ALMA memo 447

Multi-resolution FX Correlator

John Bunton
CSIRO Telecommunications and Industrial Physics
27 February 2003

Introduction

The use of polyphase filterbanks allows arbitrary filter responses to be implemented in an FX correlator. The quality of the filters implemented and the ability to oversample the output allows the filterbanks to be cascaded. Cascading the filterbanks provides multiple frequency resolutions and the resulting correlator can simultaneously generate continuum and multi-resolution spectral line data. Where only part of the full bandwidth is needed at high resolution the cost of the cascaded filterbanks is similar to that of the traditional FFT based FX correlator with the same frequency resolution. In addition, the cascaded polyphase filterbank eliminates problems associated with traditional FFT based FX correlators.

Polyphase filterbanks

The FFT is a very efficient method of transforming data between the time and frequency domains. When used to form a filterbank it constrains the possible frequency response of each channel because the impulse response of the channel must be no greater than the length of the FFT. With data processed in non-overlapping blocks this constraint on filter length leads to significant aliasing as the separation between frequency channels is equal to the width of the frequency channels. For example, the -3dB width of a Hamming window is 1.3 times the frequency channel separation. One way to solve this is to increase the output sampling rate by overlapping the section of data being processed. This can be very inefficient. For the Hamming window the output data rate increases by a factor of 3.7 if aliasing is to be kept below -40dB. In an FX correlator this would almost quadruple the cross-multiply operations and the data transfer load. As there is considerable overlap between bins in the frequency domain the load is reduced by pruning the outputs of the FFT. This will be shown to lead to a polyphase filterbank implementation.

Consider the discrete Fourier transform (DFT) *X(k)* of a data consisting of a sequence *x(n)* multiplied by the window function (filter impulse response) *h(n)*.

$$X(k) = \sum_{n=0}^{N-1} h(n).x(n).e^{-j(2\pi/N)nk}$$

For an FFT implementation k takes the values 0 to N-1. To prune the output data only a subset of the *X(k)* values need be calculated. If N can be factored as rM and only every rth value of *X(k)* is taken then the calculation is reduced to

$$X(k') = \sum_{n=0}^{N-1} h(n).x(n).e^{-j(2r\pi/N)nk'}$$ where k′ takes the values 0 to M-1.

This can be rearranged to give

$$X(k') = \sum_{m=0}^{r-1} \sum_{n=0}^{M-1} h(n+mM).x(n+mM).e^{-j(2\pi r/N)(n+mM)k'}$$
$$= \sum_{m=0}^{r-1} \sum_{n=0}^{M-1} h(n+mM).x(n+mM).e^{-j(2\pi r/N)nk'} e^{-j2\pi.m.k'}$$
$$= \sum_{m=0}^{r-1} \sum_{n=0}^{M-1} h(n+mM).x(n+mM).e^{-j(2\pi/M)nk'}.1$$

It is seen that this is a summation of *r* DFTs each of length M and that these DFTs have an FFT implementation. In each of these FFTs a section of the data of length M is multiplied by a section of the window function and transformed. The workload in windowing the data remains unchanged. The FFT workload has been reduced from a single N point FFT to *r* FFTs of length M. This provides a small reduction in the computational load but now the order of the summation can be reversed to give

$$X(k') = \sum_{m=0}^{r-1} \sum_{n=0}^{M-1} h(n+mM).x(n+mM).e^{-j(2\pi/M)nk'}$$
$$= \sum_{n=0}^{M-1} \left[\sum_{m=0}^{r-1} h(n+mM).x(n+mM) \right] .e^{-j(2\pi/M)nk'}$$

Equation 1

Again, the total workload needed to implement the windowing is unchanged but the FFT work load is reduced to a single M-point transform. This is the polyphase filterbank [Bellanger and Daguet 1974] and [Crochiere and Rabiner 1983] and can be implemented by a structure such as the one shown in Figure 1. In this filter the polyphase filters $p_n(m)$ are equal to *h*(n+mM).

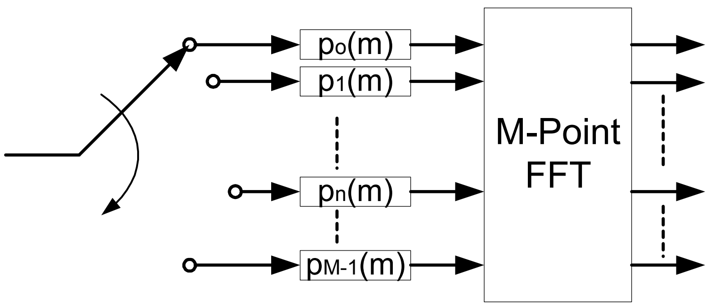

Figure 1 Polyphase Filterbank

This structure implements a filterbank with arbitrary response for a computational cost that is approximately double that of the FFT alone, for the case where the prototype lowpass filter h(n) is an order of magnitude longer than the FFT. For the above implementation the input and output data rates are identical and there is no difference in the cross multiply load between an FX correlator using this as a filterbank and a traditional FFT FX correlator.

Polyphase Filterbank Performance

For the implementation shown in Figure 1 each execution of the FFT must be on data that is displaced in time by M samples. For each frequency channel this sets the output data rate at 1/M of the input data rate. If the input data rate is f_i then the output data rate sets the two-sided bandwidth of each frequency channel to f_i/M. The spacing of outputs from the FFT is also f_i/M. Thus, the spacing between each frequency channel is identical to the width of the channel. A representative filter and its magnitude response are shown in Figure 2, together with the response for an adjacent channel. It is seen that the magnitude responses for two adjacent frequency channels overlap at a point halfway between the two centre frequencies. If the overlap was eliminated by making the bandwidth narrower then aliasing can be eliminated but at the cost of eliminating a band of frequency components at the half way point. In either case, the frequencies near the halfway point are corrupted.

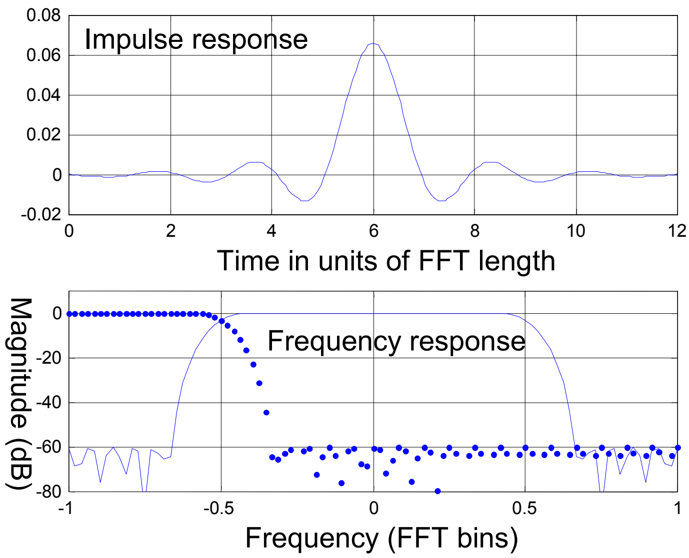

Figure 2 Example of a prototype filter impulse response and the resulting channel response for a polyphase filterbank. Also shown in the lower plot is the channel response of an adjacent channel (dotted) for a polyphase filterbank implemented with the topology shown in Figure 1.

The aliasing can be eliminated, without losing data, by raising the output sample rate from the polyphase filterbank. Inspection of equation 1 shows that it defines only one set of outputs from the polyphase filterbank and that to generate the next output the input sequence could be shifted by any number of samples. As is seen in Figure 1 the architecture is simple if the shift is made equal to the FFT length but the shift could equally well be βM ($\beta<1$). To increase the output sample rate consider a polyphase filterbank where eight simultaneous filters ($p_n(m)$) are needed to maintain the data flow into the FFT. In a single clock period the eight filters will normally process eight new input samples. These must be stored to generate later outputs, thus each filter stores every eighth input sample. By decoupling the filter operation from the data input the start of block can be made to occur on any sample that is a multiple of eight from the start of the previous block. For a 1024-point FFT this provides better than 1% accuracy for setting the oversampling ratio.

The increase in the output data rate effectively oversamples each output channel. A filterbank where the output data rate is increased in this way will be referred to as an oversampled filterbank. Use of oversampling decouples the width of the frequency bins from the spacing between frequency bins. This gives the design of oversampled filterbanks the same flexibility as is the case for digitally sampled analogue filterbanks. Thus filterbanks and cascades of filterbanks can be designed with the same level of performance as analogue filterbanks but without the stability problems and variability of analogue filters.

Cascaded Polyphase Filterbanks

For the response shown in Figure 2, setting β equal to 0.88 eliminates aliasing within the frequency range –0.5 to 0.5 f_i/M from the band centre, where f_i is the input data rate and M is the length of the FFT. This corresponds to oversampling each output channel by a factor of 1.13. With this change, the outputs from the polyphase filterbank can be processed by a second polyphase filterbank without loss or corruption of any of the original data.

However, there is some overlap in the data generated when two adjacent frequency channels from the first filterbank are processed. After a single channel is processed by the second filterbank the new output channels will span the frequency range –0.56 to $0.56f_i/M$ from the band centre of the input channel. The adjacent channel centred on a frequency f_i/M higher spans the range 0.44 to $1.56f_i/M$ after processing. There are redundant frequency channels in the range 0.44 to $0.56f_i/M$. But only those up to $0.5f_i/M$ from the first channel and above $0.5f_i/M$ for the adjacent channels are free of aliasing. The other output channels can be discarded and the frequency range is still continuously covered without any loss of data. Discarding the redundant channels prevents growth in the total data rate. If the redundant channels were not discarded a cascade of filterbanks with oversampling ratio (1+Δ) would cause the data rate to grow at (1+Δ) for each stage of filterbanks.

Table 1 Cascaded oversampling filterbanks discarding redundant data. Last stage not oversampled.

	Input data rate to filterbank (Ssample/s)	Output channel data rate (Ssample/s)	No of filter banks	Total output channels per filterbank	Total output data rate
1st stage	f_i	$(1+\Delta)\ f_i/M$	1	M	$(1+\Delta)f_i$
2nd stage	$(1+\Delta)f_i/M$	$(1+\Delta)^2 f_i/M^2$	M	$M/(1+\Delta)$	$(1+\Delta)f_i$
3rd stage	$(1+\Delta)^2 f_i/M^2$	$(1+\Delta)^3 f_i/M^3$	$M^2/(1+\Delta)$	$M/(1+\Delta)$	$(1+\Delta)f_i$
4th stage	$(1+\Delta)^3 f_i/M^2$	$(1+\Delta)^3 f_i/M^3$	$M^3/(1+\Delta)^2$	$M/(1+\Delta)$	f_i

An example is shown in Table 1 of three stages of oversampling filterbanks (oversampling ratio 1+Δ) followed by one stage of non-oversampling filterbanks. At each stage the input data rate into a single filterbank is equal to the output data rate for a single channel from the previous stage of filterbanks. The number of filterbanks grows initially by M, the number of channels in the first filterbank. At later stages the growth is reduced to $M/(1+\Delta)$ due to channels being discarded from the previous set of filterbanks. At each stage the total non-redundant output data rate is equal to the sample rate for a single output channel times the number of useful channels and the number of filterbanks in the stage. It is seen that the total data rate grows at the output of the first filterbank but then remains constant. In the last stage there is no oversampling and the total output data rate reduces, to again be equal to the input data rate.

The compute load of a filterbank is proportional to the total output data rate, including unwanted data. For the first filterbank the output data rate is $(1+\Delta)f_i$. For the second filterbank there are M^2 output channels of which $M^2/(1+\Delta)$ are actually used. Thus even though there is no increase in the total useable data the total compute load of the second set of filterbanks is $(1+\Delta)$ times greater than that of the first filterbank. From then on there is no further increase in the compute load for later stages of processing.

The errors that are introduced by cascading the filterbanks in this way are due to the channel response of preceding filterbanks. This introduces gain changes between channels that are easily calibrated, and a small magnitude slope. For the filter response shown in Figure 2 the worst case slope is 0.1dB for a 2k-channel filterbank. This error can be reduced by use of an optimised channel response.

Efficiency of Cascaded Filterbanks

Cascaded filterbanks provide a method of generating high-resolution frequency channels. If high resolution is required across the full frequency band it would be computationally more efficient to use a single polyphase filterbank with M^2 channels than a cascade of two M-channel filterbanks. The computation can be broken into two components: the polyphase filter and the FFT. If no oversampling occurs then, for a given resolution, the FFT part of the computation is the same for a single long filterbank or two cascaded filterbanks. For the polyphase component the computations needed for the first filterbank in a cascade is the same as that for a single filter bank implementation. Thus the extra computational load in implementing cascaded filterbanks is due to the polyphase calculation in the second and subsequent sets of filterbanks. Where oversampling is used the computational load increases by the oversampling ratio for the first filterbank. As an example it is estimated that a two stage cascade with a channel response as illustrated in Figure 2 and with only the first stage oversampling requires ~50% more computing power than the a single stage filterbank with the same resolution.

Implementation Advantage of Cascaded Filterbanks

However cascaded filterbanks may be easier to implement. Consider a single stage million channel filterbank. The polyphase section of the filterbank, with 8-bit sampling, must store ~20Mbytes of input data. This storage exceeds on-chip capacities and needs to be located in off-chip RAM. For a 2GHz bandwidth filterbank, such as the ATNF design [Ferris et al. 2001] with a 12-point polyphase filter, the I/O to the RAM needs a bandwidth of over 48Gbytes/s, 8-bit samples. This data rate is independent of the number of channels in the filterbank. Achieving a 48Gbyte/s data rate from an external memory is not easy. With a cascaded design the first stage could be a 1024-channel filterbank. The memory requirements are reduced to about 20kbytes, which can be accommodated within a single ASIC or FPGA.

In a cascaded design it is the next stage of filterbanks that a large memory is needed, ~20Mbytes. But now the processing and memory can be distributed over a number of smaller units. Say 10 units are used. Each unit will process 100 channels from the first stage filterbank and the storage requirement of each is ~2Mbytes. The I/O bandwidth of a single unit has also been reduced to ~4.8Gbytes/s, which is achievable with current packages. A further reduction to less than 1Gbyte/s is possible by increasing the size of the memory and processing multiple successive iterations of the polyphase filter on each frequency channel.

Application to FX correlators

A cascaded polyphase filterbank can be used as the F part of a high resolution FX correlator. (F originally stood for FFT [Chikada et al. 1984] and later Fourier transform [Chikada et al. 1987] but can equally well stand for filterbank [Ferris et al 2001] or frequency transform [Bunton 2002].) The use of polyphase filterbanks avoids many of the problems inherent in the use of the FFT as the frequency transform, such as segmentations of data, cyclic rather than linear convolution [Chikada 1991] and resultant loss of signal to noise ratio at the high frequency resolutions [Chikada 1987, Okumura et al 2001]. Other problems such as errors in the Van Vleck correction [Chikada 1991] and growth in the number of interconnections [Escoffier 1997] can be solved by the use of higher precision A/D converters, high precision signal processing, automatic level control on the quantiser following the frequency transformation [Bunton 2000] and the use of channel rerouting [Urry 2000].

A well known advantage of the FX correlator, when compared to an XF correlator, is that the number of cross-multiply accumulate (XMAC) operations needed in the correlator is independent of the frequency resolution [e.g. Bunton 2002]. As the computational load grows logarithmically with the number of channels in both FFTs and cascaded polyphase filterbank there is little difference in the cost of low and high resolution FX correlators. In either case, the problem caused by high resolution across the full frequency band is an unmanageable amount of correlation data ~64Gbytes/s for a single ALMA 2GHz full polarisation band per integration (4 Stokes parameter, 2000 baselines, 1,000,000 channels, 8 bytes per correlation). A solution to this problem [Okumura et al. 2001] for FFT based correlators is to average adjacent frequency channels after correlation. This also recovers some signal/noise performance for this lower resolution data.

For a correlator based on cascaded polyphase filterbanks frequency averaging can also be used. But a better option is to process only a subset of the outputs from the first stage filterbank in the second stage filterbank. The low resolution channels from the first stage filterbank are processed as they stand and those channels containing high-resolution features are processed in the second stage filterbanks. If a fraction α of the channels are processed then the cost of the second stage filterbanks is also reduced by the same factor α. As the cost of a full set of second stage filterbanks is approximately the same as the cost of the first stage filterbank the total cost of the filterbanks is $(1 + \alpha)$ times the cost of the first stage filterbank. This brings the cost of a cascaded filterbank down to the point where it is competitive with a straight FFT, assuming the implementations are in similar technology. Consider a million point FFT made from two 1,000-point FFTs connected by a corner turner. For equivalent frequency resolution the first stage of a cascaded filterbank might implement a 1,000-channel filterbank with a cost roughly equal to that of the one million point FFT (polyphase section of the polyphase filterbank has complexity approximately equal to that of the 1,000-point FFT). For α small, say 10%, the cost of the filterbanks is at most 20% greater than a one million point FFT, assuming the filter and FFT sections are the same cost. When the cost of the cross-multiply units and backend are added there is little difference in the two systems. However, at the highest resolution the cascaded filterbank approach can achieve a 20% higher sensitivity, which more than compensates.

The cost of the cascaded filterbank approach can be reduced even further if a small loss of continuum sensitivity is allowed when simultaneous spectral line observations are made. When operating in full continuum mode the first filterbank need not operate in an oversampling mode. Spectral line processing can still occur but there is degraded performance in the regions of overlap between the frequency channels from the first stage filterbank. To allow high quality spectral line data without gaps oversampling is needed. If no increase in the total compute capacity is possible then the input data rate must be lowered. The lowered data rate could be achieved by both changing the analogue filter and lowering the A/D sample rate or digitally filtering and decimating. The total processing power of the first stage filterbank remains constant for the two modes. With an oversampling ratio of $(1+\Delta)$ the loss of continuum sensitivity with simultaneous spectral line observation is $\sqrt{}/(1+\Delta)$ or about 6% for 13% oversampling.

Even with 10% processed at high-resolution a 1,000-channel second stage filterbank still generates 100,000 correlations. Implementing a correlator with this number of channels is not difficult if a channel reordering approach [Bunton 2001] is used. The increase in channel numbers is achieved by adding more memory, implemented in commodity RAM or DRAM. However, the data transfer rates and storage after the correlator can become excessive. This can be reduced by selectively storing the data from the second filterbanks or averaging the data after correlation. Note, full continuum sensitivity can still be achieved by processing all the data from the first stage filterbank.

As an example of the performance possible with an FX correlator using cascaded filter bank with:

Stage 1. 4000, 1000 or 250 channels either oversampling or direct with a 2GHz direct bandwidth.

Stage 2. Input 10% of first stage channels, 400, 100 or 25 input channels respectively arbitrarily chosen from the 4000, 1000 or 250 channels of stage 1. These channels are processed by non-oversampling filterbanks with either 1000, 250 or 64 frequency channels.

The resulting specifications are summarised in Table 2.

Table 2 Example of Correlator Specifications, performance limited by the filterbank

First stage mode	Continuum Bandwidth GHz	Continuum Resolution MHz	Continuum Channels Processed	Spectral line Bandwidth MHz	Spectral line resolutions kHz
Direct	2	0.5	400	200 [note]	8, 2, 0.5
Direct	2	2	100	200 [note]	32, 8, 2
Direct	2	8	25	200 [note]	128, 32, 8
Oversampling	1.8	0.45	400	180	8 ,2, 0.5
Oversampling	1.8	1.8	100	180	32, 8, 2
Oversampling	1.8	7.2	25	180	128, 32, 8

[note] degraded performance in overlap regions of first stage frequency channels.

In oversampling mode the spacing between continuum frequency channels decreases but the increase in output sampling rate maintains the total bandwidth of each output channel constant even thought the spacing between channels has decreased. Thus there is no change in spectral line resolution between direct and oversampling mode. Adding about 10% to the performance of the filterbank would allow it to operate in an oversampling mode at the full 2GHz bandwidth. This gives the specifications shown in Table 3.

Note, all resolutions are available at the same time. This is achieved by breaking the outputs from the first filterbank into subsets. Each subset is then processed by filterbanks of different lengths.

Table 3 Example of Correlator Specifications, oversampling mode at full bandwidth

First stage mode	Continuum Bandwidth GHz	Continuum Resolution MHz	Continuum Channels Processed	Spectral line Bandwidth MHz	Spectral line resolutions kHz
Oversampling	2	0.5	400	200	8.8, 2.2, 0.55
Oversampling	2	2	100	200	34, 8.8, 2.2
Oversampling	2	8	25	200	137, 34, 8.8

At the highest resolution there are 400,000 spectral line channels which would probably exceed the throughput of the correlator backend. This can be reduced by processing only a subset of the channels at the highest resolution and processing the rest at a lower resolution. For example, configuring the filterbanks as shown below reduces the total channel count to 115,000 and still maintains a maximum resolution of 550Hz.

Table 4 Possible configuration of filterbanks

	Continuum	Low resolution	Med resolution	High resolution
No of .5MHz channels	3600	160	120	120
Total Bandwidth	1.8GHz	100MHz	60MHz	40MHz
Resolution	0.5MHz	8.8khz	2.2kHz	0.55kHz
No. of Channels	3600	11,363	27,273	72,727

The throughput could be further reduced by limiting the number of spectral line channels processed by the cross-multiply accumulate modules. This will help to minimise the cost of the cross multiply accumulate module. Alternatively, total output data could be reduced by averaging across adjacent channels, which further increases the number of frequency resolutions available.

The use of cascaded filterbanks allows upgrades to the spectral line capability of the correlator to be added incrementally by adding extra second stage filterbanks. This does not alter the total data rate over the cabling between the filterbanks and the cross-multiply units. Within the cross-multiply units extra memory will be needed to accumulate the extra correlations and buffer the incoming antenna data.

Conclusion

The use of oversampled polyphase filterbanks allows filterbanks to be cascaded without the introduction of aliasing. This allows FX correlators to be built that can simultaneously operate at multiple frequency resolutions while at the same time eliminating many of the disadvantages inherent in FFT based FX correlators. The added cost of simultaneous continuum and multi-resolution spectral line observations is small if a limited amount of data is processed by the second stage of filterbanks. This reduction also limits data generated by the correlator making backend processing more manageable.

References

Bellanger, M.G., and Daguet, J.L., 'TDM-FDM Transmultipexer: Digital Polyphase and FFT", IEEE Trans. Com., Vol Com-22, No. 9, pp 1199-1205, Sept 1974.

Bunton, J.D., 'Implementing Correlators for the SKA', WARS'02 Workshop on Applications of Radio Science, Leura Australia, 20-22 February 2002

Bunton, J.D., 'An Improved FX Correlator', ALMA Memo 342, Dec 2000, http://www.alma.nrao.edu/memos/

Bunton, J.D. 'Low Cost Cross Multiply Accumulate Unit for FX Correlators', ALMA memo 392, Sep 2001, http://www.alma.nrao.edu/memos/

Chikada Y., et al., 'Digital FFT Spectro-Correlator for Radio Astronomy', in Indirect Imaging (ed J.A. Roberts), Cambridge University Press, pp387-404, 1984

Chikada Y., et al., 'A 6 X 320-MHz 1024-Channel FFT Cross-Spectrum Analyser for Radio Astronomy', Proc. IEEE, Vol 75, No 9, pp1203-1209, 1987

Chikada Y., 'Correlators for Interferometry – Today and Tomorrow', presented in Astronomical Society of the Pacific Conference Series, Vol 19, Radio Interferometry: Theory, Techniques, and Applications™ (eds Cornwell, T.J. and Perley, R.A.), 1991

Crochiere, R.E. & Rabiner, L.R., 'Multirate Digital Signal Processing', Prentice-Hall, New Jersey, 1983

Escoffier, R., 'The MMA Correlator', ALMA memo 166, April 1997, http://www.alma.nrao.edu/memos/

Ferris, R.H., Bunton, J.D., Stuber, B., 'A 2GHz Digital Filterbank Correlator', The SKA: Defining the Future, Berkeley California, July 9-12, 2001. http://www.skatelescope.org/skaberkeley/data/posters/Posters/Posters%2011-20/Posters%2011-20%20(PDF%20version)/Poster%2020/poster20.pdf or http://www.atnf.csiro.au/SKA/techdocs/DFCHandout.pdf

Okumura, S.K., Chikada, Y., Momose, M., Iguchi, S., 'Feasibility Study of the Enhanced Correlator for 3-way ALMA I', ALMA memo 350, February 2001, http://www.alma.nrao.edu/memos/

Urry, W.L., 'A Corner Turner Architecture', ATA memo 14, Nov 2000, http://astron.berkeley.edu/ral/ATAImagerDocs.html