UNIVERSITY OF LINCOLN

# CMP9780, EGR3031 & BME3002

## Short Time Fourier Transform & Wavelet Transform

# The Story so far…

- Introduction to Signals
- Sampling
- Fourier Transform
- Discrete and Fast Fourier Transform
- Convolution and Correlation

# SHORT TIME FOURIER TRANSFORM (STFT)

# Fourier Transform (Recall)

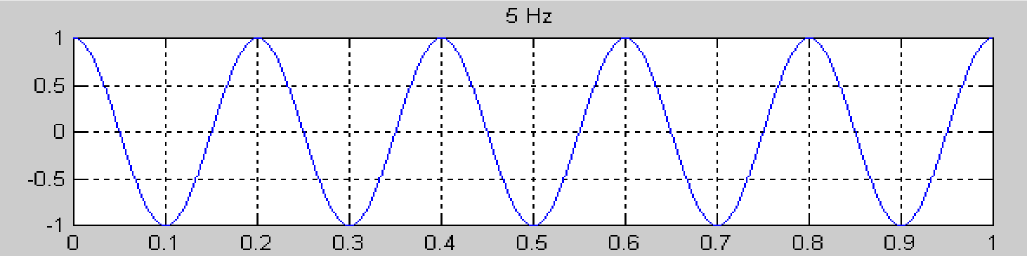
- Fourier Transform reveals **which frequency components** are present in a given function:

$$f(x) = \sum_{u=0}^{N-1} F(u) e^{\frac{j2\pi ux}{N}}, x = 0, 1, \ldots, N-1$$ (inverse DFT)

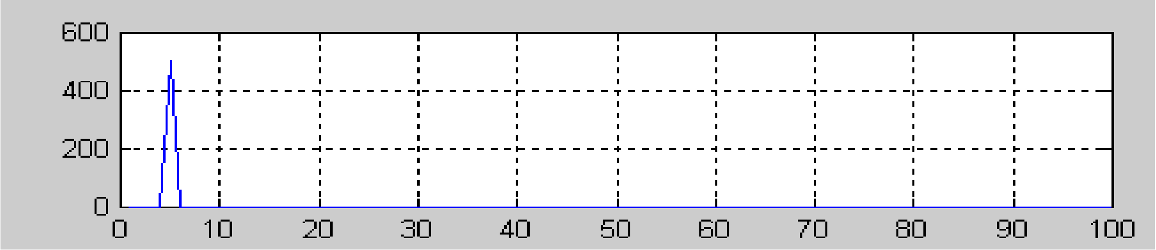
where: $$F(u) = \frac{1}{N} \sum_{x=0}^{N-1} f(x) e^{\frac{-j2\pi ux}{N}}, u = 0, 1, \ldots, N-1$$ (forward DFT)

# Examples

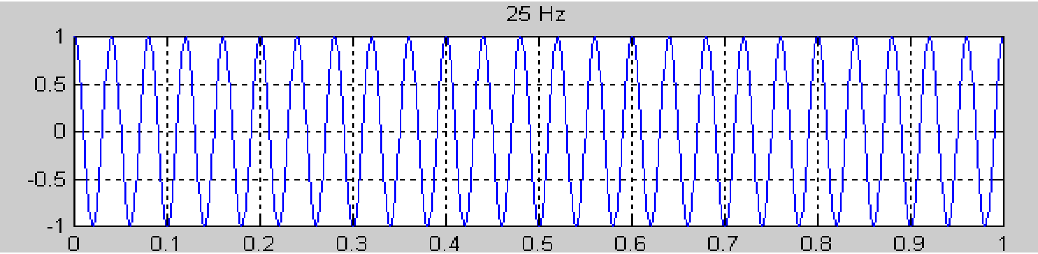
$$f_1(t) = \cos(2\pi \cdot 5 \cdot t)$$



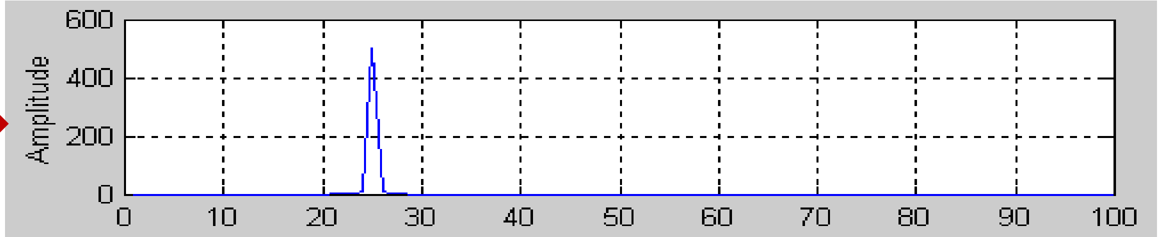

$F_1(u)$



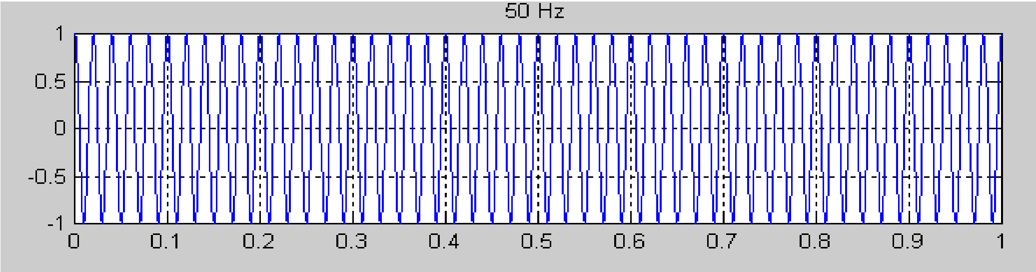
$$f_2(t) = \cos(2\pi \cdot 25 \cdot t)$$



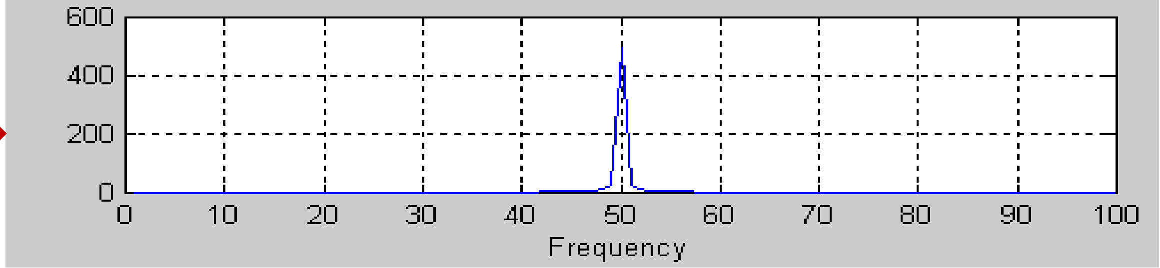

$F_2(u)$




$$f_3(t) = \cos(2\pi \cdot 50 \cdot t)$$

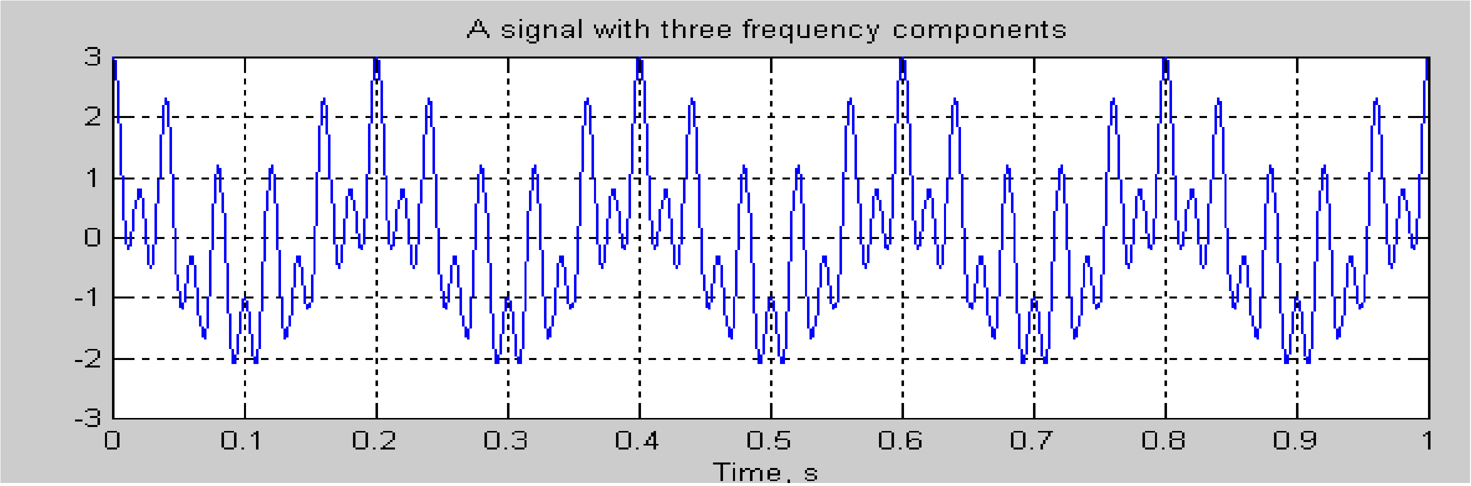



$F_3(u)$

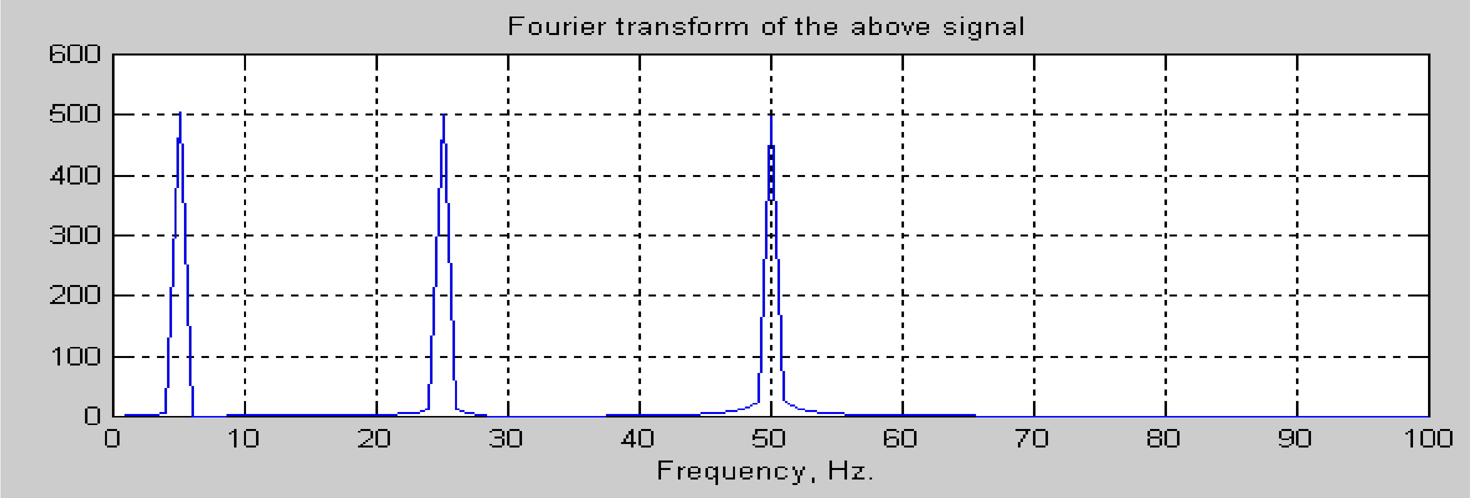


# Fourier Analysis – Examples (cont'd)

$$f_4(t) = \cos(2\pi \cdot 5 \cdot t) + \cos(2\pi \cdot 25 \cdot t) + \cos(2\pi \cdot 50 \cdot t)$$




$F_4(u)$



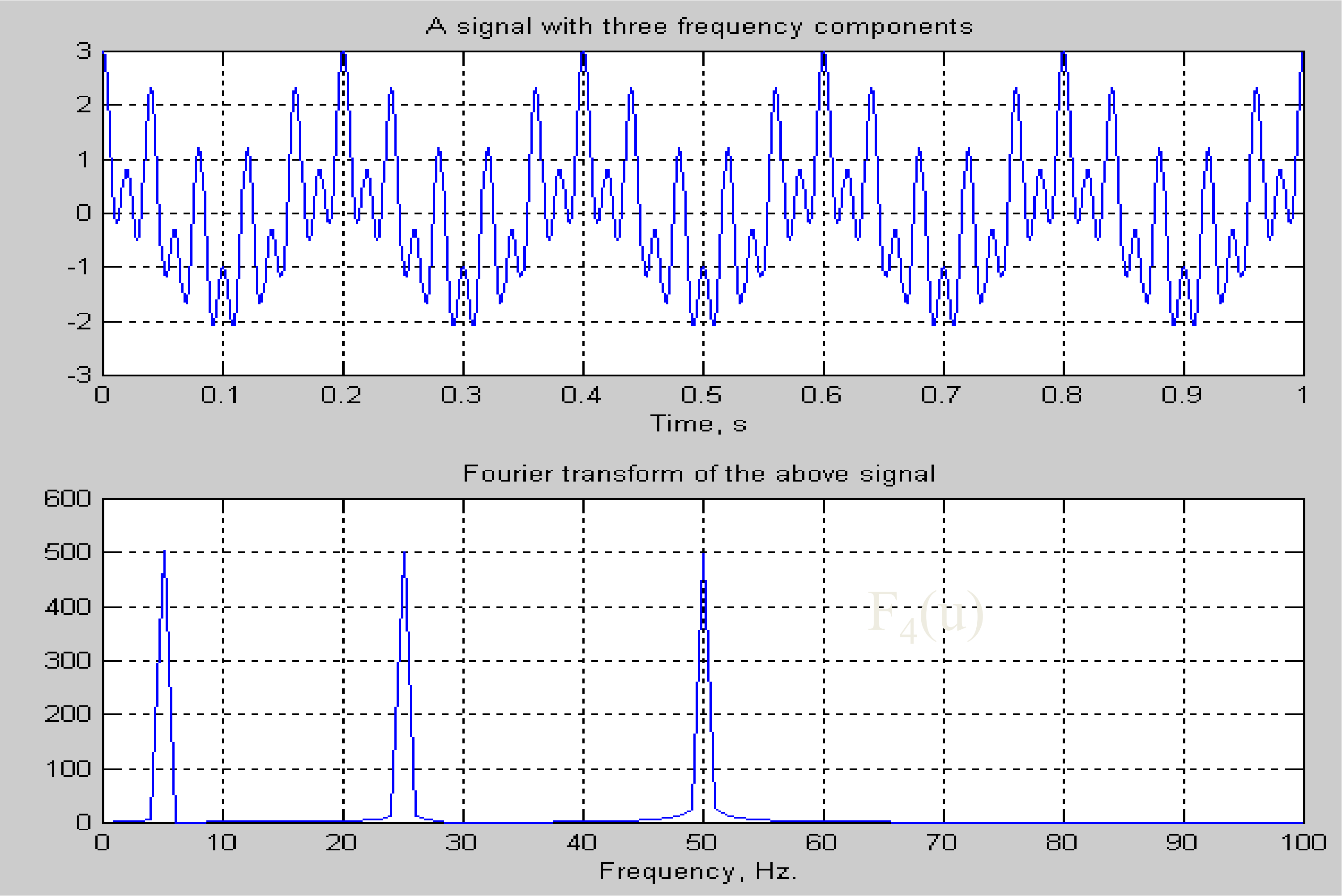
# Limitations of Fourier Transform

1. Can not provide simultaneous time and frequency localization.

# Fourier Analysis – Examples (cont'd)

$$f_4(t) = \cos(2\pi \cdot 5 \cdot t) + \cos(2\pi \cdot 25 \cdot t) + \cos(2\pi \cdot 50 \cdot t)$$

Provides excellent localization in the frequency domain but poor localization in the time domain.



# Limitations of Fourier Transform (cont'd)

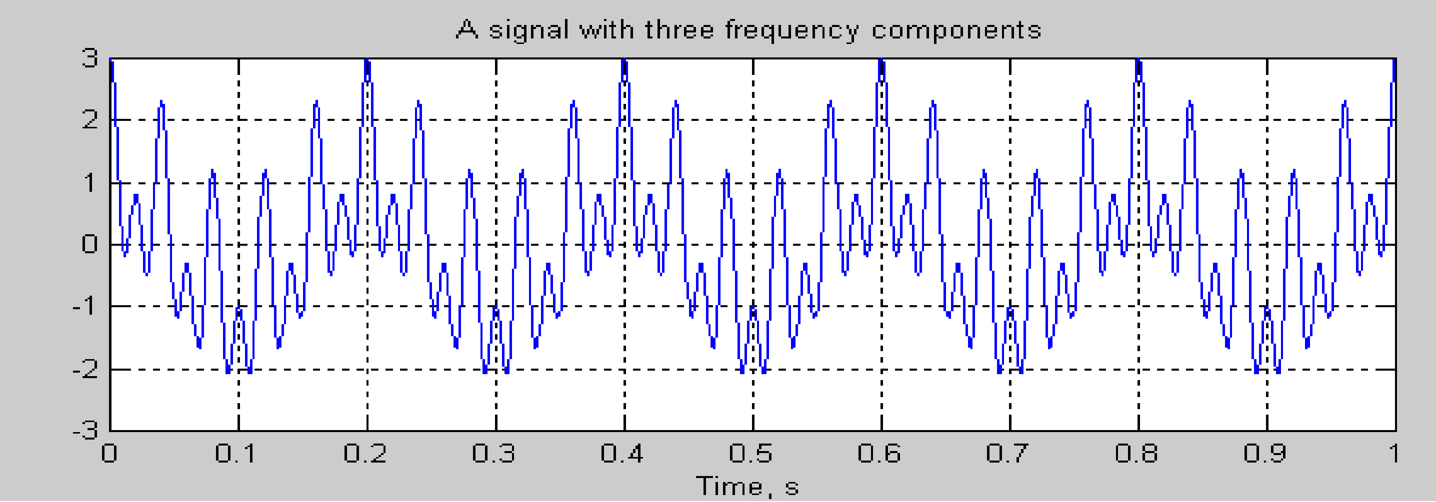
1. Can not provide simultaneous time and frequency localization.

2. Not very useful for analyzing time-variant, non-stationary signals.

# Stationary vs non-stationary signals (cont'd)

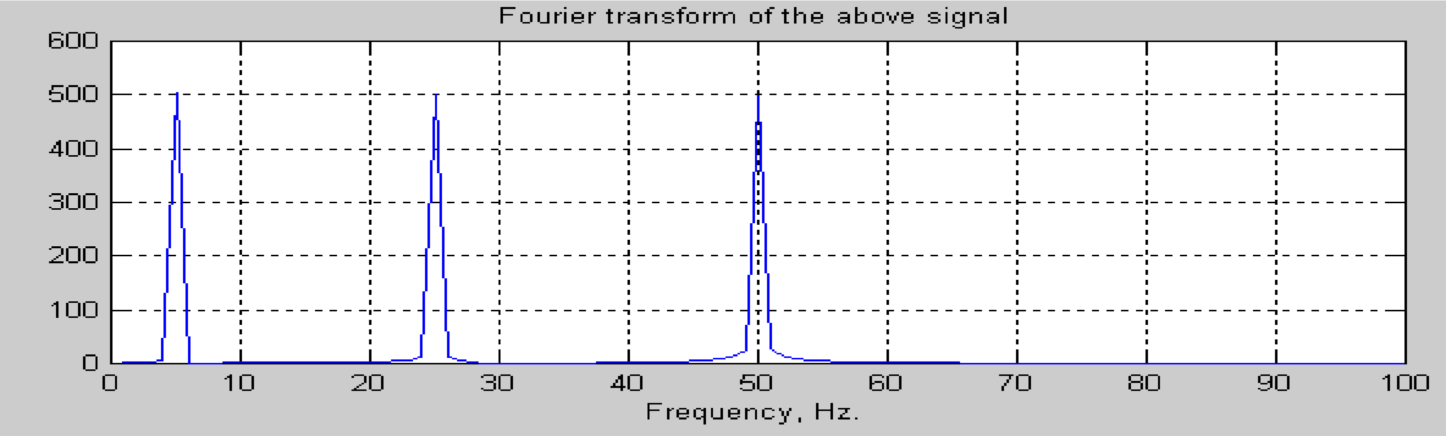
Stationary signal
(non-varying frequency):

$f_4(t)$




Three frequency components,
present at all times!

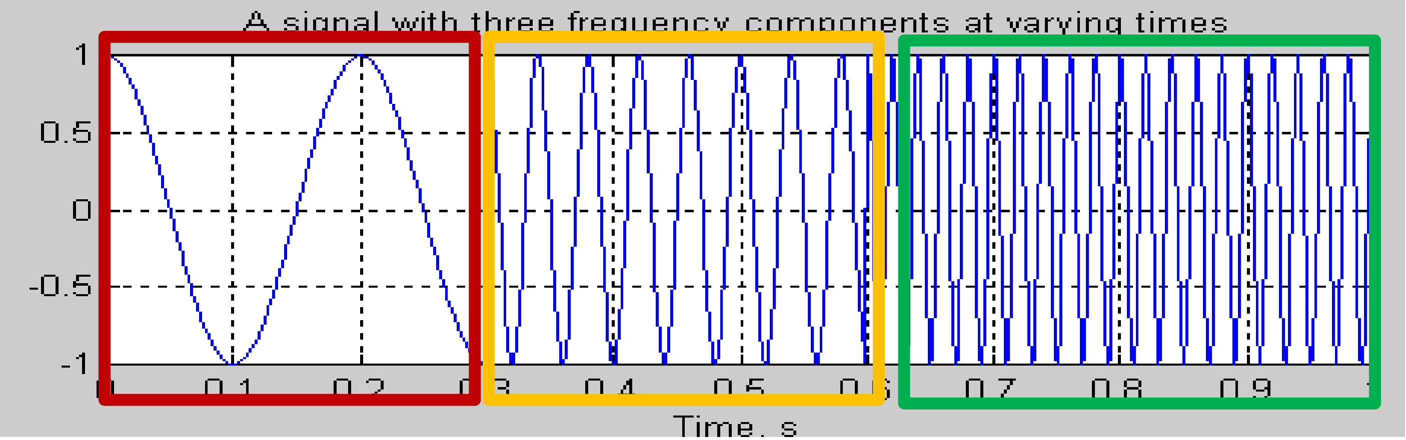
F4(u)



# Stationary vs non-stationary signals (cont'd)

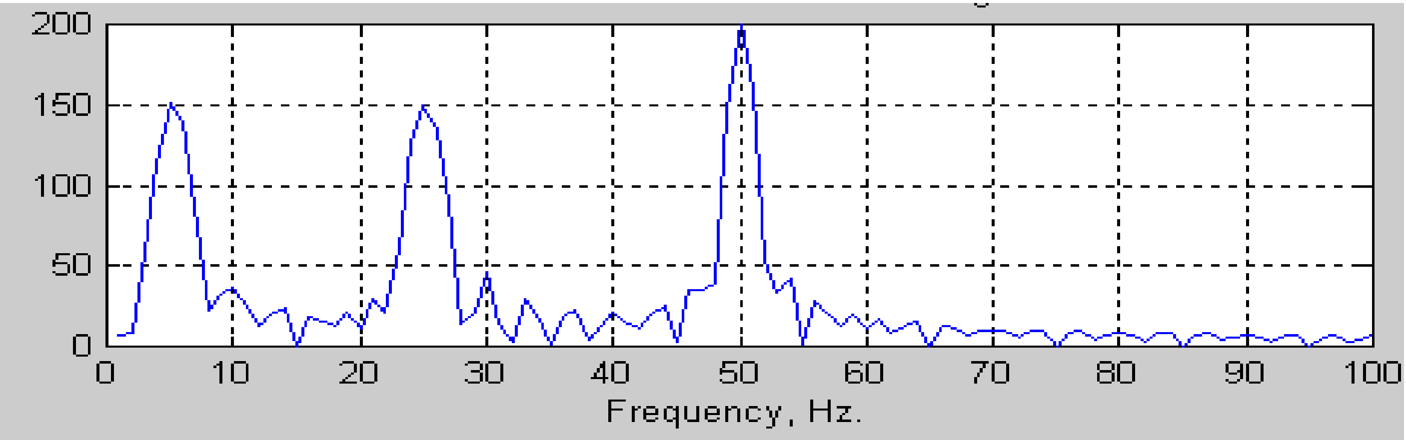
Non-stationary signal
(varying frequency):

$f_5(t)$




Three frequency components,
NOT present at all times!

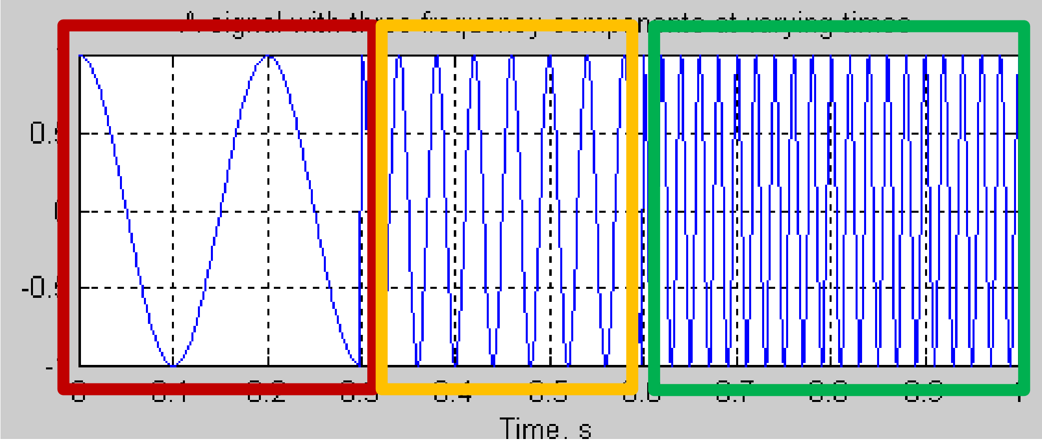
F5(u)



# Stationary vs non-stationary signals (cont'd)

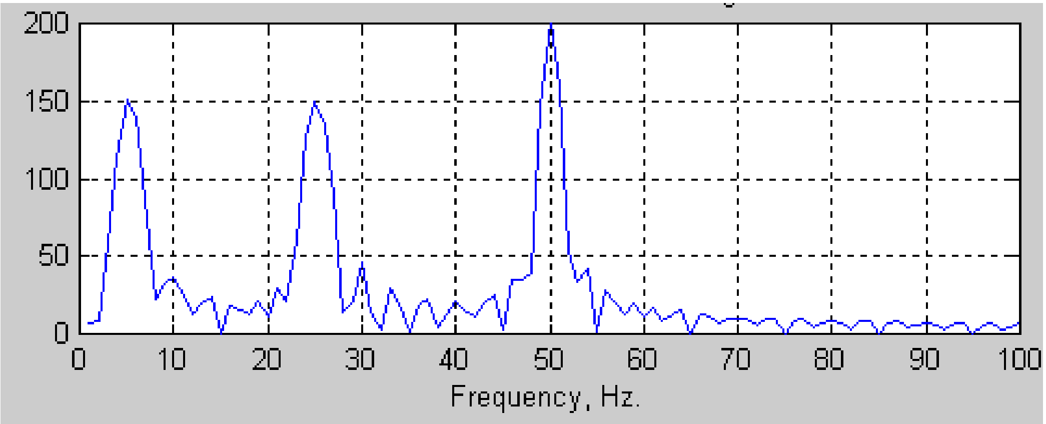
Non-stationary signal (varying frequency):

$f_5(t)$




Perfect knowledge of what frequencies exist, but no information about where these frequencies are located in time!

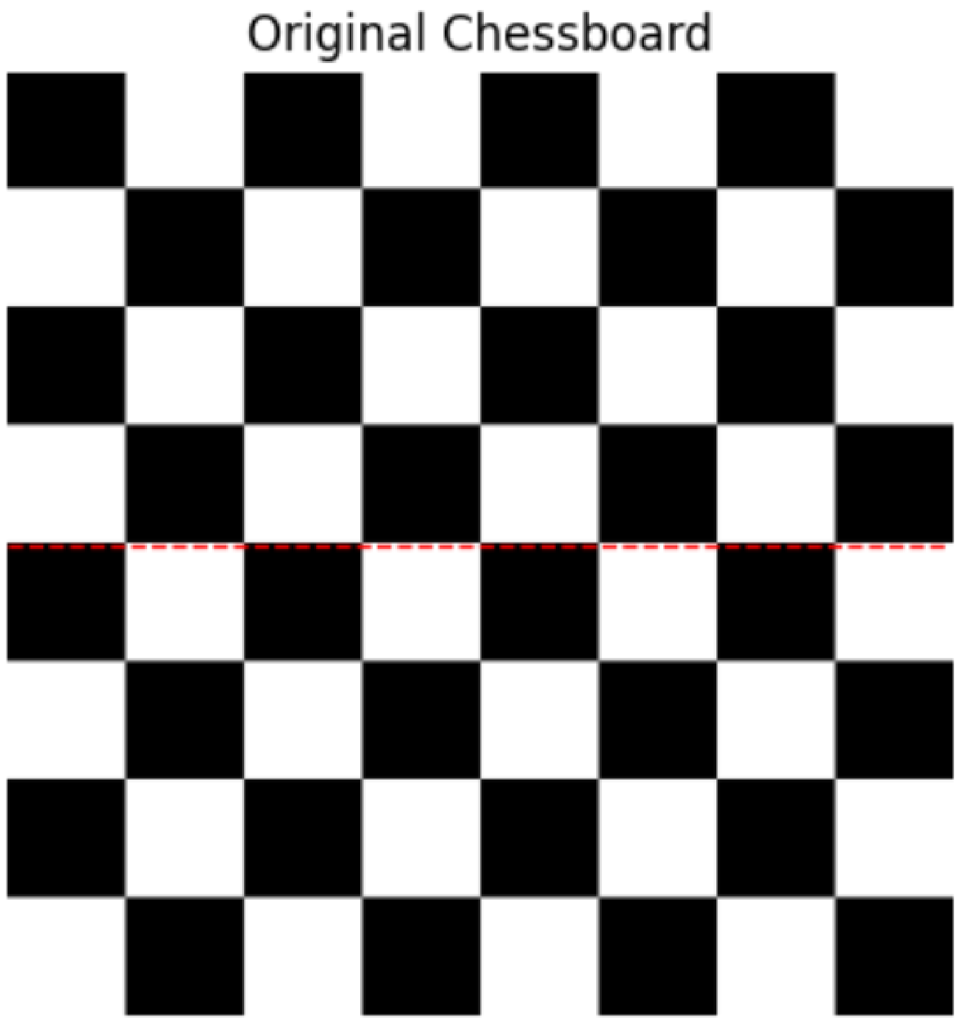
F5(u)

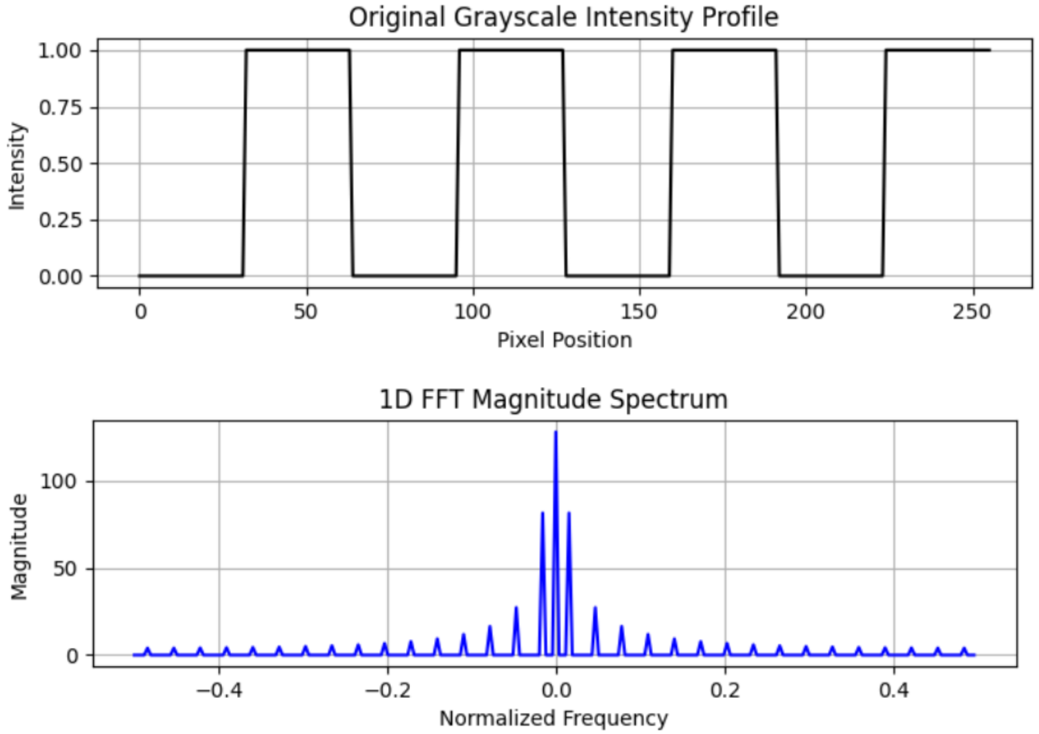


# Limitations of Fourier Transform (cont'd)

1. Cannot provide simultaneous time and frequency localization.
2. Not very useful for analyzing time-variant, non-stationary signals.
3. Not efficient for representing discontinuities or sharp corners.

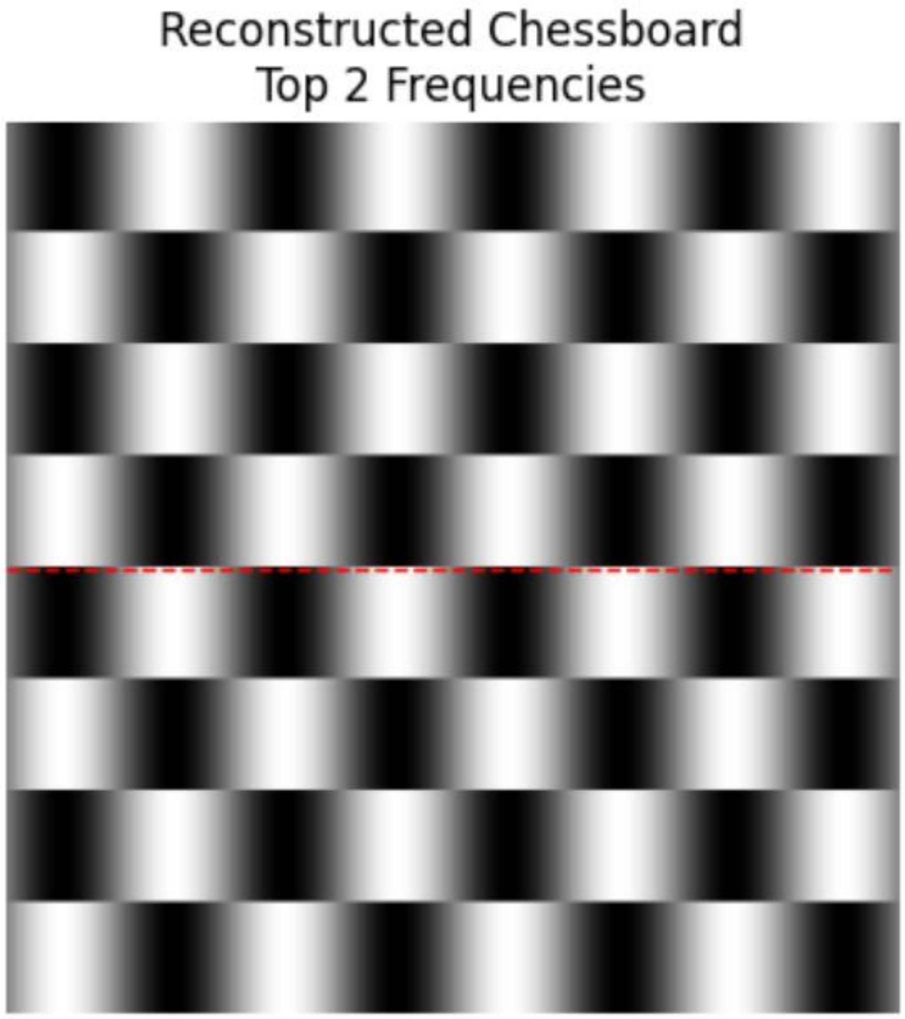
# Representing discontinuities or sharp corners

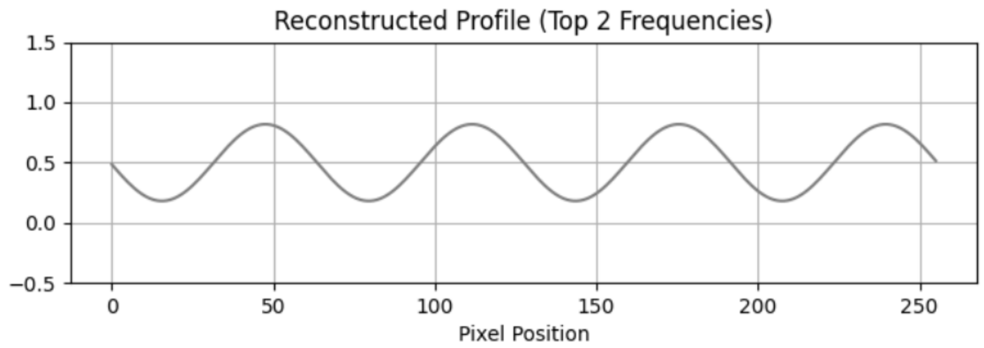


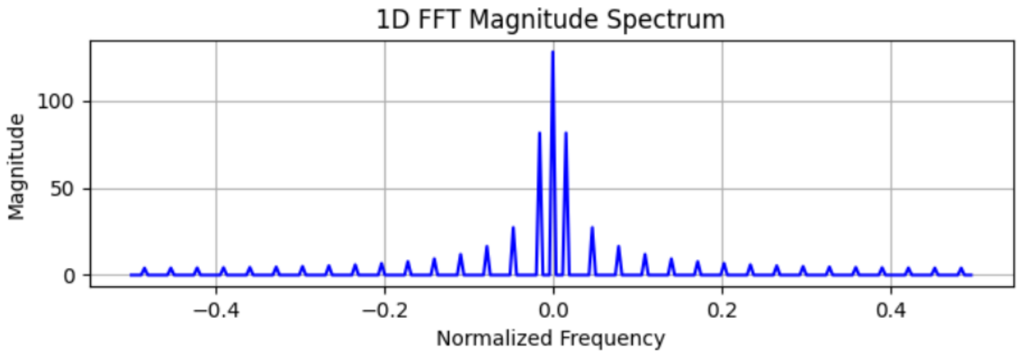




$$F(u) = \frac{1}{N}\sum_{x=0}^{N-1} f(x)e^{\frac{-j2\pi ux}{N}}, u = 0, 1, \ldots, N-1$$

# Representing discontinuities or sharp corners - Reconstruction

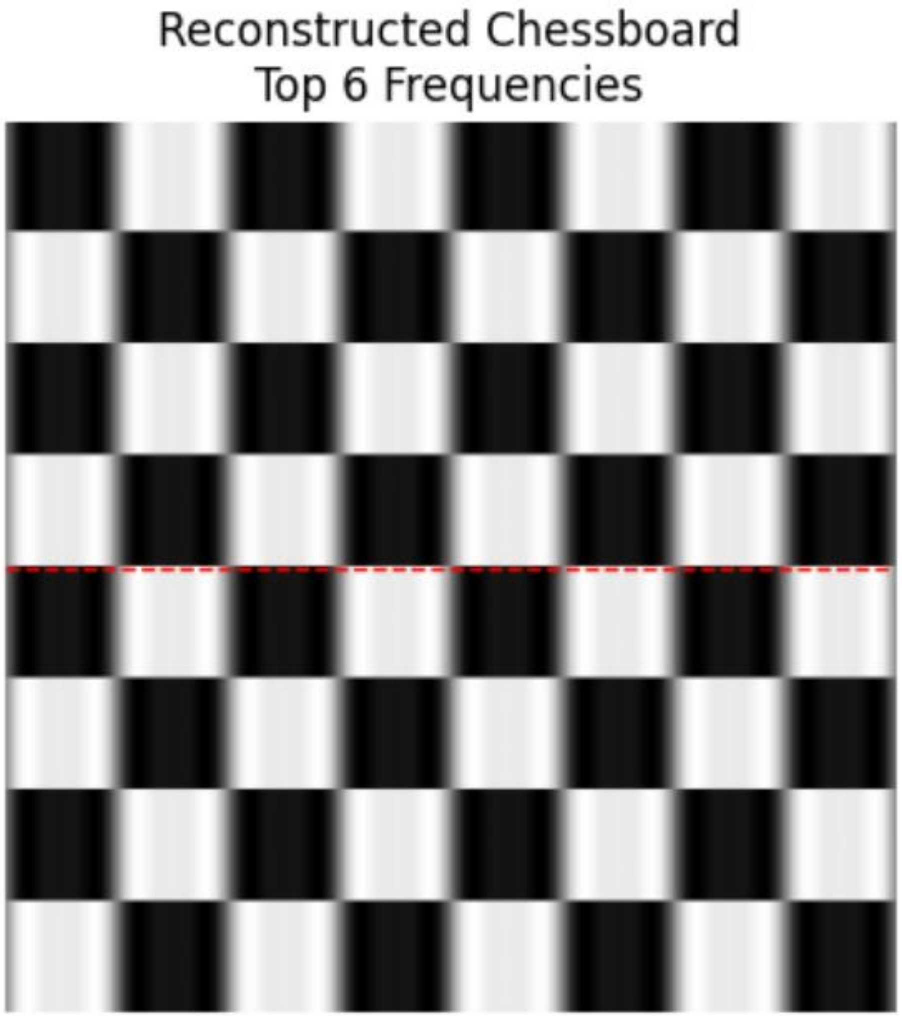


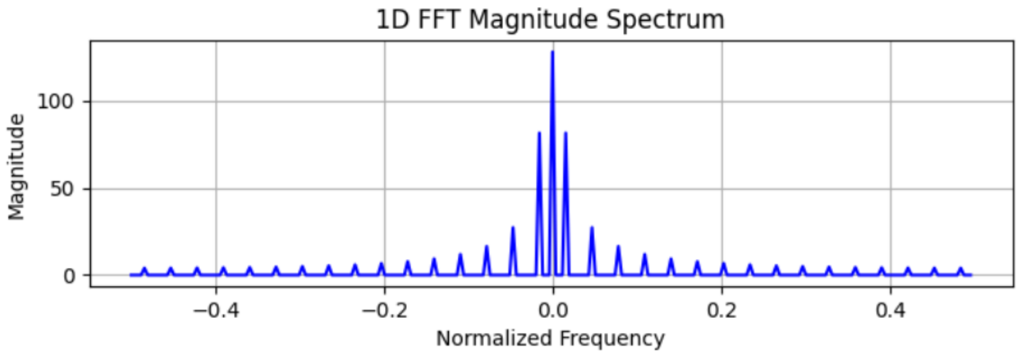




$$f(x) = \sum_{u=0}^{2} F(u) e^{\frac{j2\pi ux}{N}}, \; x = 0, 1, \ldots, N-1$$

# Representing discontinuities or sharp corners - Reconstruction

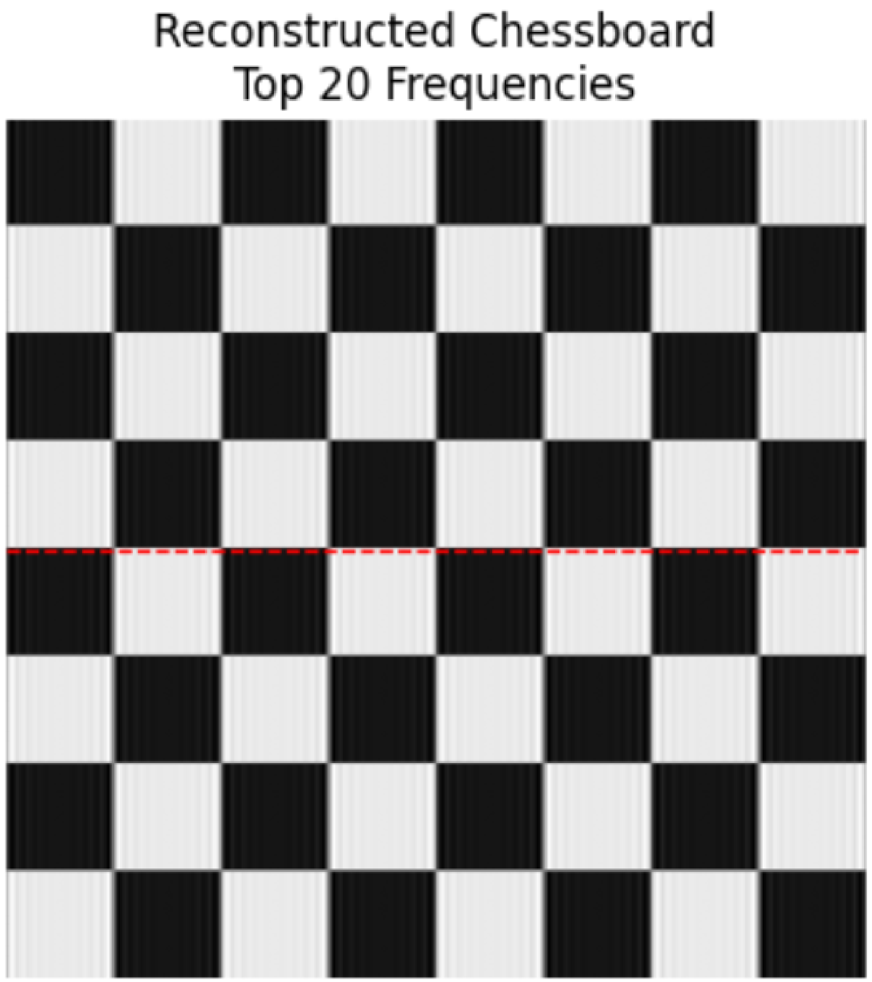




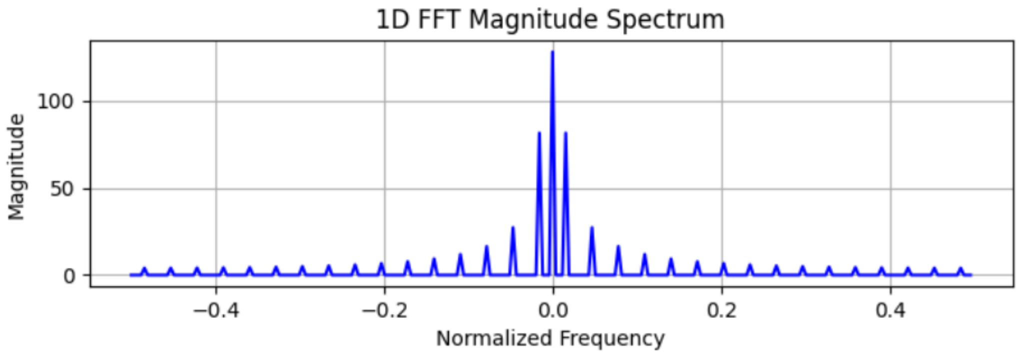



$$f(x) = \sum_{u=0}^{6} F(u)e^{\frac{j2\pi ux}{N}}, x = 0, 1, \dots, N-1$$

# Representing discontinuities or sharp corners - Reconstruction

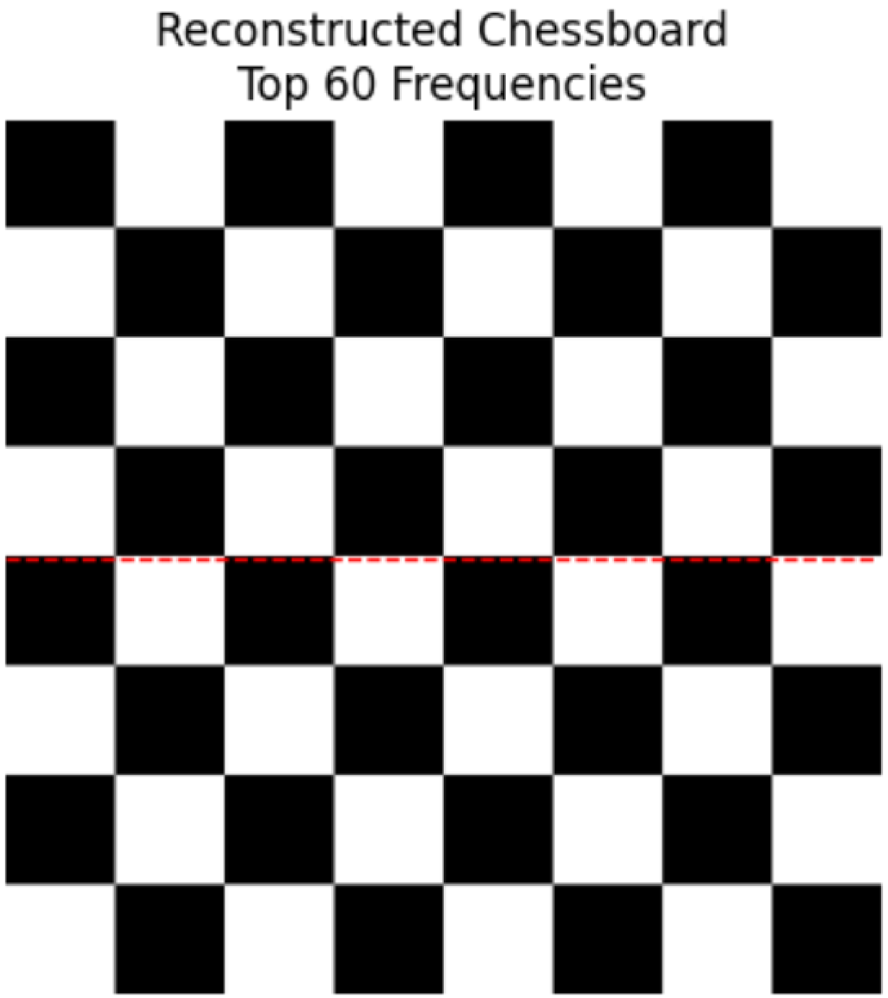


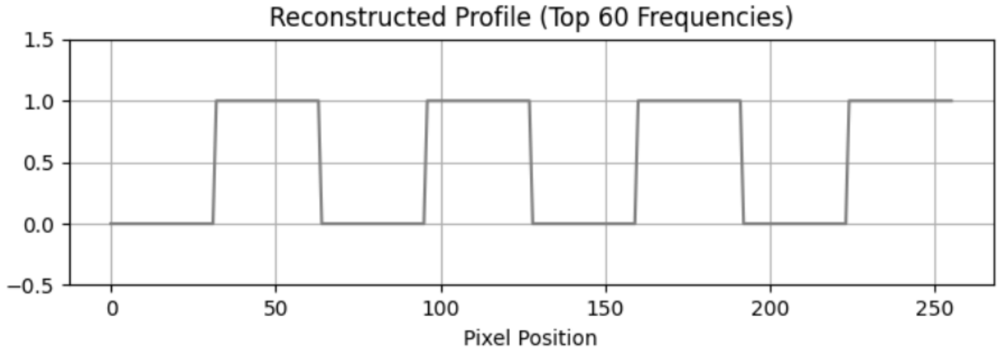


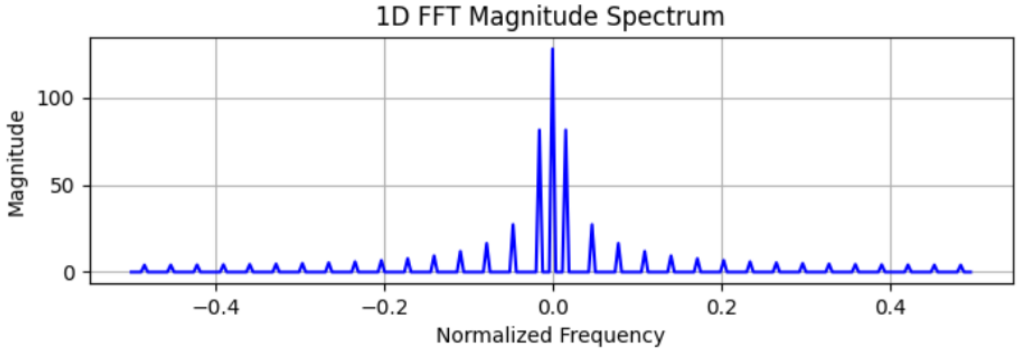



$$f(x) = \sum_{u=0}^{20} F(u) e^{\frac{j2\pi ux}{N}}, \; x = 0, 1, \ldots, N-1$$

# Representing discontinuities or sharp corners - Reconstruction

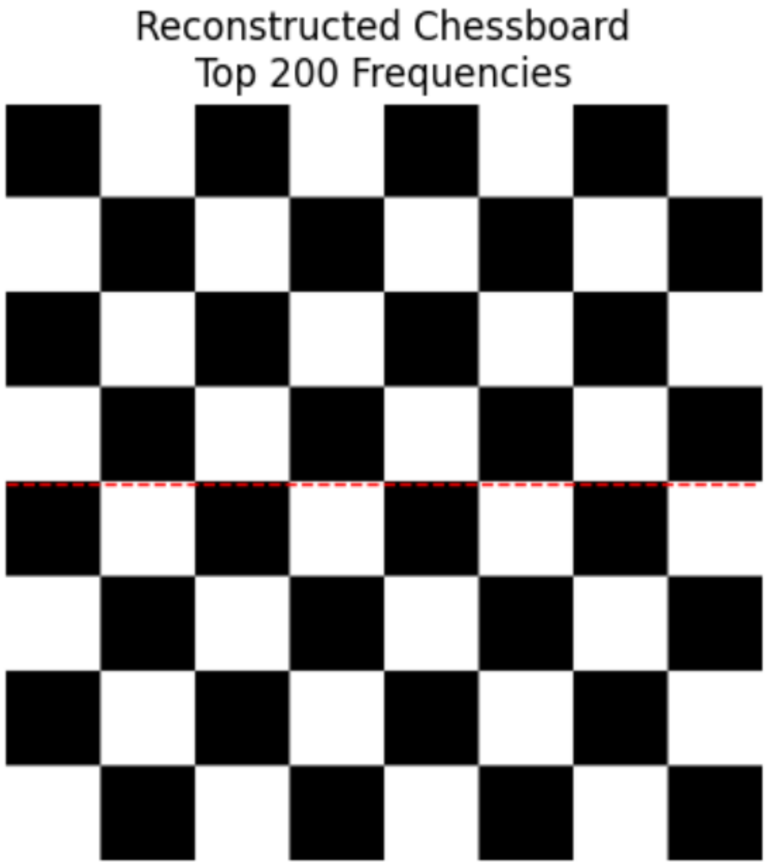


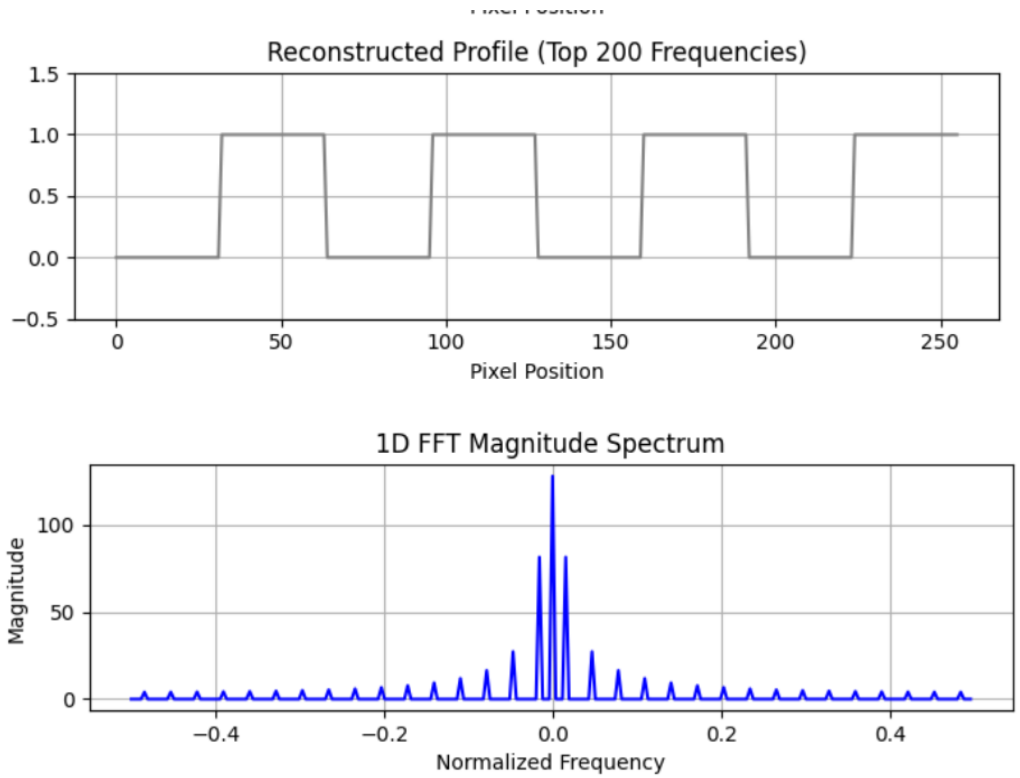




$$f(x) = \sum_{u=0}^{60} F(u) e^{\frac{j2\pi ux}{N}}, \ x = 0, 1, \ldots, N-1$$

# Representing discontinuities or sharp corners - Reconstruction



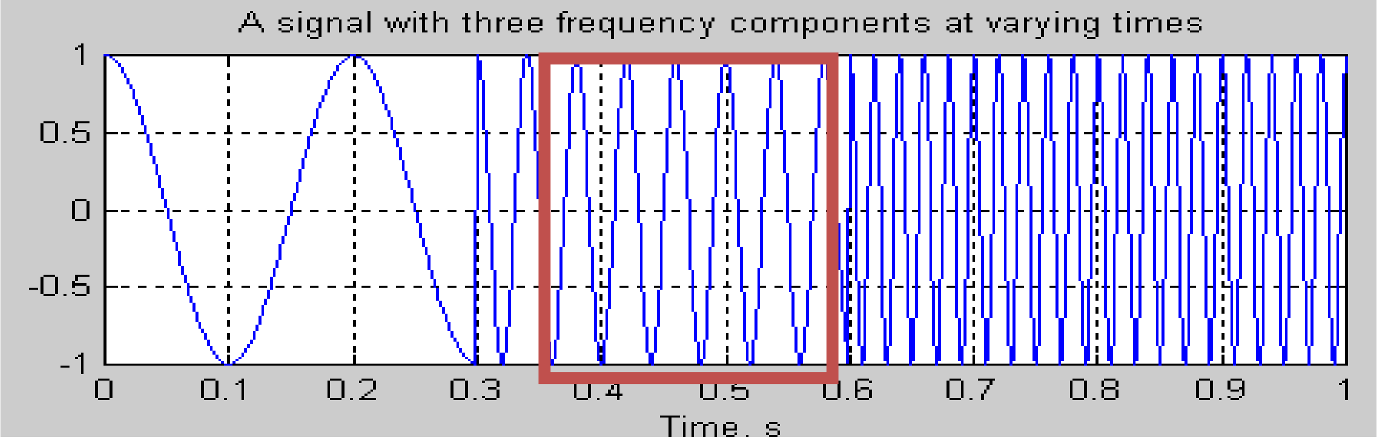




A large number of Fourier components is needed to represent discontinuities.

$$f(x) = \sum_{u=0}^{200} F(u) e^{\frac{j2\pi ux}{N}}, \quad x = 0, 1, \ldots, N-1$$

# Short Time Fourier Transform (STFT)

- Segment signal into narrow time intervals (i.e., narrow enough to be considered stationary) and take the FT of each segment.
- Each FT provides the spectral information of a separate time-slice of the signal, providing simultaneous time and frequency information.



# STFT - Steps

(1) Choose a window of finite length

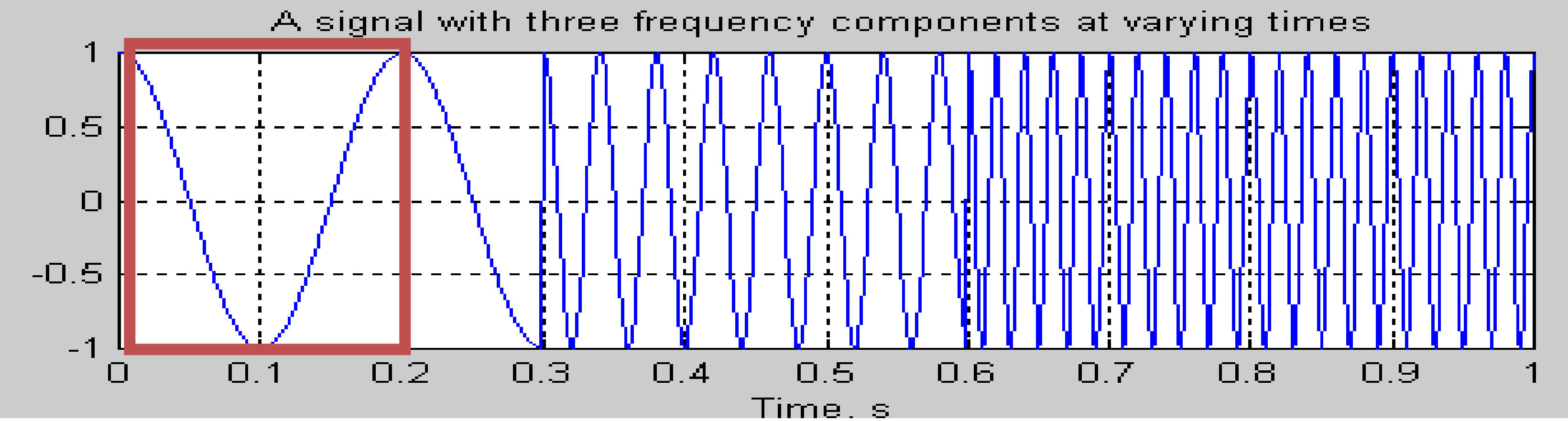
(2) Place the window on top of the signal at t=0

(3) Truncate the signal using this window

(4) Compute the FT of the truncated signal, save results.

(5) Incrementally slide the window to the right

(6) Go to step 3, until window reaches the end of the signal



# STFT - Definition

Time parameter

Frequency parameter

Signal to be analyzed

$$STFT_f^u(t',u) = \int_t \left[ f(t) \cdot W(t-t') \right] \cdot e^{-j2\pi ut} dt$$

2D function

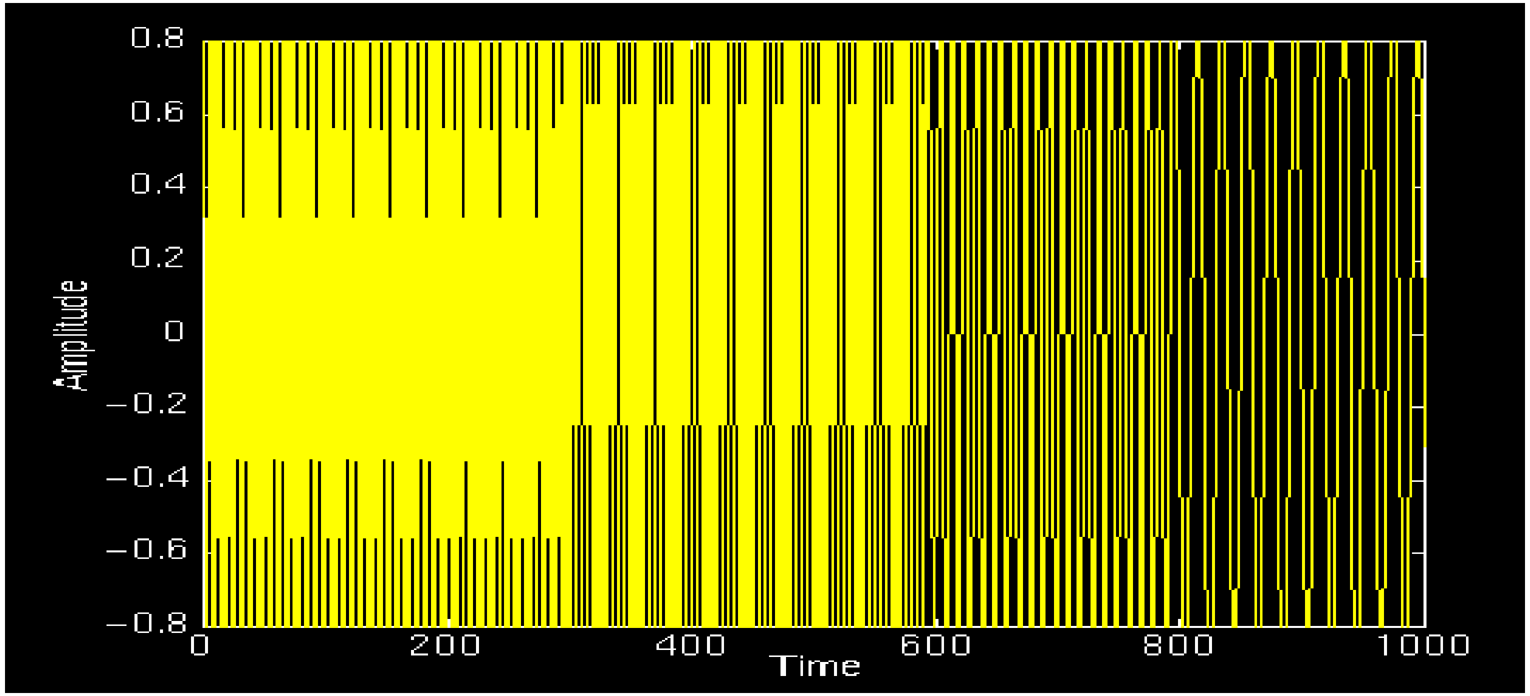
STFT of f(t): computed for each window centered at t=t'

Windowing function

Centered at t=t'

# Example

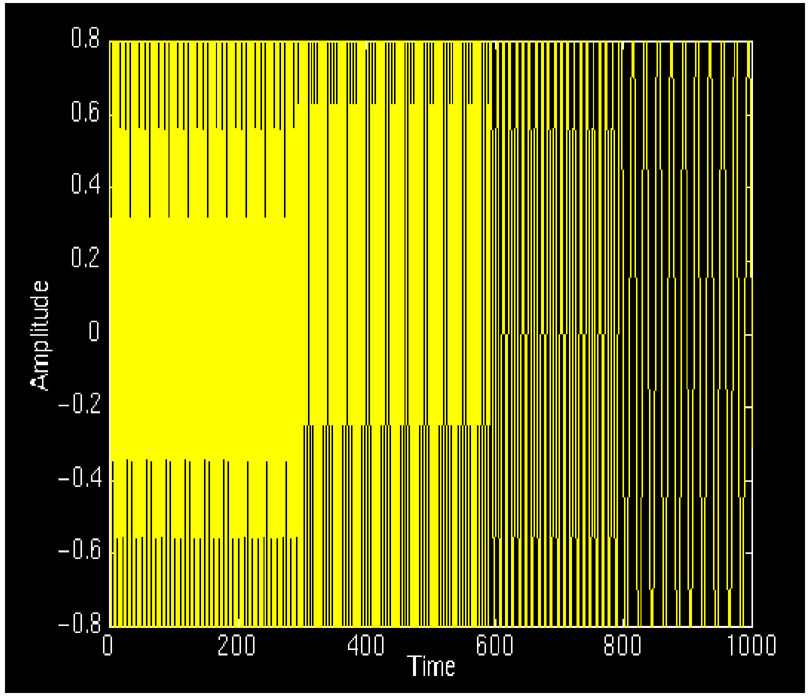
f(t)



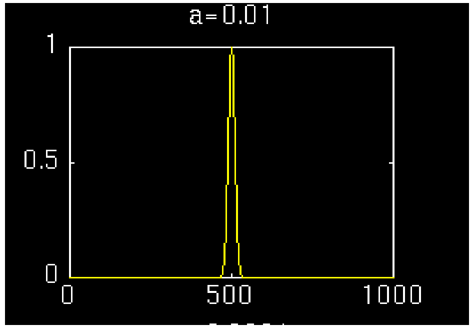

[0 – 300] ms → 75 Hz sinusoid
[300 – 600] ms → 50 Hz sinusoid
[600 – 800] ms → 25 Hz sinusoid
[800 – 1000] ms →10 Hz sinusoid

# Example

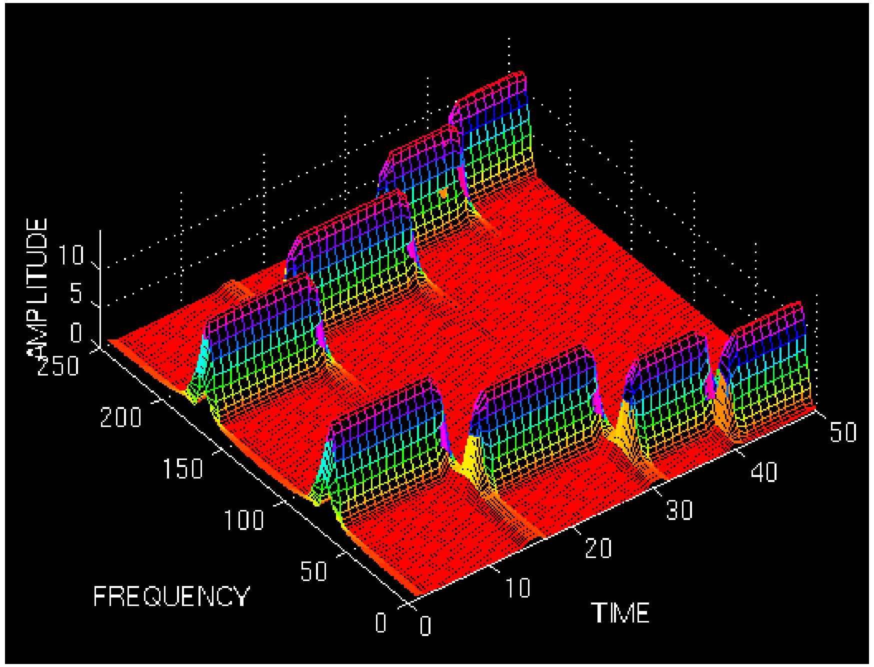
f(t)




W(t)




$$STFT_f^u(t',u)$$

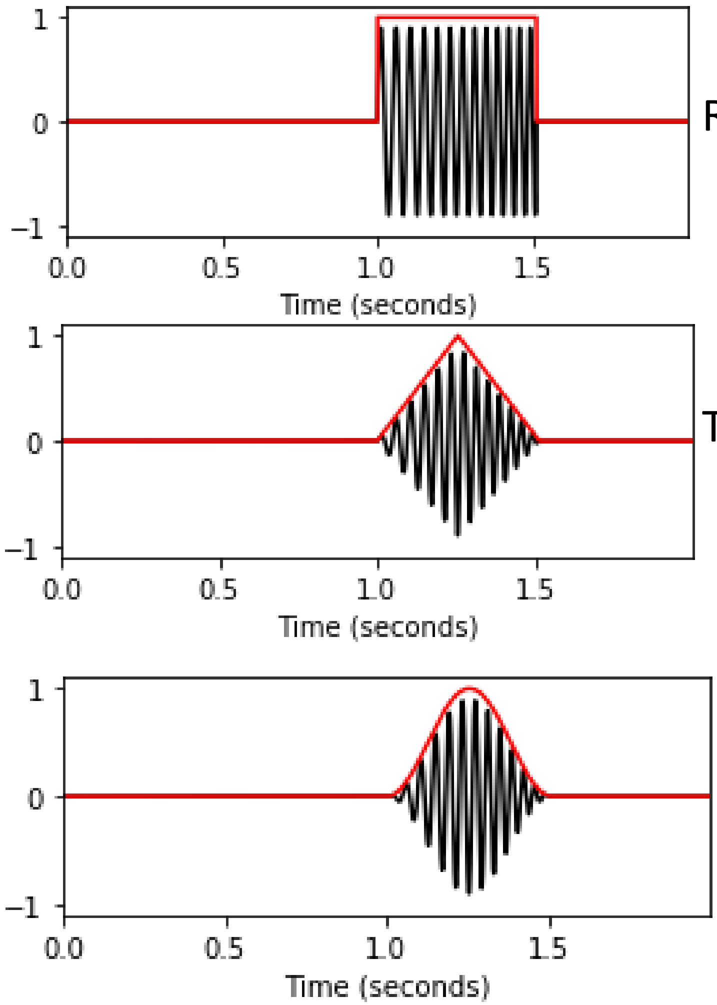



spectrogram

scaled: t/20

# Choosing Window W(t)

- Rectangular windows introduce discontinuities at section boundaries.
- These abrupt changes cause spectral artifacts spread over the entire frequency spectrum.
- To reduce boundary effects, use windows that taper smoothly to zero at the edges:
  - Triangular window
  - Hann window
  - Hamming window
- Smooth windows soften artifacts in the Fourier transform of the windowed signal.
- Some windows (e.g., Hann) may smear frequencies, making the transform appear smoother than the original signal.




Rectangular window

Triangular window

Hann window

# STFT – Window Size

- How does the choice of STFT window size influence the representation of a signal in the time-frequency domain?

# Choosing Window W(t) Size

- How wide should it be?
  - Should be narrow enough to ensure that the portion of the signal falling within the window is stationary.
  - Very narrow windows, however, do not offer good localization in the frequency domain.

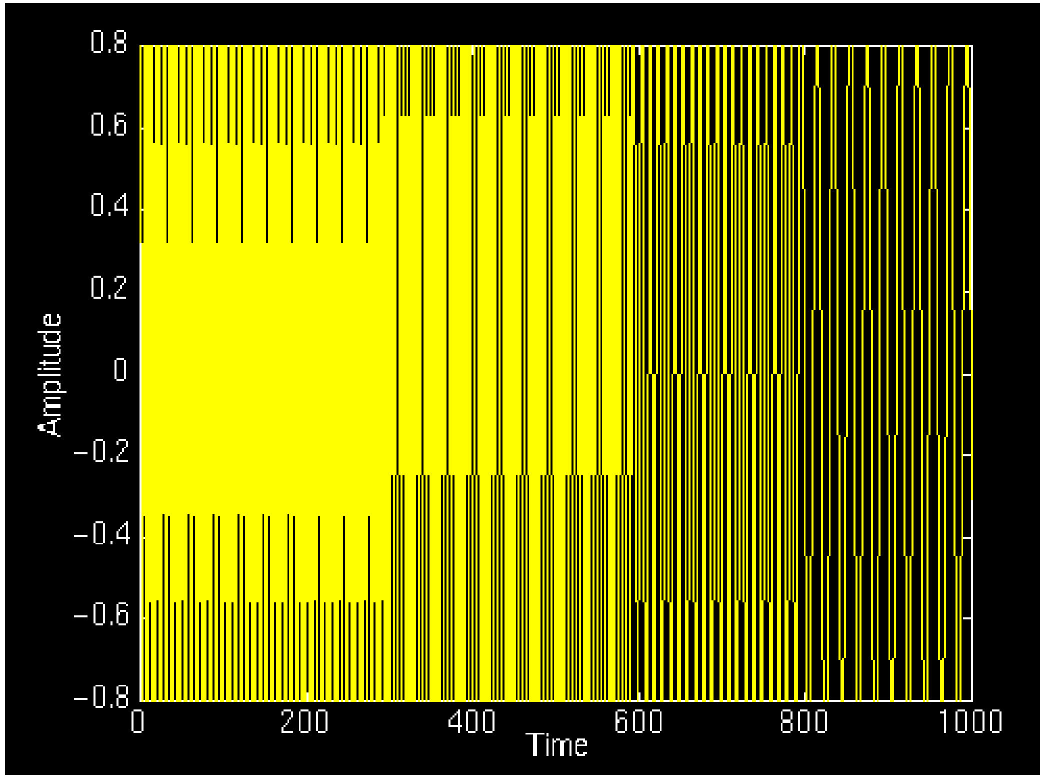
# STFT Window Size

***W(t)* infinitely long**: W(t) = 1 → STFT turns into FT, providing excellent frequency localization, but no time localization.

***W(t)* infinitely short**: W(t) = δ(t) → results in the time signal (with a phase factor), providing excellent time localization but no frequency localization.

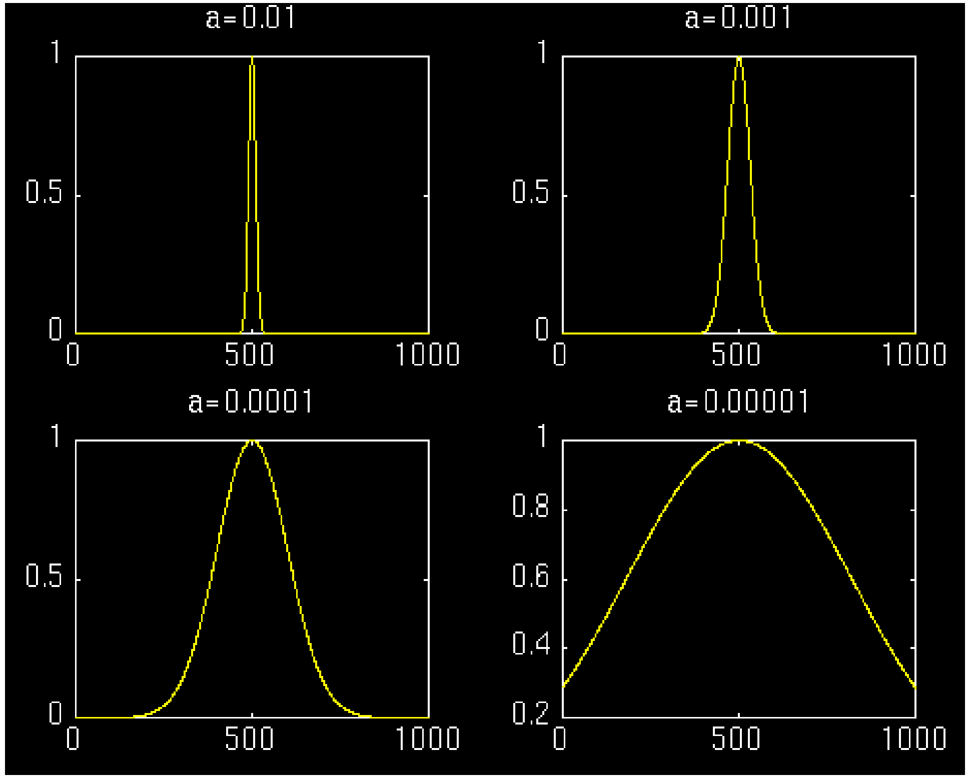
# Example

(four frequencies, non-stationary)



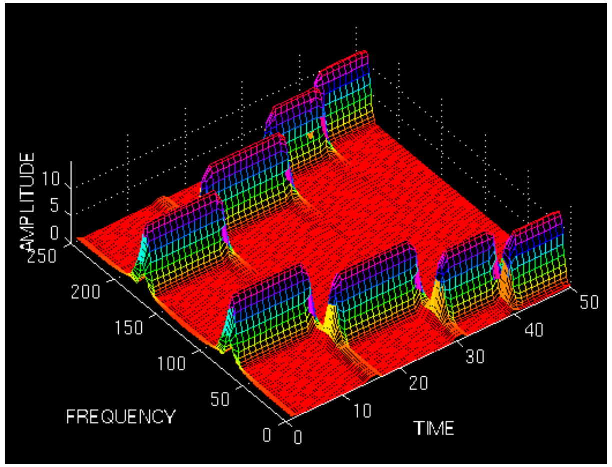

[0 – 300] ms → 75 Hz sinusoid
[300 – 600] ms → 50 Hz sinusoid
[600 – 800] ms → 25 Hz sinusoid
[800 – 1000] ms → 10 Hz sinusoid

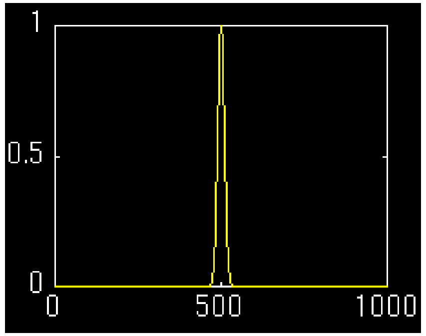
different size windows



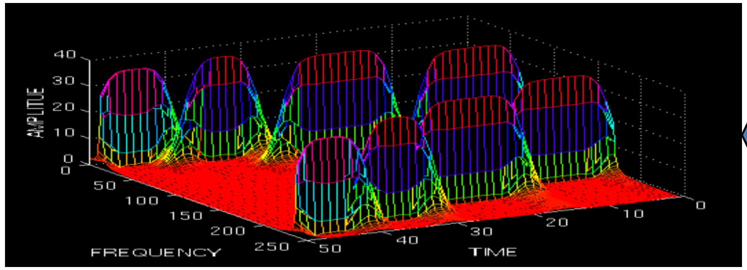
# Example (cont'd)

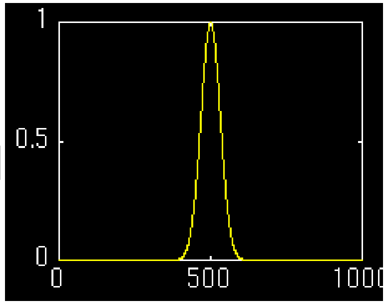
spectrogram

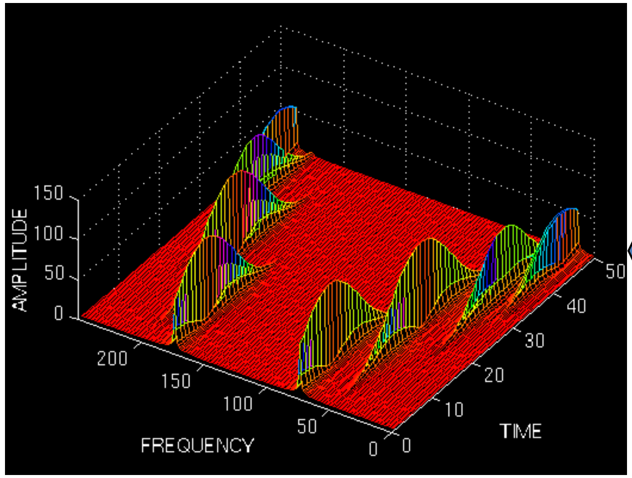


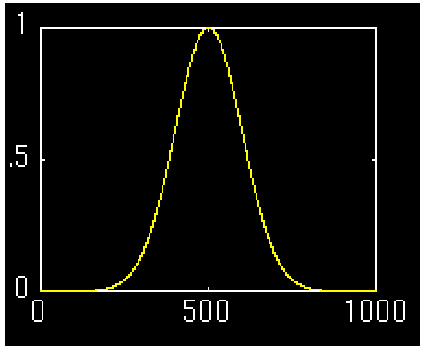



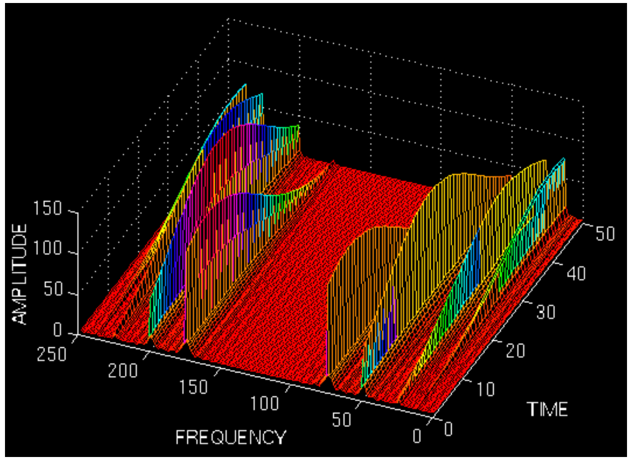

spectrogram

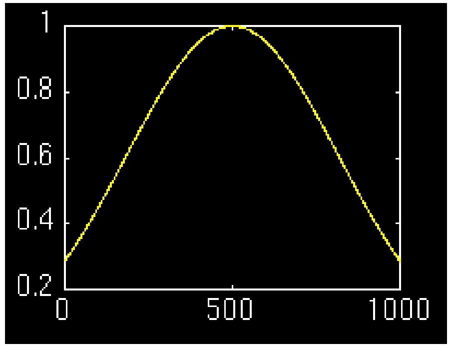


UNIVERSITY OF LINCOLN

# Wavelet transform

# The Chirp Signal

- Chirp Signals
  - They are signals in which the frequency of the signal varies with time.
  - **Can you give instances where a chirp signal will be generated?**
  - Chirp signals are characterized by their linear or nonlinear frequency sweep.
  - Application:
    - Sonar Systems:
      - Chirp signals are employed in underwater acoustics and sonar systems to locate and identify underwater objects.
    - Spectroscopy:
      - Chirp signals can be used in certain spectroscopy techniques to analyze the properties of materials and molecules by observing how the chirp interacts with the sample.

# Demo: The Chirp Signal

**Generate a Linear Chirp**

sampled at 1 kHz for 2 seconds.
The instantaneous frequency is 0 at $t$ = 0 and crosses 250 Hz at $t$ = 1 second.

**Perform the FFT and plot the results**

**Perform the STFT and plot the results**

# The Linear Chirp

```python
import numpy as np
from scipy.signal import chirp, spectrogram
import matplotlib.pyplot as plt
```

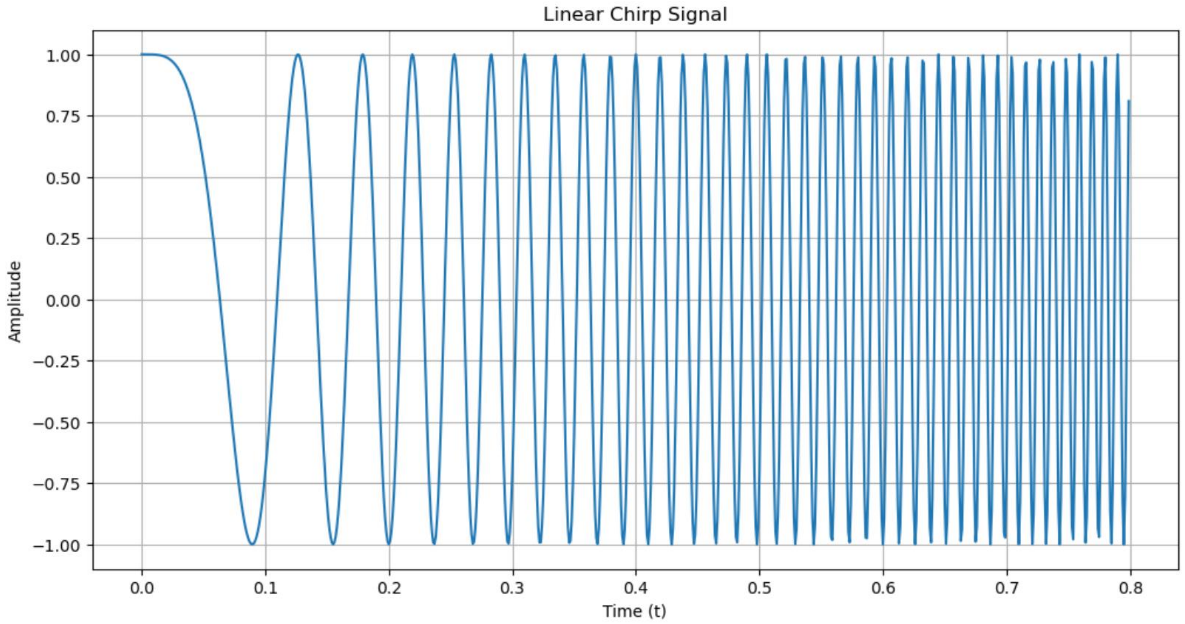
```python
fs = 1000
dt = 1/fs
t = np.arange(0, 2, dt)
```

```python
# method='linear', 'quadratic', 'logarithmic', 'hyperbolic'
```

```python
y_linear = chirp(t, f0=0, f1=250, t1=2, method='linear')
```

```python
# Plot the signals
fig = plt.figure(figsize=(12,6))

plt.plot(t[:800], y_linear[:800])
plt.title('Linear Chirp Signal')
plt.xlabel('Time (t)')
plt.ylabel('Amplitude')
plt.grid(True)
plt.show()
```

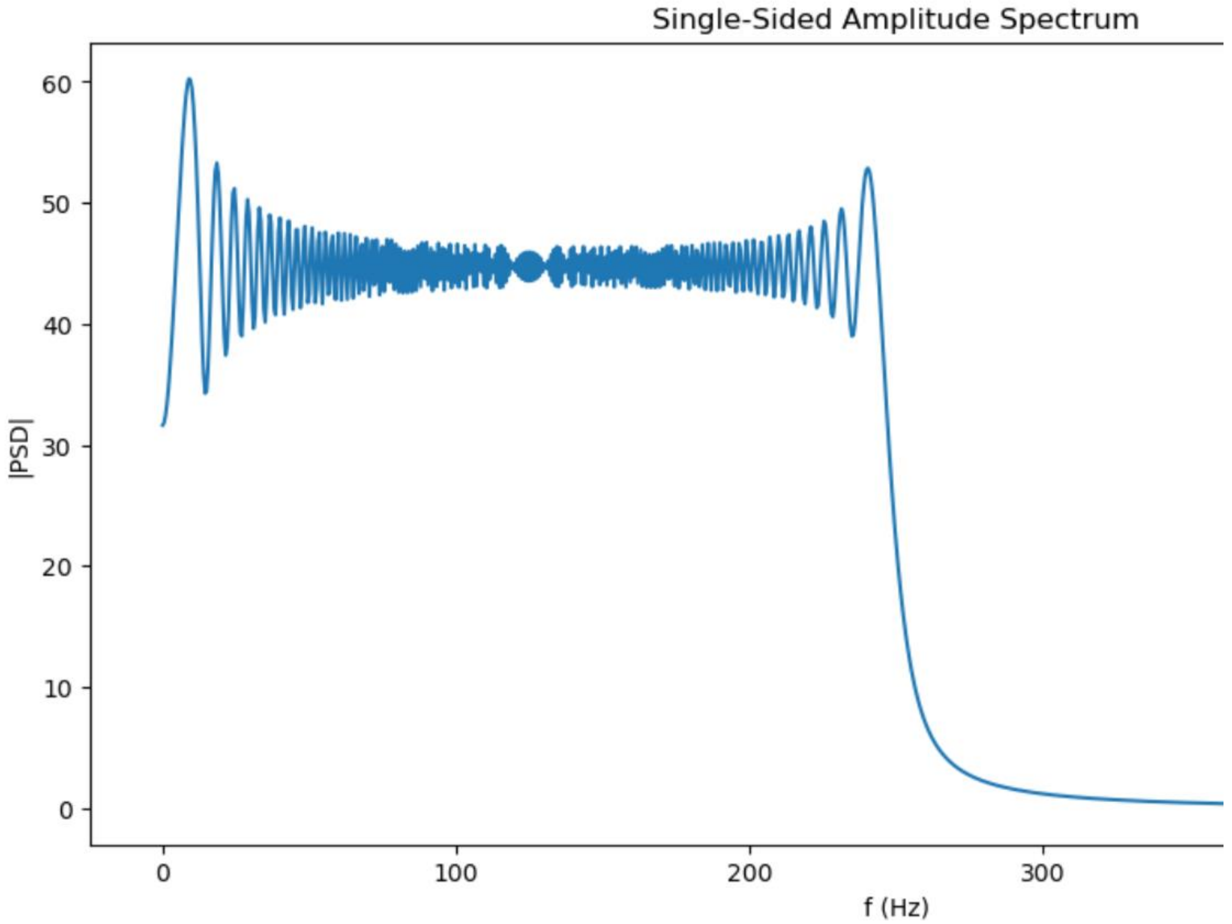


# The FFT of the Linear Chirp

```
from scipy.fft import rfft, rfftfreq, irfft
```

```
# Compute the One Sided FFT
y = rfft(y_linear)
PSD = np.abs(y) # Power spectrum Density
freqs = rfftfreq(t.size, 1/fs) # the frequenc
```

```
# Plot the FFT
fig = plt.figure(figsize=(12,6))
plt.plot(freqs, PSD)
plt.title('Single-Sided Amplitude Spectrum')
plt.xlabel('f (Hz)')
plt.ylabel('|PSD|')
```

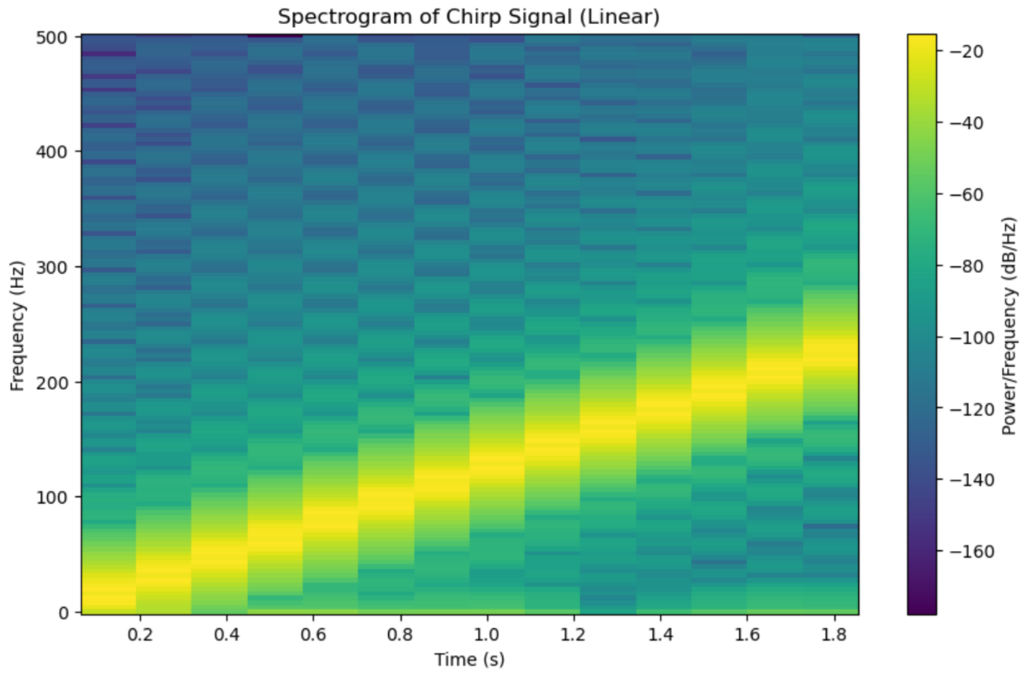


# The STFT of the Linear Chirp

```python
# Define the parameters for the spectrogram
fs = 1 / (t[1] - t[0])  # Sampling frequency
nperseg = 256  # Number of data points in each segment
noverlap = nperseg // 2  # Overlap between segments
```

```python
# Compute the spectrogram
frequencies, times, Sxx = spectrogram(y_linear, fs=fs,
                                      nperseg=nperseg, noverlap=noverlap)
```

```python
# Plot the spectrogram
plt.figure(figsize=(10, 6))
plt.pcolormesh(times, frequencies, 10 * np.log10(Sxx))
plt.title('Spectrogram of Chirp Signal (Linear)')
plt.xlabel('Time (s)')
plt.ylabel('Frequency (Hz)')
plt.colorbar(label='Power/Frequency (dB/Hz)')
plt.show()
```



# Wavelet Transform

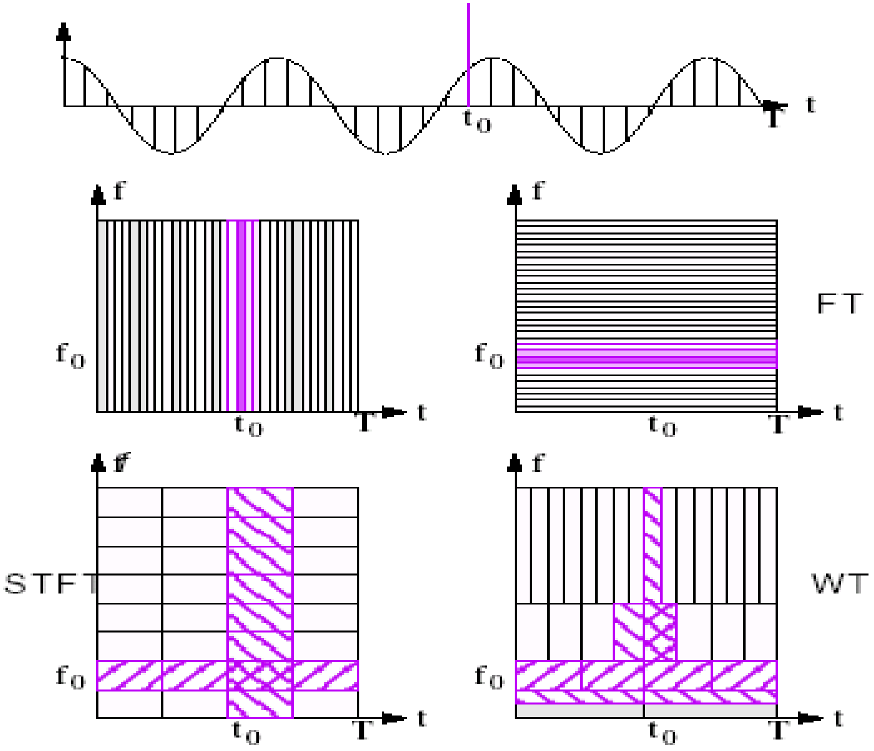
- **Time series** – temporal information of the signal (temporal resolution), but no information of frequency (uncertainty)
- **Fourier Transform** – frequency information of the signal (frequency resolution), but no temporal information(uncertainty)
- **Spectrogram** – Some information on Time and Frequency
- **Wavelet** – Multiresolution analysis –
  - **Low frequencies** last for a long time and do not change much over time
  - **High frequencies** lasts for a short period
  - At lower frequencies, the temporal resolution is coarse
  - At high frequencies, the temporal resolution is fine.

# Wavelet Transform

- Time-frequency analysis techniques
  - Wavelet transforms and
  - Short-Time Fourier Transforms (STFT)
- When to use Wavelets
  - **Non-Stationary Signals**
    - STFT uses Fixed size window making it less effective for capturing time-varying behaviour
  - Biomedical Signal Analysis:
    - EEG and ECG, where signal characteristics change over time
  - pitch detection in audio and signal processing
  - signals with both high and low-frequency components.

# Fourier Transform and DWT

- Example:



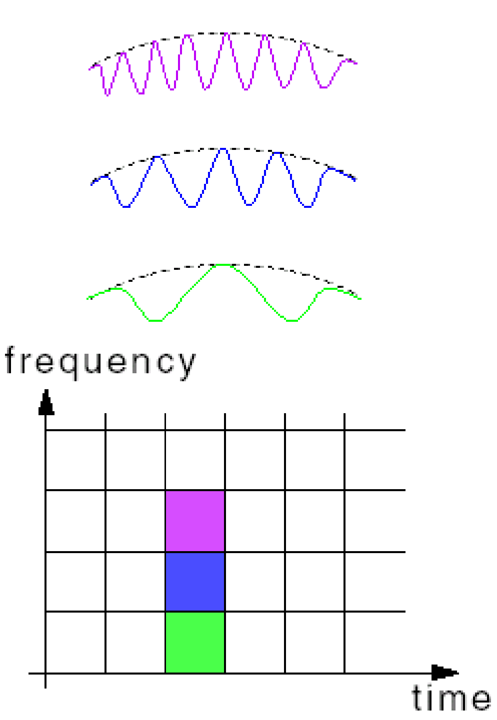

$$x[n] = \cos 2\pi f_0 + A\delta[n - t_0]$$

- FFTs captures global frequency information, which is not suitable for some signals.
  - For example, Electrocardiography (ECG) where signals have short intervals of characteristic oscillation.

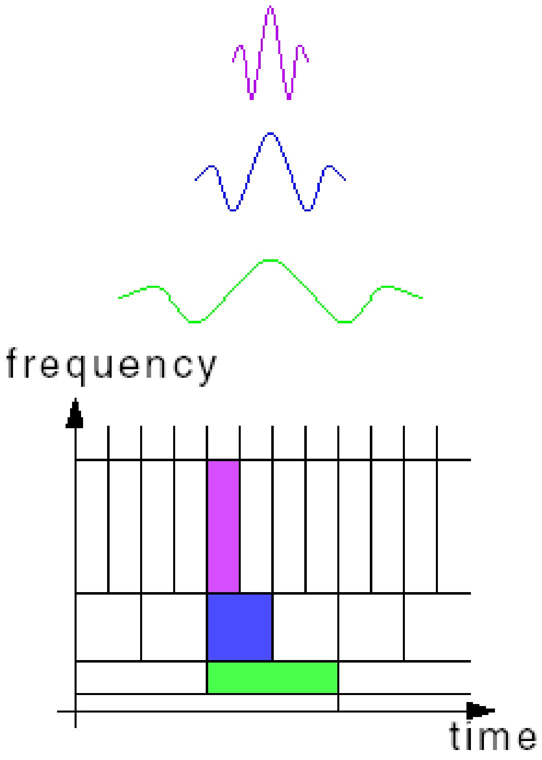
# STFT and Wavelets

1. STFT is uniform yet CWT is not.

2. STFT involves Fourier transforms but CWT only requires an **orthogonal filter bank**

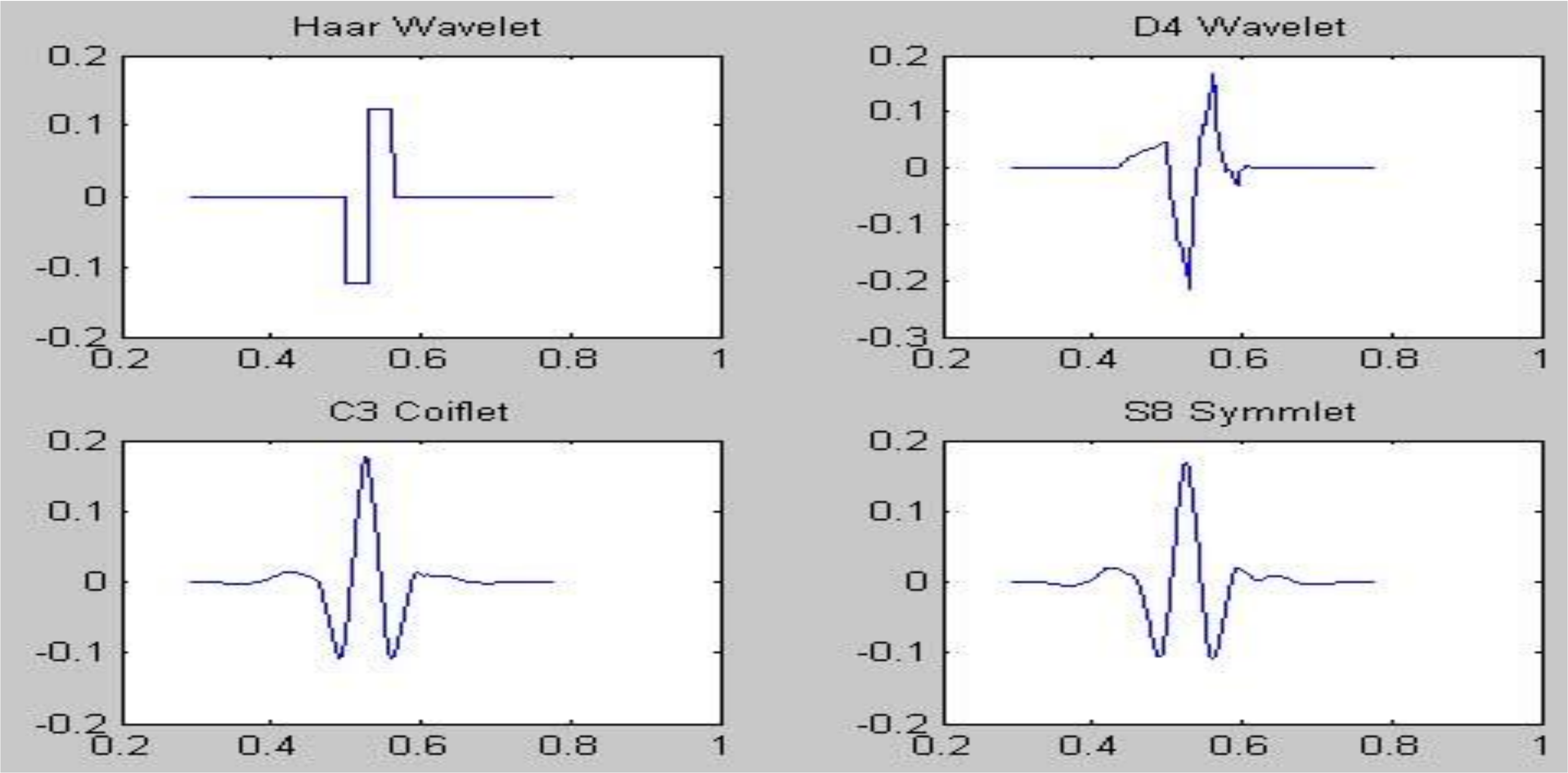



short-time Fourier transform



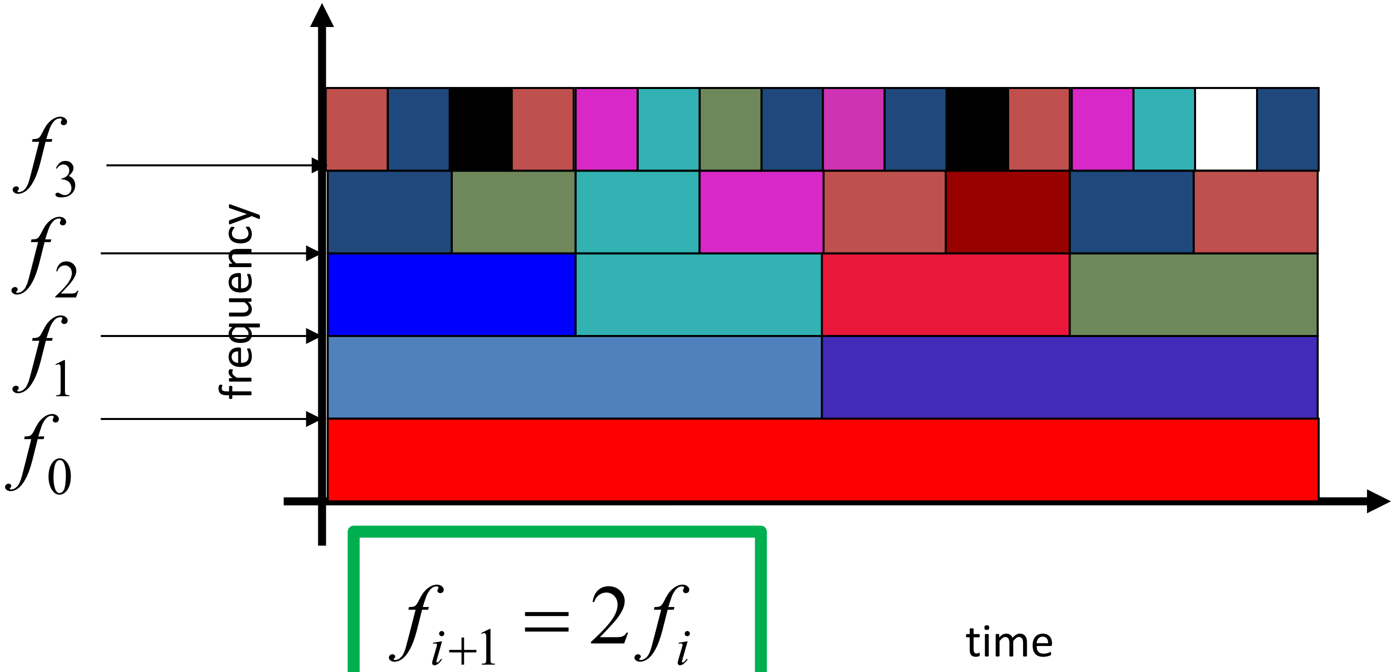

wavelet transform

# Families of wavelets



# Wavelets: Frequency Partitioning

- **Frequency partitioning** (others are possible!)




$$f_{i+1} = 2 f_i$$

# Introduction - Mathematical Form

- Class of basis functons, known as wavelets, incorporate two parameters:
  1. one for *translation* in time
  2. another for *scaling* in time

Another definition: $$\psi_{s,\tau}(t) = \frac{1}{\sqrt{s}}\psi\left(\frac{t-\tau}{s}\right)$$

A wavelet is an oscillating function whose energy is concentrated in time to better represent transient and nonstationary signals.

# Continuous Wavelet Transform (CWT)

$$X_{WT}(\tau, a) = \frac{1}{\sqrt{|a|}} \int x(t) \Psi^* \left(\frac{t-\tau}{a}\right) dt$$

- The transformed signal is a function of two variables:
  - $\tau$ **-** *translation* parameter
  - *a - scale (or dilation)* parameter
- $\Psi$ **(t)** is called the mother wavelet

# Continuous Wavelet Transform (CWT)

- In a similar way we can think of the above expression as asking the question

  *" Is there one these functions*

- $$\Psi(\frac{t-\tau}{a})$$

  *in the given signal and where is it located?".*

- The expression is a convolution, and this leads to a filtering interpretation

# Continuous Wavelet Transform (CWT)

- The mathematical form of the representation is very similar to the STFT

$$X_{WT}(\tau,a)=\frac{1}{\sqrt{|a|}}\int\Psi*(\frac{t-\tau}{a})x(t)dt$$

- Compare with the STFT we looked at earlier

$$X_{ST}(j\omega_0,\tau)=\int_{-\infty}^{\infty}(e^{j\omega_0 t})^{*}g(t-\tau)x(t)dt$$

- The similarities are strikingly obvious!!!
- **What similarities can you identify?**

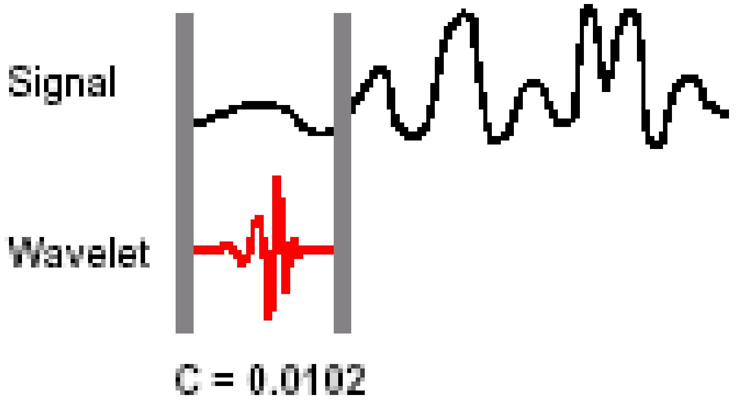
# How to Wavelet Transform

- Five Easy Steps to a Continuous Wavelet Transform

- The continuous wavelet transform is the **sum over all time of the signal** multiplied **by scaled, shifted versions of the wavelet**. This process produces wavelet coefficients that are a function of scale and position.

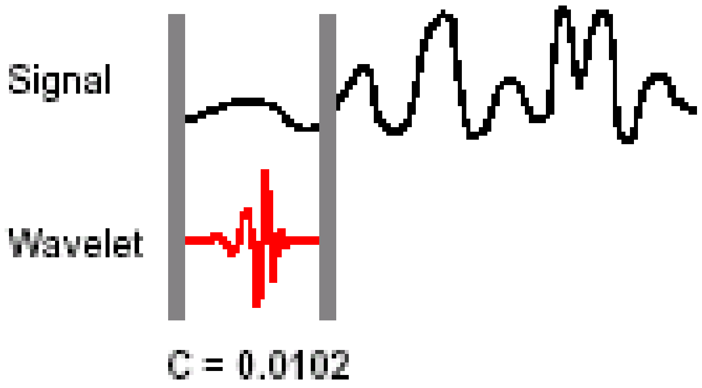
# How to Wavelet Transform

- **1** Take a wavelet and compare it to a section at the start of the original signal.



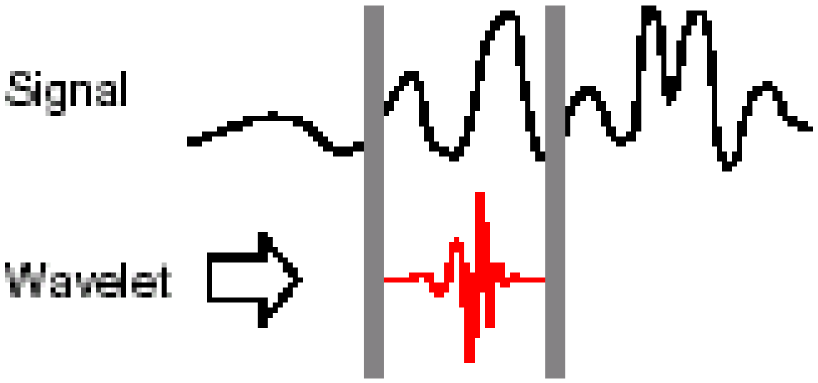
# How to Wavelet Transform

- **2** Calculate a number, C, that represents how closely correlated the wavelet is with this section of the signal. The higher C is, the more the similarity. Note that the results will depend on the shape of the wavelet you choose.



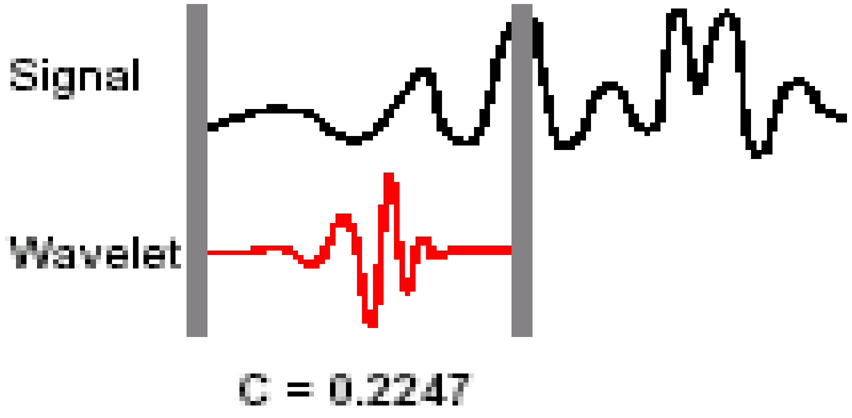
# How to Wavelet Transform

- **3** Shift the wavelet to the right and repeat steps 1 and 2 until you've covered the whole signal.



# How to Wavelet Transform

- **4** Scale (stretch) the wavelet and repeat steps 1 through 3




- **5** Repeat steps 1 through 4 for all scales.

# Demo: The Chirp Signal

- Generate a Linear Chirp
  - sampled at 1 kHz for 2 seconds.
  - The instantaneous frequency is 0 at $t$ = 0 and
  - crosses 250 Hz at $t$ = 1 second.

- Perform the FFT and plot the results
- Perform the STFT and plot the results

# DEMO: The CWT of the Linear Chirp

# Summary

- Short Time Fourier Transform
- Wavelet Transform
- Examples

# Questions

?

# Further Reading

- You should read this text:
  - Smith, Julius Orion. *Spectral Audio Signal Processing*. https://ccrma.stanford.edu/~jos/sasp/, online book, 2011 edition, accessed October 2025.

- Specifically, you should focus on the sections
  - Time-Frequency Displays
  - Wavelet Filter Banks