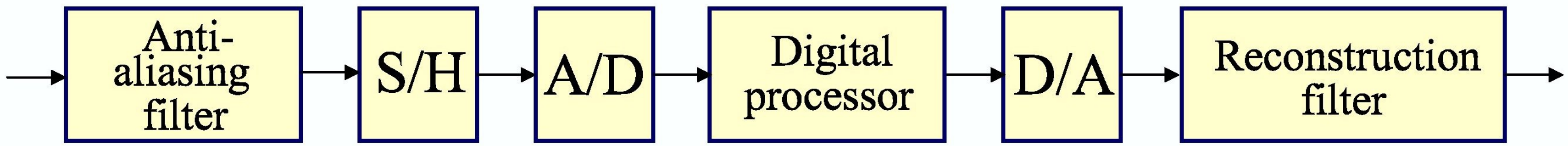
Sampling: examined in detail

- The Sampling Theorem
  - The need for sampling and the aliasing problem
  - Shannon's sampling theorem
    - Nyquist rate
  - Effect of sampling in frequency domain
  - Sampling explained graphically!
  - Recovering the original signal.
    - The sinc function

Analog → Digital → Analog

- ➲ Most signals in nature are continuous in time
  - Need a way for “digital processing of continuous-time signals.”
- ➲ A three-step procedure
  - Conversion of the continuous-time signal into a discrete-time signal
    - Anti-aliasing filter – to prevent potentially detrimental effects of sampling
    - Sample & Hold – to allow time to the A/D converter
    - Analog to Digital Converter (A/D) – actual conversion in time and amplitude
  - Processing of the discrete-time signal
    - Digital Signal Processing – Filter, digital processor
  - Conversion of the processed discrete-time signal back into a cont.-time signal
    - Digital to analog converter (D/A) – to obtain the cont. time signal
    - Reconstruction / smoothing filter - smooth out the signal from the D/A

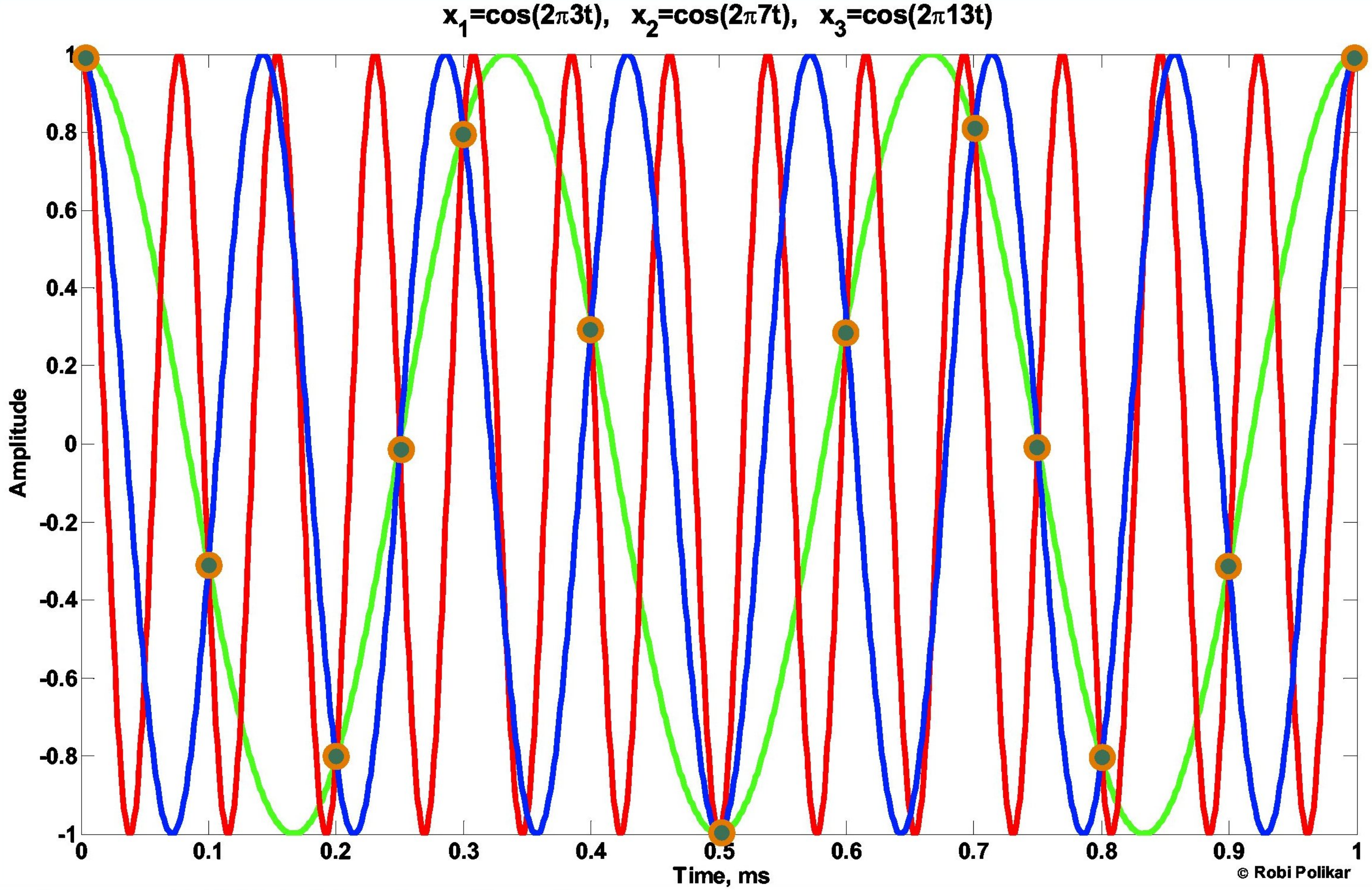


Alias:

Two names for the same person, or thing.

ALIASING

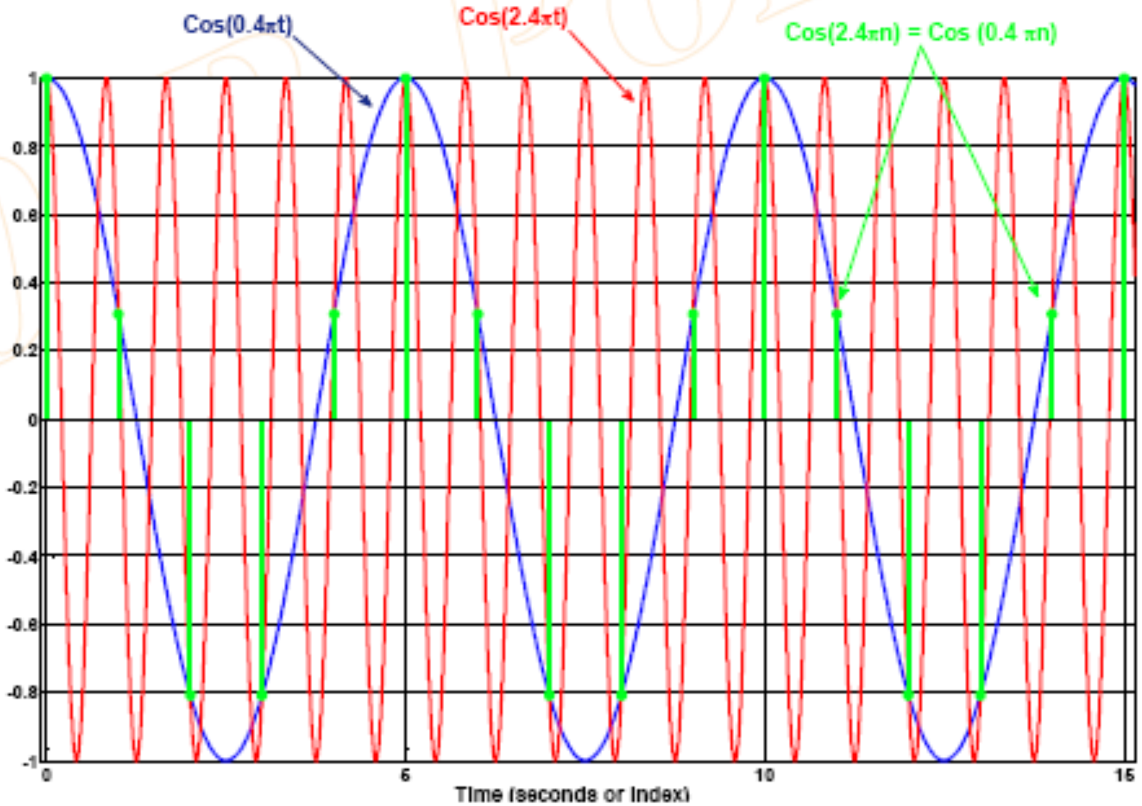
➲ Note that identical discrete-time signals may result from the sampling of more than one distinct continuous-time function. In fact, there exists an infinite number of continuous-time signals, which when sampled, lead to the same discrete-time signal



ALIASING
(ANOTHER EXAMPLE)

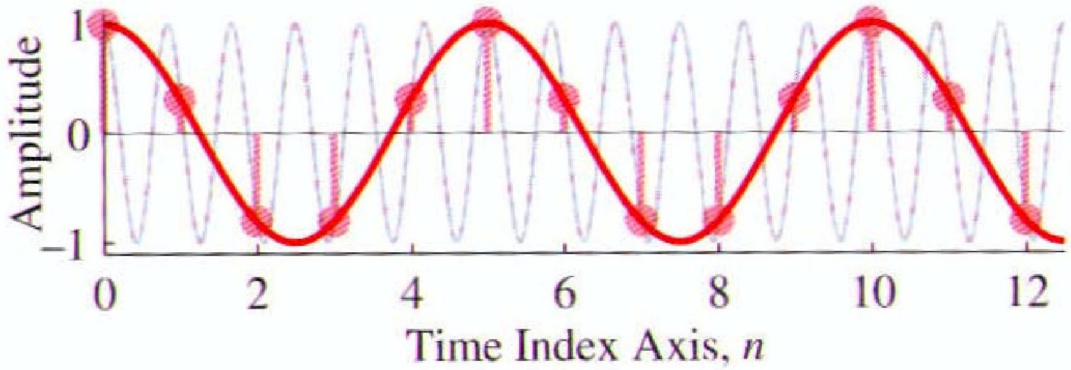
- The signals in the following plot shows two sinusoids: $x_1[n]=\cos(0.4\pi n)$ and $x_2[n]=\cos(2.4\pi n)$. Note that these two signals are distinct, as the second one clearly has a higher frequency.
- However, when sampled at, say integer values of n, they have the same values, that is $x_1[n]=x_2[n]$ for all integer n. These two signals are aliases of each other. More specifically, in the DSP jargon, we say that the frequencies $\omega_1= 0.4\pi$ and $\omega_2=2.4\pi$ are *aliases* of each other.
- This is why all signals– when represented in frequency domain – are normalized to a 2π interval

```
n=0:99;
x1=cos(0.4*pi*n); x2=cos(2.4*pi*n);
t=0:0.001:100;
y1=cos(0.4*pi*t); y2=cos(2.4*pi*t);
plot(t,y1, 'b');
hold
plot(t,y2, 'r');
stem(n,x1, 'g');
stem(n,x2, 'g');
```



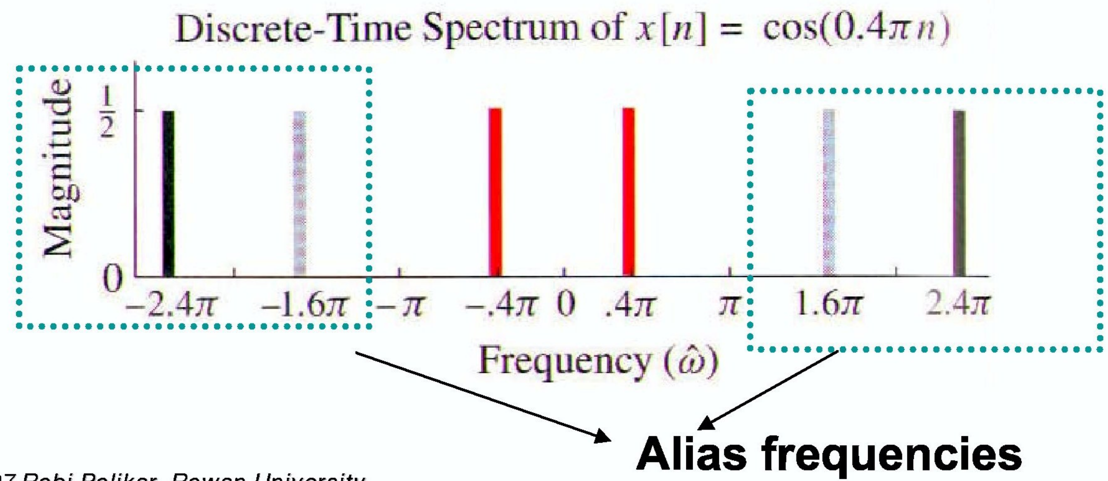
ALIASING

- Here is another example:
  - The signals in the following plot shows two sinusoids: $x_1[n]=\cos(0.4\pi n)$ and $x_2[n]=\cos(2.4\pi n)$. Note that these two signals are distinct, as the second one clearly has a higher frequency.
  - However, when sampled at, say integer values of *n* , they have the same values, that is $x_1[n]=x_2[n]$ for all integer ***n***. These two signals are aliases of each other. More specifically, in the DSP jargon, we say that the frequencies $\omega_1= 0.4\pi$ and $\omega_2=2.4\pi$ are aliases of each other.
  - This is why all signals and systems – when represented in frequency domain – are normalized to a 2π interval



ALIASING

- In general, 2π multiples added or subtracted to a sinusoid gives aliases of the same signal.
  - The one at the lowest frequency is called the ***principal alias***, whereas those at the negative frequencies are called ***folded aliases***.
  - In summary, the frequencies at $\omega_0+2\pi k$ and $2\pi k-\omega_0$ for any integer k, are aliases of each other.
  - We can further show that for folded aliases, the algebraic sign of the phase angle is opposite that of the principal alias



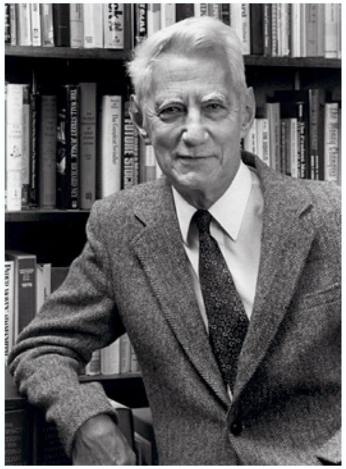
HOUSTON… WE'VE GOT A PROBLEM!*

- The fact that there exists an infinite number of continuous-time signals, which when sampled lead to the same discrete-time signal, poses a considerable dilemma in plotting and interpreting signals in the frequency domain.
- Q: If the same discrete signal can be obtained from several continuous time signals, how can we uniquely reconstruct the original continuous time signal that generated the discrete signal at hand?
- A. Under certain conditions, it is possible to relate a unique continuous-time signal to a given discrete-time signal.
  - If these conditions hold, then it is possible to recover the original continuous-time signal from its sampled values

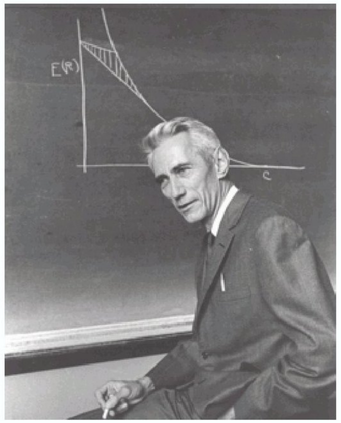
Shannon's Sampling Theorem

- The solution to this complicated and perplexing phenomenon comes from the amazingly simple *Shannon's Sampling Theorem*, one of the cornerstones of the modern communications, signal processing and control systems.
  - A continuous time signal $x(t)$, with frequencies no higher then $\Omega_{max}=2\pi f_{max}$ can be reconstructed ***exactly, precisely*** and ***uniquely*** from its samples $x[n] = x(nT_s)$, if the samples are taken at a sampling rate (frequency) of $f_s=1/T_s$ or ($\Omega_s=2\pi/T_s$) that is greater then $2f_{max}$. The frequency $\Omega_s/2$ (or $f_s/2$ or f_{max}) is called the ***Nyquist frequency*** (or ***folding frequency***), as it determines the minimum sampling frequency required. The minimum required sampling frequency is then called the ***Nyquist rate.***
  - In other words, if a continuous time signal is sampled at a rate that is at least twice as high (or higher) as the highest frequency in the signal, then it can be uniquely reconstructed from its samples.
  - Aliasing can be avoided if a signal is sampled at or above the Nyquist rate.

Claude Shannon



www.research.att.com



Shannon, Claude Elwood (1916-2001), American applied mathematician and electrical engineer, noted for his development of the theory of communication now known as information theory. Born in Gaylord, Michigan, Shannon attended the University of Michigan and in 1940 obtained his doctorate from the Massachusetts Institute of Technology, where he became a faculty member in 1956 after working at Bell Telephone Laboratories. In 1948 Shannon published "A Mathematical Theory of Communication," an article in which he presented his initial concept for a unifying theory of the transmitting and processing of information. Information in this context includes all forms of transmitted messages, including those sent along the nerve networks of living organisms; information theory is now important in many fields.

Effect of Sampling in the Frequency Domain

➲ Mathematically, we can show that the spectrum of the discrete (sampled) signal is simply a 2π replicated and $1/T_s$ normalized version of the spectrum of the original continuous time signal

$$G_p(j\Omega) = \frac{1}{T_S}\sum_{k=-\infty}^{\infty} G_a\left(j\left(\Omega + k\Omega_S\right)\right)$$

Spectrum of the sampled signal

Spectrum of the continuous time signal

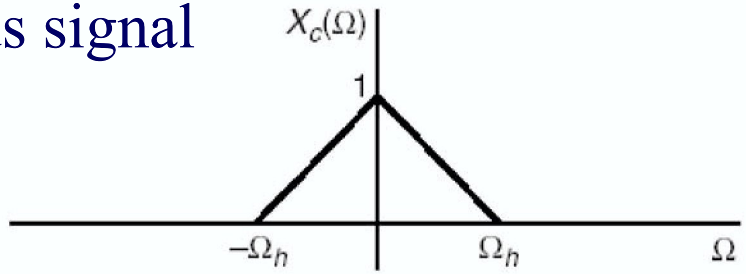
Sampling frequency

$$\boxed{\omega = \Omega T_S}$$

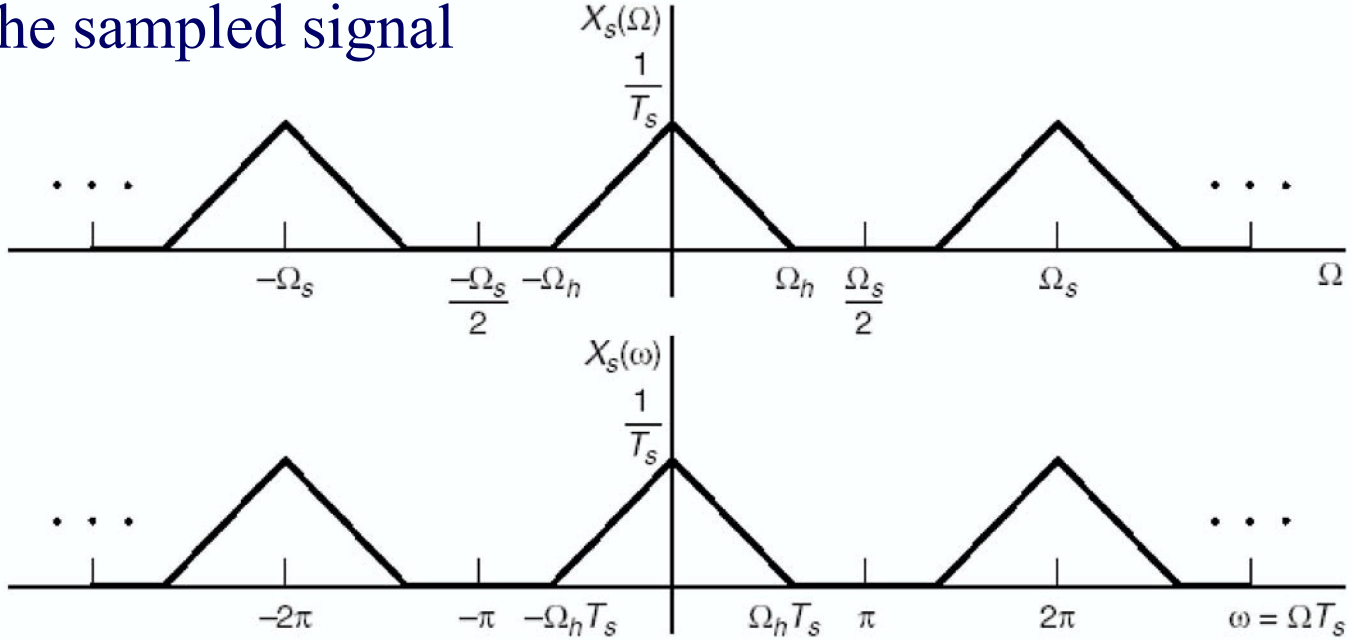
Note that if $\boxed{\Omega = \Omega_s \Rightarrow \omega = \Omega_s T_s = 2\pi f_s T_s = 2\pi \frac{1}{T_s} T_s = 2\pi}$

Effect of Sampling in the Frequency Domain

spectrum of the continous signal



spectrum of the sampled signal



Sampling Explained …

- Let $g_a(t)$ be a continuous-time signal that is sampled uniformly at $t = nT_s$, generating the sequence $g[n]$ where

$$g[n] = g_a(nT_s), \quad -\infty < n < \infty$$

- Now, the frequency-domain representation of $g_a(t)$ is given by its continuous-time Fourier transform (CFT):

$$\boxed{G_a(j\Omega) = \int_{-\infty}^{\infty} g_a(t) e^{-j\Omega t} dt}$$

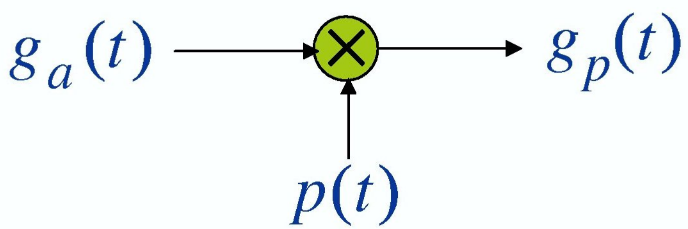
- The frequency-domain representation of $g[n]$ is given by its discrete-time Fourier transform (DTFT):

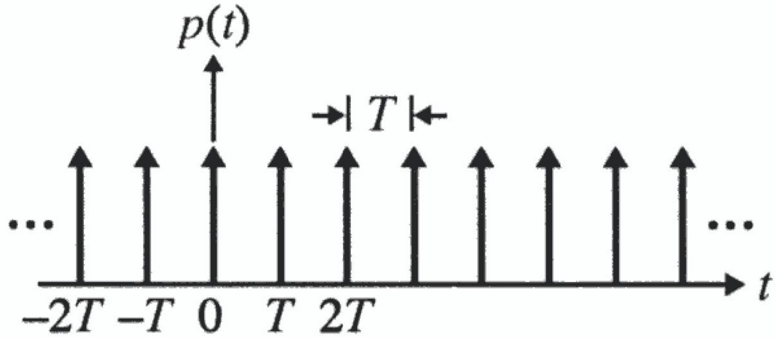
$$G(e^{j\omega}) = G(\omega) = \sum_{n=-\infty}^{\infty} g[n] e^{-j\omega n}$$

SAMPLING

➲ To establish the relationship between $G_a(\Omega)$ and $G(\omega)$, we treat the sampling operation mathematically as a multiplication of $g_a(t)$ by a *periodic impulse train* $p(t)$:

$$\boxed{p(t) = \sum_{n=-\infty}^{\infty} \delta(t - nT)}$$



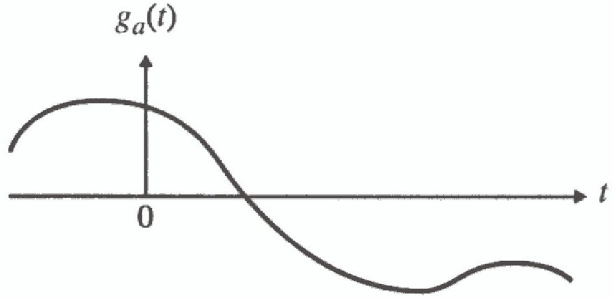


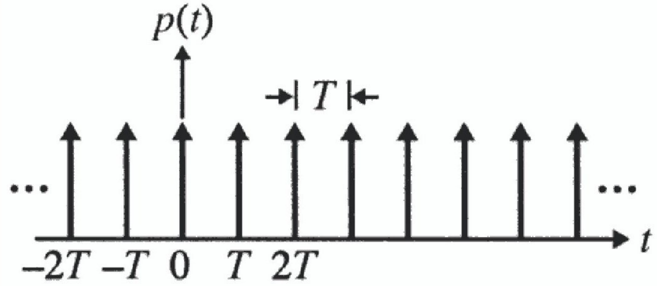
Sampling

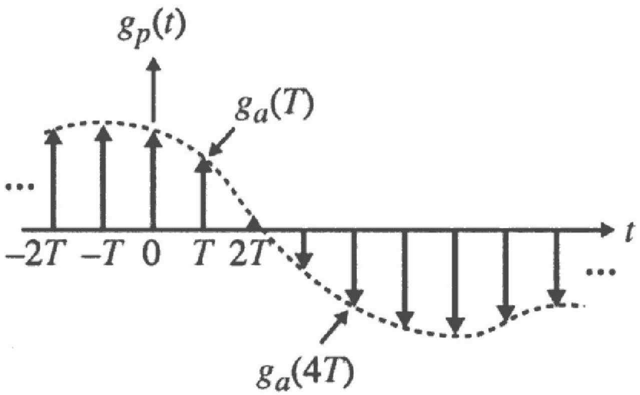
- The multiplication operation yields another impulse train:

$$g_p(t) = g_a(t)\,p(t) = \sum_{n=-\infty}^{\infty} g_a(nT)\,\delta(t - nT)$$

- $g_p(t)$ is a continuous-time signal consisting of a train of uniformly spaced impulses with the impulse at $t = nT_s$ weighted by the sampled value $g_a(nT_s)$ of $g_a(t)$ at that instant







From the convolution theorem we know that multiplication in the frequency domain corresponds to convolution in the time domain, and, similarly, that multiplication in the time domain corresponds to convolution in the frequency domain. Thus from

$$g_p(t) = g_a(t)\, p(t)$$

one gets

$$G_p(f) = G_a(f) * P(f)$$

with:

$$P(f) = F\left(\sum_{n=-\infty}^{\infty} \delta(t - nT)\right) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T}\right)$$

(F is here a symbol indicating the Fourier Transform)

Then we have for $G_p(f)$

$$G_p(f) = G_a(f) * \frac{1}{T} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T}\right) = \frac{1}{T} \sum_{n=-\infty}^{\infty} \int_{-\infty}^{\infty} G_a(f_1)\, \delta\left(\left(f - \frac{n}{T}\right) - f_1\right) df_1$$

so

$$G_p(f) = \frac{1}{T}\sum_{n=-\infty}^{\infty} G_a\left(f - \frac{n}{T}\right)$$

so, if the sampling frequency is $f_s = \frac{1}{T_s}$
then

$$G_p(f) = \frac{1}{T_s}\sum_{n=-\infty}^{\infty} G_a\left(f - \frac{n}{T_s}\right)$$

hence $G_p(f)$ is a periodic function with period $\frac{1}{T_s}$ consisting of a sum of shifted Fourier transforms of $g_a(t)$

this can be rewritten as: $$\boxed{G_p(j\Omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} G_a(j(\Omega + k\Omega_T))}$$

$G_p(j\Omega)$ is a periodic function of *Ω* consisting of a sum of shifted (by *Ω_T*) and scaled (by 1/T) replicas of *$G_a(j\Omega)$*. The term with k=1 is called the baseband portion of *$G_p(j\Omega)$*.

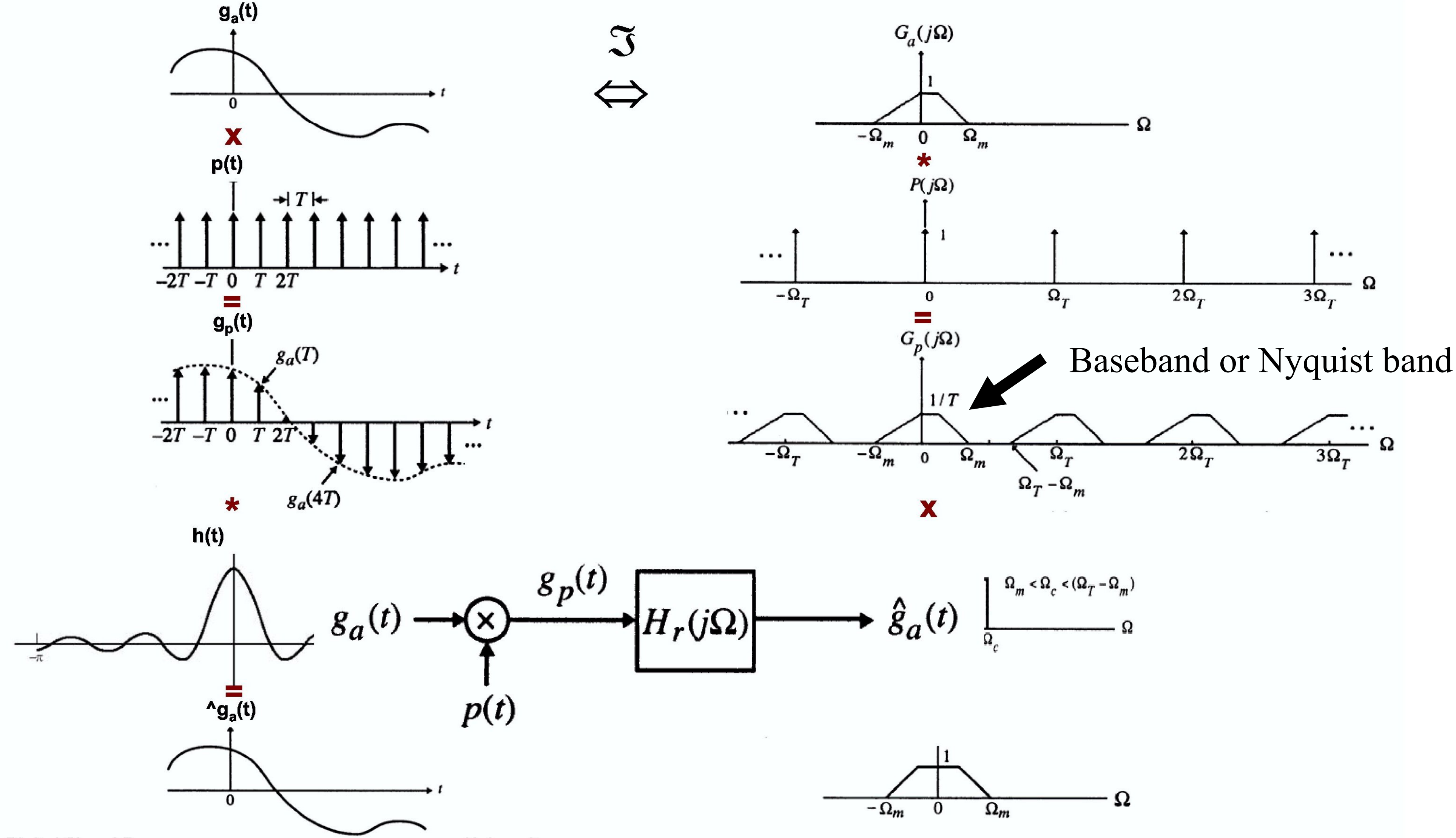
SAMPLING

- the term on the RHS of this equation for $k=0$ is called the ***baseband*** portion of $G_p(j\Omega)$

- $$\boxed{G_p(j\Omega) = \frac{1}{T}\sum_{k=-\infty}^{\infty} G_a(j(\Omega + k\Omega_T))}$$

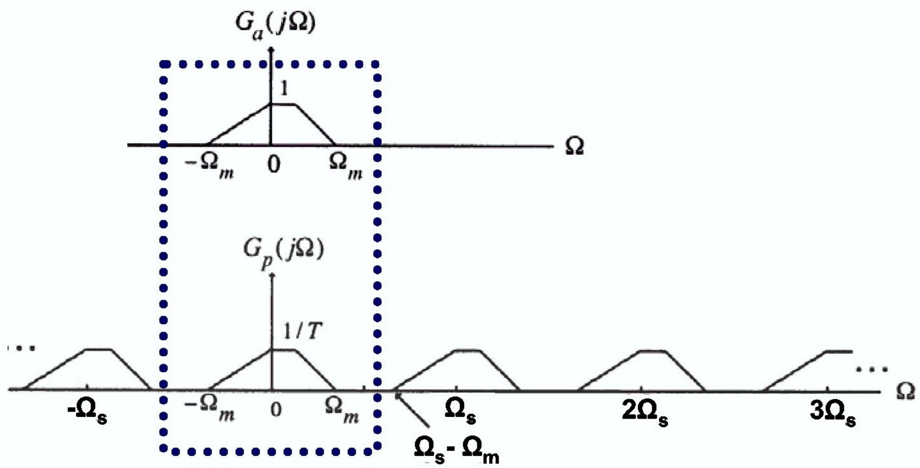
- the frequency range $-\Omega_T/2 \le -\Omega \le -\Omega_T/2$ is called the ***baseband*** or ***Nyquist band***

Sampling Explained... ...Graphically



NYQUIST RATE

- Note that the key requirement for the $G_a(\Omega)$ recovered from $G_p(\Omega)$ is that $G_p(\Omega)$ should consist of non-overlapping replicas of $G_a(\Omega)$.



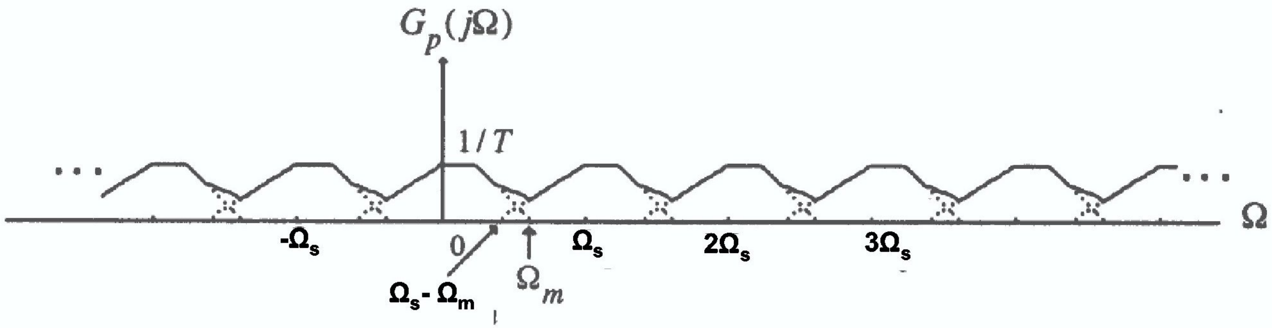
- Under what conditions would this be satisfied…?

?

If $\Omega_s \geq 2\Omega_m$, $g_a(t)$ can be recovered exactly from $g_p(t)$ by passing it through an ideal lowpass filter $H_r(\Omega)$ with a gain T_s and a cutoff frequency Ω_c greater than Ω_m and less than $\Omega_s - \Omega_m$. For simplicity, a half-band ideal filter is typically used in exercises.

ALIASING - REVISITED

- On the other hand, if $\Omega_s < 2\Omega_m$, due to the overlap of the shifted replicas of $G_a(\Omega)$ in the spectrum of $G_p(\Omega)$, the signal cannot be recovered by filtering.
  - This is simply because the filtering of overlapped sections will cause a distortion by folding, or ***aliasing***, the areas immediately outside the baseband back into the baseband.



  - The frequency $\Omega_s/2$ is known as the ***folding frequency***.

A Summary

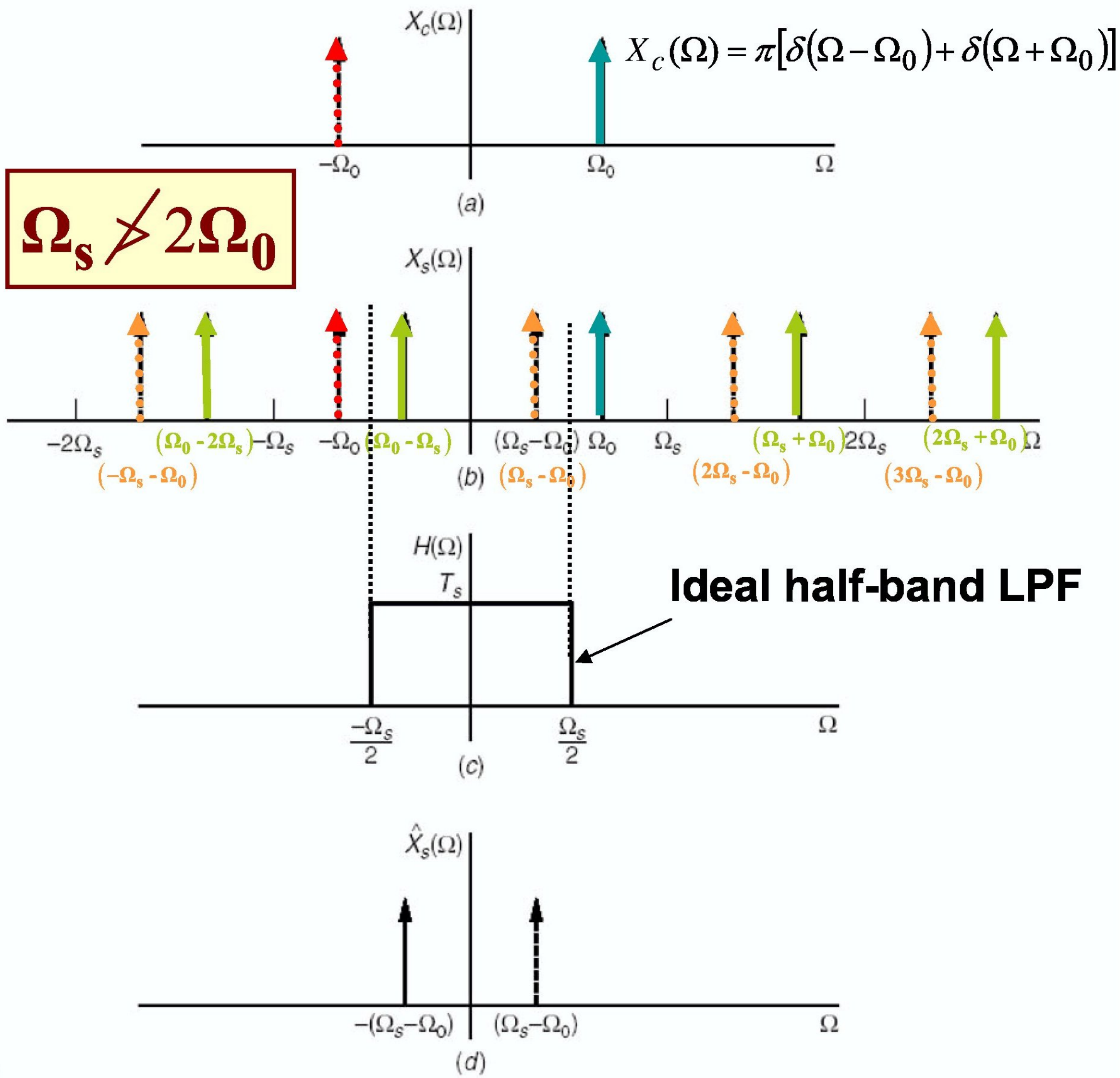
- Given the discrete samples $g_a(nT_s)$, we can recover $g_a(t)$ *exactly* by generating the impulse train

$$g_p(t) = \sum_{n=-\infty}^{\infty} g_a(nT_s)\delta(t - nT_s)$$

  and then passing it through an ideal lowpass filter $H_r(\Omega)$ with a gain T_s and a cutoff frequency Ω_c satisfying $\Omega_m \leq \Omega_c \leq \Omega_s - \Omega_m$.
- The highest frequency Ω_m contained in $g_a(t)$ is usually called the *Nyquist frequency* since it determines the minimum sampling frequency $\Omega_s = 2\,\Omega_m$ that must be used to fully recover $g_a(t)$ from its sampled version.
  - Sampling over or below the Nyquist rate is called ***oversampling*** or ***undersampling***, respectively. Sampling exactly at this rate is ***critical sampling***. A pure sinusoid may not be recoverable from critical sampling. Some amount of oversampling is usually used to allow some tolerance.
  - E.g. in phone conversations, 3.4 kHz is assumed to be the highest frequency in the speech signal, and hence the signal is sampled at 8 kHz.
  - In digital audio applications, the full range of audio frequencies of 0 ~ 20kHz is preserved. Hence, in CD audio, the signal is sampled at 44.1kHz.

More on Aliasing

- To see the full effect of aliasing, as well as to get an insight to the story behind the word "aliasing" consider the sinusoid $x_c(t)=\cos\Omega_0 t$.
- Note that the filtered signal has its spectral components at $\pm(\Omega_s-\Omega_0)$, rather then Ω_0.
- The reconstructed signal is then $\cos(\Omega_s-\Omega_0)t$, not $\cos\Omega_0 t$.
  - This is because frequencies beyond $\Omega_s/2$ have folded into the baseband area. Hence we get an ***alias*** of original signal.
  - E.g. A sinusoid at 5kHz, sampled at 8kHz would appear as 3kHz when reconstructed!

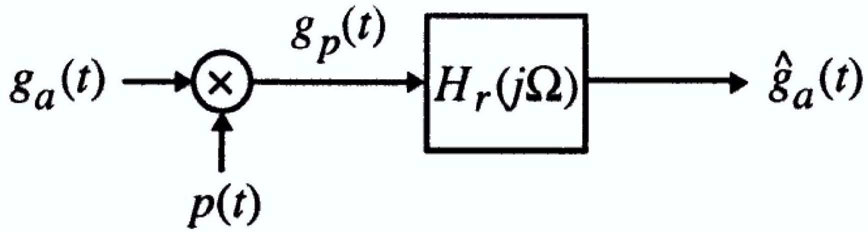


SUMMARY

- Sampling theorem – MUST sample at least twice highest frequency in the signal!
- Aliasing will occur otherwise.
- If sampled at the appropriate rate, continuous time signal can be recovered – *exactly* – from its samples using low pass filtering.

Recovering the Original Signal

- We saw in *frequency domain* that the original signal $g_a(t)$ can be recovered from its sampled version after lowpass filtering.
- One may ask the question: *How does lowpass filtering uniquely "fill-in" the spaces in between the discrete samples?*
- To find out, we need to convolve the impulse response of the low pass filter $h_r[n]$ with the sampled sequence, impulse train $g_p(t)$



Recovering the Original Signal

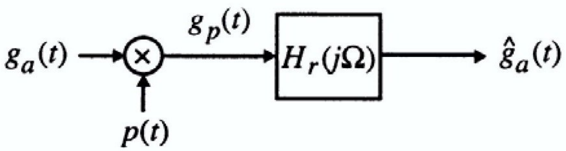
- The impulse response $h_r(t)$ of the lowpass reconstruction filter is obtained by taking the inverse DTFT of $H_r(\Omega)$:

$$H_r(\Omega) = \begin{cases} T, & |\Omega| \le \Omega_c \\ 0, & |\Omega| > \Omega_c \end{cases}$$

$$\boxed{h_r(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} H_r(\Omega) e^{j\Omega t} d\Omega = \frac{T}{2\pi}\int_{-\Omega_c}^{\Omega_c} e^{j\Omega t} d\Omega = \frac{\sin(\Omega_c t)}{\Omega_c t/2} \quad , -\infty < t < \infty}$$

- The input to the LPF is the impulse train $g_p(t)$

$$\boxed{g_p(t) = \sum_{n=-\infty}^{\infty} g[n]\delta(t - nT)}$$



- Therefore, the output in time domain is the convolution

$$\boxed{\hat{g}_a(t) = \sum_{n=-\infty}^{\infty} g[n] h_r(t - nT)}$$

Recovering the Original Signal

- Substituting $h_r(t) = \sin(\Omega_c t)/(\Omega_s t/2)$ in the above and assuming for simplicity $\Omega_c = \Omega_s/2 = \pi/T_s$ (that is assuming ideal half band filter), we get (do the math at home!!!)

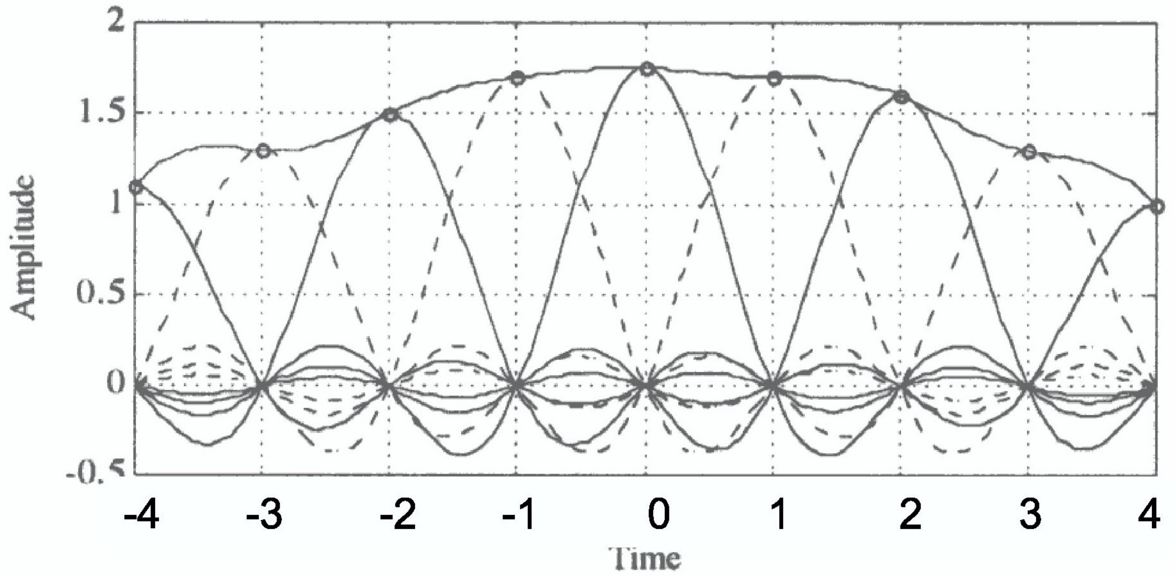
$$\boxed{\hat{g}_a(t) = \sum_{n=-\infty}^{\infty} g[n] \frac{\sin[\pi(t - nT_s)/T_s]}{\pi(t - nT_s)/T_s}}$$

Shifted versions of impulse function $h_r(t)$

- What does this mean???
  - Now recall that the impulse response of the filter is a sinc function
  - The sampled signal is a series of impulses
  - The convolution of any signal with a series of impulses can be obtained by sifting the signal to each impulse location and summing up all shifted versions of the signal

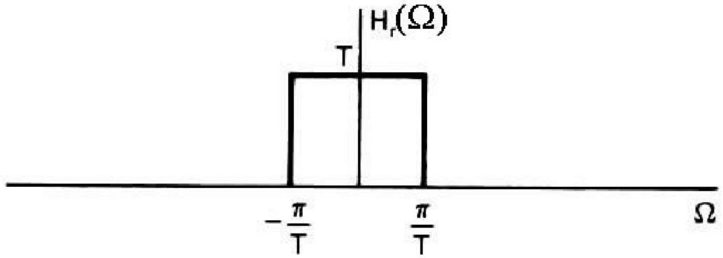
Recovering the Original Signal

- Graphically:
  - Observe that $\boldsymbol{h_r(0)=1}$ and $\boldsymbol{h_r(t\text{-}nT_s)=0}$ for all n≠0. Since at n=0, the $h_r(\boldsymbol{t\text{-}nT_s})$ is normalized by $g[n]$, we do obtain $g[0]$ at n=0. The contribution to $g[0]$ from all other $h_r(\boldsymbol{t\text{-}nT_s})$ at n=0 is zero.
  - The same can be said for all other time points of $g[n]$: For any n, only one of the shifted $h_r(\boldsymbol{t\text{-}nT_s})$ contributes at that time n, all others are zero.
  - Thus the ideal lowpass filter fills-in between the samples by interpolating using the sinc function.

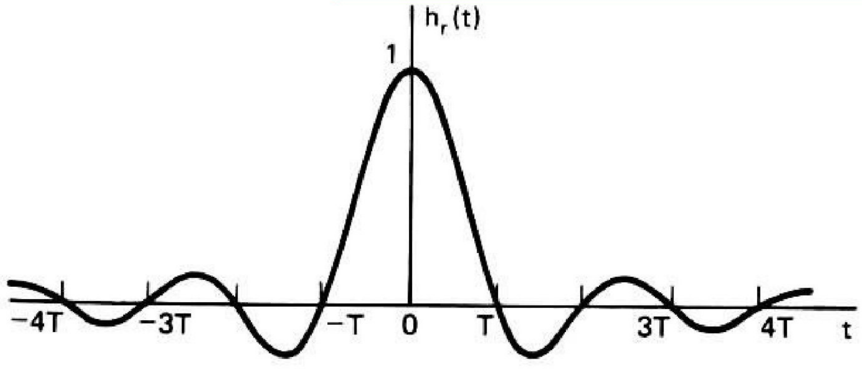


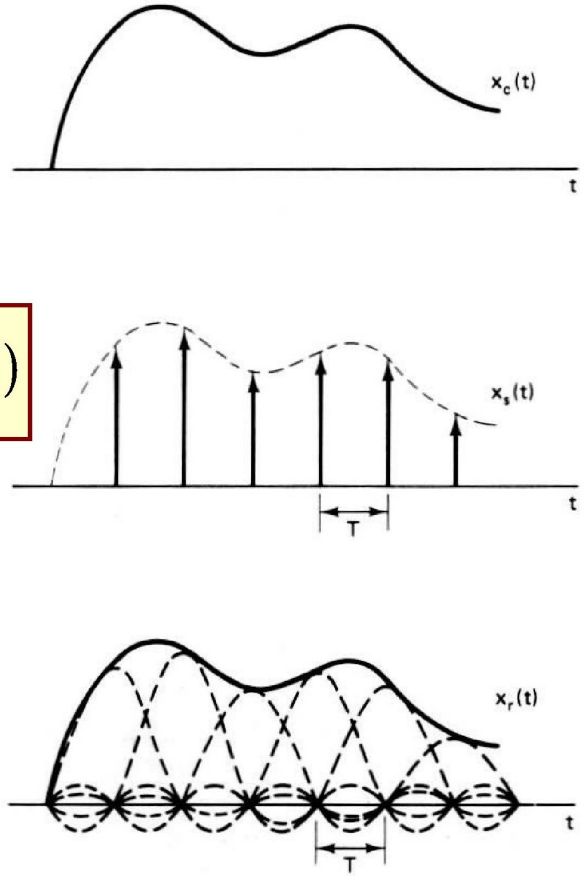
Some Key points to Remember

- The lowpass filter is essentially doing a “sinc” interpolation.
- Sinc function in Matlab computes sin(pi*x)/(pi*x)
  - The sinc function and the rectangular function are Fourier transform pairs. Therefore, the impulse response of an ideal lowpass filter is a ***sinc*** function



$$\boxed{\frac{\sin[\pi(t-nT_s)/T_s]}{\pi(t-nT_s)/T_s} = \mathrm{sinc}\big((t-nT_s)/T_s\big)}$$





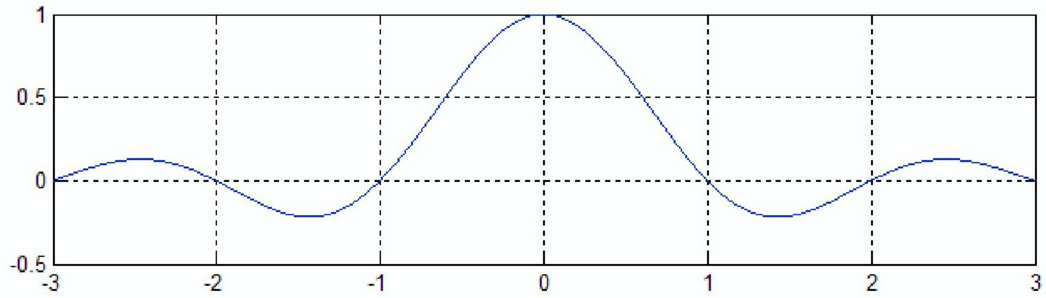
FINAL WORDS

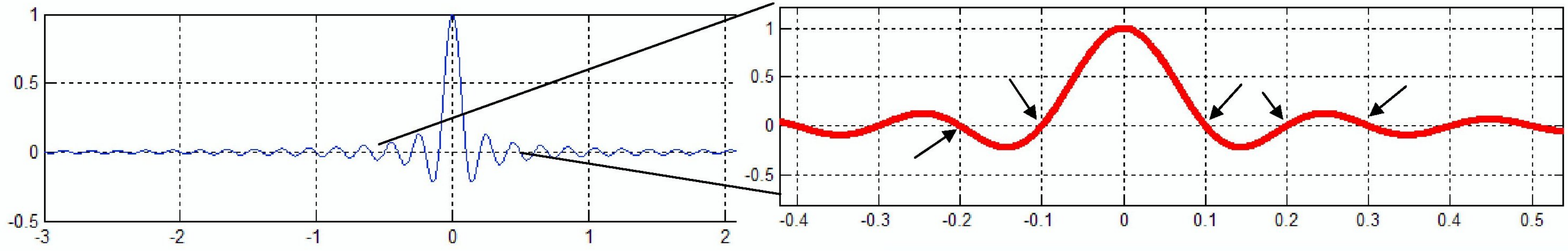
- Note that the ideal lowpass filter is infinitely long, and therefore is not realizable. A non-ideal filter will not have all the zero crossings that ensure perfect reconstruction.
- Furthermore, if a signal is not-bandlimited, it cannot be reconstructed whatever the sampling frequency is.
  - Therefore, an ***anti-aliasing filter*** is typically used before the sampling to limit the highest frequency of the signal, so that a suitable sampling frequency can be used, and so that the signal can be reconcstructed if with a non-ideal low pass filter!
- Sampling and anti-aliasing filters are covered in Chapter 4 of your text (page 209-211)

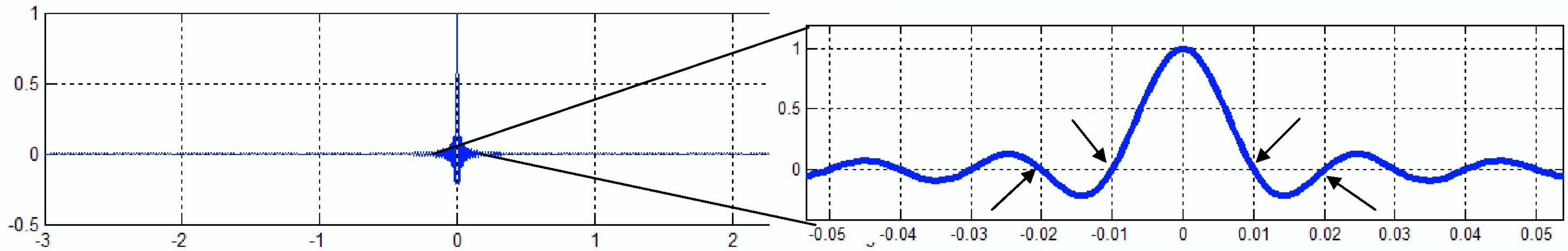
About the Sinc Function

```
t=-3:0.001:3;
xa1=sinc(1*t);
xa2=sinc(10*t);
xa3=sinc(100*t');
subplot(311); plot(t,xa1); grid
subplot(312); plot(t,xa2); grid
subplot(313); plot(t,xa3); grid
```

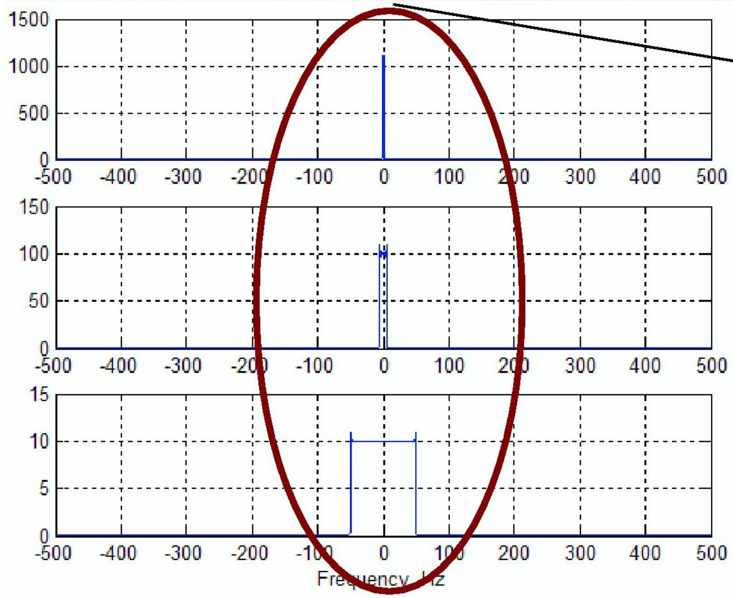
$1/T_s$

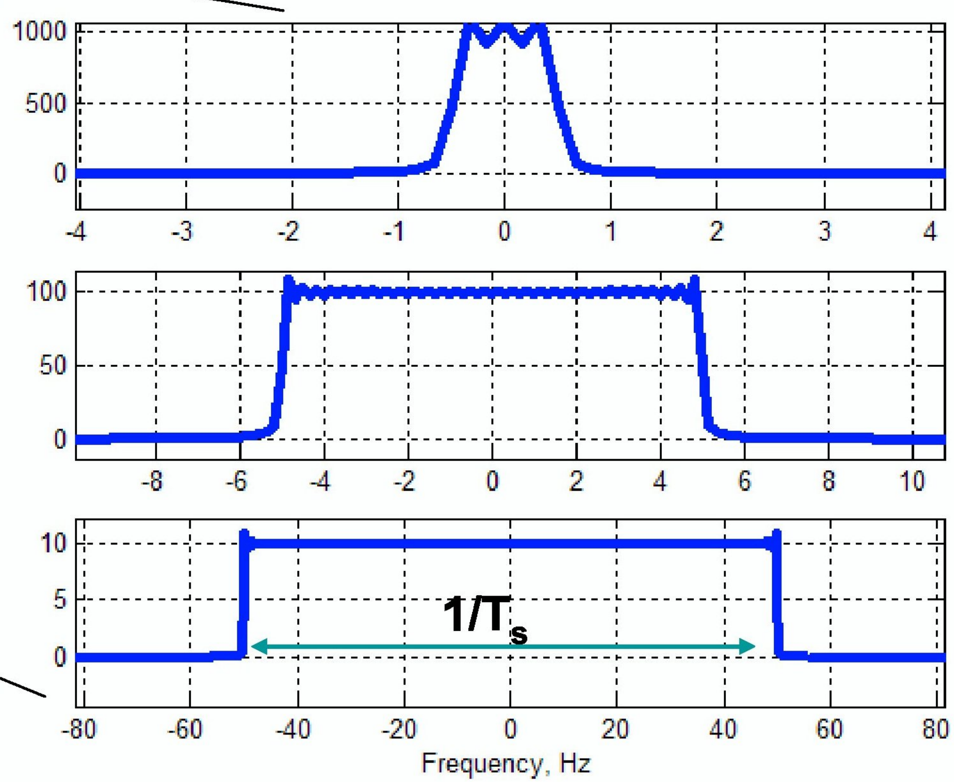






About the Sinc Function






```
XA1=abs(fft(xa1));
XA2=abs(fft(xa2));
XA3=abs(fft(xa3));
Fs=1/0.001;
f=linspace(-Fs/2, Fs/2, length(xa1));
subplot(311)
plot(f,fftshift(XA1)); grid
subplot(312)
plot(f,fftshift(XA2)) ; grid
subplot(313)
plot(f,fftshift(XA3)) ; grid
```

EXAMPLE

- Consider the analog signal $x_a(t)=\cos(20\pi t)$, $0\leq t\leq 1$. It is sampled at T_s=0.01, 0.05, 0.075 and 0.1 second intervals to obtain x[n];
  - For each T_s, plot x[n]
  - Reconstruct the analog signal $y_a(t)$ from the samples of x[n] by means of sinc interpolation (low pass filtering). Use Δt=0.001. Estimate the frequency in $y_a(t)$ from your plot. Ignore the end effects.

  - Try at home with the sinc function. What did you observe?