Dhirubhai Ambani Institute of
Information and Communication
Technology

ADSP

LAB – 7 : STFT

Name:	Manish Manojkumar Prajapati
Student ID:	202411012
Branch:	MTech ICT ML (2024 - 26)
Date:	20/04/2025

EXERCISES

Question - 1) What is STFT?

Answer

- ⇨ STFT stands for **Short Time Fourier Transform**.
- ⇨ It’s a technique used to analyse how the frequency content of the signal changes over time – especially useful for non-stationary signals like speech, music or any real-world sound.

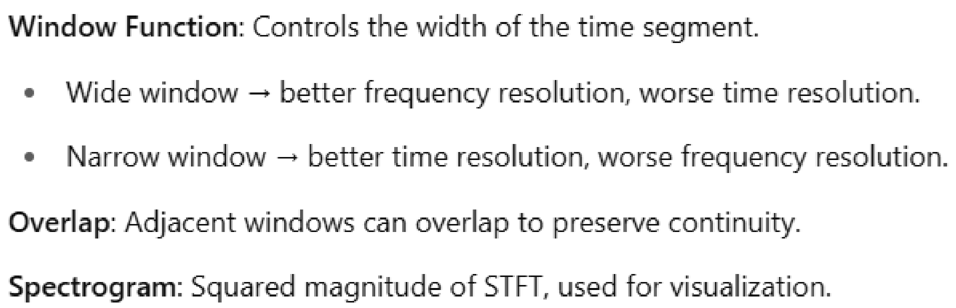
$$\mathrm{STFT}_x(t,\omega) = \int_{-\infty}^{\infty} x(\tau)w(\tau - t)e^{-j\omega\tau}\,d\tau$$

$x(\tau)$: signal

$w(\tau - t)$: window function (like Hanning, Hamming, etc.)

ω: angular frequency

t: time shift (the window center moves)



Window Function: Controls the width of the time segment.

- Wide window → better frequency resolution, worse time resolution.
- Narrow window → better time resolution, worse frequency resolution.

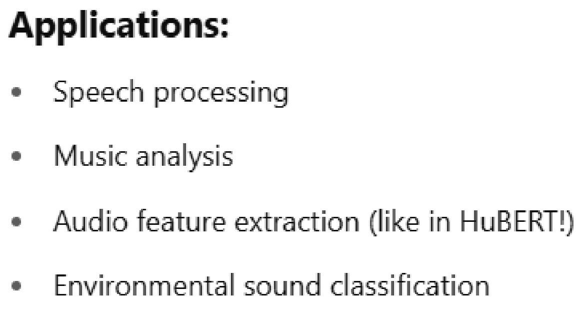
Overlap: Adjacent windows can overlap to preserve continuity.

Spectrogram: Squared magnitude of STFT, used for visualization.

Question - 2) Why we use STFT?

⇨ While a regular Fourier transform gives you all the frequencies in a signal, it loses time information. The STFT fixes that by:

1. Sliding a window across the signal
2. Taking a small segment (or 'window') at a time
3. Applying **Fourier Transform** to each segment.
4. This gives you frequency vs. time information – i.e., a time-frequency representation.



Applications:

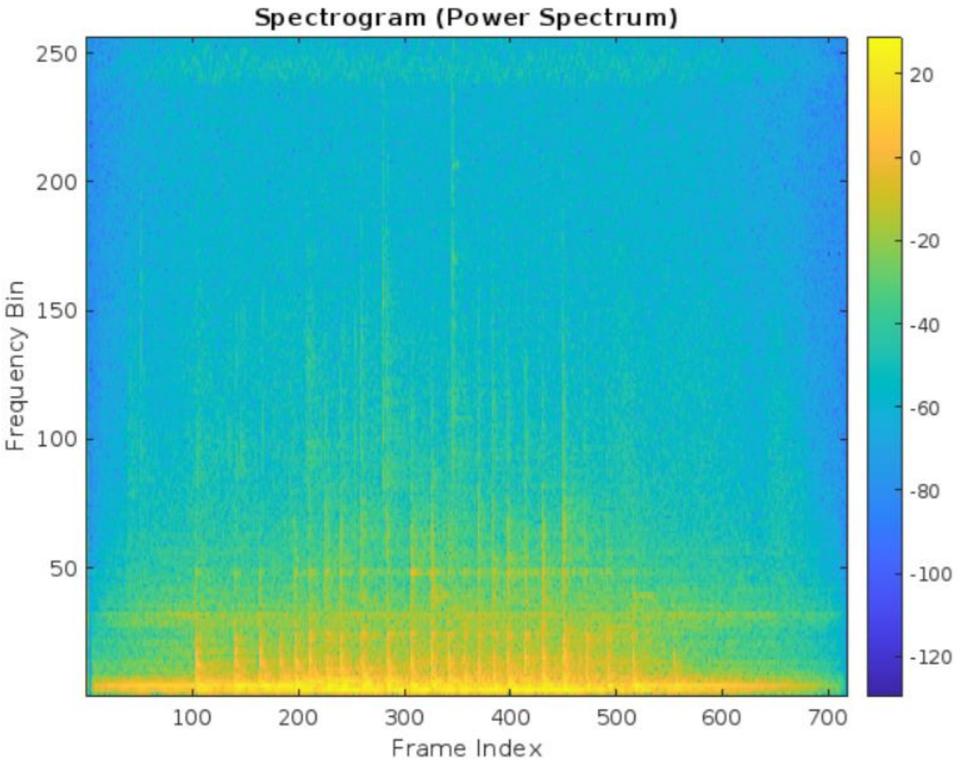
- Speech processing
- Music analysis
- Audio feature extraction (like in HuBERT!)
- Environmental sound classification

Question - 3) Take 1 audio signal. Plot STFT with frame length of 512 without overlapping.

CODE:

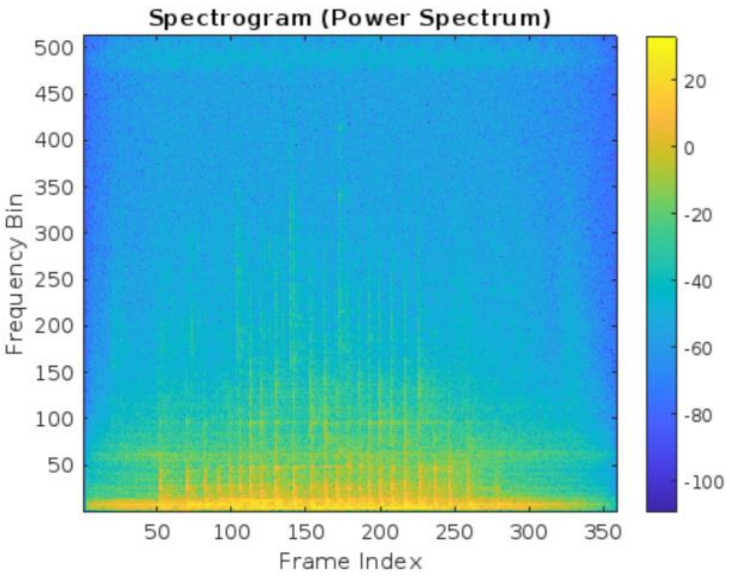
```
% 202411012
% STFT

% Reading the audio file

% Read audio
[x, Fs] = audioread('BAK.wav');     % x: audio signal, Fs: sampling frequency
x = x(:,1);                         % If stereo, take only one channel


% Parameters
frame_length = 512;               % Number of samples per frame
hop_size = 512;                   % No overlap
num_frames = floor(length(x)/frame_length);
```

```
% Initialize matrix to store power spectra
spectrogram_matrix = zeros(frame_length, num_frames);


% Loop over frames
for k = 1:num_frames
    start_idx = (k-1)*hop_size + 1;
    end_idx = start_idx + frame_length - 1;
    frame = x(start_idx:end_idx);

    % Apply window (optional, improves spectral behavior)
    windowed_frame = frame .* hanning(frame_length);

    % Compute FFT and power spectrum (magnitude squared)
    X = fft(windowed_frame);
    power_spectrum = abs(X).^2;

    % Store in matrix
    spectrogram_matrix(:,k) = power_spectrum;
end


% Plotting
imagesc(10*log10(spectrogram_matrix(1:frame_length/2, :)));  % Only plot up to
Nyquist
axis xy;
xlabel('Frame Index');
ylabel('Frequency Bin');
title('Spectrogram (Power Spectrum)');
colorbar;
```



Question - 4) Take 1 audio signal. Plot STFT with frame length of 1024 without overlapping.

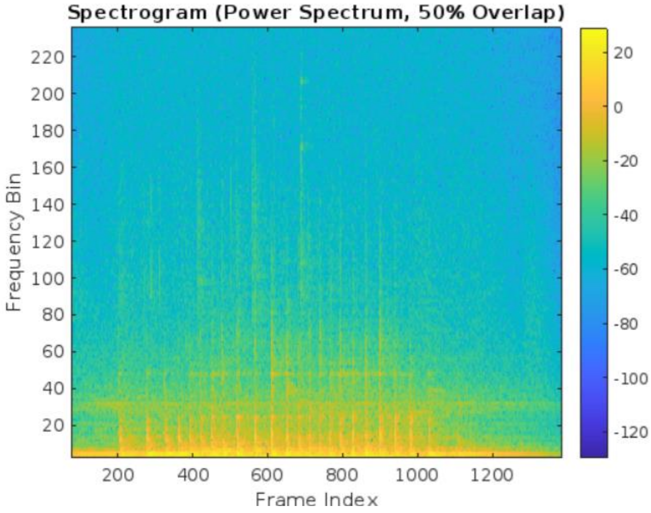
```
% 202411012
% STFT

% Reading the audio file

% Read audio
[x, Fs] = audioread('BAK.wav');    % x: audio signal, Fs: sampling frequency
x = x(:,1);                        % If stereo, take only one channel


% Parameters
frame_length = 1024;               % Number of samples per frame
hop_size = 1024;                   % No overlap
num_frames = floor(length(x)/frame_length);


% Initialize matrix to store power spectra
spectrogram_matrix = zeros(frame_length, num_frames);


% Loop over frames
for k = 1:num_frames
    start_idx = (k-1)*hop_size + 1;
    end_idx = start_idx + frame_length - 1;
    frame = x(start_idx:end_idx);

    % Apply window (optional, improves spectral behavior)
    windowed_frame = frame .* hanning(frame_length);

    % Compute FFT and power spectrum (magnitude squared)
    X = fft(windowed_frame);
    power_spectrum = abs(X).^2;

    % Store in matrix
    spectrogram_matrix(:,k) = power_spectrum;
end


% Plotting
imagesc(10*log10(spectrogram_matrix(1:frame_length/2, :)));  % Only plot up to
Nyquist
axis xy;
xlabel('Frame Index');
ylabel('Frequency Bin');
title('Spectrogram (Power Spectrum)');
colorbar;
```



Question - 5) Take 1 audio signal. Plot STFT with frame length of 512 with overlapping of 50%.

```
% 202411012
% Read audio
[x, Fs] = audioread('BAK.wav');     % x: audio signal, Fs: sampling frequency
x = x(:,1);                         % If stereo, take only one channel

% Parameters
frame_length = 512;               % Frame size
overlapping = 0.5;                % Overlappping size
hop_size = (1-overlapping)*frame_length;
num_frames = floor((length(x) - frame_length) / hop_size) + 1;

% Initialize matrix to store power spectra
spectrogram_matrix = zeros(frame_length, num_frames);

% Loop over frames
for k = 1:num_frames
    start_idx = (k-1)*hop_size + 1;
    end_idx = start_idx + frame_length - 1;
    frame = x(start_idx:end_idx);

    % Apply Hanning window
    windowed_frame = frame .* hanning(frame_length);

    % Compute FFT and power spectrum (magnitude squared)
    X = fft(windowed_frame);
    power_spectrum = abs(X).^2;

    % Store power spectrum
    spectrogram_matrix(:,k) = power_spectrum;
end

% Plot spectrogram (dB scale, only positive frequencies)
imagesc(10*log10(spectrogram_matrix(1:frame_length/2, :)));
axis xy;
```

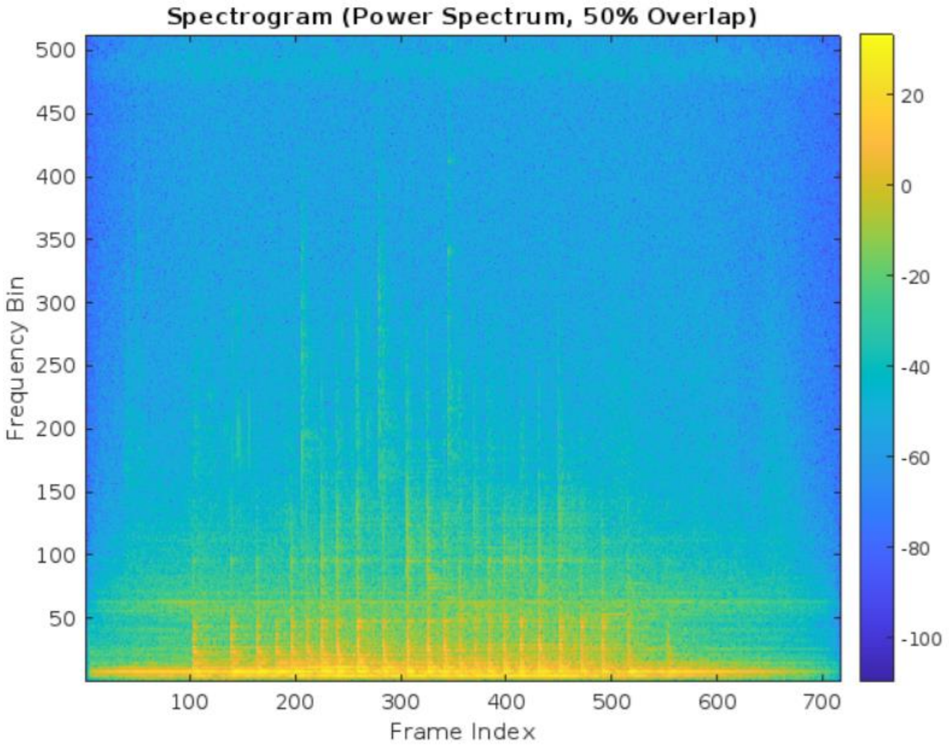
```
xlabel('Frame Index');
ylabel('Frequency Bin');
title('Spectrogram (Power Spectrum, 50% Overlap)');
colorbar;
```



Question - 6) Take 1 audio signal. Plot STFT with frame length of 1024 with overlapping of 50%.

```
% 202411012
% Read audio
[x, Fs] = audioread('BAK.wav');    % x: audio signal, Fs: sampling frequency
x = x(:,1);                        % If stereo, take only one channel

% Parameters
frame_length = 1024;               % Frame size
overlapping = 0.5;               % Overlappping size
hop_size = (1-overlapping)*frame_length;
num_frames = floor((length(x) - frame_length) / hop_size) + 1;

% Initialize matrix to store power spectra
spectrogram_matrix = zeros(frame_length, num_frames);

% Loop over frames
for k = 1:num_frames
    start_idx = (k-1)*hop_size + 1;
    end_idx = start_idx + frame_length - 1;
    frame = x(start_idx:end_idx);

    % Apply Hanning window
    windowed_frame = frame .* hanning(frame_length);

    % Compute FFT and power spectrum (magnitude squared)
    X = fft(windowed_frame);
    power_spectrum = abs(X).^2;

    % Store power spectrum
    spectrogram_matrix(:,k) = power_spectrum;
end

% Plot spectrogram (dB scale, only positive frequencies)
imagesc(10*log10(spectrogram_matrix(1:frame_length/2, :)));
axis xy;
```

```
xlabel('Frame Index');
ylabel('Frequency Bin');
title('Spectrogram (Power Spectrum, 50% Overlap)');
colorbar;
```



<><><><><><><><><><><><><><><><><><><><><><><><><><><><>

<><><><><><><><><><><><><><><><><><><><><><><><><><><><>

<><><><><><><><><><><><><><><><><><><><><><><><><><><><>