"FIR AND IIR DIGITAL FILTER DESIGN USING LABVIEW"

PROJECT REPORT

Submitted in partial fulfillment of the requirements for the award of

Bachelor of Engineering

in

Electronics and Communication Engineering

Submitted to

Visvesvaraya Technological University

Belagavi, Karnataka, 590 018

Submitted By

Praveen Kadappanavar	**2KE17EC055**
Prashanth Joshi	**2KE17EC052**
Prathik Bannimath	**2KE17EC054**
Sandeep Goudar	**2KE17EC066**

Under the Guidance of

Mr. Santhosh Hosamane

Department of Electronics and Communication Engineering

(NBA Accredited)

K. L. E. SOCIETY'S

K. L. E. INSTITUTE OF TECHNOLOGY,

HUBBALLI-580027

K. L. E. S.

2020-2021

K. L. E. SOCIETY'S

K. L. E. INSTITUTE OF TECHNOLOGY,

Hubballi-580027

K. L. E. S.

Department of Electronics and Communication Engineering

(NBA Accredited)

2020-2021

CERTIFICATE

Certified that the project work entitled **"**FIR AND IIR DIGITAL FILTER DESIGN USING LABVIEW**"** is a bonafide work carried out by **Praveen Kadappanavar (USN 2K17EC055), Prashanth Joshi (USN 2KE17EC052), Prathik Bannimath (USN 2KE17EC054), and Sandeep Goudar (USN 2KE17EC066),** in partial fulfilment for the award of degree of **Bachelor of Engineering** in **VIII Semester, Electronics and Communication Engineering of Visvesvaraya Technological University, Belagavi,** during the year **2020-21**. It is certified that all corrections/suggestions indicated for internal assessment have been incorporated in the report deposited in the department library. The project report has been approved as it satisfies the academic requirements in respect of project work prescribed for the said degree.

(Mr. Santhosh Hosamane)	**(Dr. Manu T. M.)**	**(Dr. B S. Anami)**
Guide	**H.O.D**	**Principal**

Name of the Examiners **Signature with Date**

1.

2.

DECLARATION

We, Praveen Kadappanavar (USN 2KE17EC055), Prashanth Joshi (USN 2K17EC052) Prathik Bannimath (USN 2K17EC054) Sandeep Goudar (USN 2K17EC066), students of VIII Semester B.E., K.L.E. Institute of Technology, Hubballi, hereby declare that the project work has been carried out by us and submitted in partial fulfillment of the requirements for the VIII Semester degree of **Bachelor of Engineering in Electronics and Communication Engineering** of Visvesvaraya Technological University, Belagavi during academic year 2020-2021.

Date:

Place: Hubballi

Praveen Kadappanavar

Prashanth Joshi

Prathik Bannimath

Sandeep Goudar

CERTIFICATE FROM THE GUIDE

This is to certify that the project work entitled **“**FIR AND IIR DIGITAL FILTER DESIGN USING LABVIEW**”,** is carried out by students Praveen Kadappanavar (2KE17EC055), Prashanth Joshi (2KE17EC052), Prathik Bannimath (2KE17EC054), and Sandeep Goudar (2KE17EC066) under my supervision, for the award of the degree of Bachelor of Engineering in “**Electronics & Communication”** of VTU, Belagavi, Karnataka.

To the best of my/our knowledge and belief,

(i) The work is carried out by the candidates only.

(ii) The work is not taken in its original form, from any other thesis or project reports.

(iii) The work is up to the desired standard both in the technical content and the writeup.

(iv) Plagiarism percentage is 34 %.

Date:

Mr. Santhosh Hosamane
Dept. of ECE, KLEIT,
Hubballi.

The project work as mentioned above is hereby recommended and forwarded for examination and evaluation.

Dr. Manu T. M.
HOD, Dept. of ECE, KLEIT,
Hubballi

Seal of the Dept

ACKNOWLEDGEMENT

The Project report on "FIR AND IIR DIGITAL FILTER DESIGN USING LABVIEW" is the outcome of guidance, moral support and devotion bestowed on us throughout our work. For this we acknowledge and express our profound sense of gratitude and thanks to our guide **Mr. Santhosh Hosamane**. We acknowledge that we were encouraged to take up work using LabVIEW software, which has benefited us by learning this software tool. We also acknowledge that we have learnt the importance of technical research and good presentation skills.

We feel deeply indebted to our **H.O.D. Dr. Manu T. M.** for the constant support and encouragement in all our endeavors. The academic ambience has helped us to excel in our academics.

We take this opportunity to offer our sincere thanks to our beloved **Principal Dr. B. S. Anami** who is always keen on enhancing the quality of teaching- learning and always a source of inspiration to us. We would like to thank our **Dean Academics Dr. S. G. Joshi** for his constant encouragement for our academic progress.

We take this opportunity to express our sincere thanks to **Project Coordinator Dr. R. L. Itagi** for dedicating time to arrange for critical evaluations that helped for proper flow of project work.

Last but not the least, we would like to thank our parents and friends for their well wishes for our success.

Praveen Kadappanavar
Prashanth Joshi
Prathik Bannimath
Sandeep Goudar

ABSTRACT

Digital signal processing (DSP) is the process of analyzing and modifying a signal to optimize or improve its efficiency or performance. In DSP, a digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. Image processing is a method to perform operations on an image, to get an enhanced image or to extract some useful information from it. Adaptive filters are digital filters whose coefficients change with an objective to make the filter converge to an optimal state

This project report presents different designs of digital filters, image processing techniques and adaptive filters using LabVIEW software developed by national instruments. The project work also includes study of noise removal techniques and experimental verification in LabVIEW.

LabVIEW® software tool is available in the department of Electronics & Communication Engineering, KLEIT, Hubballi, procured with the MODROB AICTE Grants, received in Dec/Jan 2020. The design method involves virtual instruments which are built in functions of LabVIEW software. The VIs are connected according to the required design such that they can take inputs and interact with each other to give the required output. The test scenarios include giving different inputs for filter parameters and images for the image processing.

The results are presented in the form of frequency responses of different filters designed in case of digital filters and images in case of noise removal image processing techniques. These results can be observed in the front panel of LabVIEW software. Students can use this report as a beginning guide to start their work in LabVIEW. The report helps the readers to know and install LabVIEW and how to set up the LabVIEW.

CONTENTS

List of Figures

Chapter 1

INTRODUCTION

Digital signal processing (DSP) is the process of analyzing and modifying a signal to optimize or improve its efficiency or performance. DSP involves applying various mathematical and computational algorithms to analog and digital signals to produce a signal that is of higher quality than the original signal. In Digital Signal Processing, noise is a general term for unwanted modifications that a signal may suffer during capture, storage, transmission, processing, or conversion. This modification may be intentional or may be due to interference.

Image processing is a method to perform some operations on an image, to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

An adaptive filter is a system with a linear filter that has a transfer function controlled by variable parameters and a means to adjust those parameters according to an optimization algorithm. Because of the complexity of the optimization algorithms, almost all adaptive filters are digital filters. Adaptive filters are required for some applications because some parameters of the desired processing operation (for instance, the locations of reflective surfaces in a reverberant space) are not known in advance or are changing.

In this chapter, brief introduction about the various topics on which the project is carried out. This chapter helps to get an overview and understanding of Digital Filters, Adaptive Filters and Different Image Processing techniques on which the project is based on.

1.1 Overview of digital filters

Digital filtering is one of the most powerful tools of DSP. Apart from the obvious advantages of virtually eliminating errors in the filter associated with passive component fluctuations over time and temperature, op amp drift (active filters), etc., digital filters are capable of performance specifications that would, at best, be extremely difficult, if not impossible, to achieve with an analog implementation.

The actual procedure for designing digital filters has the same fundamental elements as that for analog filters. First, the desired filter responses are characterized, and the filter parameters are then calculated. Characteristics such as amplitude and phase response are derived in the same way. The key difference between analog and digital filters is that instead of calculating resistor, capacitor, and inductor values for an analog filter, coefficient values are calculated for a digital filter. So, for the digital filter, numbers replace the physical resistor and capacitor components of the analog filter. These numbers reside in a memory as filter coefficients and are used with the sampled data values from the ADC to perform the filter calculations.

The real-time digital filter, because it is a discrete time function, works with digitized data as opposed to a continuous waveform, and a new data point is acquired each sampling period. Digital filters, however, are not the answer to all signal processing filtering requirements. To maintain real-time operation, the DSP processor must be able to execute all the steps in the filter routine within one sampling clock period, 1/fs [1].

1.2 Overview of Image processing

Image processing is a method to perform some operations on an image, to get an enhanced image or to extract some useful information from it. It is a type of signal processing in which input is an image and output may be image or characteristics/features associated with that image. Nowadays, image processing is among rapidly growing technologies. It forms core research area within engineering and computer science disciplines too.

Image processing basically includes the following three steps:

- Importing the image via image acquisition tools.
- Analyzing and manipulating the image.
- Output in which result can be altered image or report that is based on image analysis.

There are two types of methods used for image processing namely, analog and digital image processing. Analogue image processing can be used for the hard copies like printouts and photographs. Image analysts use various fundamentals of interpretation while using these visual techniques. Digital image processing techniques help in manipulation of the digital images by using computers. The three general phases that all types of data must undergo while using digital technique are pre-processing, enhancement, and display, information extraction [5].

1.3 Overview of adaptive filters

Adaptive filters are digital filters whose coefficients change with an objective to make the filter converge to an optimal state. The optimization criterion is a cost function, which is most commonly the mean square of the error signal between the output of the adaptive filter and the desired signal. As the filter adapts its coefficients, the mean square error (MSE) converges to its minimal value. At this state, the filter is adapted, and the coefficients have converged to a solution.

The filter output, y(k), is then said to match very closely to the desired signal, d(k). When you change the input data characteristics, sometimes called filter environment, the filter adapts to the new environment by generating a new set of coefficients for the new data [8].

General Adaptive Filter Algorithm

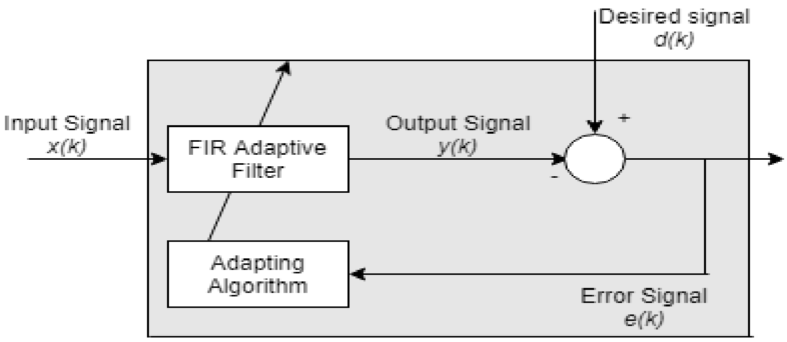


Fig. 1.1 General adaptive filter algorithm

1.4 Literature Survey

The concept of Digital Filtering, Image Processing and its implementation has been referred from IEEE papers, other journal papers and Digital signal processing textbooks. An effort is made to explain these concepts in the literature survey in this section.

Ranjushree Pal (2017)[1], in "Comparison of the design of FIR and IIR filters for a given specification and removal of phase distortion from IIR filters" presented the design of both FIR and IIR Filters. The two designs were compared with respect to circuit complexity, group delay, phase distortion and the output response. An equalizer was also designed to remove the phase distortion of the IIR filter. The results are summarized in the concluding section.

Arnd Frech, Markus Klugel and Peter Russer (2013)[8] in "Adaptive filtering for noise cancellation and signal analysis in real-time" have explained advanced digital signal algorithms for adaptive filtering applied for noise cancellation and signal analysis in real-time. For optimum noise suppression performance, different approaches of adaptive filter algorithms were investigated and enhanced with respect to implementation of the noise cancelling algorithm on field programmable gate arrays (FPGA) and enabling signal processing in real-time.

Eduardo Perez and Sam Shearman (2009)[2] in "LabVIEW DSP - A Hands-on DSP Educational Platform" have explained LabVIEW as an educational DSP platform that allows to learn DSP and advanced concepts by interfacing hardware to live signals, leverage graphical data flow (block diagram) programming to develop and encapsulate signal processing concepts. Successful academic case studies of LabVIEW as a DSP platform for real-time FPGA audio processing and multimedia communications are presented.

Divya Sharma and Rashpinder Kaur (2015)[3] in "Design & Analysis of IIR Notch Filter using LabVIEW" has presented on implementation of IIR filter using signal processing tool kit of LabVIEW software. User interface is designed using LabVIEW to obtain the simulation results of digital IIR filtering technique.

Anusha Nellutla (2018)[4] in "Image Processing Techniques Using LabVIEW" has published her work on LabVIEW which explains Image processing as a method to perform operations on an image, to get enhanced image. The paper Implements image processing using LabVIEW NI vision module. using vision module image acquisition, color transformation, edge detection and morphological operation are performed

Sambit Satpathy, Mohan Chandra Pradhan and Subrat Sharma (2015)[5] in "Comparative Study of Noise Removal Algorithms for Denoising Medical Image Using LabVIEW" Published a paper that presents the study of five types of filters like Gaussian filter, Gabor filter, Box filter, Median filter, Adaptive median filter, which are design using LabVIEW. These filters are used for removing of two types of noises like Gaussian noise, Salt & Pepper noise.

1.5 Motivation and Problem Statement

- **Motivation**

From the literature survey, it is observed that it is possible to design Digital Filters in LabVIEW and understand the work and design flow graphically instead of coding. Like these works it is possible to carryout Image processing and Adaptive filters design in LabVIEW.

- **Problem Statement**

To design digital filters, adaptive filters and noise reduction techniques for Images using LabVIEW software.

1.6 Objectives fulfilled

Objectives of the proposed project work which are fulfilled are mentioned as below-

i. FIR & IIR digital filters using LabVIEW are designed and verified.
ii. Adaptive filters are designed in LabVIEW.
iii. Filtering techniques are applied for noise removal in image.

1.7 Scope and Limitations

The Digital filters has some of the limitations which are described below:

- **Uneconomical for Simple applications:** Even for simple applications using DSP techniques and processors, there is a need to use ADC and DAC since the signals taken as input may also be analog in nature and outputs may also be either required to be in digital or analog as per requirements.
- **Limited to Low Bandwidth Signals:** The DSP techniques are most of the times limited to low bandwidth signals and it becomes difficult to apply them to high bandwidth signals, this becomes time consuming for simple applications.

1.8 Relevance and Type

The Relevance of project is **General**, which involves using of LabVIEW tool for the design of filters, the tool makes use of graphical representation of required instruments thereby removing the process of coding and making it easy for the users to implement new techniques.

Type

The project work is of **Research** type, which requires collecting available codes and design of digital filters in LabVIEW.

1.9 Organization of the Report

The Project Report is organized as mentioned below

Chapter 1: Chapter 1 gives brief introduction to Digital Filters, Image Processing and Adaptive Filters. Literature survey, Objectives and scope and limitations are also discussed.

Chapter 2: Chapter 2 presents the methodology of Designing of digital Filters, Image processing Techniques and Adaptive filters using LabVIEW.

Chapter 3: Chapter 3 presents with the results obtained from LabVIEW for the designed experiments. Results appear in the front panel of the LabVIEW software.

Chapter 4: Chapter 4 concludes the project report with conclusion and future scope of the project.

In the end, a reasonable bibliography is given for the readers help.

In the appendix LabVIEW software setup and usage is mentioned.

- **Summary**

  In this chapter we have explained overview about the project mentioning some of the literature surveys and explained in brief about the motivation and problem definition, objectives, scope and limitations.

Chapter 2

METHODOLOGY

In chapter 1, it is mentioned that LabVIEW provides a platform and tools to design and verify digital filters. In this chapter, LabVIEW's blocks called as VIs (Virtual Instruments) are explained. Design concepts of digital filters are explained.

The overall process involved in "FIR AND IIR DIGITAL FILTER DESIGN" is to study the LabVIEW blocks, placing of required blocks for the digital filter design from the function's palette, selection of input and output blocks from the function palette, control palette and are connected according to the block diagram. The outputs obtained are observed in Front panel. Noise removal techniques in image are designed and implemented using LabVIEW addons such as VAS and VDM. Also, the design and implementation of adaptive filters.

In this project work, two different platforms are used: MATLAB and LabVIEW.

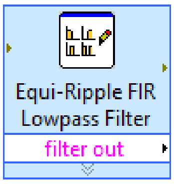
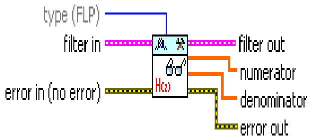
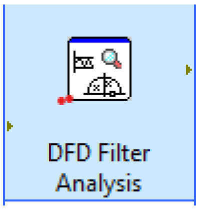
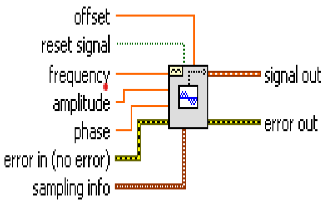
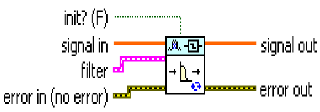
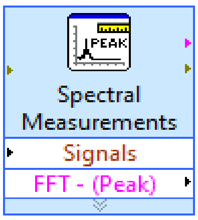
- MATLAB: With MATLAB it is possible to Prototype, test, and implement DSP algorithms on PCs, embedded processors, SoCs, and FPGAs. MATLAB can be used to Acquire, measure, and analyze signals from many sources.
- LabVIEW DSP module tool: Laboratory Virtual Instrument Engineering Workbench (LabVIEW) is a system-design platform and development environment for virtual programming language from National Instruments.

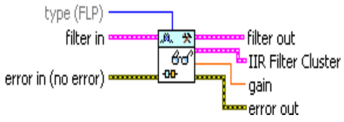
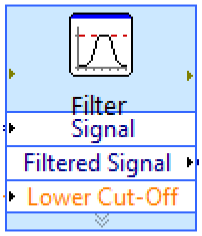
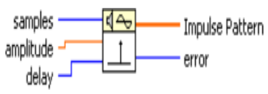
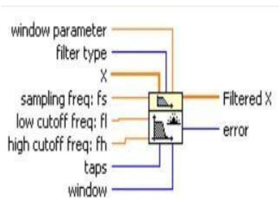
2.1 Digital filter design

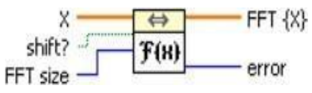
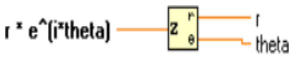
In DSP, a digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to reduce or enhance certain aspects of that signal. The General Purpose of Digital Filters are separation of signals that have been combined and restoration of signals that have been distorted in some way.

Digital filters can be implemented in two ways, by convolution (also called Finite Impulse Response or FIR) and by recursion (also called Infinite Impulse Response or IIR). Filters carried out by convolution (FIR) can have far better performance than filters using recursion (IIR) but execute much more slowly [2].

Table 2.1 VI's Used in designing digital filter

VI'S	Name of VI	Description
	Classical Filter Design Express VI	Specifies the type of filter that this VI creates.
	DFD Get Transfer Function VI	specifies the type of transfer function to retrieve.
	Digital Filter Analysis Express VI	This Analyses the specified characteristics of a filter.
	Sine Waveform VI	This type Generates a waveform containing a sine wave.
	DFD Filtering Array	This Filters an input signal continuously.
	Spectral Measurements	This Performs FFT-based spectral Measurements.

	DFD Get Cascaded Coefficient VI	It Converts a filter to an infinite impulse response (IIR) filter cluster.
	Simulate Signal Express VI	It Simulates a sinewave, square wave, triangle wave, sawtooth wave, or noise signal.
	Filter Express VI	It Processes signals through filters and windows.
	Impulse Pattern	It Generates an array containing an impulse pattern.
	FIR Windowed Filter	It Filters the input data sequence, **‘X’,** using the set of windowed FIR filter coefficients specified by the sampling frequency: fs**,** low cut-off frequency: fl**,** high cut-off frequency: fh**,** and number of taps**.**

	FFT (Fast Fourier Transform) VI	It Computes FFT of the input sequence X.
	Complex to Polar Function	It Breaks a complex number into its polar components.
	Unwrap Phase	It Unwraps the phase array by eliminating discontinuities.

2.1.1 FIR Filter Design

Finite impulse response (FIR) filters are non-recursive filters. The output depends only on a history of input values.

The Difference equation of an FIR filter is given by,

$$\boldsymbol{y(n) = \sum_{k=0}^{N-1} h(k)x(n-k)} \qquad \textbf{(2.1)}$$

where, $h(k) = h_d(k)w_N(k)$

$h_d(k)$ is the impulse response coefficient of an ideal filter.

$w_N(k)$ is the right window function

The Transfer Function of an FIR filter is,

$$\boldsymbol{H(z) = \sum_{K=0}^{N-1} h(k)Z^{-k}} \qquad \textbf{(2.2)}$$

Windowing Techniques: A window is a finite length sequence. Windowing is a process of forming a finite length sequence from an infinite length sequence by multiplying an infinite length sequence with a window.

Types of Windows: Rectangular Window, Hanning Window, Hamming Window, Blackman Window, Bartlett Window.

➢ FIR Low Pass Filter and High Pass Filter using Rectangular Window

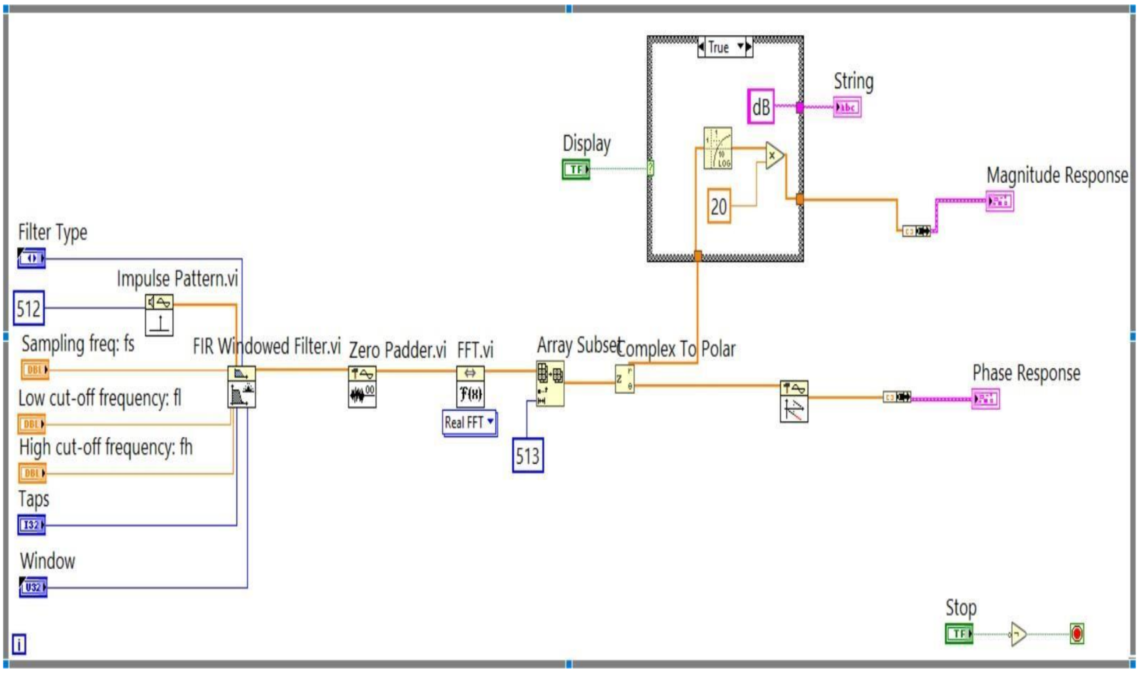


Fig. 2.1 Block Diagram of FIR VI in LabVIEW

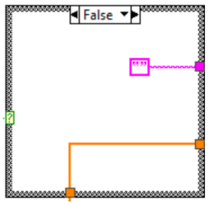


Fig. 2.1 shows all the inputs and specifications such as filter type, Sampling frequency, lower and higher cut-off frequencies, etc. that are given to the FIR Windowed Filter VI block where the filter does its processing on the inputs given. The filtered output is given to the Zero padder block which resizes the input sequence input array to the next higher valid power of 2. The output of this block is given to the FFT block which computes FFT of the sequence. This FFT output is given to the array Subset block which fits different types of responses into a single array. The output obtained from this block is complex, which has both magnitude and phase value. The Complex to polar block is used to split the same into responses. The magnitude output can be obtained in logarithmic form by using a case block with logarithmic function in it. An Unwrap phase block is used in the phase response path which unwraps the phase array by eliminating discontinuities whose absolute values exceed either pi or 180 degrees. Hence the magnitude and phase responses are obtained from the given VI block diagram.

- **FIR Filtering System**

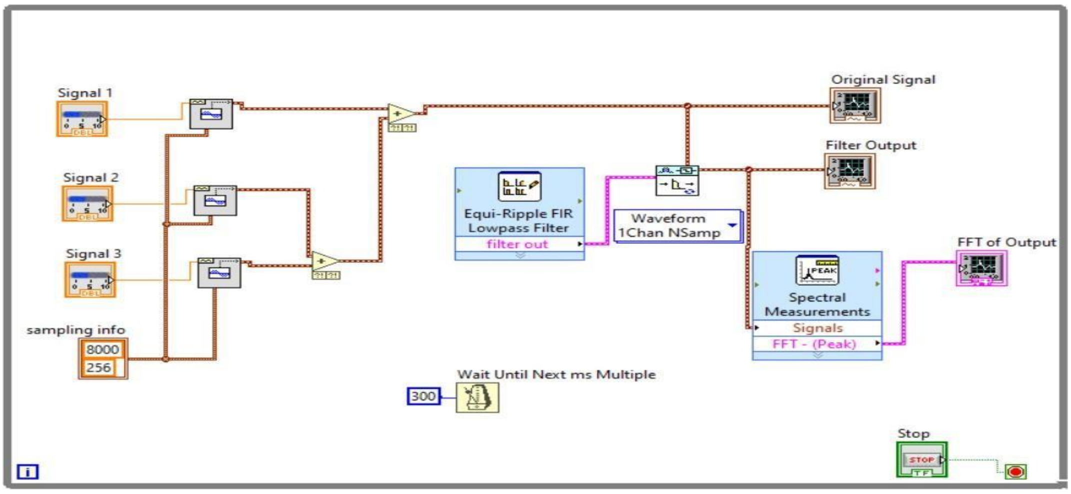


Fig. 2.2 Block Diagram of FIR filtering system VI in LabVIEW

In Fig. 2.2, three sine wave signals are taken as input with a specified sampling frequency. Out of the three, two are added together and the output of this added waves is again added to another sine wave input. This cascaded wave of three sine inputs is taken as the waveform with noise. And this noise is removed in the further process by the Equi -Ripple Lowpass filter. After the filtering of the noise wave from this filter, the original signal, the filter output and the FFT of the filtered output waveforms are obtained using the waveform blocks.

2.1.2 IIR Filter Design

The infinite impulse response (IIR) filter is a recursive filter in that the output from the filter is computed by using the current and previous inputs and previous outputs. The Difference Equation of an IIR filter is given by,

$$y(n)=\sum_{i=1}^{N} a_i\, x(n-i) + \sum_{j=1}^{N} b_j\, y(n-j) \qquad (2.3)$$

The Transfer Function of an IIR filter is,

$$H(z) = \frac{\sum_{i=1}^{N} a_i z^{-1}}{1+\sum_{j=1}^{N} b_j z^{-1}} \qquad (2.4)$$

where, N=1,2,3…… is the order of the filter

a_i, bj are the filter's Coefficients.

➢ **Butterworth IIR Low Pass Filter and High Pass Filter**

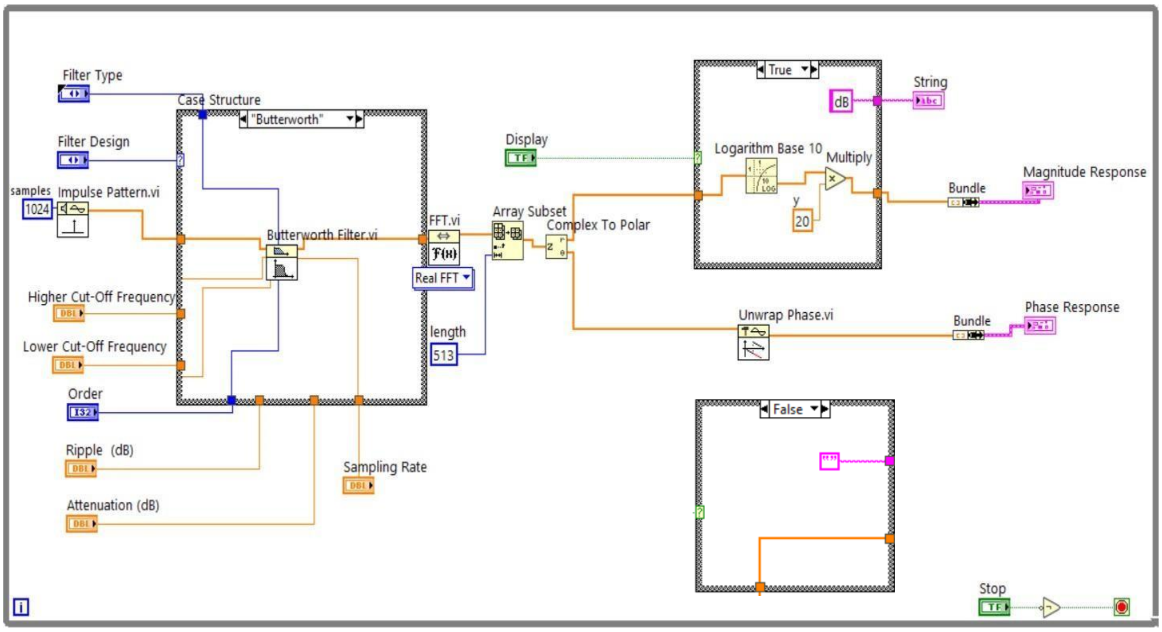


Fig. 2.3 Block Diagram of IIR Filter VI in LabVIEW

Fig. 2.3 shows all the inputs and specifications such as filter type, filter design, lower and higher cut- off frequencies, etc. that are given to the Butterworth filter VI block where the filter does its processing on the inputs given. The filtered output is given to the FFT block which computes FFT of the sequence from the filter. This FFT output is given to the array Subset block which fits different types of responses into a single array. The output obtained from this block is complex, which has both magnitude and phase value. The Complex to polar block is used to split the same into responses. The magnitude output can be obtained in logarithmic form by using a case block with logarithmic function in it. An Unwrap phase block is used in the phase response path which unwraps the phase array by eliminating discontinuities whose absolute values exceed either pi or 180 degrees. Hence the magnitude and phase responses are obtained from the given VI block diagram.

➢ **IIR Filtering system**

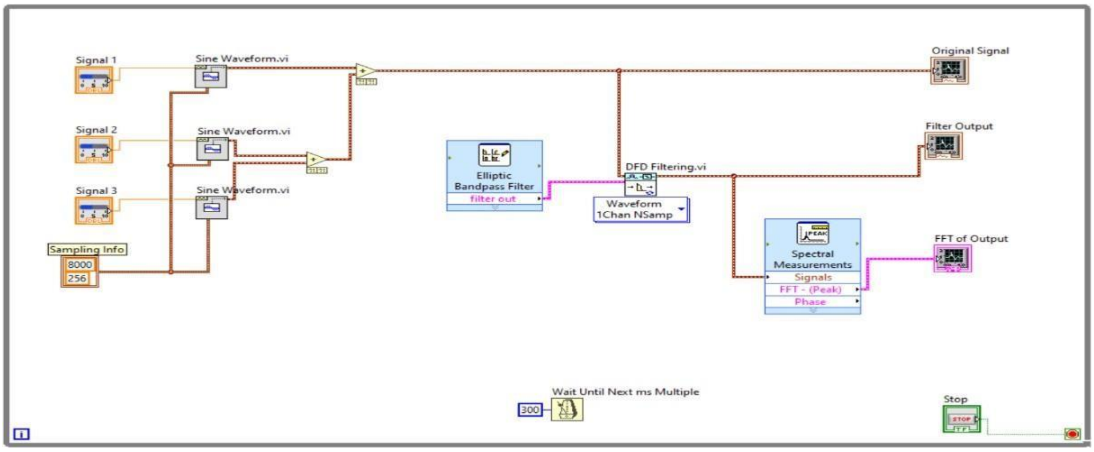


Fig. 2.4 Block Diagram of a IIR Filtering system VI in LabVIEW

In Fig, 2.4, three sine wave signals are taken as input with a specified sampling frequency. Out of the three, two are added together and the output of this added waves is again added to another sine wave input. This cascaded wave of three sine inputs is taken as the waveform with noise. And this noise is removed in the further process by the Elliptic bandpass filter. After the filtering of the noise wave from this filter, the original signal, the filter output and the FFT of the filtered output waveforms are obtained using the waveform blocks.

2.1.3 Noise removal Block

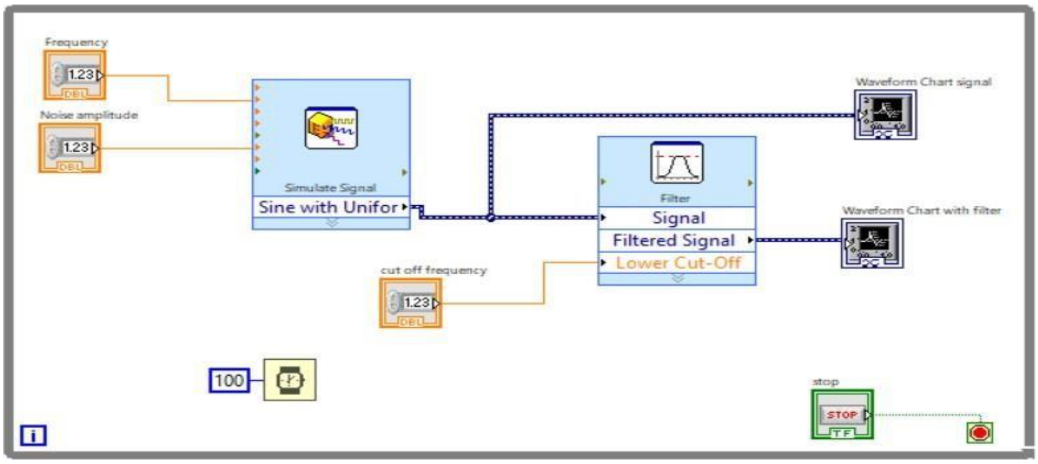


Fig. 2.5 Block Diagram of Noise removal block VI in LabVIEW

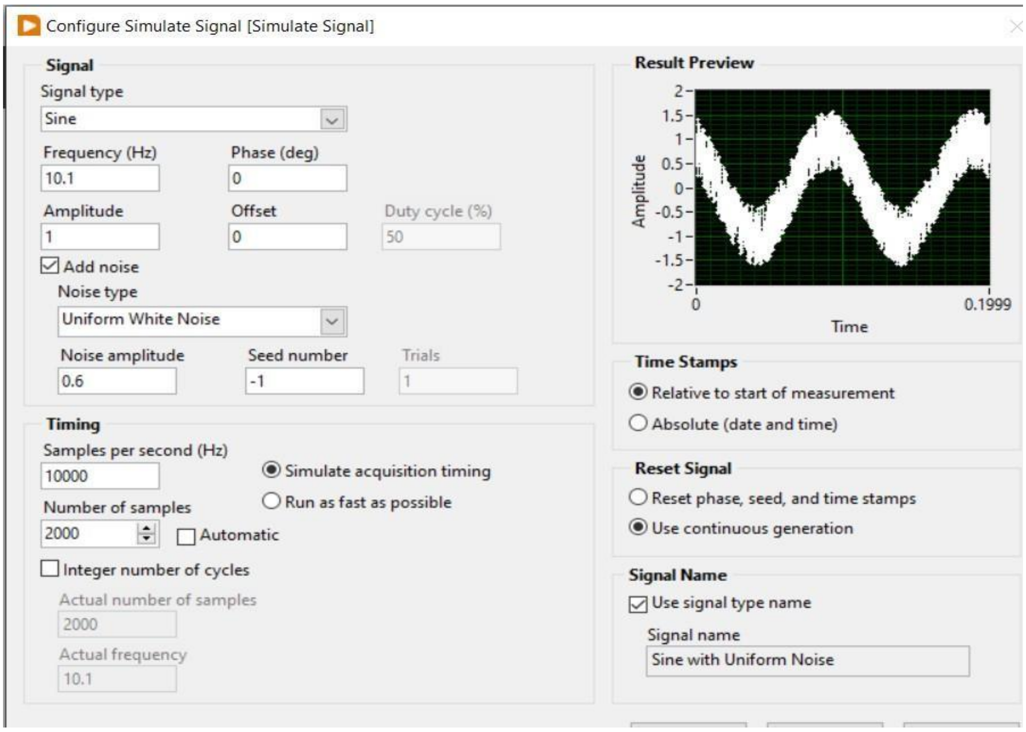


Fig. 2.6 Configuration simulate signal

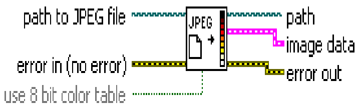
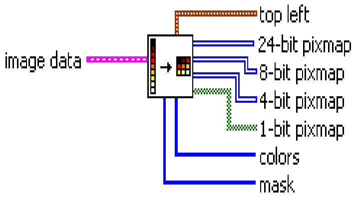
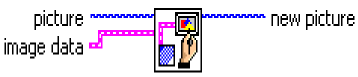
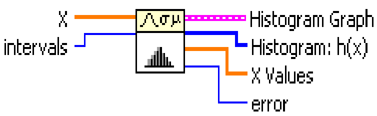
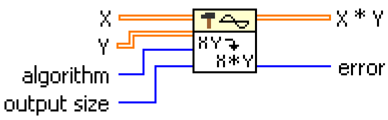
Fig. 2.5 shows the noise removal block which consists of configuration simulate signal VI and configure filter VI. From configuration simulate signal VI, signal and the noise type are selected, and the resulted signal is given to the configure filter VI. This VI consists of digital filters i.e., both FIR and IIR filters. Depending upon the application filter is decided. These filters remove the noise present in the signal and at the output original is obtained.

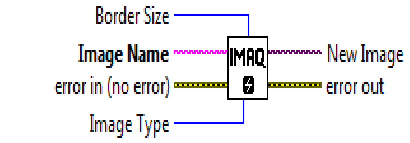
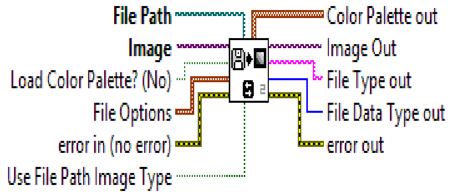
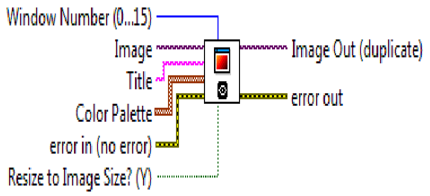
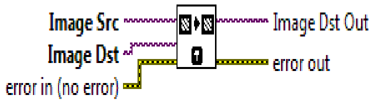
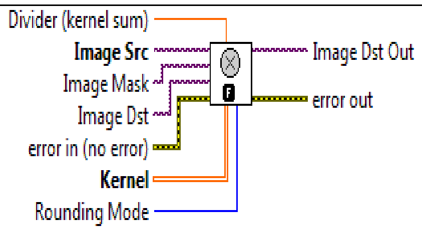
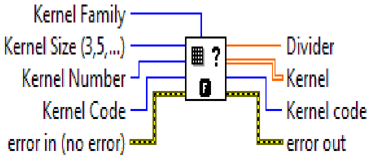
Fig. 2.6 gives the configuration of the signal where we can specify the signal type, sampling frequency, amplitude and noise type. It also shows the signal added with noise. The noise removal technique will then be applied and the results will be shown in chapter 3.

2.2 Image Processing Techniques

LabVIEW facilitates the user with addons which support Image processing VIs which are used in experiments like Edge detection, Erosion and Dilation, Median Filter and Box Filter.

Table 2.2 VI's Used in Image processing

VI'S	Name of VI	Description
	Read JPEG File VI	Reads a JPEG file and creates the data necessary to display the file in a picture control.
	Unflatten Pixmap VI	Converts a cluster of image data into a 2D array.
	Draw Flattened Pixmap VI	Draws a 1-, 4-, or 8-bit pixmap or a 24-bit RGB pixmap into a picture.
	Histogram VI	Finds the discrete histogram of the input sequence **X**.
	2D Convolution	Computes the convolution of the input sequences **X** and **Y**.

	IMAQ Create VI	Creates a temporary memory location for an image.
	IMAQ Read File 2 VI	Reads an image file.
	IMAQ Wind Draw VI	Displays an image in an image window.
	IMAQ Copy VI	Copies the specifications and pixels of one image into another image of the same type.
	IMAQ Convolute VI	Filters an image using a linear filter.
	IMAQ Get Kernel VI	Reads a predefined kernel.

2.2.1 Edge Detection of an Image

Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges.

An edge point is a point in an image with coordinates [i, j] at the location of a significant local intensity change in the image. An edge fragment corresponds to the i and j coordinates of an edge orientation θ, which may be the gradient angle. An edge detector is an algorithm that produces a set of edges from an image [7].

- **Edge Detection of an Image in LabVIEW**

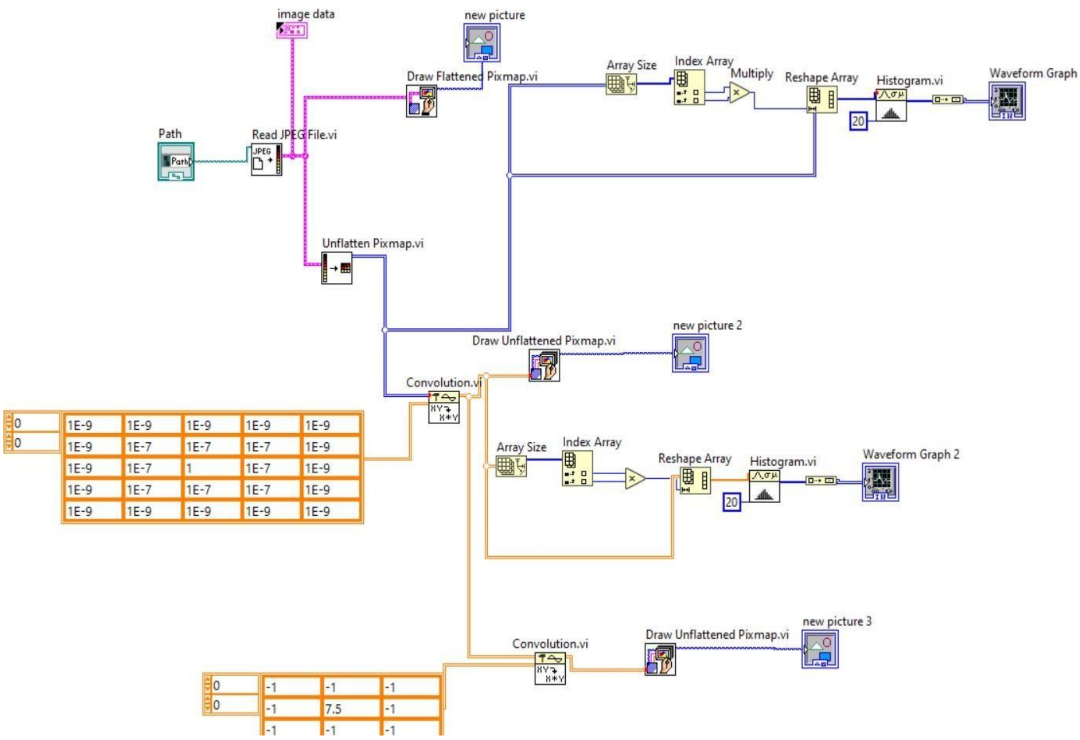


Fig. 2.7 Block Diagram of an Edge Detection VI in LabVIEW

The Edge Detection of an Image can be carried out by using the above VI block created in the LabVIEW block diagram panel. The Working of this block diagram involves 3 stages. First the path of an Image is read and unflattened using the unflatten pixmap VI. The data obtained from this VI is used to obtain the histogram of the image. The same data is again given to a convolution block and convoluted with an Image smoothening kernel. This gives a smoothened Image and again histogram of this Image is calculated which can be observed in the output block in front panel. The Image Smoothening changes can be observed in detail by looking at histogram output. Similarly, the third level includes convoluting the smoothened Image with edge detection kernel then this data obtained in flattened and given to the image window, which displays the Edges of the original Image. These outputs can be observed in the results section of this report. Note that development of this VI in LabVIEW requires addons like IMAQ Vision acquisition Software (VAS) and Vision development Module (VDM).

2.2.2 Morphology Block (Erosion and dilation)

Morphological operations apply a structuring element to an input image, creating an output image of the same size. In a morphological operation, the value of each pixel in the output image is based on a comparison of the corresponding pixel in the input image with its neighbors. Morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to greyscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest. A morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image.

- **Erosion and Dilation**

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion.

- **Rules for Dilation and Erosion**

Dilation and Erosion are the morphological operations used to sharpen or thickening the image.

- **Erosion:** The value of the output pixel is the minimum value of all pixels in the neighborhood. In a binary image, a pixel is set to 0 if any of the neighboring pixels have the value 0. Morphological erosion removes islands and small objects so that only substantive objects remain.
- **Dilation:** The value of the output pixel is the maximum value of all pixels in the neighborhood. In a binary image, a pixel is set to 1 if any of the neighboring pixels have the value 1. Morphological dilation makes objects more visible and fills in small holes in objects.

- **Image Morphology Block in LabVIEW**

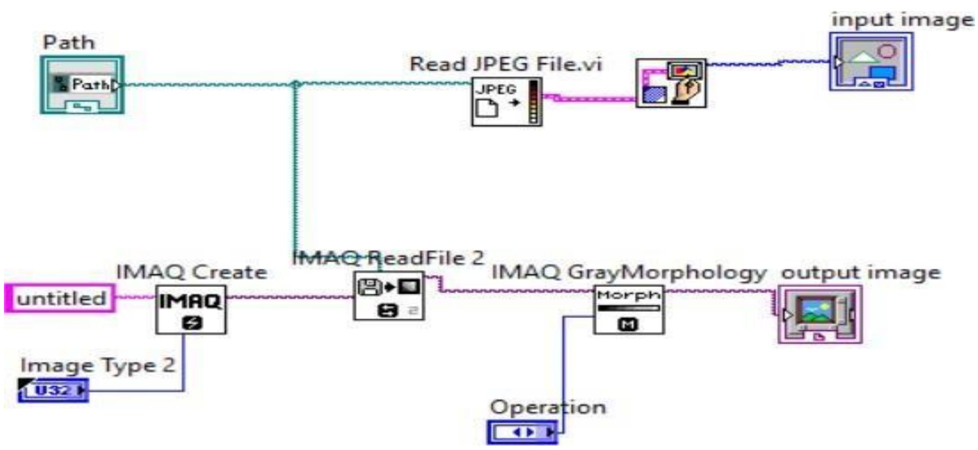


Fig. 2.8 Block Diagram of an Image Morphology block in LabVIEW

The Morphological operation on an Image can be carried out by using the block diagram created in the LabVIEW as shown in Fig. 2.8. The Working of this block diagram involves the path of an Image is read and the image data send to the IMAQ Read file 2 where it reads the image for further process. It also contains a block called IMAQ Create which is used as temporary memory storage, here the image is stored. IMAQ Read file 2 is connected to the IMAQ Gray Morphology where the process of choosing between erosion and dilation can be selected from the front panel and that operation is applied to the 8-bit image read from the path. This gives us the desired output in the Front panel.

2.2.3 Median Filter

The median filter is a non-linear digital filtering technique, often used to remove noise from an image or signal. Such noise reduction is a typical pre-processing step to improve the results of later processing of an image. Median filtering is very widely used in digital image processing because, under certain conditions, it preserves edges while removing noise and it also has applications in signal processing.

The main idea of the median filter is to run through the signal entry by entry, replacing each entry with the median of neighboring entries. The pattern of neighbors is called the "window", which slides, entry by entry, over the entire signal. For one-dimensional signals, the most obvious window is just the first few preceding and following entries, whereas for two-dimensional (or higher-dimensional) data the window must include all entries within a given radius [5].

- **Median Filter in LabVIEW**

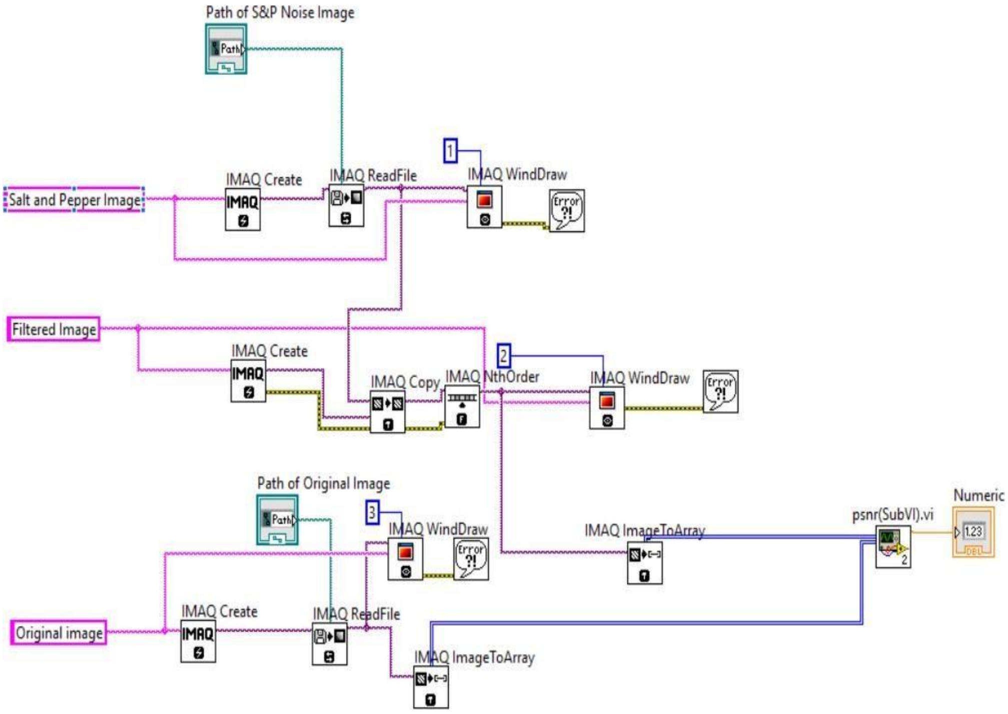


Fig. 2.9 Block Diagram of a Median Filter in LabVIEW

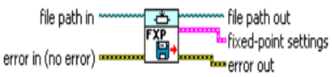
The above block diagram shown is of Median Filter. This Median Filter takes an 8-bit Image consisting of Salt and Pepper noise and applies the Median Filter operation to it to obtain the denoised image.

The block reads an Image of 8-bit from the file path and reads it using IMAQ Read file and a memory is created for that Image using IMAQ Create VI. IMAQ Window draw VI is used to create a new window to display the Image and Error VI checks for any errors in the given input VI. The IMAQ Nth order VI performs the Median Filter operation on the noisy image and gives the filtered image. Original Image as well as Filtered Images are compared to check the PSN ratio between the Images. The output in the Front panel Include 3 Image displaying windows of original image, Noisy Image and Filtered Image.

2.3 Adaptive Filter

An adaptive filter is a computational device that iteratively models the relationship between the input and output signals of a filter. It is a system with a linear filter that has a transfer function controlled by variable parameters and a means to adjust those parameters according to an optimization algorithm.

Table 2.3 VIs used in Adaptive Filter

VI'S	Name of VI	Description
	AFT Simulate Fixed-Point FIR LMS	It Configures the settings of a fixed-point adaptive filter and simulates the performance of the adaptive filter
	AFT Save Fixed-Point FIR LMS	Loads the settings of a fixed-point adaptive filter.

2.3.1 LMS Adaptive Filter

The LMS algorithm is an adaptive algorithm among others which adjusts the coefficients of FIR filters iteratively. Least mean squares (LMS) algorithms are a class of adaptive filter used to mimic a desired filter by finding the filter coefficients that relate to producing the least mean square of the error signal (difference between the desired and the actual signal).

- **LMS Adaptive Filter in LabVIEW**

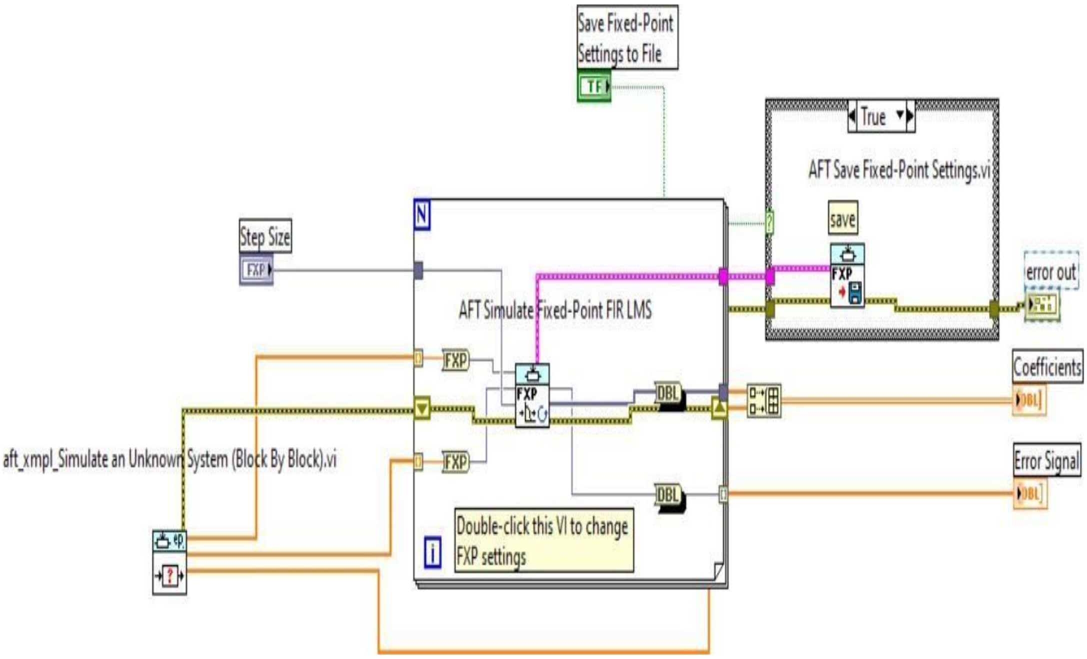


Fig. 2.10 Block diagram of LMS Adaptive Filter in LabVIEW

The above figure is a block diagram of LMS Adaptive Filter. The block diagram is constructed by connecting an AFT Simulate Fixed-Point FIR LMS with a sub VI which consists of all the inputs for the LMS Filter in its front panel. The Inputs are converted into Fixed point data before giving to AFT Simulate VI. The AFT VI performs the operation of Adaptive filter based on LMS algorithm by taking the Fixed-point Inputs. The outputs obtained are converted into double type data and given to the graphs of coefficients and error signal. The settings of the AFT VI can be saved into another VI called AFT Fixed-point Settings VI. The output is observed in the Front Panel.

2.4 Software Details

The following topics give brief Introduction about the LabVIEW software.

2.4.1 Introduction to LabVIEW

LabVIEW (Laboratory Virtual Instrument Engineering Workbench) is a graphical programming language that uses icons instead of lines of text to create applications. In contrast to text-based programming languages that use instructions to determine the order of program execution, LabVIEW uses dataflow programming. In data flow programming, the flow of data through the nodes on the block diagram determines the execution order of the VIs and functions. VIs, or virtual instruments, are LabVIEW programs that imitate physical instruments. In LabVIEW, you build a user interface by using a set of tools and objects. The user interface is known as the front panel. After you build the front panel, you add code using graphical representations of functions to control the front panel objects. You add this graphical code, also known as G code or block diagram code, to the block diagram. The block diagram somewhat resembles a flowchart. The block diagram, front panel, and graphical representations of code compose a VI. LabVIEW is unique for its capabilities to interface with and control different types of hardware modules and instruments from NI (National Instruments). LabVIEW is a software that does not require much of coding proficiency from the user [6].

2.4.2 LabVIEW ADD-ONS

Since LabVIEW is a graphical Programming Language it is difficult to include all the facilities under a single bundle of software. For that reason, NI (national Instruments) have introduced a concept called Add-Ons.

- **DFD (Digital Filter Design) Toolkit**

As the name implies this software is mainly focused on helping users to design and develop Digital Filters of different parameters. The toolkit also supports user with options required to develop in different ways also.

The Main Features of DFD Toolkit Involves:

a) Comprehensive Analysis Tools: You can use the Filter Analysis VIs to evaluate the characteristics of digital filters

b) Large Selection of Filter Structures: When you design digital filters with the Digital Filter Design Toolkit, you can select from one of 23 possible filter structures

c) Special Digital Filter Design: The Special Filter Design VIs help you design IIR notch/peak filters, IIR comb filters, maximally flat filters, narrowband filters, and group delay compensators.

d) Adaptive Filter Design: The Adaptive Filters VIs such as AFT Create FIR LMS which creates a FIR adaptive filter with standard least mean square algorithm.

- **Vision Acquisition Software (VAS)**

The Vision Acquisition Software (VAS) is a set of drivers and utilities used to acquire, display, and save images from a wide range of camera types, including cameras using GigE Vision, IEEE 1394 (FireWire), USB 2.0, USB 3 Vision, or the Camera Link standard.

Components: VAS is made up of the following three drivers.

- NI-IMAQ - Acquisition from National Instruments frame grabbers, as well as general display, file saving and acquisition functions
- NI-IMAQdx - Acquisition from GigE Vision, IEEE-1394 cameras, USB cameras that are Direct Show compliant (VAS 2009 onwards), and some IP cameras.
- NI-IMAQ I/O - Configuration of FPGA included in the PCI-8254R, PCI-8255R, Compact Vision System.

- **Vision Development Module (VDM)**

Description: Vision Development Module (VDM) provides machine vision and image processing functions for LabVIEW, C/C++, Visual Basic, and .NET environments.

Included in: VDM is a separately licensed module installed into LabVIEW.

Components: VDM includes the Vision Assistant tool, a prototyping and code generation tool like NI Vision Builder in its menu-driven interface. Vision Assistant is a useful tool for developing and testing a series of image analysis and processing steps from which code can be generated.

2.4.3 LABVIEW VIRTUAL INSTRUMENTS (VIs)

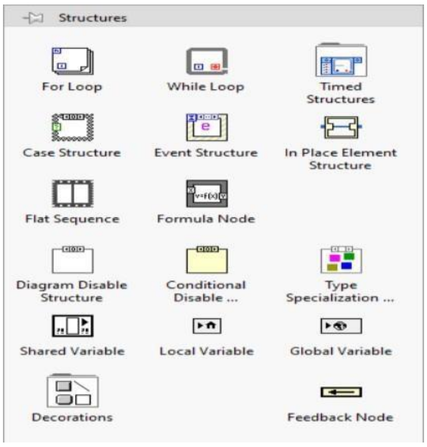
There are many Virtual Instruments (VI's) in the LabVIEW and some of the important ones are mentioned below, these can be accessed by Function's palette and Controls palette by right clicking on Block diagram and Front panel. The Functions chosen on block diagram appear as their respective controls on front panel.

I. Structures

Owning Palette- Programming VIs and Functions.

Requires- Base Development System. It consists of For loop, while loop, case structures.

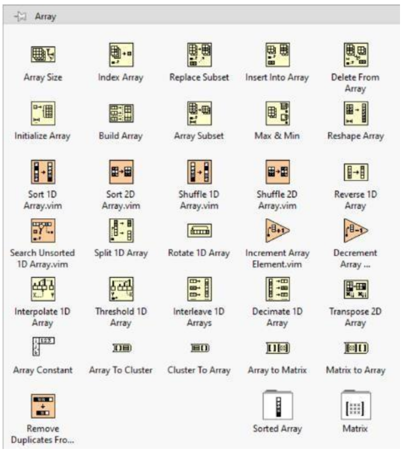
These structures are used to enclose the VIs in a loop or implement different cases for a same basic circuit.



II. Array

Owning Palette- Programming VIs and Functions.

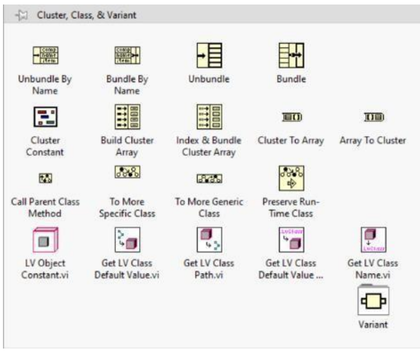
Requires- Base Development System. It consists of different types of arrays as shown in the figure. Used to store multiple dimensions of array.



III. Clusters, class, and variant

Owning Palette- Programming VIs and Functions.

Requires- Base Development System. It consists of VIs used to form a cluster of elements from selection.



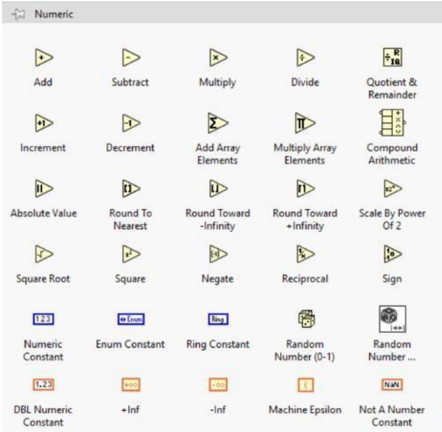
IV. Numeric

Owning Palette-

Programming VIs and Functions.

Requires- Base Development System.

It consists of Numeric type VIs which are used to perform mathematical operations such as addition, multiplication etc.



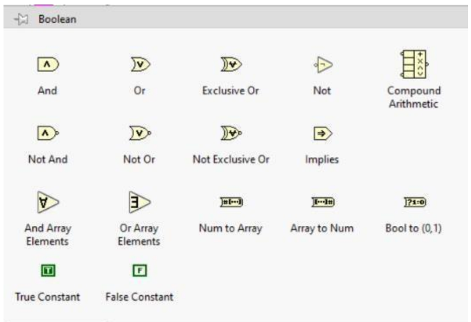
V. Boolean

Owning Palette-

Programming VIs and Functions.

Requires- Base Development System.

It consists of VIs used to perform logical operations like and, not, or, and true or false operations.



VI. Vision and Motion

Owning Palette- Functions.

Requires- Vision Acquisition Software and Vision Development Module.

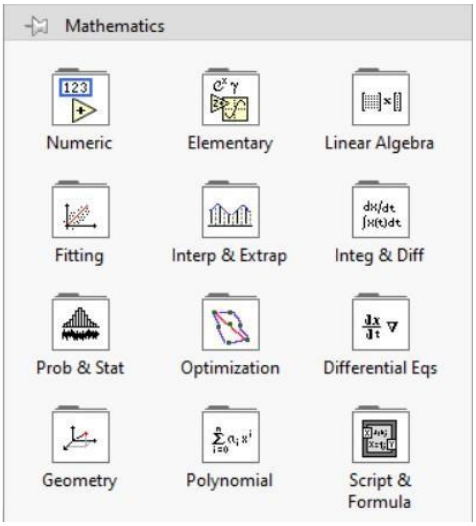
The respective folders contain many other VIs which are mentioned in the upcoming experiments.

VII. Mathematics

Owning Palette- Functions.

Requires- Base Development system.

It consists of mathematical VIs that can be used to perform mathematical operations on data as well as digital signals

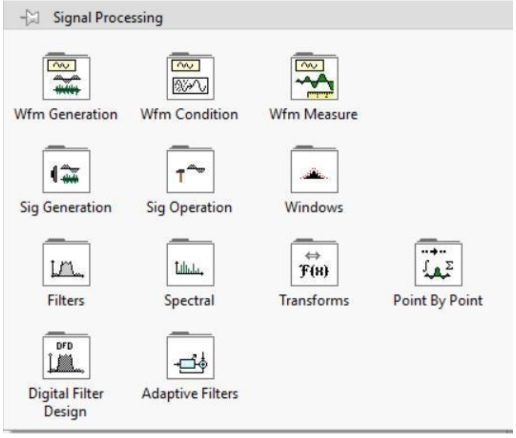


VIII. Signal processing

Owning Palette- Functions.

Requires- DFD Toolkit.

It consists of VIs responsible for signal generation, waveform visualization, signal operations and Filters both digital and analog.



• Summary

The Methodology involved studying the VIs available in the LabVIEW and choosing appropriate VIs for designing the digital filters. The VIs were obtained from the toolkits like Digital Filter Design toolkit (DFD), Vision Acquisition Software (VAS) and Vision Development Module (VDM).

Chapter 3

RESULTS AND DISCUSSION

In chapter 2, LabVIEW's blocks called as VIs are explained along with their functions. Design concepts of Digital filters, Image processing and Adaptive filter are explained. In this project titled "FIR AND IIR DIGITAL FILTER DESIGN USING LABVIEW" presents the designs of Digital filters and Image processing Techniques. The Block diagrams (back-end design) are discussed in the previous sections of methodology. In this chapter, let us analyze the results obtained in the Front panel, where a user can work with set of different test cases.

3.1 Results of Digital Filters

- **FIR LPF and HPF using rectangular window**

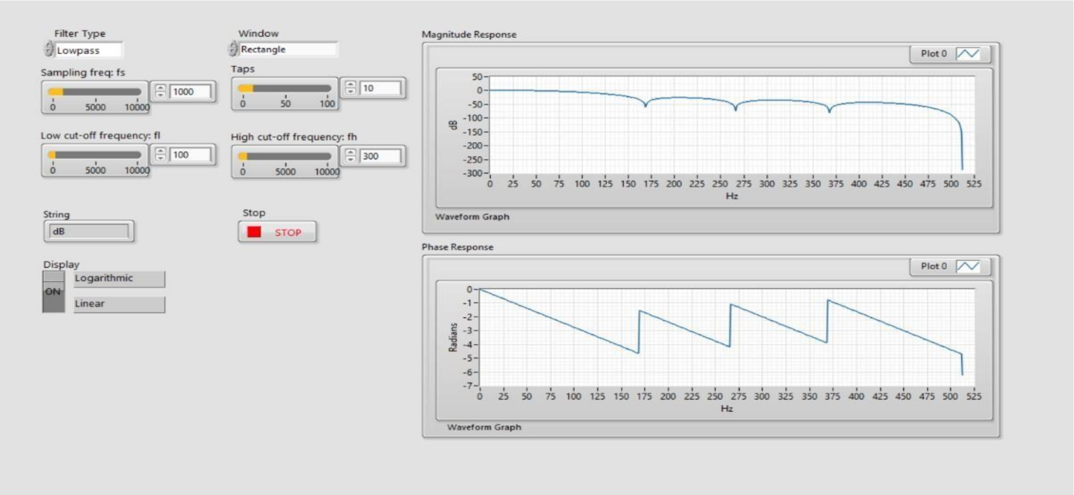


Fig. 3.1 Front panel of FIR LPF Using Rectangular Window

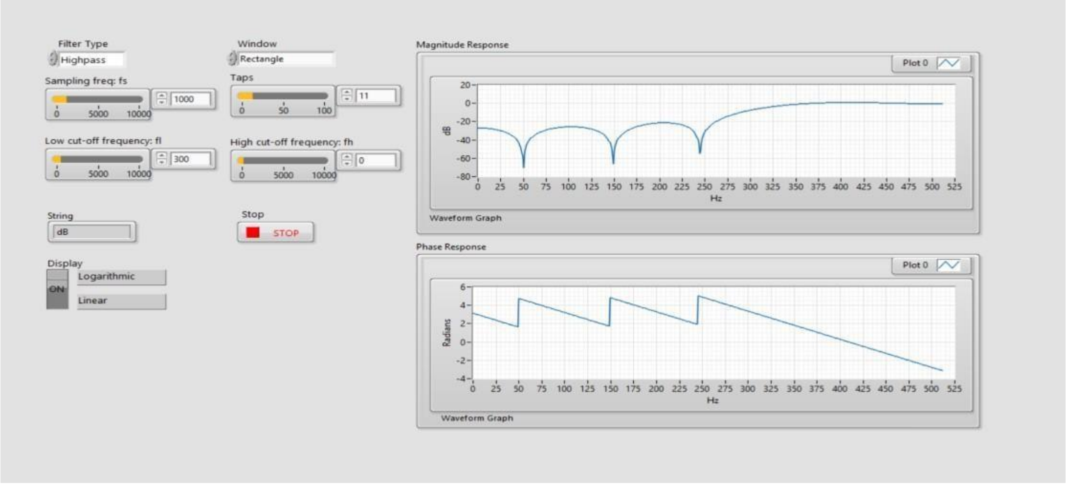


Fig. 3.2 Front panel of FIR HPF Using Rectangular Window

The Fig. 3.1 and 3.2 shows the output of the FIR LPF and HPF using rectangular window. According to the requirements, the types of filters can be changed like Lowpass, High pass, Bandpass and Band stop filters. Also, the type of windowing technique such as rectangular, hanning, hamming windows can be varied accordingly. The Lower cut- off, Higher cut-off and the sampling frequency are specified to get the magnitude and phase response of the given filter.

- **FIR Filtering System**

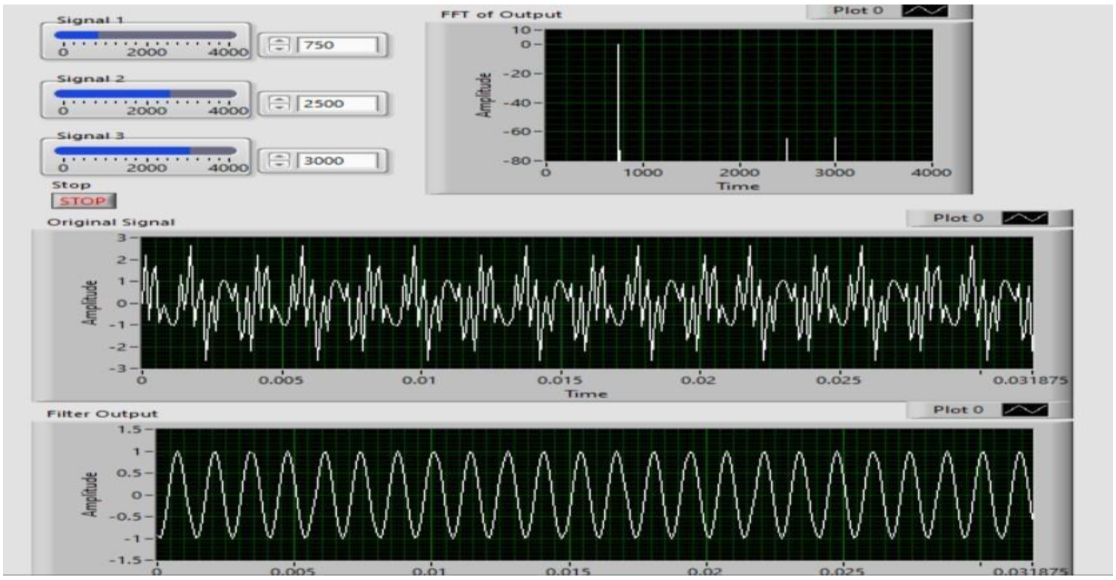


Fig. 3.3 Front panel of FIR Filtering System

The Fig. 3.3 is the front panel of the FIR filtering system output. The original signal is the signal with noise and after filtering, filter output waveform is the signal after removing the noise (In this filter, the higher frequencies are classified as noise and are removed).

➢ Butterworth IIR Low Pass Filter and High Pass Filter

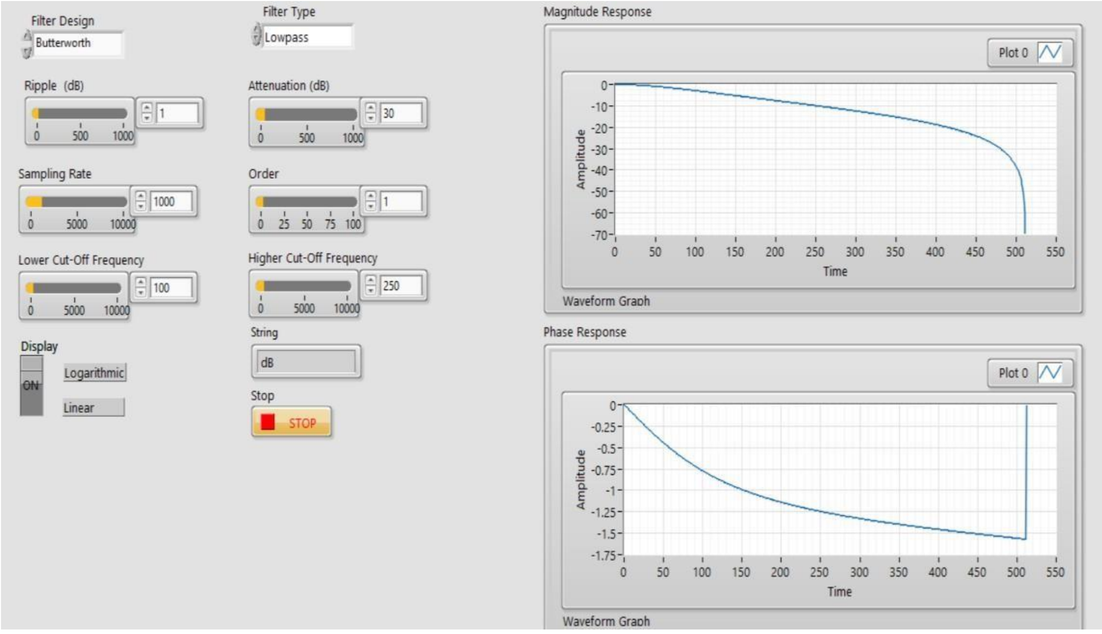


Fig. 3.4 Front panel of Butterworth LPF

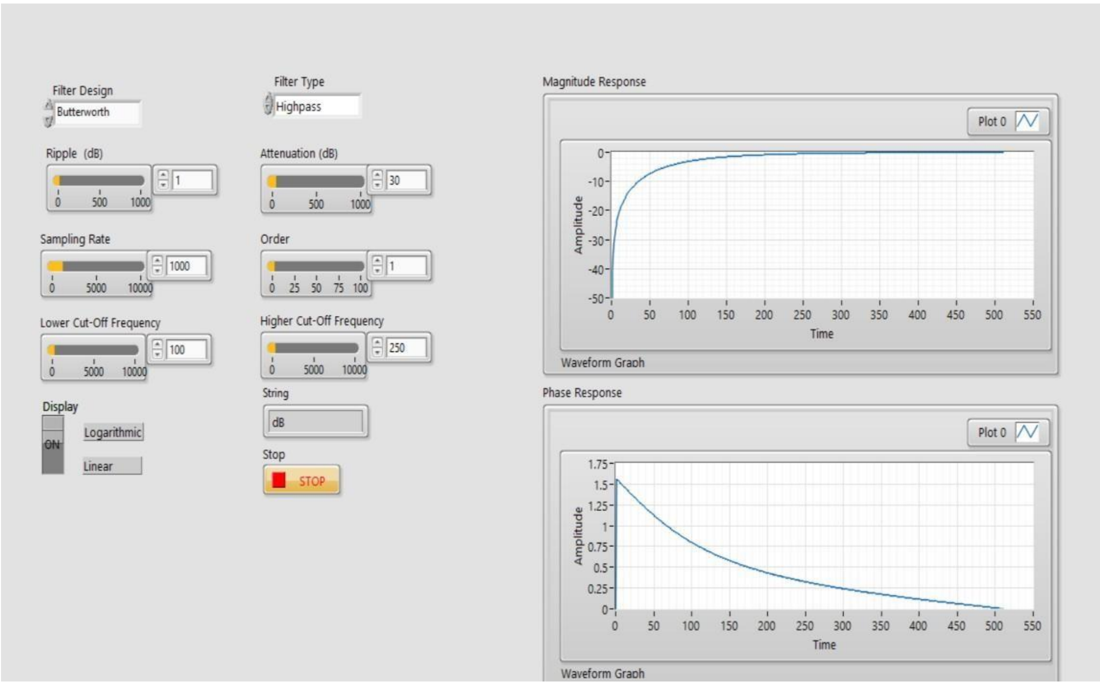


Fig. 3.5 Front panel of Butterworth HPF

The Fig. 3.4 and 3.5 shows front panel output of IIR filters. According to the requirements, the types of filters can be changed like Lowpass, High pass, Bandpass and Band stop filters. Also, the filter design such as Butterworth, Chebyshev, etc. can be varied accordingly. The Lower cut-off, Higher cut-off and the sampling frequency are specified to get the magnitude and phase response of the given filter.

➢ IIR Filtering system

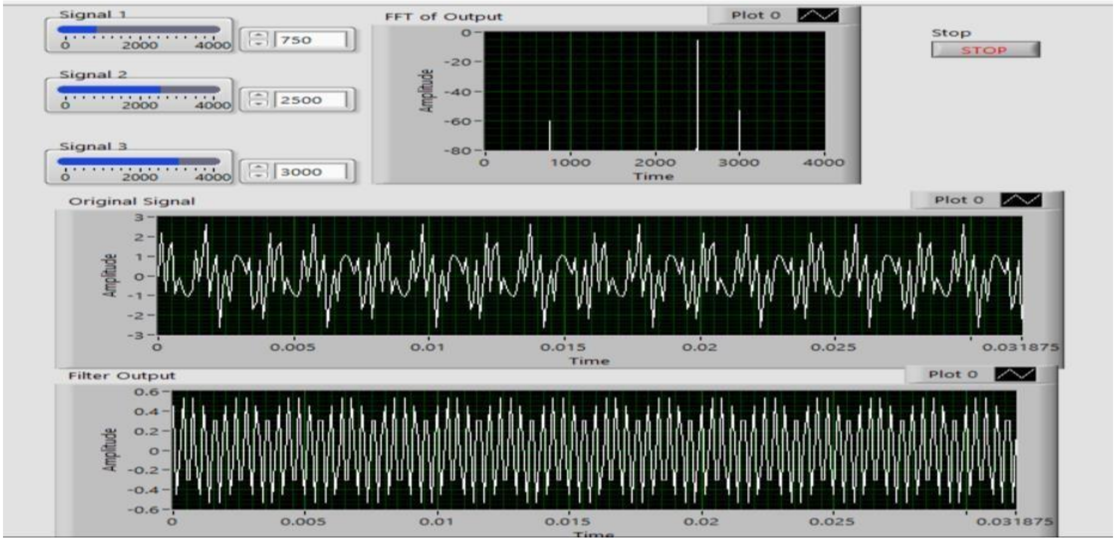


Fig. 3.6 Front panel of IIR Filtering system

The Fig. 3.6 shows the front panel of the IIR filtering system output. Here, the original signal is the signal with noise and after filtering, filter output waveform is the signal after removing the noise. (Here bandpass filter is used, so the signals which do not pass through it are classified as noise and are removed.)

➢ Noise Removal Block

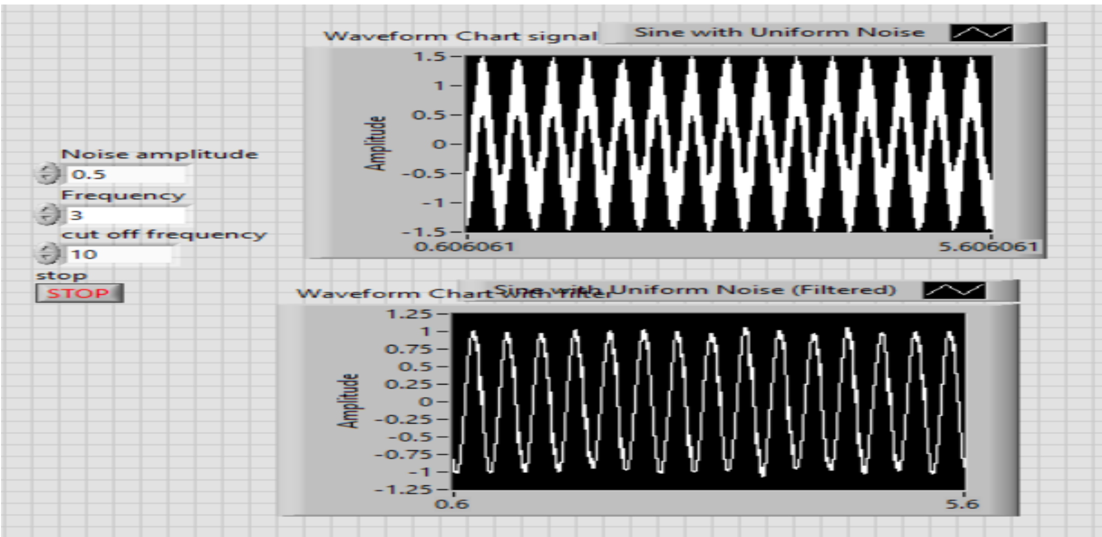


Fig. 3.7 Front Panel of Noise Removal Block

Fig. 3.7 is the front panel output of the noise removal block. The original signal is the sine wave with uniform noise. After filtering, the output obtained is the sine wave by filtering out the uniform noise.

3.2 Results of Image Processing Techniques

Front Panels of different Image Processing Techniques are discussed.

- **Edge Detection of Image**

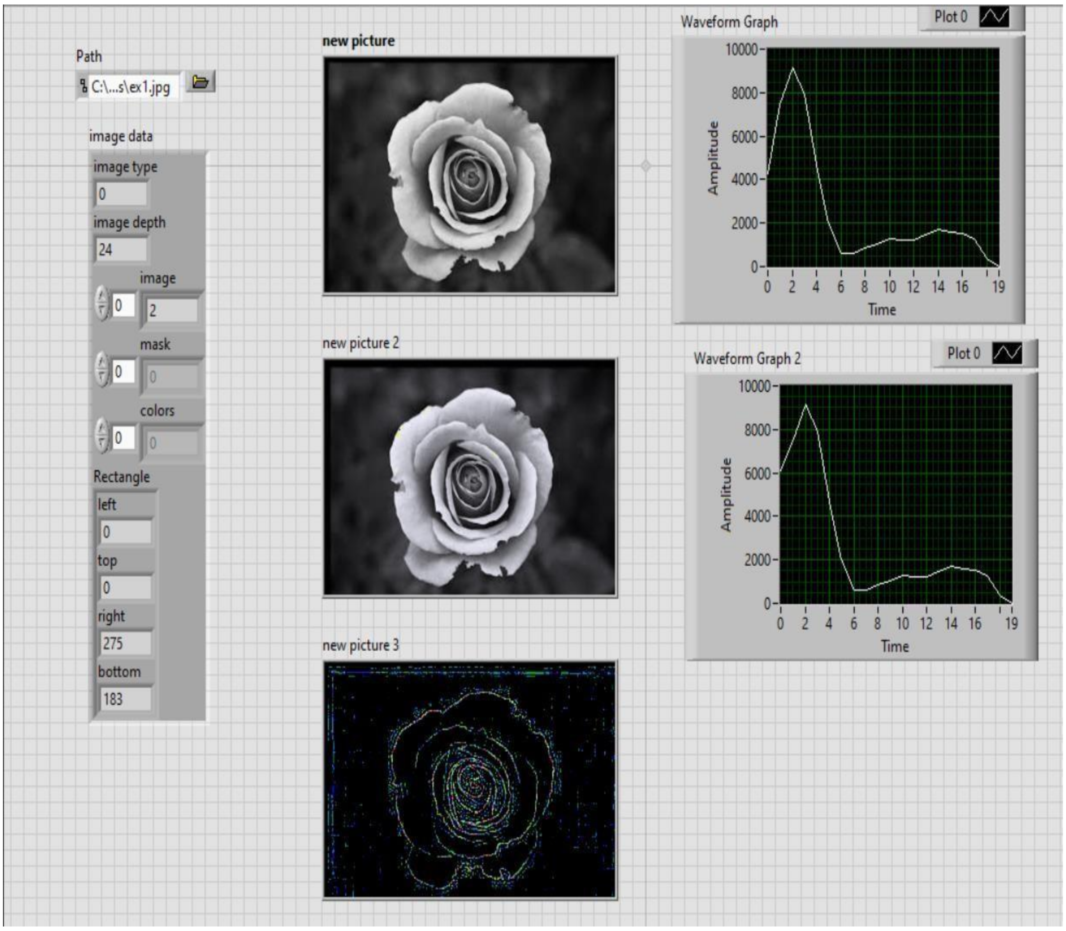


Fig. 3.8 Front Panel of Edge Detection of an Image

The Front Panel shown in Fig. 3.8 includes of a File Path from which the Image is read, and the window "new picture" displays the original Image. The Histogram of this Image can be seen in "waveform Graph", the "new picture 2" displays the sharpened image and "Waveform Graph 2" displays the histogram of that image. The Image data displays number of parameters which can be seen in the Image. Result can be seen in "new picture 3" window which displays the Edge detection of the input Image.

The Edge detection output is such that it can highlight the edges of the Image and suppressing the other parts of the Image. The Sharpening of Image is necessary to perform to avoid losing edges of the Image in detection process.

- **Image Morphology Block**

a. Erosion

Fig. 3.9 Front Panel of Erosion Morphological process

b. Dilation

Fig. 3.10 Front Panel of Dilation Morphological process

Fig. 3.9 and 3.10 depict the Front Panel of the Image Morphological Process. It can be observed from the figures that it takes inputs as Image type, in this case it is Grayscale Image along with path of the Image and the operation need to be performed on that Image. In Fig. 3.9 it is Erosion operation and in 3.10 it is Dilation process. Erosion process removes pixels on object boundaries and Dilation process adds pixels to the boundaries of objects in an image.

- **Median Filter**

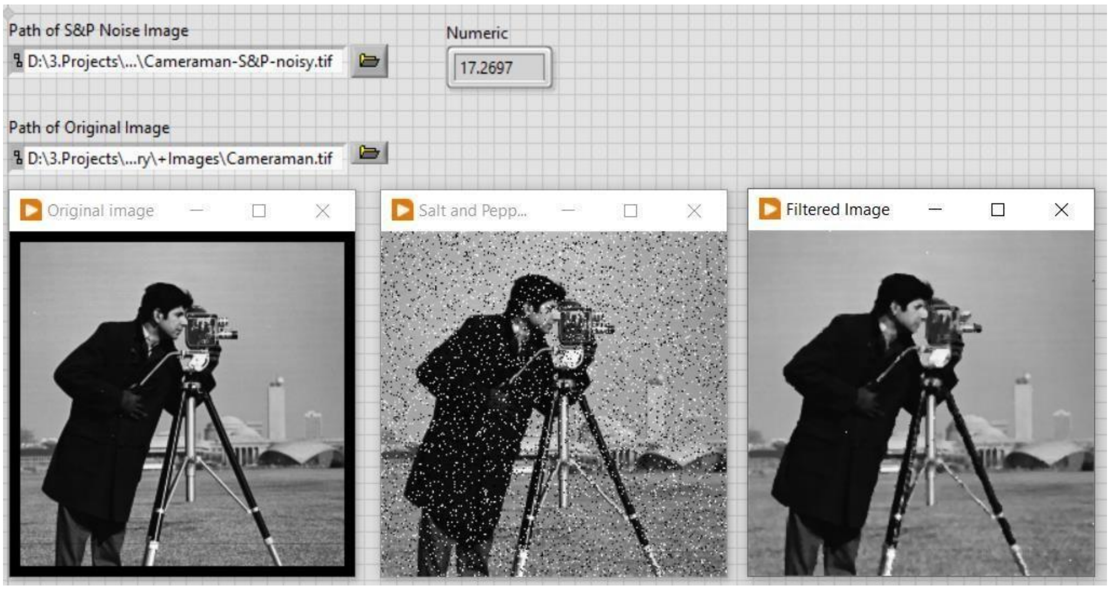


Fig. 3.11 Front Panel of Median Filter operation

The above front panel includes paths of original Image and Noisy Image as Inputs. It can be observed that 3 Image Windows displaying the Image are fetched from the paths and the output Image. The "Numeric" value displays the PSNR value of the Images compared. The Image path is of an 8-bit S&P noisy Image and path of original Image fetches the original Image without any noise. The IMAQ Nth order VI performs the median filter operation on the Image and gives out the output Image as Filtered Image. This completes the Median Filtering of an 8-bit S&P noisy Image.

3.3 Results of Adaptive Filter

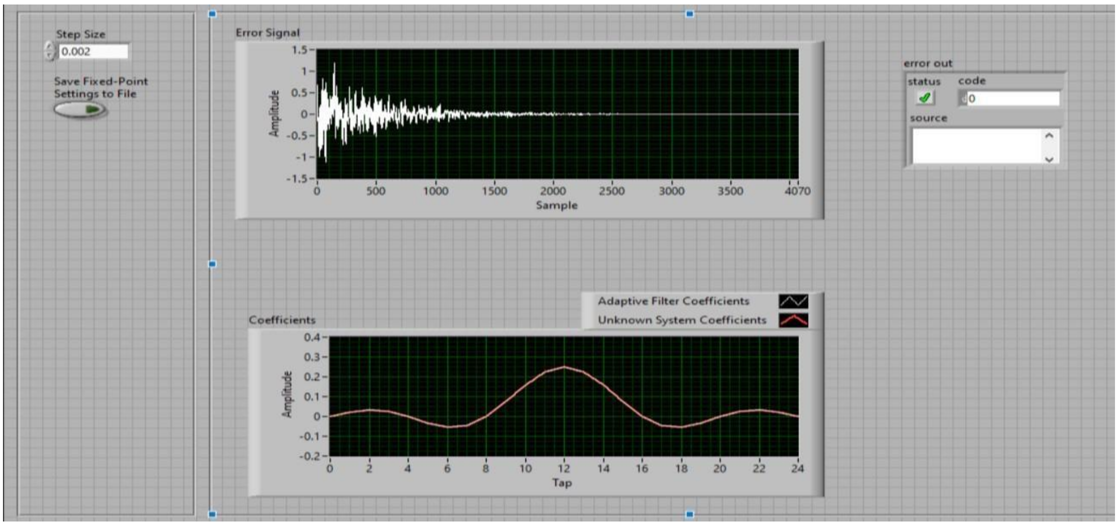


Fig. 3.12 Front Panel of LMS Adaptive Filter

In Fig. 3.12, the output of an Adaptive Filter designed using LMS algorithm is observed. It consists of two graphs, Error Signal and co-efficient. The Error signal is gradually decreased as the iterations of the LMS adaptive Filter starts processing. It is observed that after a certain iteration, the error signal significantly reduces and become flat. Also, the co-efficient signal is also moving according to the removal of error as seen in the above front panel.

- **Summary**

In this chapter, the outputs of the Digital filters, Image processing, and adaptive filter designed in the block diagram panel are observed in the front panel with varying parameters of input.

Chapter 4

CONCLUSION AND FUTURE SCOPE

Verification of Digital Filter design, Image processing techniques and adaptive filters were carried out and results were observed using LabVIEW platform. LabVIEW® software tool is available in the department of Electronics & Communication Engineering, KLEIT, Hubballi, procured with the MODROB AICTE Grants, received in Dec/Jan 2020.

4.1 Conclusion

With verification of Digital Filter design, Image processing techniques and adaptive filters, following points are observed.

i. Virtual Instruments (VI's) in LabVIEW® platform are used in the design of digital filters. The FIR and IIR Digital Filters are designed and simulated according to different specifications using LabVIEW. This approach was selected to eliminate necessity of coding and to provide better understanding about the working of digital filters and the results are presented as in sections 2.1 and 3.1.

ii. The Image processing techniques such as Edge Detection, Morphological Processing (Erosion and Dilation) and Median Filtering for a noisy Image are applied and the results are discussed as in sections 2.2 and 3.2.

iii. The Adaptive filter using LMS algorithm is created and simulated as in Fig. 2.10 and 3.12.

iv. This approach of design and verification in LabVIEW, enables the user to better visualize the working and results of the digital filters and image processing. The user can provide desired inputs through the front panel of the LabVIEW software and observe the results in the same panel.

v. A beginner as a learner in LabVIEW usage, can refer to this report and take benefit of a quick reference.

4.2 Scope for Future Work

Different possibilities of taking up future work are mentioned as below-

- This work can be further extended to create similar designs which can be helpful in visualizing the practical aspects of digital signal processing.
- LabVIEW can also be used in developing embedded systems.
- It can also be used for Machine learning and IOT applications.
- The above designs can also be implemented on processors developed by national instruments, by doing so it is possible to implement the above designs in real time.

BIBLIOGRAPHY

[1] Ranjushree Pal, "Comparison of the design of FIR and IIR filters for a given specification and removal of phase distortion from IIR filters", International Conference on Advances in Computing, Communication and Control (ICAC3), Mumbai, 2017, pp. 1-3.

[2] Eduardo Perez and Sam Shearman, "LabVIEW DSP - A Hands-on DSP Educational Platform", IEEE 13th Digital Signal Processing Workshop and 5th IEEE Signal Processing Education Workshop, Marco Island, FL, 2009, pp. 232-237, doi: 10.1109/DSP.2009.4785927.

[3] Divya Sharma and Rashpinder Kaur, "Design and Analysis of IIR Notch Filter using LabVIEW", 2015 IEEE International Conference on Computational Intelligence & Communication Technology, 2015, pp. 612-616, doi: 10.1109/CICT.2015.52.

[4] Anusha Nellutla, "Image Processing Techniques Using LabVIEW", International Journal of Latest Technology in Engineering, Management & Applied Science (IJLTEMAS) Vol. VII, Issue VIII, Aug. 2018.

[5] Sambit Satpathy, Mohan Chandra Pradhan and Subrat Sharma, "Comparative Study of Noise Removal Algorithms for Denoising Medical Image Using LabVIEW", International Conference on Computational Intelligence and Communication Networks (CICN), 2015, pp. 300- 305, doi: 10.1109/CICN.2015.67.

[6] Mahesh L. Chugani, Abhay R. Samant, Michael Cerna, "LabVIEW Signal Processing", 4th Edition, Upper saddle River USA, Prentice Hall PTR, 1998.

[7] Thomas Klinger, "Image Processing with LabVIEW and IMAQ Vision", 2nd Edition, Upper Saddle River USA, Prentice Hall PTR, 2003.

[8] Arnd Frech, Markus Klugel and Peter Russer, "Adaptive filtering for noise cancellation and signal analysis in real-time", 2013 European Microwave Conference, Nuremberg, 2013, pp. 1123-1126.

APPENDIX A: LabVIEW Set up details

- Download the latest version of LabVIEW from the National Instruments official website.
- After the download, start the installation process.
- After successful installation, Install required add-ons from the NI Package manager.
- Open the software and click on “New Project”, to get new block diagram and front panel windows.
- Add the graphical code, to the block diagram window.
- In the Front Panel, controls need to be given by the user for the block diagram.
- After this, simulate the circuit on front panel using Run option.
- Observe the outputs like graphs in the front panel.