The FFT as a Filter Bank
Wilbert Lynn Urry

The Fourier transform makes a terrible filter bank. The pass-band shape of each of its' channels exhibits bad overlap and the channels have bad side-lobes. Its' bad characteristics are due to the fact that a square window is implied for the input vector in the usual form of the discrete Fourier transform. Many authors have studied the problem of using a more favorable input window to improve its' performance[1], however performance improvement usually requires overlapping the input sample vectors[2]. The result is at least a factor of two loss in throughput speed. By adding a simple circuit to the input of a normal discrete Fourier transform, overlap is not required and the pass-band shape can be improved to any extent. I describe only one approach to, what is now, a general method.[3,4]

The discrete Fourier transform is usually written as

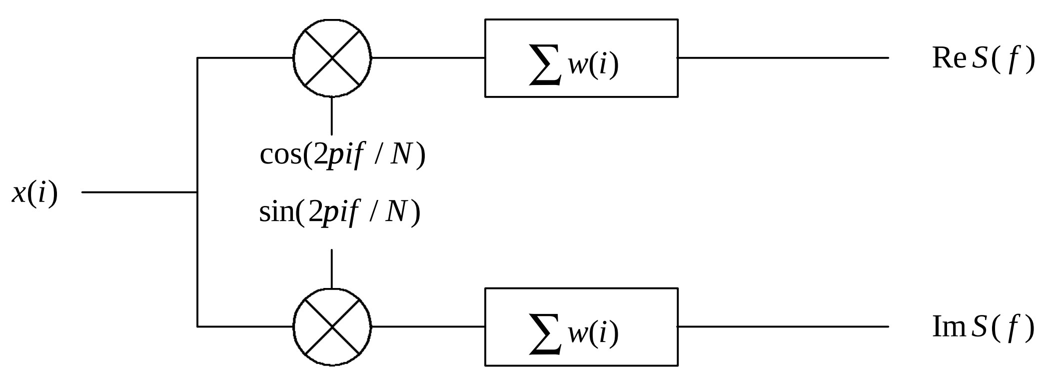
$$S(f) = \frac{1}{N}\sum_{i=0}^{N-1} w(i)x(i)e^{-j2pif/N} \qquad (1)$$

The $w(i)$ term is a windowing function that determines the pass-band shape of each of the bins.

Note that

$$e^{-j2pif/N} = \cos(2pif/N) + j\sin(2pif/N) \qquad (2)$$

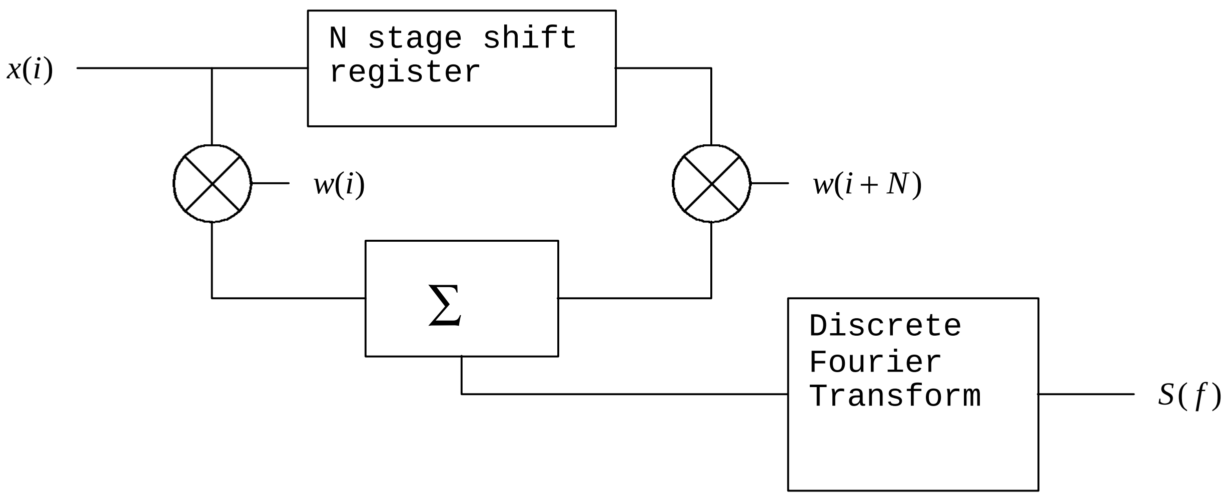
The discrete Fourier transform can be viewed as many quadrature down converters.



There is one down converter for each value of f for $f = 0 \rightarrow N-1$. Note that the Fourier transform is periodic. Whenever $if = N$ the "local oscillator" has gone through one complete turn and the whole thing repeats. A more general discrete Fourier transform may be written

$$S(f) = \frac{1}{N}\sum_{i=0}^{N-1}\left(\sum_{a=0}^{b} w(i+aN)x(i+aN)\right)e^{-j2pif/N} \qquad (3)$$

Any number of input samples may be used in the construction of $S(f)$ and the window may be unlimited in extent. An implementation of this scheme for b=1 is shown below.



In order to window the samples of $x(i)$ as they come in, they must be multiplied by the sampled window function $w(i)$. The window function samples are stored in a look-up table and multiply the input samples as they go by. The result is entered into the Fourier transform in sequence. This arithmetic is needed if any windowing is required. In order to expand the size of the input window, the input sequence is passed to an N stage shift register which delays the input sequence by N, the period of the Fourier transform. Another look-up table is used containing the samples of the extended window. A second multiplier multiplies the delayed input and an adder adds the result to the result of the first multiplier. The sum then enters the Fourier transform in sequence. Any degree of expansion may be added by adding as many shift registers and multipliers as are needed.

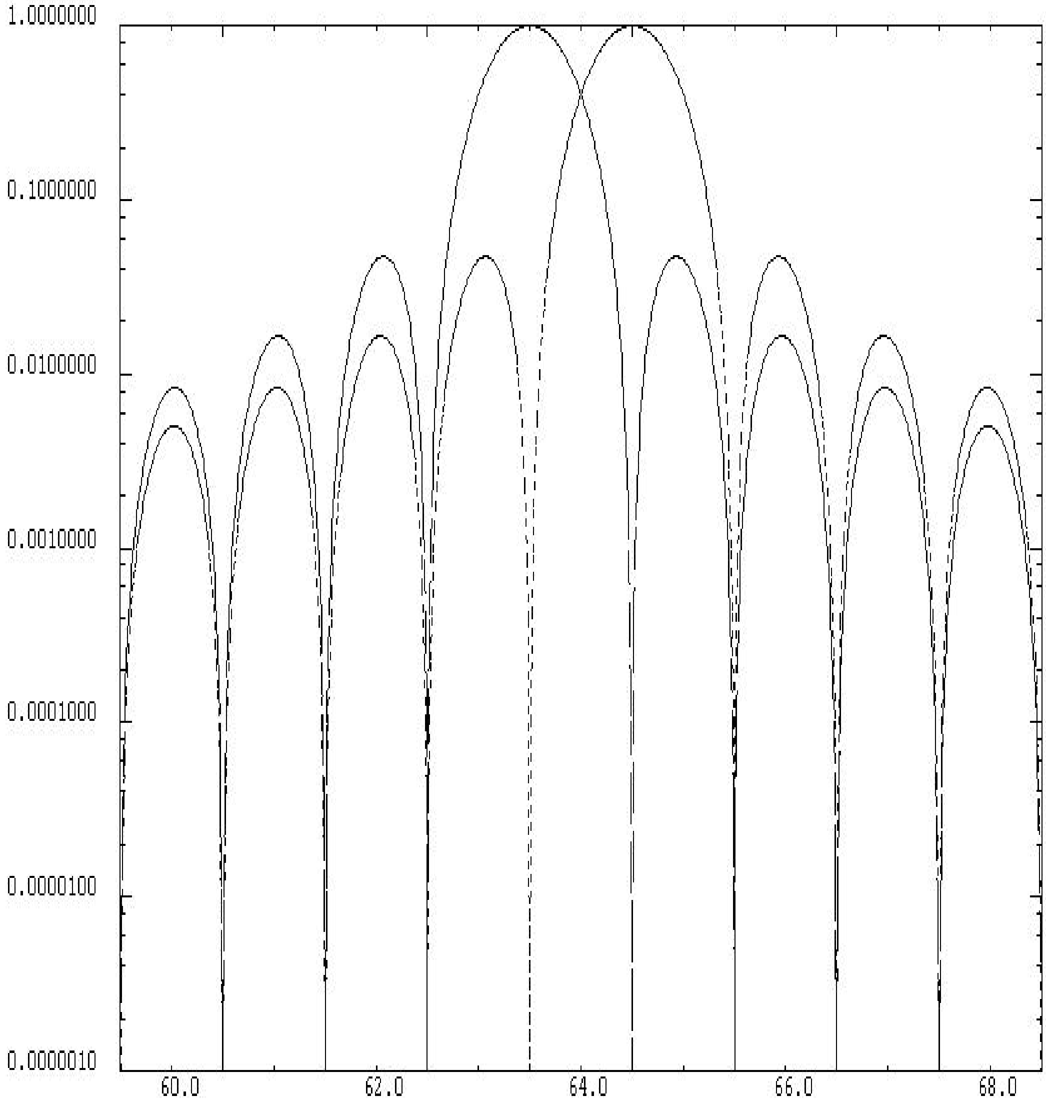


Figure 1

This is a plot of the overlapping pass-band shape for channels 63 and 64 of a 256 channel discrete Fourier transform showing the interfering spectral leakage or sidelobes of those channels caused by a rectangular window that is 256 channels wide. The highest sidelobe level is only 13 db below the main channel.

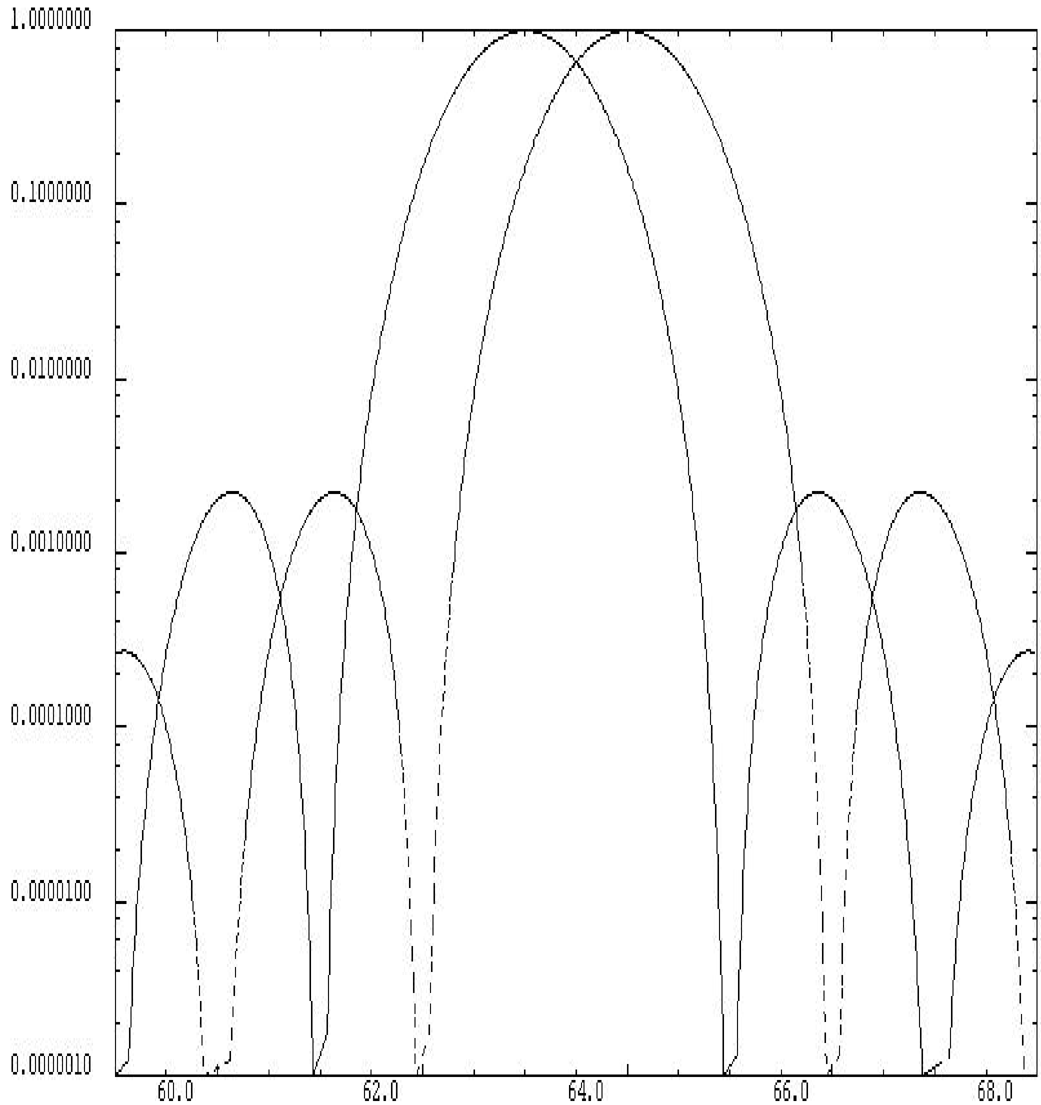


Figure 2

This plot shows the lower sidelobe level but wider pass-band shape caused by a 256 sample triangular window. The highest sidelobe is now 27 db below the peak of the main channel. In order to maintain the noise sensitivity of the discrete Fourier Transform, it is necessary to overlap the input vector by at least 50%. Overlapping causes a reduction in processing speed.

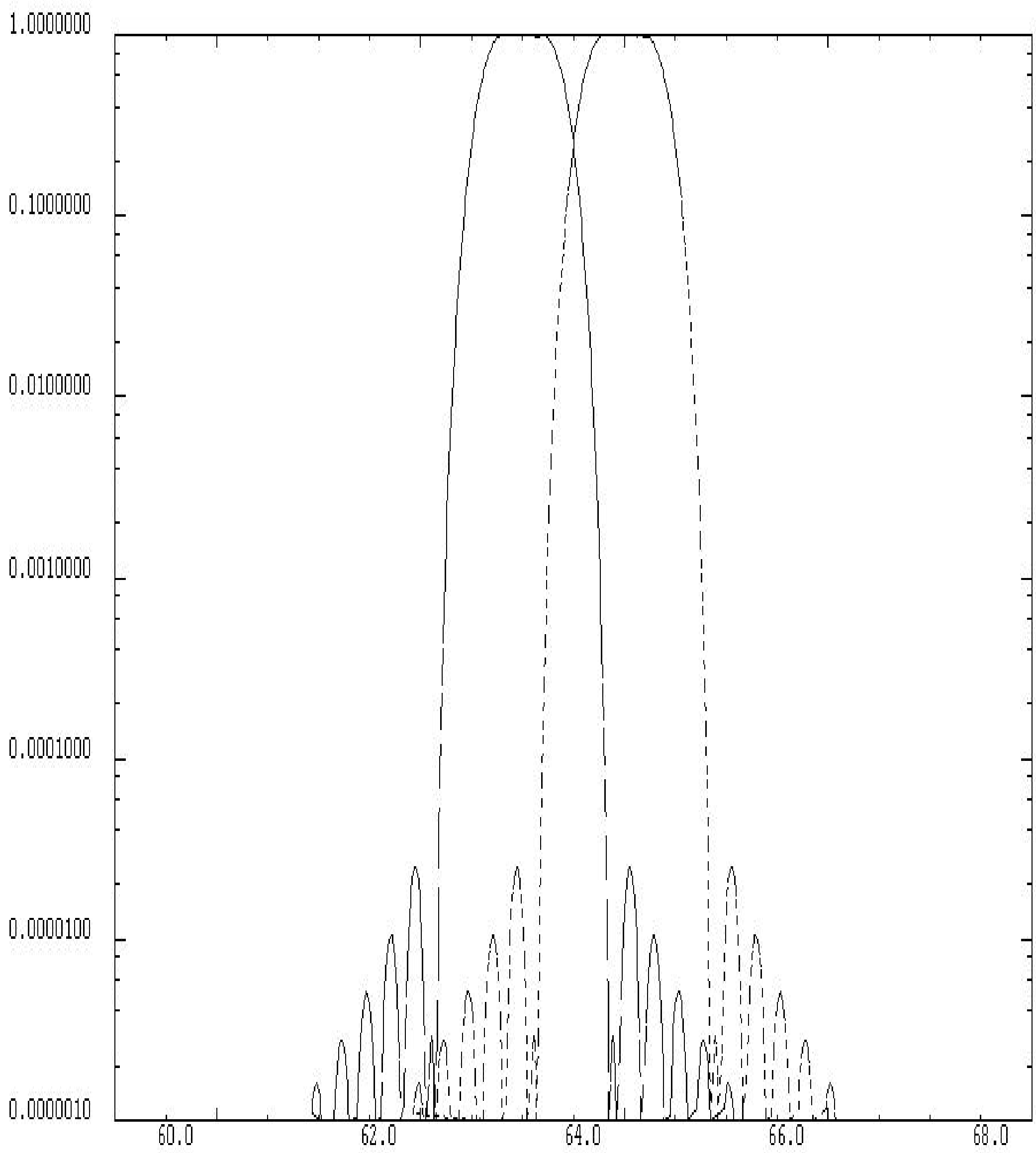


Figure 3

This plot shows the results of extending the window from the usual 256 samples as in the previous examples to 1024 samples. A Gaussian shape was used producing sidelobes that are 47 db down. The windowing required four multipliers, a summer and three 256 stage shift registers. A Gaussian was used because it was handy and these results should not be viewed as an optimum of what this method can accomplish.

[1] Harris, F.J. “On the Use of Windows for Harmonic Analysis with the Discrete Fourier Transform” Proceedings of the IEEE, vol. 66, No. 1, Jan. 1978

[2] Welch P.D. “The Use of Fast Fourier Transform for the Estimation of Power Spectra: A Method Based on Time Averaging Over Short, Modified Periodograms,” IEEE Trans. Audio and Electroacoust., vol. AU-15, pp. 70-73, June 1967

[3] Lim, J.S. and Oppenheim, A.V. “Advanced Topics in Signal Processing.” Prentice Hall, New York, 1988

[4] Fiore, P.D. “Low-Complexity Implementation of a Polyphase Filter Bank,” Digital Signal Processing 8, 126-135 1998