severely alters the sample statistics; the samples being requantized do *not* follow a Gaussian distribution, and hence the optimum "threshold voltage" v_0 (the positive input level separating the high and low quantization states, in units of the standard deviation of the input) need not coincide with that appropriate for the first quantization; it may also change between spectral modes. Cross-correlation statistics, which determine the final SNR, are also significantly affected. Detailed, bit-accurate simulations of the above processing pipeline were performed to accurately determine the cross-correlation statistics and SNR variations (as a function of v_0) for each of the spectral modes. Efficiency also depends on the number of bits retained between decimation stages. To gauge SNR dependence on the latter, simulations maintaining 6, 7, or 8 bits of precision between stages were performed (input to filter HB was limited to 4, 5, or 6 bits, respectively, as it will dominate the FPGA logic usage). For comparison, full precision simulations—quite impractical in terms of logic usage!—were also done.

Sample input was created by drawing pairs of samples from a bivariate Gaussian (correlation factor $\rho = 0.1$), band-limiting the two input streams to model the analog pre-filter, and then quantizing them to two bits. The initial quantization used a threshold voltage $v_0 = 1.00$, the optimum value for full 2-bit cross-correlation (spot testing with $v_0 = 0.90$ showed little change in the final SNR). The samples were then processed as described above. After each stage, power spectra, population histograms, and cross-correlation statistics were calculated, for a series of values of v_0 between 0 and 2. The SNR results are summarized in Table 2. The final SNR is seen to depend sensitively on the number of bits retained between filtering stages, especially at the highest decimation ratios, which contain the most filtering stages. Although the actual SNR for each mode cannot be determined until the full VHDL implementation is complete, a 6-bit implementation should be regarded as the worst-case. Thus efficiency losses are predicted to be 3-6%, or less.

4. Discussion

The preceding analysis assumes as-is reuse of COBRA hardware. In that case, decimators based on digital frequency modulation appear to be the most advantageous, although they are somewhat more complicated to implement than decimators employing band-pass filters. The latter approach could be viable for the 31 MHz mode if two of the digitizer FPGAs were replaced with larger (10K200E) devices, but as the frequency modulation option can also support modes narrower than 2 MHz with relative ease, the added algorithmic complexity provides flexibility as well as cost-effectiveness. In fact, the use of 10K200E devices would allow the 62 MHz band to be created digitally as well, as long as frequency modulation is employed.

The filters listed in Table 1 are not cast in stone, and changes would add negligibly to implementation costs; improved performance may be possible after further experimentation. In particular, a more intelligent (yet practical) approach to maximizing SNR would be to minimize truncation during the early, simple stages, and to round results only as needed in the final one or two stages. Such issues are best investigated after the initial VHDL implementation is available, when total logic usage can be accurately determined.

Table 1. FIR Filter Coefficients

Filter	Expansion[a]	Coefficients
$F1(8)$	4	{1, 2, 2, 3, 3, 2, 2, 1}
$F1(16)$	5	{1, 2, 2, 2, 2, 3, 4, 4, 3, 2, 2, 2, 2, 1}
$F2$	2	{1, 2, 1}
$F4$	6	{-3, 0, 19, 32, 19, 0, -3}
HB	9	{ -1, 0, 1, 0, -1, 0, 1, 0, -2, 0, 2, 0, -2, 0, 2, 0, -2, 0, 2, 0, -3, 0, 3, 0, -3, 0, 3, 0, -3, 0, 4, 0, -4, 0, 4, 0, -5, 0, 5, 0, -6, 0, 7, 0, -8, 0, 9, 0, -11, 0, 13, 0, -17, 0, 24, 0, -40, 0, 122, 192, 122, 0, -40, 0, 24, ... }

[a]The maximum number of additional bits needed to represent the filtered samples, relative to the input bit width.

Table 2. Estimated CARMA FIR Spectral Mode Efficiency

	6-bit[a]		7-bit[a]		8-bit[a]		full[a]	
Bandwidth	α[b]	SNR[c]	α[b]	SNR[c]	α[b]	SNR[c]	α[b]	SNR[c]
62 MHz	···	···	···	···	···	···	3.022	0.876
31 MHz	3.596	0.843	3.499	0.846	3.450	0.849	3.409	0.849
8 MHz	3.236	0.815	3.036	0.830	3.153	0.834	3.263	0.837
2 MHz	3.181	0.807	3.010	0.830	3.122	0.834	3.238	0.835

[a] The number of bits maintained between filtering stages (for the input to *HB* only, subtract two bits); 'full' means no truncation.

[b] The cross-correlation normalization factor, $\alpha(\rho;v_0) = \langle XY\rangle/\rho$, where ρ is the true correlation between X and Y, and v_0 is the threshold voltage. Values are for $\rho = 0.1$ and $v_0 = 0.90$.

[c] Maximum cross-correlation signal-to-noise ratio relative to a single, infinite-precision quantization. A deleted-inner-product scheme is assumed. For comparison, the maximum SNR for a single 2-bit quantization is 0.878 ($\rho = 0.1$). Maximum SNR was achieved at $v_0 = 1.00$ for the 62 MHz band and at $v_0 = 0.90$ for all remaining bands.

REFERENCES

Beasley, A.J., Hawkins, D.W., Rauch, K.P., and Woody, D.P., 2003, CARMA Memo 11.

Beasley, A.J., Woody, D.P., and Hawkins, D.W., 2002, CARMA Memo 3.

Crochiere, R., and Rabiner, L. 1983, *Multirate Digital Signal Processing* (Englewood Cliffs: Prentice-Hall), Chapter 5.

Dick, C. 1998, "Implementing Area Optimized Narrow-Band FIR Filters Using Xilinx FPGAs," white paper available from www.xilinx.com.

Hagen, J.B., and Farley, D.T. 1973, "Digital-correlation techniques in radio science," *Radio Science*, 8, 775.

Rauch, K.P., 2003, CARMA Memo 9.

Vaidyanathan, P.P. 1993, *Multirate Systems and Filter Banks* (Englewood Cliffs: Prentice-Hall).