SIGNAL CONDITIONING FOR DIGITAL CONTROL

By David A. Haessig & Matthew Domier

DIGITAL SIGNAL PROCESSING IN CONTROL

Digital signal processing (DSP) is the manipulation of signal data contained in the form of discrete samples. If the samples represent an observation from a real system, they are created by the conversion of a continuous-time, analog signal to the discrete-time digital domain. If this is a control application, the digital samples produced by the controller are typically converted back to an analog, continuous-time signal driving an actuator that is part of the plant. Understanding these conversion processes requires an understanding of the mathematical processes of sampling and aliasing, as well as that of continuous signal reconstruction from samples.

This chapter begins with an overview of sampling theory, describing the effect of signal aliasing on the continuous to discrete conversion process, and defining the Nyquist Theorem and Nyquist frequency. A summary and brief overview of the various transforms (Fourier, *z*, etc.) involved in these developments is provided in an appendix at the end of this chapter, along with the application of the Discrete and Fast Fourier Transforms (DFT and FFT) in digital signal analysis. The construction of ideal and realizable anti-alias filters is described with the help of several examples. The carrying of signal information as discrete samples and their conversion back to a continuous-time form without distortion via ideal filters, and with acceptable distortion with realizable filters is described.

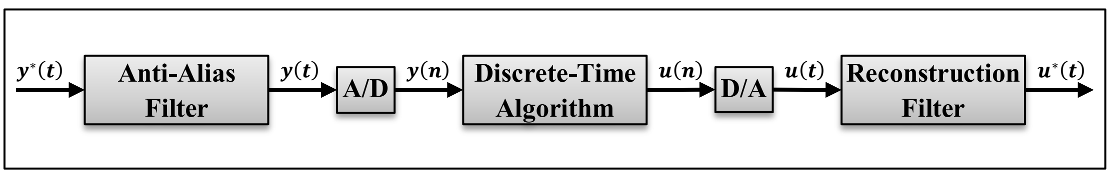


Figure 3–1: Conversion from a continuous input signal $y^*(t)$ to discrete samples $y(n)$ and $u(n)$, and back to a continuous control output signal $u^*(t)$

MATHEMATICAL MODEL OF DISCRETE-TIME SAMPLING

This section describes the effect of sampling on the incoming signal $y(t)$. This information will enable definition of requirements on the anti-alias filter producing $y(t)$ from $y^*(t)$. Another objective of this section is the definition of the spectrum at the output of the D/A converter, which will enable the specification of requirements on the reconstruction filter that follows.

In Figure 3-2, mathematical models of the A/D and D/A conversion processes are proposed. The A/D is represented as a continuous time ideal sampling process and continuous-to-discrete (C/D) conversion. The D/A is discrete to continuous conversion (D/C) followed by a hold function. C/D conversion is the generation of a numerical value from the continuous signal captured at the instant of sampling. D/C conversion is the opposite, converting from a numerical value to a signal level, an impulse size, that drives the system having an impulse response that holds that level for *T* units of time. The concept of the impulse, a mathematical abstraction that does not exist in the physical world, is used in generating both the mathematical description of the input and output spectral, with $y_\delta(t)$ and $u_\delta(t)$ being streams of delta functions in time, spaced at the sample time *T*.

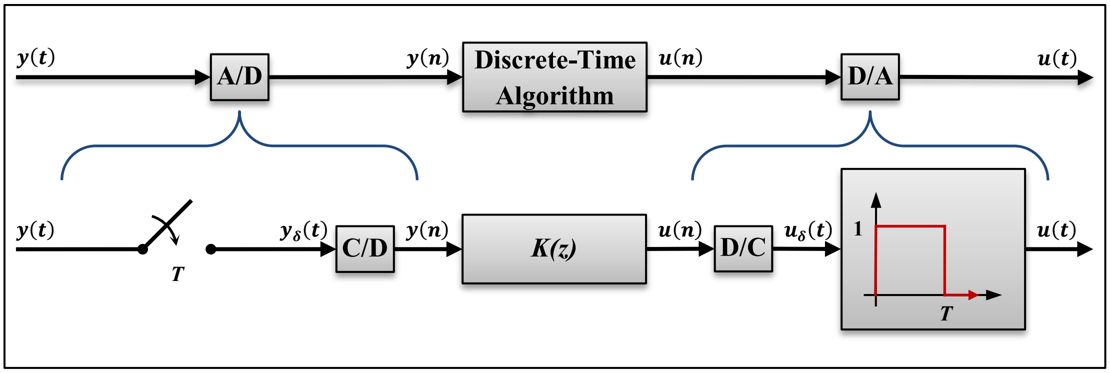


Figure 3–2: Mathematical modeling of sampling and discrete to continuous conversion processes

SAMPLING THEOREM AND ALIASING

This development of the Shannon-Nyquist Sampling Theorem and aliasing is built upon a property of the Fourier transform when applied to the convolution operation – convolution in the time domain gives rise to the multiplication of their transforms in the frequency domain. See Appendix 3A below for a description of this. The dual is also true – that multiplication of two signals in the time domain produces a convolution of their Fourier transforms – or to be more succinct, "Convolution in the time domain results in multiplication in the frequency domain", and visa-versa, "Multiplication in the time domain results in convolution in the frequency domain".

We begin with the development of a mathematical model suitable for describing the fundamental operation of an analog to digital converter, with the ultimate goal of analyzing the effect of an A/D on a signal.

The effect of an ideal sampler on an analog signal can be described as a momentary switch that is being periodically operated at the sampling rate of the A/D clock…

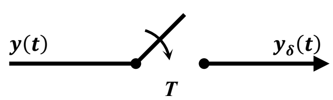


Figure 3–3: An ideal sampler modeled as a momentary switch

By decreasing the time the switch is held closed, until it is infinitesimally small, the action of the switch's pulses can be modeled as the product of $y(t)$ with a "comb" process—an infinite series of Dirac delta functions, $\delta(t - nT)$…

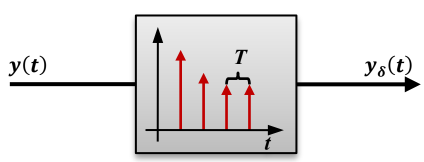


Figure 3–4: An ideal sampler modeled as the product of $y(t)$ and an infinite series of Dirace delta functions

$$y_\delta(t) = y(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{3–1}$$

$$y_\delta(t) = \sum_{n=-\infty}^{\infty} y(t)\delta(t - nT)$$

$$y_\delta(t) = \sum_{n=-\infty}^{\infty} y(nT)\delta(t - nT)$$

$$y_\delta(t) = \sum_{n=-\infty}^{\infty} y(n)\delta(t - nT) \tag{3–2}$$

Thus, $y(n)$ in (3–2) represents the weights of the digital sample weights carried on delta function $\delta(t - nT)$.

Consider the ideal sampler alone:

$$\delta_T(t) = \sum_{n=-\infty}^{\infty} y(t)\delta(t - nT) \tag{3–3}$$

We know from Fourier transform theory that,

$$\mathcal{F}\{\delta_T(t)\} = \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T}\right) \tag{3–4}$$

Remember, $2\pi f = \omega$, therefore $X(\omega/2\pi) = X(f)$. We use $X(f)$ here because it has units of $1/time$.

Now consider the ptoduct of $y(t)$ and $\delta_T(t)$. Multiplication in the time domain results in convolution in the frequency domain, therefore,

$$Y_\delta(f) = Y(f) * \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T}\right) \tag{3–5}$$

Recall the "sifting" property of $\delta(f - x)$ when convoluted with $Y(f)$,

$$Y(f) * \delta(f - x) = Y(f - x) \tag{3–6}$$

Thus,

$$Y_\delta(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} Y\left(f - \frac{n}{T}\right) \tag{3–7}$$

If the generating Fourier Transform $Y(f)$ is bandlimited, meaning that it is zero outside a specified bandwidth B, then if the spacing of the aliased replicas of $Y(f)$ exceeds $2B$, there will be no overlap in these replicas and the summation at any f will only involve one of the generating functions at most. This can be guaranteed by making the sample frequency $1/T$ at least twice the bandwidth B.

The Shannon – Nyquist sampling theorem – According to Wikipedia, Shannon's version of the sampling theorem is as follows:

> If a function $x(t)$ contains no frequencies higher than B hertz, it is completely determined by giving its ordinates at a series of points spaced $1/(2B)$ seconds apart.

Or, in other words…

> A bandlimited analog signal that has been sampled can be perfectly reconstructed if the sampling rate exceeds $2B$ samples per second, where B is the highest frequency of the original signal, and where an infinite number of samples are provided.

Note the word exceeds. When, for example, the sample rate precisely equals twice the frequency of a pure sin wave, then an infinite length sampling process will not provide the information defining the amplitude and phase of the sine wave. This is clear when examining Figure 3–5, which shows that sampling precisely at two times the sample frequency is insufficient.

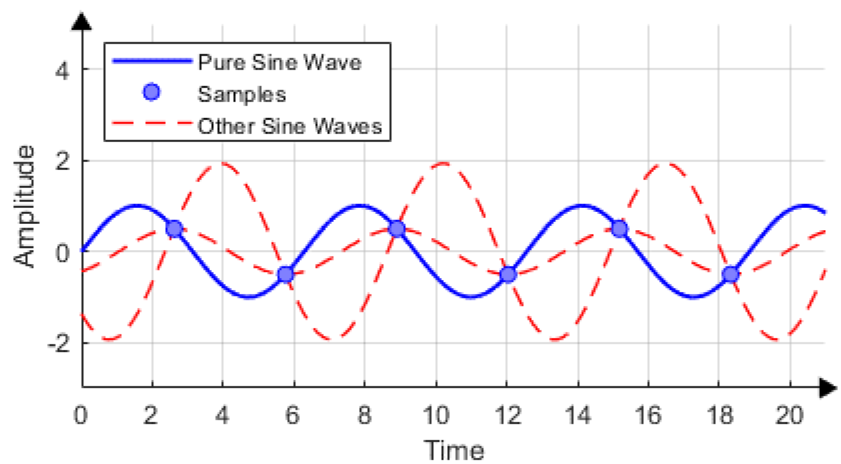


Figure 3–5: Sampling of a bandlimited pure sine wave at exactly twice the signal's frequency

Sampling at twice the signal's frequency does not satisfy the Shannon-Nyquist theorem, as we see here. A number of different sine waves all pass through these samples. The sample rate must exceed the highest frequency in the analog signal, and the sample record must be infinite in duration.

Nyquist frequency and time interval -- the Nyquist frequency is defined to be twice the sampled signal's bandwidth, or $2B$, and outside the frequency range of $\pm B$ the signal's Fourier Transform is zero. The Nyquist time interval is $1/(2B)$.

ANTI-ALIAS FILTERING

The anti-alias filter is applied upstream of the sampling and digital conversion process, as shown in Figure 3–1. Generally speaking, this filter should pass the signal representative of the physical variable being sensed, and it should reject (i.e. attenuate) any signals out beyond $\pm f_s/2$ to avoid aliasing. This is depicted in Figure 3–6 below.

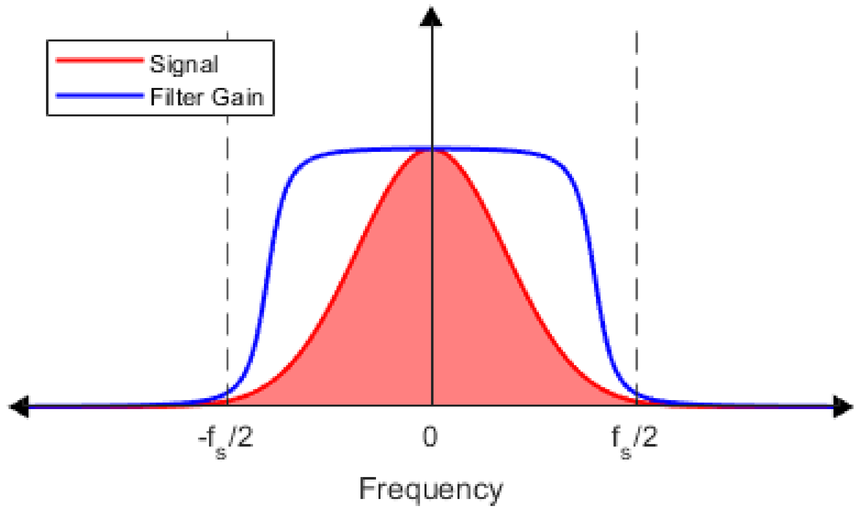


Figure 3–6: An anti-alias filter passing the sensed signal and rejecting frequencies beyond $\pm f_s/2$, one half the sample frequency

The ideal anti-alias filter provides a frequency response that ideally is flat over the bandwidth of the sensed signal and introduces no phase shift, i.e. the phase response is 0 and therefore the signal delay through the filter is 0. This filter also perfectly rejects signal energy out beyond the Nyquist frequency f_s divided by 2, this being the signal that will be aliased back into the range $-f_s/2$ to $+f_s/2$. The ideal is the brick wall filter that is perfectly flat out to $\pm f_s/2$, and exactly zero for all frequencies outside $\pm f_s/2$.

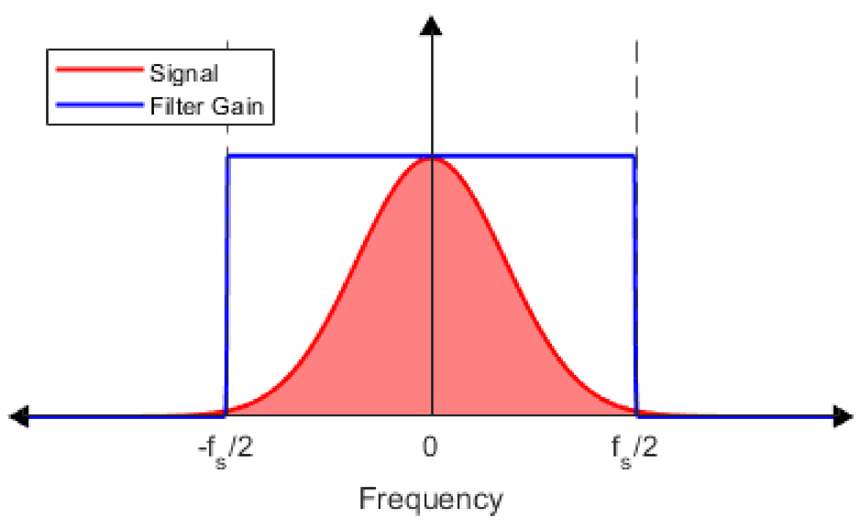


Figure 3–7: An ideal anti-alias filter as a brickwall filter spanning the range $-f_s/2$ to $+f_s/2$

The ideal anti-alias filter is not realizable for two primary reasons—consider the impulse response of the brick wall filter:

- It is a sync function, infinite in duration and involving an infinite number of filter taps (poles), making it impossible to build
- If it is to cause no delay it is anti-causal, which is also not possible

A realizable anti-alias filter is typically a low pass filter constructed of discrete components selected to provide the frequency characteristics desired. It is a causal system and is therefore realizable. What would this look like on a log-log plot of magnitude and semilog plot of phase – see Figure 3–8:

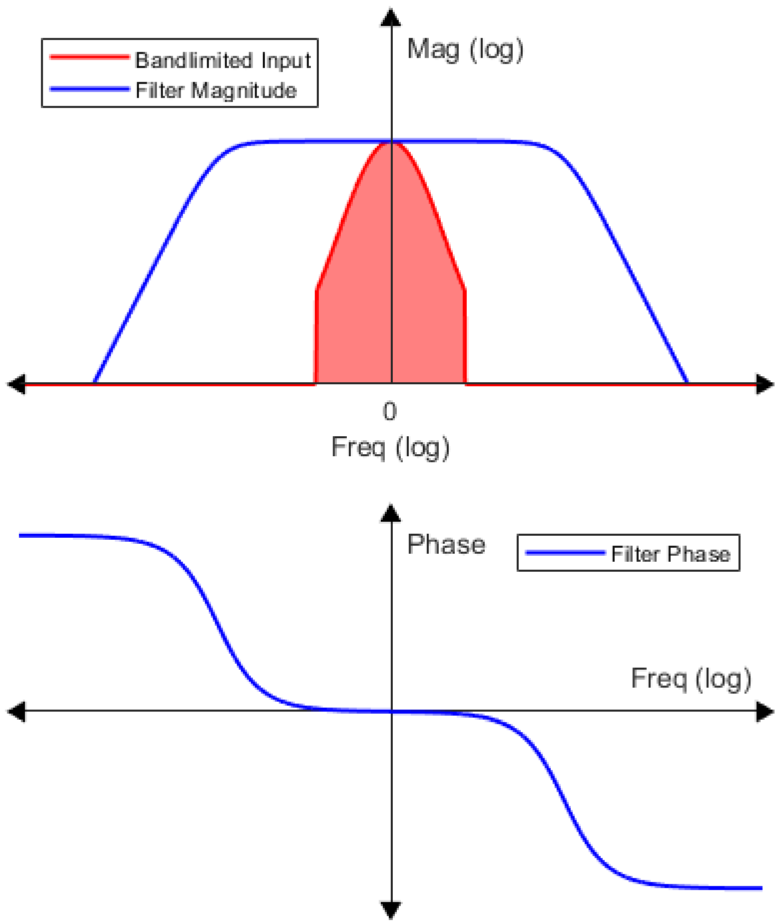
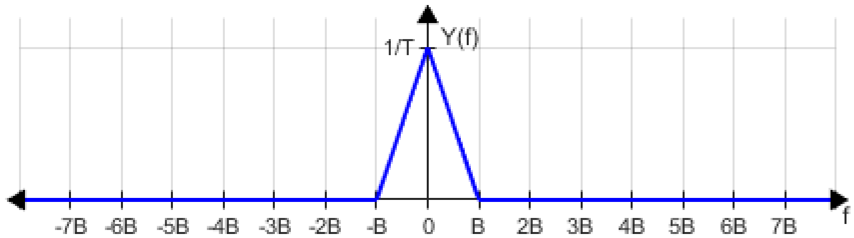


Figure 3–8: Example of a realizable anti-alias filter and "bandlimited" input signal

A low pass filter is represented in the frequency domain (i.e. bode plot), along with the spectral plot of the signal magnitude. The low pass is nominally flat over the signal bandwidth, and the phase is about zero. With any real filter there will be phase loss, or negative phase angle in the region near and outside the passband where the magnitude rolls off. Note that these plots are shown double sided with negative frequencies. This can be somewhat confusing. For real signals which is what we are dealing with, only positive frequencies apply, and we generally only deal with the positive range of frequencies when designing control systems. The two-sided spectra and response apply when the signals of interest are complex; this is typically the case with communication systems after conversion from a real signal to a baseband complex signal (having a spectrum centered on zero).

Alias Example 1: No aliasing

In this example we are given a signal $y(t)$ having a Fourier transform that is zero outside the band $\pm B$ and having the magnitude shown below.



Note that this shape is symmetric about the origin. A property of the Fourier transform of real signals -- the transform magnitude is symmetric about 0. The signal is multiplied in the time domain by the comb process,

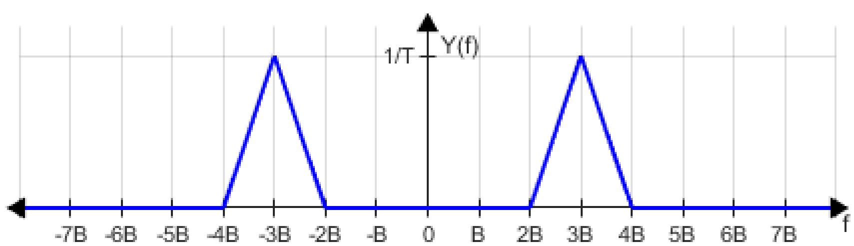
$$\delta_T(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT) \tag{3–8}$$

resulting in

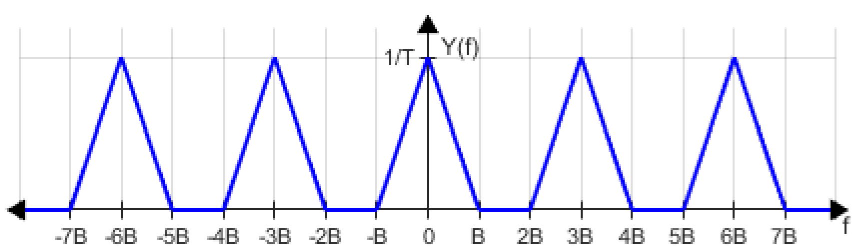
$$y_\delta(t) = y(t)\delta_T(t) \tag{3–9}$$

The sampling frequency equals the bandwidth B times 3; i.e. $T = 1/(3B)$. Plot the Fourier transform of the continuous time signal $y_\delta(t)$.

Solution – We know that the sampling process results in a spreading of the original signals spectrum, with identical replicas of that spectra appearing at integer multiples of the sampling frequency. First let's plot the original spectra shifted by $\pm 3B$.



Clearly when this is added to the original there is no overlap, i.e. no aliasing. Subsequent terms in the series are at $\pm 6B$, $\pm 9B$, $\pm 12B$, and so on.

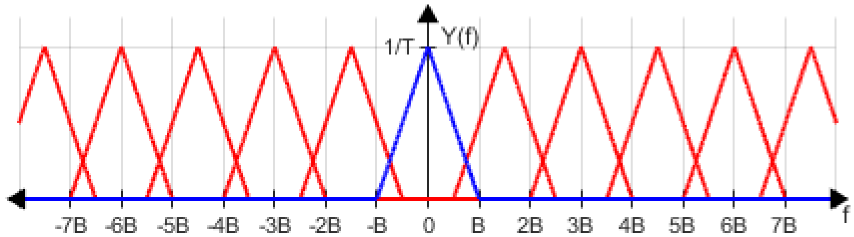


The amplitude of the spectrum is $1/T$ that of the original transform's amplitude.

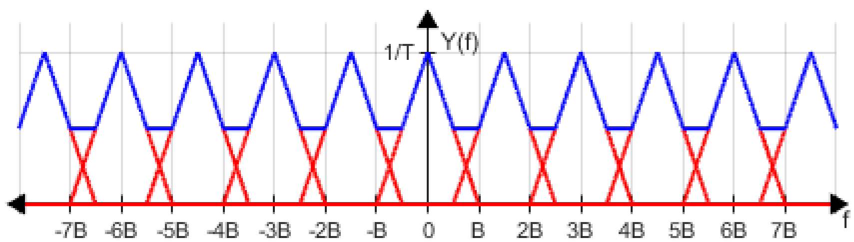
Alias Example 2: Sampling at Less Than the Nyquist Frequency

We are given the same problem as that directly above except that the sample frequency is reduced to $1.5B$. Plot the Fourier transform of the continuous time signal $y_\delta(t)$.

Solution – Begin by sketching the original signals spectrum (scaled by 1/T) in blue. To that we add the signal shifted by $\pm 1.5B$, $\pm 3B$, $\pm 4.5B$, and so on in red.



There are regions of overlap; the region between $B/2$ and B, for example. In those regions, the original spectrum and any overlapping spectral replicas will be summed together, as shown in equation (3–7) above. The resulting magnitude of the superimposed transform components will be:



In the overlapping region the magnitude plots shape will depend on the generating function thus it is plotted with a squiggle to depict that might be occurring in the overlapping region where aliasing is causing distortion.

Alias Example 3: Sinusoids at 100 and 1200 Hz, T = 1/2000 sec

We are given a time domain signal given by:

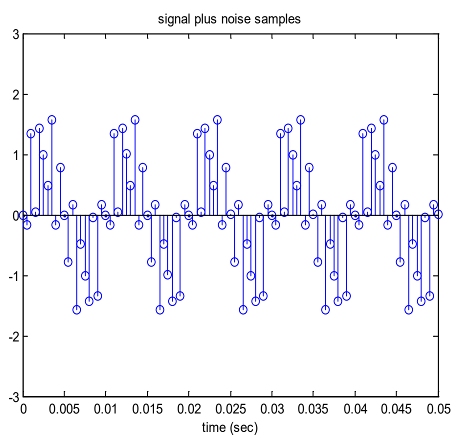
$$x(t) = \sin(2\pi 100t) + \sin(2\pi 1200t) \tag{3–10}$$

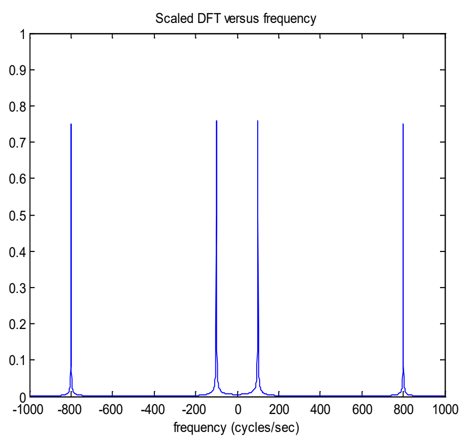
which is sampled at 2000 Hz. Process 2048 samples using an FFT and plot the spectrum from $-f_s/2$ to $+f_s/2$. Decide if there was aliasing. Where does the signal at 1200 Hz appear?

Solution – Generate an .m file to produce the samples and generate the magnitude of the FFT:

```
N = 2048;  % number of samples
T = 1/2000; % sample frequency = 2000 Hz
df = (2*pi/N/T) / (2*pi);
fo = 100;
f1 = 1200;

% create samples from time domain signal
t = (0:1:(N-1))*T;
x = sin(2*pi*fo*t) + 0.8*sin(2*pi*f1*t);

% Plot the signal and noise time domain data
figure(1)
stem(t,x)
```

```
xlabel('time (sec)')
title('signal plus noise samples')
axis([0 0.05 -3 3])

% Compute the FFT and find the magnitude
X = fft(x);
absX = abs(X);

% Plot with FFT zero frequency appearing at the center of the
spectrum
absXs = fftshift(absX);
fs = fftshift(f);
figure(4)
plot(f-(1/2/T),absXs*2/N)
xlabel('frequency (cycles/sec)')
title('Scaled DFT versus frequency')
axis([-1000 1000 0 1])
```



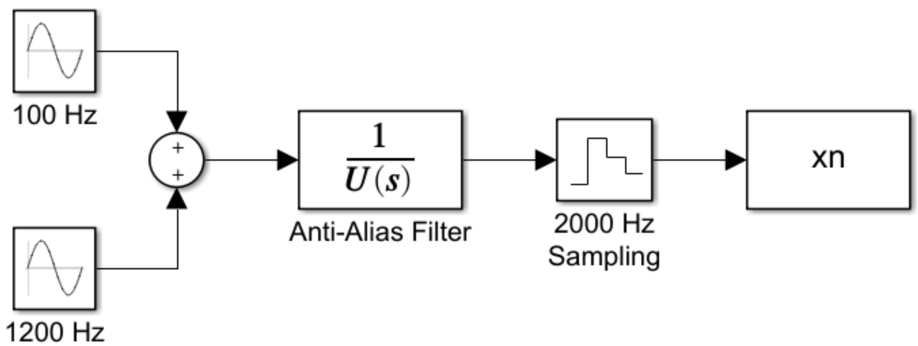


The signal components appear at 100 and 800 Hz. The unsampled signal at 1200 Hz produces an aliased signal at 800 Hz.

Alias Example 4: Sinusoids at 100 and 1200 Hz, T = 1/2000 sec, with a 200Hz anti-alias filter

To the example given above add an anti-alias filter to the time-domain signal that consists of a series of 2 first-order filters having break frequencies of 200 Hz. Repeat the generation of the sample domain plot and FFT.

Solution – Generate a Simulink model representing the system. We use Simulink because of the ease with which we can add the anti-alias filter. That system is:

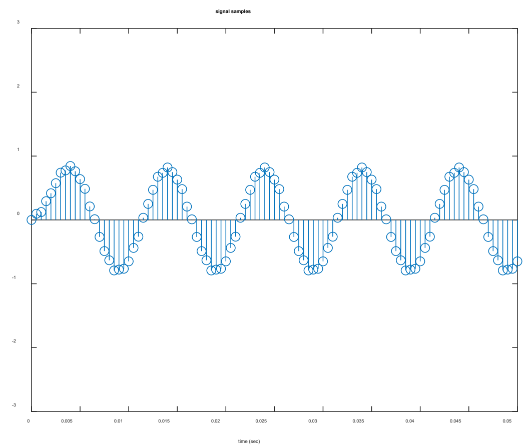


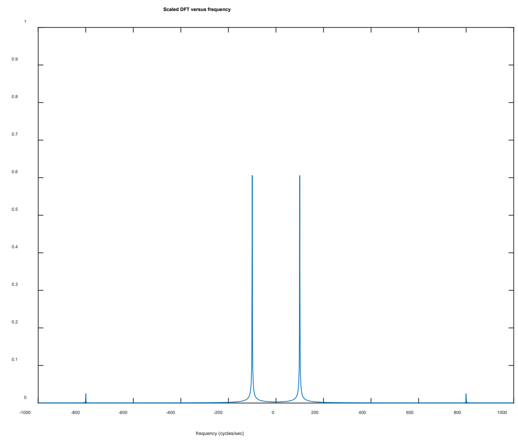
Where,

$$U(s) = \left(\frac{s}{2\pi 200} + 1\right)\left(\frac{s}{2\pi 200} + 1\right) \tag{3–11}$$

The transfer function adds the anti-alias filtering. The zero-order hold performs the sampling and generation of the sample stream $x(n)$. A total of 2048 samples are stored in the workspace array *xn*.

```
% Plot the signal time domain data
figure(1)
Ts = 1/2000;
t = 0:Ts:(length(xn)-1)*Ts;
stem(t,xn)
xlabel('time (sec)')
title('signal samples')
axis([0 0.05 -3 3])

% Compute the FFT and find the magnitude
X = fft(xn(1:2048));
absX = abs(X);

% Plot with FFT zero frequency at the center of the spectrum
absXs = fftshift(absX);
df = 2000/2048;
f = (0:2047)*df;
absXs = fftshift(absX);
figure(2), plot(f-(1/2/Ts),absXs*2/N)
xlabel('frequency (cycles/sec)')
title('Scaled DFT versus frequency')
axis([-1000 1000 0 1])
```



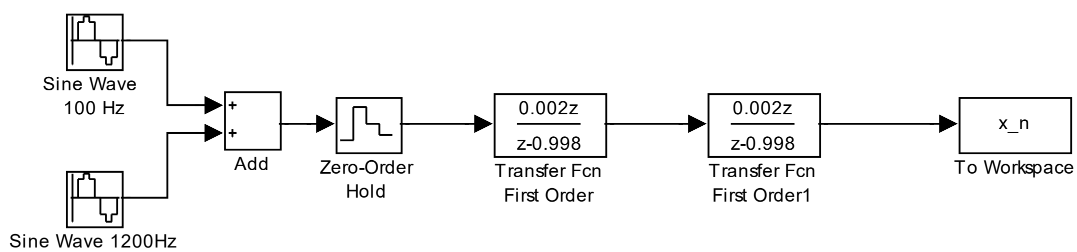


The 100 Hz signal is clearly present, and the FFT shows that the signal amplitude of the 1200 Hz component aliased to 800 Hz is low.

Alias Example 5: Alias Example 3 with 1/T increased from 2 to 100 kHz and digital filtering

For the same input conditions as that of Alias Example 3, the sample frequency is increased to 100 kHz. An analog anti-alias filter is not needed. Process these digital samples with a low pass 2^{nd} order digital filter consisting of two 1^{st} order filters in cascade. Plot these samples over 50 msec.

Solution – A Simulink model sampling the 100 and 1200 Hz signals at 100 kHz is constructed. Two first order digital filters with poles at 200 Hz are added.

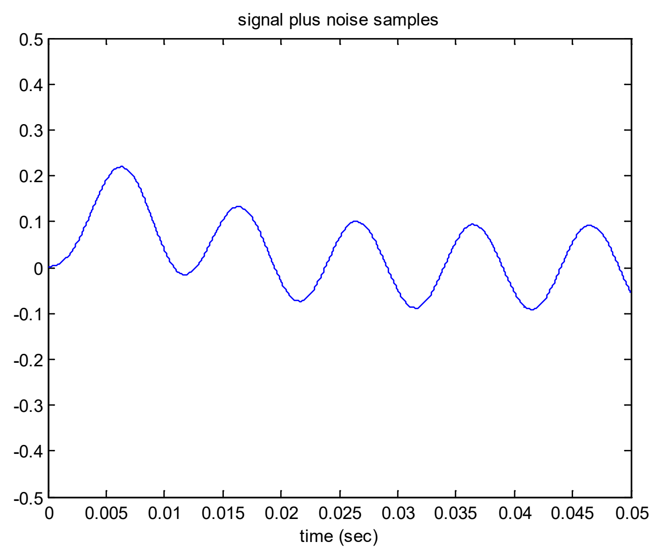


The following code generates the plot:

```
T = 1/100000; % sample frequency = 100 kHz
N = 0.050/T

% create samples from time domain signal
t = (0:1:(N-1))*T;
x = x_n(1:N);

% add noise

%x = x + 3*randn(size(t));

% Plot the signal and noise time domain data
```

```
figure(1)
%stem(t,x)
plot(t,x)
xlabel('time (sec)')
title('signal samples')
axis([0 0.05 -0.5 0.5])
```

From this we see that the 200 Hz signal is clearly present and the 1200 Hz signal suppressed. Thus the need for anti-alias filtering is diminished with faster ADC sampling and the addition of digital sampling to remove unwanted high frequency interference.

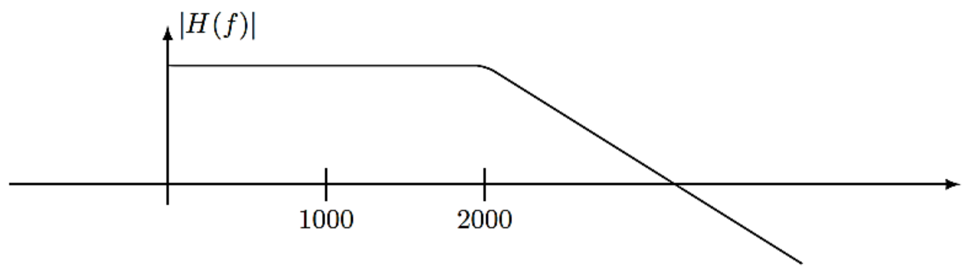


Alias Example 6: How many stages?

An analog sensor provides an output consisting of the true signal having a bandwidth of 1000 Hz plus a continuous-wave (CW) interferer at 20 kHz with an amplitude of 1 volt. An analog anti-alias filter is to be constructed by cascading a series of 1st-order low pass filters each having a break frequency of 2 kHz, and together creating a filter with unity gain in the pass band. How many of these cascaded filters are needed to knock the 20khz signal down in amplitude to 0.0005 volts?

Solution – The problem is depicted in the sketch below showing the Fourier transform of the true signal $Y(f)$, and of the sinusoidal interferer at 20 kHz. We only sketch these spectra for positive frequencies since the spectrum is symmetric about 0.

Each first-order filter is approximately flat with unity gain out to 2000 Hz, and from 2000 to 20,000 each will roll off by a factor of 10x (this is reviewed in Chapter 5 which covers frequency response concepts). Thus, the first by itself attenuates the interferer by 10x to 0.1 volt.



The second provides another 10x of attenuation, bringing it to 0.01 volts, a third to 0.001 volt, and finally a fourth bringing it to 0.0001 volts. Thus, a 4-stage filter is required.

SIGNAL RECONSTRUCTION FROM DIGITAL SAMPLES

The signal reconstruction process is depicted in Figure 3–9, with the conversion of discrete to continuous samples followed by an impulse response that holds the signal constant for duration *T*.

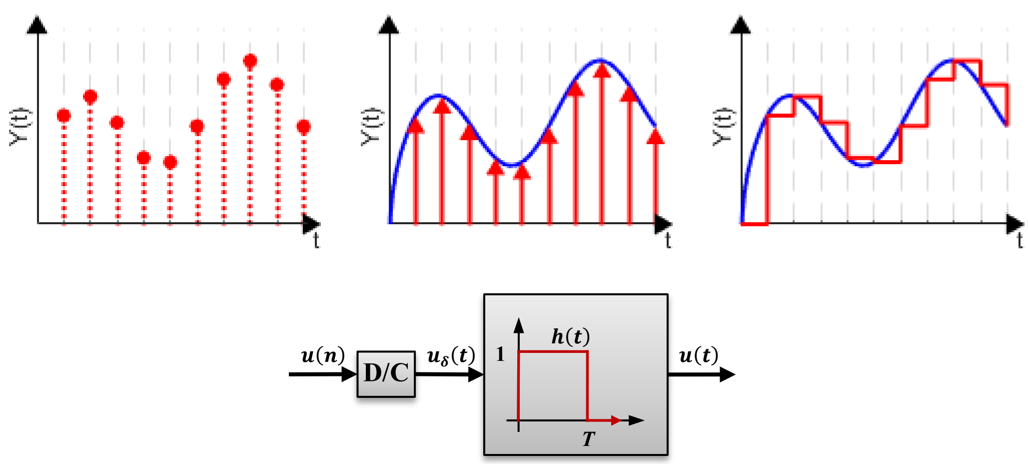


Figure 3–9: Analog signal reconstruction

The discrete-time samples *u(n)*, weight the Dirac-Delta stream $\delta_T(t)$ to produce

$$u_\delta(t) = \sum_{n=-\infty}^{\infty} u(n)\delta(t - nT) \tag{3–12}$$

The Fourier transform of this signal is like that derived above for the same model that arose from the sampling theorem description. Rearranging this to be in the form of a multiplication of a time domain signal:

$$u_\delta(t) = \sum_{n=-\infty}^{\infty} u(nT)\delta(t-nT) \tag{3–13}$$

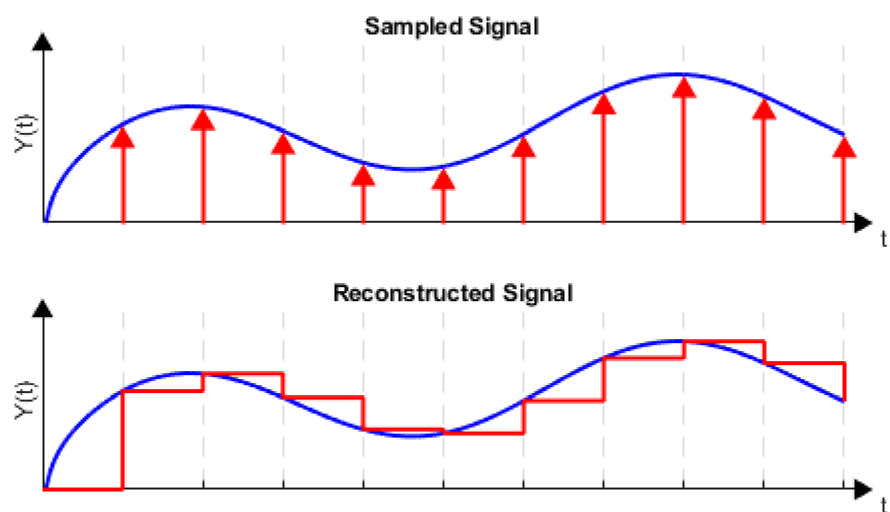
$$u_\delta(t) = u(t)\sum_{n=-\infty}^{\infty} \delta(t-nT) \tag{3–14}$$

which is a multiplication in the time domain, leading to convolution in the frequency domain, simplifying to:

$$U_\delta(f) = \sum_{n=-\infty}^{\infty} U\left(f-\frac{n}{T}\right) \tag{3–15}$$

So this function, existing only as a mathematical abstraction, is the Fourier transform of $u(t)$ repeated at integer multiples of the sample frequency $1/T$.

Passing this through the hold function $h(t)$ is given by the convolution of $u_\delta(t)$ with $h(t)$, which is equivalent to the multiplication of their respective Fourier transforms. Since $h(t)$ is a rectangular function with a duration of T, its transform $H(f)$ is a sync function with zero crossings at $\pm n/T$. The multiplication of $H(f)$, which diminishes with frequency, with the response $U_\delta(f)$, which repeats with period $1/T$ for an infinite duration, results in an output signal spectrum that decays with increasing frequency. This is illustrated in Figure 3–10.



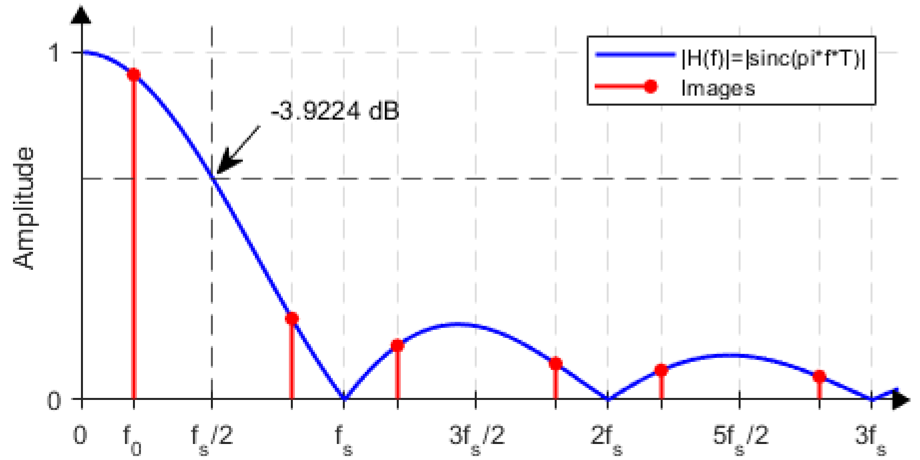


Figure 3–10: Reconstructed signal spectrum directly out of the DAC, but before any reconstruction filtering

Base signal appearing at f_0 and images appearing at $nf_s \pm f_0$ are attenuated by the amplitude response of a sync function.

Reconstruction Example 1: Spectral content

A sample sequence $u(n)$ has the discrete-time Fourier transform depicted here. Drawn below it is the frequency response of the impulse reponse, $h(t)$, the hold function that holds for $1/(3B)$. The frequency response of the output signal is sketched below.

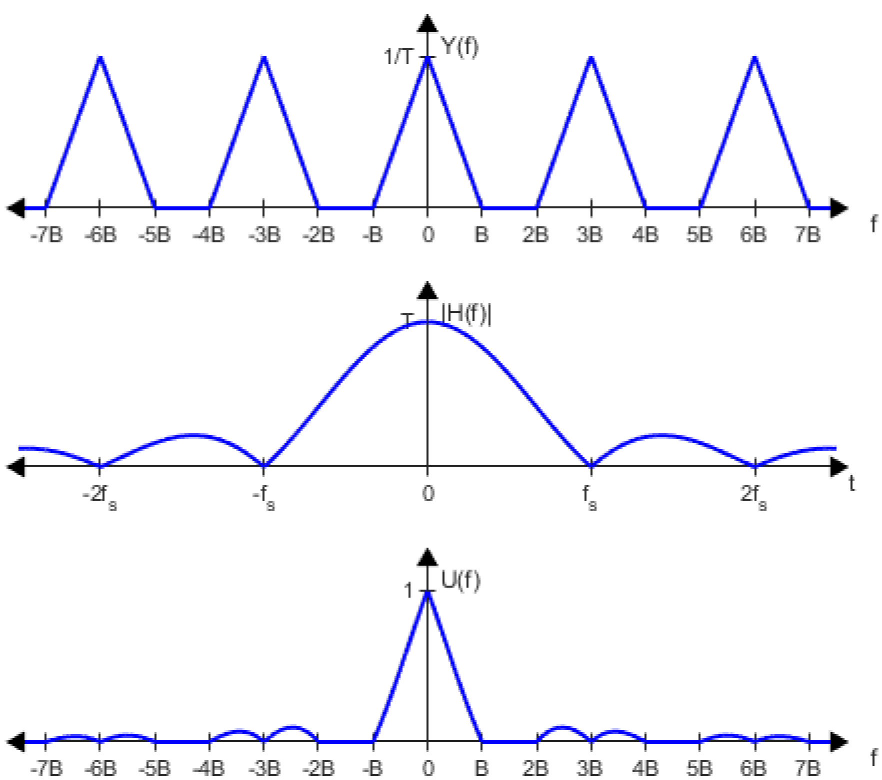


Figure 3-11 – Spectral content through various stages of the reconstruction process

Where,

$$Y_\delta(f) = \frac{1}{T} \sum_{n=-\infty}^{\infty} Y\left(f - \frac{n}{T}\right) \quad (3\text{–}16)$$

$$\left|H(f)\right| = T\left|\mathrm{sinc}(\pi f T)\right| \quad (3\text{–}17)$$

$$U(f) = \left|H(f)\right| Y_\delta(f) = \left|\mathrm{sinc}(\pi f T)\right| \sum_{n=-\infty}^{\infty} Y\left(f - \frac{n}{T}\right) \quad (3\text{–}18)$$

APPENDIX 3A: A REVIEW OF LINEAR CONVOLUTION

Consider first the linear system $g_2(t)$ driven by signal $g_1(t)$:

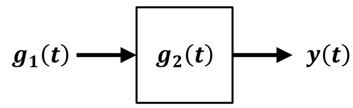


Figure 3–11: A linear system

Recall that $y(t)$ is the convolution of input $g_1(t)$ with impulse response $g_2(t)$.

$$y(t) = \int_{-\infty}^{\infty} g_1(\tau) g_2(t-\tau)\, d\tau \tag{3–19}$$

Of which the Fourier transform is:

$$\mathcal{F}\{y(t)\} = Y(\omega) = \int_{-\infty}^{\infty} \left[\int_{-\infty}^{\infty} g_1(\tau) g_2(t-\tau)\, d\tau \right] e^{-j\omega t}\, dt \tag{3–20}$$

Changing variables; let $t - \tau = \xi \ldots$

Then $\quad dt = d\xi, \qquad$ and $\quad t \Big|_{-\infty}^{\infty} \longrightarrow \xi \Big|_{-\infty}^{\infty}.$

$$Y(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_1(\tau) g_2(\xi)\, d\tau\, e^{-j\omega(\xi+\tau)}\, d\xi \tag{3–21}$$

$$Y(\omega) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g_1(\tau) g_2(\xi)\, e^{-j\omega\xi} e^{-j\omega\tau}\, d\tau\, d\xi \tag{3–22}$$

$$Y(\omega) = \int_{-\infty}^{\infty} g_1(\tau) e^{-j\omega\tau}\, d\tau \int_{-\infty}^{\infty} g_2(\xi) e^{-j\omega\xi}\, d\xi \tag{3–23}$$

$$Y(\omega) = G_1(\omega) G_2(\omega) \tag{3–24}$$

Therefore, convolution in the time domains leads to multiplication in the frequency domain.

APPENDIX 3B: TRANSFORMS FOR SPECTRAL ANALYSIS

This appendix summarizes the five transform types that are used in the material of Chapter 3 and subsequent chapters. They are the:

- Fourier transform (FT)
- Laplace transform (LT)
- Discrete-time Fourier transform (DTFT)
- z-transform (ZT)
- Discrete Fourier transform (DFT)

Also a special form of the DFT, the Fast Fourier Transform (FFT), used extensively in digital signal analysis and algorithm realization, is described.

For time domain signals the Fourier and Laplace transforms apply (see Table 3-1). It can be shown that the FT is a special case of the LT, and that the LT is an extension of the FT that permits signals involving singularities (poles) on the imaginary axis, singularities that pose a convergence problem for the FT analysis. Further information can be found in http://en.wikipedia.org/wiki/Laplace_transform.

For sample domain (discrete-time) signals the Discrete-time Fourier transform and z-transform apply. Like their time domain counterparts, the DTFT is a special case of the ZT, with the ZT being an extension of the DTFT permitting analysis of systems involving singularities in this case on the unit circle, singularities posing a convergence issue for DTFT analysis. Further information is also available in http://en.wikipedia.org/wiki/Z-transform.

The DTFT is derived from the FT by its application to and operation on time domain signals that are sampled at the sampling time interval T. This sampling introduces a finite resolution in time, producing a sequence of samples, $x(n)$, defined over the sample indices $-\infty < n < \infty$, with $x(n)$ containing the value of $x(t)$ at discrete times $t = nT$.

Finally, we limit the number of discrete samples involved in the computation of DTFT $X(e^{j\Omega})$, to the range $0 \leq n \leq (N-1)$, giving rise to the Discrete Fourier Transform $X(e^{j\Omega})$. Computation of the function $X(e^{j\Omega})$, a continuous function of frequency Ω, requires an infinite number of samples n, and provides therefore a function having infinitely fine (i.e. continuous) resolution. Restricting the number of samples to the finite quantity N limits the resolution to N discrete frequencies,

$$\frac{f_s}{2(N-1)}k \tag{3–25}$$

Where $0 \leq k \leq N-1$.

The frequency resolution, $f_s/\big(2(N-1)\big)$, can be controlled through the choice of sample frequency f_s and sequence duration N. We see that the DFT represents the DTFT at the finite set of discrete frequencies evaluated. The DFT is therefore the transform therefore used predominantly in digital signal processing.

Table 3–1: Transform Analysis Types

Time/Sample Span	$-\infty < t < \infty$		$-\infty < n < \infty$		$0 \leq n \leq N-1$
Frequency Span	$-\infty < f < \infty$		$-\pi < \Omega \leq \pi$		$0 \leq k \leq N-1$
Domain: Frequency	$X(f)$		$X(e^{j\Omega})$		$X(k)$
	$\uparrow$ $\mathcal{F}$ $\uparrow$		$\uparrow$ DFDT $\uparrow$		$\uparrow$ DFT $\uparrow$
Domain: Time & Sample	$x(t)$	$\xrightarrow{\Sigma\delta(t-nT)}$	$x(n)$	$\xrightarrow{0\leq n\leq(N-1)}$	$x(n)$
	$\downarrow$ $\mathcal{L}$ $\downarrow$		$\downarrow$ $\mathcal{Z}$ $\downarrow$		

s & z	$X(s)$	$X(z)$

The Fourier Transform is defined by the transform pair:

$$X(f) = \int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}\,dt \tag{3–26}$$

$$x(t) = \int_{-\infty}^{\infty} X(f)e^{+j2\pi ft}\,df \tag{3–27}$$

where the first equation defines the transform of time domain signal $x(t)$ to a function defined over the range of real frequencies $-\infty < f < \infty$ where f has units of Hertz. The inverse transform converts this function of frequency back to $x(t)$. Note that this transform pair is also often stated as follows, using frequency in radians/second rather than in Hz, which results in a 2π term appearing in the inverse transform:

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}\,dt \tag{3–28}$$

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)e^{+j\omega t}\,d\omega \tag{3–29}$$

The Laplace Transform is given by the following:

$$X(s) = \int_{-\infty}^{\infty} x(t)e^{-st}\,dt \tag{3–30}$$

$$x(t) = \frac{1}{2\pi j}\lim_{\beta\to\infty}\int_{\gamma-j\beta}^{\gamma+j\beta} X(s)e^{+st}\,ds \tag{3–31}$$

Where $s = \sigma + j\omega$ and where γ is a real number chosen with a region of convergence in mind. One immediately notes the similarity of these two transforms and sees that the LT equals the FT on the imaginary axes where $s = j\omega$. We will use this fact when computing the frequency response of a transfer function given in the Laplace domain, evaluating the LT with $s = j\omega$.

The Discrete-Time Fourier Transform is defined by the pair:

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x(n)e^{-j\Omega n} \tag{3–32}$$

$$x(n) = \frac{1}{2\pi}\int_{0}^{2\pi} X(e^{j\Omega})e^{-j\Omega n}\,d\Omega \tag{3–33}$$

Where the forward transform is an infinite summation over all non-zero sample $x(n)$. We see that the DTFT is a periodic function of Ω with a period of 2π. Thus it is shown as a function of $e^{j\Omega}$ as a reminder that this function is periodic, as $e^{j\Omega}$ is periodic. Also, it is often represented over one period, $-\pi < \Omega \leq \pi$, where Ω has units of radians/sample. What is the relationship between the angular frequency variable Ω, having units of radians/sample, and that of ω of Equation 4.1 and having a frequency with units of rad/sec? This is easily determined using the sample time interval *T*. The angular rotation of the spinning vector $e^{j\Omega}$ is equal to the angular traversed over one interval *T*. Thus we have

$$\Omega = \omega T$$

and we can compute the range of continuous frequencies ω associated with the angular frequency Ω as follows:

$$-\frac{\pi}{T} < \Omega \leq \frac{\pi}{T}$$

An example of this is given in the body of Chapter 4.

Next, the z-Transform:

$$X(z) = \sum_{n=-\infty}^{\infty} x(n)z^{-n} \quad (3\text{–}34)$$

$$x(n) = \frac{1}{2\pi j}\oint_C X(z)z^{n-1}\,dz \quad (3\text{–}35)$$

Which again involves an infinite summation of the sample values, multiplied by the independent variable z^{-n}. Information on the inverse transform and the basis for the contour integral can be found in: http://en.wikipedia.org/wiki/Z_transform. The z-transform equals the DTFT on the unit circle contour where $z = e^{j\omega}$.

Finally, the equations defining the Discrete Fourier Transform (DFT) and inverse DFT:

$$X(k) = \sum_{n=0}^{N-1} x(n)e^{-j2\pi\frac{nk}{N}} \quad (3\text{–}36)$$

$$x(n) = \frac{1}{N}\sum_{n=0}^{N-1} X(k)e^{j2\pi\frac{nk}{N}} \quad (3\text{–}37)$$

The similarity of this transform pair with that of the DTFT is evident. Considering the case in which $x(n)$ is limited in sample duration to the range $0 \leq n < (N-1)$, then the DFT and DTFT are equal at the N discrete frequencies $2\pi k/N$. Thus we can compute the DTFT at regularly spaced intervals in angular frequency $2\pi/N$ using the DFT. This interval, $2\pi/N$, is therefore the resolution (also called the bin spacing) of the DFT in radians/sample.

We'll define

$$\Delta\Omega = \frac{2\pi}{N}$$

The bin size in terms of the time domain frequency is

$$\Delta\omega = \frac{2\pi}{NT} = \frac{2\pi f_s}{N}$$

where $\Delta\omega$ has units of radians/second. Representing this also with units of Hertz:

$$\Delta f = \frac{\Delta\omega}{2\pi} = \frac{f_s}{N}$$

How is the DFT vector to be interpreted? Its interpretation is provided by an examination of the equations directly above. Notice that the complex exponential is a spinning complex vector moving at one of the discrete angular increments, $\Delta\Omega^*k$, and equivalently at the discrete frequencies, $\Delta\Omega^*k$. At each value of k, the DFT multiplies the k^{th} sample by the spinning complex vector:

$$e^{-j\Delta\Omega^*kn}$$

which increments by the angle $\Delta\Omega^*k$ and coherently accumulates up all of the signal energy occurring at that frequency, placing that into the transformed vector element $X(k)$. That element contains a measure of the signal power at that bin, index k, and at frequency $\Delta\Omega^*k$, that measure being the square root of the power.

Finally, what is the Fast Fourier Transform (FFT) and its relationship to the DFT? Simply put, the FFT is the DFT, and it is formally defined no differently; however, when the number of samples N is a power of 2, then a particularly advantageous computational algorithm can be applied which significantly speeds up the computation of the transform vector $X(k)$. It is the Cooley-Tukey algorithm providing this computational advantage, reducing the number of computations from the order of N^2 to $N \log_2 N$. See http://en.wikipedia.org/wiki/Fast_Fourier_transform for further details.

SPECTRAL ANALYSIS

When the DFT and FFT are used for spectral analysis, a finite length of discrete-time samples $x(n)$ are used to represent the continuous-time signal $x(t)$ during the same time period. We demonstrated above that taking the Fourier transform of the signal sampled by the ideal sampler (3–1), results in the generation of the discrete-time Fourier transform, and that this gives rise to a potential source of distortion called aliasing.. The DTFT is a continuous function of linear frequency ω, infinite in resolution in ω. It therefore requires an infinitely large quantity of samples $x(n)$ in its computation. When processing only a finite number of samples as is always the case with real data and with limited computation resources, only a total of N samples are considered, the samples $x(n)$ existing over the interval $0 \le n < (N-1)$. The availability of a limited number of samples produces a limited resolution along the frequency axis of the discrete Fourier transform (DFT), limiting it to a set of N discrete samples $X(k)$, actually samples of the DTFT $X(e^{j\Omega})$ at N points on the unit circle $e^{j\Omega}$. This gives rise to another type of distortion called leakage, the phenomena in which signal energy in a particular DFT bin contributes to the DFT values in all other bins to some degree. This is associated with the finite resolution of the DFT and is illustrated by example below. As expected, this affect is reduced by increasing the number of samples N of $x(n)$, and reducing the bin size.

FFT Example 1: Computation of bin resolution

Consider sample stream $x(n)$ generated by sampling a continuous time signal $x(n)$ at 1000 Hz. A sequence of 1024 samples is to be processed in order to be compatible with the FFT (1024 is a factor of 2). What is the frequency resolution in radians/sample, also called the bin spacing? Express this same spacing in terms of the time domain frequency having units of radians/sec? Express the later also in Hz.

Solution – The number of samples is N = 1024, thus the frequency resolution is

$$\Delta\Omega = \frac{2\pi}{N} = \frac{2\pi}{1024} = 0.0061359 \text{ radians/sample}$$

The time domain frequency resolution depends on the sample time interval…

$$T = 1/1000 = 0.001 \text{ seconds/sample}$$

$$\Delta\omega = \frac{2\pi}{NT} = \frac{0.0061359}{0.001} = 6.1359 \text{ radians/second}$$

Converting this to cycles/sec:

$$\Delta f = 6.1359 \frac{\text{rad}}{\text{sec}} \cdot \frac{1}{2\pi} \frac{\text{cycle}}{\text{rad}} = 0.9766 \text{ Hz}$$

Alternatively, we can use the bin spacing formula $\Delta f = f_s/N$

$$\Delta f = \frac{1000}{1024} = 0.9766 \text{ Hz}$$

FFT Example 2: Computation and plotting of the DFT

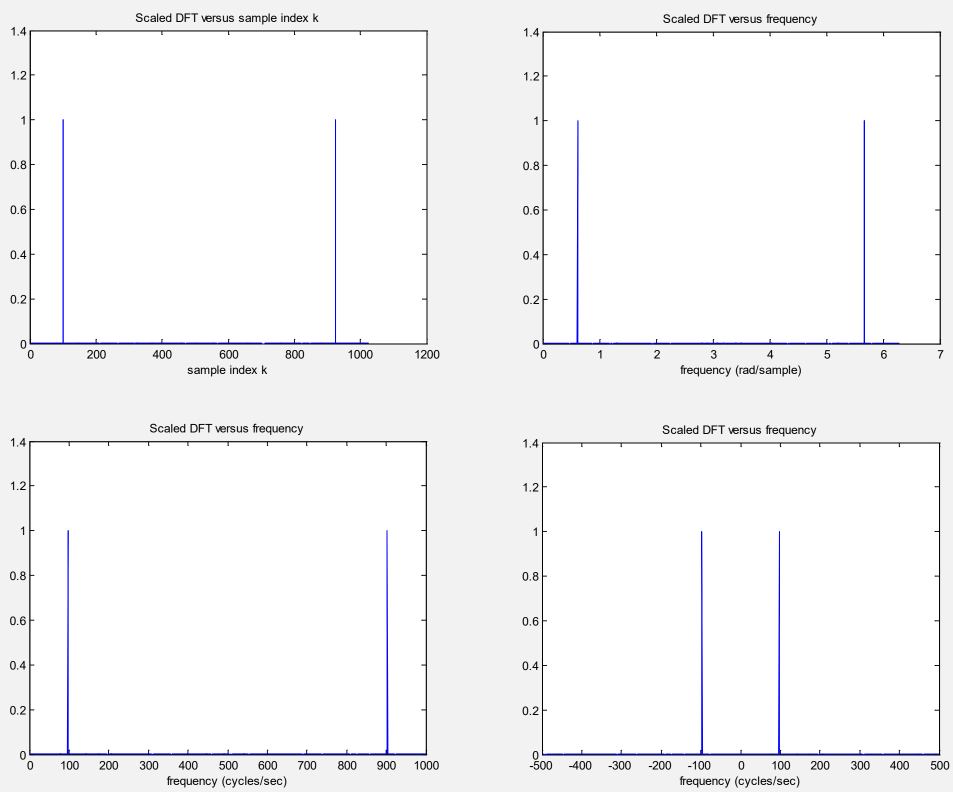
Continuing the example above, $x(t)$ is a pure sinusoid with a frequency of $100 \times \Delta\omega = 97.6$ Hz, placing the signal directly at the center of the 101st bin. Compute the FFT of this sample stream with the Matlab fft() function; divide this result by $N/2$ to scale the output to signal amplitude. Plot the scaled discrete Fourier transform $2|X(k)|/N$ as a function of:

a) sample index k,
b) sample domain bin frequency Ω,
c) time domain bin frequency f, and
d) time domain bin frequency f over the range -500 to +500 Hz.

Solution – The following Matlab .m file was created to produce the required plots. The results are as follows:

```
%% FFT example 2, Ch 4 -- Signal Conditioning for Digital Control
N = 1024;  % number of samples
T = 1/1000; % sample frequency = 1000 Hz
df = (2*pi/N/T) / (2*pi);
fo = 100*df;

% create samples from time domain signal
t = (0:1:(N-1))*T;
x = sin(2*pi*fo*t);

%% Compute the FFT and find the magnitude
X = fft(x);
absX = abs(X);

plot(absX*2/N)
```

```
xlabel('sample index k')
title('Scaled DFT versus sample index k')

%% Plot as function of bin frequency
dOmega = 2*pi/N;
Omega = (0:(N-1))*dOmega;
f = (0:(N-1))*df;

figure(2)
plot(Omega,absX*2/N)
xlabel('frequency (rad/sample)')
title('Scaled DFT versus frequency')

figure(3)
plot(f,absX*2/N)
xlabel('frequency (cycles/sec)')
title('Scaled DFT versus frequency')

%% Plot with FFT zero frequency appearing at the center of the
spectrum
absXs = fftshift(absX);
fs = fftshift(f);

figure(4)
plot(f-(1/2/T),absXs*2/N)
xlabel('frequency (cycles/sec)')
title('Scaled DFT versus frequency')
```



FFT Example 3: Illustration of the effects of leakage

The carrier frequency of the sinusoid is changed to a frequency of $100 \times \Delta\omega = 97.6$ Hz, placing the signal directly at the center between the 101st and 102nd bins. Plot the magnitude from 75 to 125 Hz using a stem plot to zoom in and clearly display the leakage effect.

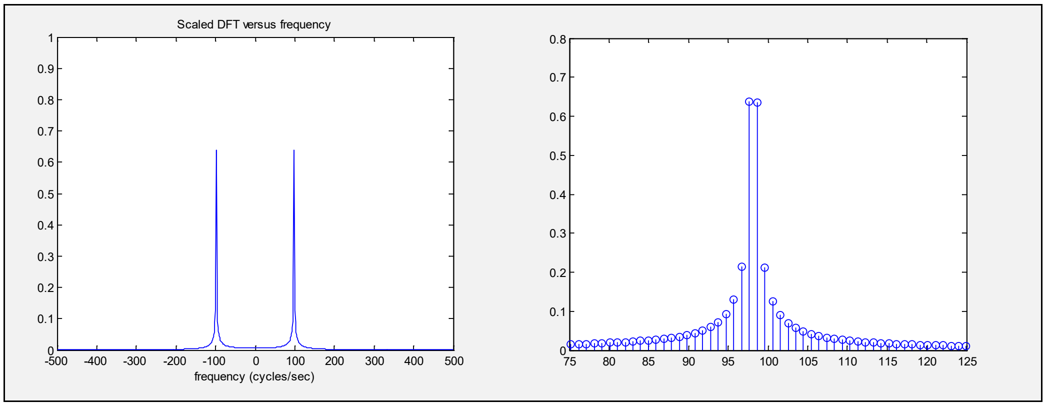
Solution – In the example above, line 5 of the .m file was changed to reflect the change in frequency:

```
fo = 100.5*df;
```

Nothing else was changed, except the following code was added to zoom in:

```
figure(5)
stem(f-(1/2/T),absXs*2/N)
axis([75 125 0 0.8])
```

The result, the two plots:



Notice that the energy has been distributed over the bins, reducing the amplitude of the signal at bin 101 from 1 to about 0.67, and spreading signal energy to all of the other bins.

FFT Example 4: Identifying signal components in random noise

Use the FFT to identify the presence and frequency of two signals hidden in Gaussian noise. For the parameters of the example above, generate a sample sequence given by:

$$x(t) = \sin(2\pi f_0 t) + 0.8\sin(2\pi f_1 t)$$

with $f_0 = 58.6$ Hz, $f_1 = 195.3.6$ Hz, and sampled at $T = 1000$ Hz.

To this add noise samples $\eta(n)$ having a standard deviation of 3 (i.e. a variance of 9):

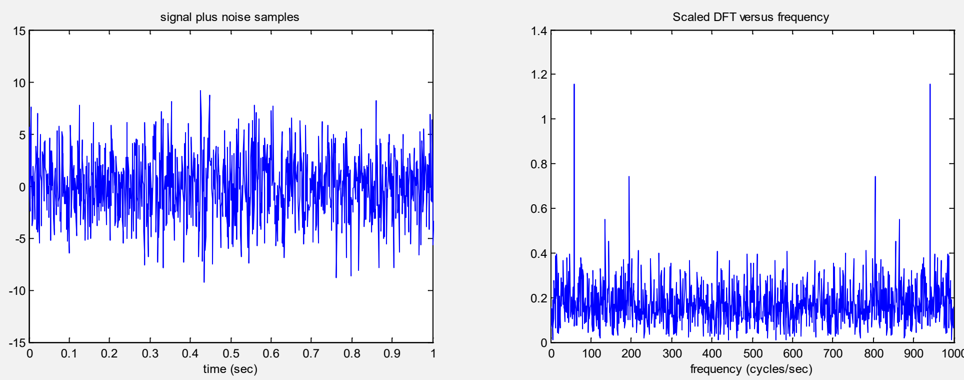
$$x(n) = x(n) + \eta(n)$$

Plot the time domain samples and the DFT of the signal.

Solution – We generate the samples as follows:

```
%% FFT example 4, Ch 4 -- Signal Conditioning for Digital Control
N = 1024;  % number of samples
T = 1/1000; % sample frequency = 1000 Hz
df = (2*pi/N/T) / (2*pi);
fo = 60.*df;
f1 = 200.*df;
```

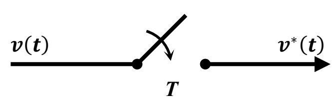
```
% create samples from time domain signal
t = (0:1:(N-1))*T;
x = sin(2*pi*fo*t) + 0.8*sin(2*pi*f1*t);

% add noise

x = x + 3*randn(size(t));

% Plot the signal and noise time domain data
figure(1)
plot(t,x)
xlabel('time (sec)')
title('signal plus noise samples')
axis([0 1 -15 15])

%% Compute the FFT and find the magnitude
X = fft(x);
absX = abs(X);
plot(absX*2/N)
xlabel('sample index k')
title('Scaled DFT versus sample index k')
```



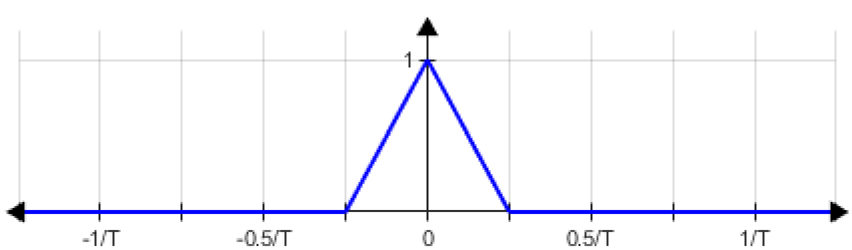
Homework Chapter 3

Sampling, Aliasing and Anti-Alias Filtering

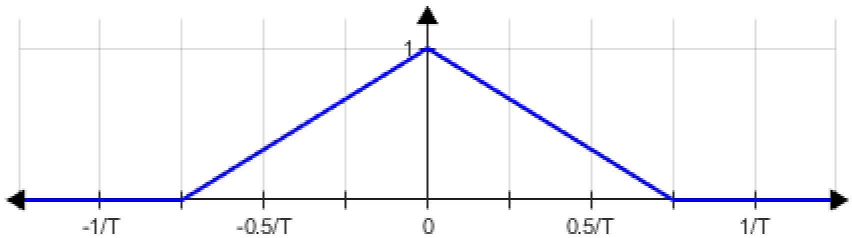
Samp–3–1: A continuous-time signal $v(t)$ is sampled by an ideal sampling device at a sampling frequency $1/T$, producing the sampled signal $v^*(t)$ as shown in the model:



The Fourier Transform of $v(t)$ has a magnitude plot which is zero everywhere except within the range $-0.25/T$ to $+0.25/T$ as shown:



Plot the Fourier Transform of the sampled signal from $-2/T$ to $+2/T$ clearly labeling amplitude and frequency axis. Repeat for the following:



Samp–3–2: We are given a time domain signal given by:

$$x(t) = \sin(2\pi 40t) + 0.8\sin(2\pi 2400t)$$

which is sampled at 2000 Hz. Plot the signal from 0 to 1.5 seconds. Process 2048 samples using an FFT and plot the spectrum from $-f_s/2$ to $+f_s/2$. Decide if there is aliasing. Where does the signal at 2400 Hz appear? Add an anti-alias filter to the time-domain model. Use a second order filter consisting of two first-order filters having break frequencies of 200 Hz. Repeat the generation of the sample domain plot and FFT.

Spectral Plotting and Analysis with FFT, Reconstruction Filtering

FFT–3–1: Consider sample stream $x(n)$ generated by sampling a continuous time signal $x(t)$ at 10,000 Hz. A sequence of 2048 samples is to be processed via the FFT. What is the frequency resolution (also called the bin spacing) in radians/sample, radians/second, and Hertz?

FFT–3–2: A signal $x(t)$ of problem FFT–3–1 is a pure sinusoid with a frequency of $200^*\omega$, placing the signal directly at the center of the 201st bin. Compute the FFT of this sample stream with the Matlab fft() function. Plot the absolute value and scaled discrete Fourier transform $2|X(k)|/N$ as a function of

a) sample index k,
b) sample domain bin frequency Ω,
c) time domain bin frequency from 0 to 10,000 Hz, and
d) time domain bin frequency over the range –500 to +500 Hz.

FFT–3–3: Determine the sample frequency required in Hz to achieve a frequency resolution of 0.1 Hz when a 1024 sample FFT is used?

Rec-1: What is the purpose of the reconstruction filter on the output of a DAC?

Rec-2: What is the shape of the transfer function that is associated with the sample and hold function occurring at the output of a DAC? Sketch the magnitude response for the case in which the sample time is 10 kHz.