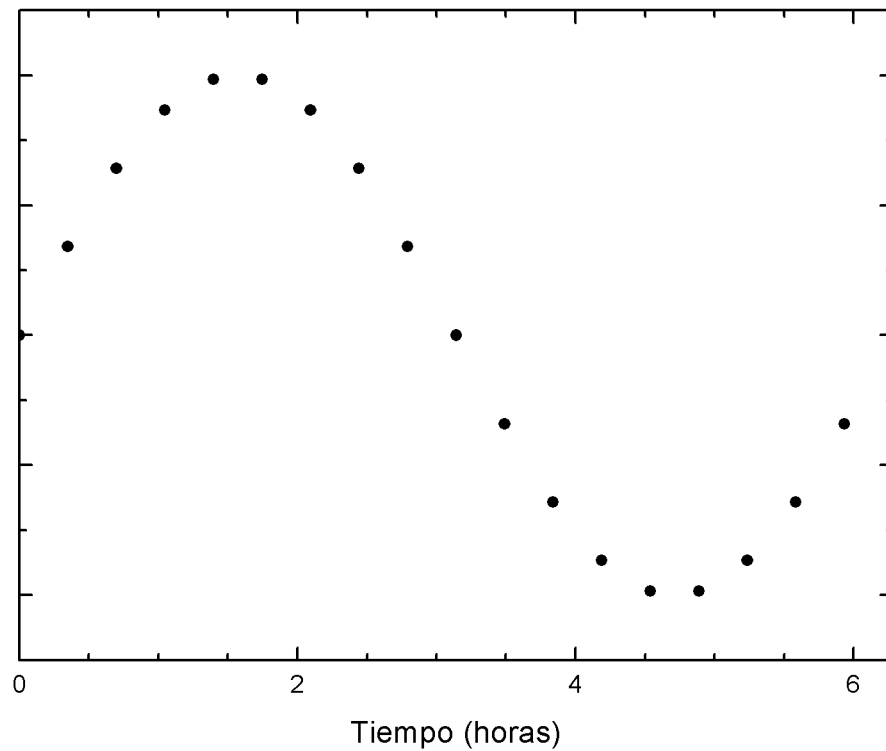


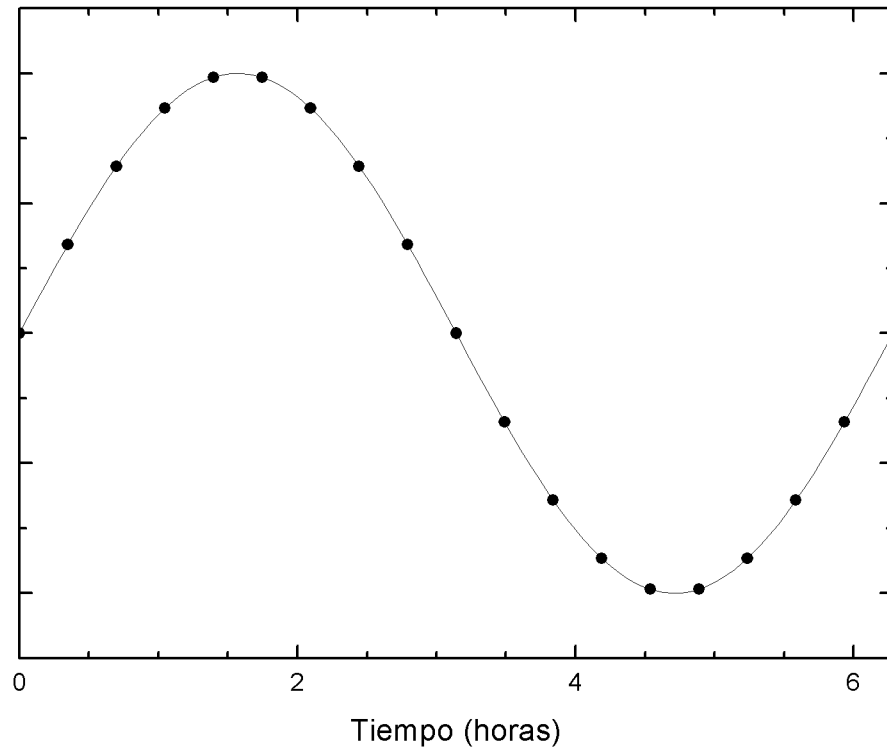
Aliasing

Supongamos que medimos un conjunto de datos cada tiempo Δ



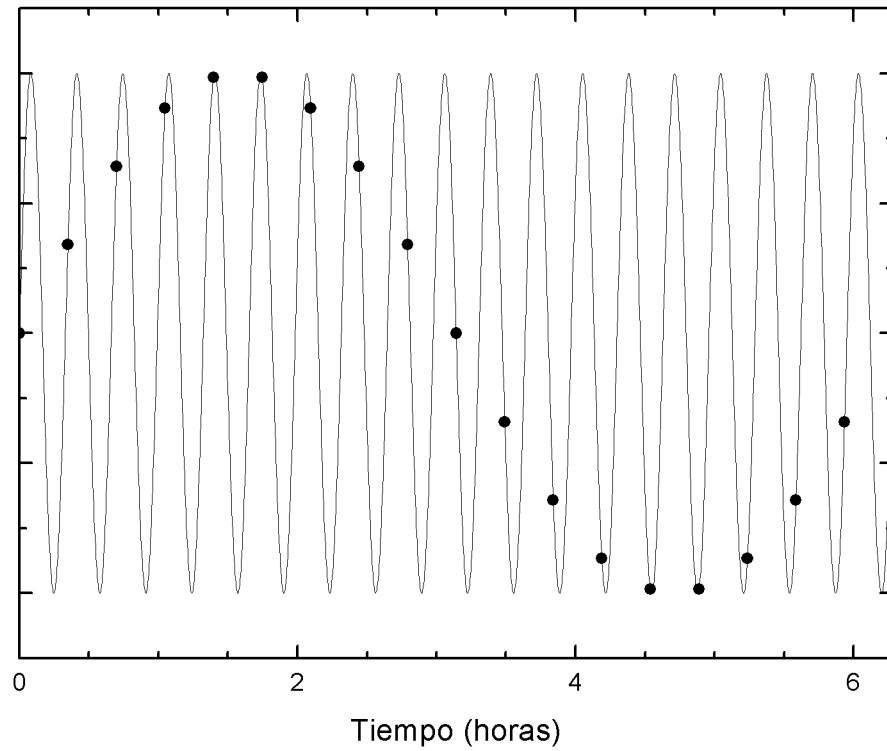
Aliasing

Estos datos están muy bien descritos por una frecuencia f



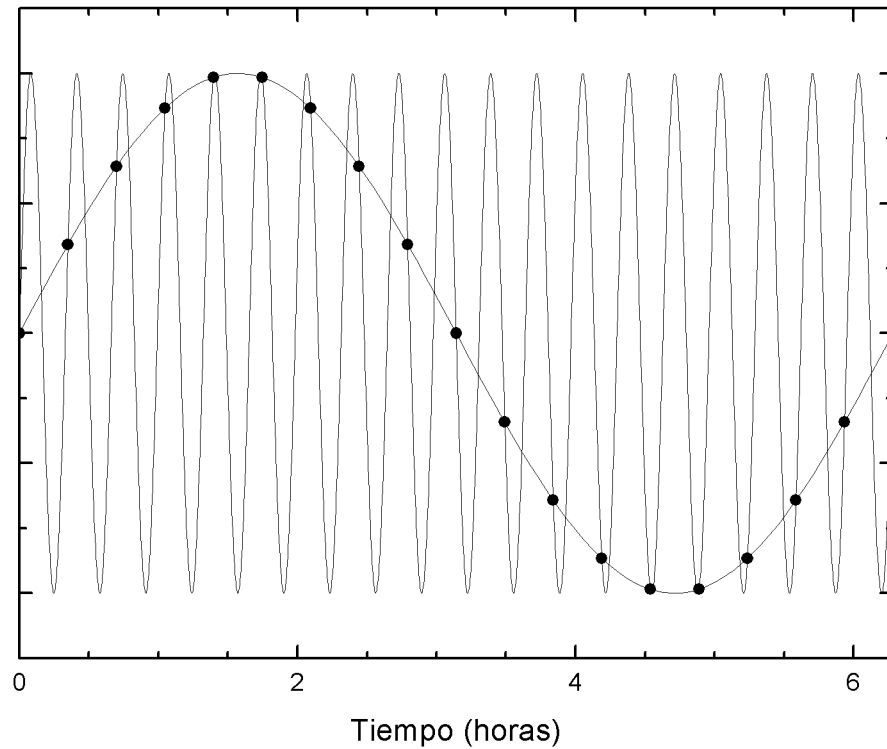
Aliasing

Pero también por una frecuencia $f + \frac{1}{\Delta}$



Aliasing

Si la info que tenemos está medida cada Δ ,
no podemos distinguir entre las frecuencias f y $f + N\frac{1}{\Delta}$



Aliasing

Ejemplos de aliasing en video

Helicóptero (empezar en 1:28)

Hélice de avión

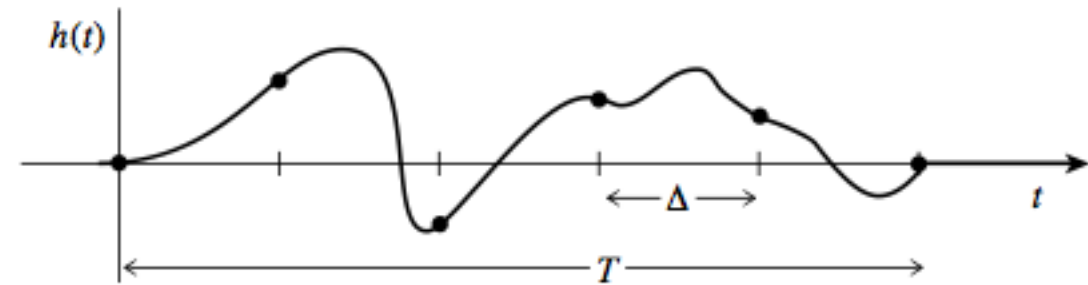
Frecuencia de Nyquist

Sea la frecuencia $f_N = \frac{1}{2\Delta}$

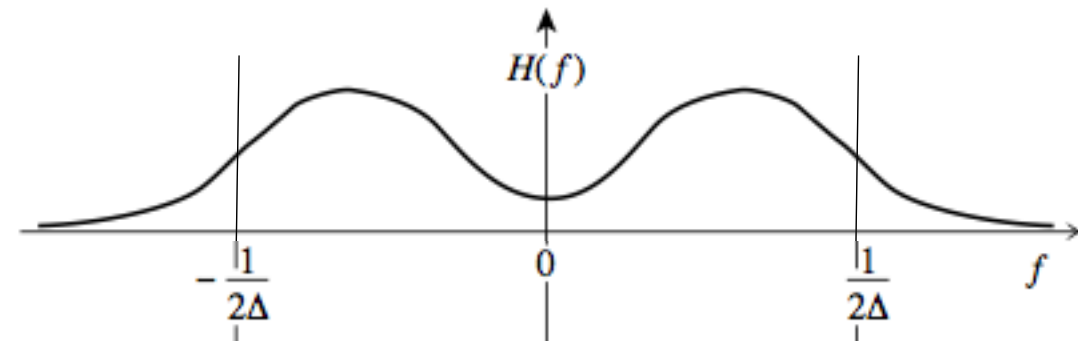
Teorema del muestreo (Sampling theorem): Si una función $h(t)$ tiene un contenido espectral de ancho de banda menor a f_N , entonces al medirla con un muestreo Δ , se obtiene “**toda**” la información de la misma. La función está determinada por los valores medidos h_n a tiempos $N\Delta$, con $-\infty < N < \infty$

$$h(t) = \Delta \sum_{n=-\infty}^{+\infty} h_n \frac{\sin[2\pi f_c(t - n\Delta)]}{\pi(t - n\Delta)}$$

Que pasa si el ancho de banda de $h(t)$ no está limitado por f_N ?



(a)



(b)

En este caso hay
frecuencias por
fuera de $-f_N < f < f_N$

y el aliasing nos
vuelca las f altas
adentro del rango
 $-f_N < f < f_N$

Transformada de Fourier discreta

Si se muestrean N valores de $h(t)$, a tiempos $n\Delta$, con $n = 0, \dots, N-1$, podemos evaluar la transformada de Fourier, $H_n = H(f_n)$, discreta de la forma

$$H_n \equiv \sum_{k=0}^{N-1} h_k e^{2\pi i k n / N}$$

donde $h_k = h(k\Delta)$ y los valores de f_n son $f_n = \frac{n}{N\Delta} \quad -\frac{N}{2} < n \leq \frac{N}{2}$

Es importante darse cuenta que el rango de frecuencias es $2f_N$, y que la ubicación es arbitraria. Podría haberse elegido $0 \leq n \leq N-1$

La transformada inversa es

$$h_k = \frac{1}{N} \sum_{n=0}^{N-1} H_n e^{-2\pi i k n / N}$$

Transformada de Fourier rápida (Fast Fourier Transform – FFT)

Reescribiendo

$$H_n \equiv \sum_{k=0}^{N-1} h_k e^{2\pi i k n / N} \quad W \equiv e^{2\pi i / N} \quad \rightarrow \quad H_n = \sum_{k=0}^{N-1} W^{nk} h_k$$

Esto tiene la pinta de una multiplicación matricial, con $O(N^2)$ multiplicaciones

FFT es un algoritmo para evaluar en forma rápida la transformada de Fourier discreta, que resulta ser $O(N \log_2 N)$

Si $N = 10^6$ entonces la diferencia es 10^{12} contra 2×10^7 !!!!!

Transformada de Fourier rápida (Fast Fourier Transform – FFT)

Separemos la suma en los términos pares y los impares F es lo mismo que H

$$\begin{aligned} F_k &= \sum_{j=0}^{N-1} e^{2\pi i j k / N} f_j \\ &= \sum_{j=0}^{N/2-1} e^{2\pi i k (2j) / N} f_{2j} + \sum_{j=0}^{N/2-1} e^{2\pi i k (2j+1) / N} f_{2j+1} \\ &= \sum_{j=0}^{N/2-1} e^{2\pi i k j / (N/2)} f_{2j} + W^k \sum_{j=0}^{N/2-1} e^{2\pi i k j / (N/2)} f_{2j+1} \\ &= F_k^e + W^k F_k^o \end{aligned}$$

Transformada de Fourier rápida (Fast Fourier Transform – FFT)

$$F_k = F_k^e + W^k F_k^o$$

y seguimos separando cada F_k en la suma de los términos pares y los impares

$$F_k = (F_k^{ee} + W^k F_k^{eo}) + W^k (F_k^{oe} + W^k F_k^{oo})$$

hasta que queda $F_k^{eooeoo.....eoeoe}$ compuesta por un único f_j , para el que $F_k = f_j$

Cual j corresponde a una determinada combinación $eooe.....eoooo$?

Si asignamos $e=0$ y $o=1$, y escribimos la combinación revertida

$$eooe.....eoooo = 0110.....00111 \rightarrow 11100.....0110$$

... nos queda la representación binaria de j !!!!

El algoritmo consiste en :

- 1) Ordenar los f_j por orden binario invertido de j
- 2) Combinar los adjacentes con la formula $f_i + W f_{i+1}$
- 3) Seguir hasta haber combinado todos

OJO!!! Sólo funciona si $N=2^m$