Dhirubhai Ambani Institute of Information and Communication Technology

# ADSP

# LAB – 5 : Gabor filter-bank application


**Name:** Manish Manojkumar Prajapati

**Student ID:** 202411012

**Branch:** MTech ICT ML (2024 - 26)

**Date:** 27/03/2025


## EXERCISES

**Question) Apply the Gabor filterbank to 1D & 2D signal. And prove the Heisenberg Uncertainty principle.**

**1D -> Speech/Audio Signal**

**2D -> Image**

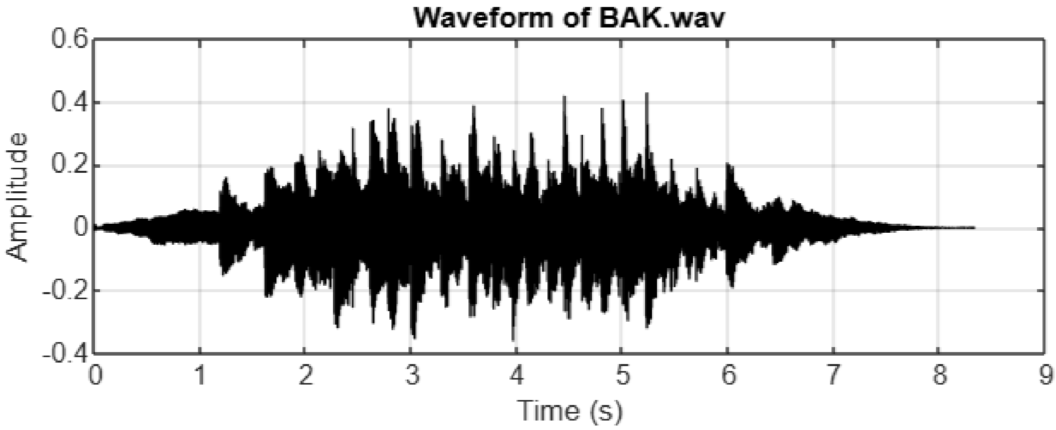
CODE:

***1-D Speech or Audio Signal***

### i) Loading the audio file

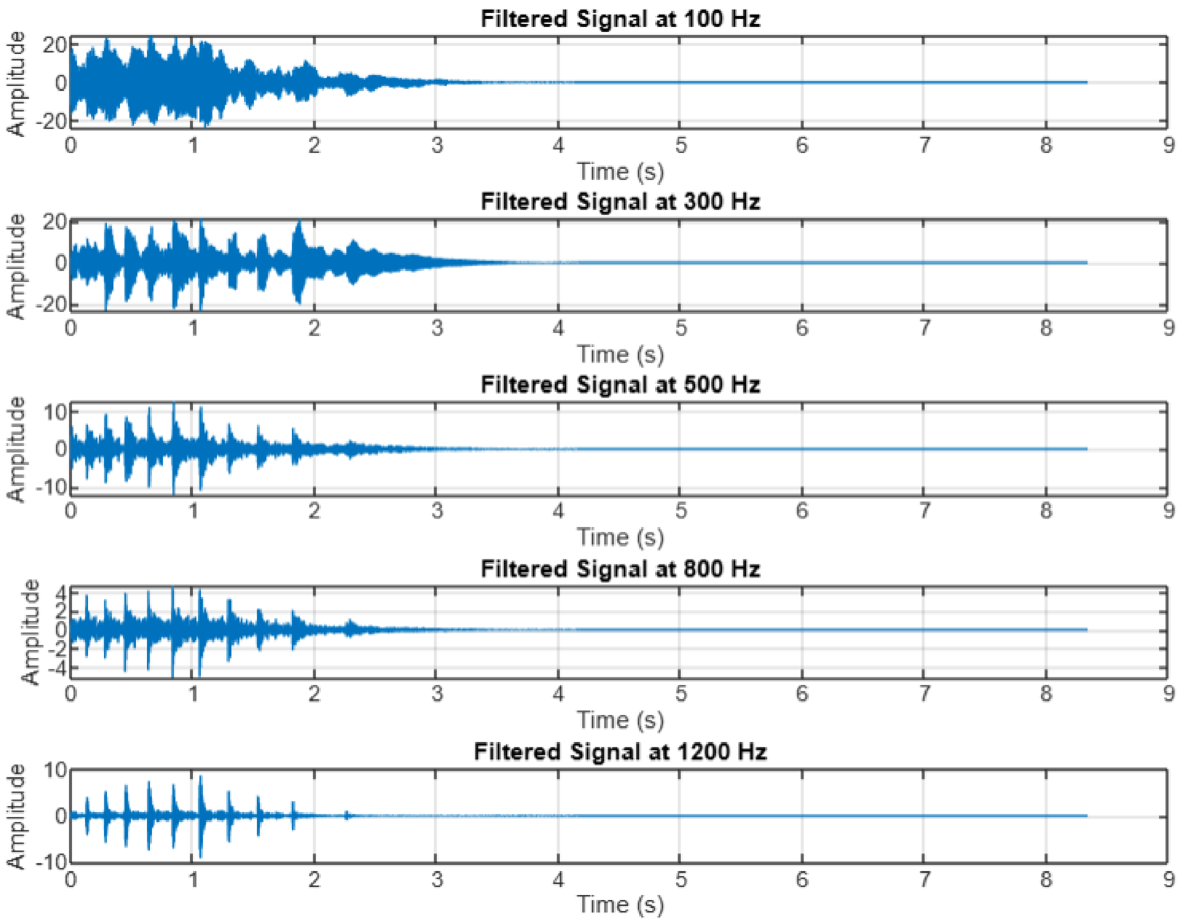
```
% 202411012
% Plotting raw signal

%% Loading the Audio file

[audio_signal, fs] = audioread("BAK.wav");

% Defining Time-Vector
t = (0: length(audio_signal)-1) / fs;

%% Plotting the Time-Domain Signal
```

```
figure;
subplot(2, 1, 1);
plot(t, audio_signal, 'k');
xlabel('Time (s)');
ylabel('Amplitude');
title('Waveform of BAK.wav');
grid on;
```




## ii) Applying Gabor filter and visualizing the transform waveform

```
%% Define Gabor Filter Parameters
f0 = [100, 300, 500, 800, 1200]; % Center frequencies in Hz
sigma_t = 0.005;  % Time-domain spread (adjustable)

filtered_signals = zeros(length(f0), length(audio_signal));

%% Apply Gabor Filters
for i = 1:length(f0)
    % Generate Gabor filter (Gaussian * Sinusoid)
    gabor_filter = exp(-t.^2 / (2 * sigma_t^2)) .* cos(2 * pi * f0(i) * t);

    % Convolve Gabor filter with audio signal
    filtered_signals(i, :) = conv(audio_signal(:,1), gabor_filter, 'same');
end

%% Visualize Filtered Waveforms
figure;
for i = 1:length(f0)
    subplot(length(f0),1,i)
    plot(t, filtered_signals(i, :));
    title(['Filtered Signal at ', num2str(f0(i)), ' Hz'])
    xlabel('Time (s)')
    ylabel('Amplitude')
    grid on;
end
```



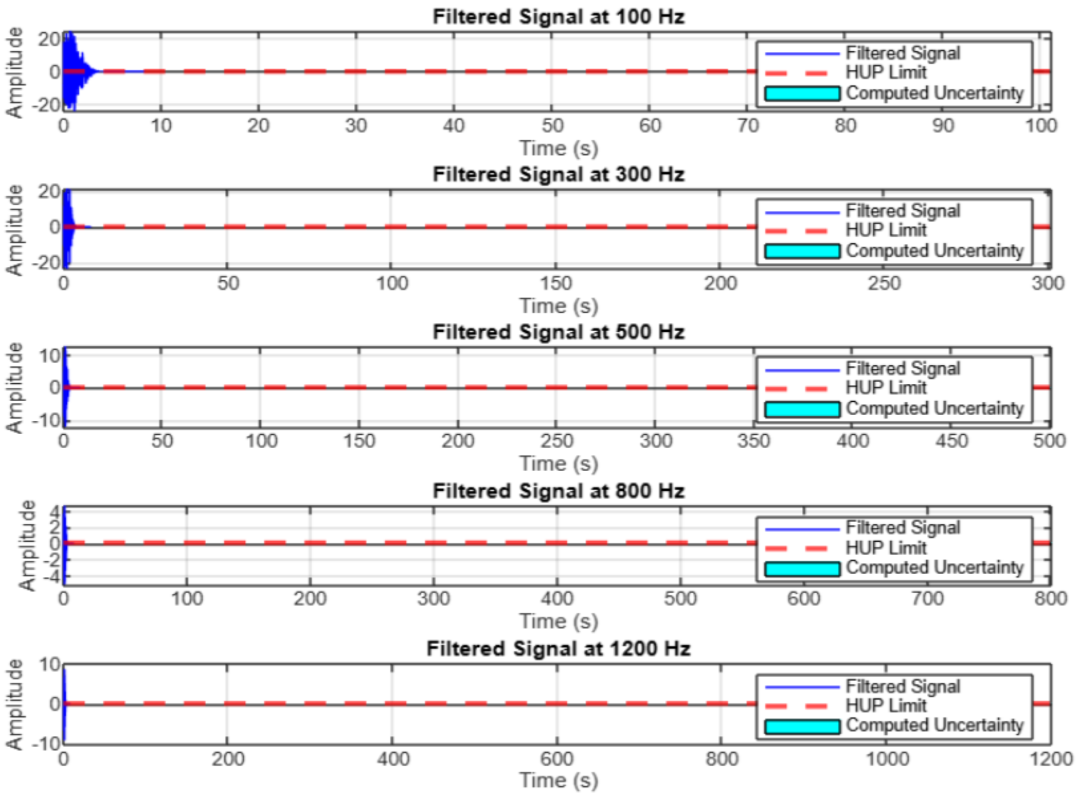

## iii) Verifying the Heisenberg Uncertainty Principle

```
%% Verify Heisenberg Uncertainty Principle

HUP_limit = 1 / (4 * pi);  % Theoretical bound
uncertainty_values = zeros(1, length(f0));  % Store σ_t * σ_f values
match_status = all(uncertainty_values >= HUP_limit); % Check match



%% Computing Time-Frequency Uncertainty

for i=1:length(f0)

    % Derived for Gaussian function
    sigma_f = 1 / (2*pi*sigma_t);

    % Compute Uncertainty
    uncertainty_values(i) = sigma_t * sigma_f;

end



%% Theoretical HUP Limit
HUP_limit = 1 / (4 * pi);
```

```
%% Visualization of Uncertainty for Each Filter
figure;
for i = 1:length(f0)
    subplot(length(f0),1,i);

    % Plot Filtered Signal
    plot(t, filtered_signals(i, :), 'b');
    hold on;

    % Add HUP Limit as a horizontal line
    yline(HUP_limit, 'r--', 'LineWidth', 2); % HUP Limit
    bar(f0(i), uncertainty_values(i), 'FaceColor', 'c'); % Computed value

    % Labels
    title(['Filtered Signal at ', num2str(f0(i)), ' Hz']);
    xlabel('Time (s)');
    ylabel('Amplitude');
    legend('Filtered Signal', 'HUP Limit', 'Computed Uncertainty');
    grid on;
end

%% Display Result
if all(uncertainty_values >= HUP_limit)
    disp('Heisenberg Uncertainty Principle holds for all Gabor filters!');
else
    disp('Difference exists! Some filters violate the HUP.');
end
```

## Output:

⇨ Heisenberg Uncertainty Principle holds for all the Gabor filters!



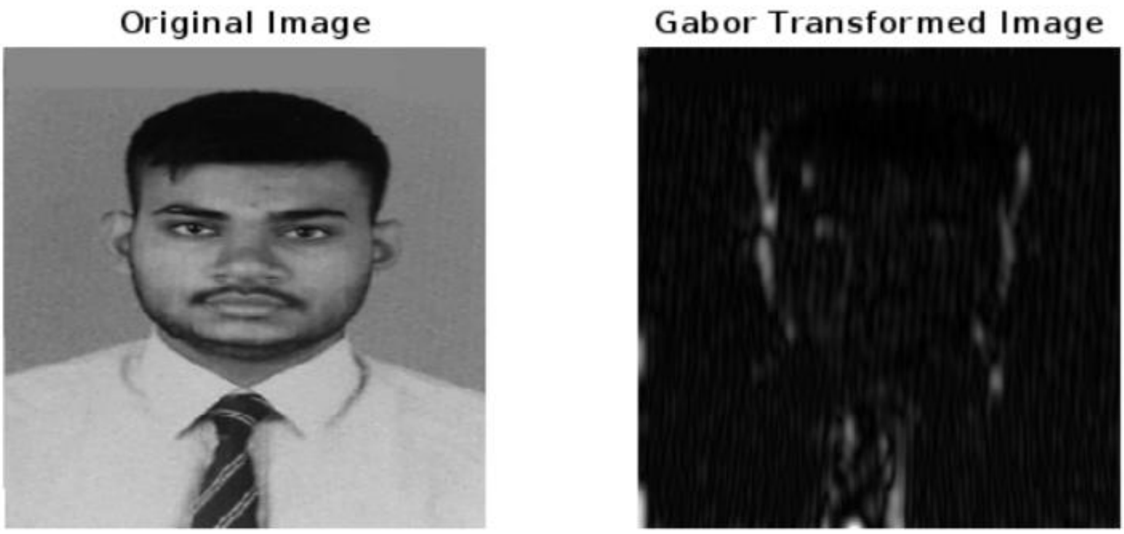
## *2-D Image*

### i) Loading the image file

```
% Loading and Converting the Image to GrayScale
img = imread('My_pic_tie_resized.png');
img = rgb2gray(img);  % Converting the image to grayscale
img = im2double(img); % Normalizing the image to [0, 1]

imshow(img);
```

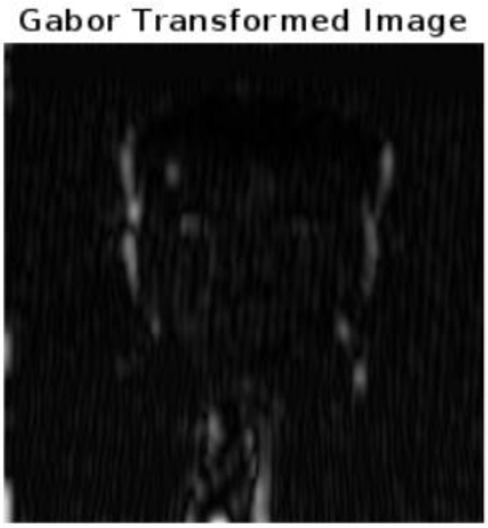
### ii) Applying the Gabor filter and visualizing the transformed image

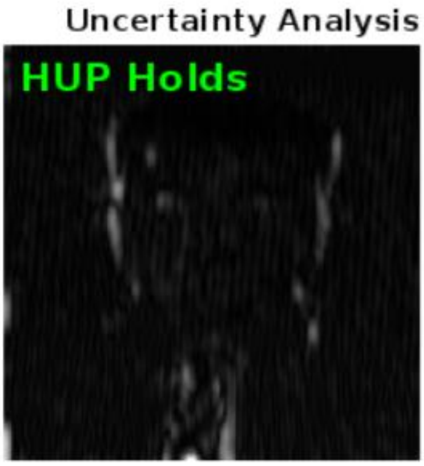
```
%% Define Gabor Filter Parameters
lambda = 4; % Wavelength (controls frequency)
theta = 0; % Orientation in degrees
psi = 0; % Phase offset
gamma = 0.5; % Aspect ratio
bw = 1; % Bandwidth (affects spread)
sigma = lambda; % Standard deviation of Gaussian envelope

%% Create and Apply Gabor Filter
gabor_filter = gabor(lambda, theta);
filtered_img = imgaborfilt(img, gabor_filter);

%% Visualize Original and Transformed Images
figure;

subplot(1,2,1);
imshow(img, []);
title('Original Image');

subplot(1,2,2);
imshow(filtered_img, []);
title('Gabor Transformed Image');
```




### iii) Verifying the Heisenberg Uncertainty Principle

```
%% Compute Spatial and Frequency Spread
sigma_x = sigma; % Spatial spread (standard deviation in space domain)
sigma_f = 1 / (2 * pi * sigma_x); % Frequency spread (Fourier domain)
uncertainty = sigma_x * sigma_f; % Compute HUP uncertainty

%% Heisenberg's Uncertainty Principle Limit
HUP_limit = 1 / (4 * pi);

%% Visualizing Results
figure;

% Plot the Gabor Transformed Image
subplot(1,2,1);
imshow(filtered_img, []);
title('Gabor Transformed Image');

% Highlighting Uncertainty Differences
subplot(1,2,2);
hold on;
imshow(filtered_img, []);
title('Uncertainty Analysis');

% Check if HUP is violated
if uncertainty < HUP_limit
    disp('Heisenberg Uncertainty Principle is violated for this transformation!');
    text(10, 20, 'HUP Violated!', 'Color', 'red', 'FontSize', 14, 'FontWeight',
'bold');
else
    disp('HUP holds for this transformation.');
    text(10, 20, 'HUP Holds', 'Color', 'green', 'FontSize', 14, 'FontWeight',
'bold');
end

hold off;
```




---

**Question – 1: What is Gabor filter bank? Why is it useful?**

Answer:

Thursday

# LAB-5.

27/3/25

Q: What is gabor filterbank? Why it is used?

Ans: → A Gabor filterbank is a collection of Gabor filters with different frequencies, orientations, and scales.

→ Instead of applying just one Gabor filter, we apply multiple filters tuned to capture various ~~filters~~ features in an image or signal.

→ A Gabor filter is a Gaussian-modulated sinusodial wave, which can be written as:

$$\boxed{g(x,y) = \exp\left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2}\right) \cdot \cos\left(2\pi\frac{x'}{\lambda} + \phi\right)}$$

$x', y'$ = Rotated coordinates

$\lambda$ = wavelength [controls frequency]

$\theta$ = Orientation [angle of filter]

$\sigma$ = Standard deviation [controls spread]

$\phi$ = phase offset

$\gamma$ = aspect ratio [elongation control]

# Why to use a Gabor Filter Bank?

→ Instead of using a single Gabor filter, we apply multiple Gabor filters with:-

+ Different orientations:-
  e.g.:- 0°, 45°, 90°, 135°

* Different frequencies:-
  e.g.:- low frequency for coarse details,
  high frequency for fine textures.

⇒ This helps in:-

↳ Feature Extraction:- Extracts edges, textures, and frequency components of signals/images.

↳ Pattern Recognition:- Used in speech processing, face recognition, texture analysis, and medical imaging.

↳ Time Frequency Analysis:- Provides a localized representation of time and frequency components.

# Applications of Gabor filter banks:-

→ Speech Processing:- Time-frequency analysis of audio signals.

→ Image Processing:- Texture analysis,

fingerprint recognition, iris recognition.

→ Biomedical Imaging:- Brain MRI texture analysis.

→ Object recognition:- Feature extraction in machine learning.

⇒ Relating to Heisenberg's Uncertainty Principle:-

↳ Gabor filters are optimal in satisfying the time-frequency tradeoffs.

↳ They minimize uncertainty because they have optimal localization in both spatial and frequency domains.

---

**Question – 2: What is Heisenberg Uncertainty Principle?**

Answer:

Q:- What is Heisenberg's Uncertainty Principle?

⇒ The Heisenberg Uncertainty Principle (HUP) states that it is impossible to simultaneously determine the exact position and momentum of a particle with absolute precision.

→ Instead, the more precisely you know one, the less ~~precise~~ precisely you can know the other.

→ Mathematically; it is expressed as:-

$$\boxed{\sigma_x \cdot \sigma_p \geq \frac{\hbar}{2}}$$

where; $\sigma_x$ = uncertainty in position
$\sigma_p$ = uncertainty in momentum
$\hbar$ = Reduced Planck's constant [$h/2\pi$]

→ This means that increasing precision in one domain increases uncertainty in the other.

⇒ Heisenberg's Uncertainty in Time-Frequency Analysis:-

↳ In signal processing, the HUP translates to time and frequency uncertainty, meaning:-

$$\sigma_t \cdot \sigma_f \geq \frac{1}{4\pi}$$

where;
$\sigma_t$ = Uncertainty in time [temporal resolution]
$\sigma_f$ = Uncertainty in frequency [spectral resolution]

↳ This means that a signal cannot be perfectly localized in both time and frequency domains:-

+ Short pulses: [good time localization]
  ↳ Poor time localization.

+ Long-duration signals:- [good freq. localization] → Poor time localization.

→ This trade-off is the reason why Gabor filters are widely used – they are designed to minimize this uncertainty

e.g.: Audio signals: A sharp impulse in time has broad frequency content [e.g., drum hit].

→ Spectrograms:- Higher time resolution means lower frequency resolution, and vice-versa.

→ Quantum Mechanics:- Electron positions in an atom follow probability distributions due to HUP.