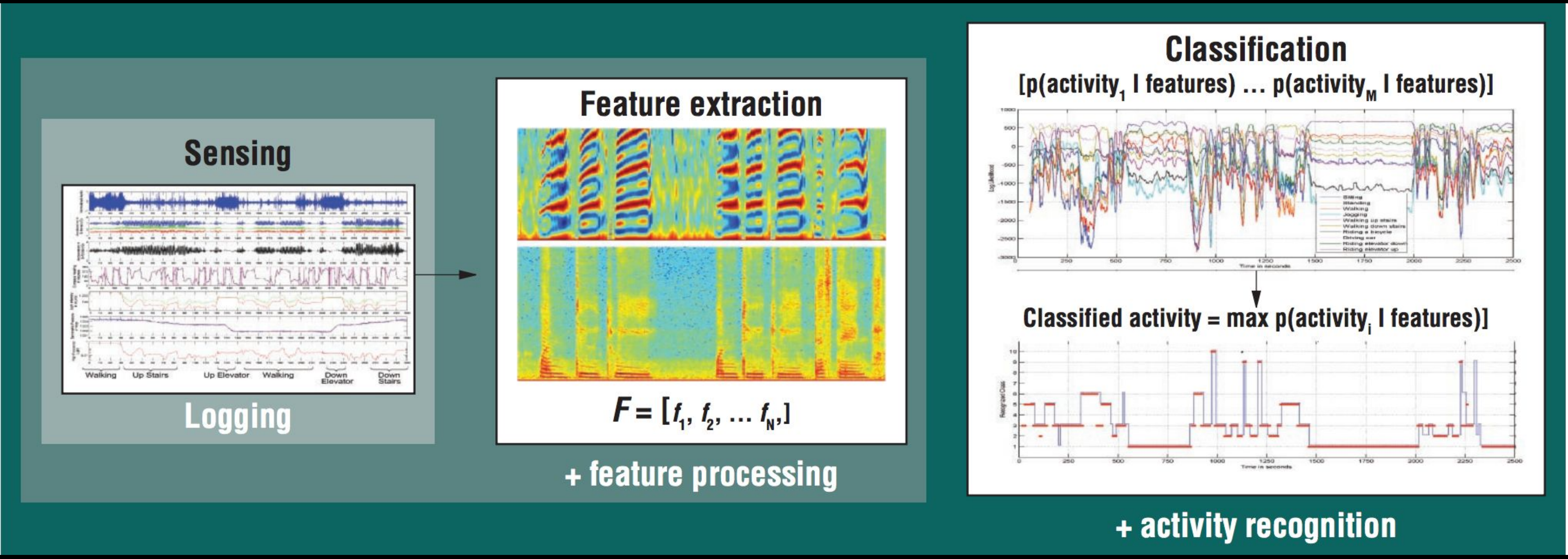
COMPSCI 590U: Feature Engineering and Building Classifiers


Sensing
Walking
Up Stairs
Up Elevator
Walking
Down Elevator
Down Stairs
Logging
Feature extraction
F = [f1, f2, … fN,]
+ feature processing
Classification
[p(activity1 l features) … p(activityM l features)]
Classified activity = max p(activityi l features)]
+ activity recognition

Why do Feature Engineering?

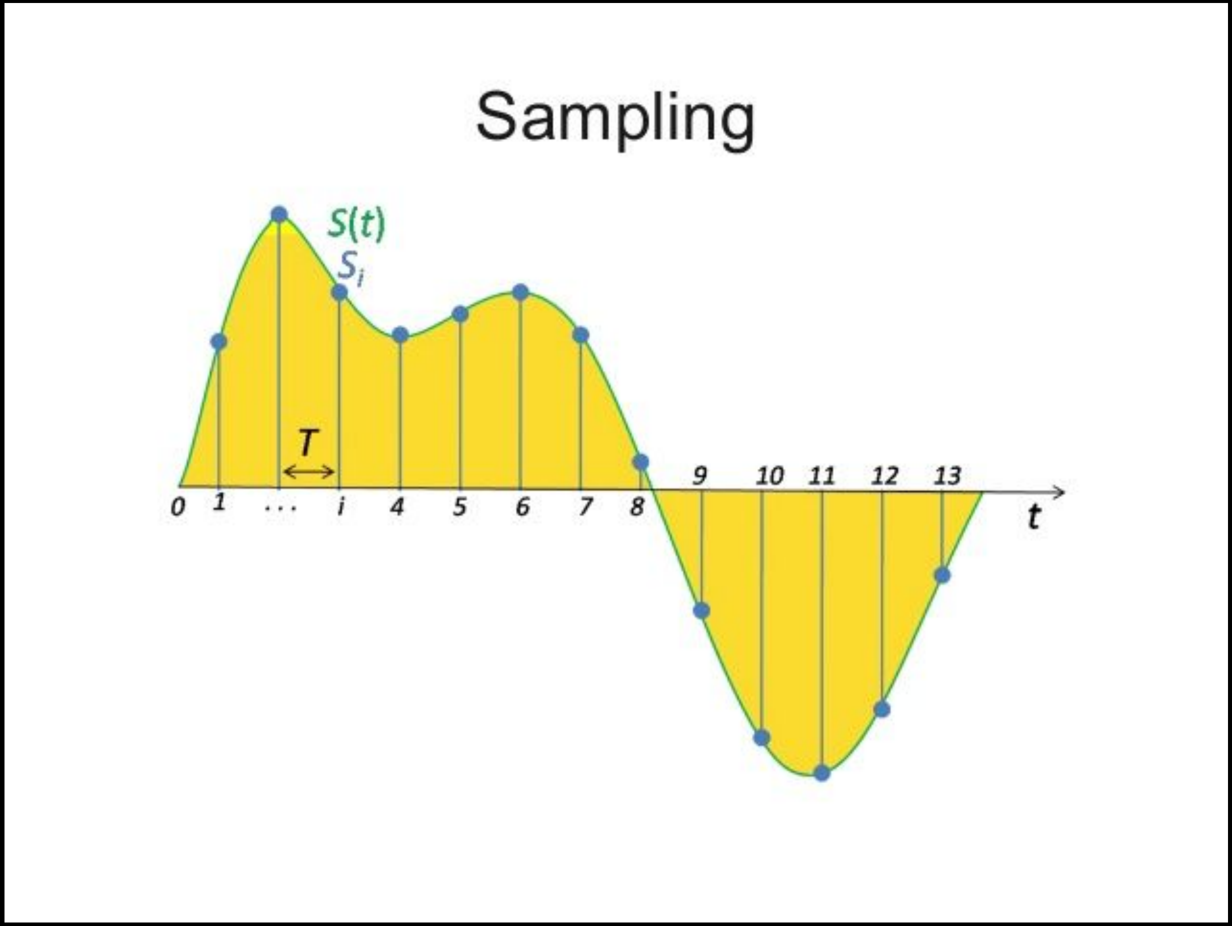
- Raw recorded data is typically not the most useful variables that characterizes what we want to detect/model.
- Transformation of the raw data is required.
- The primary way to inject human knowledge into the recognition model

Features

- Properties of the variable that we think will help distinguish or describe the target class/label of interest
- This can be a function, transformation or combination of the raw data

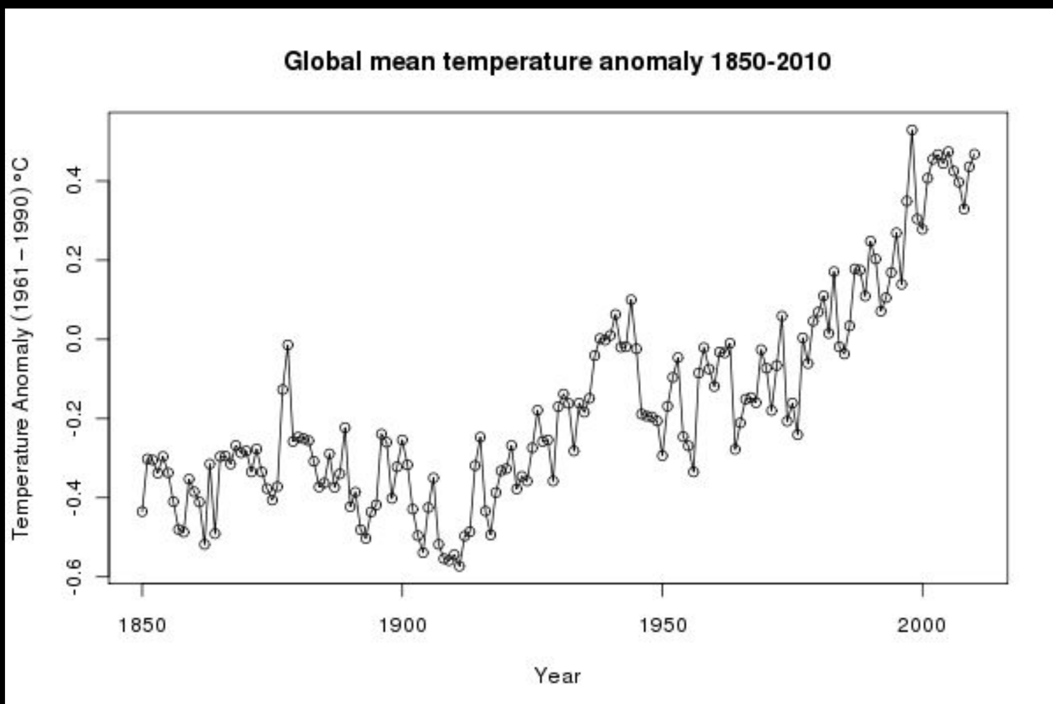
Recording the raw data

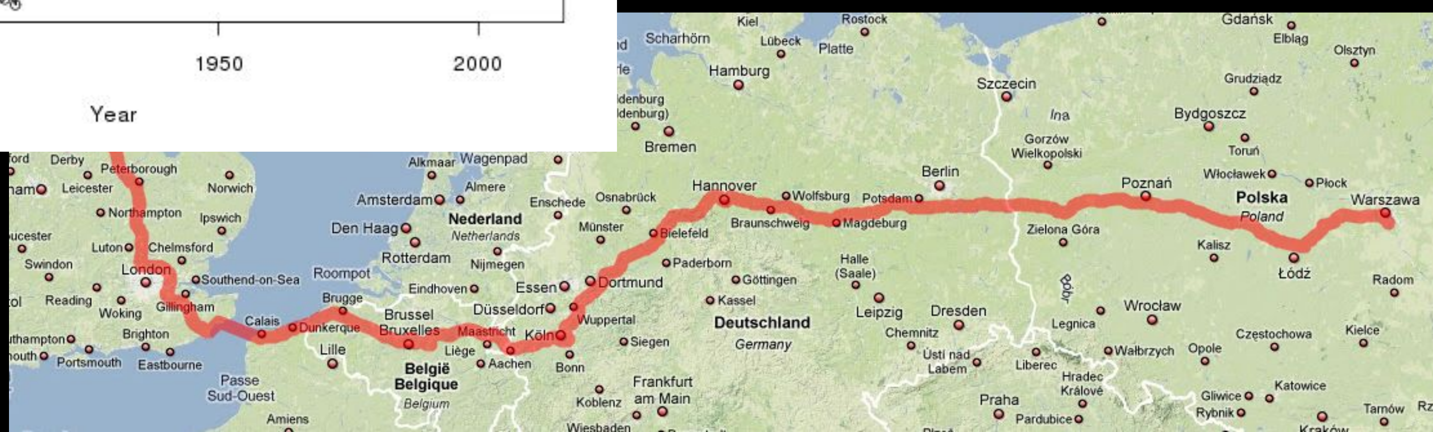
- Making an observation of a variable



Recording the raw data

- Making an observation of a variable





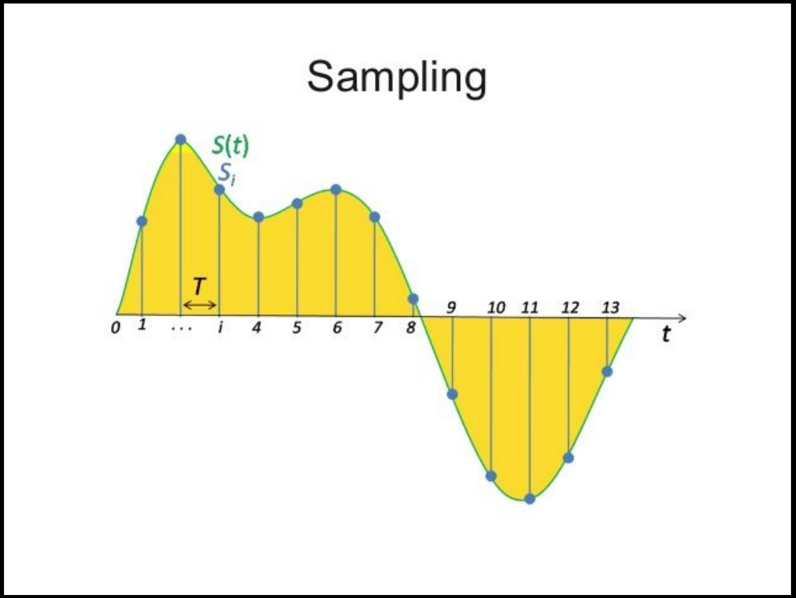
Sampling Rate/Frequency

- The number of times per second a continuous variable is recorded.

  fs = number of samples/second

- What is Sampling Period?

  T = 1/fs



Sampling Rate/Frequency

- A higher sampling rate allows a better approximation of the underlying continuous variable.

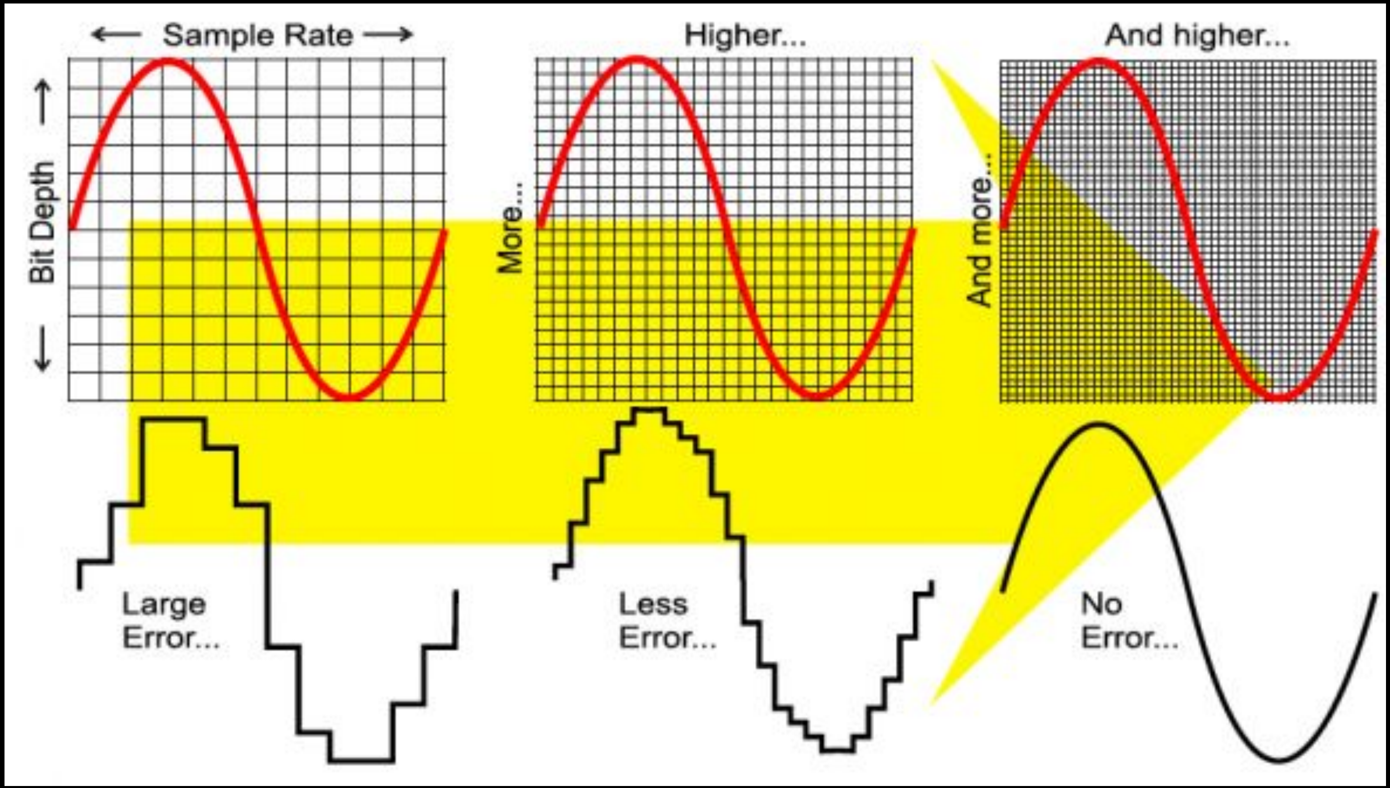


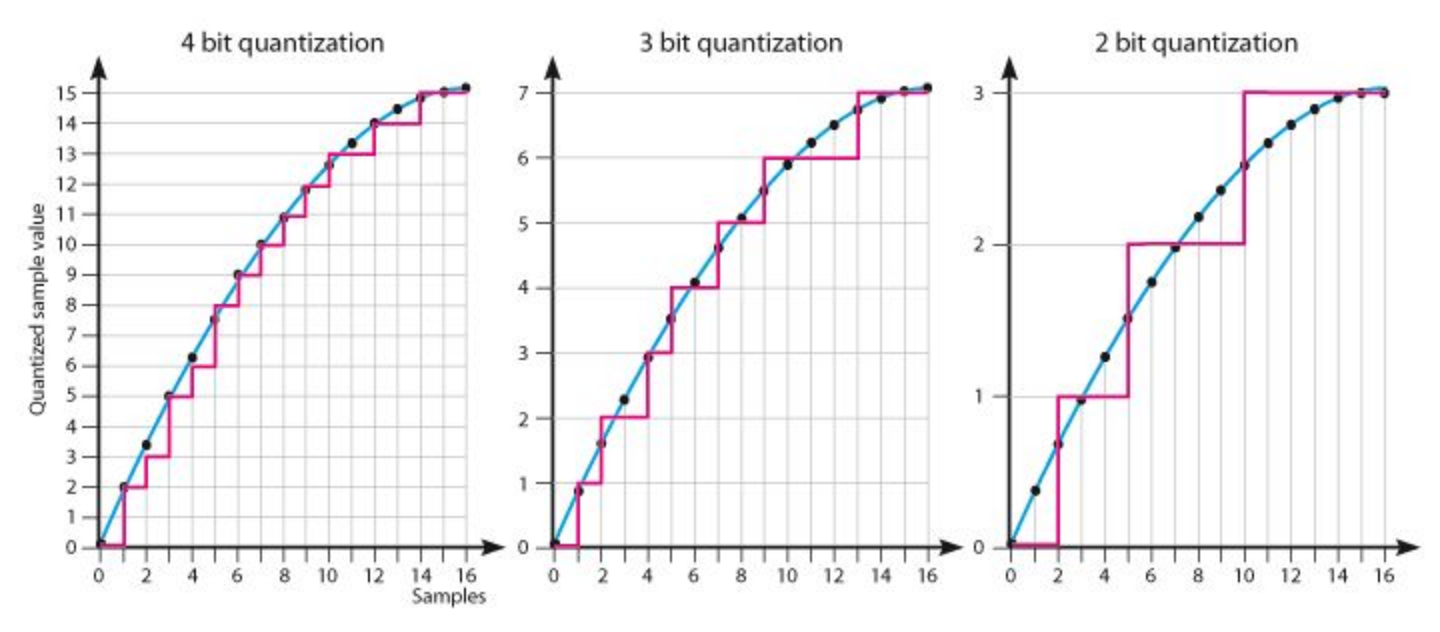
Image Source: https://www.izotope.com/en/learn/digital-audio-basics-sample-rate-and-bit-depth.html

Quantization

- In electronics, an analog-to-digital converter (ADC) is a system that converts an analog signal, such as a sound picked up by a microphone or light entering a digital camera, into a digital signal.
- Bit depth or resolution of the ADC refers to the number bits used to convert the analog voltage or current value to a digital signal.
- The resolution also indicates the number of discrete values it can produce over the range of analog values.
- For example, an ADC with a resolution of 8 bits can encode an analog input to one in 256 different levels ($2^8 = 256$). The values can represent the ranges from 0 to 255 (i.e. unsigned integer) or from –128 to 127 (i.e. signed integer).

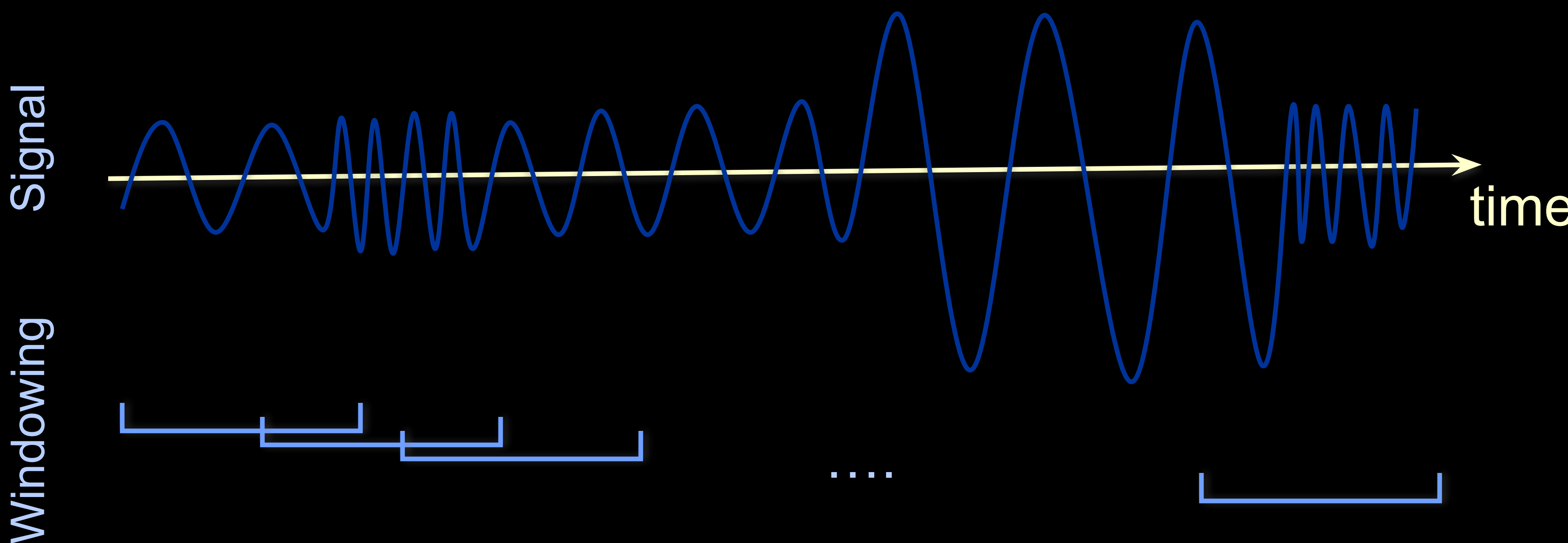
Quantization

- With the increase of resolution or bit depth, quantization error decreases.



Windowing

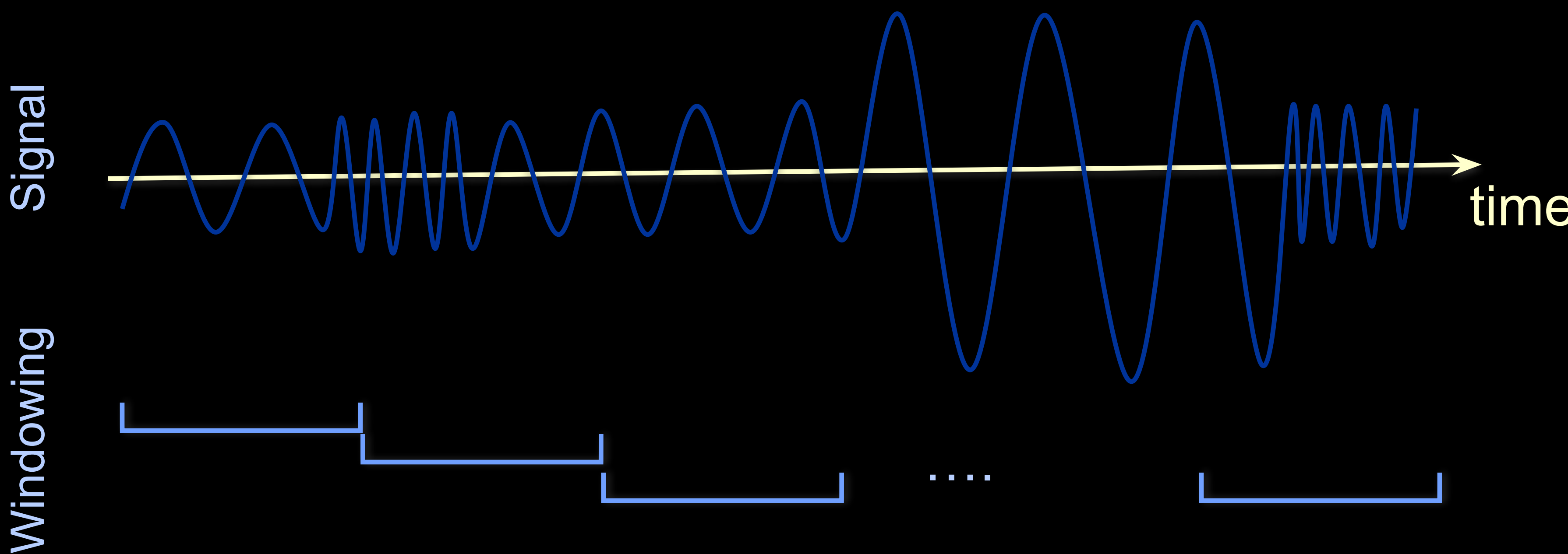
- A method of taking data from predefined intervals within the signal



Overlapping Window

Windowing

- A method of taking data from predefined intervals within the signal



Non-overlapping Window

Considerations for window size selection

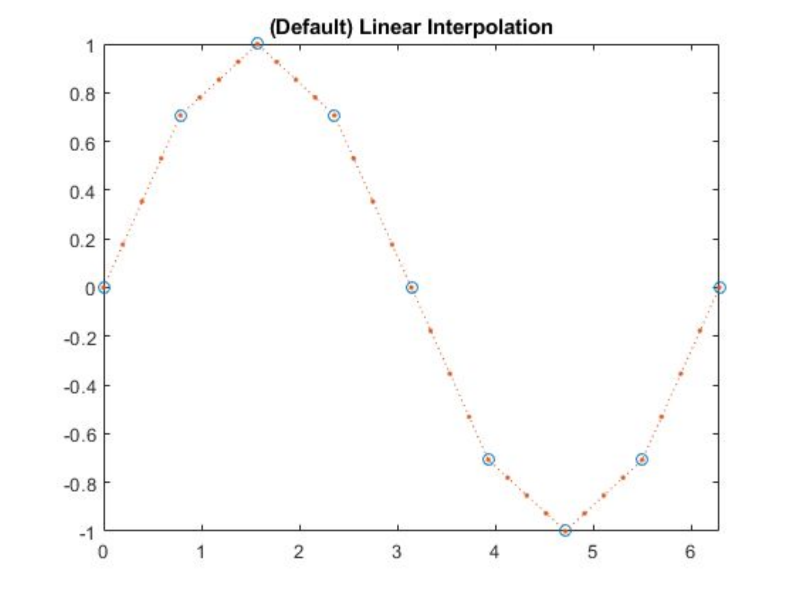
	Small Window	**Large Window**
Temporal Resolution	• High Temporal Resolution • Can track sudden changes in time domain signal.	• Low Temporal Resolution • Fail to track sudden changes • Has smearing/smoothing effect
Frequency Resolution	• Low Frequency Resolution • Fail to resolve two/more tightly spaced frequencies	• High Frequency Resolution • Can resolve two/more tightly spaced frequencies

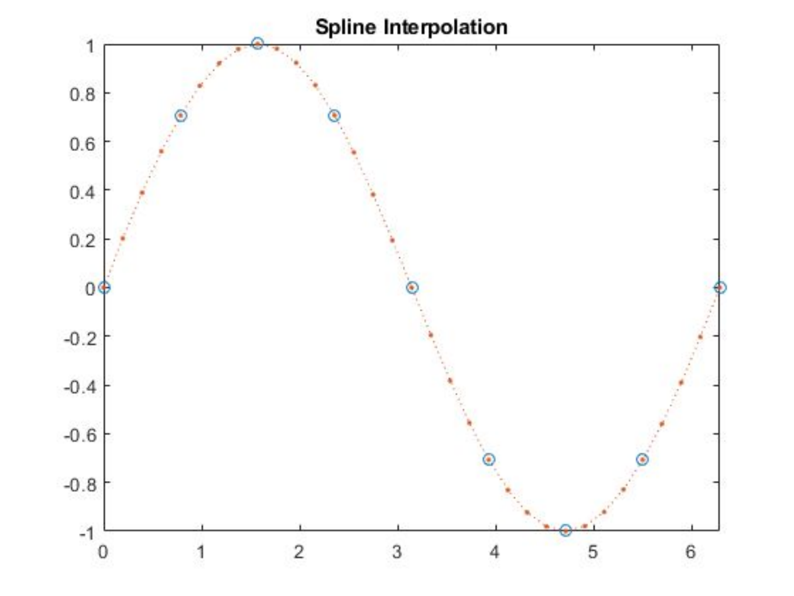
Resampling and Interpolation

- Resampling is used to either increase the sample rate or decrease it.
- Interpolation is the process of calculating values between sample points.
- Due to device/sensor heterogeneity or due to operating system preferences, you may have a certain sensor data that has been collected with different sampling frequencies.
- Before feature extraction, you want to ensure that all the data within a certain sensor stream has the same sampling rate.

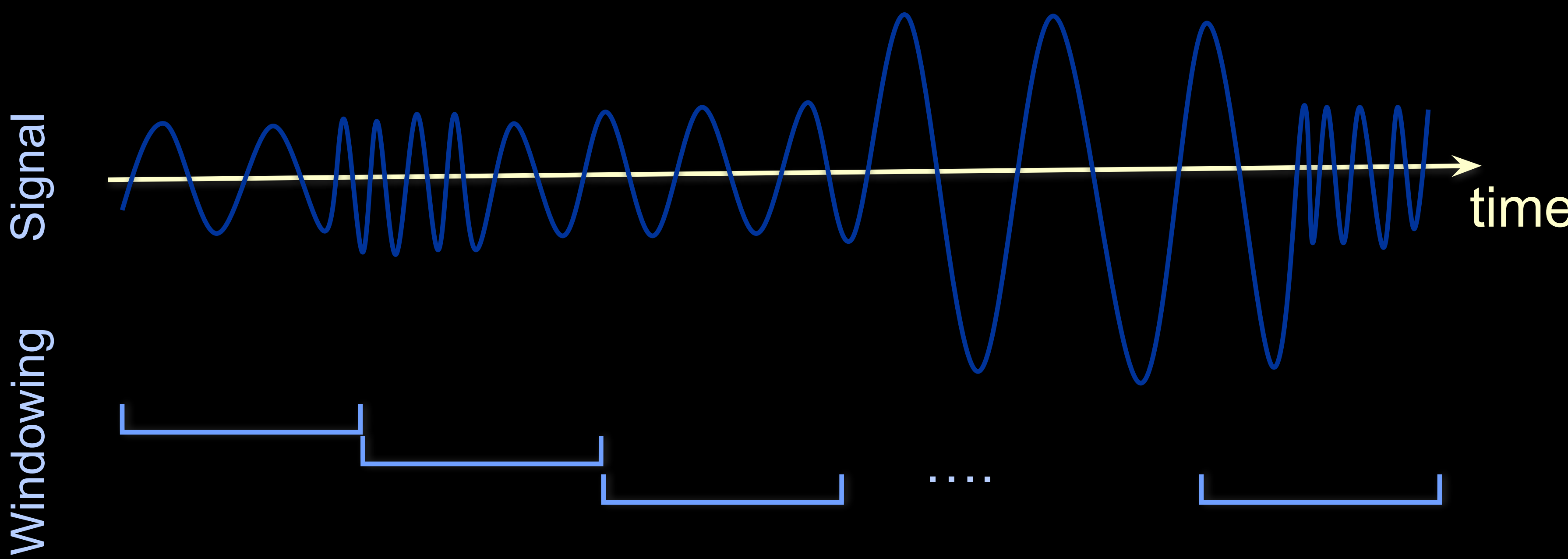
Resampling and Interpolation

- interp1() function in Matlab allows you to achieve 1D interpolation with a wide variety of interpolation methods along with optional extrapolation criteria.





Feature Extraction



$f([x(1), x(2), .. x(n)])$

Two types of functions:

1. Time Domain Functions
2. Frequency Domain Functions

Examples of Time Domain Functions

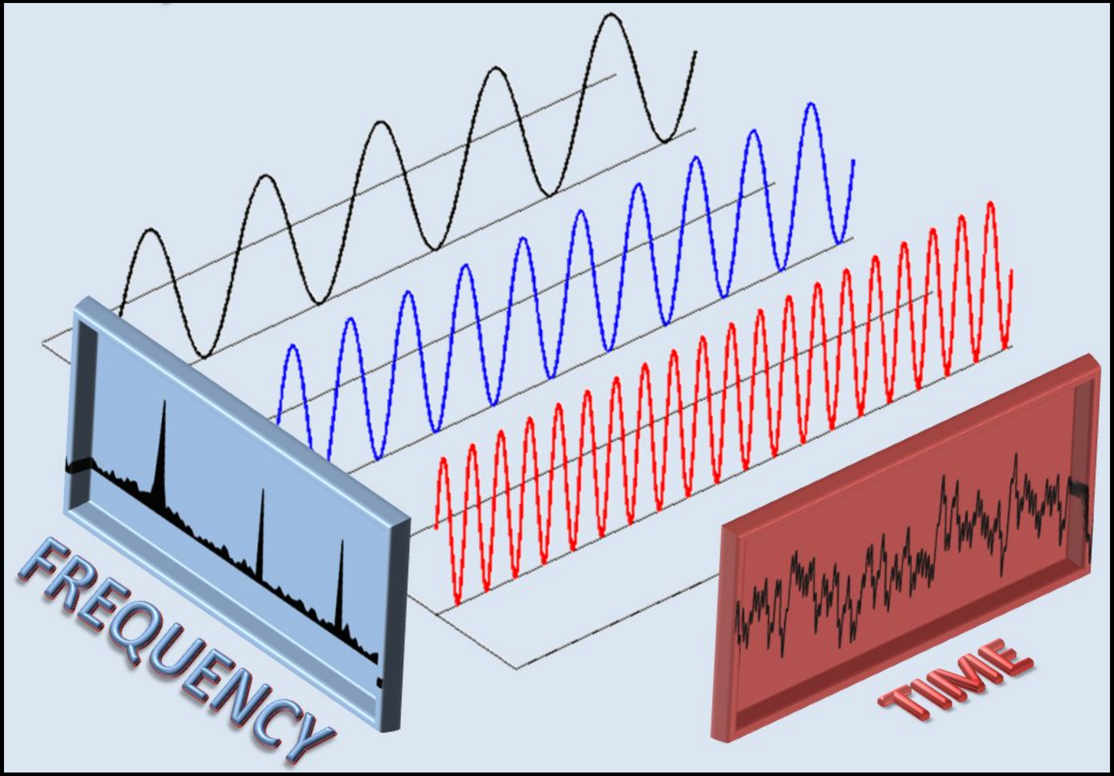
- Measure of average: *Mean*, mod, quartiles, percentiles, median
- Measure of spread: Variance, standard deviation, range
- Rates of change (first and second order differentials)
- Shape (e.g., slope of the raw waveform)
- Extreme values: *Max*, *Min*

Examples of Frequency Domain Functions

- Fourier Transformation
  - raw spectra
  - Spectral features
  - filter bank features
  - Nonlinear transformations (e.g., MFCC)
- Fundamental frequency

Fourier Analysis

- Fourier Analysis
  - All signal composed of linear combination of sinusoids.



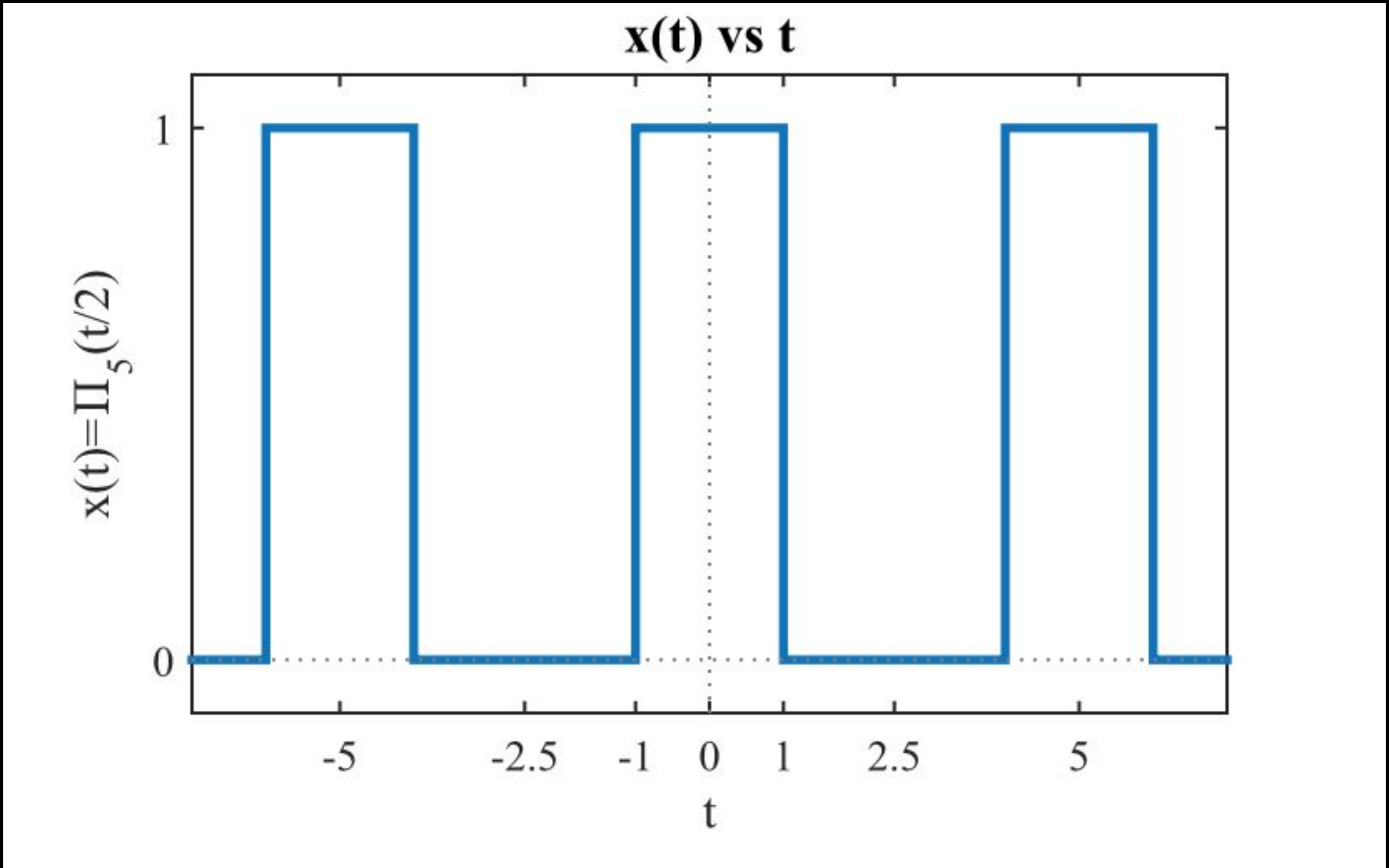
Fourier Series

- For periodic time domain signal $x_T(t) = x_T(t+T)$

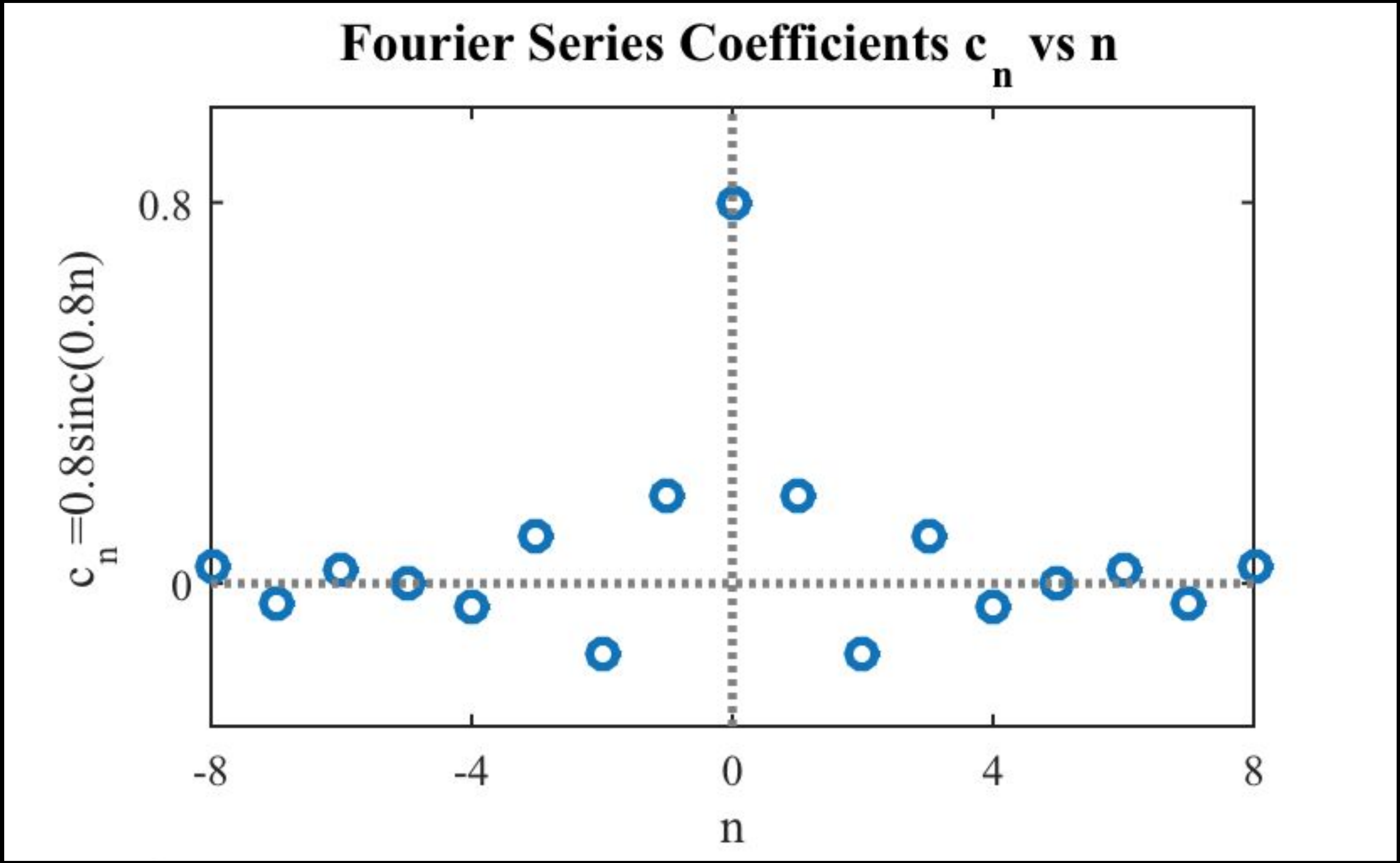
Synthesis: $$x_T(t) = \mathcal{F}^{-1}[X[k]] = \sum_{k=-\infty}^{\infty} X[k]e^{jk\omega_0 t}$$

Analysis: $$X[k] = \mathcal{F}[x_T(t)] = \frac{1}{T}\int_T x_T(t)e^{-jk\omega_0 t}dt \quad (k = 0, \pm 1, \pm 2, \cdots)$$

Fourier Series Example



Fourier Series Example



Fourier Transfom

- For aperiodic time domain signal

$$x_T(t) = x_T(t+T) \qquad \& \qquad T \to \infty \Longrightarrow \omega_0 = 2\pi/T \to 0$$

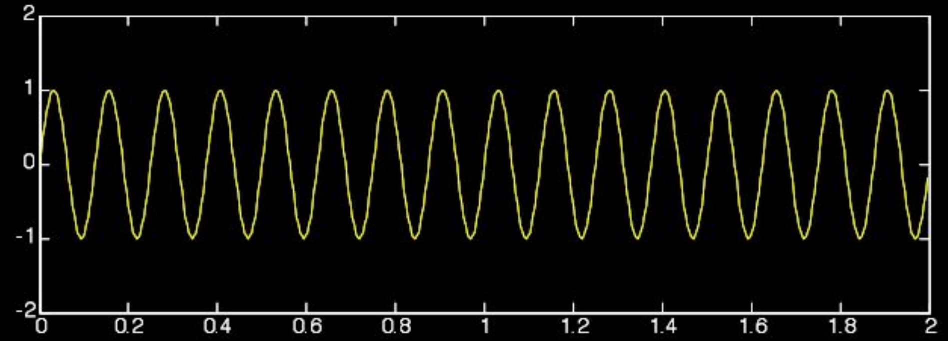
Synthesis:

$$x(t) \triangleq \lim_{T\to\infty} x_T(t) = \lim_{\omega_0\to 0} \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} X(k\omega_0) e^{jk\omega_0 t} \omega_0 = \frac{1}{2\pi} \int_{-\infty}^{\infty} X(\omega) e^{j\omega t} d\omega$$

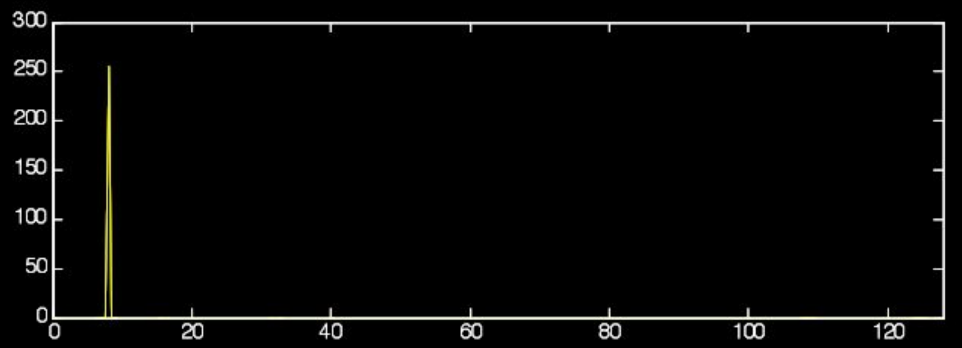
Analysis:

$$X(\omega) \triangleq \lim_{T\to\infty} X(k\omega_0) = \lim_{T\to\infty} \int_T x_T(t) e^{-jk\omega_0 t} dt = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

Famous Fourier Transforms

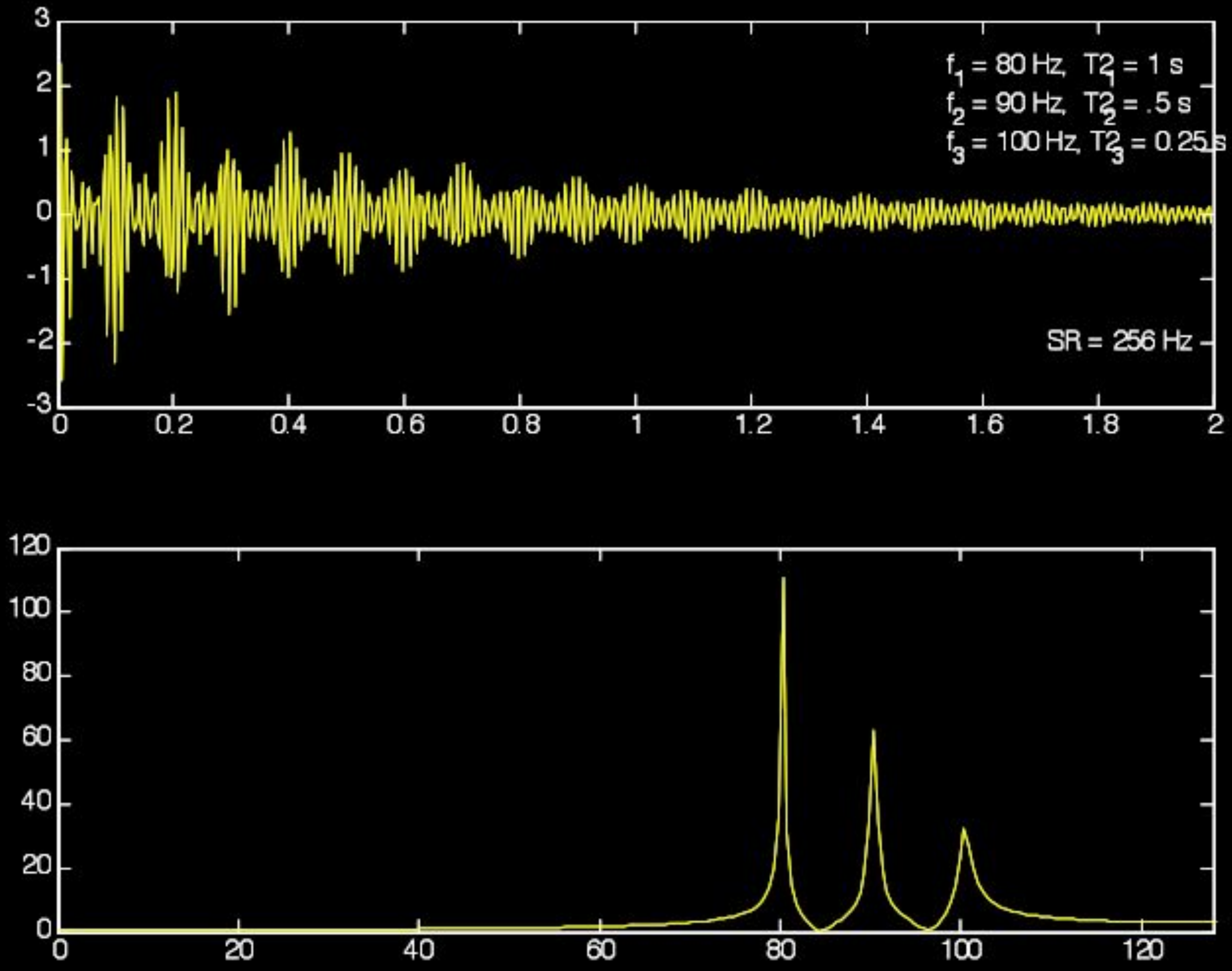


Sine wave



Delta function

Measuring multiple frequencies



Fast Fourier Transformation

Demo

Fast Fourier Transformation

Matlab Code:

```
fs = 1000;% hz... Sampling frequency
t = 0:1/fs:10;

f1 = 100; %hz... The freq of signal 1
f2 = 250; %hz... The freq of signal 2
x1 = sin(2*pi*f1*t) + sin(2*pi*f2*t)+ randn(size(t));

figure;
subplot(2,1,1)
plot(t,x1);
xlabel('time');
ylabel('signal');

FFTLen = 1024; % Length of FFT
y1 = abs(fft(x1,FFTLen));
y1 = y1 (1:FFTLen/2+1); % We will plot the first half of it as the second half is merely the reflection of the first half
y1(2:end) = y1(2:end)*2;

subplot(2,1,2);
plot([0:FFTLen/2]*fs/FFTLen,y1);
xlabel('Frequency');
ylabel('Energy');
```

Fast Fourier Transformation

Matlab Code:

```
FFTLen = 1024; % Length of FFT
y1 = abs(fft(x1,FFTLen));
y1 = y1 (1:FFTLen/2+1); % We will plot the first half of it as the second half is merely the reflection of the first half
y1(2:end) = y1(2:end)*2;

subplot(2,1,2);
plot([0:FFTLen/2]*fs/FFTLen,y1);
xlabel('Frequency');
ylabel('Energy');
```

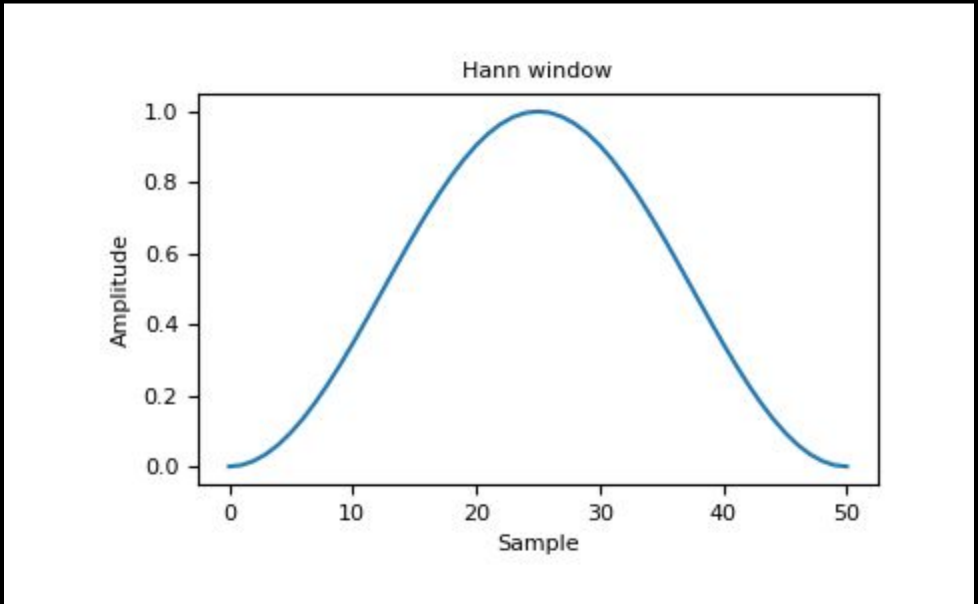
Considerations for window size selection

	Small Window	**Large Window**
Temporal Resolution	• High Temporal Resolution • Can track sudden changes in time domain signal.	• Low Temporal Resolution • Fail to track sudden changes • Has smearing/smoothing effect
Frequency Resolution	• Low Frequency Resolution • Fail to resolve two/more tightly spaced frequencies	• High Frequency Resolution • Can resolve two/more tightly spaced frequencies

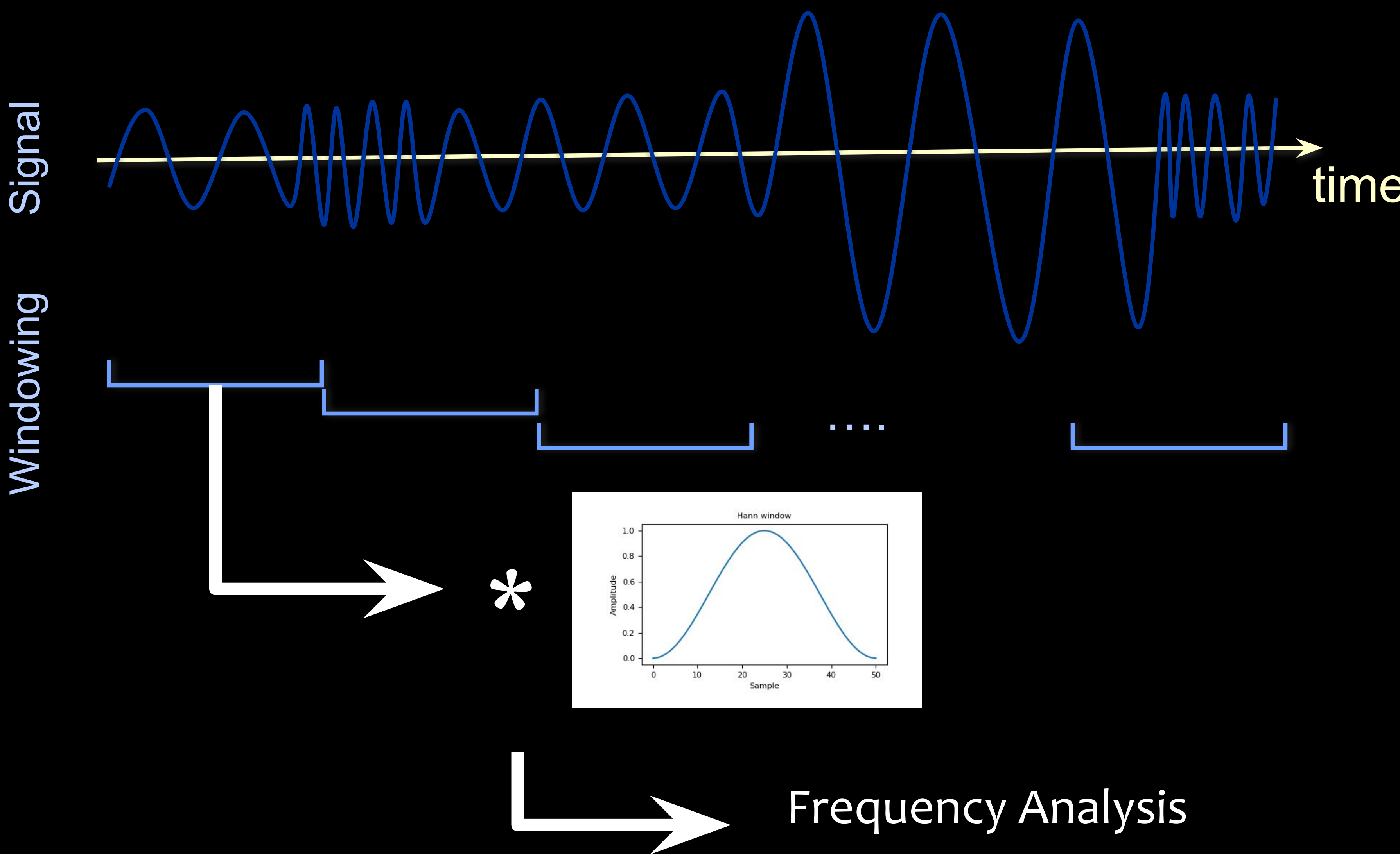
Considerations for window function

Hanning Function

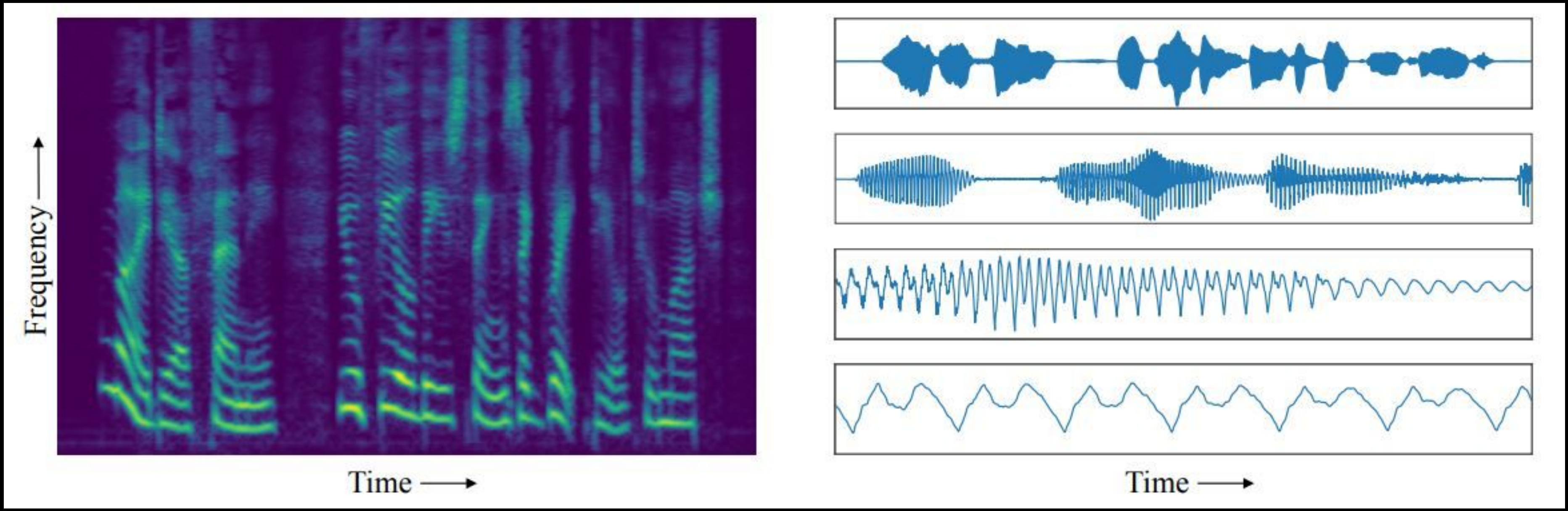
$$w(n) = 0.5 - 0.5cos\left(\frac{2\pi n}{M-1}\right) \qquad 0 \leq n \leq M-1$$



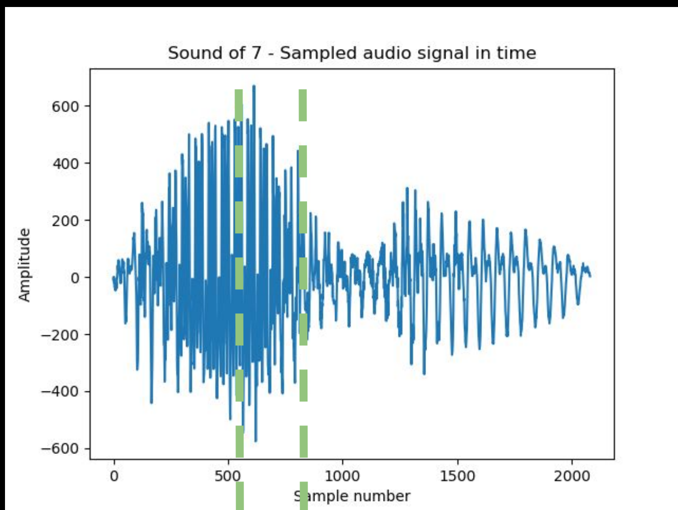
Feature Extraction



Spectrogram



Spectrogram

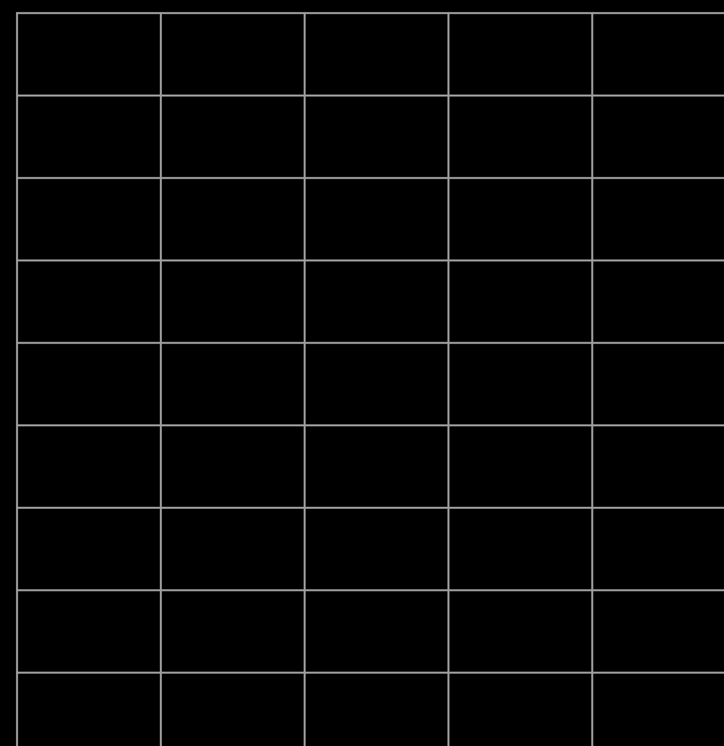


Windowed Time Domain Signal

FFT

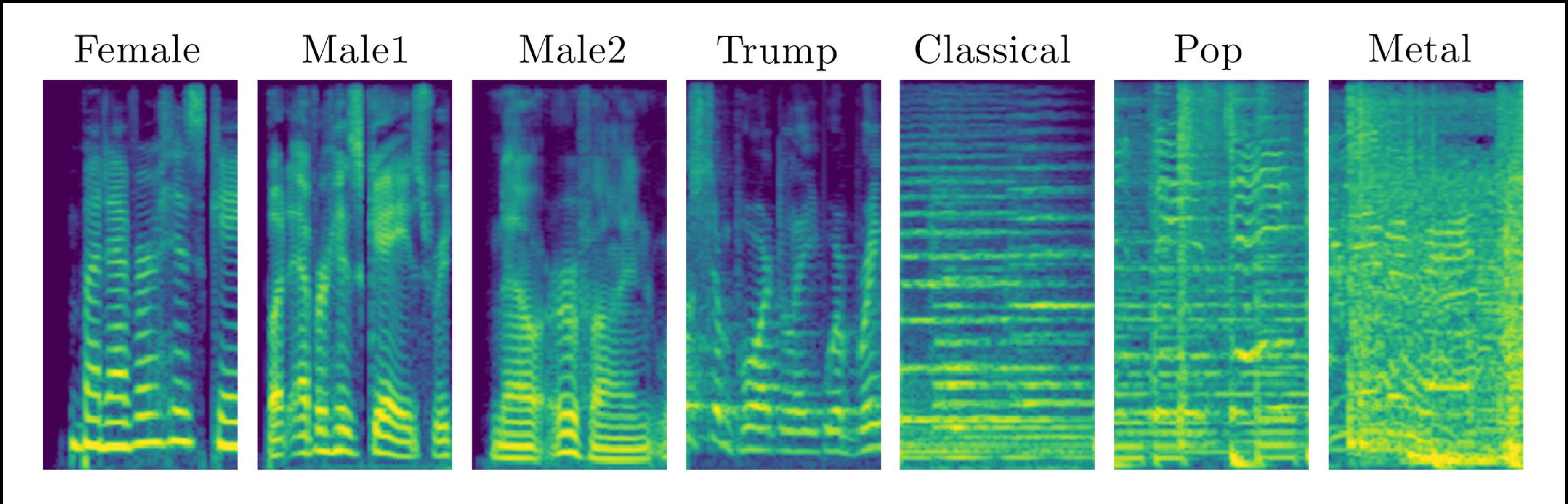
Frequency Domain Signal

Frames: 1st 2nd ….. nth …..

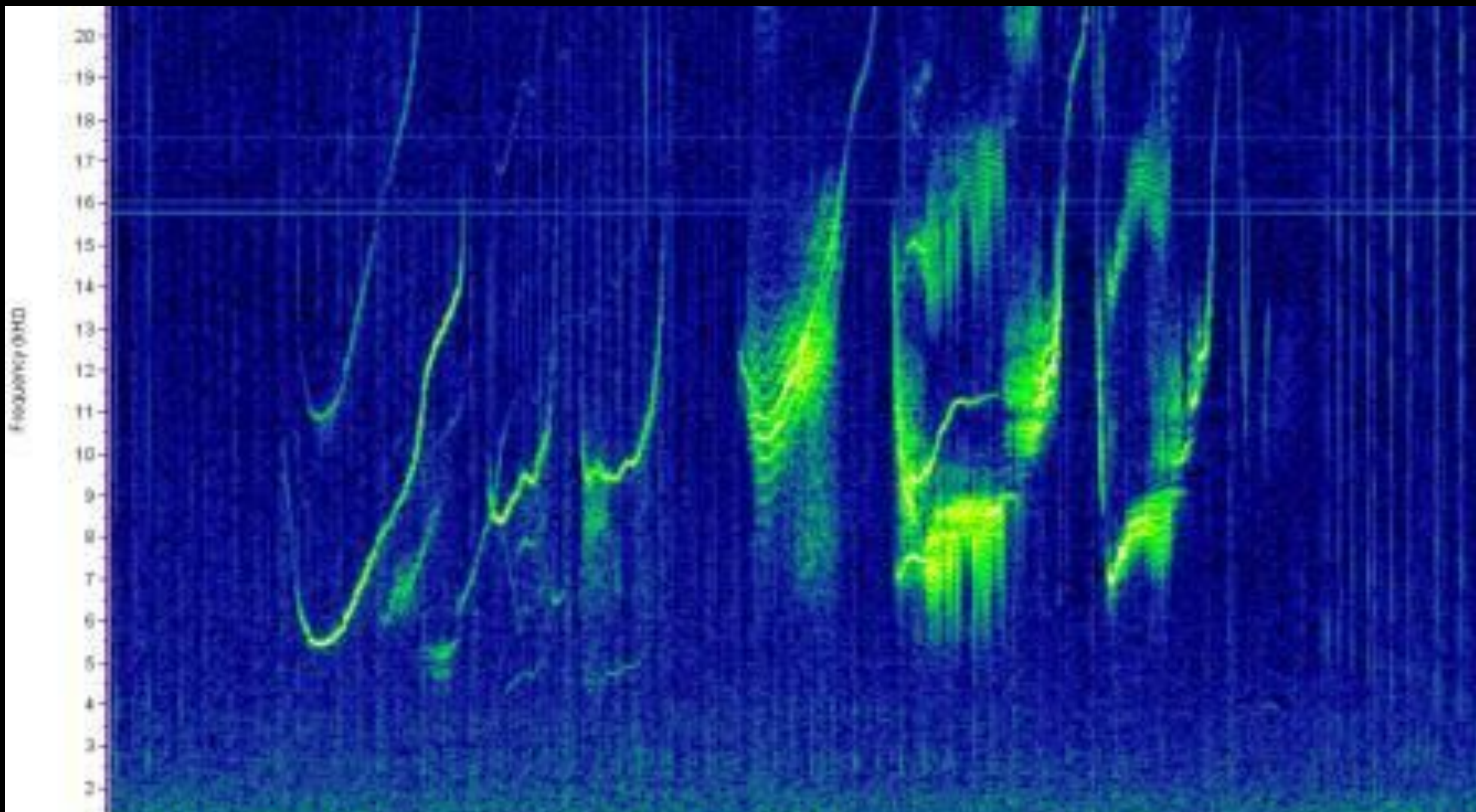


Frequencies

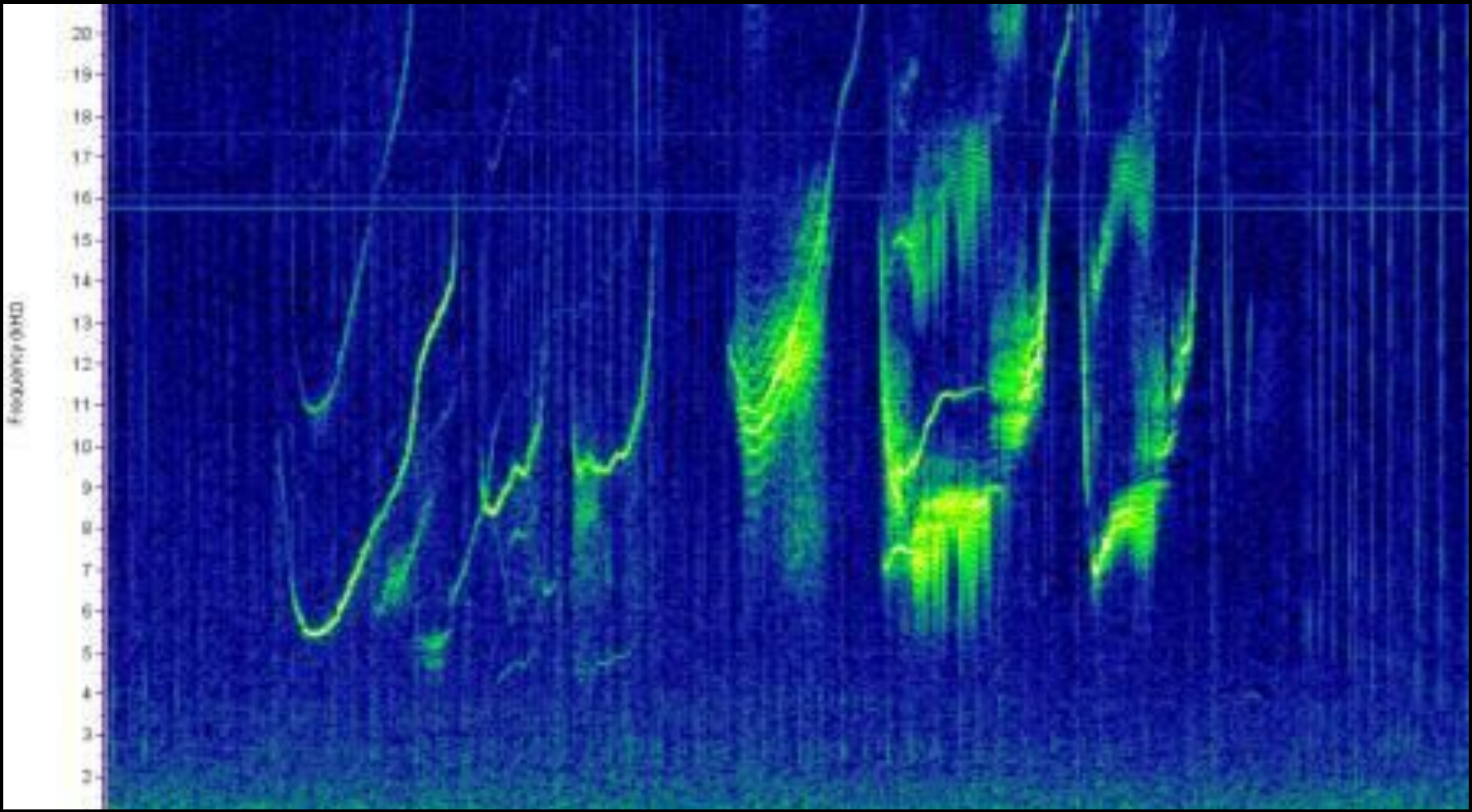
Spectrogram



?


20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2

Dolphin



Spectral Features

Spectral Centroid

The spectral centroid is a measure that indicates where the "center of mass" of the spectrum is located.

$$\text{Centroid} = \frac{\sum_{n=0}^{N-1} f(n)x(n)}{\sum_{n=0}^{N-1} x(n)}$$

Here, f(n) refers to the nth frequency and x(n) refers to the power or weight associated with nth frequency.

Spectral Features

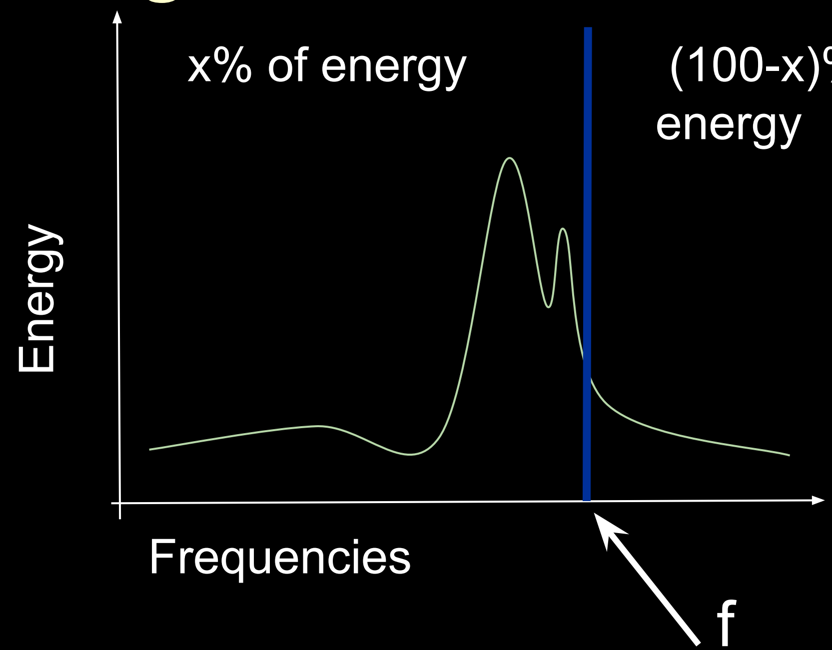
Spectral Slope

The spectral "slope" can be quantified by applying linear regression to the Fourier magnitude spectrum of the signal, which produces a single number indicating the slope of the line-of-best-fit through the spectral data.

Spectral Features

Spectral Roll off x% (e.g., 95% or 75%)

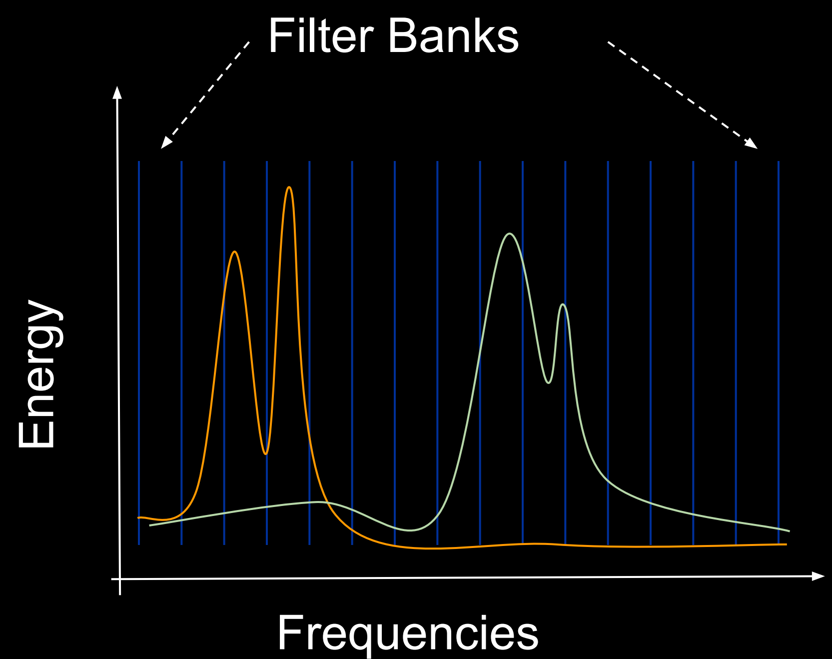
This refers to the frequency (f) below which x% of the signal energy lie.



Spectral Features

Filter Bank

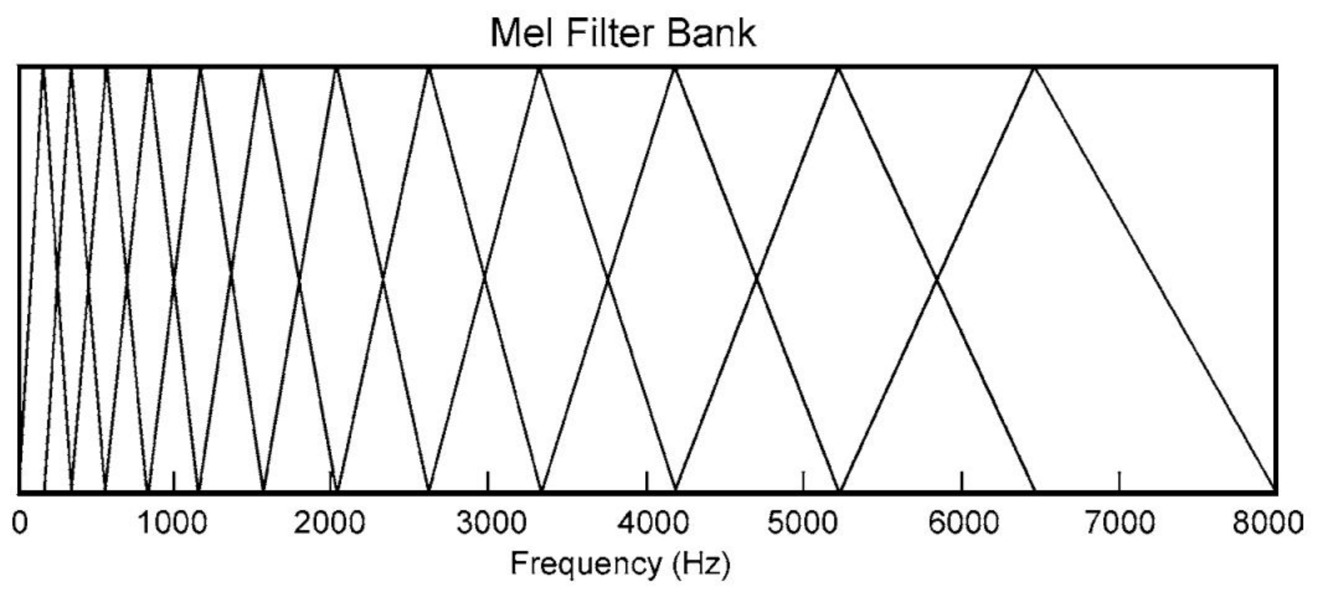
This refers to splitting the spectra into several frequency bands and estimating the total energy in each of these bands.



Spectral Features

Filter Bank

Different filters within a filter bank can have different bandwidth.



Spectral Features

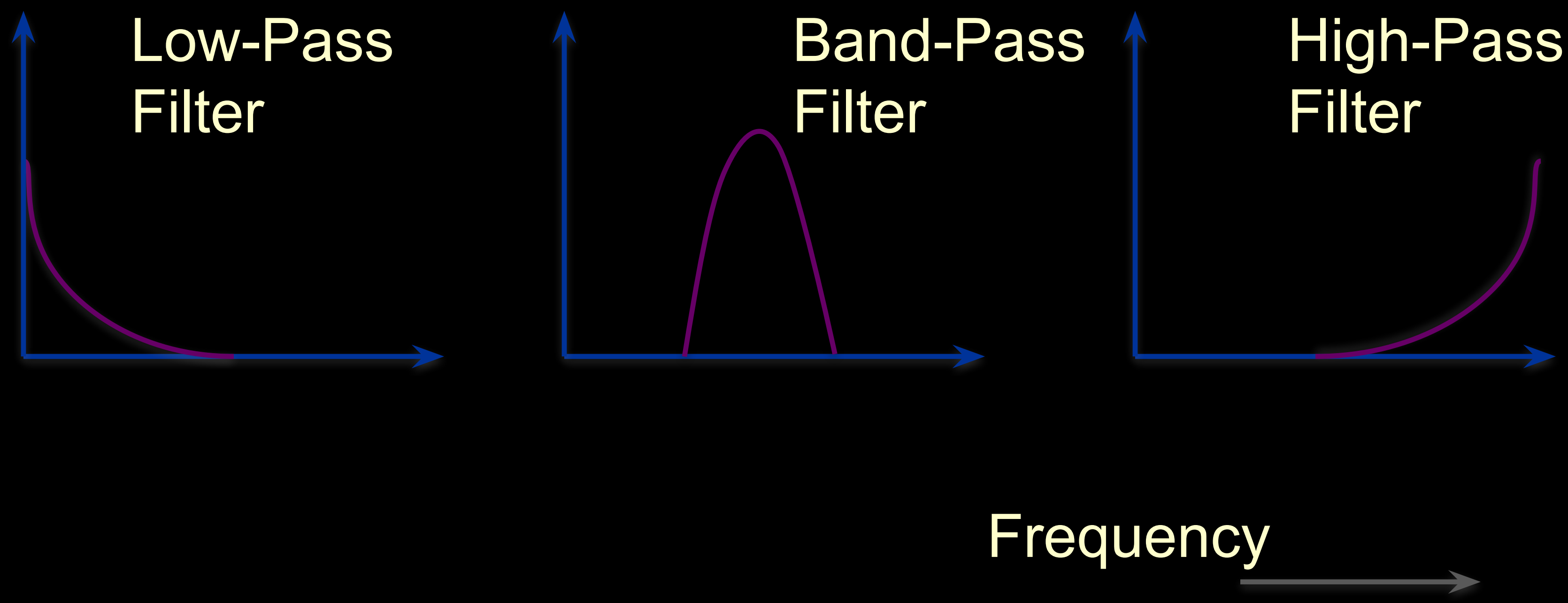
A detailed list of different spectral features can be found here.

http://docs.twoears.eu/en/latest/afe/available-processors/spectral-features/

How to engineer good features?

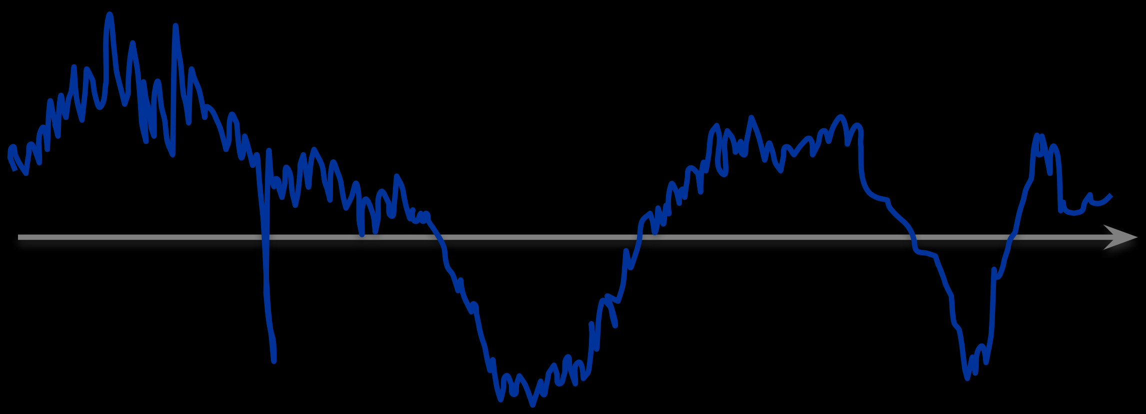
- Get to know your data well (Visualize)
  - Human intuitions based on thorough observations
- Exploit current understanding
  - Talk with domain experts
  - Do a good literature review
- Begin with a large number of functions to extract a huge feature set and then use an automatic feature selection method to help you find a good feature subset.

Fundamentals of Filters

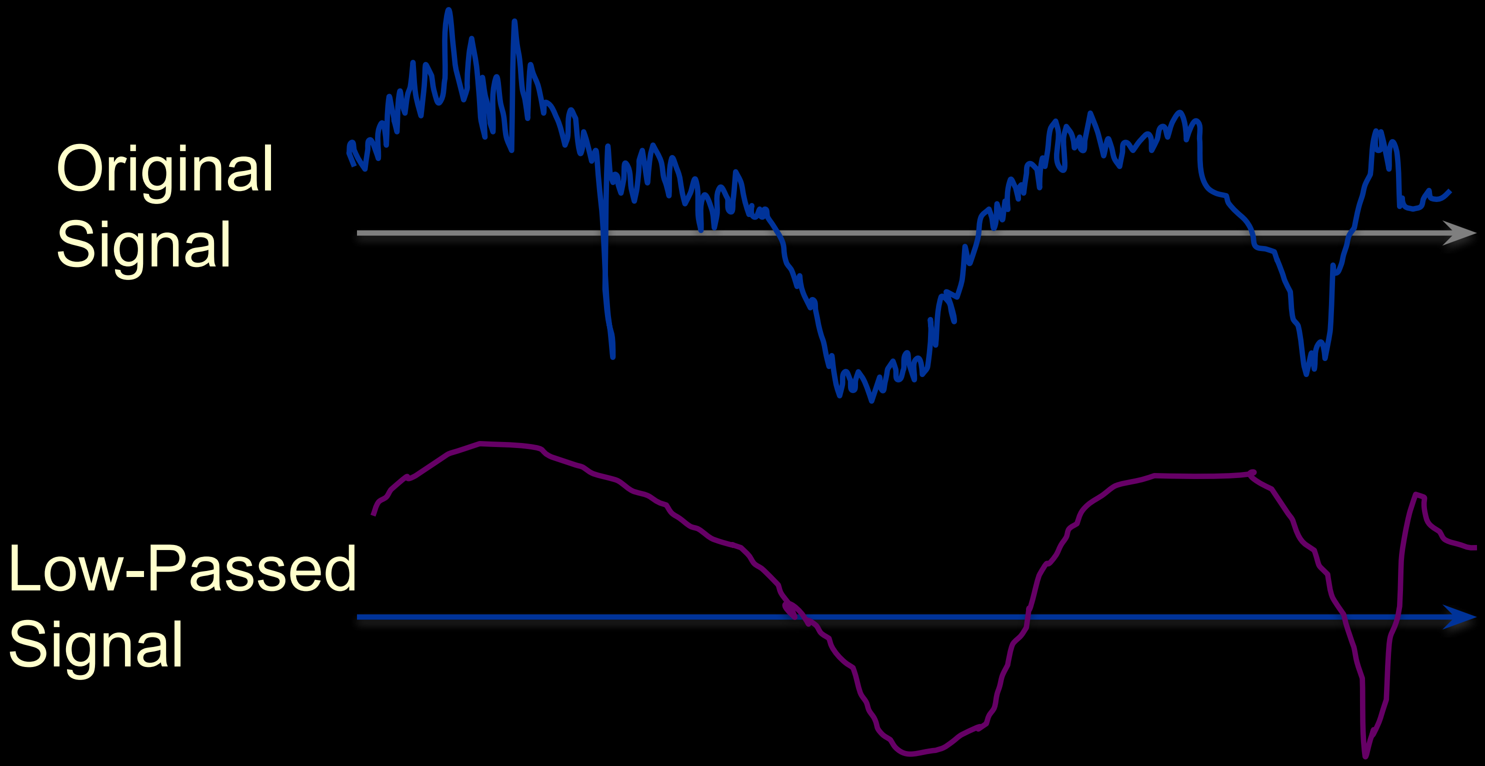


Fundamentals of Filters

Original Signal



Fundamentals of Filters



Fundamentals of Filters