COMP-4147

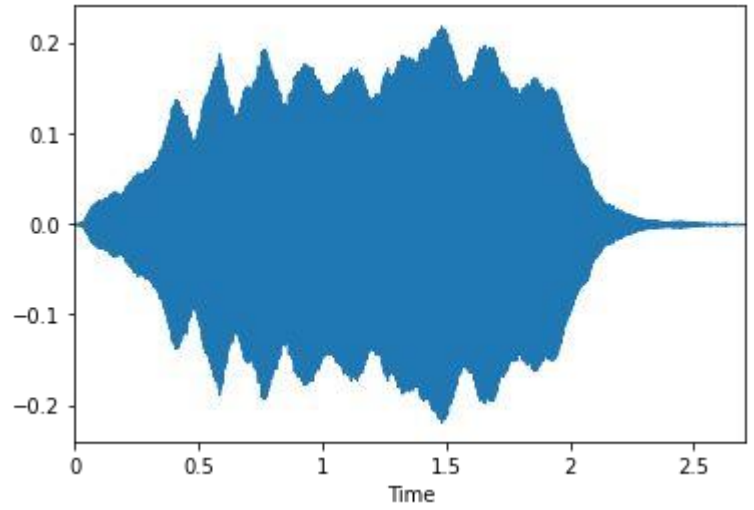
Basic concepts and techniques (Part 1)

Outline

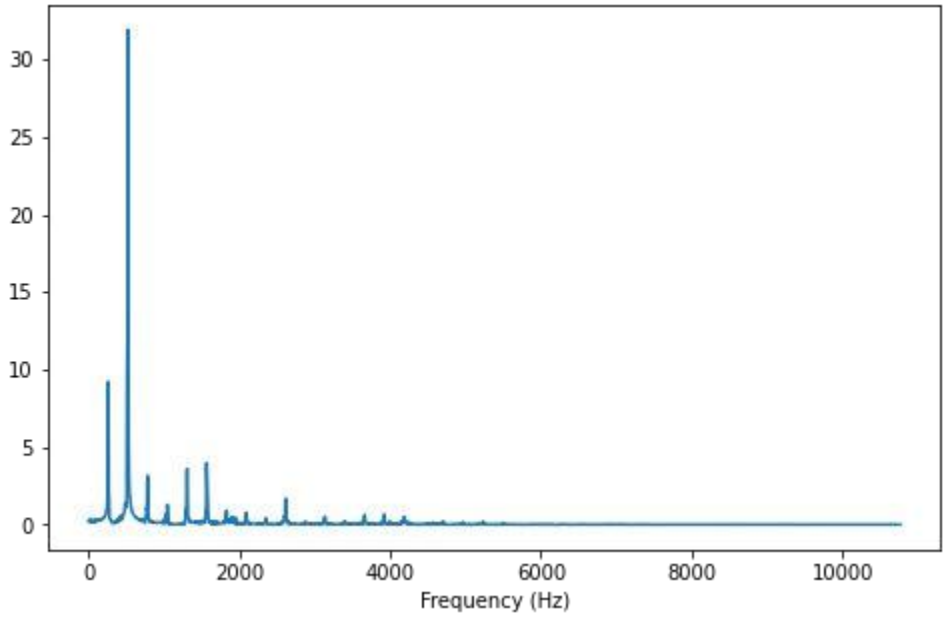
- Short-Time Fourier Transform
- Spectrogram
- MFCC

Short-Time Fourier Transform (STFT)

Previously



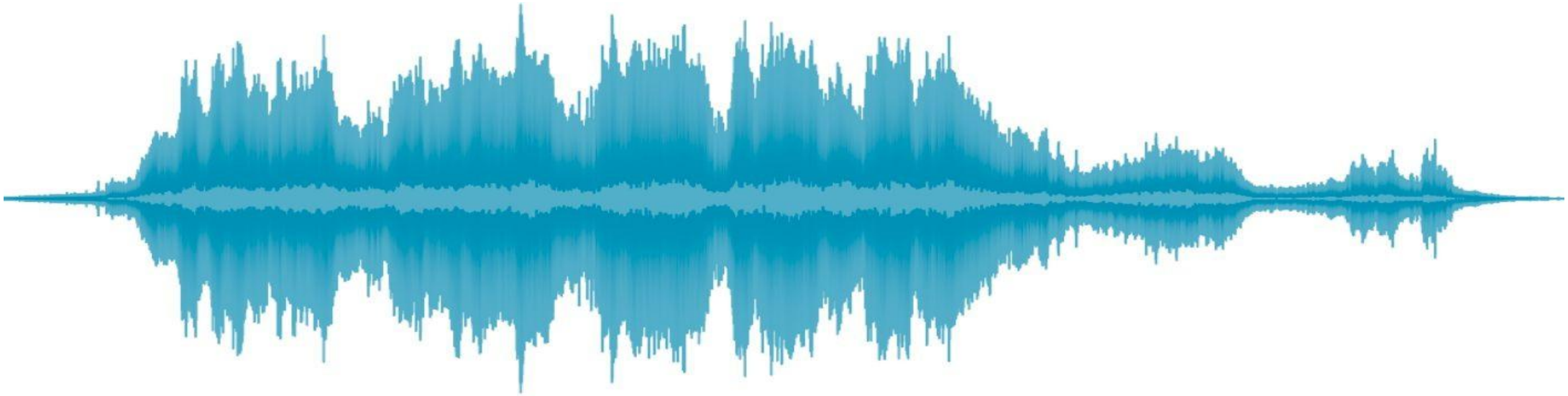
Fourier transform



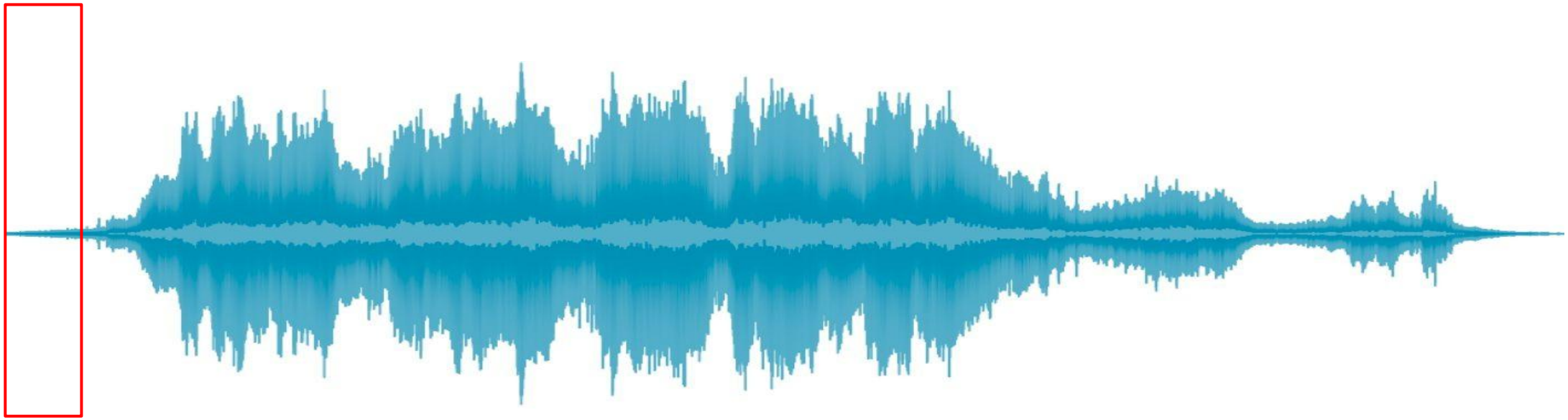
Fourier transform problem

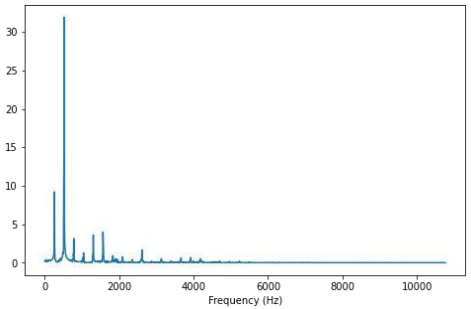
- Loss of time information
    - In an audio clip of a song, the FT might show peaks at certain notes, but it won't distinguish if a high-pitched note happens at the beginning or end.
- Sensitivity to noise and transients
    - Transient events (sudden bursts, like a drum hit) get smeared across the entire spectrum, masking other components.
- Inaccurate analysis
    - In an audio clip of a song, the FT might show peaks at certain notes, but it won't distinguish if a high-pitched note happens at the beginning or end.
- Spectral leakage
    - The continuous FT assumes the signal is periodic and extends infinitely in time, which isn't true for finite, real signals. This leads to artifacts like spectral leakage (energy spreading into adjacent frequency bins) unless windowing is applied.

STFT intuition



STFT intuition

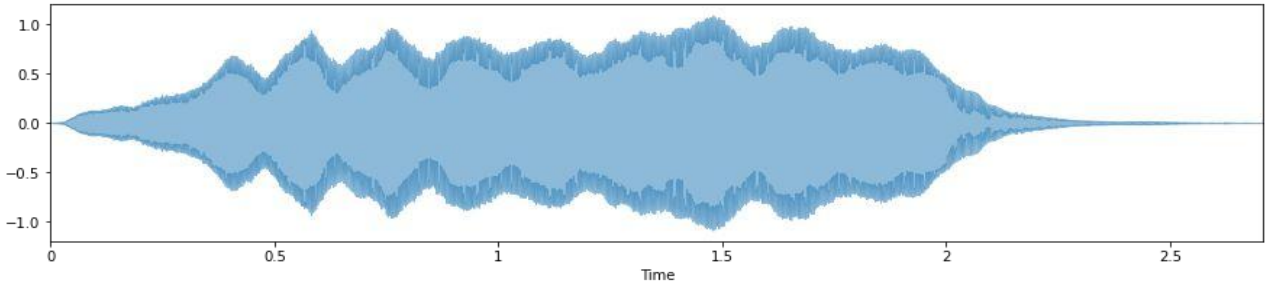




STFT intuition

STFT intuition

Windowing



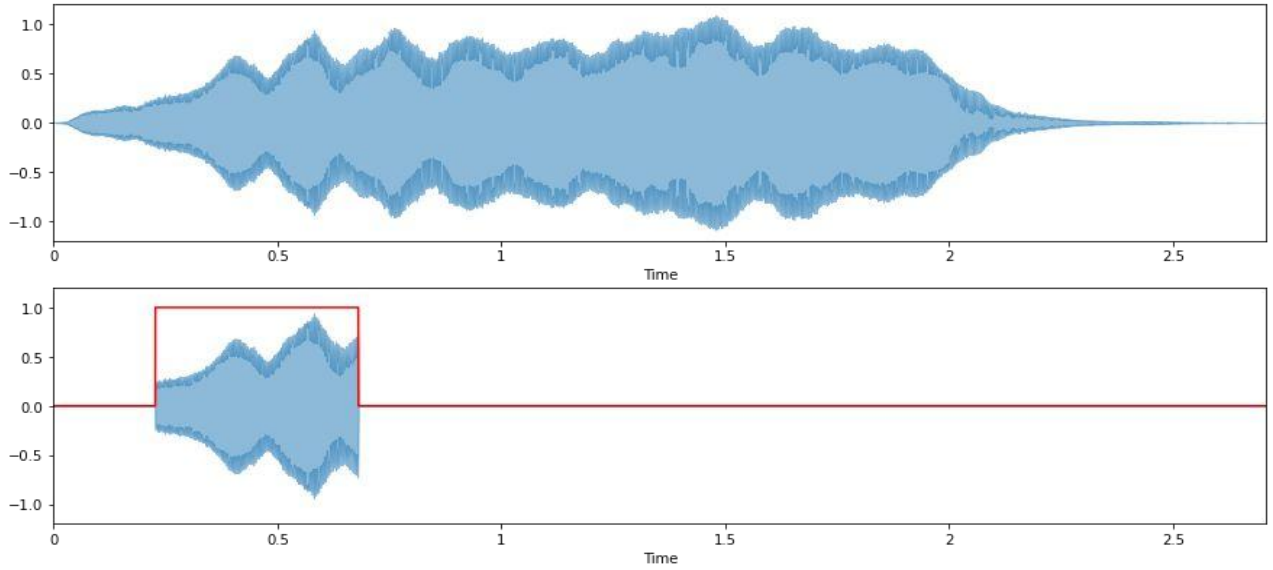
- Apply the windowing function to the signal.

$$x_w(k) = x(k) \cdot w(k)$$

Windowing

- Apply the windowing function to the signal.

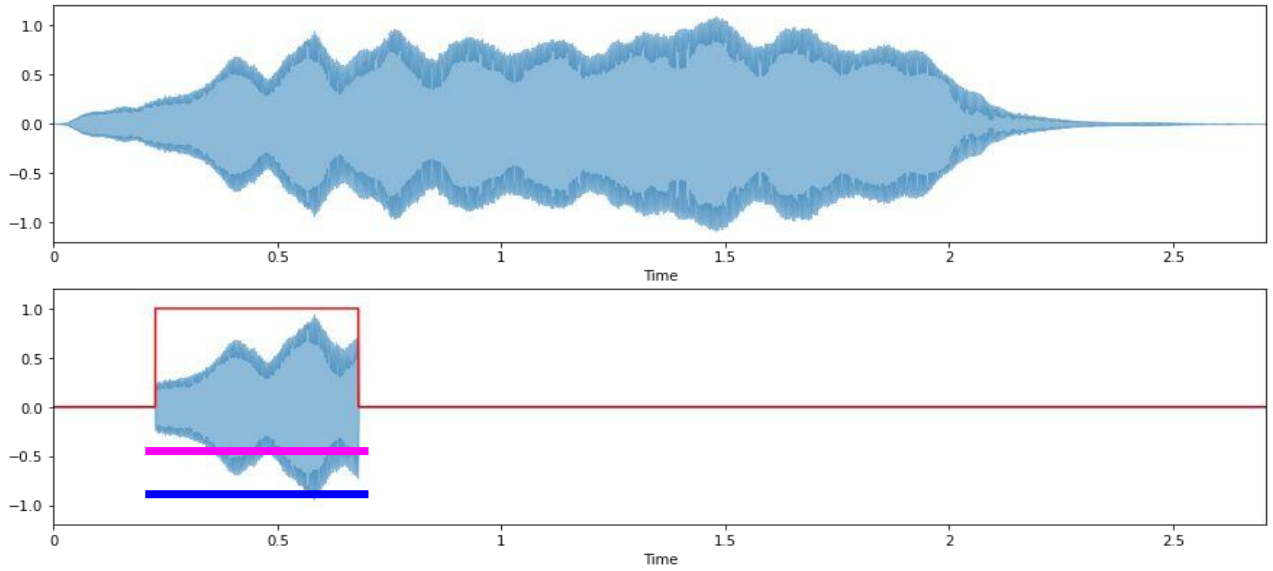
$$x_w(k) = x(k) \cdot w(k)$$



Windowing

- Apply the windowing function to the signal.

$$x_w(k) = x(k) \cdot w(k)$$



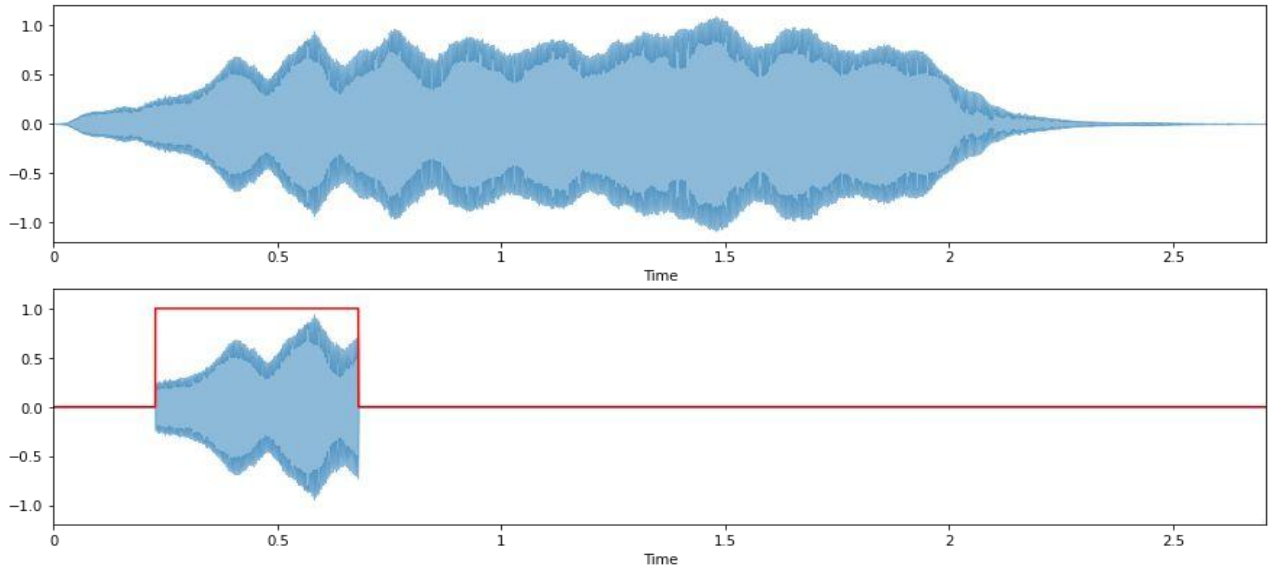
window size = frame size

STFT

- Apply the windowing function to the signal.

$$x_w(k) = x(k) \cdot w(k)$$

- Apply the Fourier transform to the signal segments.

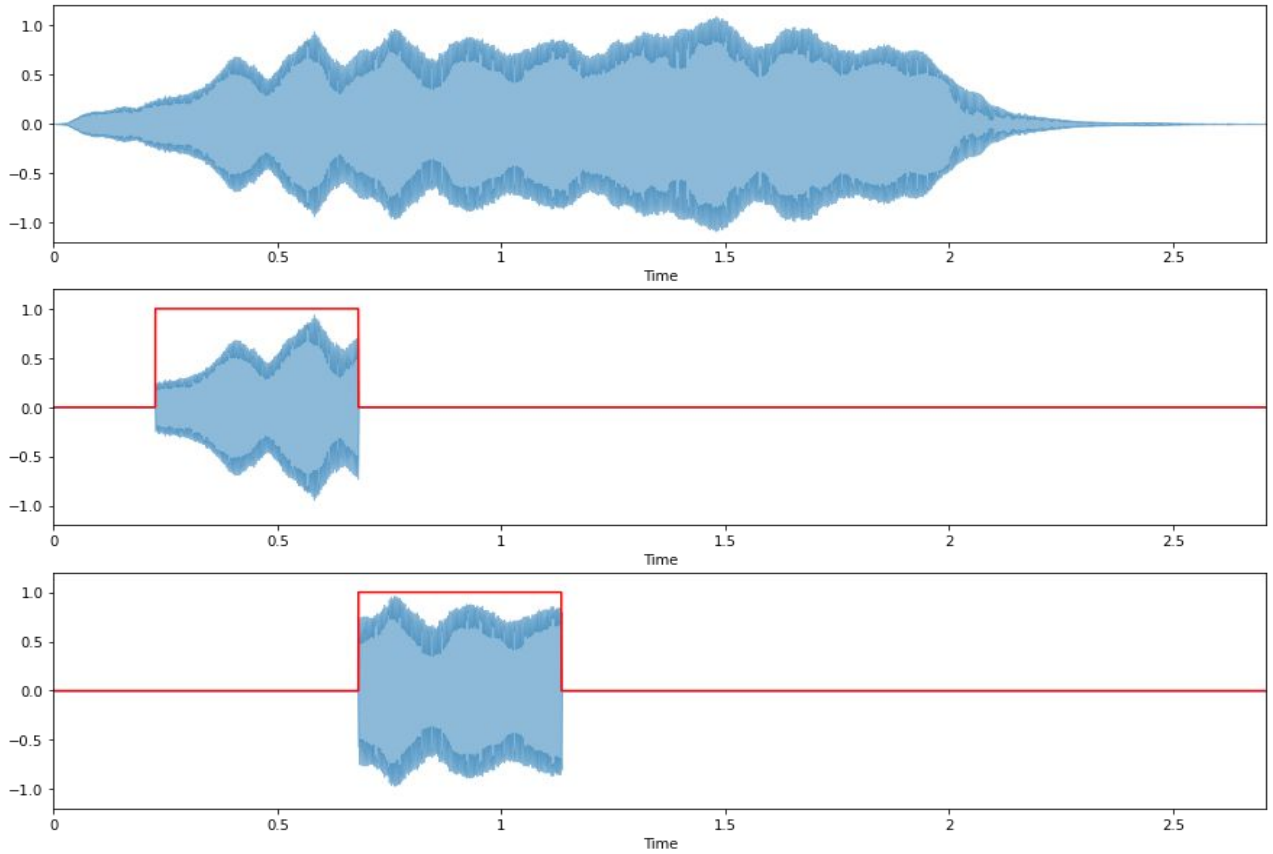


STFT

- Apply the windowing function to the signal.

$$x_w(k) = x(k) \cdot w(k)$$

- Apply the Fourier transform to the signal segments.

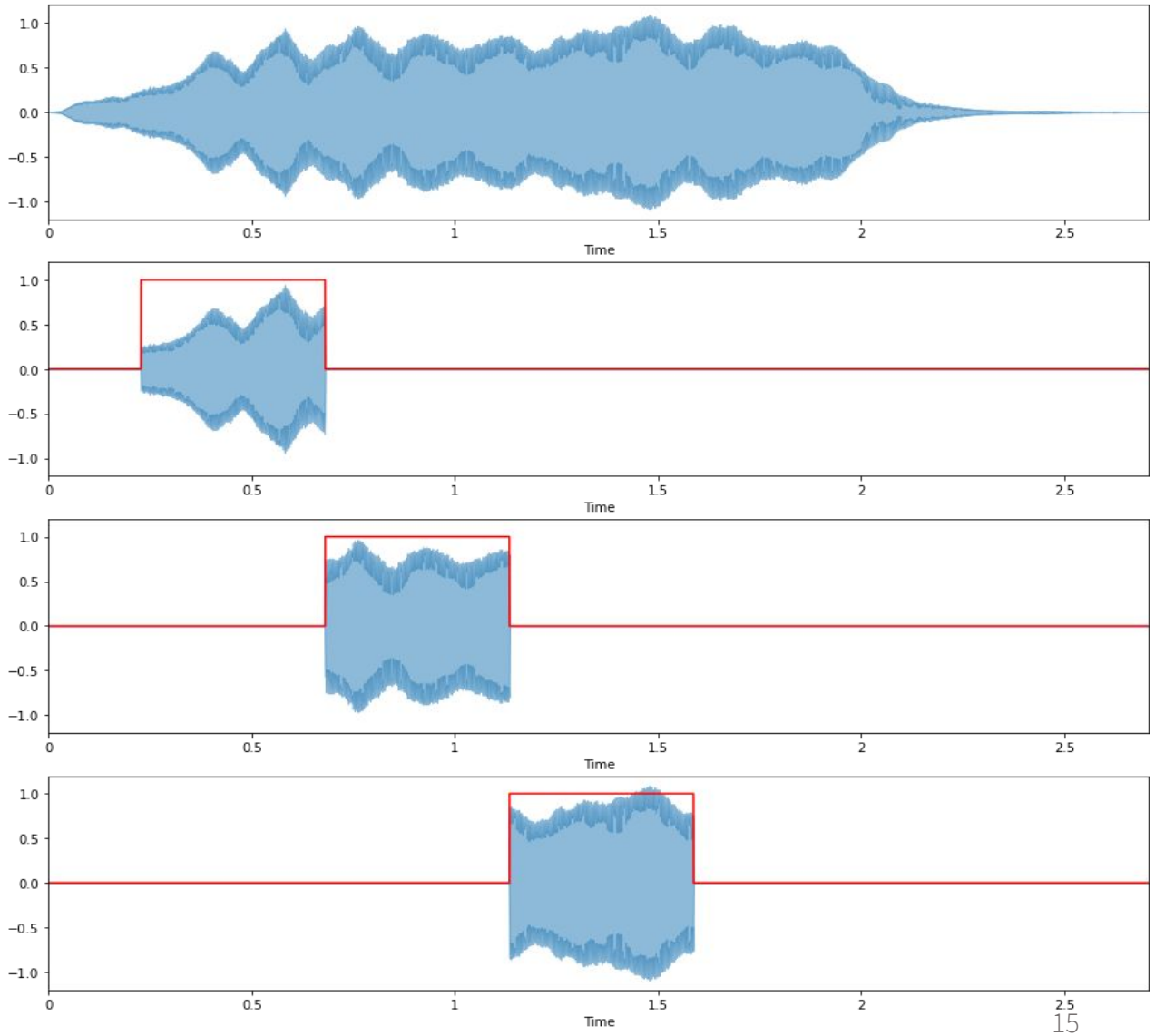


STFT

- Apply the windowing function to the signal.

$$x_w(k) = x(k) \cdot w(k)$$

- Apply the Fourier transform to the signal segments.

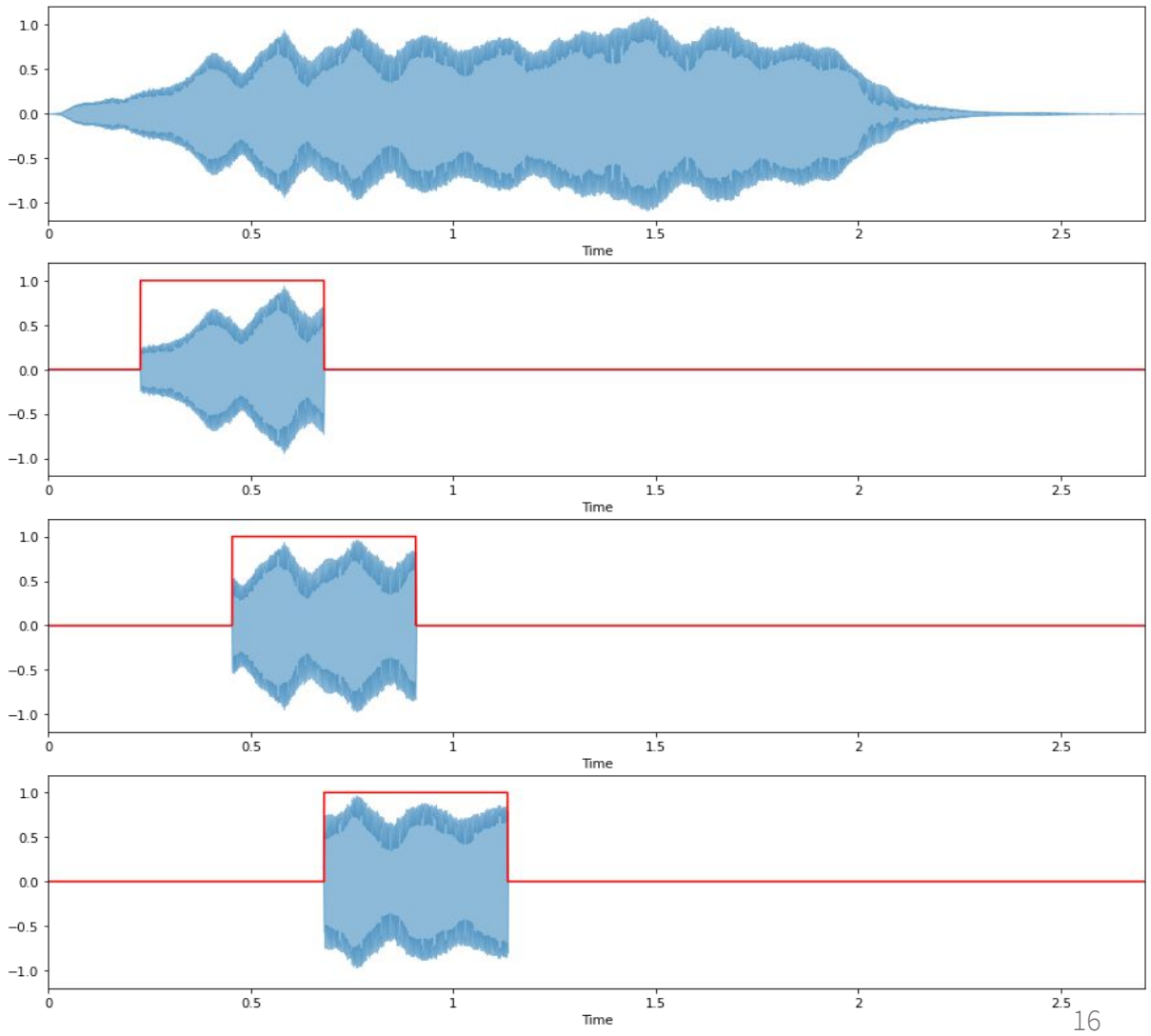


Overlapping windows

- Apply the windowing function to the signal.

$$x_w(k) = x(k) \cdot w(k)$$

- Apply the Fourier transform to the signal segments.

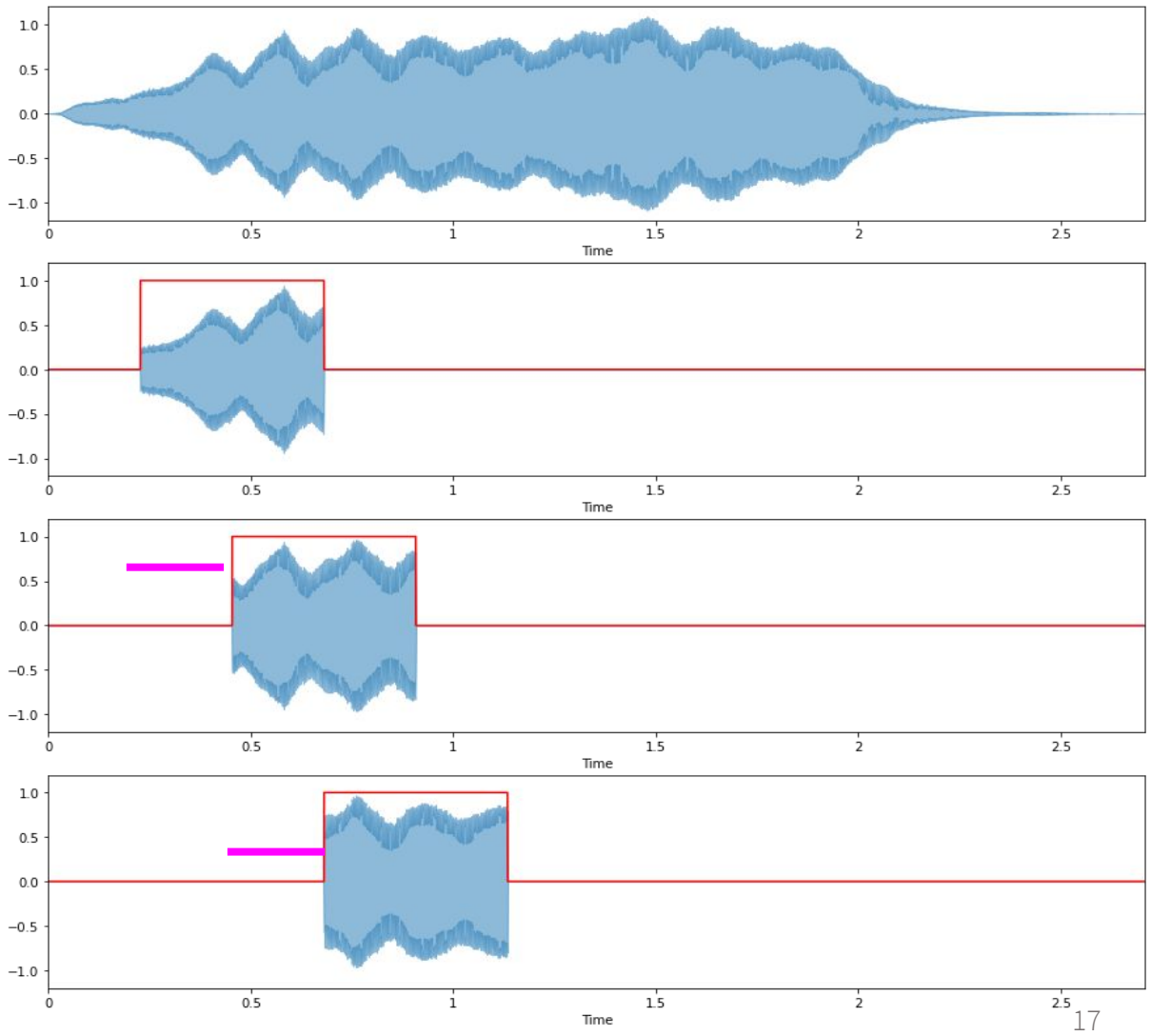


Overlapping windows

- Apply the windowing function to the signal.

$$x_w(k) = x(k) \cdot w(k)$$

- Apply the Fourier transform to the signal segments.



From DFT to STFT

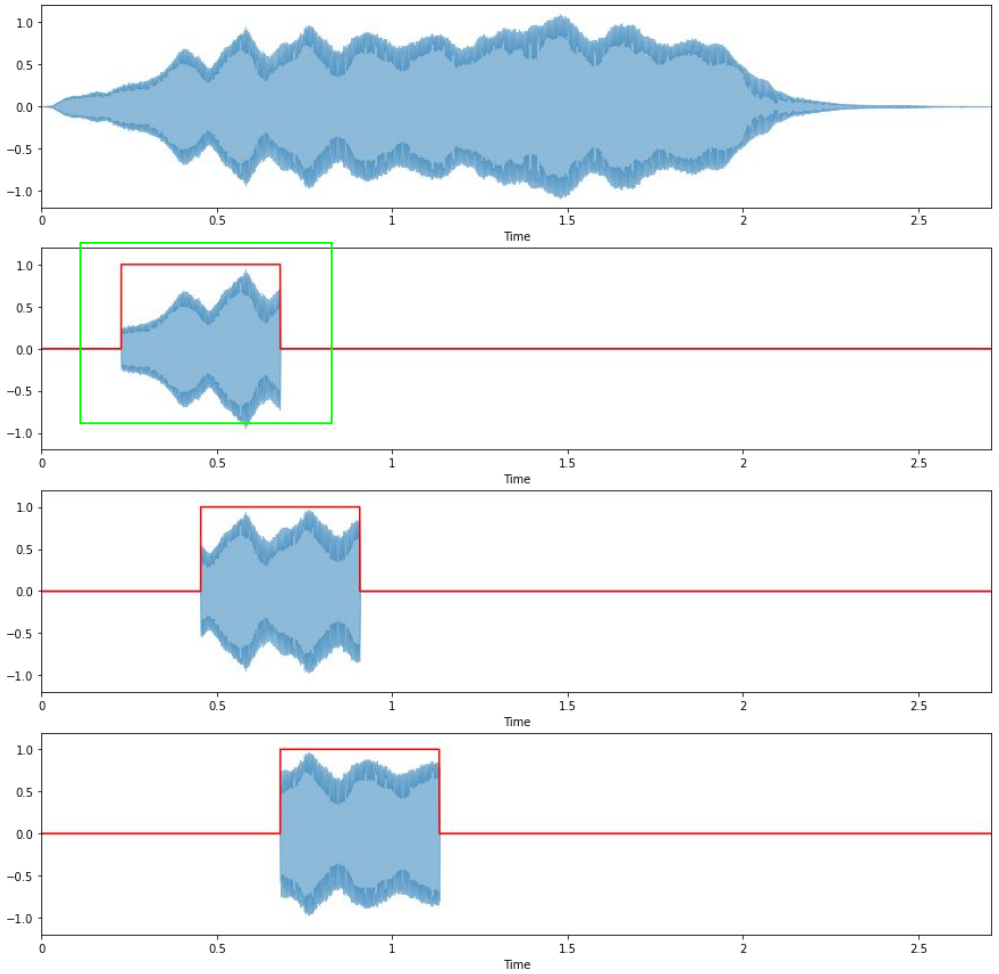
- DFT

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

- STFT

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

m = 1



From DFT to STFT

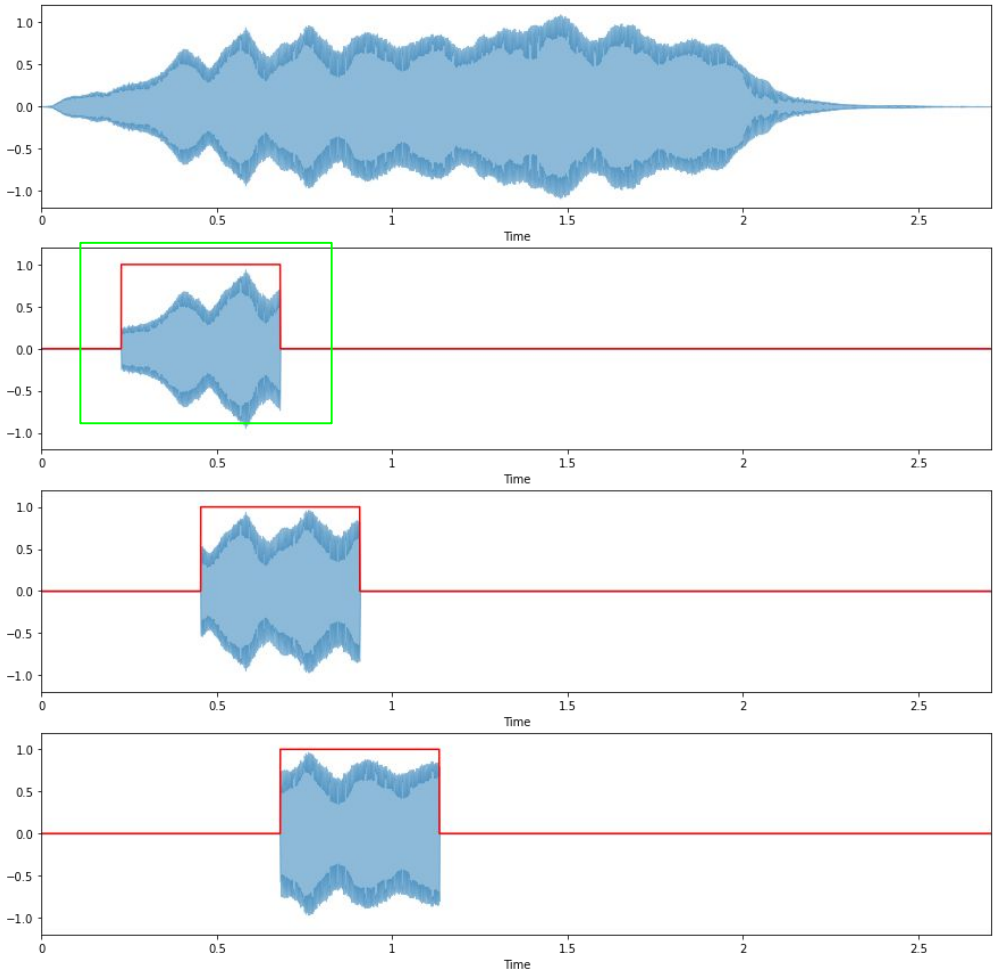
- DFT

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

- STFT

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

m = 1



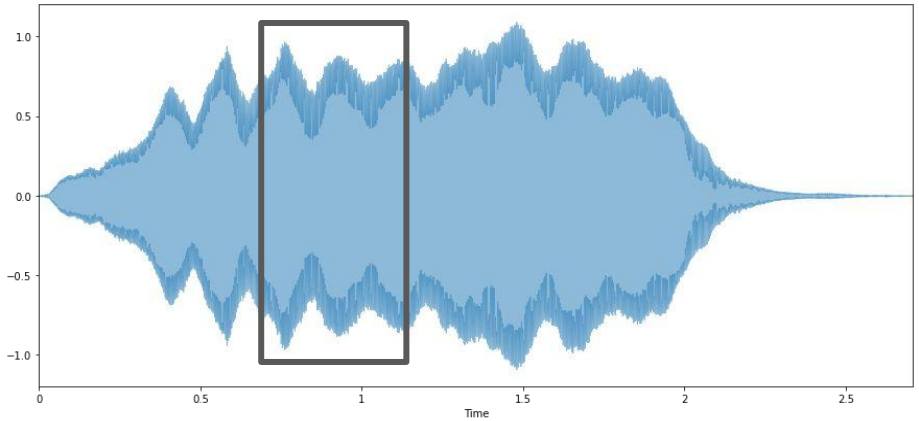
From DFT to STFT

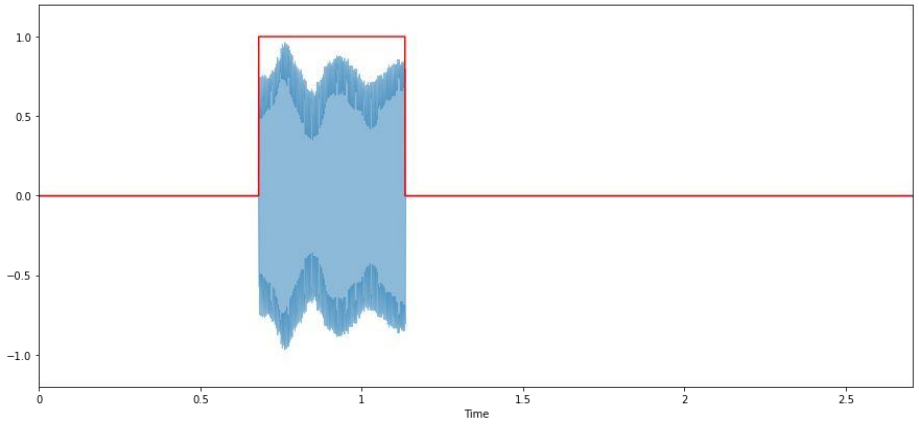
- DFT

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

- STFT

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n \frac{k}{N}}$$





From DFT to STFT

- DFT

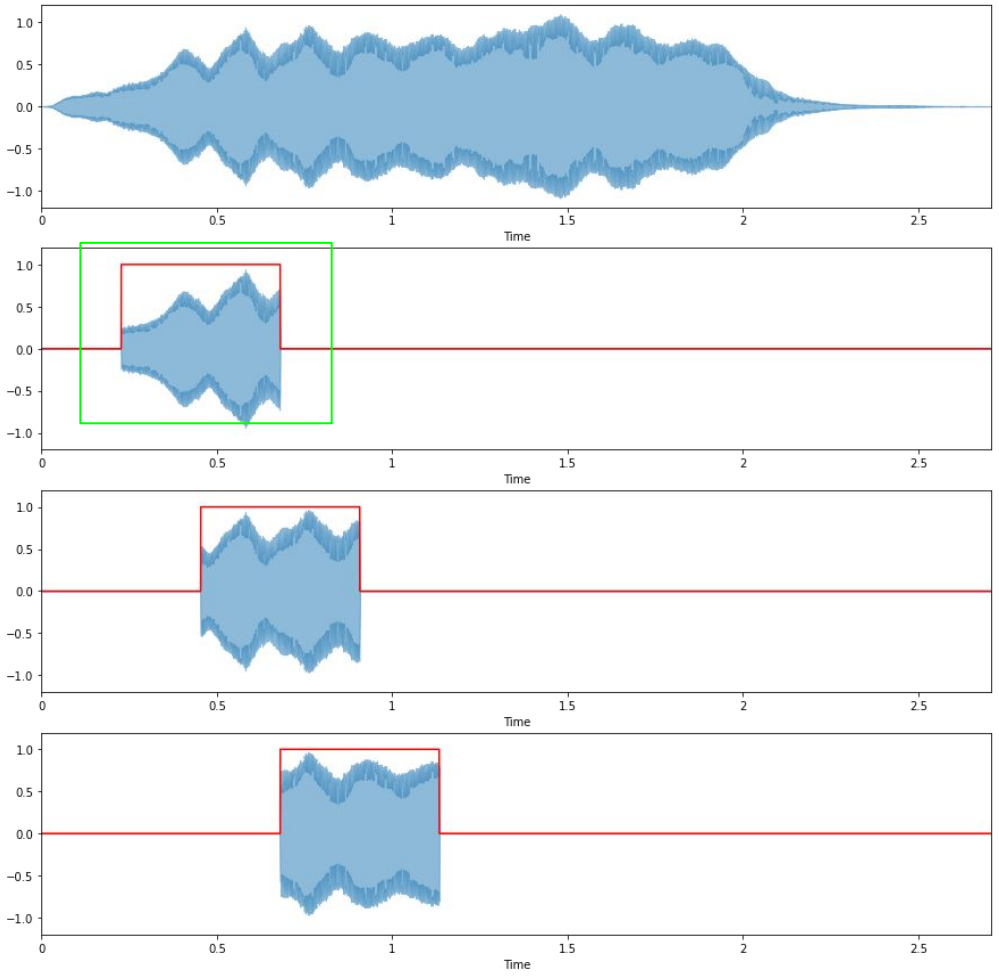
$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

- STFT

$$S(m,k) = \sum_{n=0}^{N-1} x(n + mH) \cdot w(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

Starting sample of the current frame

m = 1



From DFT to STFT

- DFT

$$\hat{x}(k) = \sum_{n=0}^{N-1} x(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

- STFT

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

- DFT
  - Spectral vector (# frequency bins)
  - N complex Fourier coefficients
- STFT
  - Spectral matrix (# frequency bins, # frames)
  - Complex Fourier coefficients

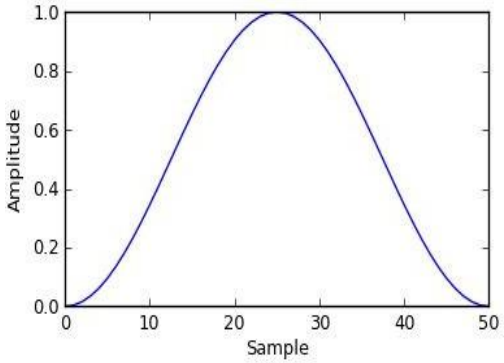
STFT Parameter

- STFT

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

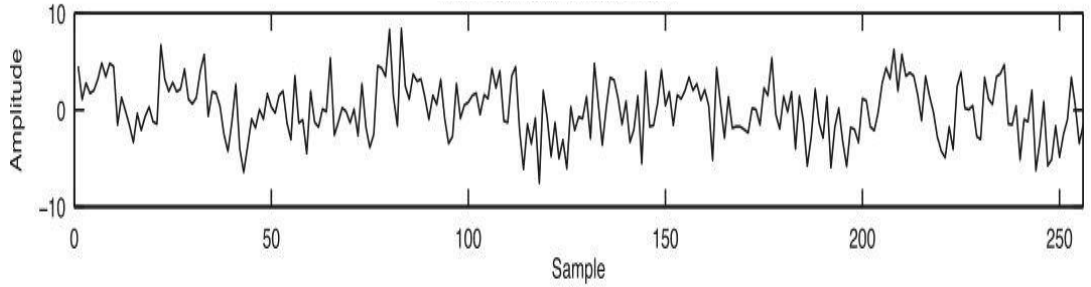
  - Windowing function: w(n)
  - Frame size: (n)
  - Hope size: (H)

- Windowing function
  - Hann window

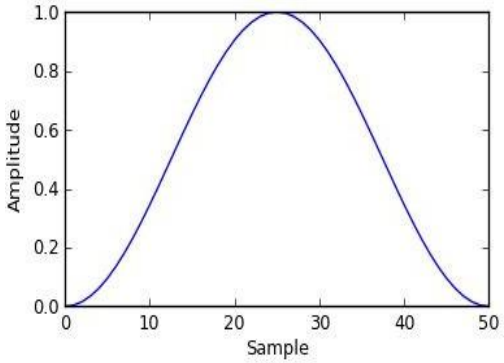


$$w(k) = 0.5 \cdot (1 - \cos(\frac{2\pi k}{K-1})), k = 1...K$$

Windowing function

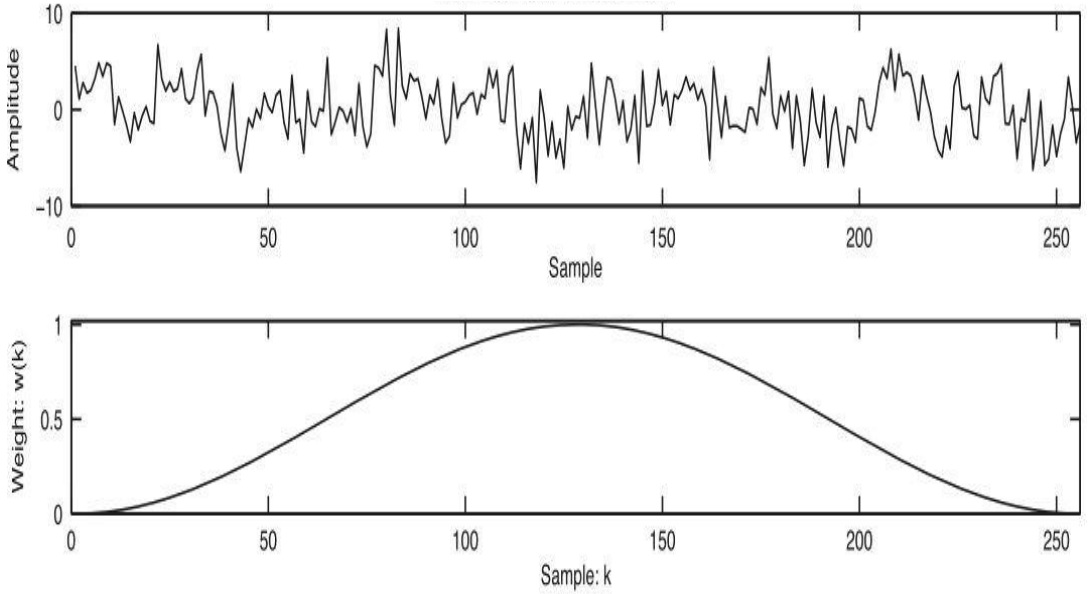


- Windowing function
  - Hann window

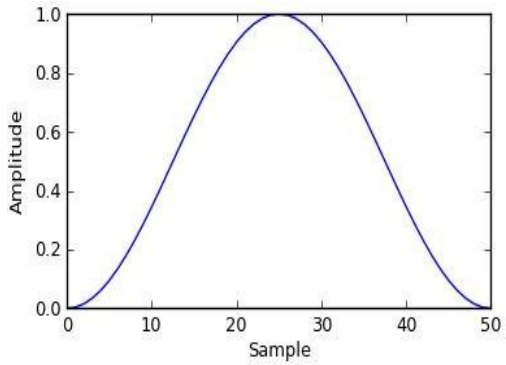


$$w(k) = 0.5 \cdot (1 - \cos(\frac{2\pi k}{K-1})), k = 1...K$$

Windowing function

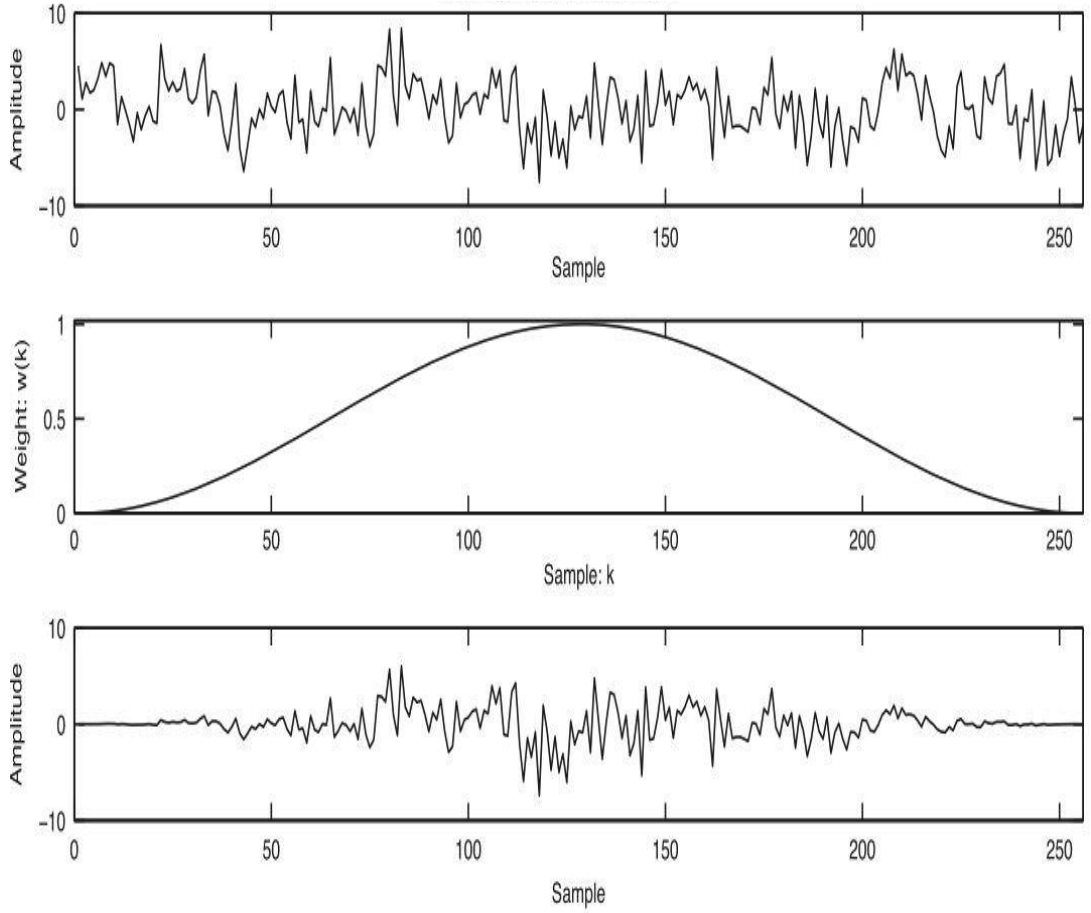


- Windowing function
  - Hann window

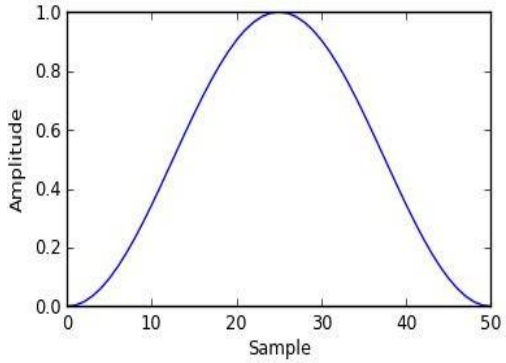


$$w(k) = 0.5 \cdot (1 - \cos(\frac{2\pi k}{K-1})), k = 1...K$$

Windowing function

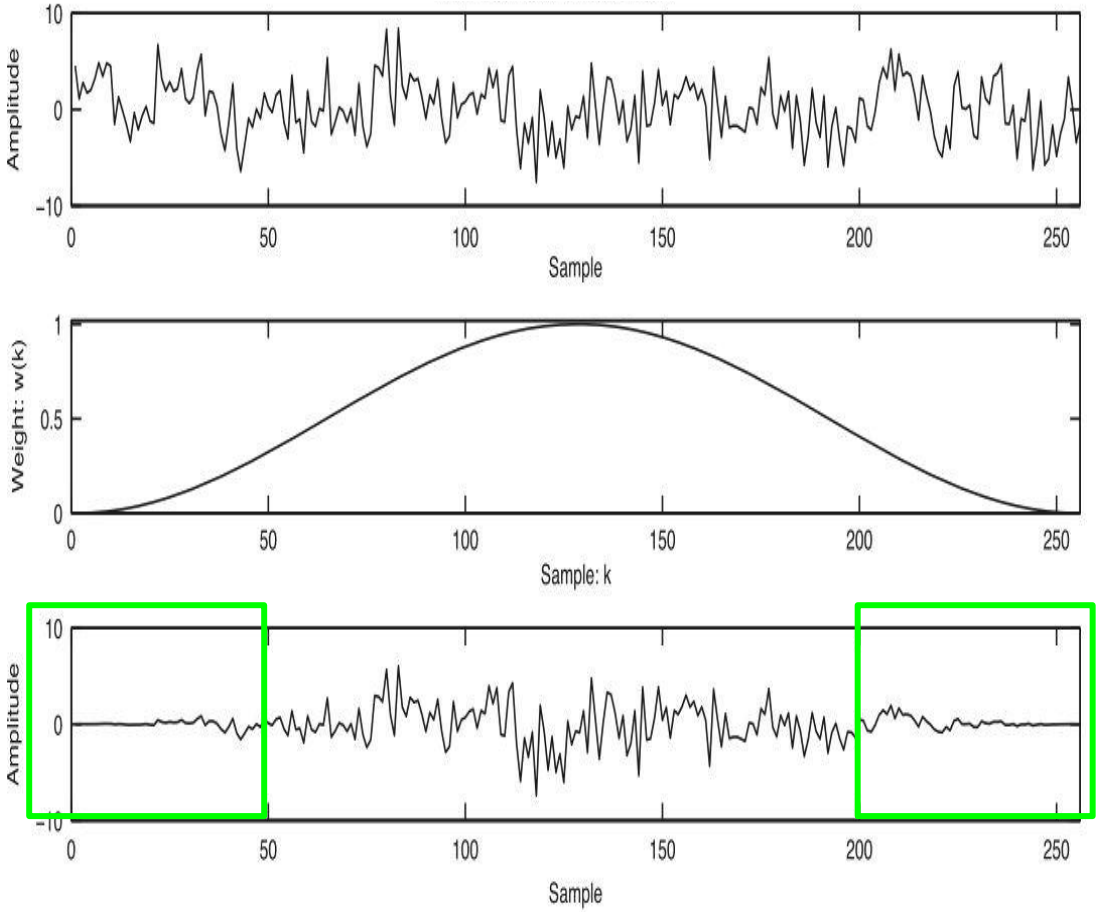


- Windowing function
  - Hann window

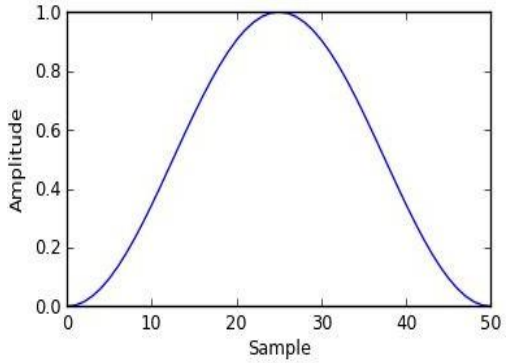


$$w(k) = 0.5 \cdot (1 - \cos(\frac{2\pi k}{K-1})), k = 1...K$$

Windowing function



- Windowing function
  - Hann window



$$w(k) = 0.5 \cdot (1 - \cos(\frac{2\pi k}{K-1})), k = 1...K$$

STFT Parameter

- STFT

$$S(m,k) = \sum_{n=0}^{N-1} x(n+mH) \cdot w(n) \cdot e^{-i2\pi n \frac{k}{N}}$$

  - Windowing function: w(n)
  - Frame size: (n)
  - Hope size: (H)

- Spectral matrix (# frequency bins, # frames)
- Complex Fourier coefficients

- Size of STFT
- # frequency bins = $\frac{frame\ size}{2} + 1$
- # frames = $\frac{samples - frame\ size}{hop\ size} + 1$

Example

- Signal: 10 k samples
- Frequency size: 1000
- Hop size: 500
- What is the STFT output size?

- Number of Frequency bins:
- Number of Frames:
- The size of STFT:

Example

- Signal: 10 k samples
- Frequency size: 1000
- Hop size: 500
- What is the STFT output size?

- Number of Frequency bins:
  - 1000/2+1 = 501
  - Map to (0, sampling rate /2)
- Number of Frames:
  - (10000 – 1000) / 500 + 1 = 19
- The size of STFT:
  - 501 x 19 complex matrix

STFT parameters: Frame size

- Frame size:
  - 512, 1024, 2048, 4096, 8192, etc.

- Time – frequency trade-off
  - When the frame size increases, the frequency resolution increases, but the time resolution decreases.
  - When the frame size decreases, the frequency resolution decreases, but the time resolution increases.
  - *Heisenberg Uncertainty Principle*: Impossible to simultaneously know a subatomic particle's exact position and exact momentum (or velocity)

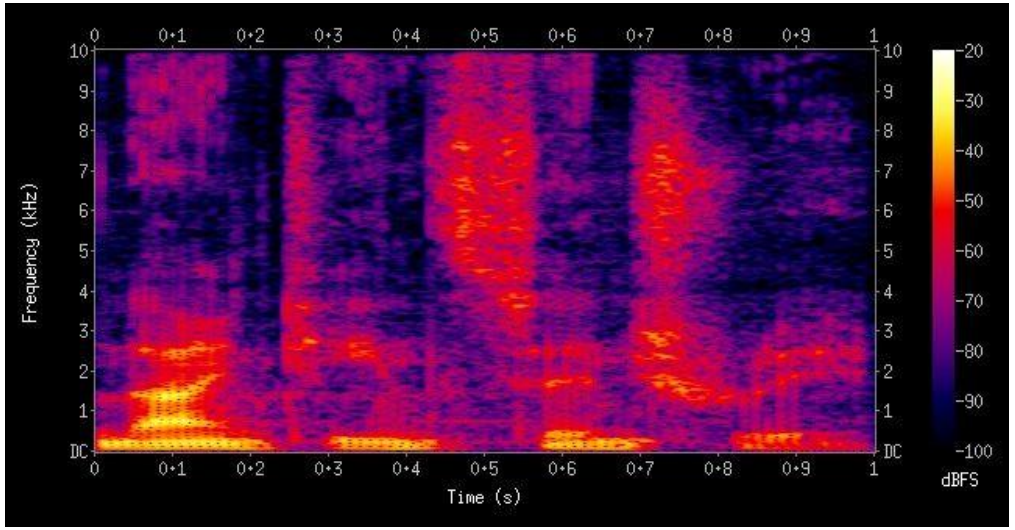
STFT parameters: Hop size

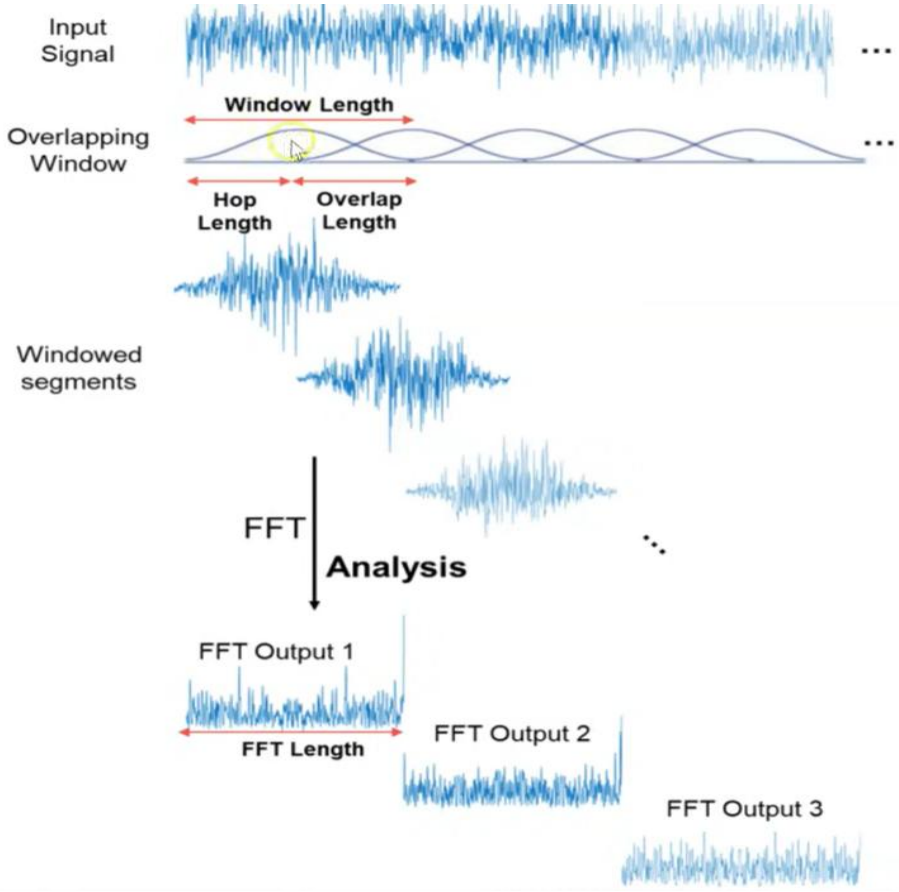
- Frame size:
  - 512, 1024, 2048, 4096, 8192, etc.
- Hop size:
  - 256, 512, 1024, 2048, 4096, etc.
  - 1/8, 1/4, 1/2 of the frame size

Visualizing sound: Spectrogram

- Examine the spectrum of STFT

$$Y(m,k) = |S(m,k)|^2$$



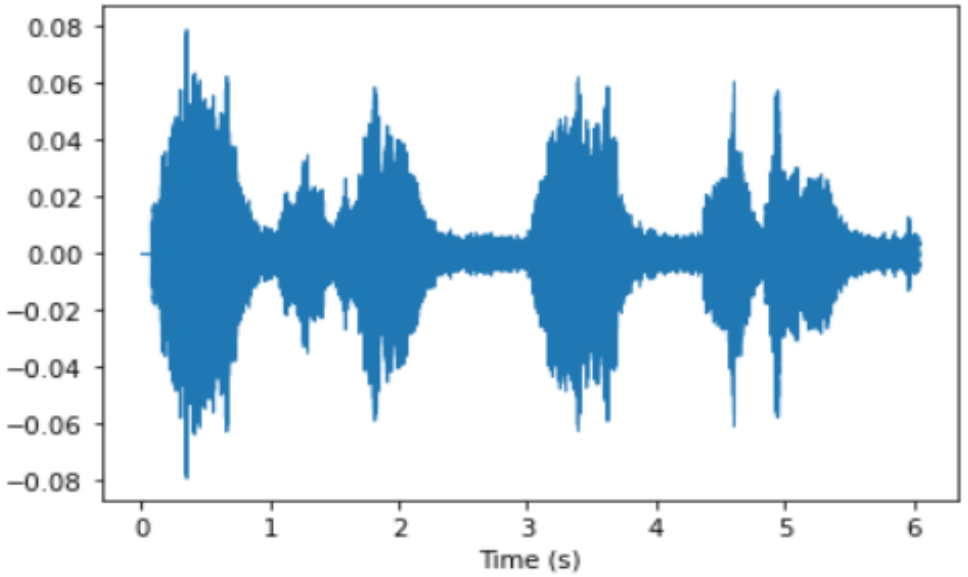


Programming STFT and spectrogram

Load sample audio file and display the raw waveform (time domain)

In [2]:
```
y, sr = librosa.load('h_1.wav', sr=32000)
librosa.display.waveshow(y, sr= sr, x_axis='s')
print("The sampled audio is returned as a numpy array (time series) and has ", y.shape, " number of samples")
print("The 10 randomly picked consequitive samples of the audio are: ", y[3000:3010])
```

```
The sampled audio is returned as a numpy array (time series) and has  (193195,)  number of samples
The 10 randomly picked consequitive samples of the audio are:  [-0.00938309 -0.01124619 -0.00986272 -0.00690551 -0.00444599 -
0.00405673
 -0.00585411 -0.00779615 -0.00679207 -0.00436785]
```



In [3]:
```
# Hear the audio
Audio('h_1.wav')
```

In [4]:

```
# Size of the Fast Fourier Transform (FFT), which will also be used as the window length
n_fft=1024

# Step or stride between windows. If the step is smaller than the window length, the windows will overlap
hop_length=320

# Specify the window type for FFT/STFT
window_type ='hann'

# Calculate the spectrogram as the square of the complex magnitude of the STFT
spectrogram_librosa = np.abs(librosa.stft(y, n_fft=n_fft, hop_length=hop_length, win_length=n_fft, window=window_type)) ** 2

print("The shape of spectrogram_librosa is: ", spectrogram_librosa.shape)
print("The size of the spectrogram is ([(frame_size/2) + 1 x number of frames])")
print("The frame size that we have specified is the number of samples to consider for the STFT. In our case, it is equal to
print("The number of frames depends on the total length of the sampled signal, the number of samples in each frame and the h
```

```
The shape of spectrogram_librosa is:  (513, 604)
The size of the spectrogram is ([(frame_size/2) + 1 x number of frames])
The frame size that we have specified is the number of samples to consider for the STFT. In our case, it is equal to the n_ff
t 1024  samples
The number of frames depends on the total length of the sampled signal, the number of samples in each frame and the hop lengt
h.
```

Reference

- A. V. Oppenheim, and A. S. Willsky. Signals & Systems. Pearson Education, 2013. (Chapter 1-3, 5, 7)
- L. R. Rabiner, and R. W. Schafer, Introduction to Digital Speech Processing, Foundations and Trends in Signal Processing 1 (1-2), 1-194, 2007.