Frequency Domain Theory And Applications

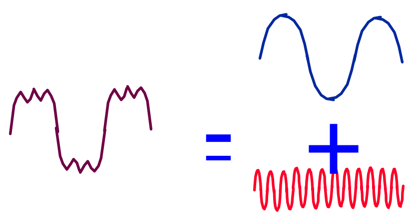
JOHN EDWARDS, BEng. (Hons), CEng., MIEE, MIEEE
Email : jedwards@numerix-dsp.com)
Director, Numerix Ltd, 7 Dauphine Close, Coalville, Leics, LE67 4QQ, UK.

Abstract

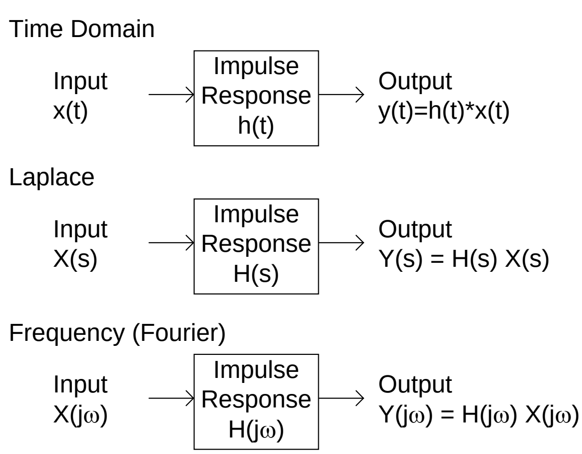
Over 80% of all DSP applications require some form of frequency domain processing and there are many techniques for performing this kind of operation. This two part workshop will serve as an introduction to frequency domain signal processing. In this first session in a two session series, we will cover the theory behind Fourier Transforms and frequency domain processing, while the second half will look at specific frequency domain techniques from an application perspective and show how frequency domain techniques can often provide information that is difficult or sometimes impossible to realise in the time domain. Topics to be discussed in this session include: continuous and discrete Fourier transforms and FFTs; How the FFT works; The complex exponential; Windowing equations and effects.

1. Introduction

All signals have a frequency domain representation and in 1822, Baron Jean Baptiste Fourier detailed the theory that any real world waveform can be generated by the addition of sinusoidal waves. This was arguably developed first by Gauss in 1805. The following diagram shows an example of this process :

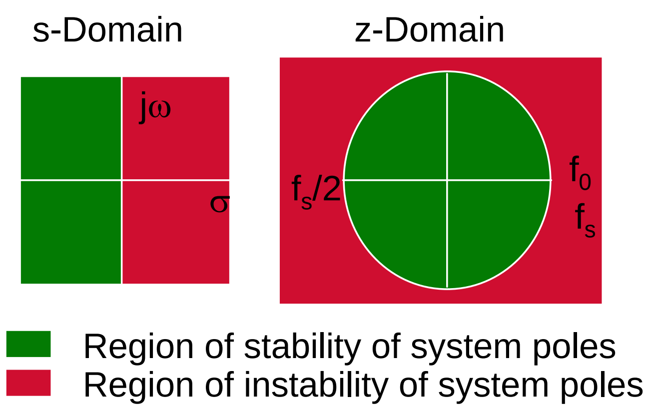


Signals can be transformed between the time and the frequency domain through various transforms. The signals can be processed within these domains and each process in one domain has a corollary in the other, as shown :



The most important process translation between the time and frequency domain is that convolution in the time domain is the equivalent to multiplication in the frequency domain and V.V.

Within the “real”, continuous time world, systems are defined in the s-domain. In the digital world, these systems are translated to the s-domain, as shown in the following diagrams :



Stable systems have their poles (feedback elements) located in the left hand half of the s-plane and these are mapped to a region that lies within the “unit-circle” of the z-domain. Zeros (feedforward elements) can lie anywhere on either plane.

On the imaginary axis of the s plane : $\sigma=0$

$$\therefore \qquad z|_{\sigma=0} = e^{jwT}$$

Thus the jω axis of the s plane maps to a circle of unit radius in the z plane. As ω increases from $-\infty$ through 0 to $+\infty$ the unit circle is retraced every $2\pi^c$.

In the continuous time domain, e^{-sT} is a unit delay, This is represented by z^{-1} in the discrete time domain, where $z = e^{sT}$

An arbitrary delay : $\delta(t-nT)$ is represented by :

$$e^{-nTs}$$

and

$$z^{-n}$$

i.e. : The Laplace transform is equivalent to the Fourier Transform for $s = j\omega$. The Laplace transform is given in the following equation :

$$L(t) = \int_{-\infty}^{\infty} x(t) e^{-st} dt$$

and this translates to the following in the z-domain :

$$X(z) = \sum_{n=-\infty}^{-1} x(n) z^{-n} + \sum_{n=0}^{\infty} x(n) z^{-n}$$

This is referred to as the two sided z-transform however, for most applications it can be simplified by assuming that at time t=0 then the output is at 0.

Any real world signal or function has a related z-transform however there are many important z-transforms that are worth remembering. Some of them are listed here, with $t = nt_s$ written as n.

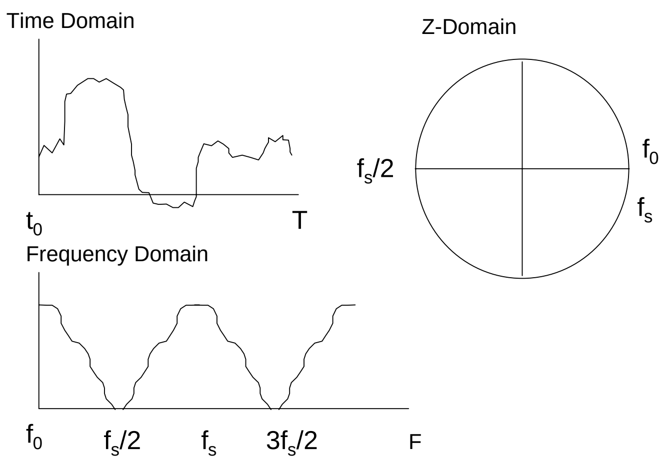
Theorem	***f(n)***	***f(z)***
Linearity	$Ax(n)+By(n)$	$AX(z)+BY(z)$
Shifting	$x(n+n0)$	$z^{n0}X(z)$
Multiplication by an exponential	$a^n x(n)$	$X(z/a)$
Multiplication by time	$nx(n)$	$-z\ (dX(z) / dz\ )$
Convolution	$x(n)*y(n)$	$X(z).Y(z)$
Initial value	$x(n) = 0$ for $n < 0$	$x(0) = \lim_{z \to \infty} X(z)$
Final value	$\lim_{x \to \infty} x(n)$	$\lim_{z \to \infty} [X(z)(1-z^{-1})]$

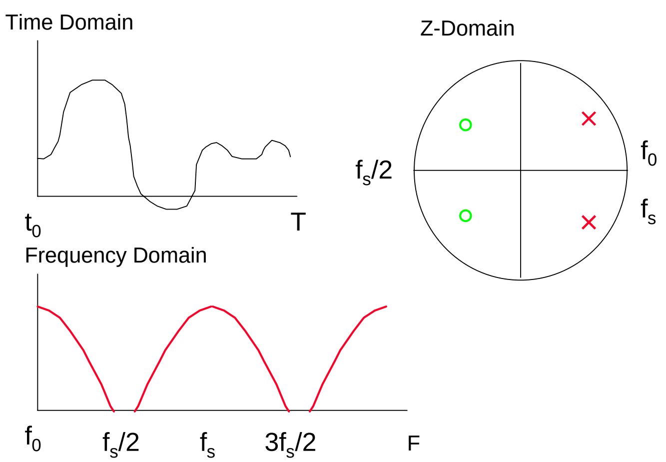
Replacing s = jω in the Laplace transform gives the Fourier transform :

$$\boldsymbol{F} \{x(t)\} = \int_{-\infty} x(t) e^{-j\omega t} dt$$

The FT is equivalent to the LT for $x(t) = 0$ for $t < 0$ and if $x(t)$ converges at $t = \infty$.

As an example of how we might use the different domains, let us consider the following signal and its representation.





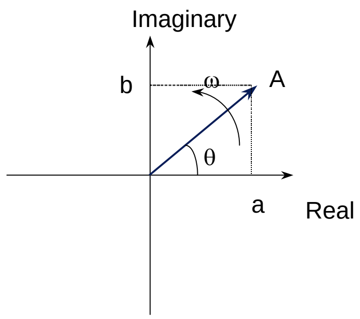
If we now filter the signal with the filter defined in the z-domain, we see the following results :

The poles represent "gain" and the zeros "attenuation. The effects of applying the filter to the signal are frequency dependent and so we see that the filter has a low-pass effect and the signal is smoothed. This operation of the system on the signal is performed by either convolution in the time domain or multiplication in the frequency domain.

2. Representing Signals

Signals can be represented in many different ways. From Fourier's theory, we know that we can represent any real world signal by the combination of two or more sinusoids. Therefore, we need to be able to understand how sinusoids work, in order that we can understand how the complex signals operate.

In the complex domain, we can think of a fundamental signal as a rotating phasor. A phasor is a rotating vector in the complex plane with magnitude A and rotational speed ω radians per second



($\omega = 2\pi f$).

At time t :

$$x(t)=a+jb$$

Where :

$$A = \sqrt{a^2 + b^2}$$

$$\theta = \omega t = \tan^{-1}\frac{b}{a}$$

In polar format :

$$X(t) = Ae^{j\omega t}$$

$$= A\left(\cos(\omega t) + j\sin(\omega t)\right)$$

A complex exponential can be represented as :

If the complex exponential function is defined as :

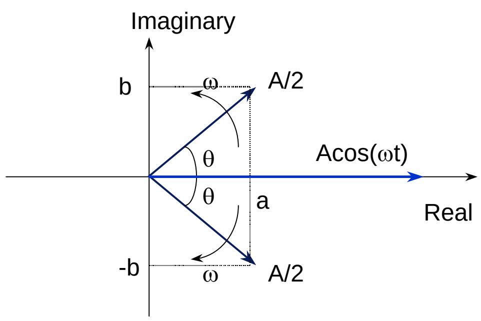
$$e^{j\omega t} = A\left(\cos(\omega t) + j\sin(\omega t)\right)$$

Then :

$$\cos(\omega t) = \frac{e^{j\omega t} + e^{-j\omega t}}{2}$$

$$\sin(\omega t) = \frac{e^{j\omega t} - e^{-j\omega t}}{2j}$$

A cosinusoid can be represented by a conjugate pair of phasors with a purely real result, similarly a sinusoid is represented by a conjugate pair of phasors with a purely imaginary result, as shown :



Phasors rotate in two directions and have the following characteristics. Positively rotating phasors rotate anti-clockwise and represent positive frequencies, whilst negatively rotating phasors rotate clockwise and represent negative frequencies.

Referring to the Fourier transform, this just splits the signals up into the fundamental phasors, or complex exponential components.

Signals can be processed or systems analysed in both the time and frequency domains, the following table shows the various theorems and how they relate to each other.

Theorem	***x(t)***	***X(f)***
Dependence	$x(t), y(t)$	$X(\omega), Y(\omega)$
Linearity	$Ax(t)+By(t)$	$AX(\omega)+BY(\omega)$
Time Shifting	$x(t-t0)$	$e^{-j\ \omega t0}\ X(\omega)$
Frequency Shifting	$e^{-j\ \omega t0}\ x(t)$	$X(\omega-\omega_0)$
Convolution	$x(n)*y(n)$	$X(\omega).Y(\omega)$
Even Real *x(t)*	$x(t) = x(-t)$	$ImX\ (\omega) = 0$
Odd Real *x(t)*	$x(t) = -x(-t)$	$ReX\ (\omega) = 0$

So far, we have primarily considered the continuous domains but for DSP we need to consider the discrete equivalents. The discrete Fourier transform (DFT) is given in the following equation and it shows that for every frequency, the Fourier Transform $X(k)$ determines the contribution of a *complex sinusoid* of that frequency in the composition of the signal $x(n)$.

$$X(k) \quad = \quad \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi nk}{N}} \qquad for\ 0 \le k \le N-1$$

The corollary of the Fourier transform is the inverse Fourier transform, as follows :

$$x(n) = \frac{1}{N}\sum_{k=0}^{N-1} X(k)\, e^{j\frac{2\pi\, nk}{N}} \quad for\ 0 \le k \le N-1$$

When using the FT, it is important to be aware of several issues, including :

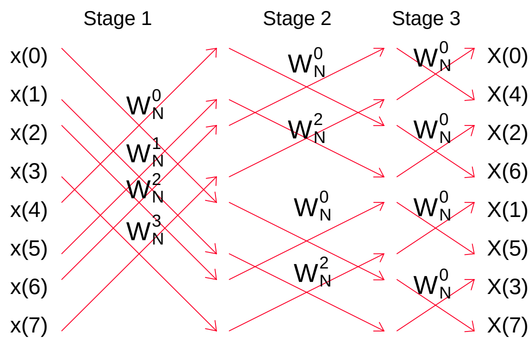
- The phase (the sign of the sine term)
    - An Engineers forward FT is the same as a Physicians inverse
- The scaling (1/N)
- Uncertainty Principle
    - Increased frequency domain resolution == reduced time domain resolution and v.v.
- Continuity
    - Time domain continuity == frequency domain discontinuity and v.v.
    - E.G. A continuous time domain sinusoid is a frequency domain impulse

3. The Fast Fourier Transform (FFT)

There are many ways to approach an understanding of the FFT however there are some heuristic approached that explain the operation and can be used to extend the techniques. The Fourier transform can be considered to be a bank of band-pass filters that takes in a signal and the magnitude of the output of each filter is proportional to the total input energy into that filter. Each of these filters is convolving the input with a set of filter coefficients that are sinusoidal in nature, with the frequency of oscillation equal to the centre frequency of the filter. When performing the convolution over all the banks, many of the multiplications of data and coefficient values are repeated and therefore redundant. Computation saving can be made by implementing a single Fourier transform as 2 half sized Fourier transforms (i.e. Two N/2 point DFTs are faster than one N point DFT). Extrapolating this to its limit, a 2 point DFT is often the optimum process and larger FT operations can be constructed from this small building block. This leads to the fact that FFT lengths are usually powers of 2. This is a generalisation and there maybe architectural reasons why other lengths are preferential and it is not unknown to mix the sizes of the blocks – termed mixed radix transforms.

For the purposes of this paper we will only discuss radix-2 transforms. When looking at the computational loading of FTs and FFTs, the former requires order N^2 operations and the latter order $N/2 \log_2 N$ operations.

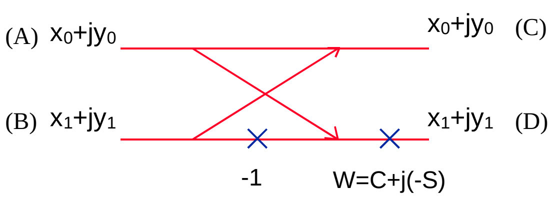
So how do we combine the small FT building blocks into a complete Fourier transform ? The process starts by sub-dividing (decimating) the complete operations and this can be performed at either the time or frequency end of the operation. Taking an 8 point FT, the decimation in frequency operation is as follows :



The WN values are the coefficients of the FFT and are often referred to as *twiddle factors*. The twiddle factors are essentially the complex exponential values, according to the following equations.

$$W_N = e^{j2\pi/N} \quad \text{where} \quad X(k) = \sum_{n=0}^{N-1} x(n) W_N^{-nk}$$

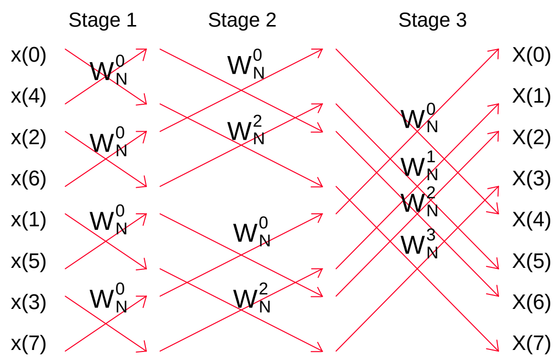
Each of the building blocks has the following structure :



- The corresponding equations are :
- Cr = Ar + Br
- Ci = Ai + Bi
- Dr = (Ar-Br) * Cos(θ) + (Ai-Bi) * Sin(θ)
- Di = (Ai-Bi) * Cos(θ) - (Ar-Br) * Sin(θ)

These structures are referred to as *FFT butterflies* – for obvious reasons !

The decimation in time operation has the following structure :



With the forward and inverse transforms, either the input or output data sets must be in a non-linear format that uses a bit reversed address style. Most modern DSPs are able to handle this directly in hardware so converting the data back into a linear order often requires no overhead to the operation.

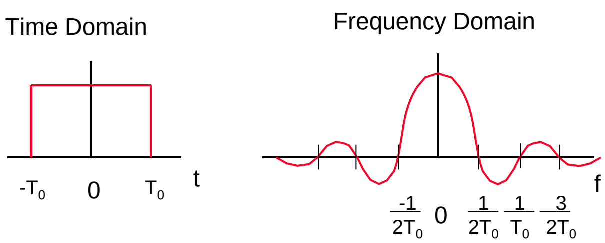
The output of an FFT is limited in both resolution, and dynamic range. Resolution is defined as the gap between to adjacent frequency components (bins) and the dynamic range is the ratio of the smallest signal to the largest signal detectable.

$$\text{resolution} = \frac{\text{sample rate}}{\text{number of samples in buffer}}$$

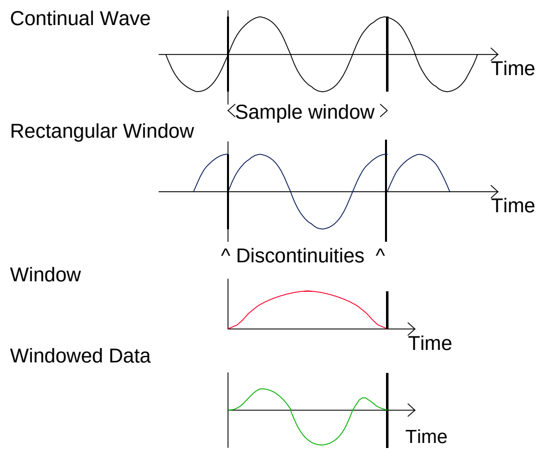
$$\text{dynamic range} = 20.\log_{10}\left(\frac{\text{smallest signal}}{\text{largest signal}}\right)$$

4. FFT Effects And Windowing

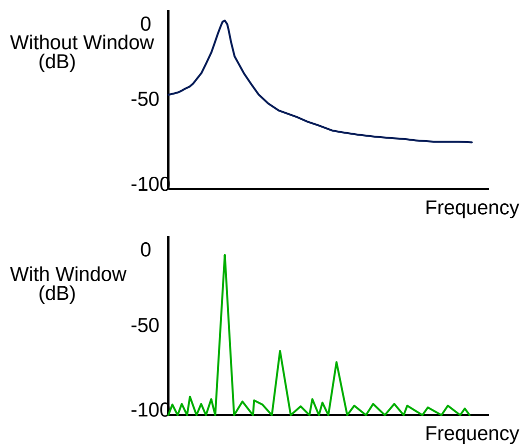
When performing the FFT, the operation is processing a block of data. The frequency domain effect of this is defined by the equation sin(x)/x therefore the magnitude of the sidelobes is independent of the window length. Increasing the window length decreases the sidelobe width only, not the height. The effect is shown in the following diagrams :



Looking at the effects from a different perspective we see the *edge effect* more clearly. Using this approach we can also see how to remove the edge effect by using a window that tapers to zero at the extremities.



The frequency domain effect is also clear :

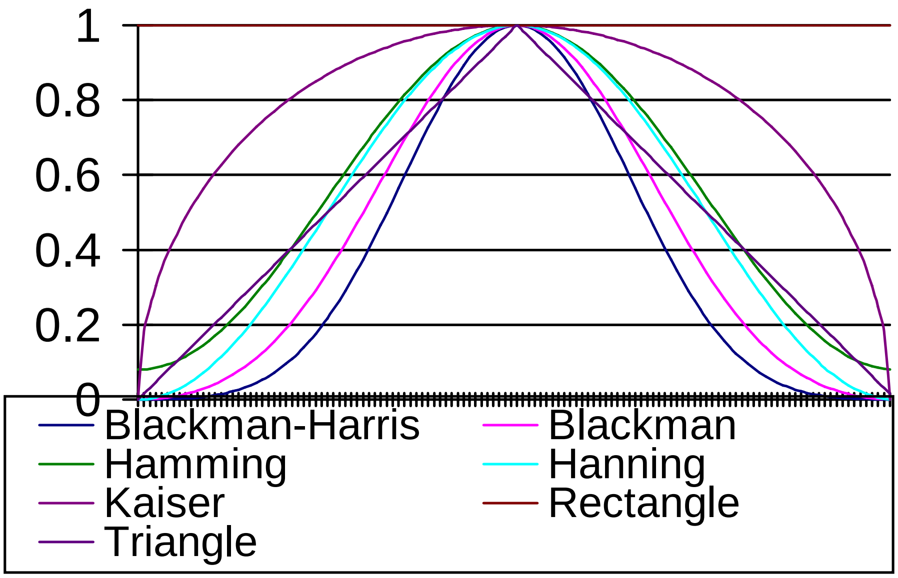


The windowing functions are typically developed from frequency domain requirements and a typical example is the Hanning Window, which is defined by the following equation :

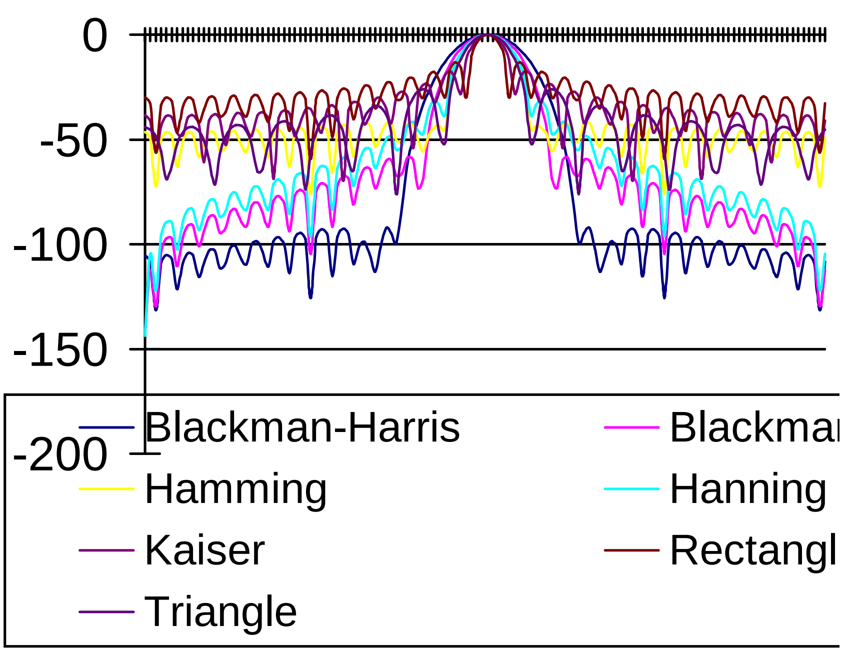
$$w(t) = \frac{1}{2}\left(1 - \cos\left(\frac{2\pi t}{T_c}\right)\right)$$

There are many “standard” windowing functions and the time and frequency domain performances are shown.

Time Domain Windowing Effects



Frequency Domain Windowing Effects



The performance of the windows is measurable via various parameters, including :

- Central peak width
- 6-dB point
  - Indicates how close signals can be before they can no longer be resolved
- Highest sidelobe
- Sidelobe fall off
- Equivalent noise bandwidth (ENBW)
- Specified how concentrated the spectral information is

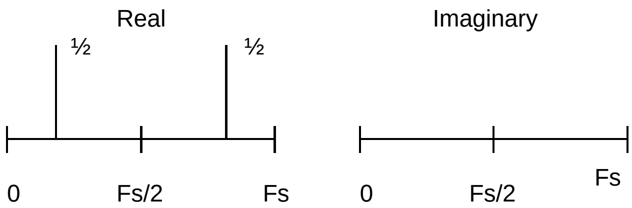
The important parameters for the main window functions are :

- Triangle
  - Simple computation - no lookup table
  - Narrow spectral peak (-6dB @ 1.21 bins)
  - Large sidelobe
- Cosine Bell
  - Moderate sidelobe rejection (-32 dB 1st lobe)
  - Narrow peak
- Hamming
  - Moderate peak width
  - Poor sidelobe rejection
- Blackman
  - Good sidelobe rejection (-60 dB 1st lobe)
  - Broad central peak
- Kaiser
  - Very Good sidelobe rejection (-70 dB 1st lobe)
  - Broad central peak (-6dB @ 2.39 bins)
  - excellent ENBW

Some window functions, like the triangular window, are very easy to calculate on-the-fly but with modern DSPs it is more common to place the data in a buffer in memory.

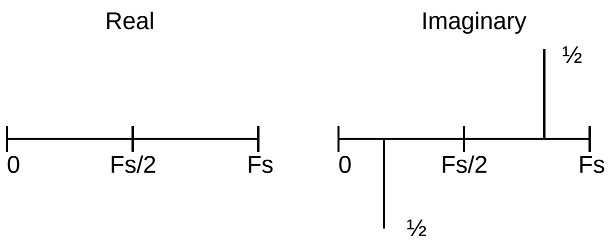
In order to understand the output of the FFT, it is important to analyse the outputs generated by the basic signals. These can be easily related to the phasor systems. Note where the DC component lies, the examples have a wavelength integer number of bins. These components represent the phasors (magnitude, frequency and phase).

FFT of a pure cosine waveform :



Real component contains +ve and -ve frequencies

FFT of a pure sine waveform :



Imaginary component contains +ve and -ve frequencies

5. Post FFT Processing

FFT processing is not useful in itself, it is the post FFT processing that is usually the important issue, it defines what information can be extracted from the information. The following equations show how to calculate the magnitude and phase of the signals.

$$\text{Magnitude} = \sqrt{\text{Real}^2 + \text{Imaginary}^2}$$

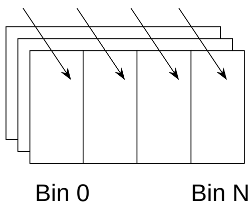
$$\text{Log Magnitude} \quad = 20 * \log_{10}\left(\sqrt{real^2 + imag^2}\right)$$

$$= 10 * \log_{10}\left(real^2 + imag^2\right)$$

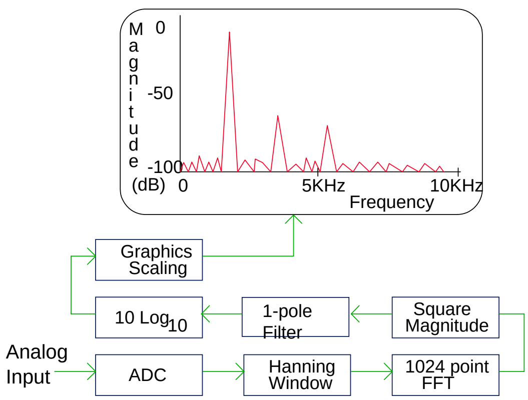
$$\text{Angle} = \tan^{-1}\left(\frac{\text{Imaginary}}{\text{Real}}\right)$$

Power Spectrum Estimation is one of the common post FFT calculations and is calculated by averaging the outputs from successive FFTs. This is usually implemented by applying a one-pole averaging filter across frames, as shown :

$$y(n) = x(n) * \alpha + (1- \alpha)*y(n-1) \quad 0 < \alpha < 1$$



The final result – an FFT based spectrum analyser would look like the following :

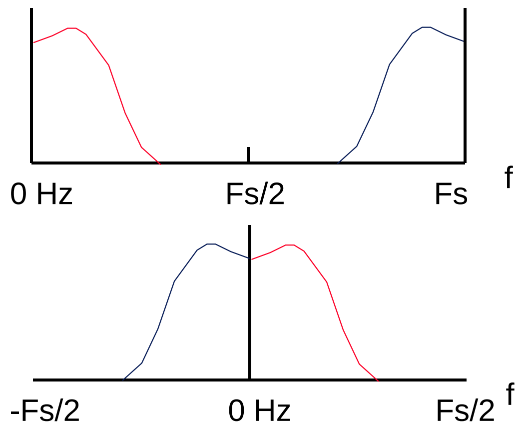


When implementing an FFT based system there are some standard suggestions that should be considered, these include :

- Pre-calculate the coefficients
- Window coefficients (N)
- Twiddle factors (3 * N) / 4 sine wave
- Memory
- Use internal memory where possible
- Align buffers on circular buffer boundary
- Consider dynamic range / scaling issues
- For PSD $Z = \sqrt{(x^2 + y^2)}$
- Therefore ½ output redundant - Saves computation
- Turn on the cache

6. FFT Output Analysis

The output of an FFT can be viewed in two basic formats. The difference is the location of the DC component and this is shown in the following diagrams :

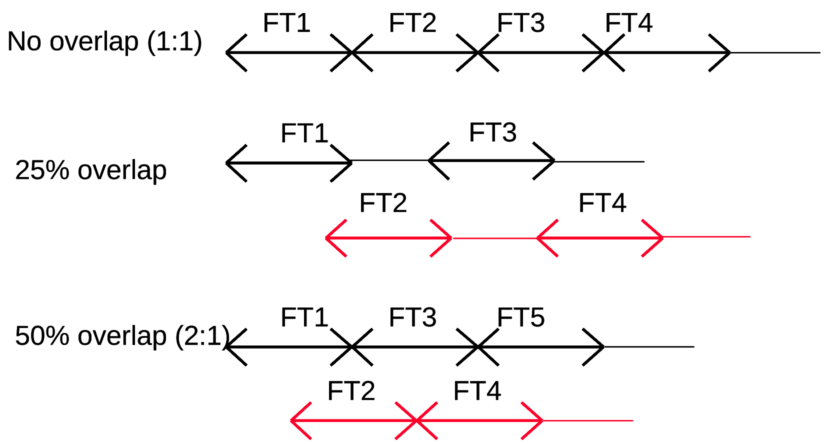


The first format is often used in baseband applications e.g. speech and system analysis, where as the second is more common in H.F. communication applications, where f_c is is hetrodyned down to D.C. The results are no different but sometimes make post processing easier.

The "noise floor" of the FFT output depends on the size of the FFT. Averaging the signal means that the coherent signal (e.g. sinusoid) power increases by 6dB for doubling the number of samples. The average noise power however increases by 3dB for twice for the same increase in the number of samples. The SNR gain is therefore 3dB per order.

Parceval's theorem says that the average power of a periodic waveform is equal to the sum of the average powers carried by its separate frequency components individually. This is often used to calculate relative powers in-band and out-of-band e.g. Signal-to-Noise-Ratio (SNR).

The FFT is a short time Fourier transform, edge effects cause discontinuities, to overcome these effects it is necessary to window the data. Unfortunately windowing can mask some important artefacts within the signal i.e. there can be a loss of vital information from continuous input at block edges. The solution is to overlap the FFTs. For input only applications we need to overlap inputs, for applications with continuous inputs and outputs it is necessary to overlap both and add the intermediate results. The following diagram shows the process :



•Trade-off - MIPS for throughput

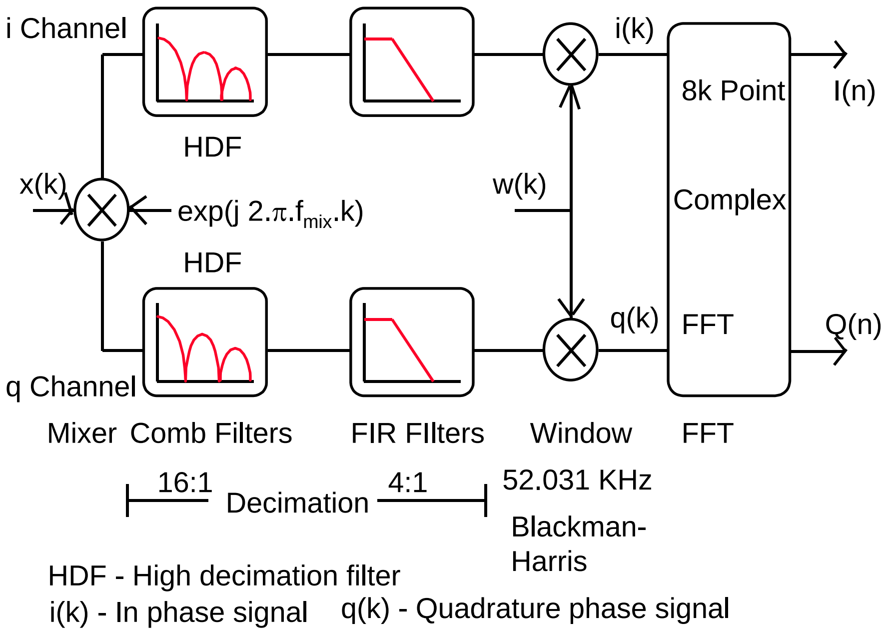
Some applications only require the detection (or generation in the case of the IFFT) of a few frequencies, a useful technique is to "prune" the FFT structure, in such a way that only the essential calculations are executed.

Very often, it is necessary to process data vectors that do not contain a number of samples that is an integer power of 2, in order to process this information, it is common to pad the data with zeros to the correct length. Zero padding does not affect the spectral content or the frequency resolution of the result but it does interpolate the original sample set across more points. This often leads to a more computationally efficient solution for non power of 2 buffer lengths.

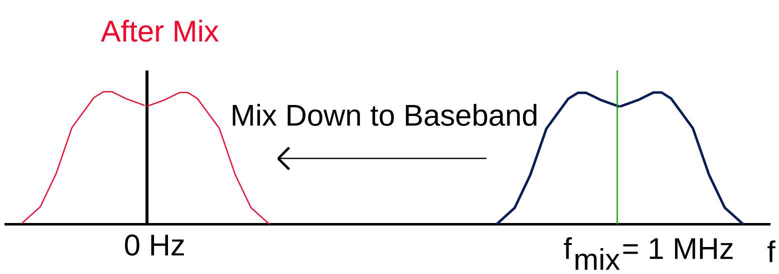
7. The Zoom-FFT

The zoom-FFT utilises complex frequency shifting and decimation to zoom in on a particular frequency band of interest. This increases the frequency domain resolution, increases spectral range and reduces the system hardware cost and complexity.

The following block diagram shows the structure of the zoom-FFT algorithm :



The effect on the signal is :



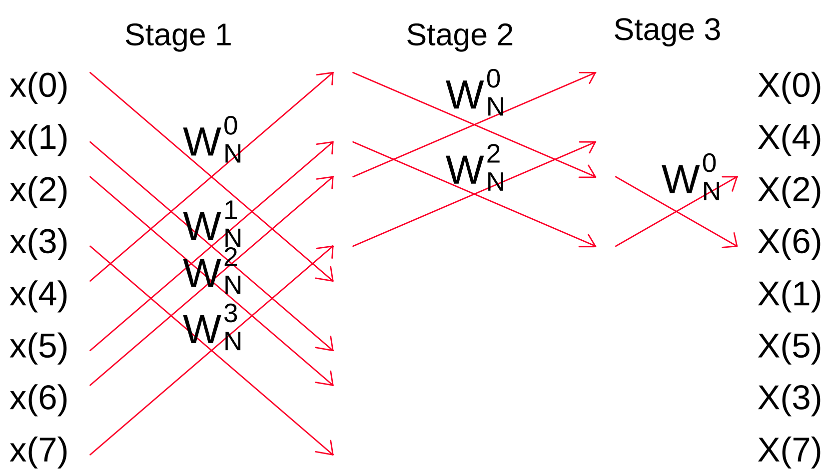
Typical applications of the zoom-FFT include; ultrasonic blood flow analysis, R.F. communications, mechanical stress analysis and Doppler radar.

This technique was used in an application to detect air bubbles suspended in a liquid, which resonate at frequencies that are proportional to their size, this resonance makes the bubbles act as diode mixers consequently when the liquid is stimulated with two signals of different frequencies they are modulated together. The energy in the mixed signal is related to the wavelength of the incident signals and the size of the bubble, therefore the bubble size can be detected by looking at the returned signal.

The ultrasound carrier operates at 1 MHz, which necessitated a high speed (3.33 MHz) sampling front end. The high performance (120 MFLOPS) DSP sub-system and easy to use PC-DOS based Man Machine Interface.

The system has to produce a Fast Fourier Transform (FFT) of a region of the frequency spectrum centred around a frequency of approximately 1 MHz. The frequency band of interest is approximately 25 kHz either side of the 1 MHz centre frequency, giving a total bandwidth of approximately 50 kHz. The required frequency resolution is 5 Hz. A Zoom FFT method is used to generate the required spectral data, since in order to obtain the required spectral resolution using a standard FFT would require an unfeasible large transform.

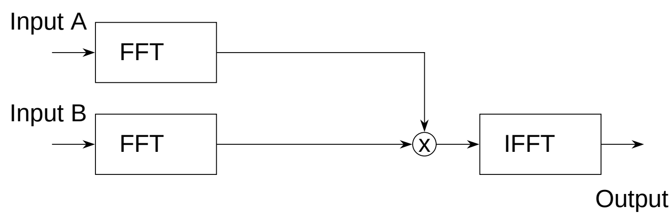
Another technique for reducing the processing requirement for some applications is to prune the FFT. This is useful when only a few frequencies need to be analysed.



8. Frequency Domain Convolution And Correlation

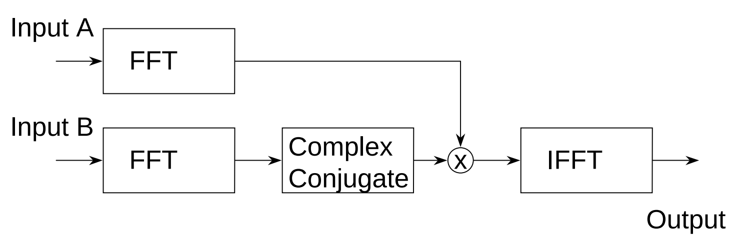
Convolution in the time domain is equal to multiplication in the frequency domain and vice versa. Correlation in the frequency domain is equivalent to convolution, with one array time reversed. The output length must be greater than N + M – 1, where N and M are the lengths of the input vectors. It is important that the FFT length is also therefore greater than N + M – 1. The first thing that is required is that the inputs require zero-padding, otherwise the result is circular convolution / correlation. The benefits of this are the potential large computational savings. For many applications it is possible to pre-compute the convolution or correlation kernel FFT for more efficiency.

The following block diagram shows the operation :



The FFT of a system (e.g. a filter) impulse response is the transfer function, the finite length transfer function is applied to an “infinite length” signal so an overlap method will be required (E.G. Overlap and add). It is important to use the correct coefficients E.G. for a low-pass filter can not just set high frequency coefficients to zero as this generates a time domain sin(x)/x function.

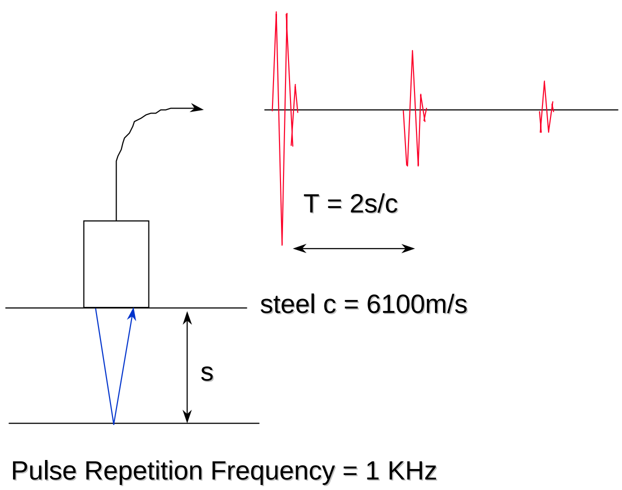
The following diagram shows the configuration of the frequency domain correlation operation.



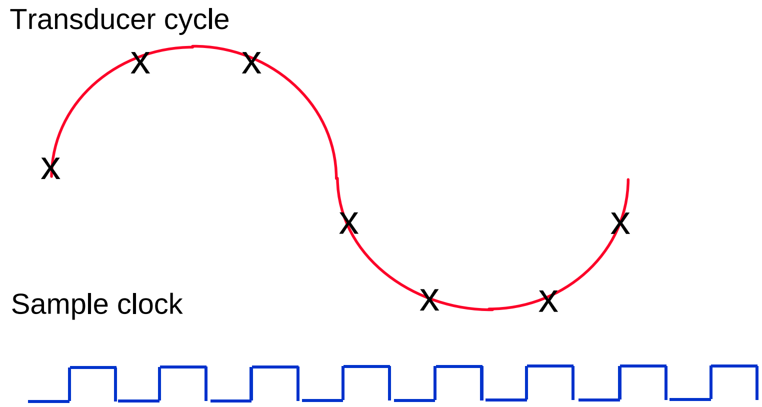
9. An Ultrasonic Time-Of-Flight System

There is a need within the steel industry for accurate measurements of steel sheet thickness. This can be achieved by making ultrasonic pulse-echo measurements through the steel and calculating the time-of-flight of the ultrasound. Since the velocity of the ultrasound in the steel is known, the thickness of the sheet can thus be calculated. A further requirement of the steel industry is the assessment of the quality of the sheet steel. One of the flaws that is often found within steel is due to a phenomenon known as residual stress. This is caused by local deformations in the crystalline structure of the steel which occur during the cooling phase after hot rolling. Residual stresses within a structure cause local variations in the velocity of the ultrasound, therefore a time-of-flight measurement which falls outside a given range may also be due to such flaws within the steel.

The following diagram shows a typical system configuration for a sheet steel thickness test system :



The analog signal from the transducer is sampled at 10 MHz and passed to the Dual Sharc DSP system for processing, in a production line application this would be a continuous process. The algorithm then uses a method which involves coherent averaging of the incoming data, which is then interpolated using a frequency-domain technique (in order to increase the temporal resolution of the dataset). The time taken between successive reverberations of the ultrasound within the steel can then be detected using a process such as autocorrelation. The distance between the peaks in the autocorrelation corresponding to the time between reverberations. The

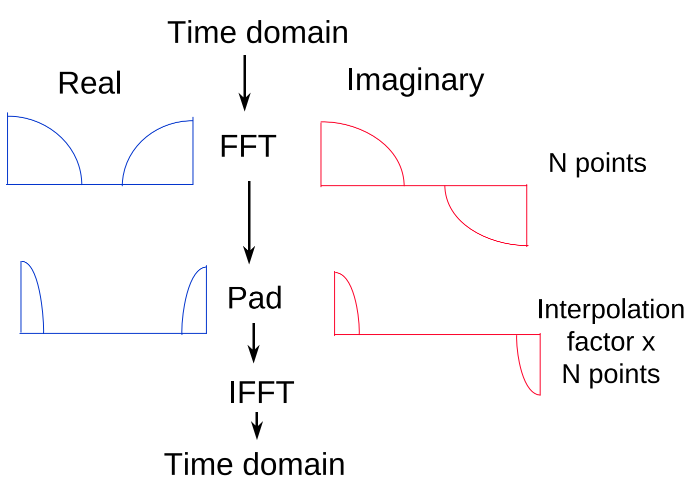


spatial resolution:

s = +/- 0.25 x c x d

frequency domain interpolation process is shown graphically as :

Zero padding of the frequency domain samples is equivalent to sin(x)/x interpolation in the time domain and this helps with more accurate location of the zero crossing points. The operation is shown below :



The far face reverberations can then be detected by autocorrelation.

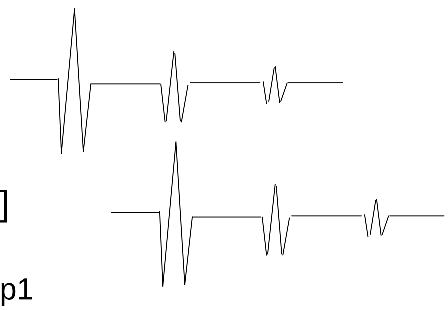
$$y[k] = \sum x[n] \cdot x[n+k]$$

x[n]

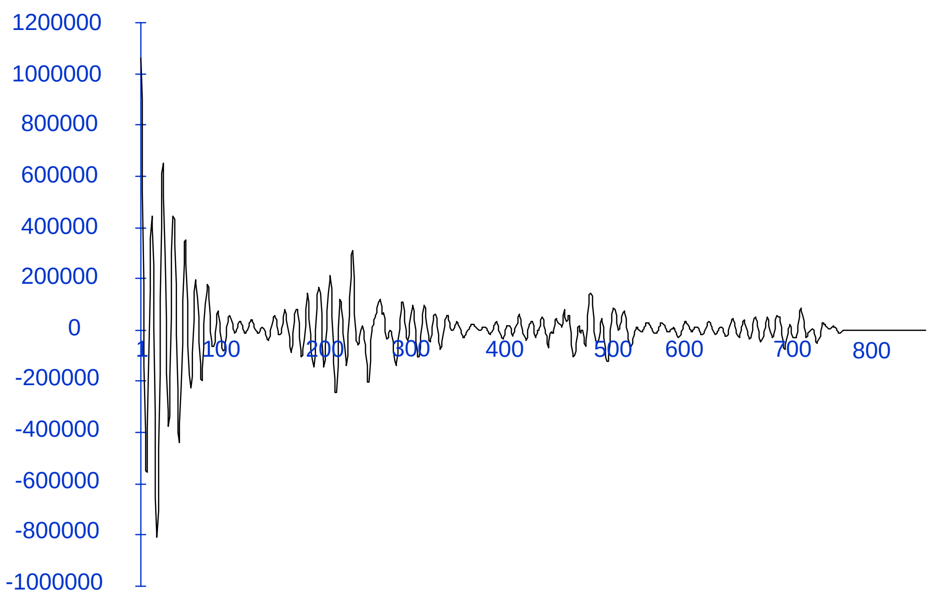
x[n+k]

where

k = p2 - p1

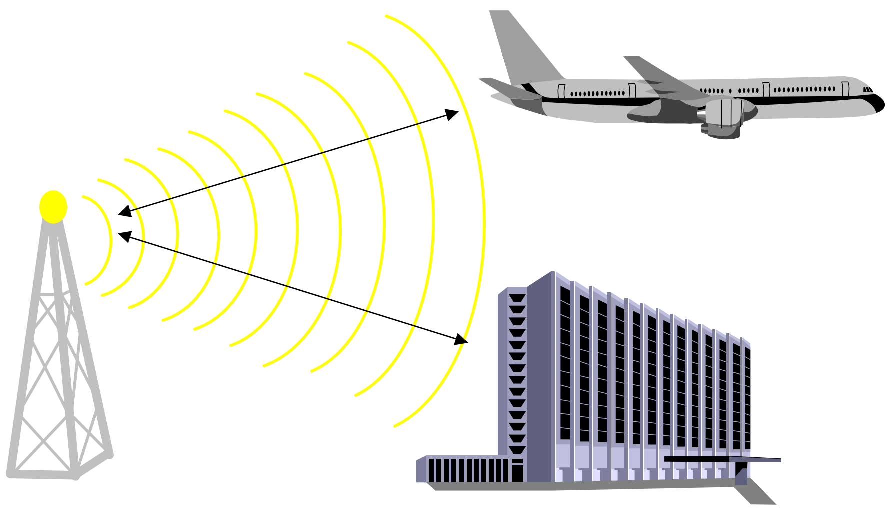


This final plot shows the output from a single generated pulse :

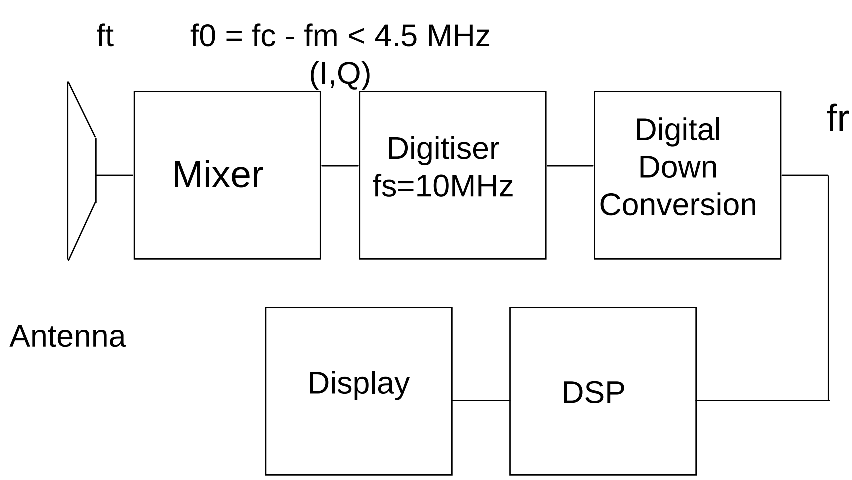


10. Radar Signal Processing

The system specification was to meet a 40 μ seconds pulse repetition interval (PRI), with 1 to 32 range gates. The processing was to be FFT based and required 300 MFLOPS, which mapped onto six 50 MHz TMS320C40s. The system required a 10 MHz sample rate and 12 bit quantisation. The following diagram shows a typical radar example, with stationary and moving targets :



The system configuration is shown in the following diagram :



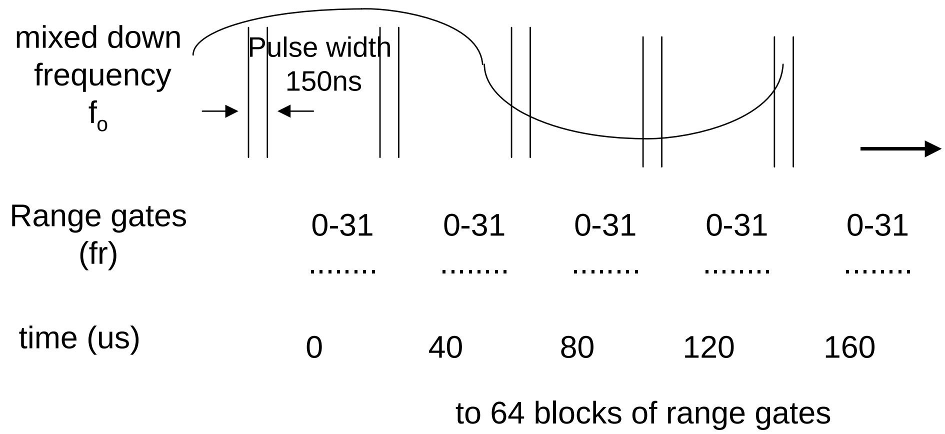
The Doppler shifted frequency from a moving object is shown in the example :

$f_d = 2 \times vr \times f_o/c$ ($c = 3 \times 10^8$ m/s)

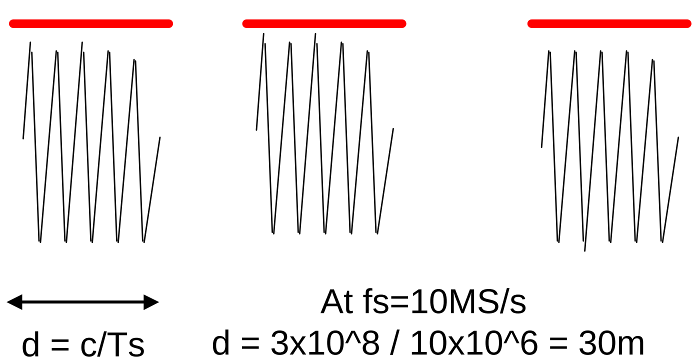
E.G. $f_o = 3$ GHz, vr = 300 m/s (almost mach 1)

$f_d = 2 \times 300 \times (3 \times 10^9)/(3 \times 10^8) = 6$ KHz

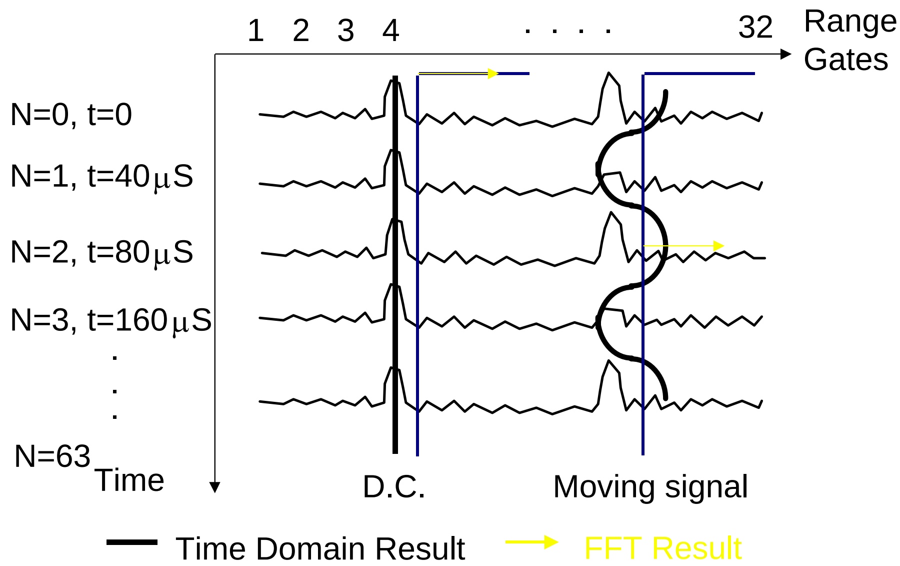
Range gating is the technique for analysing the motion of objects at different ranges from the radar system. The radar outputs pulses at given time spacings and starts sampling the input after a delay. The larger the delay, the greater the distance from the radar. For each block of samples taken the nth sample is at a particular range – over a sequence of sampled blocks it is possible to look at all the samples at a particular range, this is referred to as the range gate.



Each range gate corresponds to a number of cycles at the transmitted frequency, as shown below.

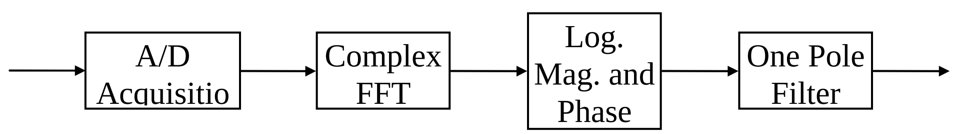


The whole process is shown pictorially below, along with the FFT magnitude results from the house and aeroplane example :

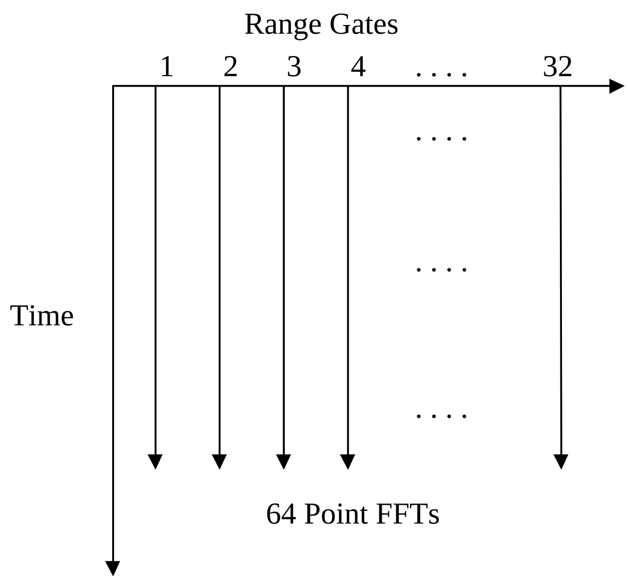


The sinusoidal result seen above is due to the phase of the returned pulses being modified by moving target. The D.C. signal is generated by the stationary target (the house). The moving signal is generated by the moving target (aeroplane). The magnitude of the FFT gives the size of the target, the FFT bin gives the speed of the target.

The DSP process is shown below :



The Arrangement Of The FFTs is :



Range gate resolution (meters) = wavefront velocity * sample rate

$= 3 \times 10^{8} \times 100 \times 10^{-9}$

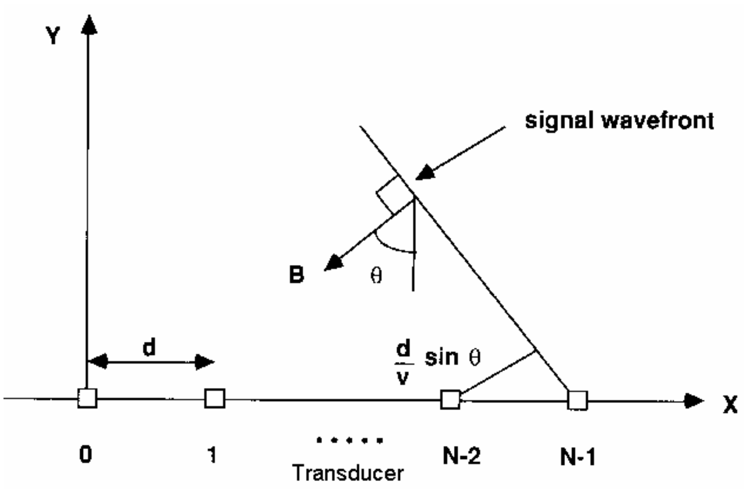
= 30 M

Total range = range gate resolution * number of gates

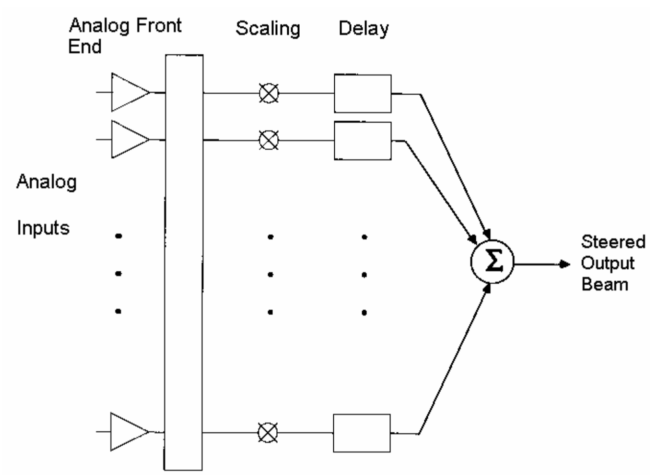
=32 x 30 = 960 m

11. Frequency Domain Beamforming

Beamforming is the acquisition of a signal using an array of transducers and then applying a delay (at its most simplest), to “steer” the direction of the “beam” in a particular direction. Essentially the direction of the beam is relative to the sampling phase of the transducers, as shown in the diagram below :



In the time domain, the beams are steered in a single direction, using the following system :

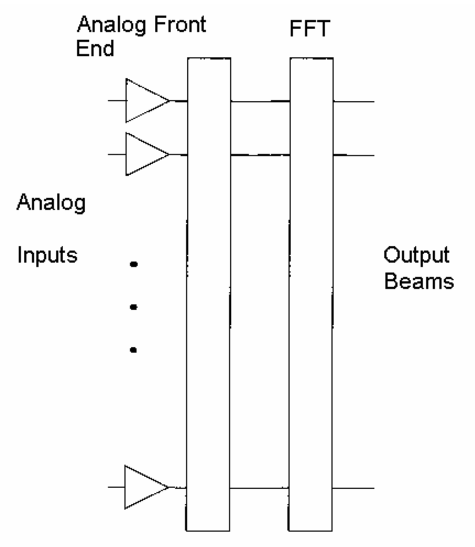


One of the limitations of time domain beamsteering is that one bank of delays can only steer the beam in one single direction. The solution is to use the fact that time delays in the time domain are equivalent to complex exponential multiplication in the frequency domain, using the following equation.

$$\text{If :} \quad x(n) \leftrightarrow X(e^{j\omega})$$

$$\text{Then :} \quad x(n-m) \leftrightarrow e^{-j\omega m} X(e^{j\omega})$$

Essentially this says that we can translate the signals to the frequency domain and cross multiply by the exponential function and the signals will be delayed by the appropriate amount. The benefit of performing this operation in the frequency domain is that we can effectively steer the beam in all directions at the same time.



12. Goertzel filters

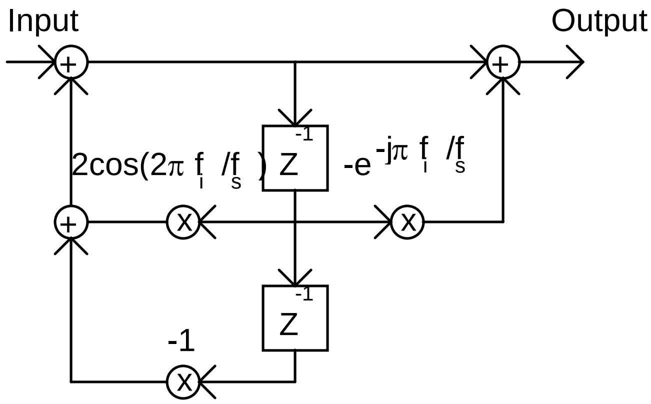
A Goertzel algorithm is a very efficient technique for selecting a particular pass band in a filtered signal. The Goertzel algorithm is defined by the equation :

$$H f_i(z) = \frac{1 - e^{\frac{2\pi f_i}{f_s}} z^{-1}}{1 - 2\cos\left(\frac{2\pi f_i}{f_s}\right) z^{-1} + z^{-2}}$$

Where f_i is the frequency of interest and f_S is the sampling frequency.

The algorithm is most commonly implemented as a second order recursive IIR filter, as shown in the following flow diagram.

Second Order Recursive Goertzel



This filter does not maintain the complex (phase) information but the Goertzel filter is often used to detect particular individual frequencies, a common application is the detection of DTMF tones. This algorithm is very efficient, when compared with the regular FFT, especially when the requirement is only to detect a few individual frequencies.

Conclusion

Translating signals from the time to the frequency domain can allow a whole new and powerful area of signal processing. All real signals can be generated from the sum of the component sinusoids. The Fourier transform will extract the information of phase and magnitude. It is important to be aware of the effects of scaling and edge effects. To remove the edge effects of the rectangular window, it is important to use the appropriate windowing function for the application.

Appendix A - Decibels

In many applications, the use of a logarithmic representation of values gives many advantages, this section discusses the use of decibels (dB).

The decibel is defined as :

$$dB = 10\log_{10}\frac{P_1}{P_2}$$

Where P_1 and P_2 are measures of power.

In many applications the voltage (V) or current (I) is measured, power is proportional to V^2 or I^2, giving the equivalent dB equations of :

$$dB = 20\log_{10}\frac{V_1}{V_2}$$

and

$$dB = 20\log_{10}\frac{I_1}{I_2}$$

For many applications, P_2 and V_2 assume standard values, for example in telecommunications P_2 = 1milliWatt in 600 Ohms, in this case the logarithmic scale is dBm. Note that dBA is not relative to Amps but is used as a measure of sound intensity. One of the advantages of using logarithms is that a multiplication of the linear values can be achieved by the addition of logarithms.

To revert back to a linear scale the following equation is used for power :

$$\frac{P_1}{P_2} = 10^{\frac{dB}{10}}$$

with the equivalent voltage equation being :

$$\frac{V_1}{V_2} = 10^{\frac{dB}{20}}$$

The following table shows some of the basic equivalents between absolute power and dB.

$\frac{P_1}{P_2}$ (dB)	$\frac{P_1}{P_2}$ (absolute)	$\frac{V_1}{V_2}$ and $\frac{I_1}{I_2}$
30	$10^3 = 1000$	10π
20	$10^2 = 100$	10
10	$10^1 = 10$	π
9	8	2√2
8	2π	2.5
7	5	2.24
6	4	2
5	π	1.77
4	2.5	π/2
3	2	√2
2	π/2	1.5
1	1.25	1.12
0	$10^0 = 1$	1
-10	$10^{-1} = 0.1$	π/10
-20	$10^{-2} = 0.01$	0.1
-30	$10^{-3} = 0.001$	π/100

Notes :

The use of π is an approximation of √10 that is useful for quick calculations and accurate to 2 significant figures.

The table shows the absolute power doubles for a 3 dB increase and halves for a 3dB drop. The voltage and current however double for a 6 dB increase.

This table can be used to give all other dB / absolute equivalents :

16 dB = 10 dB + 6 dB = 10 x 4 = > $\frac{P_1}{P_2}$ (absolute) = 40

-74 dB = -(70 + 4) = 1/(10^7x2.5) => $\frac{P_1}{P_2}$ (absolute) = $4x10^{-8}$

Abbreviations

ADC	Analog to digital converter
ADPCM	Adaptive Differential Pulse Coded Modulation
ALU	Arithmetic Logic Unit. The part of the processor that performs the mathematics
AM	Amplitude Modulation
APC	Adaptive Predictive Coding
ASIC	Application Specific Integrated Circuit
BPF	Band-pass filter
CCITT	International Telegraph and Telephone Consultative Committee (Now called ITU)
CDMA	Code Division Multiple Access
CELP	Code Excited Linear Predictive Coding
CODEC	COder-DECoder - used in analog signal sampling
COMPANDER	COMpressor-exPANDER
CPU	Central Processing Unit - the main part of the DSP that executes the instructions
CVSD	Continuously Variable Slope Delta modulator
CW	Continuous Wave
DAC	Digital to analog converter
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DIF	Decimation In Frequency (FFT)
DIT	Decimation In Time (FFT)
DSP	Digital Signal Processing OR a Digital Signal Processor

DTMF	Dual Tone Multi-Frequency - telephone dialling standard
f_s	The sample rate (or frequency) of the system.
FDM	Frequency Division Multiplexing
FDMA	Frequency Division Multiple Access
FFT	Fast Fourier Transform
FIR	Finite Impulse Response filter, one containing no feedback elements
FSK	Frequency Shift Keying - Digital frequency modulation
GMSK	Gaussian Minimum Shift Keying
HF	High frequency - usually refers to applications such as radio communications
LMS	Least Mean Squares - a technique for adapting FIR filter coefficients
HPF	High-pass filter
IDCT	Inverse Discrete Cosine Transform
IDFT	Inverse Discrete Fourier Transform
IFFT	Inverse Fast Fourier Transform
IIR	Infinite Impulse Response filter, one containing feedback elements
ITU	International Telegraph Union (formerly CCITT)
JPEG	Joint Photographic Expert Group (Still image compression standard)
LPC	Linear Predictive Coding
LPF	Low-pass filter
MAC	Multiply Accumulate
MIPS	Millions of Instructions Per Second
MFLOPS	Million Floating Point Operations Per Second, a typical measure of floating-point DSP performance
MODEM	MODulator / DEModulator

MPEG	Moving Pictures Expert Group (Moving video compression standard)
MUX	Multiplexer
PCM	Pulse Coded Modulation
PSK	Phase Shift Keying
PWM	Pulse Width Modulation
QAM	Quadrature Amplitude Modulation
QMF	Quadrature Mirror Filter
QPSK	Quadrature Phase Shift Keying
RELP	Residual Excited Linear Predictive coder
RF	Radio Frequency
SBC	Sub-Band Coding
S/H	Sample and Hold
SNR	Signal to Noise Ratio, common measure of performance for ADCs
TDM	Time Division Multiplexing - A communications system that divides a single communications channel into several smaller ones, using discrete time slots.
TDMA	Time Division Multiple Access
ZOH	Zero Order Hold, an effect of analog signal reconstruction

Glossary

Adaptive Differential Pulse Coded Modulation (ADPCM)	A speech compression algorithm that adaptively filters the difference between two successive PCM samples. This technique typically gives a data rate of about 32 Kbps.
Adaptive equalisation	A filtering system that can allow for the effects of a changing communications medium to be cancelled.
Adaptive filter	A filter that can adapt its coefficients to model a system.
Adaptive predictive coding	An LPC based speech compression technique that uses an adaptive predictive voice source.
Aliasing	The effect on a signal when it has been sampled at less than twice its highest frequency.
Amplitude Modulation	A communications scheme that modifies the amplitude of a carrier signal according to the amplitude of the modulating signal.
Anti-aliasing filter	An analog filter that is used prior to sampling to limit the signal bandwidth to less than half the sample rate (generally low pass) to prevent aliasing distortion.
Asynchronous communications	A communications system where the transmitter and receiver run independently. The beginning and end of the data packet are usually indicated by start and stop bits in the data stream.
Attenuation	Decrease in magnitude.
Autocorrelation	The correlation of a signal with a delayed version of itself.
Band-pass filter	A filter that only allows a single range of frequencies to pass through.
Band-stop filter	A filter that removes a single range of frequencies.
Bandwidth	The range of frequencies that make up a more complex signal.
Barrel shifter	Part of the ALU that allows single cycle shifting and rotating of data words.
Baseband	Signals that have a frequency spectrum based around 0 Hz.

	E.G. speech.
Baud rate	The rate at which symbols are transmitted over a communications channel. A symbol may contain one or more bits of information.
Bit rate	The rate at which bits are transmitted and equals the baud rate * the number of bits per baud.
Biquad	Typical 'building block' of IIR filters - from the bi-quadratic equation.
Butterfly	The smallest constituent part of an FFT, it represents a cross multiplication, incorporating multiplication, sum and difference operations. The name is derived from the shape of the signal flow diagram.
Companding	A logarithmic scheme for sampling analog signals that increases the resolution of signals with a low amplitude. Common standards include A-Law and u-Law.
Convolution	An identical operation to Finite Impulse Response filtering.
Correlation	The comparison of two signals in time, to extract a measure of their similarity.
Data flow architecture	A multi-processing architecture where individual processing elements perform multiple instructions on a many pieces of data.
Discrete Fourier Transform (DFT)	A transform that gives the frequency domain representation of a time domain sequence.
Discrete sample	A single sample of a continuously variable signal that is taken at a fixed point in time.
Echo canceller	A filter that will remove reflected signals on a transmission line that are caused by impedance mismatches.
Equalisation	A filter that will compensate for the effects of a communications channel.
Fast Fourier Transform (FFT)	An optimised version of the DFT.
Finite Impulse Response (FIR)	A filter that includes no feedback and is unconditionally

Filter	stable.
Floating-point	A number scheme that codes a value with a fraction and an exponent and allows a high signal dynamic range.
Frequency Division Multiplexing (FDM)	A communications system that divides a single channel into smaller ones with discrete frequency bands.
Frequency domain	The representation of the amplitude of a signal with respect to frequency.
Frequency Shift Keying (FSK)	A digital modulation scheme that uses a different frequency to represent different binary levels.
Full duplex	Communications in two directions simultaneously.
Gain	Amplification or increase in magnitude.
Half duplex	Communications in two directions, but only one at a time.
Harvard Architecture	A microprocessor architecture that uses separate busses for program and data, this is typically used on DSPs to optimise the data throughput.
High pass filter	A filter that allows high frequencies to pass through.
Hybrid	An analog 2 wire to 4 wire (and vice versa) converter.
Infinite Impulse Response (IIR) filter	A filter that incorporates data feedback. Also called a recursive filter.
Linearity	A measure of the performance of an ADC or DAC to convert signals with different amplitudes, to the same degree of accuracy.
Linear Predictive Coding (LPC)	A speech compression technique that is based on modelling the vocal tract with a time varying filter.
Low pass filter	A filter that allows low frequencies to pass through.
Multi-processing	The division of a process across several processors to improve the performance of the system.

Multi-tasking	The division of processor across several tasks, such that each one is able to receive its required number of processor cycles.
Multiple Instruction Multiple Data (MIMD)	See data flow architecture.
Modulation	The modification of the characteristics of a signal so that it might carry the information contained in another signal.
Parallel processing	The execution of tasks in parallel, either on a single processor via multi-tasking or across several processors by multi-processing.
Pass band	The frequency range of a filter through which a signal may pass with little or no attenuation.
Phase Shift Keying (PSK)	A digital modulation scheme that uses a constant frequency carrier with a variable phase.
Pipelining	A technique commonly used in high performance microprocessors that allows an instruction to begin execution before previous ones have been completed.
Pole	Artefact leading to frequency dependent gain in a signal. Generated by a feedback element in a filter.
Pulse Code Modulation (PCM)	The effect of sampling an analog signal.
Quadrature Amplitude Modulation (QAM)	A variation of PSK that incorporates AM to increase the number of bits per baud.
Recursive filter	See Infinite Impulse Response filter.
Resolution	The accuracy of and ADC or DAC circuit.
Sampling	The conversion of a continuous time analog signal into a discrete time signal.
Sample rate	The inverse of the time between successive samples of an analog signal.

Single Instruction Multiple Data (SIMD)	A multi-processing architecture where individual processing elements perform the same instruction on many pieces of data, also referred to as a systolic array.
Spectrum analyser	An instrument that displays the frequency domain representation of a signal.
Stop band	The frequency range of a filter through which a signal may NOT pass and where it experiences large attenuation.
Synchronous communications	A communications system where the data is transmitted and received at discrete times, which are usually synchronized by a clock signal.
Systolic array	See Single Instruction Multiple Data.
Time domain	The representation of the amplitude of a signal with respect to time.
Transducer	A piece of equipment that converts a physical signal into an electrical signal.
Twiddle factor	The coefficients of the FFT algorithm, typically a ¾ sine table.
Von-Neumann architecture	A traditional microprocessor architecture that uses the same bus for program and data.
z-domain	The discrete frequency domain, in which the $j\omega$ axis on the continuous time s-plane is mapped to a unit circle in the z-domain.
Zero	Artefact leading to frequency dependent attenuation in a signal. Generated by a feed-forward element in a filter.

Bibliography

Analog Devices, *DSP Applications Using The ADSP-2100 Family*, Prentice Hall, USA.

Antoniou, A., (1993), *Digital Filters : Analysis, Design and Applications*, McGraw-Hill, USA.

Bellanger, M., (1984), *Digital Processing Of Signals, Theory And Practice*, John Wiley & Sons, UK.

Burrus, C. S. and Parks, T. W., (1985), *DFT/FFT and Convolution Algorithms*, Wiley Interscience, NY, USA.

Cooley, J.W. and Tukey, J.W., (1965), *An Algorithm for the Machine Calculation of Complex Fourier Series*, Mathematics of Computation, The American Mathematical Society, USA.

Defatta, D. J., Lucas, J. G. and Hodgkiss, W. S., (1988), *Digital Signal Processing - A system Design Approach*, John Wiley & Sons Inc., USA.

Higgins, R. J., (1990), *Digital Signal Processing in VLSI*, Prentice-Hall, Englewood Cliffs, N.J., USA.

Hamming, R. W. (1989), *Digital Filters*, Prentice-Hall, Englewood Cliffs, N.J., USA.

Harris, F.J., (1978), *On The Use of Windows For Harmonic Analysis With the Discrete Fourier Transform*, Proc. Inst. Electrical and Electronic Engineers, January.

Haykin, S. (1991), *Adaptive Filter Theory*, Prentice-Hall, Englewood Cliffs, N.J., USA.

Kay, *Modern Spectral Estimation*, Prentice-Hall Inc., USA

Knuth, D. E., (1981), *Seminumerical Algorithms, vol 2 of The Art of Computer Programming*, Addison-Wesley, USA.

Park S., *Principles of Sigma-Delta Modulation for Analog-to-Digital Converters (APR8.PDF)*, Motorola Inc., USA

Marple, *Digital Spectral Analysis with Applications*, Prentice-Hall Inc., USA

Nussbaumer H., *Fast Fourier Transform and Convolution Algorithms*

Qian, S and Chen, D., (1996), *Joint Time-Frequency Analysis*, Prentice-Hall Inc., USA.

Oppenheim, A. V. and Schafer, R. W., (1989), *Discrete Time Signal Processing*, Prentice-Hall Inc., USA.

Papamichalis, P. (ed.), (1991), *Digital Signal Processing Applications with the TMS320 Family, Volume 3*, Macmillan Inc., USA.

Park, S. (1990), *Principles of Sigma-Delta Modulation for Analog-to-Digital Converters*, Motorola Inc., USA.

Proakis, J. G., and Manolakis, D. G., (1992), *Digital Signal Processing - Principles, Algorithms and Applications (2ed)*, Prentice-Hall, USA.

Proakis, J. G., Rader C. M., Ling, F. and Nikias, C. L., (1992), *Advanced Digital Signal Processing*, Macmillan Inc., USA.

Sorensen, H. V., Jones, D. L., Heideman, M. T. & Burrus, C. S., (1987), and Sig. Proc., Vol ASSP-35, No *Real-valued Fast Fourier Transform Algorithms*, IEEE Tran. Acoustics, Speech 6 pgs 849-863. (1981)

Stimson, G. W., (1998), Introduction To Airborne Radar (2ed), Scitech Publishing, USA.

World Wide Web : http://www.numerix.co.uk/hotlinks.html