**PROJECT BASED LEARNING REPORT**

on

**“DENOISING MEDICAL IMAGES USING FIR AND IIR FILTER ”**

Submitted in the partial fulfillment of the requirements

for the Project based learning (PBL) in **DIGITAL SIGNAL PROCESSING**

in

Electronics & Communication Engineering

By

PRN: Name of the Student:

2214110411 SOUMYA DEEP

2214110426 PRIYANSHU RAJ 2214110441 VISHAL KUMAR

Under the guidance of Course In-charge

Prof. S.R. MADKAR



Department of Electronics & Communication Engineering

Bharati Vidyapeeth (Deemed to be University)

College of Engineering,

Pune – 4110043

**Academic Year: 2024-25**

**CERTIFICATE**

Certified that the Project Based Learning report entitled, “**Denoising Medical Images using FIR Filter (Finite Impulse Response ) and IIR Filter (Infinite Impulse Response )”** is work done by:

**2214110411 SOUMYA DEEP**

**2214110426 PRIYANSHU RAJ**

**2214110441 VISHAL KUMAR**

in partial fulfillment of the requirements for the award of credits for Project Based Learning (PBL) in **Digital Signal Processing** of Bachelor of Technology Semester V, in Department of Electronics and Communication Engineering.

**Date:**

**Prof. S. R. MADKAR Dr. Arundhati A. Shinde**

**Course In-charge Professor & Head**

**INDEX:**

|  |  |  |
| --- | --- | --- |
| **SR NO.** | **TITLE** | **PAGE NO.** |
| 1. | Problem Statement | 1 |
| 2. | Introduction to filters | 2 |
| 3. | Types of filters | 8 |
| 4. | Software used | 12 |
| 5. | Steps to execute the program | 18 |
| 6. | Code | 19 |
| 7. | Results | 20 |
| 8. | Project outcome and conclusion | 22 |

**PROBLEM STATEMENT:**

**Implement a Denoising algorithm for medical images using Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters.**

Denoising is a crucial step in medical image processing, as it helps improve the quality and reliability of diagnostic information extracted from noisy images. Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters are commonly used techniques for image denoising.

**INTRODUCTION TO FILTERS :**

Filters in the context of image processing refer to operations applied to an image to enhance or modify its visual appearance or extract specific features. There are various types of filters used for different each designed for specific purposes. Here are some common types of filters:

IIR (Infinite Impulse Response) filters and FIR (Finite Impulse Response) filters are two types of digital filters used in signal processing.

They have different characteristics, design methods, and applications.IIR Filters (Infinite Impulse Response):

**Characteristics of FIR Filters**

1.Finite Impulse Response:

The impulse response of FIR filters is nonzero for a finite duration, meaning they respond to inputs for a limited number of samples.

2.Linear Phase Response:

FIR filters can be designed to have a linear phase response, ensuring that all frequency components are delayed by the same amount of time, preserving the signal shape.

3.Stability:

FIR filters are always stable because they do not use feedback, which eliminates the possibility of oscillation.

4.Design Simplicity:

FIR filters can be easier to design, especially when targeting specific frequency characteristics, and there are established techniques for their design.

5.Memory Requirements:

The memory requirement is proportional to the number of filter taps (coefficients), making them suitable for real-time applications where memory is limited.

6.Coefficient Quantization:

FIR filters are less sensitive to coefficient quantization errors compared to IIR filters, enhancing their reliability in practical implementations.

**Characteristics of IIR Filters**

1.Infinite Impulse Response:

The impulse response of IIR filters theoretically lasts indefinitely, meaning they can respond to an input signal for an infinite amount of time.

2.Nonlinear Phase Response:

IIR filters typically do not have a linear phase response, which can introduce phase distortion in the filtered signal.

3.Potential Stability Issues:

IIR filters can become unstable due to feedback, requiring careful design and analysis to ensure stability.

4.Efficient in Design:

IIR filters often require fewer coefficients than FIR filters to achieve a similar frequency response, making them more efficient in terms of computational resources.

5.Feedback Mechanism:

IIR filters use feedback, which allows them to use past output values to compute the current output, creating a more complex structure.

6.Sensitivity to Coefficient Quantization:

IIR filters are more sensitive to quantization errors in their coefficients, which can lead to instability or degradation of filter performance.

|  |  |  |
| --- | --- | --- |
| **Characterstic** | **Finite** | **Infinite** |
| Impulse Response | Finite | Infinite |
| Phase Response | Linear | Non Linear |
| Stability | Always Stable | Potentially Unstable |
| Design Complexity | Simpler | More Complex |
| Memory Requirements | Proportional to taps | Fewer taps for similar requirements |
| Coefficient Sensitivity | Less Sensitive | More Sensitive |

**Design of FIR Filters:**

FIR filter design can be approached using various methods. Here are some common techniques:

1. **Windowing Method:**

Design the ideal filter's impulse response and apply a window function to reduce ripples and truncation effects.

* Steps:

Define the desired frequency response.

Compute the ideal impulse response ℎ[𝑛].

Select a window function (e.g., Hamming, Hanning, Blackman).

Multiply the ideal response by the window function.

Normalize the coefficients if necessary.

1. **Frequency Sampling Method:**

Specify the desired frequency response at discrete frequency points and compute the inverse Fourier transform to get the time-domain coefficients.

* Steps:

Define the desired frequency response.

Sample the frequency response at 𝑁 points.

Use the inverse discrete Fourier transform (IDFT) to obtain the impulse response.

1. **Least Squares Method:**

Minimize the error between the desired frequency response and the actual response.

* Steps:

Define the desired frequency response.

Formulate the error function.

Use optimization techniques to find the filter coefficients.

1. **Parks-McClellan Algorithm:**

An optimal method for designing equiripple FIR filters.

* Steps:

Specify the desired frequency bands and ripples.

Use the Parks-McClellan algorithm to compute the filter coefficients.

**Design of IIR Filters:**

IIR filter design is more complex due to the feedback involved. Common methods include:

1. **Analog Filter Design:**

Design an analog filter using classical methods (e.g., Butterworth, Chebyshev, Elliptic).

Use techniques like transformations (e.g., bilinear transform) to convert analog filters into digital IIR filters.

* Steps:

Design the desired analog filter.

Apply the bilinear transformation to obtain the digital filter coefficients.

1. **Bilinear Transform Method:**

A mathematical technique to map analog filter poles and zeros to the digital domain.

* Steps:

Specify the desired analog filter's transfer function.

Apply the bilinear transformation to convert to the digital domain.

1. **Impulse Invariance Method:**

Preserve the impulse response of the analog filter when converting it to a digital filter.

* Steps:

Design the analog filter.

Sample the impulse response at a certain frequency to obtain the digital filter.

1. **Direct Design:**

Specify the desired poles and zeros directly in the digital domain.

* Steps:

Choose pole and zero locations based on the desired frequency response.

Calculate the filter coefficients from these poles and zeros.

| **Characteristic** | **FIR Filters** | **IIR Filters** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| Impulse Response | Finite | Infinite |

|  |  |  |
| --- | --- | --- |
| Magnitude Response | Can be designed to be flat | Dependent on poles/zeros |

|  |  |  |
| --- | --- | --- |
| Phase Response | Can be linear | Usually nonlinear |

|  |  |  |
| --- | --- | --- |
| Stability | Always stable | Can be unstable |

**PHASE RESPONSE OF FIR FILTER AND IIR FILTER :**

| **Characteristic** | **FIR Filters** | **IIR Filters** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| Phase Response Type | Linear | Generally Nonlinear |

|  |  |  |
| --- | --- | --- |
| Group Delay | Constant | Varies with frequency |

|  |  |  |
| --- | --- | --- |
| Impact on Signal Shape | Preserves shape | May distort shape |

**Computational Complexity of FIR and IIR Filters:**

The computational complexity of digital filters is an important factor in their design, especially for real-time applications. The complexity is generally measured in terms of the number of multiplications and additions required to process a signal.

**FIR Filters**

1. Structure

FIR filters have a finite impulse response characterized by a set of coefficients. The output of an FIR filter is computed as a weighted sum of its input samples.

2. Computational Complexity

For an FIR filter with N coefficients (or taps), the output

[𝑛] can be expressed as: [𝑛]=∑𝑘=0𝑁−1ℎ[𝑘]𝑥[𝑛−𝑘]

Where:

ℎ[𝑘] are the filter coefficients,

[𝑛−𝑘] are the input samples.

Operations Count:

Multiplications:

𝑁 (one for each coefficient)

Additions:

𝑁−1 (to sum the products)

Thus, the total number of operations (multiplications + additions) for each output sample is:

3.Complexity

FIR=𝑁+(𝑁−1)=2𝑁−1≈𝑂(𝑁)

**IIR Filters**

1. Structure

IIR filters have an infinite impulse response due to feedback, characterized by both numerator and denominator coefficients.

2. Computational Complexity

The output of an IIR filter can be represented as:

[𝑛]=∑𝑘=0[𝑘][𝑛−𝑘]−∑𝑗=1𝐿𝑎[𝑗]𝑦[𝑛−𝑗]

y[n]= k=0∑Mb[k]x[n−k]− j=1∑La[j]y[n−j]

Where:

[𝑘] are the numerator coefficients,

[𝑗] are the denominator coefficients,

𝑀 is the order of the filter,

𝐿 is the number of feedback coefficients.

Operations Count:

Multiplications for feedforward path:

𝑀+1 (for 𝑀 numerator coefficients and the current input)

Multiplications for feedback path:

(for)

L feedback coefficients)

Additions:

(𝑀+1+𝐿−1)

Thus, the total number of operations (multiplications + additions) for each output sample is:

Complexity

IIR

​

=(M+1+L)+(M+L−1)=2M+2L≈O(M+L)

**Applications of FIR and IIR Filters:**

Both FIR and IIR filters have a wide range of applications across various fields due to their unique characteristics. Here are some common applications for each type of filter:

Applications of FIR Filters

**Audio Processing:**

Equalization: FIR filters are used in audio equalizers to adjust the balance between frequency components.

Noise Reduction: They can help eliminate unwanted noise from audio signals.

Digital Communication:

Pulse Shaping: FIR filters are used to shape pulses in digital communication to minimize inter-symbol interference (ISI).

Channel Equalization: FIR filters are used to compensate for distortion introduced by the communication channel.

**Image Processing:**

Edge Detection: FIR filters can be designed as convolution kernels to detect edges in images.

Smoothing and Blurring: FIR filters are often used for image smoothing to reduce noise.

Biomedical Applications:

ECG Signal Processing: FIR filters are used to filter out noise from electrocardiogram (ECG) signals.

Biomedical Imaging: Used in filtering techniques for enhancing images in medical imaging

**Control Systems:**

Signal Conditioning: FIR filters can be used to condition signals in control applications, ensuring stability and performance.

Applications of IIR Filters

Audio Processing:

Reverb and Echo Effects: IIR filters are often used in digital audio effects to create reverberation and echo effects due to their feedback capabilities.

Active Noise Cancellation: IIR filters can adaptively filter noise in headphones or audio devices.

**Digital Communication:**

Decimation and Interpolation: IIR filters are used in systems that require changing the sampling rate of a signal efficiently.

Adaptive Equalization: Used in adaptive equalizers for channel compensation in communication systems.

Control Systems:

Feedback Control: IIR filters are suitable for designing controllers in feedback loops due to their recursive structure.

Sensor Filtering: They are used in filtering sensor data in real-time control applications.

Image Processing:

Image Enhancement: IIR filters can be used for real-time image enhancement applications, such as sharpening and contrast adjustment.

Biomedical Applications:

Real-Time Signal Processing: IIR filters are often used in applications like blood pressure monitoring, where low-latency filtering is crucial.

Summary of Differences in Applications

FIR Filters:

Preferred for applications requiring linear phase response, stability, and simple implementation.

Common in audio processing, image processing, and biomedical applications where precision and predictability are vital.

IIR Filters:

More efficient for applications needing lower computational load with fewer coefficients.

Suitable for real-time applications in audio effects, adaptive filtering, and feedback control systems.

Both FIR and IIR filters play crucial roles in modern signal processing, and the choice between them depends on specific application requirements, including performance, computational efficiency, and system stability.

**Types of FIR Filters:**

**Linear Phase FIR Filters:**

Designed to have a constant group delay across all frequencies.

Ensure that all frequency components are delayed by the same amount, preserving the waveform shape.

Commonly used in applications where phase distortion must be minimized.

**Symmetric FIR Filters:**

Coefficients are symmetric around the center tap.

This symmetry results in a linear phase response.

Example:

h[n]=h[N−n]

**Antisymmetric FIR Filters:**

Coefficients are antisymmetric, leading to a linear phase response with a phase shift.

Example:

h[n]=−h[N−n]

**Windowed FIR Filters:**

Designed using a window function applied to an ideal filter response.

Common window functions include Hamming, Hanning, and Blackman windows.

Useful for controlling side lobes and achieving a desired frequency response.

**Multirate FIR Filters:**

Used in applications requiring sampling rate changes (decimation or interpolation).

These filters help to minimize aliasing effects during sampling rate conversions.

**Types of IIR Filters:**

**Butterworth Filters:**

Known for their maximally flat frequency response in the passband.

They have a smooth transition from passband to stopband with no ripple.

**Chebyshev Filters:**

Designed to have a ripple in the passband (Type I) or stopband (Type II).

Chebyshev Type I filters offer a steeper roll-off than Butterworth filters.

**Elliptic Filters:**

Provide the steepest roll-off for a given order and ripple in both passband and stopband.

They are characterized by having ripples in both the passband and stopband, offering high performance in limited space.

**Bessel Filters:**

Known for their maximally flat group delay, making them suitable for applications where phase linearity is important.

Often used in audio and video applications to preserve the waveform shape.

**All-Pass Filters:**

Designed to pass all frequencies with equal gain but can introduce a phase shift.

Useful for phase manipulation in various applications.

| **Filter Type** | **FIR Filters** | **IIR Filters** |
| --- | --- | --- |

|  |  |  |
| --- | --- | --- |
| Linear Phase | Designed for constant group delay | Not typically linear phase |

|  |  |  |
| --- | --- | --- |
| Symmetric | Coefficients are symmetric | Not applicable |

|  |  |  |
| --- | --- | --- |
| Antisymmetric | Coefficients are antisymmetric | Not applicable |

|  |  |  |
| --- | --- | --- |
| Windowed | Created using window functions | Not applicable |

|  |  |  |
| --- | --- | --- |
| Butterworth | Not applicable | Flat passband |

|  |  |  |
| --- | --- | --- |
| Chebyshev | Not applicable | Ripple in passband |

|  |  |  |
| --- | --- | --- |
| Elliptic | Not applicable | Steepest roll-off |

|  |  |  |
| --- | --- | --- |
| Bessel | Not applicable | Flat group delay |

|  |  |  |
| --- | --- | --- |
| All-Pass | Not applicable | Equal gain phase shift |

**Advantages of FIR and IIR Filters:**

Both FIR and IIR filters have their unique advantages, making them suitable for different applications. Here’s a comparison of the advantages of each type:

**Advantages of FIR Filters**

1. **Linear Phase Response :**

* FIR filters can be designed to have a linear phase response, ensuring that all frequency components are delayed equally. This preserves the shape of the signal waveform, which is crucial in many applications.

1. **Stability**:

* FIR filters are inherently stable because they do not use feedback in their structure. As a result, they cannot produce oscillations, making them suitable for critical applications.

1. **Flexibility in Design**:

* FIR filters can be designed to achieve a wide range of frequency responses, including sharp transitions and specific shapes, using techniques such as windowing or the Parks-McClellan algorithm.

1. **Simple Implementation**:

* The implementation of FIR filters is straightforward, especially in fixed-point arithmetic, as they involve only additions and multiplications of current and past input samples.

1. **No Limitations on Order**:

* The performance can be improved by simply increasing the number of taps (coefficients) without concerns about stability.

1. **Less Sensitivity to Coefficient Quantization**:

* FIR filters tend to be less sensitive to quantization errors in the coefficients compared to IIR filters, which is advantageous in fixed-point implementations.

### **Advantages of IIR Filters**

1. **Efficiency**:

* IIR filters typically require fewer coefficients than FIR filters to achieve a similar frequency response, making them computationally efficient. This is especially beneficial in real-time applications.

1. **Steep Roll-off**:

* IIR filters can achieve a steeper roll-off between the passband and stopband compared to FIR filters, allowing for a more compact design in terms of filter order.

1. **Low Latency**:

* Because IIR filters can provide a desired frequency response with fewer taps, they often result in lower latency, making them suitable for real-time applications where speed is crucial.

1. **Recursive Structure**:

* The feedback mechanism in IIR filters can produce complex frequency responses that are difficult to achieve with FIR filters, making them versatile for various applications.

1. **Widely Used in Analog Filters**:

* The design of IIR filters is often straightforward due to their relationship with analog filter designs (such as Butterworth, Chebyshev), facilitating easy transition from analog to digital.

**Disadvantages of FIR and IIR Filters:**

While both FIR and IIR filters offer distinct advantages, they also come with certain drawbacks. Here’s an overview of the disadvantages of each type:

**Disadvantages of FIR Filters:**

* Higher Computational Complexity:

FIR filters generally require a larger number of coefficients (taps) to achieve a given frequency response compared to IIR filters. This leads to more multiplications and additions, increasing the computational burden.

* Increased Memory Usage:

The need for more coefficients also means that FIR filters may require more memory for storage, which can be a concern in resource-constrained environments.

* Slower Response in Real-Time Applications:

Due to the increased number of calculations required, FIR filters may exhibit a slower response time in real-time applications compared to IIR filters.

* Limitations on Filter Order:

To achieve sharp transitions in frequency response, FIR filters often require a higher order, which may not be practical in all scenarios.

* Potential for Group Delay Variability:

Although FIR filters can have linear phase responses, achieving this may require careful design, and not all FIR filters will inherently have constant group delay.

**Disadvantages of IIR Filters:**

* Stability Concerns:

IIR filters can be unstable due to their feedback structure. If not designed carefully, they may produce oscillations or divergent outputs, especially for high-order filters.

* Non-Linear Phase Response:

IIR filters do not guarantee a linear phase response, which can distort the signal waveform and affect the integrity of the processed signal in applications where phase linearity is crucial.

* Sensitivity to Coefficient Quantization:

IIR filters are more sensitive to quantization errors in the coefficients, which can lead to performance degradation in fixed-point implementations.

* Complex Design Process:

Designing IIR filters can be more complex compared to FIR filters, especially when ensuring stability and achieving the desired frequency response.

* Difficulties in Realizing Higher-Order Filters:

As the order of the filter increases, the design and implementation of IIR filters become more complicated, and it may be challenging to maintain stability.

**SOFTWARE USED:**

**Introduction to MATLAB**

MATLAB (Matrix Laboratory) is a high-level programming language and interactive environment used primarily for numerical computation, visualization, and programming. It was developed by MathWorks and has become a standard tool in various fields, including engineering, physics, finance, and data science. MATLAB is particularly known for its powerful matrix manipulation capabilities and extensive built-in functions.

**Key Features of MATLAB**

Matrix and Array Mathematics:

MATLAB is designed for matrix and array operations. It supports a variety of matrix operations, making it ideal for linear algebra applications.

* Built-in Functions:

MATLAB includes a vast library of built-in functions for mathematical operations, statistics, optimization, and more, which simplifies the development process.

* Visualization:

MATLAB provides extensive tools for data visualization. Users can create 2D and 3D plots, animations, and interactive graphics to analyze data effectively.

* Toolboxes:

MATLAB offers numerous specialized toolboxes that extend its functionality for specific applications, such as Signal Processing, Image Processing, Control Systems, Machine Learning, and more.

**Simulink:**

Simulink is an add-on product to MATLAB that provides an environment for modeling, simulating, and analyzing dynamic systems. It uses a graphical block diagram approach for system design.

* High-Level Programming Language:

MATLAB allows users to write scripts and functions in a high-level language, making it easier to implement complex algorithms compared to lower-level languages.

* Interactivity:

Users can run commands interactively in the MATLAB command window, making it easy to test snippets of code and explore data quickly.

* Integration:

MATLAB can interface with other programming languages (such as C, C++, and Python) and software (such as Excel and databases) to enhance its capabilities.

**Applications of MATLAB**

* Engineering and Scientific Research:

MATLAB is widely used in engineering disciplines for simulations, model development, and system design. It is essential in fields like electrical, mechanical, and civil engineering.

* Data Analysis and Visualization:

Researchers and data analysts use MATLAB to manipulate data sets, perform statistical analyses, and visualize results effectively.

* Control System Design:

Control engineers utilize MATLAB and Simulink for modeling, simulating, and designing control systems, including PID controllers and state-space models.

* Signal and Image Processing:

MATLAB is a powerful tool for processing and analyzing signals and images, with built-in functions and toolboxes for filtering, transformation, and feature extraction.

* Machine Learning and Artificial Intelligence:

The Statistics and Machine Learning Toolbox in MATLAB provides functions for implementing various machine learning algorithms, facilitating tasks like classification, regression, and clustering.

* Finance and Econometrics:

Financial analysts use MATLAB for quantitative finance, risk management, and econometric modeling. Its capabilities include time-series analysis and option pricing.

* Education:

MATLAB is widely used in academic institutions for teaching mathematics, engineering, and computer science concepts due to its intuitive syntax and visual capabilities.

Getting Started with MATLAB

**Installation:**

MATLAB can be installed on various operating systems, including Windows, macOS, and Linux. Users can obtain a license through academic institutions, companies, or personal licenses.

**MATLAB Environment:**

The MATLAB environment includes:

Command Window: For executing commands and scripts interactively.

Workspace: Displays the variables currently in memory.

Command History: Logs previously executed commands.

Editor: For writing and editing scripts and functions.

* Basic Syntax:

Variables are created using the assignment operator (=), and functions are called using parentheses.

For example:

x = 5; % Assigning a value

y = sin(x); % Calling a function

* Scripts and Functions:

Scripts are files with a .m extension that contain a series of MATLAB commands. Functions are defined using the function keyword and can take input arguments and return outputs.

Example of a simple function:

function result = square(num)

result = num^2;

end

**Visualization in MATLAB**

MATLAB’s visualization capabilities are one of its standout features. Users can create various types of plots and customize them easily. Here are some common plotting functions:

* 2D Plotting:

Basic 2D plots can be created using the plot function:

x = 0:0.1:10; % Generate values from 0 to 10

y = sin(x); % Compute the sine of x

plot(x, y); % Create a 2D plot

title('Sine Function');

xlabel('x');

ylabel('sin(x)');

* 3D Plotting:

3D plots can be created using functions like mesh, surf, and scatter3.

[X, Y] = meshgrid(-5:0.5:5, -5:0.5:5);

Z = sqrt(X.^2 + Y.^2);

surf(X, Y, Z);

title('3D Surface Plot');

* Customizing Plots:

MATLAB provides extensive options for customizing plots, including colors, markers, line styles, legends, and grid options.

**Toolboxes**

MATLAB offers numerous toolboxes that enhance its functionality for specific applications. Some popular toolboxes include:

* Signal Processing Toolbox:

Provides algorithms and functions for analyzing, preprocessing, and transforming signals.

* Image Processing Toolbox:

Contains tools for image analysis, enhancement, and visualization.

* Statistics and Machine Learning Toolbox:

Includes functions for statistical analysis, machine learning algorithms, and predictive modeling.

* Control System Toolbox:

Offers functions for designing and analyzing control systems, including Bode plots, root locus, and state-space analysis.

* Optimization Toolbox:

Provides functions for solving optimization problems, including linear programming and nonlinear optimization.

**Simulink**

Simulink is an integral part of MATLAB, offering a graphical environment for modeling and simulating dynamic systems. Key features include:

* Block Diagrams:

Users can create block diagrams representing systems and subsystems, making it easier to visualize complex interactions.

Simulation:

Simulink allows users to simulate the behavior of systems over time, enabling analysis of performance under various conditions.

* Model-Based Design:

Facilitates model-based design processes, allowing for rapid prototyping and testing of algorithms before implementation.

* Integration with MATLAB:

Users can integrate MATLAB code and Simulink models, leveraging the strengths of both environments.

**STEPS TO EXECUTE THE PROGRAM :**

**Step 1**: Load the Medical Image

Load the medical image you want to denoise. You can use a library like OpenCV in Python or imread in MATLAB.

**Step 2**: Add Noise

Add artificial noise to the loaded image to simulate a noisy medical image. You can use functions like np.random.normal in Python or randn in MATLAB to generate random noise.

**Step 3:** Define FIR Kernel

If you're using FIR filtering, define a suitable FIR kernel. This kernel will be convolved with the noisy image to perform denoising. Common types of FIR kernels include Gaussian, box, or custom-designed filters.

**Step 4**: Define IIR Filter Coefficients

If you're using IIR filtering, design the filter using appropriate techniques. Common IIR filters include Butterworth, Chebyshev, etc. You'll need to define filter coefficients b and a.

**Step 5**: Apply FIR Filter

Convolve the noisy image with the FIR kernel. This is done using a convolution operation, which can be implemented using functions like scipy.signal.convolve2d in Python or imfilter in MATLAB.

**Step 6**: Apply IIR Filter

Apply the IIR filter to the noisy image using a function like lfilter in Python or filter in MATLAB.

**Step 7:** Display the Results

Display the original image, noisy image, and denoised images side by side for comparison.You can use functions like imshow in MATLAB or libraries like Matplotlib in Python.

**Step 8**: Experiment and Fine-tune

Experiment with different FIR kernels, IIR filter coefficients, and noise levels to find the best denoising results for your specific medical images.

**Step 9**: Save the Denoised Image

Once you're satisfied with the denoised image, save it for further analysis or clinical use.

**CODE :-**

% Define the URL of the image.

image\_url = 'https://imgs.search.brave.com/iSKBDdnwz-zji-aN\_eA7DETQrNGJXWgSH9ZZL-IvAFk/rs:fit:500:0:0:0/g:ce/aHR0cHM6Ly9tZWRp/YS5pc3RvY2twaG90/by5jb20vaWQvMTE3/MzE1OTc0MC9waG90/by9jdC1zY2FuLW9m/LXRob3JheC1hbmQt/YWJkb21lbi5qcGc\_/cz02MTJ4NjEyJnc9/MCZrPTIwJmM9MmNW/d1RQdmNSam9QZERE/WnBXb1NOaFZPSkts/RlFMVlVjQm1mbnR6/Y2QzMD0';

% Define a local path to save the downloaded image.

local\_image\_path = 'C:\Users\91880\OneDrive\Documents\SEM-5\DSP.jpeg';

% Download the image from the URL and save it locally.

websave(local\_image\_path, image\_url);

% Read the image from the local file.

original\_image = imread(local\_image\_path);

% Convert to grayscale if the image is in color.

if size(original\_image, 3) == 3

original\_image = rgb2gray(original\_image);

end

% Add Gaussian noise to the original image.

noisy\_image = imnoise(original\_image, 'gaussian', 0.1);

% Define FIR filter (e.g., Gaussian filter).

sigma = 1;

% Standard deviation for Gaussian filter.

filter\_size = 2 \* ceil(3 \* sigma) + 1;

% Choose an appropriate size.

filter\_fir = fspecial('gaussian', filter\_size, sigma);

% Apply FIR filter.

denoised\_image\_fir = imfilter(noisy\_image, filter\_fir);

% Define IIR filter (Butterworth filter).

n = 2;

% Order of the Butterworth filter.

D0 = 0.1 \* min(size(original\_image));

% Normalized cutoff frequency.

% Create Butterworth filter.

[U, V] = dftuv(size(original\_image, 1), size(original\_image, 2));

D = sqrt(U.^2 + V.^2);

butterworth\_filter = 1 ./ (1 + (D ./ D0).^(2 \* n));

% Apply IIR filter in the frequency domain.

noisy\_image\_fft = fft2(double(noisy\_image));

denoised\_image\_iir\_fft = noisy\_image\_fft .\* butterworth\_filter;

denoised\_image\_iir = real(ifft2(denoised\_image\_iir\_fft));

denoised\_image\_iir = uint8(denoised\_image\_iir);

% Display the images.

figure;

subplot(3, 1, 1), imshow(original\_image), title('Original Image');

subplot(3, 1, 2), imshow(noisy\_image), title('Noisy Image');

subplot(3, 1, 3), imshow(denoised\_image\_fir), title('Denoised using FIR');

figure;

subplot(3, 1, 1), imshow(original\_image), title('Original Image');

subplot(3, 1, 2), imshow(noisy\_image), title('Noisy Image');

subplot(3, 1, 3), imshow(denoised\_image\_iir), title('Denoised using IIR');

function [U, V] = dftuv(M, N)

% Computes meshgrid frequency matrices.

u = 0:(M - 1);

v = 0:(N - 1);

idx = find(u > M / 2);

u(idx) = u(idx) - M;

idy = find(v > N / 2);

v(idy) = v(idy) - N;

[V, U] = meshgrid(v, u);

end

**RESULTS:-**

**X-RAY :**

X-rays are a form of electromagnetic radiation, similar to visible light but with much higher energy. They are widely used in medicine, industry, and research due to their ability to penetrate various materials, including human tissues.

**Medical Applications**

* Diagnostic Imaging:

X-rays are commonly used for imaging the skeletal system (e.g., fractures, infections) and soft tissues (e.g., tumors, infections).

* Computed Tomography (CT):

CT scans use X-rays to create cross-sectional images of the body, providing more detailed information than standard X-ray images.

* Fluoroscopy:

This technique uses continuous X-ray imaging to visualize movement within the body, such as the digestive tract or blood flow.

* Radiation Therapy:

High-energy X-rays are used in cancer treatment to target and destroy malignant cells.

Industrial Applications

* Non-Destructive Testing (NDT):

X-rays are used to inspect materials and structures for flaws without causing damage. This is common in aerospace, manufacturing, and construction.

* Quality Control:

X-ray imaging is employed in quality control processes to ensure the integrity of components and assemblies.

* Security Screening:

X-ray machines are used in airports and other security settings to scan luggage and cargo for prohibited items.

**Safety and Risks**

* Radiation Exposure:

While X-rays are valuable for diagnostics and treatment, exposure carries a risk of radiation-induced damage. Safety measures include minimizing exposure time, increasing distance, and using protective shielding.

* Regulatory Standards:

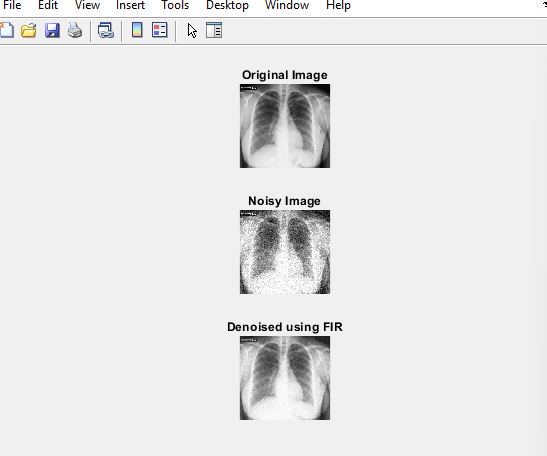
Regulatory bodies, such as the FDA and OSHA, set guidelines for safe use, including equipment maintenance, personnel training, and patient protection protocols.

* Risk-Benefit Analysis:

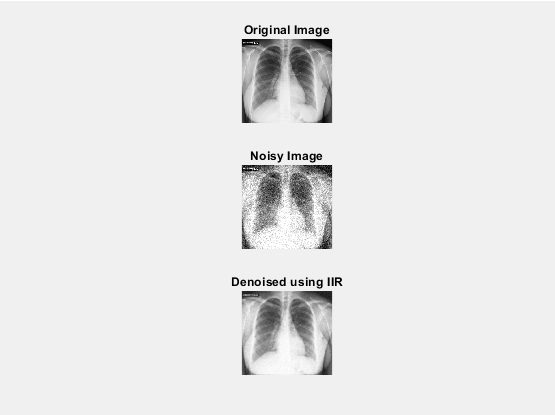
The benefits of X-ray imaging must be weighed against the potential risks of radiation exposure, especially in vulnerable populations like children and pregnant women.

X-rays are a crucial tool in medicine and industry, enabling the visualization of internal structures and the detection of abnormalities. While they provide significant benefits, careful consideration of safety and exposure guidelines is essential to minimize risks associated with ionizing radiation. As technology advances, the applications and effectiveness of X-rays continue to evolve, enhancing their role in diagnostics, treatment, and various industrial processes.

**FIR IMAGE :**

****

**IIR IMAGE :**

****

**CONCLUSION:**

We have excuted the program for the implementation of a denoising algorithm for medical images such as **X-RAY using Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) filters in MATLAB**.). FIR filters are straightforward and maintain image details, while IIR filters can provide substantial noise reduction with fewer coefficients. Choosing between them depends on specific application requirements. Proper parameter selection and design are crucial for optimal results. **This work lays the foundation for advancing** **medical image processing, benefiting healthcare and diagnostics.**

**COURSE OUTCOME:**

**CO2 :** Design and realize appropriate linear FIR filters based on frequency domain specifications.

**CO3 :** Design and realize appropriate digital IIR filters through the classical approach of analog filter design.

Hence through this Project Based Learning **CO2 & CO3** verified.