Aliasing

Introduction

Sampling is a nonlinear process. As a result, it distorts the frequency content of a signal. Aliasing is when that distortation changes the low-frequecy content of a signal, meaning that the processor is getting faulty readings.

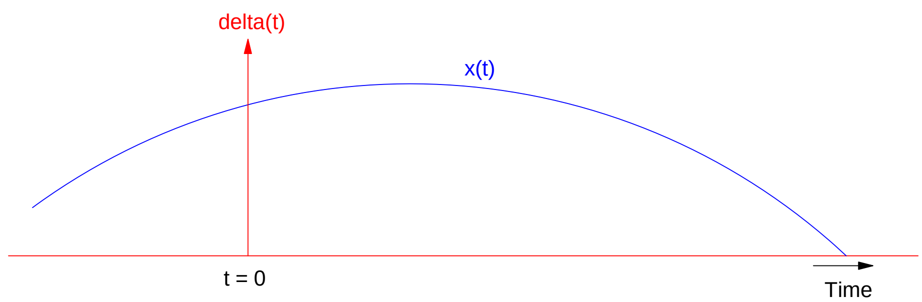
Convolution

The delta function is defined as

$$\delta(t) = \lim_{\varepsilon \to 0} \begin{cases} 0 & |t| > \varepsilon \\ \frac{1}{2\varepsilon} & |t| < \varepsilon \end{cases}$$

If you multuply a signal by the delta function, you get that function at t=0

$$x(t) \cdot \delta(t) = x(0) \cdot \delta(t)$$



Given a signal, x(t), it can be written as the convolution of x(t) with the delta function

$$x(t) = \int_{-\infty}^{\infty} x(\tau) \cdot \delta(t-\tau) \cdot d\tau$$

If you apply a delta function to a filter, you get its impulse response, h(t). Similarly, if you apply any other function to a filter, you get a convolution

$$y(t) = \int_{-\infty}^{\infty} x(\tau) \cdot h(t-\tau) \cdot d\tau$$

$$y(t) = x(t) ** h(t)$$

where ** denotes convolution.

If you convert to the frequency domain (i.e. take LaPlace transforms), the you get

$$Y(s) = H(s) \cdot X(s)$$

LaPlace Transforms convert convolution in the time-domain in to multiplication in the frequency domain.

The assumption is that multiplication is easier than convolution. Hence, the popularity of LaPlace Transforms.

Convolution and Sampling

The time-domain and the frequency-domain are related.

- If you multiply in the frequency domain, you use convolution in the time domain.
- If you multiply in the time domain, you use convolution in the frequency domain.

Sampling is essentially multiplying by a series of delta functions in the time-domain

$$y(kT) = x(t) \cdot \delta(t - kT)$$

< figure - multiplying in the time domain >

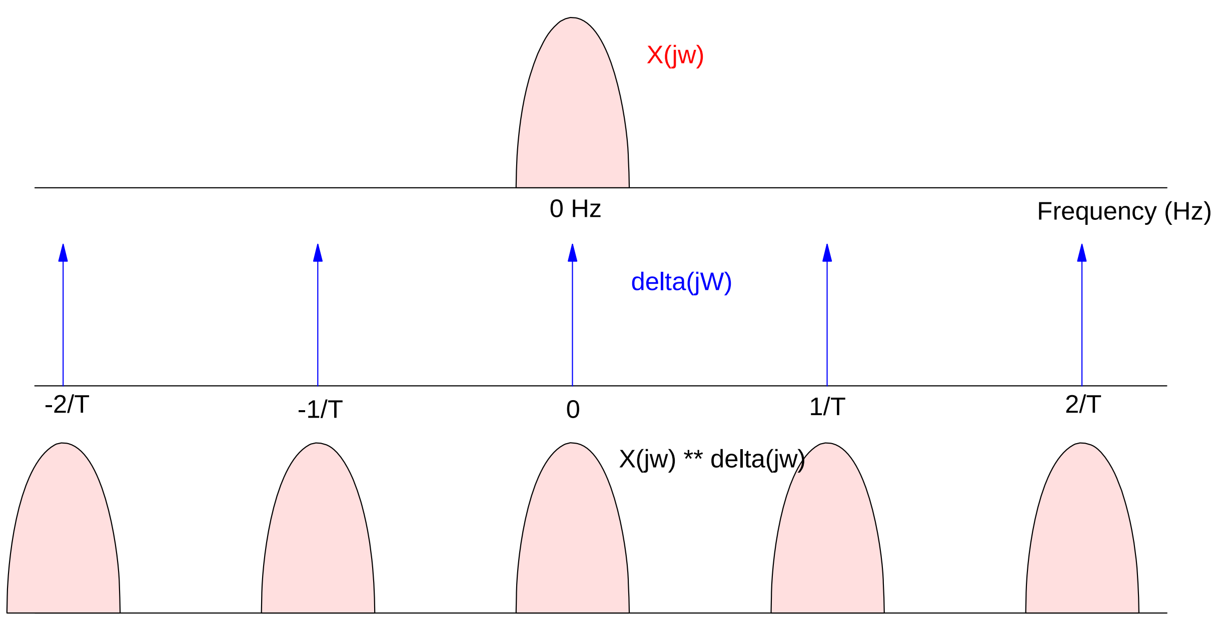
This means you are using convolution in the frequency domain

$$Y(s) = X(s) * * \delta(s)$$

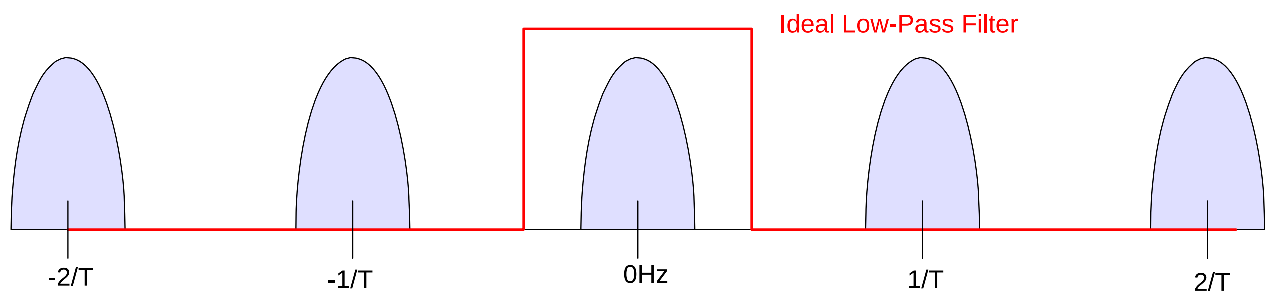
The Fourier Transform for a delta train is one:

$$\delta(t - kT) = \sum_{n=0}^{\infty} 1 \cdot \cos(n\omega_0 t)$$

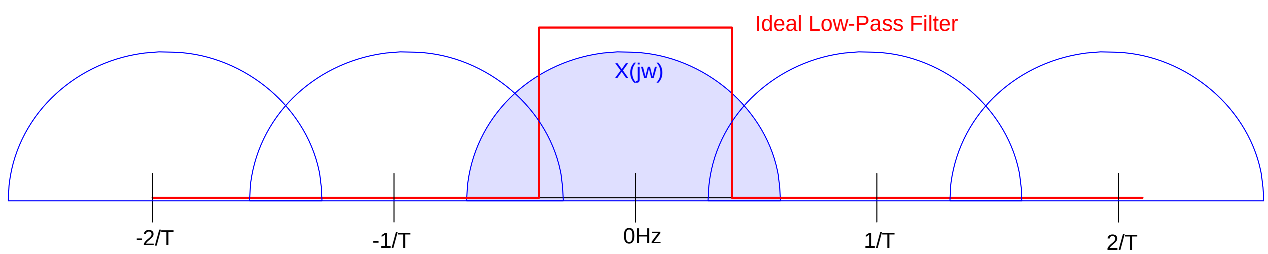
This means that the frequency content of a sampled signal is the sum of that signal, repeated every 1/T Hz



If the signals do not over-lap, it is possible to recover x(t) from the sampled signal



If the signals *do* over-lap, it is no longer possible to recover x(t) from its sampled signal



This is what aliasing is: being unable to recover the orignal signal after sampling.

The condition necessary for a signal to be recoverable is

- **The highest frequency contained in a signal must be less than 1/2 of the sampling rate**

a.k.a. the Nyquiste limit, or said differently

- **The sampling rate must be at least 2x the highest frequency component of the signal being sampled**

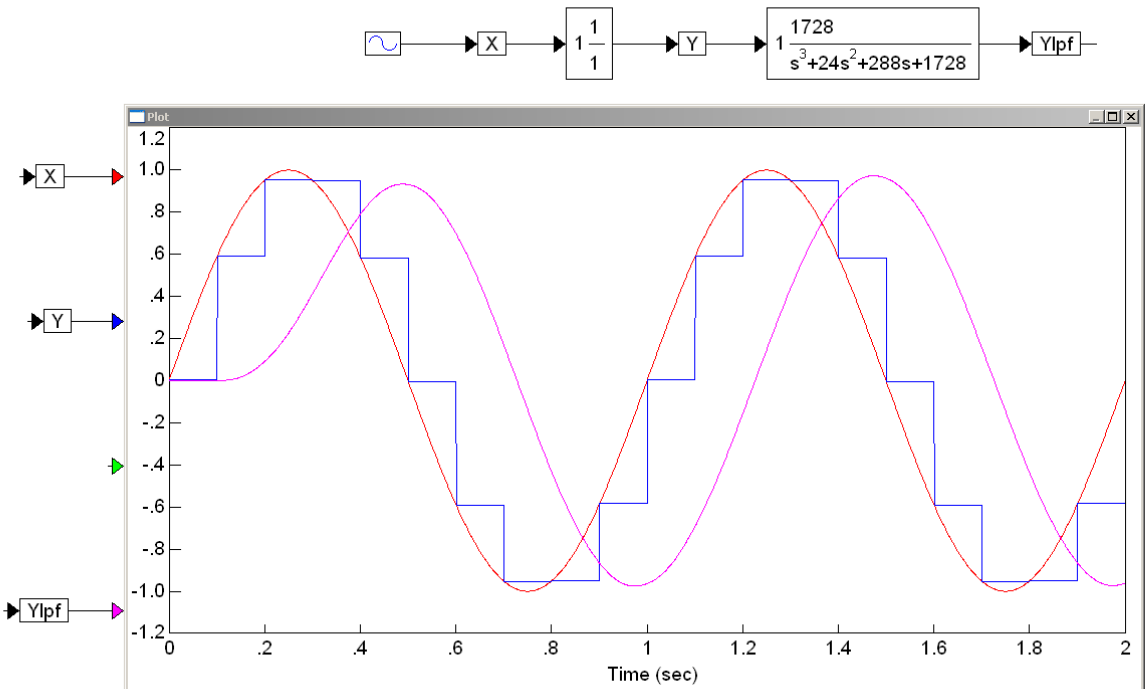
a.k.a. Shannon's Sampling Theorem

Example:

- Sample a 1Hz sine wave at 10Hz.
- Pass the result through a low-pass filter with a corner at 2Hz

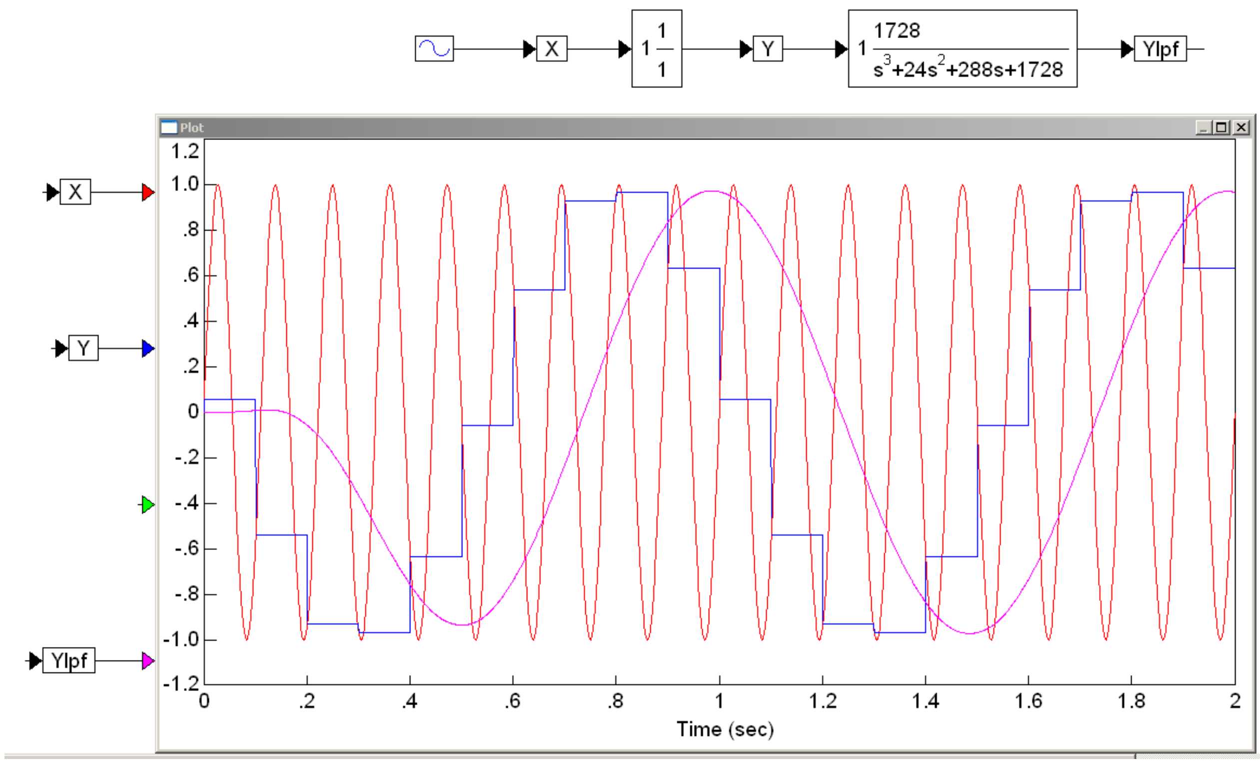
In this case, the highest freuquency of the input (1Hz) is less than 1/2 of the sampling rate (1/2 * 10Hz).

You can recover the orignal signal with filtering (A low-pass filter with a corner at 2Hz used here)



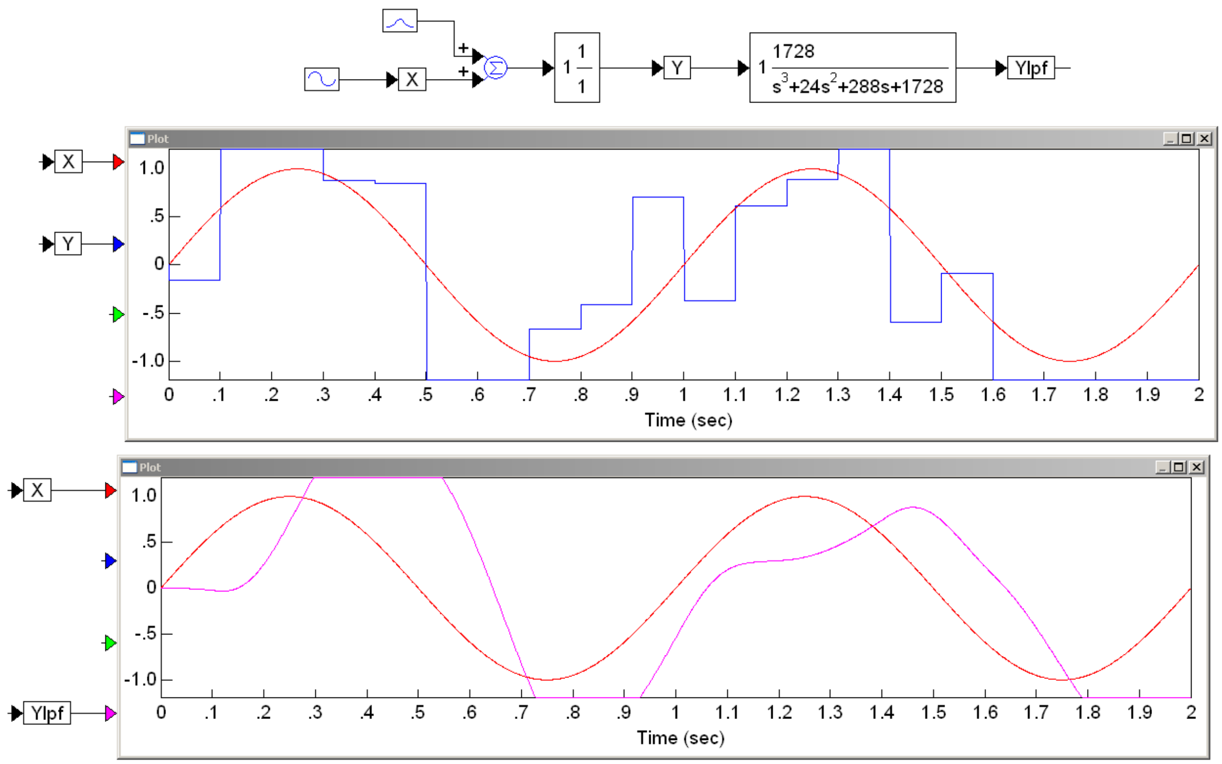
Example 2: Sample a 9 Hz sine wave at 10Hz.

In this case, the inpt is past the Nyquize limit (5Hz). You cannot recover the orignal signal with filtering.



Example 3: 1Hz Sine Wave + Noise, Sampled at 10Hz

White noise contains all frequencies with the same amplitude. Once sampled, you cannot separate the noise from the signal.



Anti-Aliasing Filters

In order to avoid distortion, you have to make sure there are no inputs past 1/2 of the sampling rate.

If your signal *does* have signals past 1/2 of the sampling rate, you have to filter these out *before* sampling.