Chapter 5

Signal Processing Fundamentals

Signals can be described and modified in different domains such as time and frequency. Some signals are more easily modelled in one domain than in another. For example, impulsive noise due to switching is readily described by a rapidly rising and falling pulse in the time domain. The same noise may be modelled in the frequency domain in terms of its spectral components, but is less easy to do. The choice facing the designer of a line simulator is in which domain is it easiest to model and modify signals associated with DSL transmission? It is not a simple choice, as ‘ease’ encompasses many areas such as accuracy, complexity, versatility and cost to name but a few. To complicate the choice further, different categories of signal may be more suited to one domain rather than another, giving rise to a model described in more than one domain. This chapter will consider the two basic domains of time and frequency and develop signal processing principles for specific modelling implementations.

5.1 The Modelling Requirement

As discussed in previous chapters, a DSL line simulator must model the physical effects of the access line in addition to noise and crosstalk. The end to end description is of a transmitted signal from a DSL modem, x(t), modified by a physical line described by its impulse response, h(t), arriving at a receiving modem with the addition of noise, n(t), and crosstalk, c(t). In the time domain, x(t) is convolved[1] with h(t) and summed with n(t) and c(t). In the frequency domain, x(t)’s Fourier transform, X(f), is multiplied by h(t)’s Fourier transform, H(f), and summed with the noise and crosstalk transforms, N(f) and C(f). Figure 5.1 shows these two alternative descriptions.

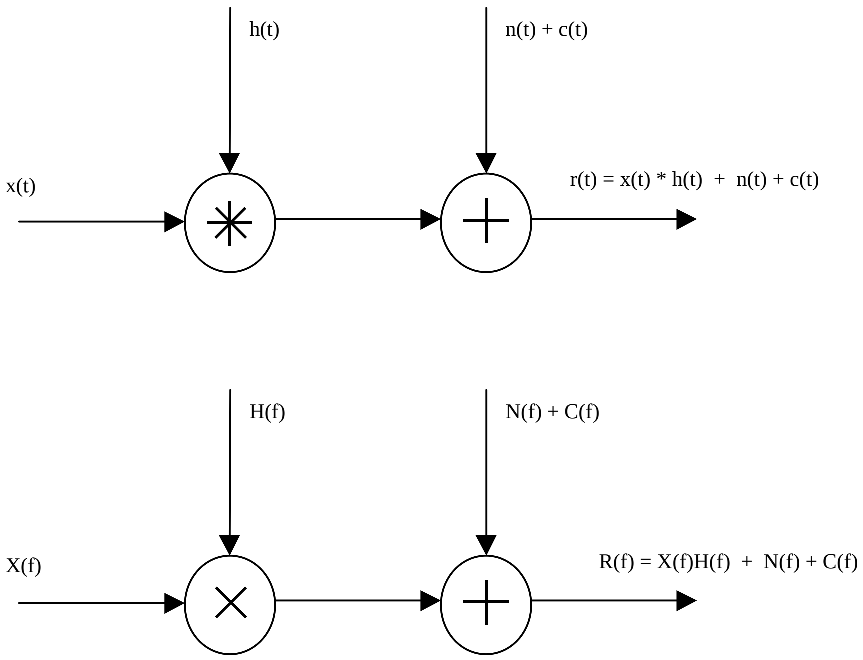

Figure 5.1 Time and frequency domain description of transmission process

5.2 Signal Domains

5.2.1 Physical Line Effects

Shown only the insertion loss of figure 2.2, it would be reasonable to conclude the response is one of a monotonic filter such as a Butterworth design. If this were the case, simulating the response in the time domain using a digital filter would be the obvious choice. However, if one also considers the phase response of figure 2.3, a designer would be challenged to reproduce it with a time domain digital filter. A linear phase response FIR filter could be used to model the insertion loss, but the phase response would be very difficult to recreate with a time domain digital filter.

An alternative to filtering in the time domain, which instead of convolving the signal and impulse response of the filter, is to multiply the signal and filter transfer function in the frequency domain. This is equivalent due the transform property of the Fourier transform; convolution in one domain is the same as multiplication in the other. Filtering in the frequency domain, is actually the preferred method for channel equalisation in practical receivers[2] where a Frequency Domain Equalisation (FEQ) filter multiplies the frequency samples given by a DFT of the incoming signal with the inverse frequency response of the channel it is connected to. Frequency domain filtering has also been employed extensively in the field of medical CT and MRI[3] scanning where processing is non-real time and requires tight control of phase response. The major problem with frequency domain filtering in the telecommunication's arena is the requirement of real time processing. To filter in the frequency domain, a discrete frequency representation of the sampled time domain signal must be computed, then modified for each frequency sample then transformed back to the time domain. If the two transforms can be achieved in real time, frequency domain filtering provides a very powerful tool[4].

Figure 5.2 shows how a single tone's magnitude and phase can be modified using frequency domain multiplication, with an attenuation factor of 'B' and phase shift of 90°.

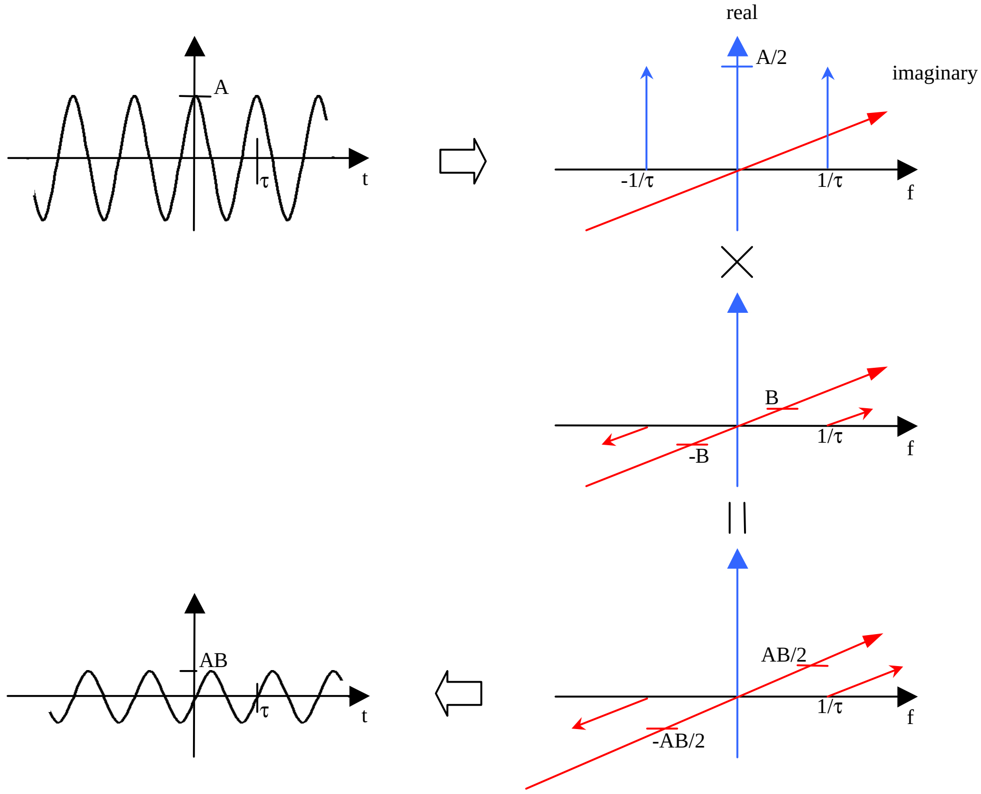

Figure 5.2 Magnitude and controlled 90° phase shift of a single tone

To produce the 90° phase shift, the Fourier transform of the signal is multiplied by a purely complex conjugate pair. The attenuation of the tone is given by the magnitude of the multiplying complex number.

Figure 5.3 shows a monotone attenuated and shifted by an arbitrary phase through multiplication with a complex conjugate pair with non-zero real and imaginary parts. The resultant phase shift of a tone at f_1 due to the filter's transfer function H(f) is given by

$$j_{f_1} = \tan^{-1}\left(\frac{\mathrm{Im}(H(f_1))}{\mathrm{Re}(H(f_1))}\right)$$

The behaviour of a filter across a band of frequencies can be produced by multiplying each frequency component of a signal by different complex conjugate pairs. Clearly the control of the phase response makes frequency filtering an ideal choice to model the phase response of an access line at the extended DSL frequencies.

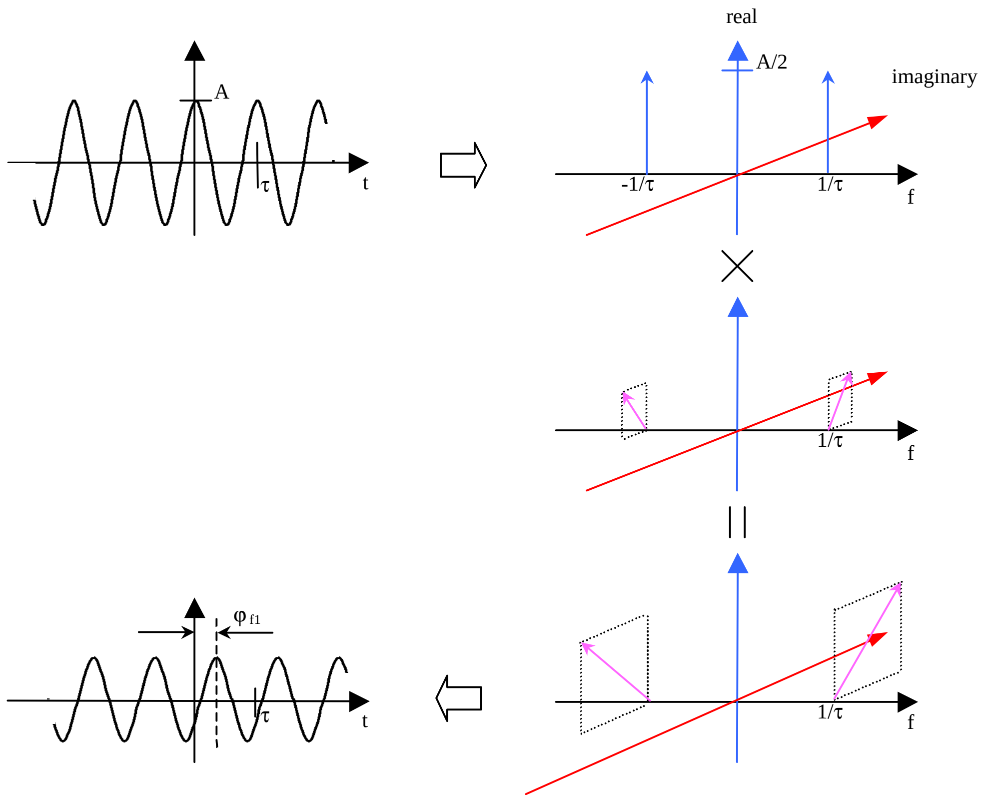

Figure 5.3 Magnitude and controlled arbitrary phase shift of a single tone

5.2.2 Crosstalk

DSL signals are characterised by their PSD which is a frequency domain description. Crosstalk, which is dependent on the PSDs of the disturbing and disturbed signals, can therefore easily be modelled in the frequency domain by the scaled addition of the disturber's PSD to the disturbed signal's PSD. With discrete transforms this corresponds to adding a scaled disturbing signal's DFT to the disturbed signal's DFT.

Self crosstalk to a DSL signal can be simply simulated by the addition of a delayed and attenuated copy of the DFT of the same signal. However, since the simulator will first sample the time waveform, self crosstalk can just as easily be produced in the time domain by the addition of a delayed and attenuated copy of the time sampled signal instead of the DFT of the signal under simulation.

Although both approaches are relatively straight forward, modelling in the frequency domain enables foreign crosstalk to be modelled more easily as knowledge of other DSL signals' PSD functions is generally known, whereas a time domain approach would require generation of line codes. Figure 5.4 shows both frequency and time domain self and foreign crosstalk modelling methods.

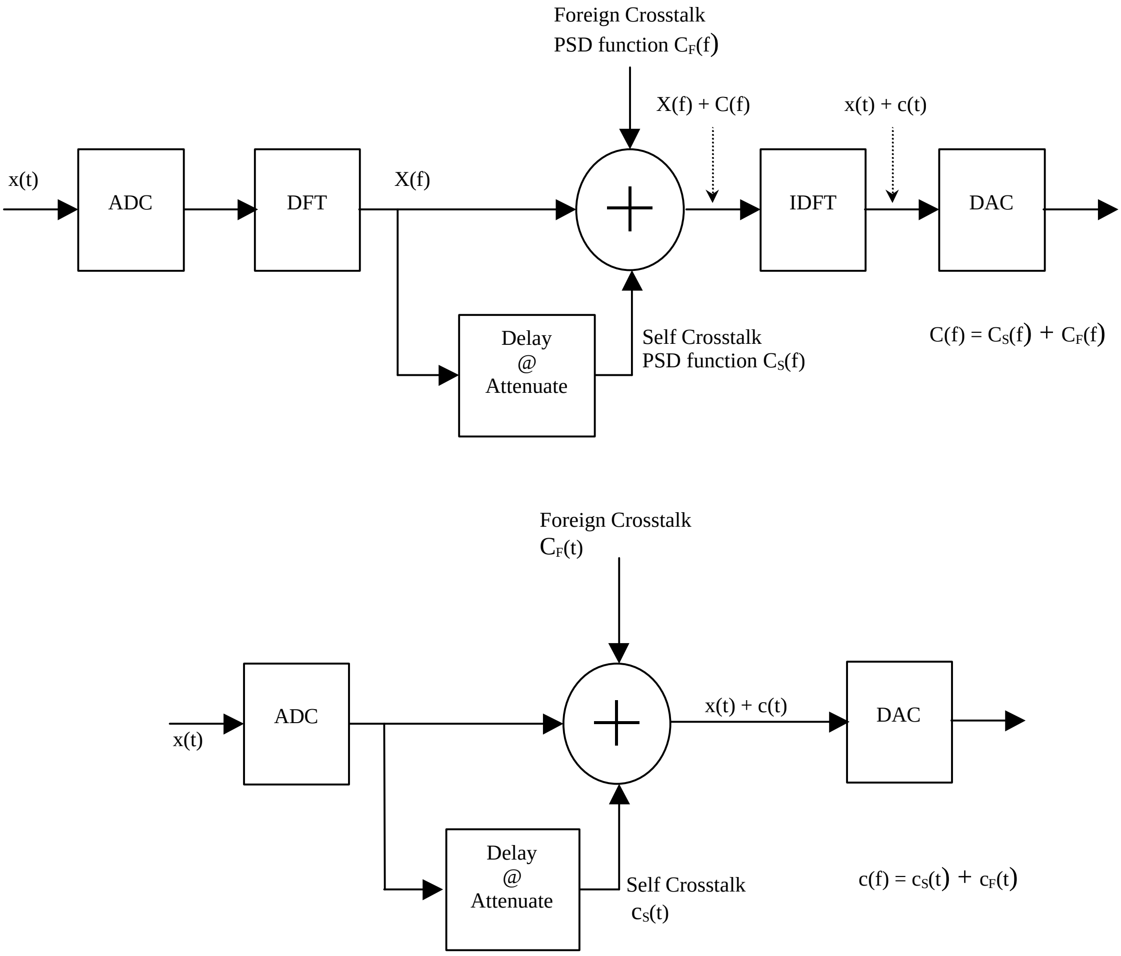

Figure 5.4 Frequency and time domain crosstalk modelling

5.2.3 Noise

5.2.3.1 Additive Gaussian White Noise

AGWN can be modelled in both time and frequency domains. In the frequency domain, AGWN can be generated as random sinusoids with random phase through additions of single complex conjugate points to the DSL signal's DFT.

5.2.3.2 Coloured Noise

By its very definition, coloured noise which is noise occupying a specific spectral band, is described in the frequency domain by its PSD function. Since coloured noise is a set of different frequency sinusoids, it could be modelled in the time domain, but would require additions to all time samples whereas in the frequency domain modelling requires additions to only the few frequency samples its occupies.

Modelled in the time domain[5], coloured or bandpass noise can be viewed as a sum of two sinusoids

$$n(t) = n_1(t)\cos\omega t + n_2(t)\sin\omega t$$

Alternatively it can also be viewed as a single sinusoid with randomly fluctuating amplitude and phase

$$n(t) = r(t)\cos[\omega t + \phi_n(t)]$$

5.2.3.3 Impulsive Noise

Impulsive noise is generally produced by switched currents ranging from semiconductor switching to motor contact arcing. Figure 5.5 shows a typical time waveform of impulse induced noise.

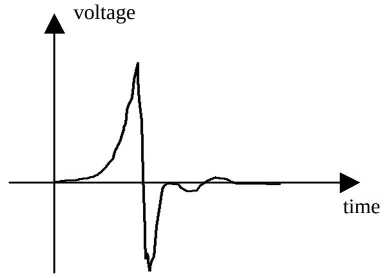

Figure 5.5 Impulsive noise waveform

Whilst repetitive impulsive noise bursts may occupy a fairly well defined spectral mask, random, irregular bursts of impulsive noise are more readily modelled in the time domain. Since both domains are useful in modelling impulsive noise, the combination of both time and frequency techniques shown in figure 5.6 gives maximum flexibility.

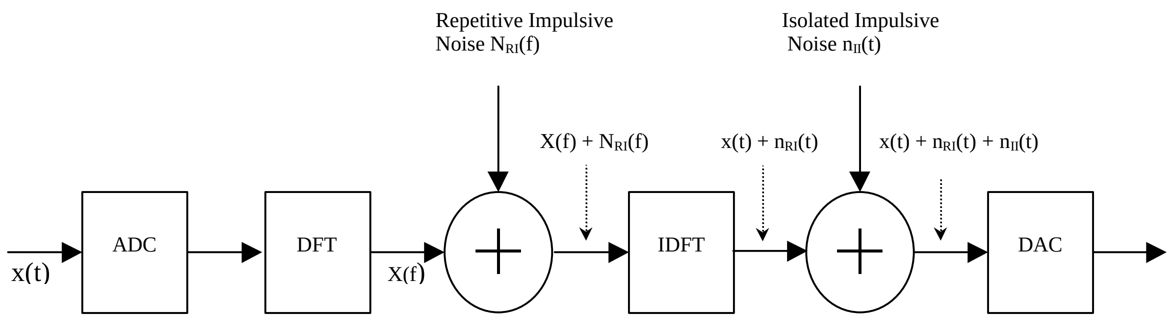

Figure 5.6 Impulsive noise modelling

5.2.4 Overall Simulation Model

Combining the modelling approaches of the previous sections, conceptually the physical line, crosstalk and noise environment can be simulated using the method shown in figure 5.7.

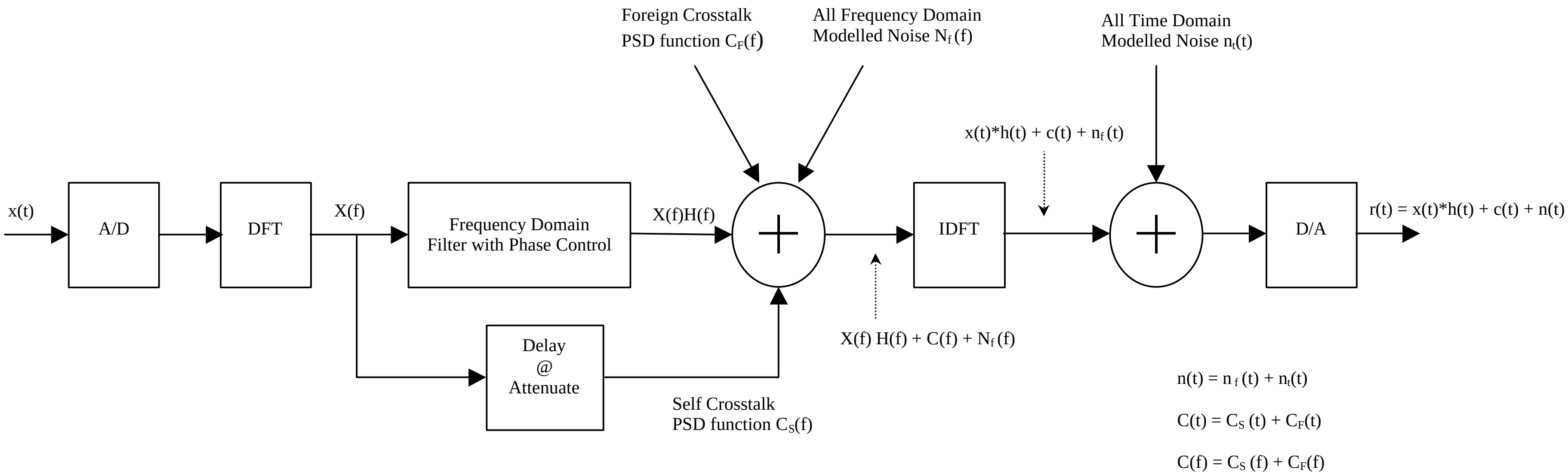

Figure 5.7 Overall simulation method

5.3 Discrete Fourier Transform

The Discrete Fourier Transform[6] (DFT) is the discrete equivalent of a signal's frequency domain description given by the continuous Fourier transform and is defined as

$$X\left(\frac{n}{NT_s}\right) = \sum_{k=0}^{N-1} x(kT_s)e^{-j2\pi nk/N} \qquad n = 0, 1, \ldots, N-1$$

Application of this equation gives frequency components from DC to (N-1)/NT_s Hz in steps of the transform resolution, 1/NT_s . The components from 1/2T_s to (N-1)/NT_s are actually negative frequency components[7]. In addition, if the discrete transform is scaled the same as the continuous version, the DFT can be written

$$X\left(\frac{n}{NT_s}\right) = T_s \sum_{k=0}^{N-1} x(kT_s)e^{-j2\pi nk/N} \qquad n = 0, \pm 1, \pm 1 \ldots, \pm N/2$$

Computed using the basic equation above, the computation can handle any number of time sample points, N, but tends to be computationally intensive and thus slow. Many algorithms have been developed dramatically reducing the computation time at the expense of setting conditions on the input vector's length. Speed optimised algorithms exploit the computational repetition in the full DFT that can occur for constrained data lengths. For example, the original Fast Fourier Transform (FFT) developed by J. Cooley and J. Tukey[8], the radix 2 FFT, requires an input vector of length $N = 2^n$. Multiplications comprise the large majority of the machine code instructions performed to compute a DFT using a digital processor. The radix 2 FFT algorithm's increase in computational speed over the basic DFT can be appreciated by considering the number of multiplications required by the two algorithms for a given time vector, shown in figure 5.8.

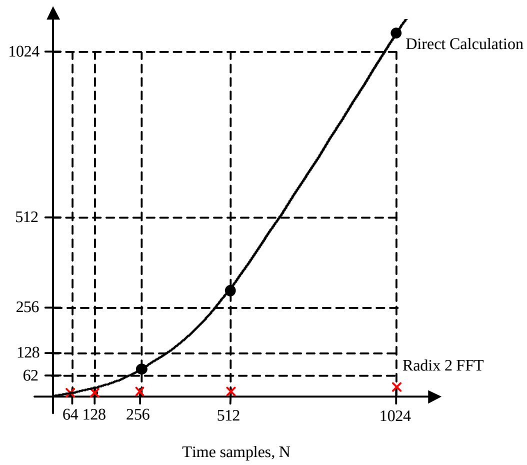

Figure 5.8 DFT and radix 2 FFT multiplication comparison

Other fast algorithms are commonly radix 4, requiring input data to be of length $N = 4^n$. Clearly if a data vector is conditioned for use with a radix 4 FFT it will also be suitable for a radix 2 algorithm.

5.3.1 DMT ADSL Fast Fourier Transform

Recalling the requirements set out in chapter 4 for the simulator's frequency transform, a DFT with 1024 points sampled at 4.416 MSPS has 512 cells each of width 4.31265 kHz and a maximum observable frequency of 2.208 MHz. Over the 1.1 MHz DMT bandwidth this gives 256 frequency samples. A 1024 point transform can be computed using either a 2 or 4 radix transform (2^{10} =1024 and 4^5 = 1024).

5.3.2 CAP ADSL Fast Fourier Transform

Since the DFT described for DMT ADSL has a maximum observable frequency of 2.208 MHz, it is also suitable for a CAP ADSL simulator that has the same resolution requirements and maximum observable frequency of 1.5 MHz. Sampling at just over 2 MSPS gives a 500 kHz guard band to avoid the effects of aliasing which occur when sampling at exactly twice the Nyquist limit is employed due to the physical impossibility of realising a brick wall filter necessary to finally recover the signal.

5.3.3 VDSL Fast Fourier Transform

A full VDSL simulator must have a minimum observable frequency of at least 20 MHz. The Nyquist sampling limit is therefore 40MSPS, although with the addition of a 5 MHz guard band, sampling at 50 MSPS is the practical minimum.

From chapter 4, section 5, short lines of just 1 kft (300 m), required 741 cells across the full rate 20 MHz bandwidth, which was constrained by the phase variation limit of 10%, not insertion loss variation across the first relevant frequency interval. Since there is a 5 MHz guard band, the transform must have a minimum of 926 cells across the 25 MHz bandwidth (741 x 25/20). This corresponds to a 2048 point DFT for a radix 2 FFT (24.4 kHz resolution) or 4096 points if a radix 4 algorithm is employed to compute the discrete transform (12.2 kHz resolution).

5.4 ADC and DAC Conversion

Before any DFT can be performed, the analogue signal from a DSL modem must be sampled in the time domain. Conversely, after taking the Inverse DFT (IDFT) of the modified signal, digital to analogue conversion must be performed. Practically all conversion must be to a finite quantisation interval described by a limited number of bits to be processed by any digital circuitry.

For a simulator, the ADC and DAC resolution must be at least as good as that used in the DSL modems themselves or else the simulator's sampling will distort the signal far more than the sampling within the modems. The sampling precision required in modems must be fine enough so as not to limit the channel capacity further due to its presence. Practical fast ADC and DAC conversion is achieved using flash converters which due to their architecture have a limited number of quantisation bits and hence precision.

5.4.1 Theoretical ADC and DAC Precision

Extensive work by Dr. Walter Chen[9,10] on analogue front end precision requirements in DSL modems has been conducted on which the sampling precision for the simulator has been based. The author identifies three fundamental factors determining the required DAC resolution for passband QAM based transmitters

- The constellation size
- The desired signal-to-noise ratio or error rate
- The peak to average voltage ratio of digital shaping filters

The author draws particular attenuation to the effect of the digital filtering and the maximum transmission error rate on the required conversion precision. For example, a passband transmitter with a constellation of 16 points requires 4-bit quantisation ($2^4 = 16$) without digital filtering, which increases to 6 bits with filtering and a BER limit of between 10^{-6} and 10^{-8}.

For ADSL with a maximum constellation size of 32 768 points the minimum DAC resolution is 14 bits, but practically should be 15 to 16 bits to minimise the effect of quantisation noise compared with other transmission impairments[11]. The same author proposes a minimum 16-bit ADC resolution for lines upto 16 kft (3 miles) long.

5.4.2 Practical ADC and DAC Precision

As previously mentioned, a line simulator’s sampling resolution must be at least as good as that used in the modems it will be connected to. Modem manufacturers however are very unlikely to divulge what is inside their equipment, but an indication of the conversion precision employed can be gained from considering recently released DSL AFE chip-sets that perform filtering and signal conversion. One such device, the ADSL KeyWave™ AFE manufactured by Fujitsu[12], is designed for both full and G.Lite applications and contains 15-bit ADC and DAC conversion blocks, sampling at some factor of the 17.664 MHz reference clock, probably 8.832 MSPS (17.664/2) or 4.416 MSPS (17.664/4). The exact reference clock division to the converters can only be predicted from the advance data sheet as the device isn’t finalised yet, but will almost certainly be some factor of two.

References

[1] Walter Chen, “DSL”, Macmillan, 1998, p.107.

[2] Denis J. Rauschmayer, “ADSL / VDSL Principles”, Macmillan, 1999, p208, figure 7.2.

[3] O. Helenon, M. Laval-Jeantet, J. Frija , “Artifacts on lung CT scans: removal with Fourier filtration”, Radiology, 171(2), May 1989, pp.572-4.

[4] Edward Cunningham, “Digital Filtering, An Introduction”, Wiley, 1995, p346.

[5] John O’Reilly, Communication Systems Modelling Lecture Course, University College London, November 1998.

[6] E. Oran Brigham, “The Fast Fourier Transform and Its Applications”, Prentice Hall, 1988, p97.

[7] E. Oran Brigham, “The Fast Fourier Transform and Its Applications”, Prentice Hall, 1988, p169.

[8] J. Cooley and J Tukey, “An Algorithm for Machine Calculation of Complex Fourier Series”, Math. Computation, April 1965, Vol. 19, pp.297-301.

[9] W. Chen, “A Calculation of the Required A/D Precision for ADSL”, Bellcore T1E1 Contribution, T1E1.4/92-082, May 1992.

[10] Walter Chen, “DSL”, Macmillan, 1998, Chapter 7 (Analog Front-End Precision).

[11] N. Al-Dhahir and J. Coiffi, “On the Uniform ADC Bit Precision and Chip Level Computation for a Gaussian Signal”, IEEE Trans. On Signal Processing, vol. 44, no. 2 , February 1998, pp.434-438.

[12] Product Preview, KeyWave™ AFE, Fujitsu Microelectronics UK limited, December 1998.