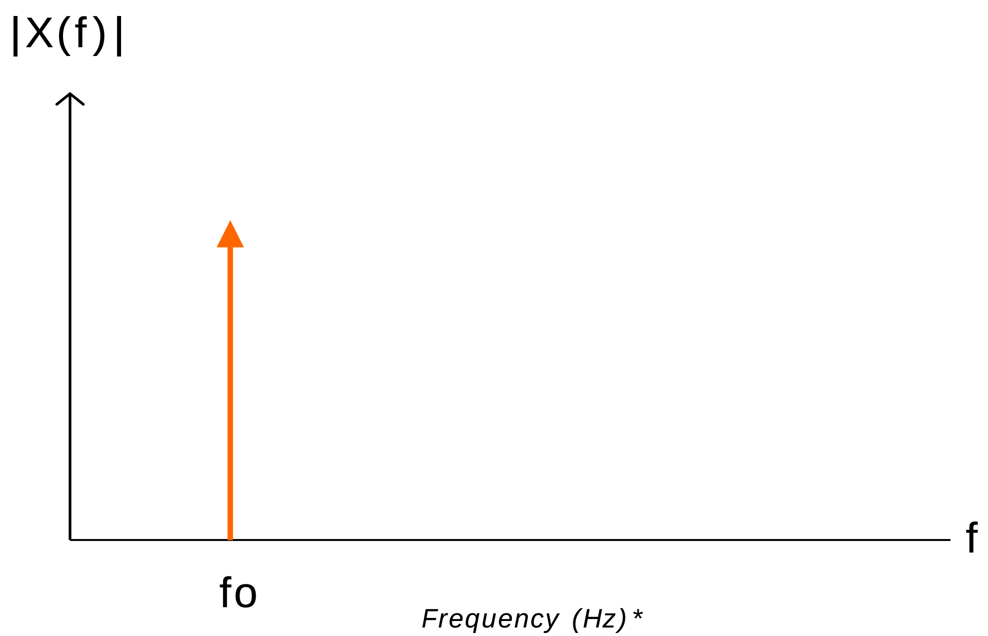
03 Speech Signal Fundamentals II

Deviation from Ideal Spectra

$$x(t) = \cos(2\pi f_0 t)$$

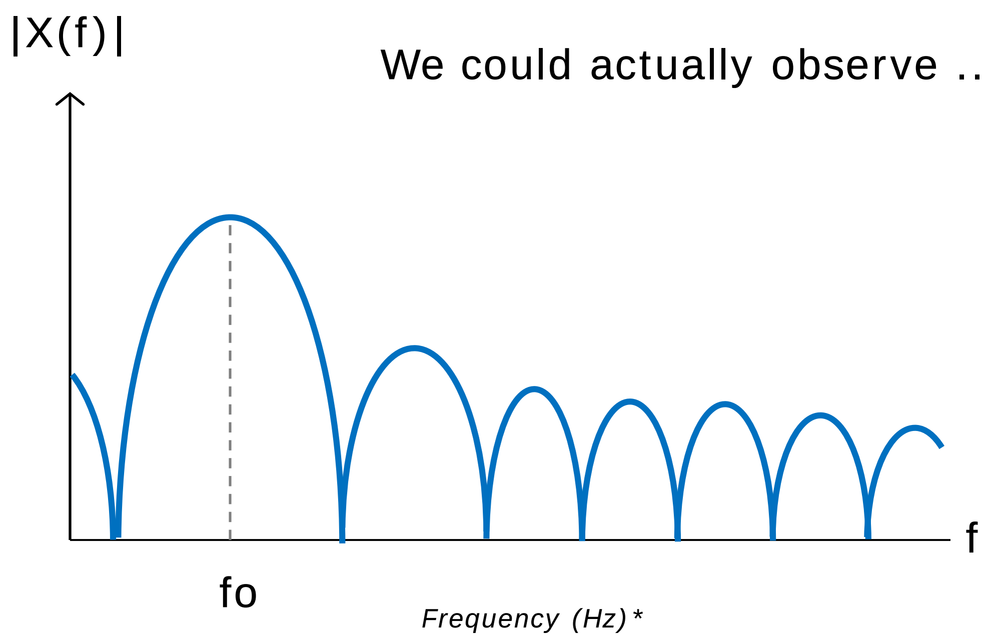
x(t) has infinite length.



* Negative frequencies omitted

Deviation from Ideal Spectra

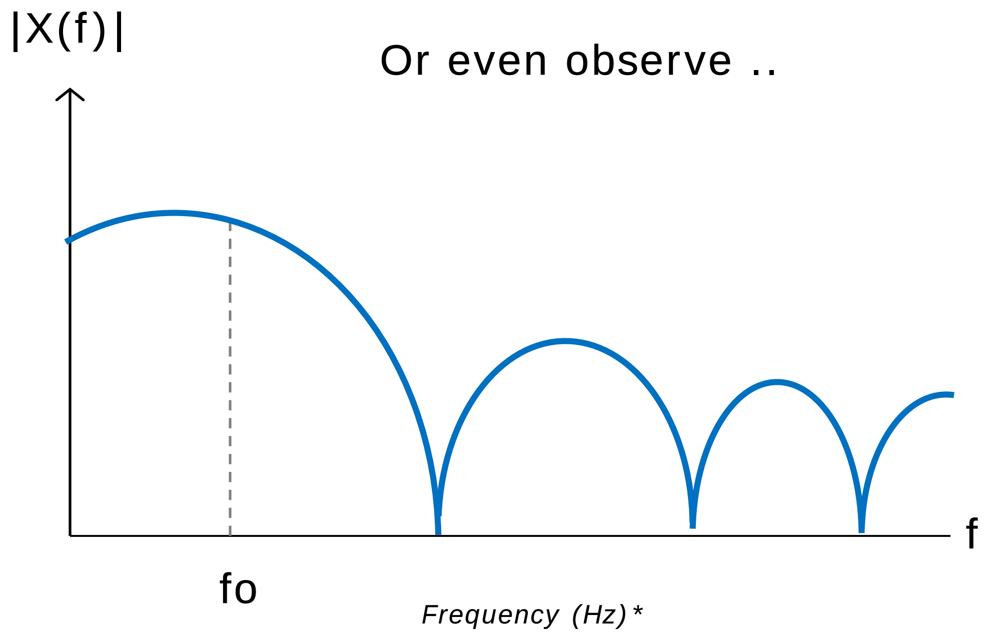
$$x(t) = \cos(2\pi f_0 t)$$



* Negative frequencies omitted

Deviation from Ideal Spectra

$$x(t) = \cos(2\pi f_0 t)$$



* Negative frequencies omitted


Photo by Paul on Unsplash

Finite-length = Time-windowed

$$x(t) = \cos(2\pi f_0 t)$$ → x(t) has infinite length.

But in real life, we can never have a signal of infinite length.

A portion of sine wave with finite length can be considered as a windowed version of the infinite length sine wave.

E.g.: A sine wave from t_1 to t_2, $x_w(t)$ can be written as:

$$\boxed{x_w(t) = \cos(2\pi f_0 t) w(t)}$$

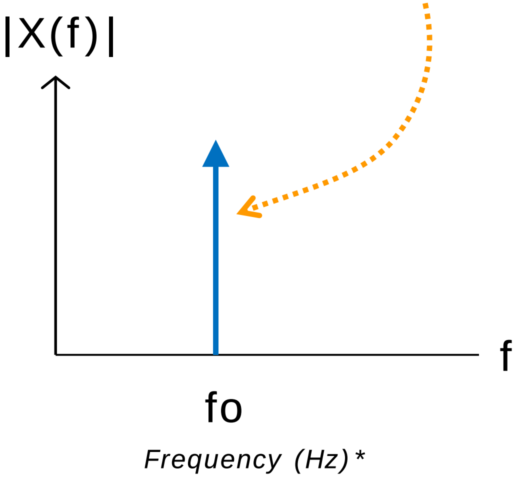
Where:

$$w(t) = \begin{cases} 1 & ; t_1 \le t < t_2 \\ 0 & ; otherwise \end{cases}$$

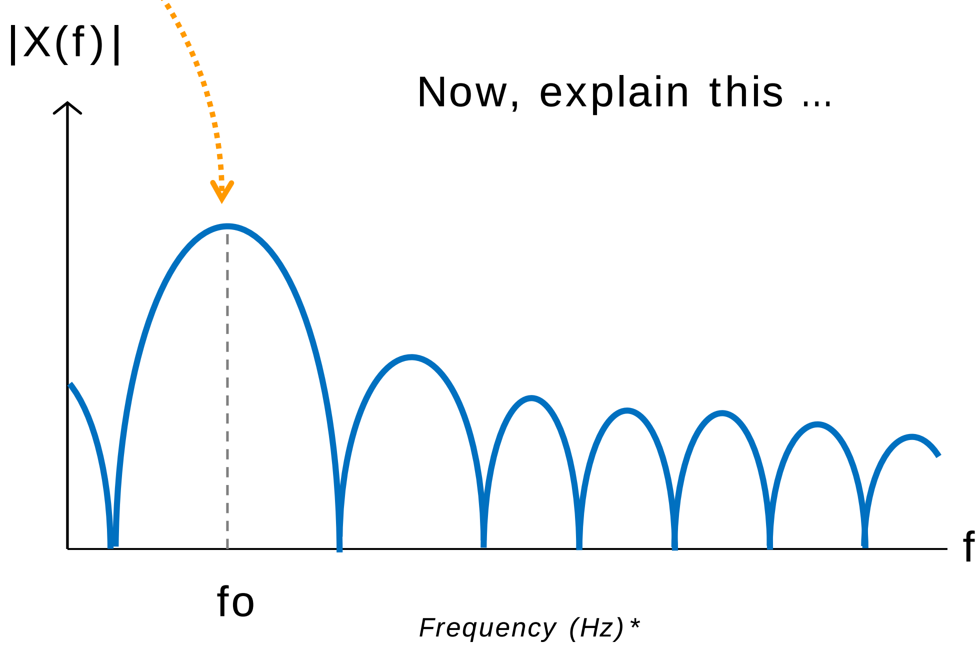
Recall that ...

$$x(t)y(t) \Leftrightarrow X(f) * Y(f)$$

$$x_w(t) = \cos(2\pi f_0 t)w(t)$$

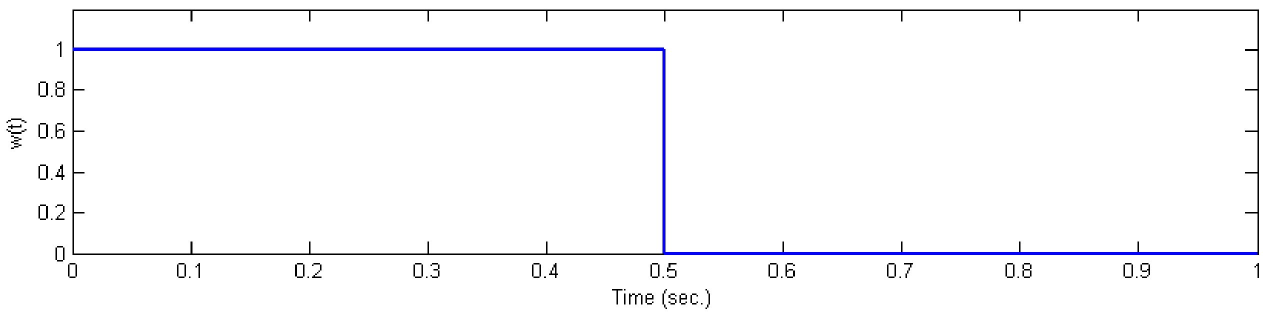


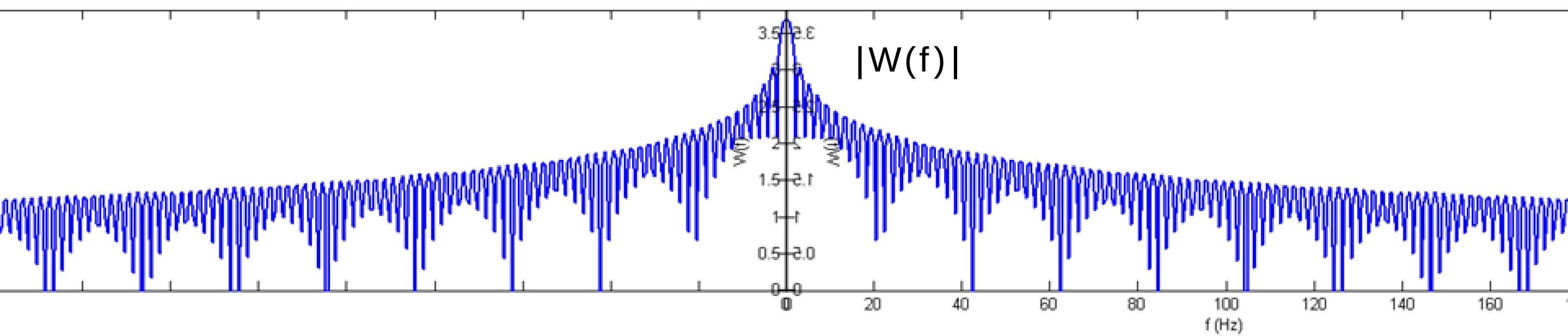
Now, explain this ...



Square Window

$$w(t) = \begin{cases} 1 & ; 0 \le t < 0.5\,\text{sec.} \\ 0 & ; otherwise \end{cases}$$

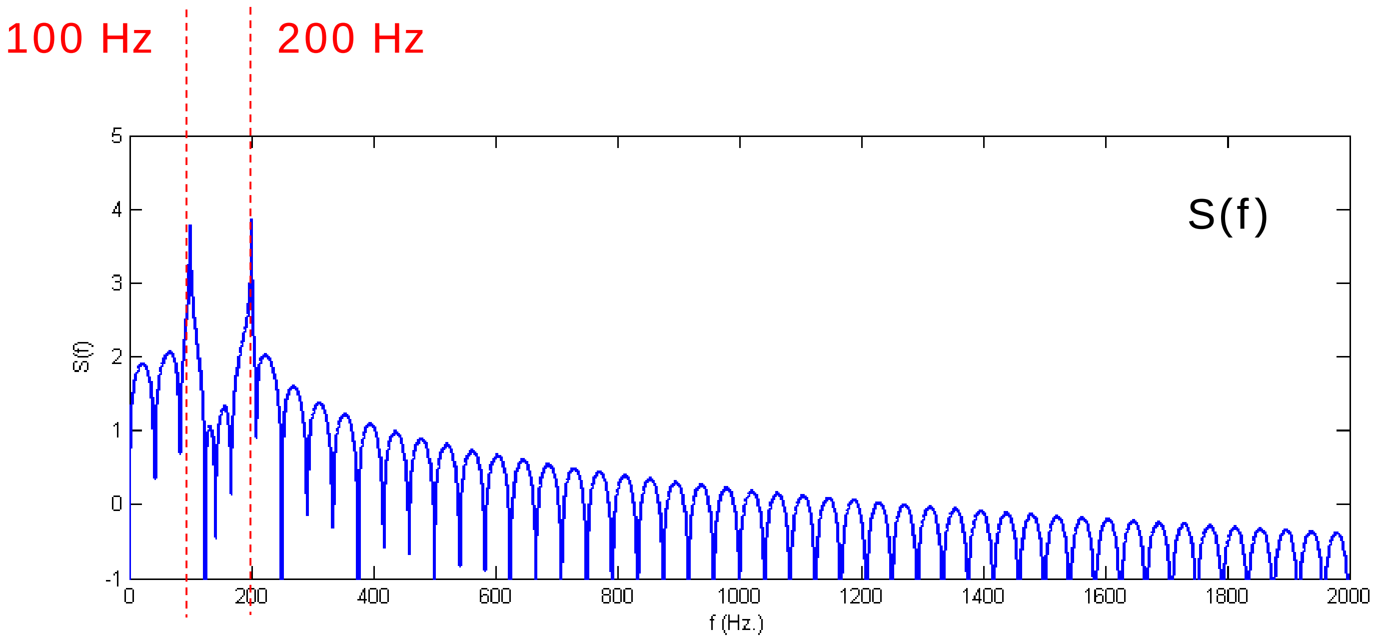




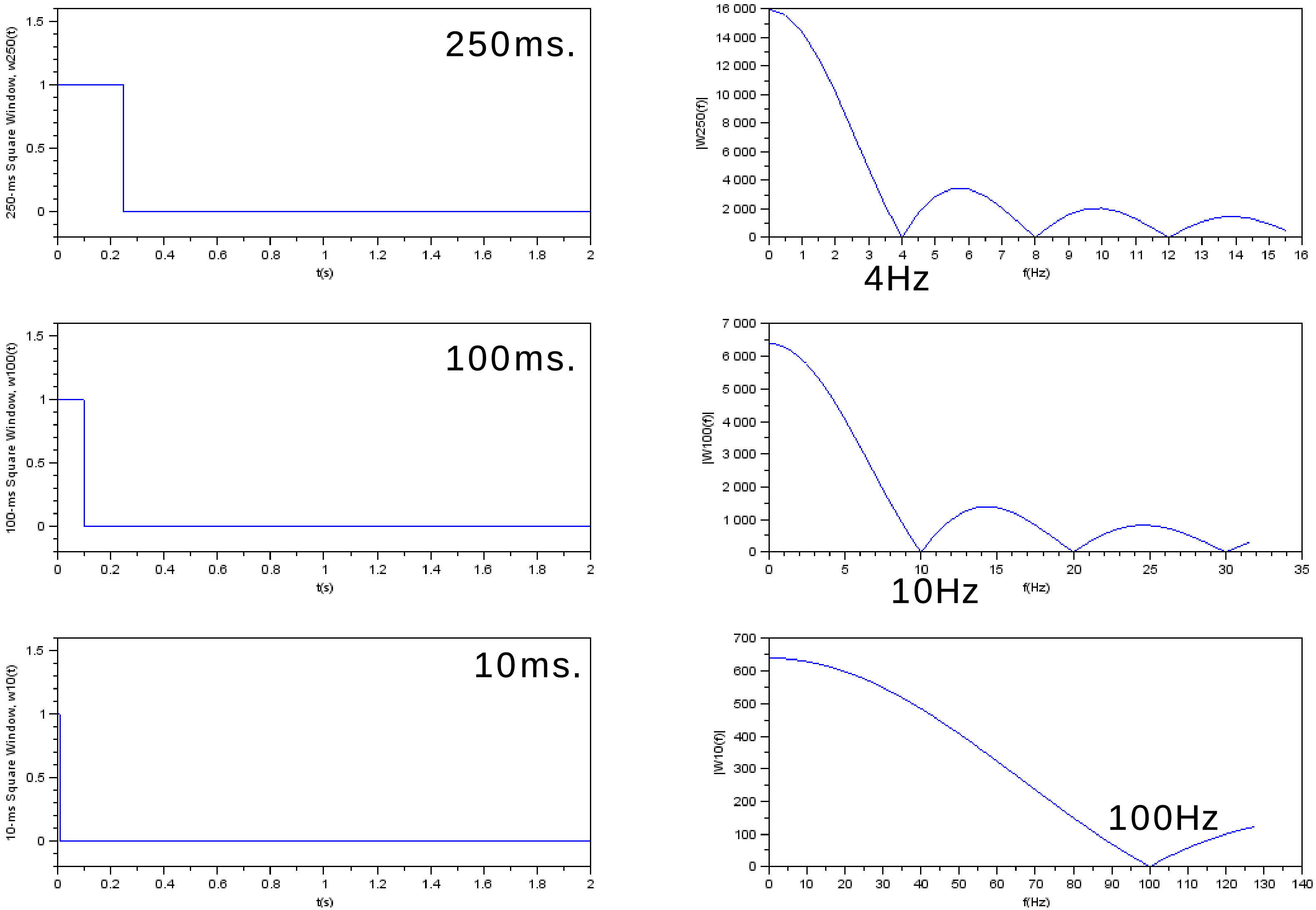
Effect of Time-windowing

$$s(t) = (\sin(2\pi(100)t) + \sin(2\pi(200)t))w(t)$$

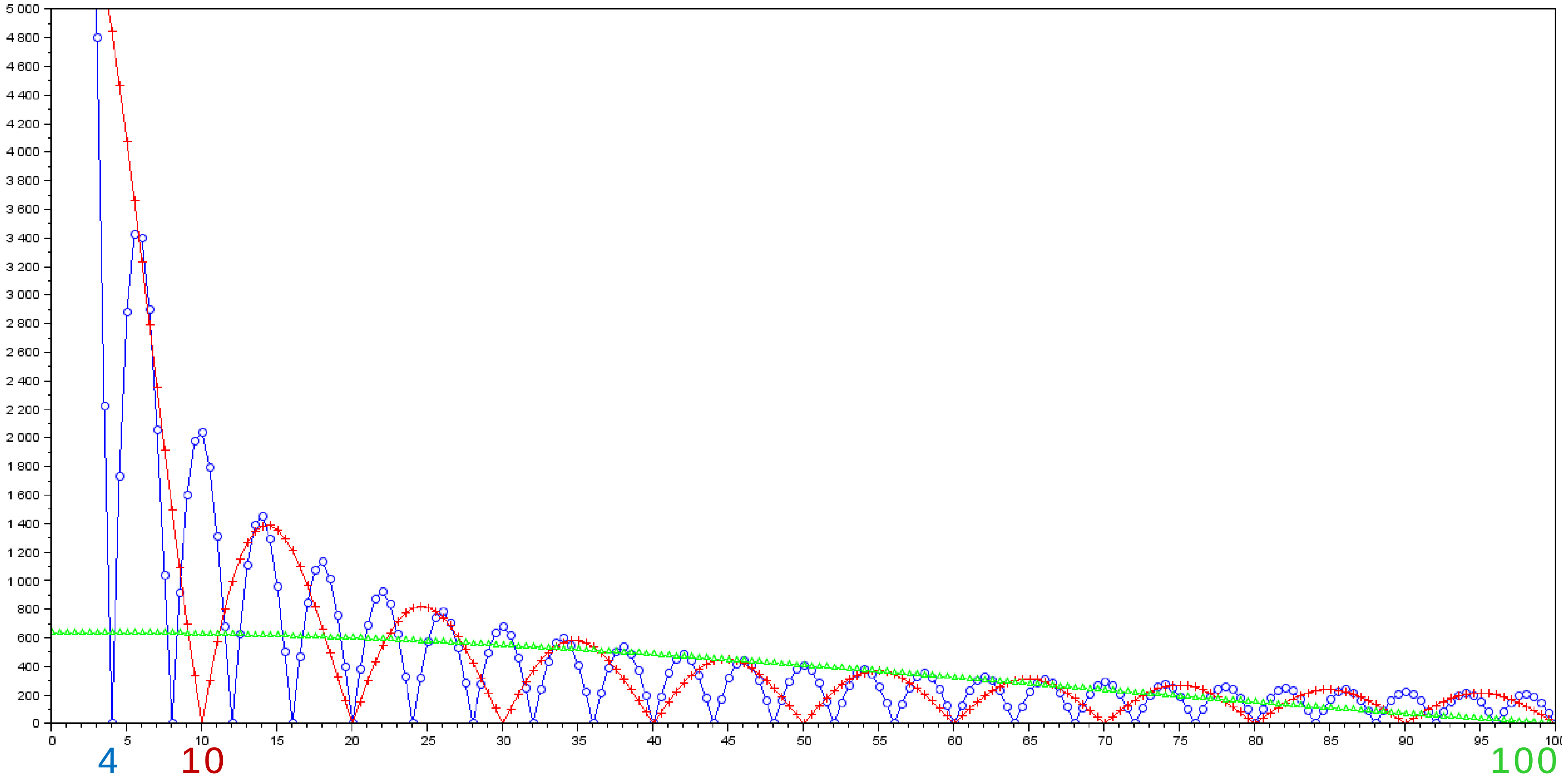
w(t) is a 1-second-long square window



Window Length Vs. Frequency Resolution

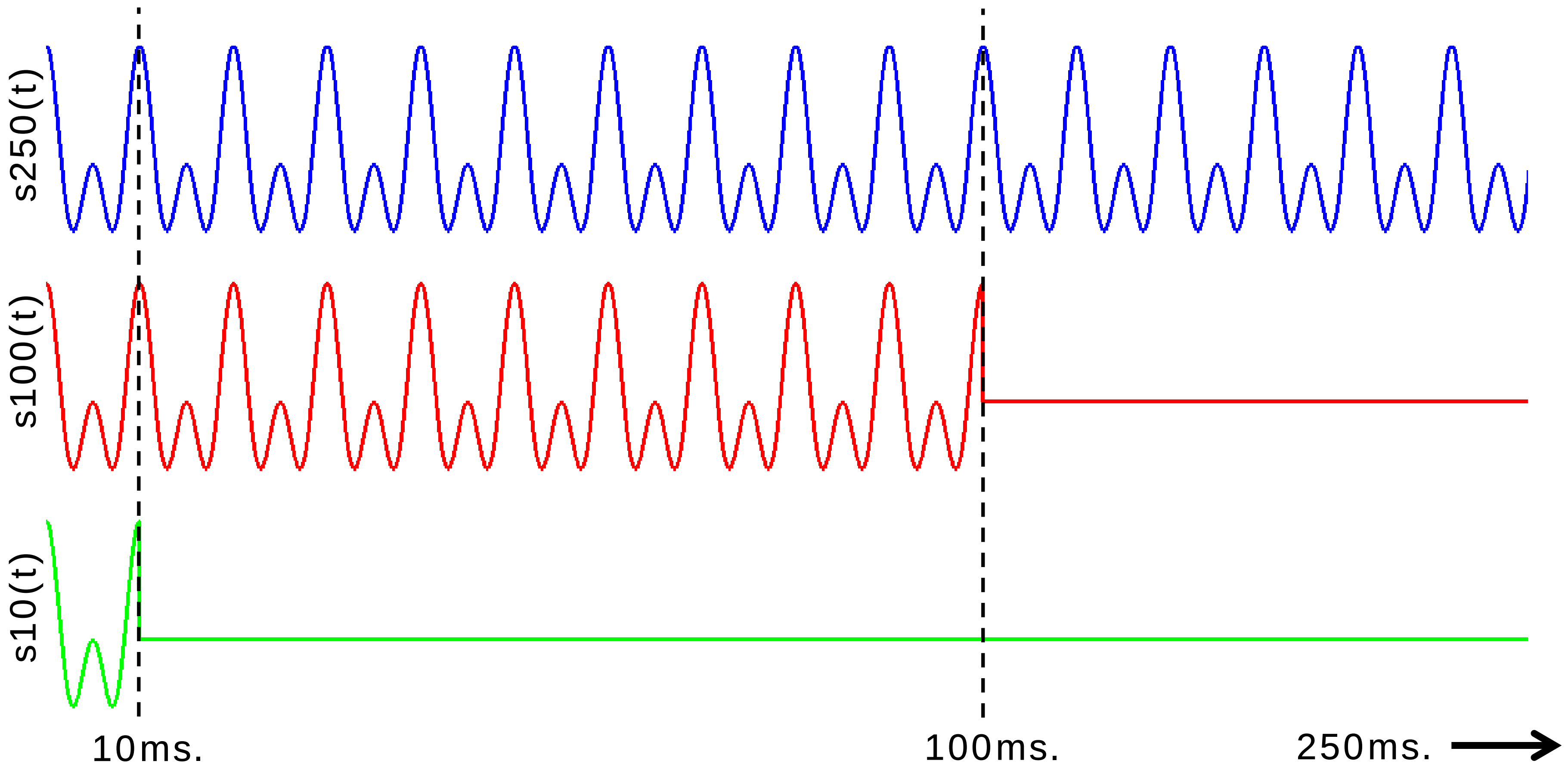


Window Length Vs. Frequency Resolution

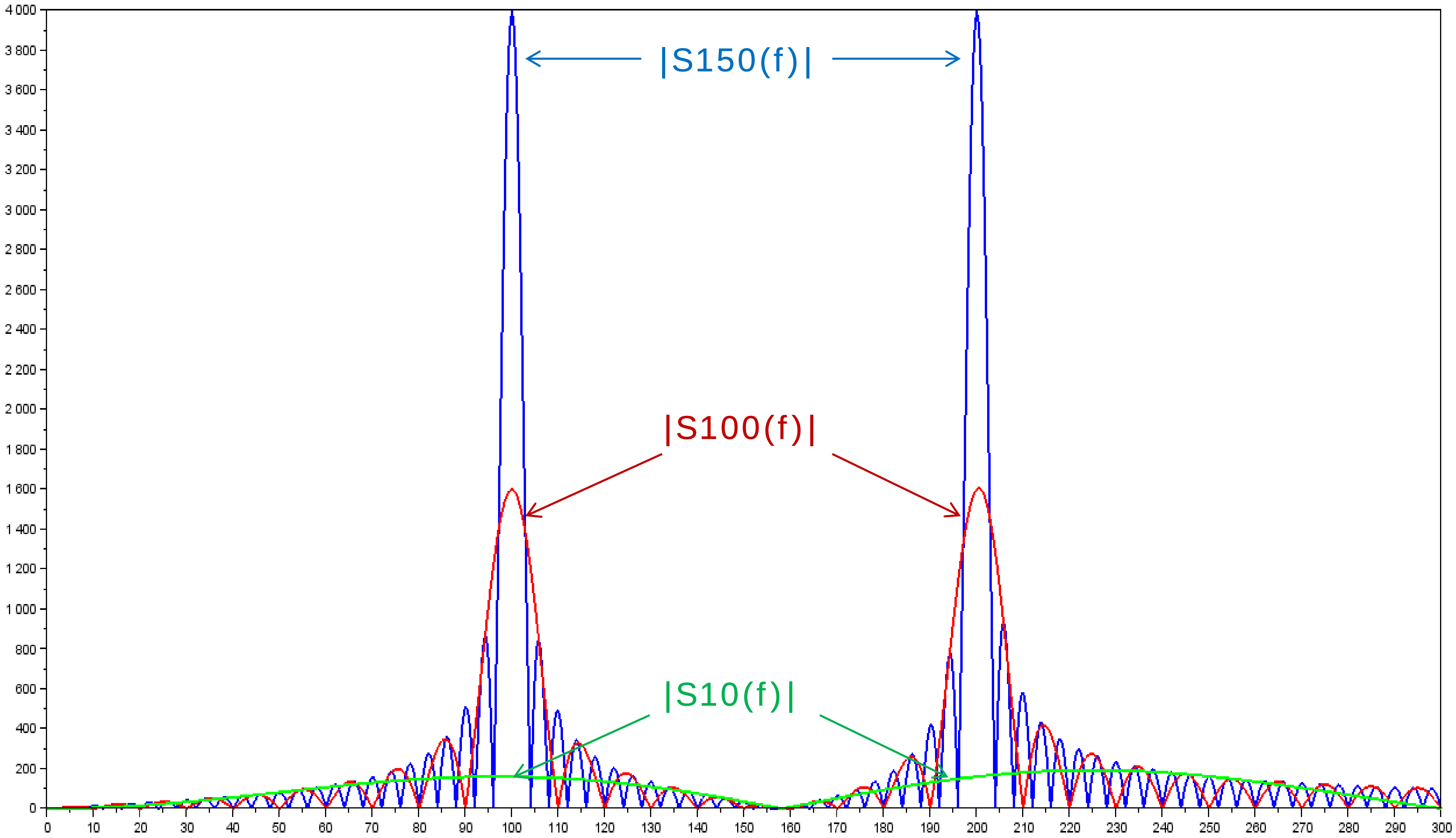


Window Length Vs. Frequency Resolution

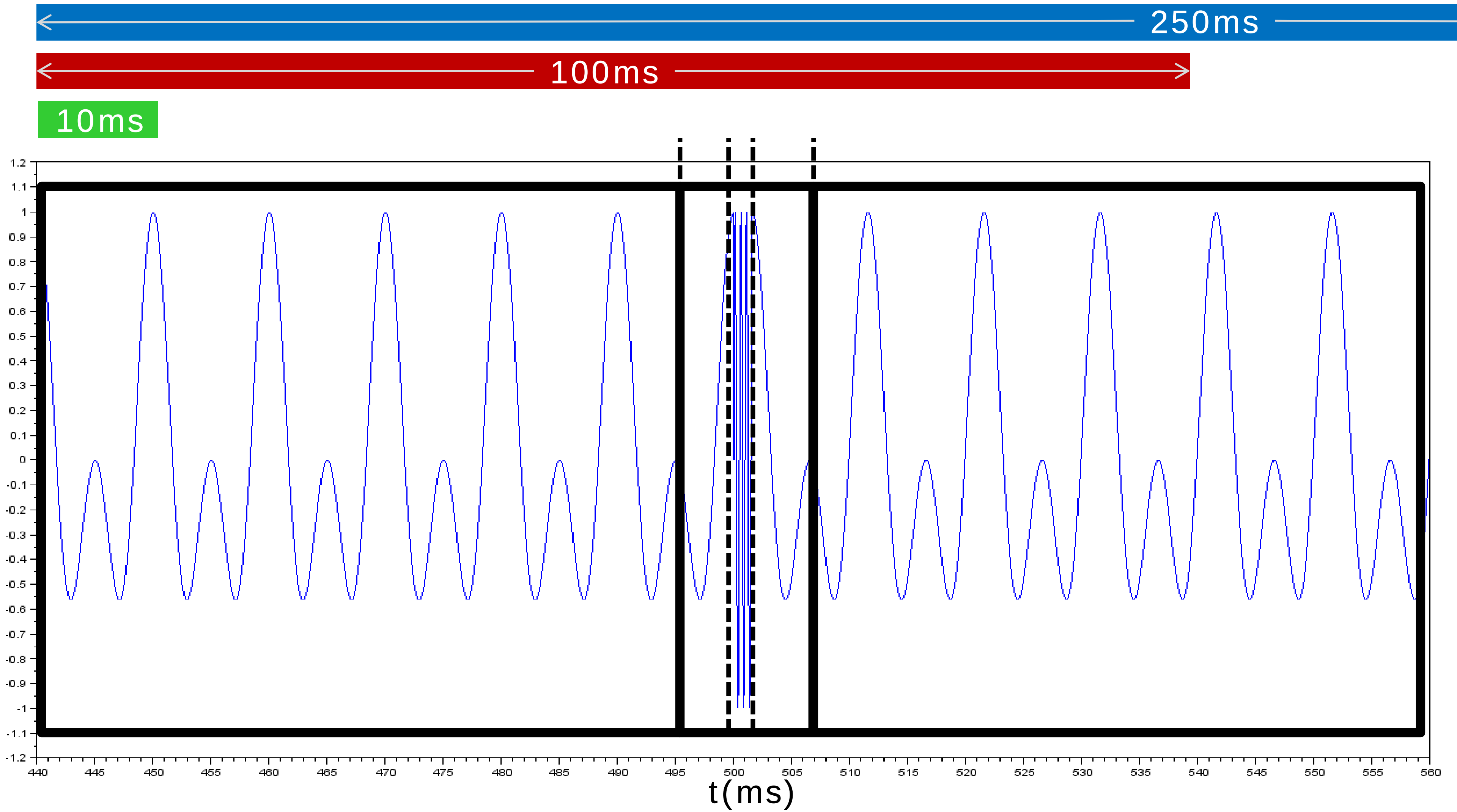
$$s(t) = (\sin(2\pi(100)t) + \sin(2\pi(200)t))w(t)$$



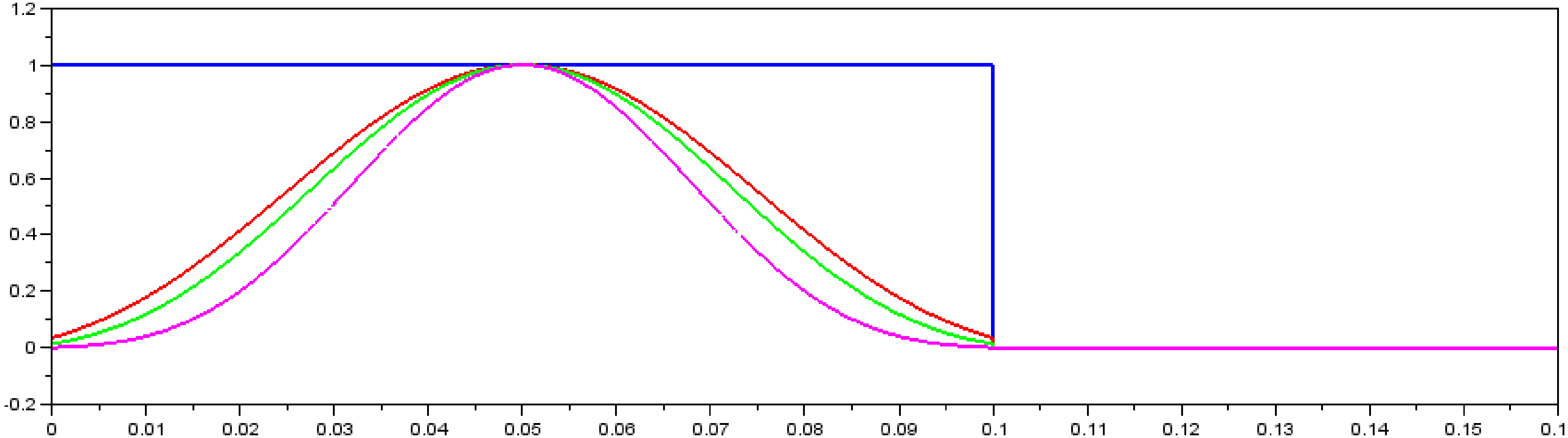
Window Length Vs. Freq.Resolution

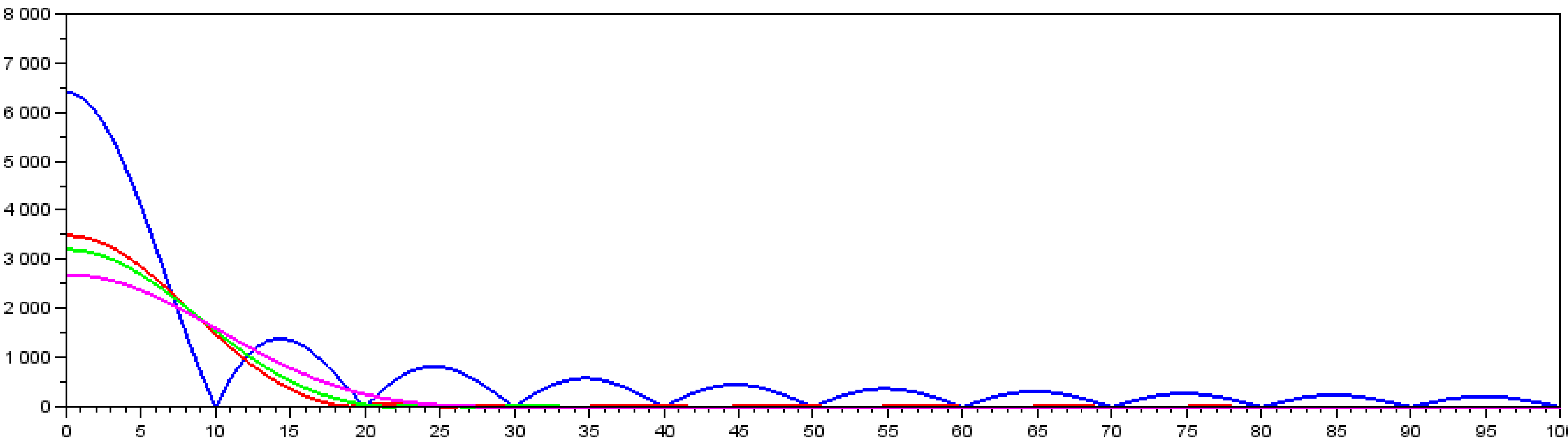


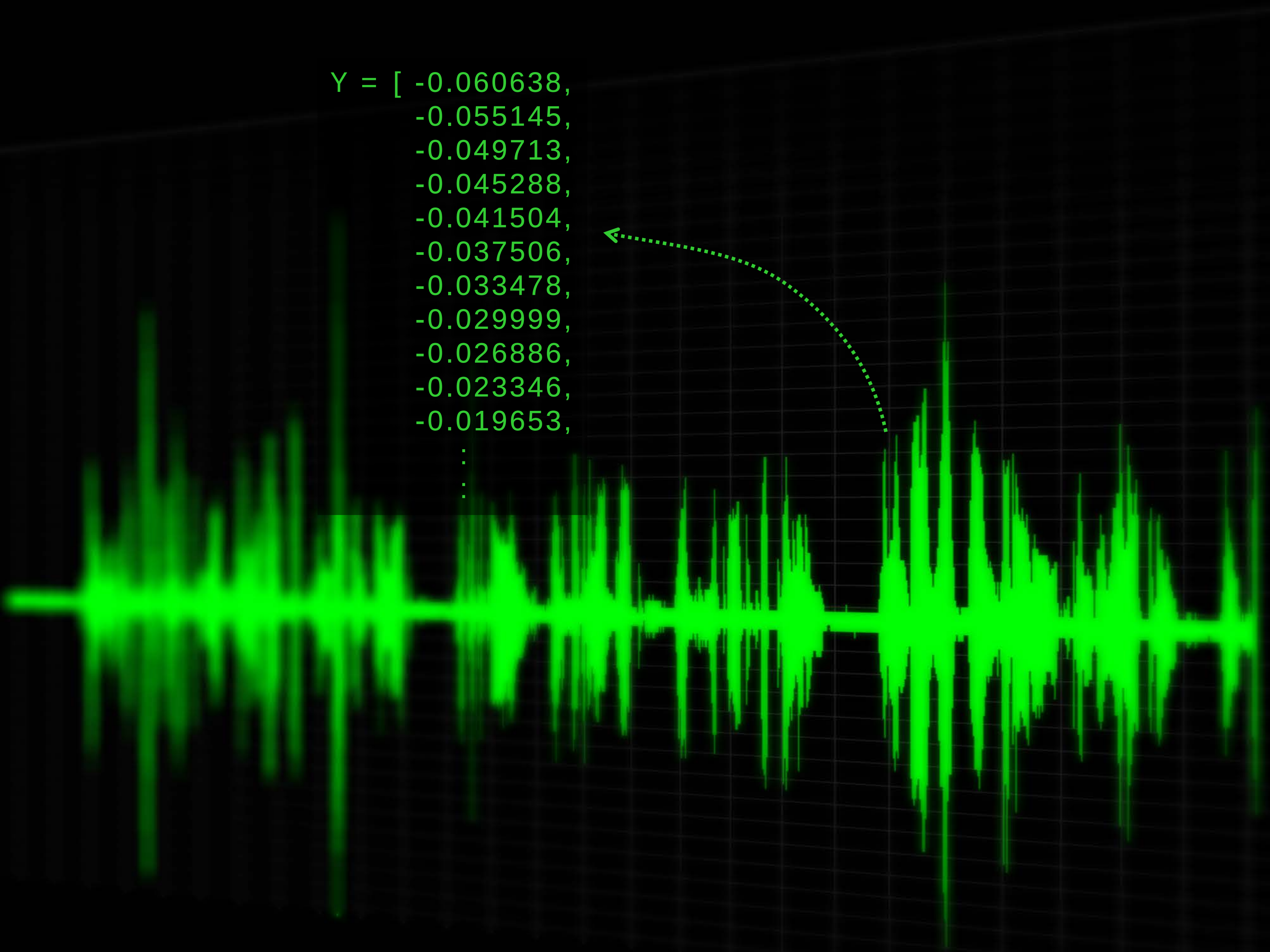
Time Resolution Vs. Frequency Resolution



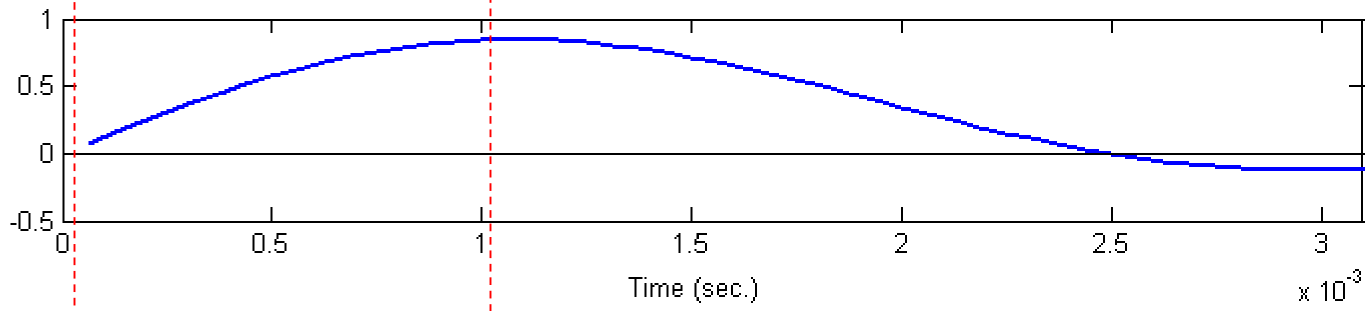
Different Types of Window






Y = [-0.060638,
-0.055145,
-0.049713,
-0.045288,
-0.041504,
-0.037506,
-0.033478,
-0.029999,
-0.026886,
-0.023346,
-0.019653,
⋮

Sampling

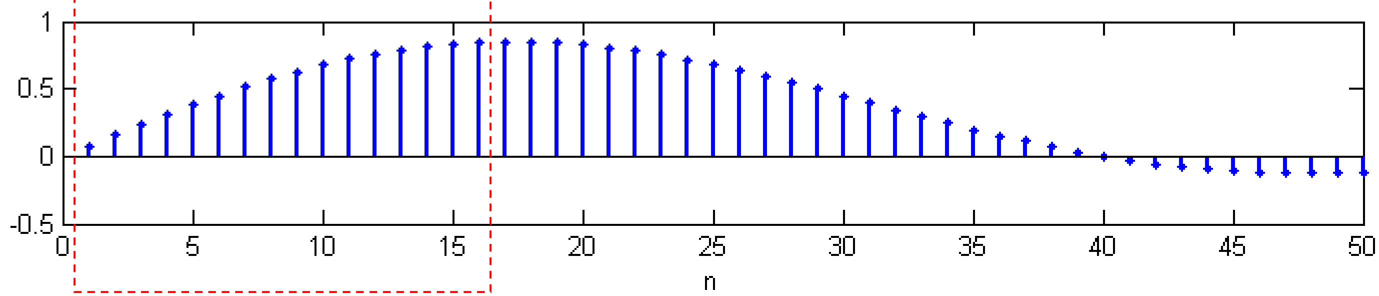


In 1 x 10^{-3} sec.
16 samples are picked

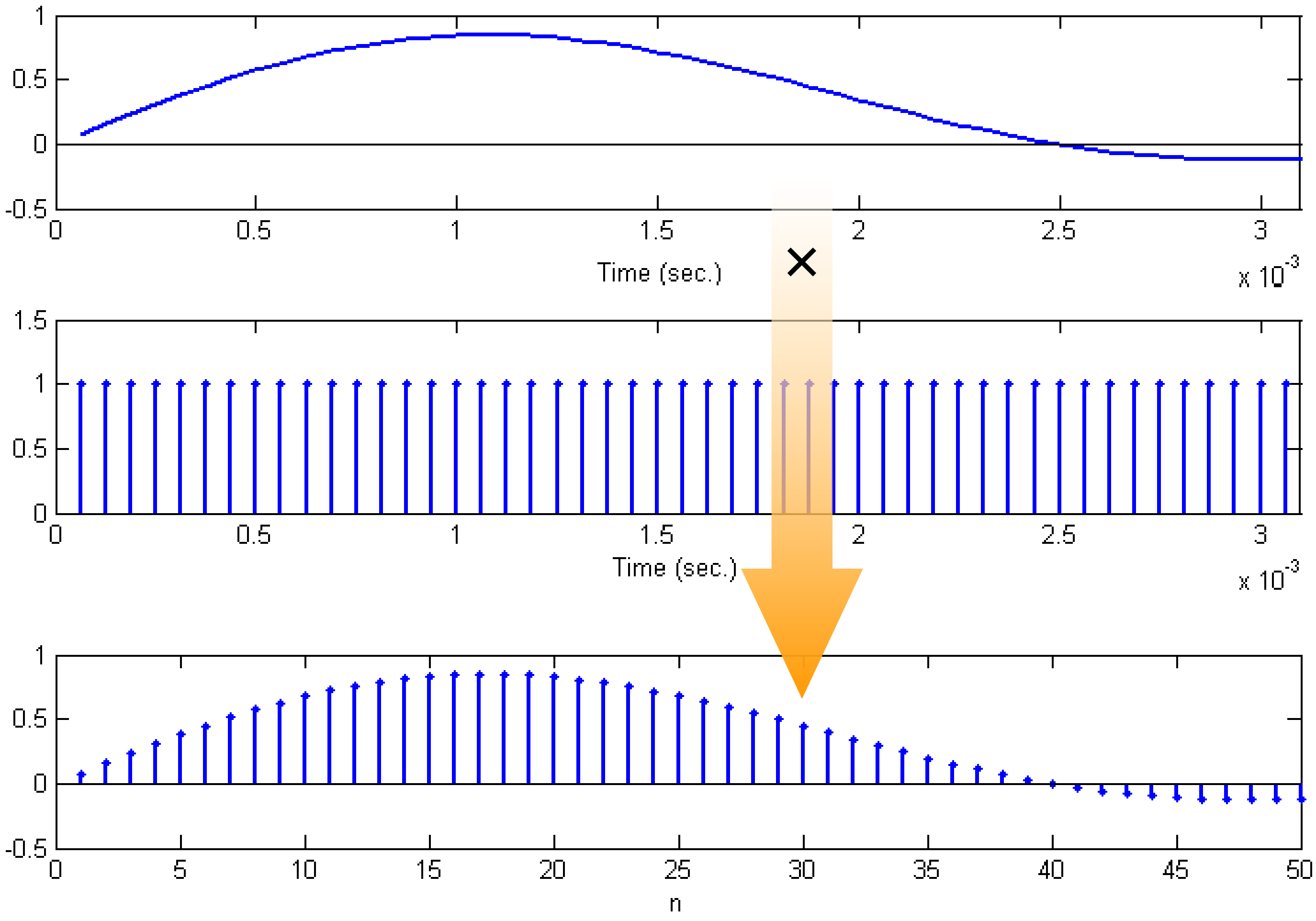
$\therefore$Fs = 16,000 samples/sec.
= 16 kHz.

Sampling Frequency, Fs

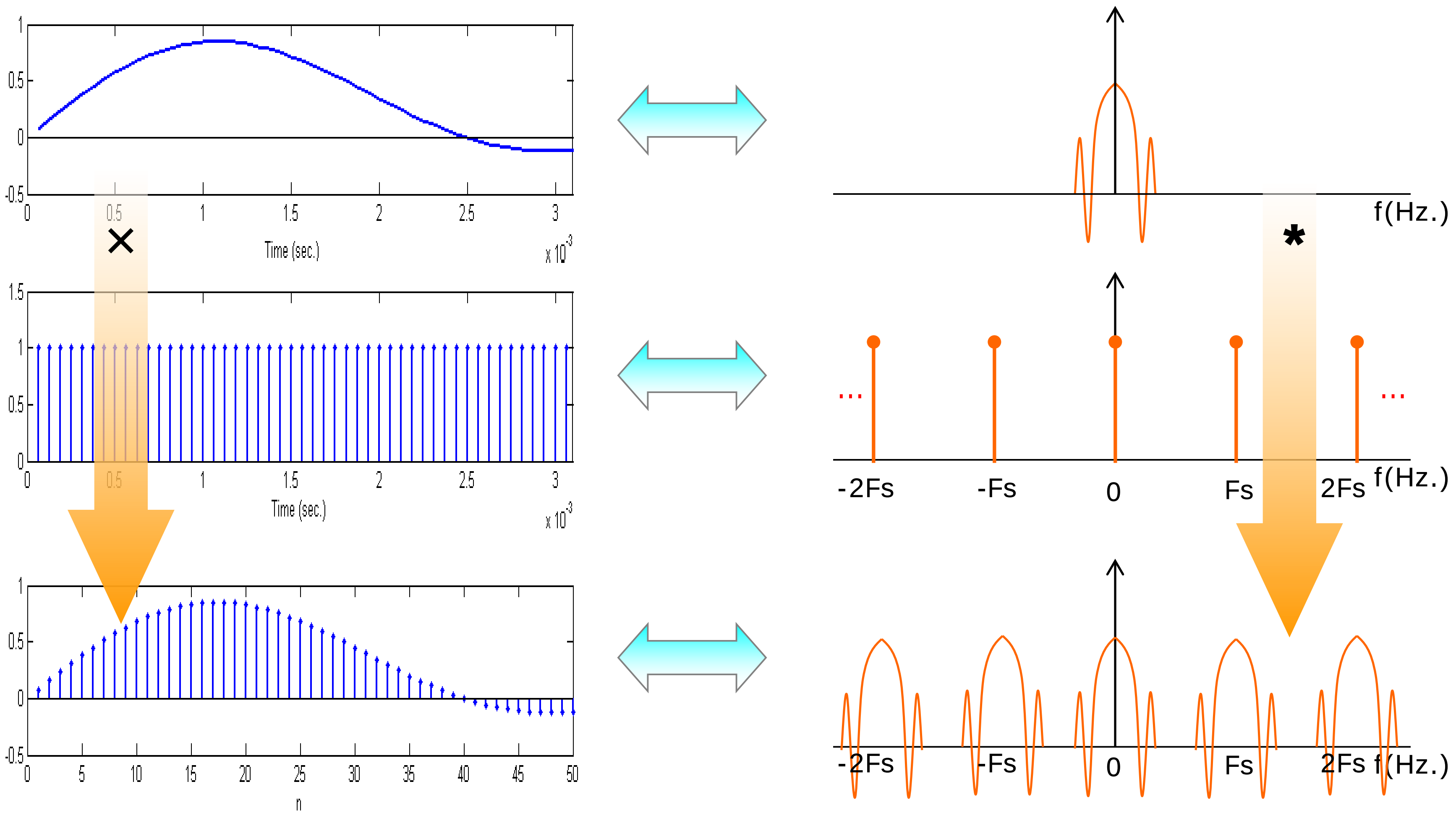
= how many samples are picked in the interval of 1 second.



Sampling

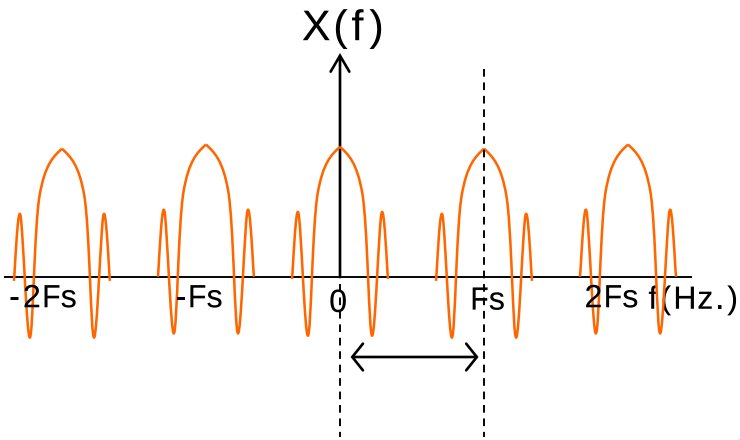

1
0.5
0
-0.5
0 0.5 1 1.5 2 2.5 3
Time (sec.)
x 10^-3
×
1.5
1
0.5
0
0 0.5 1 1.5 2 2.5 3
Time (sec.)
x 10^-3
1
0.5
0
-0.5
0 5 10 15 20 25 30 35 40 45 50
n

Sampling



Normalized Frequency

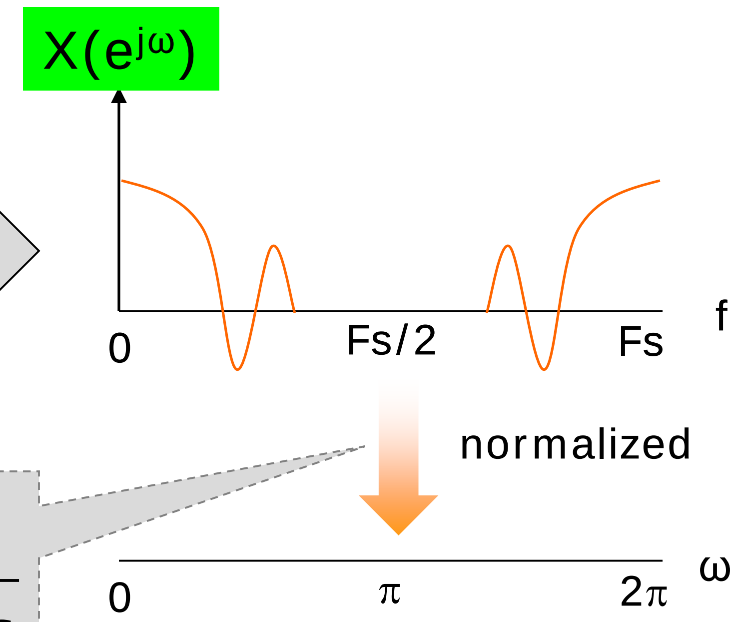
Fourier Transform



Repeats every Fs

f : frequency (Hz.)

Discrete-time Fourier Transform



$$\omega = 2\pi \frac{f}{Fs}$$

ω : Normalized frequency
(Radian per sample)

Discrete-time Fourier Transform

$$\boxed{x[n] \Leftrightarrow X(e^{j\omega})}$$

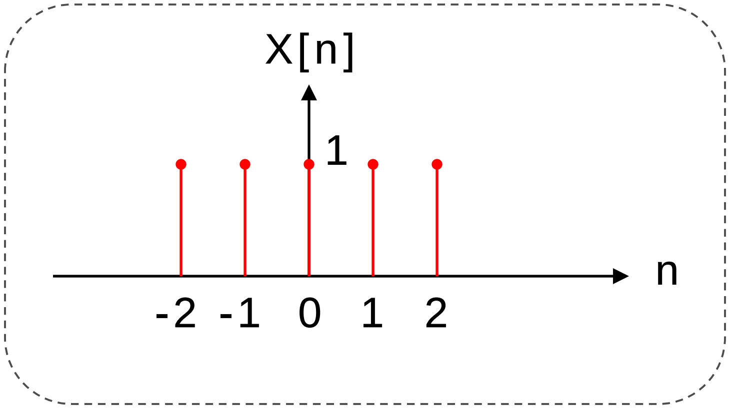
Discrete-time Fourier Transform

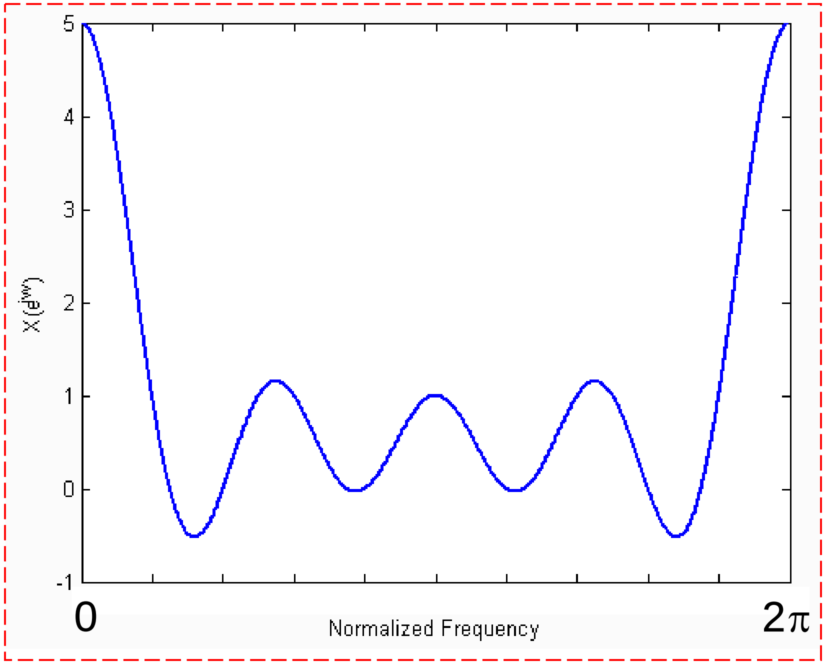
$$X(e^{j\omega}) = \sum_{n=-\infty}^{\infty} x[n] e^{-j\omega n}$$

Inverse Discrete-time Fourier Transform

$$x[n] = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\omega}) e^{j\omega n} d\omega$$

Example





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

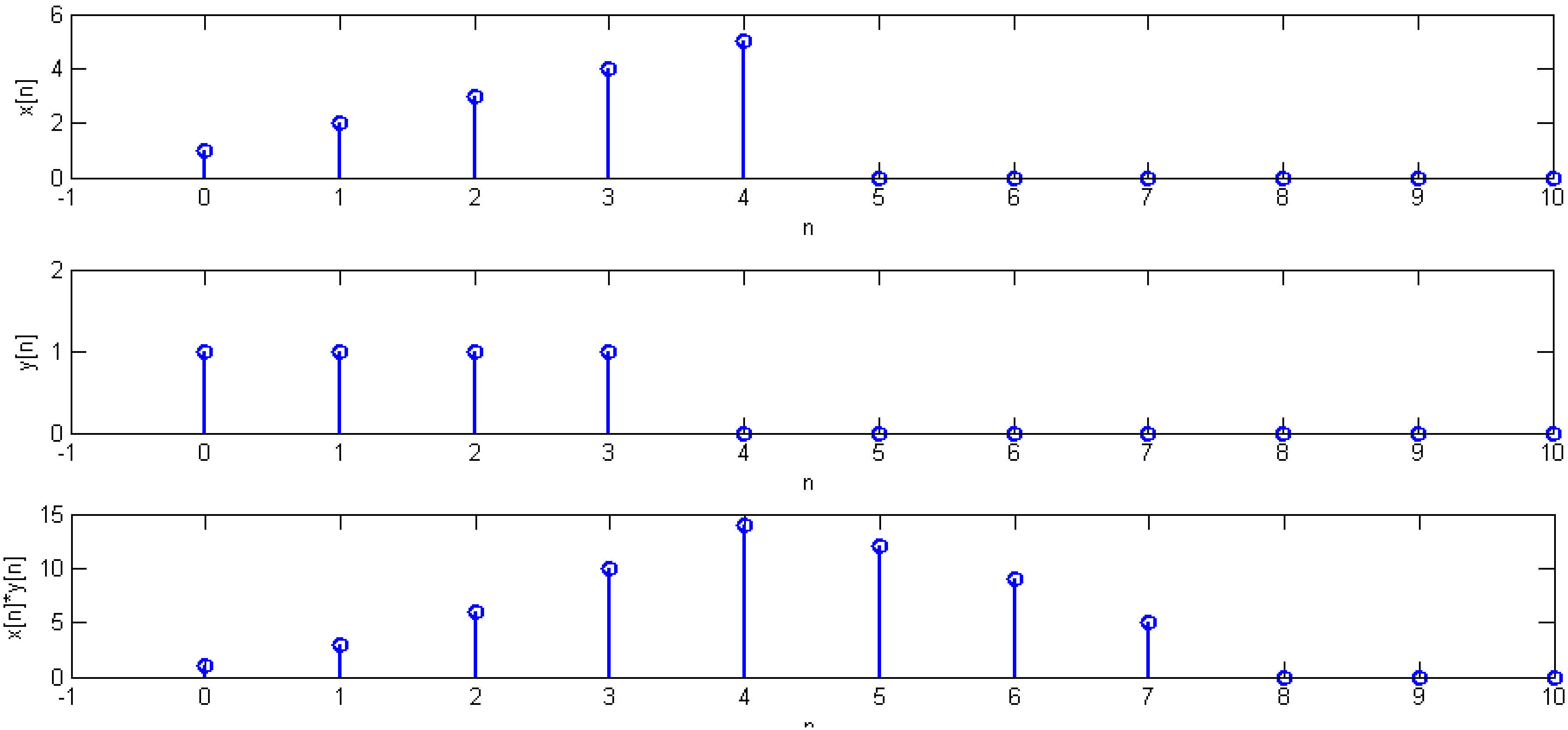
$$= x[-2]e^{-j\omega(-2)} + x[-1]e^{-j\omega(-1)} + x[0]e^{-j\omega(0)} + x[1]e^{-j\omega(1)} + x[2]e^{-j\omega(2)}$$

$$= e^{j\omega(2)} + e^{j\omega} + 1 + e^{-j\omega} + e^{-j\omega(2)}$$

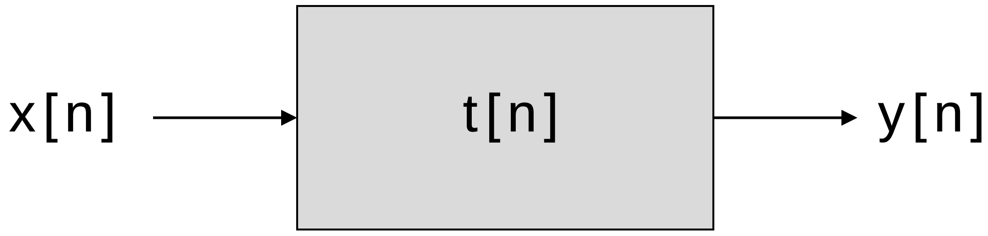
$$= \boxed{1 + 2\cos(\omega) + 2\cos(2\omega)}$$

Discrete-time Convolution

$$x[n] * y[n] = \sum_{m=-\infty}^{\infty} x[m]y[n-m]$$



Discrete-time Filter



$$Y(e^{j\omega}) = X(e^{j\omega})T(e^{j\omega}) \quad \Leftrightarrow \quad y[n] = x[n] * t[n]$$

Family of Fourier Transform

- "Discreteness" in one domain implies "Periodicity" in the other domain.
- "Continuity" in one domain implies "Aperiodicity" in the other domain.

Transform	Time	Freq.
Cont. Fourier Trans.	Cont. & aperiodic	Cont. & aperiodic
Fourier Series	Cont. & periodic	Disc. & aperiodic
DTFT	Disc. & aperiodic	Cont. & periodic
DFT	Disc. & periodic	Disc & periodic

Discrete Fourier Transform

$$x[n] \Leftrightarrow X[k]$$

DFT

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi kn}{N}} \qquad k = 0,1,\ldots, N-1$$

IDFT

$$X[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[k] e^{j\frac{2\pi kn}{N}} \qquad n = 0,1,\ldots, N-1$$

Discrete Fourier Transform

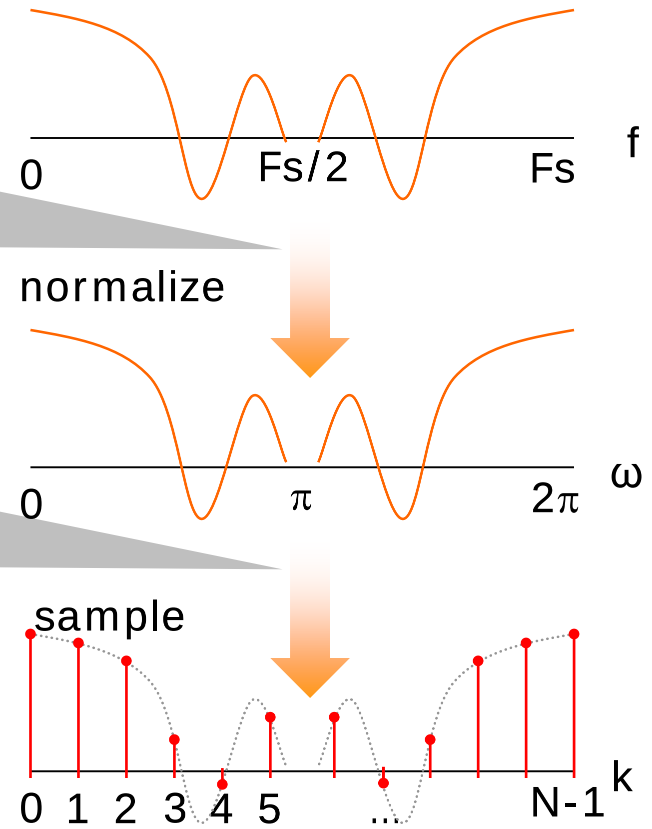
$$X(f)=\int_{-\infty}^{\infty} x(t)e^{-j2\pi ft}\,dt$$

$$\omega = 2\pi\frac{f}{Fs}$$

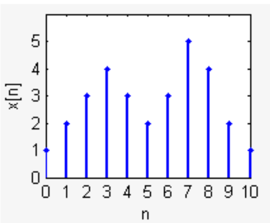
$$X(e^{j\omega})=\sum_{n=-\infty}^{\infty} x[n]e^{-j\omega n}$$

$$k=\frac{\omega}{2\pi}N$$

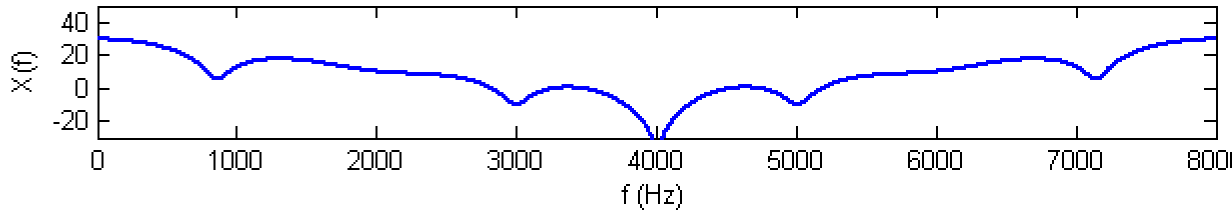
$$X[k]=\sum_{n=0}^{N-1} x[n]e^{-j\frac{2\pi kn}{N}}$$



N DFT

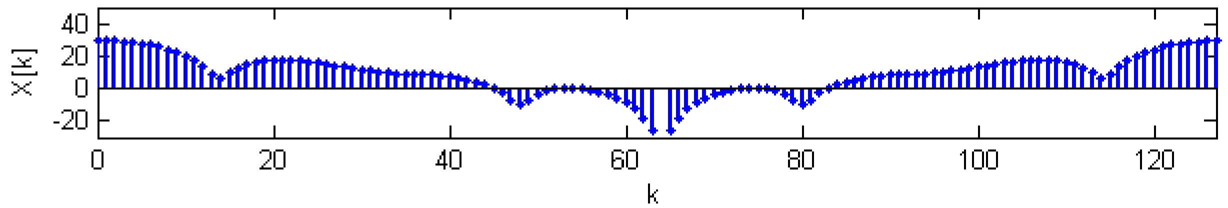


DTFT



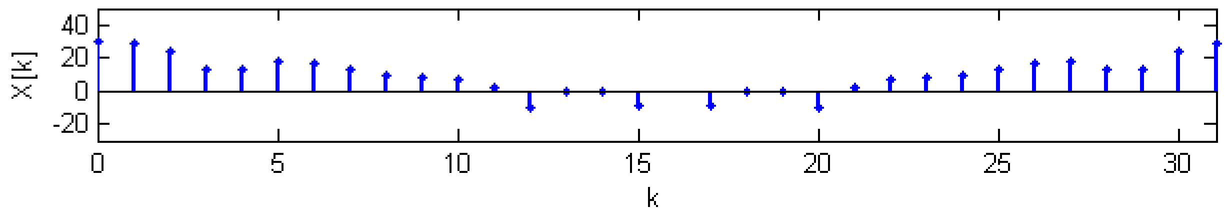
DFT

N=128



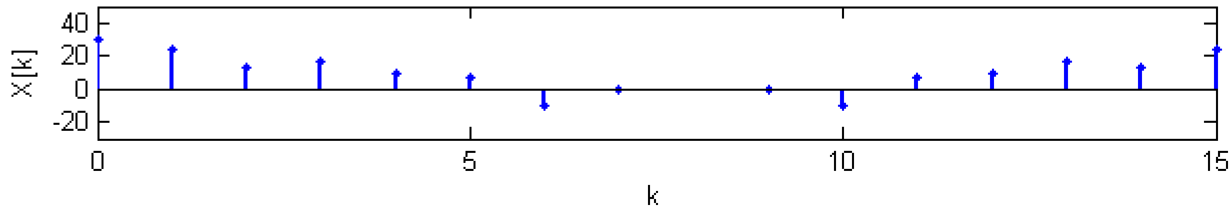
DFT

N=32



DFT

N=16



Numerical Computing Environment

Millions of Engineers and Scientists Trust MATLAB

MATLAB combines a desktop environment tuned for iterative analysis and design processes with a programming language that expresses matrix and array mathematics directly.

Professionally Built

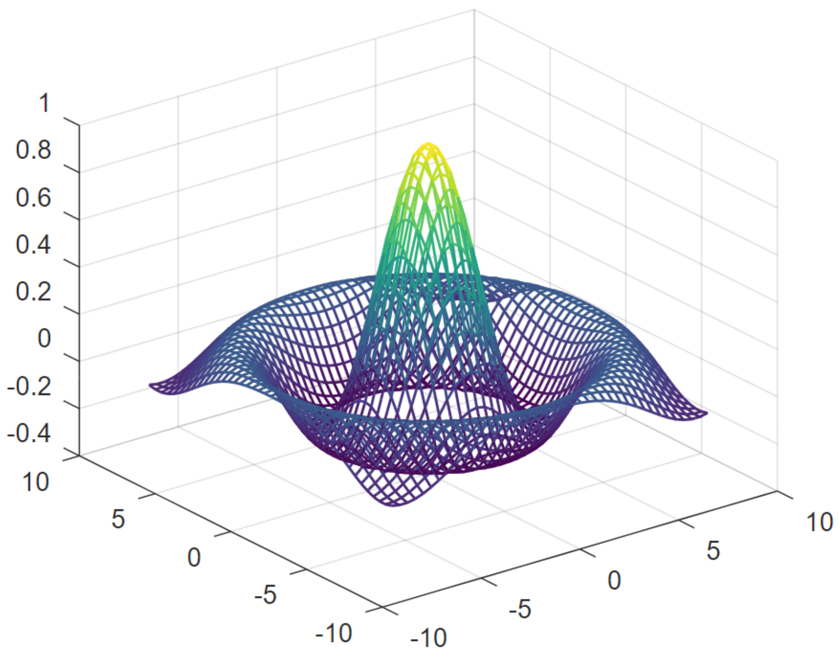
MATLAB toolboxes are professionally developed, rigorously tested, and fully documented.

With Interactive Apps

MATLAB apps let you see how different algorithms work with your data. Iterate until you've got the results you want, then automatically generate a MATLAB program to reproduce or automate your work.

Retrieved from: https://www.mathworks.com/products/matlab.html
On January 31st, 2018

GNU Octave



GNU Octave

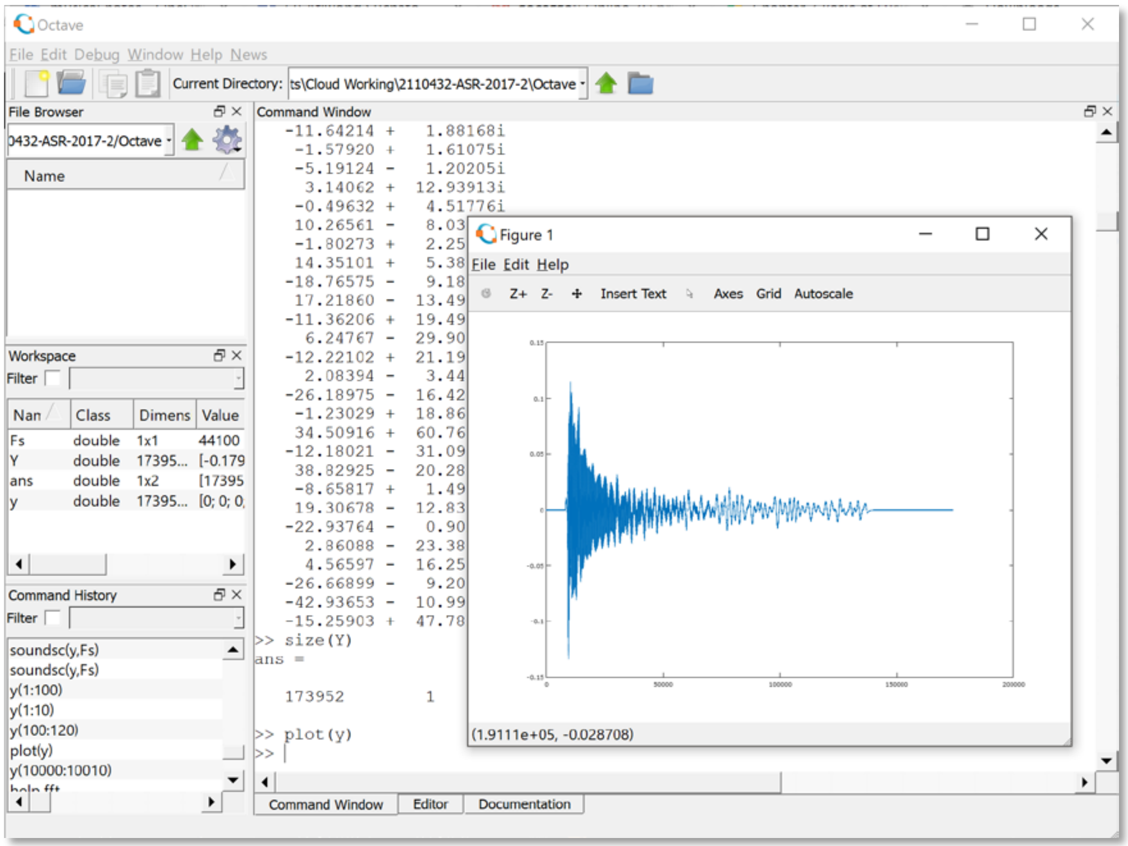
Scientific Programming Language

- Powerful mathematics-oriented syntax with built-in plotting and visualization tools
- Free software, runs on GNU/Linux, macOS, BSD, and Windows
- Drop-in compatible with many Matlab scripts

Download | Docs

Retrieved from: https://www.gnu.org/software/octave/
On January 31st, 2018

Octave Demo



- Basic operations
  - Variables
  - Colon operator
  - Matrix manipulation
  - Convolution
- Arrays
- Plotting
- Loading/Saving audio
- Help
- Control structures