

# MINOR PROJECT REPORT on DIGITAL FILTER DESIGN in MATLAB

**Team Members:**

1. **AKSHIT MRIDUL(1504439)**
2. **AYAN DEBNATH(1504472)**
3. **SANTRUPTA MISHRA (1504421)**
4. **SHUBHAM SAURAV(1504429)**

Under the guidance of

**Dr. SANANDA KUMAR**

|  |  |  |
| --- | --- | --- |
| School of Electronics Engineering | | |
| **KALINGA INSTITUTE OF INDUSTRIAL TECHNOLOGY**  (Deemed to be University)  Bhubaneshwar, April 2018 | | |
|  | | |
|  | | |
| **CERTIFICATE**  This is to certify that the project report entitled **“DIGITAL FILTER DESIGN in MATLAB”** submitted by :   |  |  | | --- | --- | | **AKSHIT MRIDUL**  **AYAN DEBNATH**  **SANTRUPTA MISHRA**  **SHUBHAM SAURAV** | **1504439**  **1504472**  **1504421**  **1504429** |   in partial fulfilment of the requirements for the award of the **Degree of Bachelor of Technology** in **Electronics and Telecommunication Engineering**  is a bonafide record of the work carried out under my(our) guidance and supervision at School of Electronics Engineering, KIIT University. | | | | |
| Signature of Supervisor  Prof. **Dr. Sananda Kumar**  School of Electronics Engineering,  KIIT University | |
|  | | | | |
|  | | | | |
| **The Project was evaluated by us on \_\_\_\_\_\_\_\_\_\_\_\_\_** | | | | |
|  | | | | |
| EXAMINER 1 | | | EXAMINER 2 | |
| EXAMINER 3 | | | EXAMINER 4 | |

**ACKNOWLEDGEMENTS**

We feel immense pleasure and feel privileged in expressing our deepest and most sincere gratitude to our supervisor **Dr. Sananda Kumar**, for his excellent guidance throughout our project work. His kindness, dedication, hard work and attention to detail have been a great inspiration to us. Our heartfelt thanks to you sir for the unlimited support and patience shown to us. We would particularly like to thank him for all his help in patiently and carefully correcting all our manuscripts.

We are also very thankful to **Professor Ghanashyam Rout** B.tech project coordinator (E&TC), Associate Dean Professor **Dr. Amlan Datta** and **Professor Dr. Arun Kumar Ray**, Dean (School Of Electronics) for their support and suggestions during our course of the project work in the pre-final year of our undergraduate course.

AKSHIT MRIDUL (1504439)

AYAN DEBNATH (1504472)

SANTRUPTA MISHRA (1504421)

SHUBHAM SAURAV (1504429)

|  |  |
| --- | --- |
|  |  |

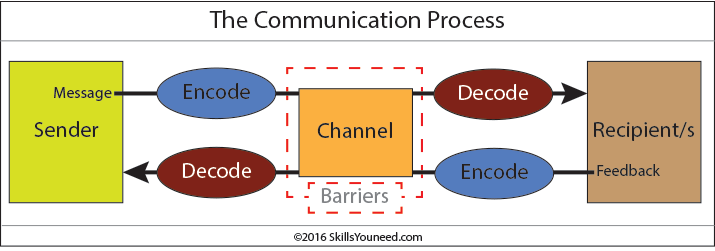
**Table of Contents:**

|  |  |  |
| --- | --- | --- |
| **Sr. no.** | **Topic** | **Page no.** |
| 1. | Abstract | 3 |
| 2. | Threats in Communication | 3 |
| 3. | What are filters? | 4 |
| 4. | Analog filters | 5 |
| 5. | Digital filters | 6 |
| 6. | Why digital filters? | 7 |
| 7. | Infinite impulse response filters | 7 |
| 8. | Finite impulse response filters | 8 |
| 9. | Why FIR? | 9 |
| 10. | Advantages over Infinite Impulse Response | 10 |
| 11. | Band pass filter | 10 |
| 12. | Low pass filter | 12 |
| 13. | Band stop filter | 13 |
| 14. | High pass filter | 15 |
| 15. | Objectives | 16 |

**Abstract:**

FIR filters are a practical and versatile tool used in signal processing for various applications. They are used to process an input signal and modify it according to the user’s needs e.g. for the removal of noise in the channel. Digital filters are a cost effective and convenient way to realise this. Through the adjustment of the filter parameters, one can remove distortions in a transmitted signal that is sent through a channel or medium. Since FIR filters are easy to realise digitally, MATLAB is the preferred software for the designing of such filters. The Filter Designer App (FDA) is a convenient, user friendly tool that provides a GUI for easy observation of output spectra for different inputs. Noise removal is a key aspect in telecommunication, especially in telephone lines. We shall design different static FIR filters for noise removal and later on realise a channel equaliser using DSP kit DSK6713 and the tools mentioned above.

**Communication:**



In today’s modern society, fast and effective communication has become a necessity in everyday life. The emergence of smartphones has cemented this sentiment, with people having easy access to technology that enables communication over long distances. In such situations, it is imperative that accurate and clear information reaches the receiver in order to avoid confusion.

Voice is a popular means of communication lately. The effect of noise is especially significant on it. Hence certain methods need to be implemented to filter out the noise. With humans having a listening range of 20 Hz-20 kHz and the usual range of telephony extending upto 4 kHz, there are noise frequencies that can interfere with communication. In the rest of the report, we shall deal with removing this noise using static digital filters.

**Threats in Communication:**

The major threats in the process of digital communication are:

* Presence of noise in the channel.
* Intersymbol interference.
* Multipath propagation.

Channel equalisation deals with eliminating these parameters. In this minor project, we shall deal with the first aspect i.e. removal of noise from an input signal.

**What are filters?:**

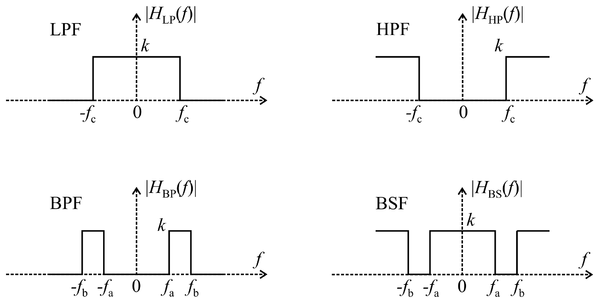


Fig 1: Different filters used in Digital Signal Processing

In signal processing, a filter is a device or process that removes some unwanted components or features from a signal. Most often, this means eliminating certain frequencies or frequency bands. However, filters do not only act in the frequency domain; especially in the field of image processing many other targets for filtering exist. Correlations can be removed for certain frequency components and not for others without having to function in the frequency domain. Filters are widely used in electronics and telecommunication, in radio, television, audio recording, radar, control systems, music synthesis, image processing, and computer graphics.

**Analog filters:**

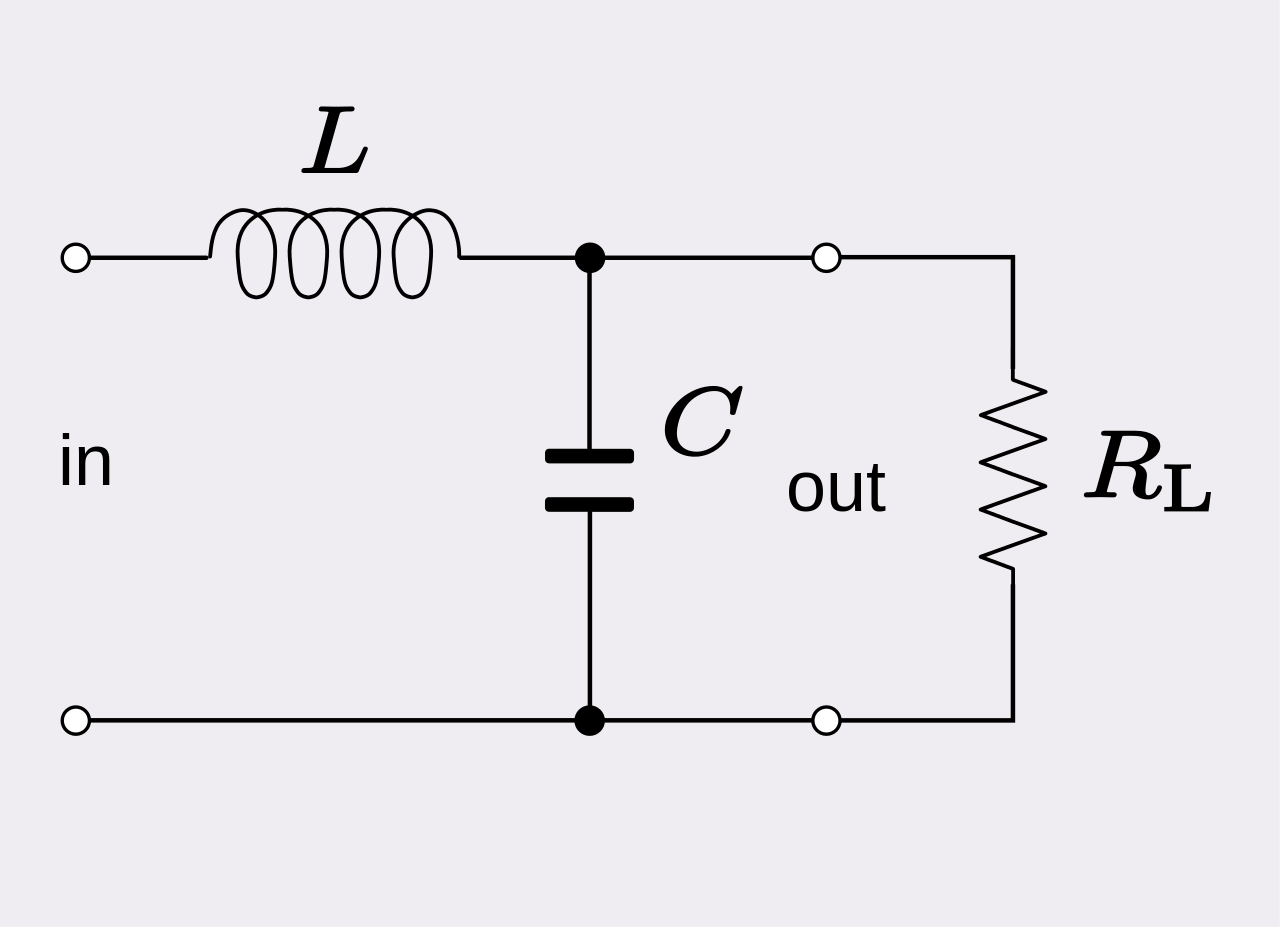


Fig 2: An analog low pass filter designed using an RLC circuit

Analog filters are a basic building block of signal processing widely used in electronics. They are designed to be IIR filters. Amongst their many applications are the separation of an audio signal before application to bass, mid-range and tweeter loudspeakers. The combining and later separation of multiple telephone conversations onto a single channel. The selection of a chosen radio station in a radio receiver and rejection of others.

Passive linear electronic analog filters are those filters which can be described with linear differential equations (linear). They are composed of capacitors, inductors and, occasionally, resistors (passive) and are made to operate on continuously varying (analog) signals.

**Digital Filters:**

In signal processing, a digital filter is a system that performs mathematical operations on a sampled, discrete-time signal to change certain parameters of that signal. This is different to analog filters, which are electronic circuits operating on continuous-time analog signals.

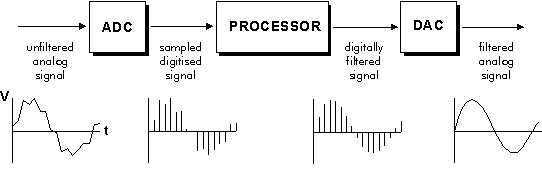


Fig 3: Block diagram of a digital filter

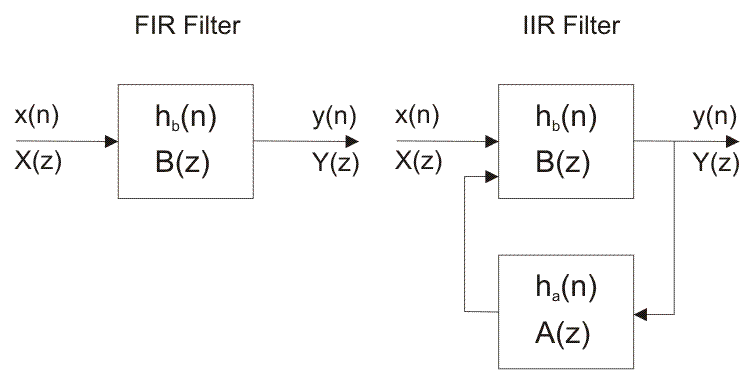
A digital filter system generally consists of an analog-to-digital converter to sample the input signal, followed by a microprocessor and some peripheral components such as memory to store data and filter coefficients. Finally, a digital-to-analog converter is used to obtain the output. Program Instructions (software) running on the microprocessor are used to enable the functioning of the digital filter by performing the necessary mathematical operations on the numbers received from the ADC. In some high performance applications, an FPGA or ASIC is used instead of a general purpose microprocessor, or a specialized DSP, such as the DSK6713,with specific paralleled architecture for expediting operations such as filtering.

**Why Digital filters?:**

* They have linear phase response.
* Thermal and environmental variation cannot affect the performance.
* They are called adaptive filters because the frequency response can be adjusted automatically with implementation of a programmable processor.
* It is possible to filter several input sequences without any hardware replication.
* All data can be stored.
* They have repeatable performance unit to unit.
* Due to them operating at low frequencies, they are used where the use of analog system is impractical.

**Infinite Impulse Response Filters:**

IIR filters are digital filters with infinite impulse response. Unlike FIR filters, the output is fed back to the input and are therefore known as recursive digital filters.

Figure 4.1: Block diagrams of FIR and IIR filters

For this reason IIR filters have much better frequency response than FIR filters of the same order. Unlike FIR filters, their phase characteristic is not linear which might cause a problem to the systems which need phase linearity. Due to this, it is not preferable to use IIR filters in digital signal processing when the phase is important.  
  
Otherwise, when the linear phase characteristic is unimportant, the use of IIR filters is an excellent option.

**Finite Impulse Response Filters:**

In digital signal processing, an FIR filter is one whose impulse response is of finite period, as a result of it settling to zero in finite time. This is often in contrast to IIR filters, which can have internal feedback and will still respond indefinitely. The impulse response of an Nth order discrete time FIR filter takes exactly N+1 samples before it then settles to zero. FIR filters are [most popular kind of filters](https://www.elprocus.com/types-of-chebyshev-filters/) executed in software and these filters can be continuous time, analog or digital and discrete time. Special types of FIR filters are namely, Boxcar, Hilbert Transformer, Differentiator, Lth-Band and Raised-Cosine.

A FIR filter is used to implement almost any type of digital frequency response. Usually these filters are designed with a multiplier, adders and a series of delays to create the output of the filter. The following figure shows the basic FIR filter diagram with N length. The result of delays operates on input samples. The values of hk are the coefficients which are used for multiplication. So that the o/p at a time and that is the summation of all the delayed samples multiplied by the appropriate coefficients.

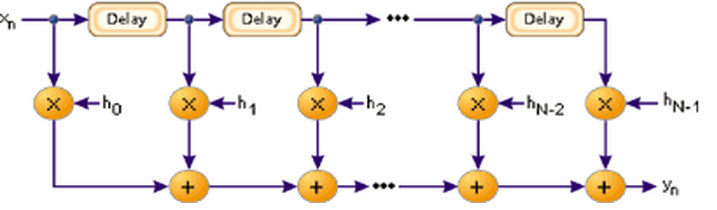


Fig 4.2: A basic FIR filter

The [filter design can be defined](https://en.wikipedia.org/wiki/Filter_(signal_processing)) as, it is the process of choosing the length and coefficients of the filter. The intention is to set the parameters so that the required parameters like a stop band and pass band will give the result from running the filter. Most of the engineers use MATLAB software to design the filter.

Usually, filters are defined by their responses to the separate frequency [components that found](https://www.elprocus.com/basic-components-used-electronics-electrical/) the i/p signal The responses of the filters a classified into three types based on the frequencies such as stop band, pass band and transition band. The response of the passband is the filter’s effect on frequency components that are delivered through mostly unaffected.

Frequencies in a filter’s stopband are, by difference, highly reduced. The transition band signifies the frequencies in the middle, which may receive some reduction, but are not detached completely from the o/p signal.

**Why FIR?:**

* They can easily be designed to be “linear phase” (and usually are). Put simply, linear-phase filters delay the input signal but don’t distort its phase.
* They are simple to implement. On most DSP microprocessors, the FIR calculation can be done by looping a single instruction.
* They are suited to multi-rate applications. By multi-rate, we mean either “decimation” (reducing the sampling rate), “interpolation” (increasing the sampling rate), or both. Whether decimating or interpolating, the use of FIR filters allows some of the calculations to be omitted, thus providing an important computational efficiency. In contrast, if IIR filters are used, each output must be individually calculated, even if it that output will discard (so the feedback will be incorporated into the filter).
* They have desirable numeric properties. In practice, all DSP filters must be implemented using finite-precision arithmetic, that is, a limited number of bits. The use of finite-precision arithmetic in IIR filters can cause significant problems due to the use of feedback, but FIR filters without feedback can usually be implemented using fewer bits, and the designer has fewer practical problems to solve related to non-ideal arithmetic.
* They can be implemented using fractional arithmetic. Unlike IIR filters, it is always possible to implement a FIR filter using coefficients with magnitude of less than 1.0. (The overall gain of the FIR filter can be adjusted at its output, if desired.) This is an important consideration when using fixed-point DSPs, because it makes the implementation much simpler.

**Advantages over Infinite Impulse Response filters:**

* Because of feedback present in IIR, they are somewhat difficult to control.
* FIR filter consume low power and IIR filter need more power due to more coefficients in the design.
* IIR filters require more adders and multipliers.
* FIR filter have linear phase characteristics whereas IIR are non linear.

**Band Pass Filters:**

A band-pass filter (also bandpass filter, BPF) is a device that passes frequencies within a certain range and rejects (attenuates) frequencies outside that range.

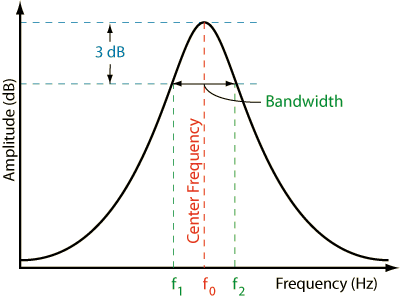


Fig 5: Band pass filter parameters

An example of an analog electronic band-pass filter is an RLC circuit (a resistor–inductor–capacitor circuit). These filters can also be created by combining a low-pass filter with a high-pass filter.

Bandpass is an adjective that describes a type of filter or filtering process; it is to be distinguished from passband, which refers to the actual portion of affected spectrum. Hence, one might say "A dual bandpass filter has two passbands." A bandpass signal is a signal containing a band of frequencies not adjacent to zero frequency, such as a signal that comes out of a bandpass filter.

An ideal bandpass filter would have a completely flat passband (e.g. with no gain/attenuation throughout) and would completely attenuate all frequencies outside the passband. Additionally, the transition out of the passband would have brickwall characteristics.

In practice, no bandpass filter is ideal. The filter does not attenuate all frequencies outside the desired frequency range completely; in particular, there is a region just outside the intended passband where frequencies are attenuated, but not rejected. This is known as the filter roll-off, and it is usually expressed in dB of attenuation per octave or decade of frequency. Generally, the design of a filter seeks to make the roll-off as narrow as possible, thus allowing the filter to perform as close as possible to its intended design. Often, this is achieved at the expense of pass-band or stop-band ripple.

**Low Pass Filters:**

A Low Pass Filter is a circuit that can be designed to modify, reshape or reject all unwanted high frequencies of an electrical signal and accept or pass only those signals wanted by the circuits designer.

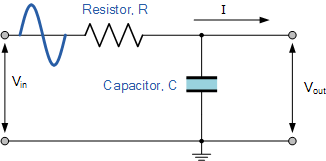


Fig 6: A low pass filter designed using a resistor and capacitor

A simple passive RC Low Pass Filter or LPF, can be easily made by connecting together in series a single Resistor with a single Capacitor as shown above. In this type of filter arrangement the input signal ( Vin ) is applied to the series combination (both the Resistor and Capacitor together) but the output signal ( Vout ) is taken across the capacitor only.

This type of filter is known generally as a “first-order filter” or “one-pole filter”, why first-order or single-pole?, because it has only “one” reactive component, the capacitor, in the circuit.

The reactance of a capacitor varies inversely with frequency, while the value of the resistor remains constant as the frequency changes. At low frequencies the capacitive reactance, ( Xc< ) of the capacitor will be very large compared to the resistive value of the resistor, R.

This means that the voltage potential, Vc across the capacitor will be much larger than the voltage drop, Vr developed across the resistor. At high frequencies the reverse is true with Vc being small and Vr being large due to the change in the capacitive reactance value.

While the circuit above is that of an RC Low Pass Filter circuit, it can also be thought of as a frequency dependant variable potential divider circuit.

where: ; the total resistance of the circuit

We also know that the capacitive reactance of a capacitor in an AC circuit is given as:

Opposition to current flow in an AC circuit is called **impedance**, symbol Z and for a series circuit consisting of a single resistor in series with a single capacitor, the circuit impedance is calculated as:

**Band Stop Filters:**

The **band stop filter** blocks signals falling within a certain frequency band set up between two points while allowing both the lower and higher frequencies either side of this frequency band.

It is formed by the combination of low pass and high pass filters with a parallel connection instead of cascading connection.

Since it eliminates frequencies, it is also called as band elimination filter or band reject filter or notch filter.

Unlike high pass and low pass filters, band pass and band stop filters have two cut-off frequencies. It will pass above and below a particular range of frequencies whose cut off frequencies are predetermined depending upon the value of the components used in the circuit design. Any frequencies in between these two cut-off frequencies are attenuated. It has two pass bands and one stop band. The ideal characteristics of the Band pass filter are as shown below.

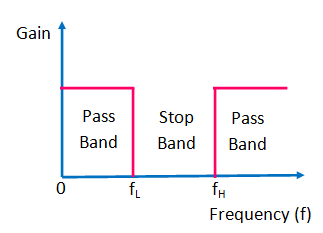


Fig 7: Ideal band stop filter characteristics

Where :

*fL* indicates the cut off frequency of the low pass filter.

*fH*is the cut off frequency of the high pass filter.

The centre frequency  *fc = √( fL x fH)*

When the input signal is given, the low frequencies are passed through the low pass filter in the band stop circuit and the high frequencies are passed through the high pass filter in the circuit. This is shown in below block diagram.

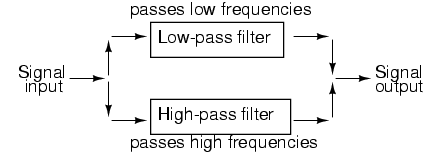
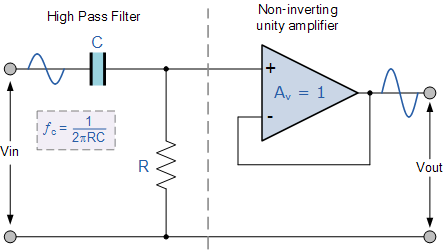


Fig 8: Band stop filter using a low pass and high pass filter

The R-C band stop filter is similar to a band pass filter in which the shunt arm is replaced by the series arm  and the series arm is replaced by the shunt one.

**High Pass Filter:**

An Active High Pass Filter can be created by combining a passive RC filter network with an operational amplifier to produce a high pass filter with amplification

****

**Fig 9: Active high pass filter**

The basic operation of an Active High Pass Filter (HPF) is the same as for its equivalent RC passive high pass filter circuit, except this time the circuit has an operational amplifier or included within its design providing amplification and gain control.

Like the previous active low pass filter circuit, the simplest form of an *active high pass filter* is to connect a standard inverting or non-inverting operational amplifier to the basic RC high pass passive filter circuit as shown.

Technically, there is no such thing as an active high pass filter. Unlike Passive High Pass Filters which have an “infinite” frequency response, the maximum pass band frequency response of an active high pass filter is limited by the open-loop characteristics or bandwidth of the operational amplifier being used, making them appear as if they are band pass filters with a high frequency cut-off determined by the selection of op-amp and gain.

In the Operational Amplifier tutorial we saw that the maximum frequency response of an op-amp is limited to the Gain/Bandwidth product or open loop voltage gain ( AV ) of the operational amplifier being used giving it a bandwidth limitation, where the closed loop response of the op amp intersects the open loop response.

A commonly available operational amplifier such as the uA741 has a typical “open-loop” (without any feedback) DC voltage gain of about 100dB maximum reducing at a roll off rate of -20dB/Decade (-6db/Octave) as the input frequency increases. The gain of the uA741 reduces until it reaches unity gain, (0dB) or its “transition frequency” ( ƒt ) which is about 1MHz. This causes the op-amp to have a frequency response curve very similar to that of a first-order low pass filter and this is shown below.

**Significance of order in a Filter:**

Generally speaking, the higher the order, some combination of the following can be achieved:

* + 1. More attenuation between the pass band and stop band
    2. Narrow transition band.
    3. Flatter pass band (less ripple)

On the other hand, these are the drawbacks we face while raising the order :

* + 1. More compute intensive
    2. Higher resource cost (SW or HW)
    3. Longer group delay
    4. High chance for instability in case of IIR filters
    5. Higher dynamic range requiring more costly data format: single precision to double precision, integer to floating points.

Thus it is important to find a balance between accuracy and resource management. Designing a filter of too high order might lead to too much computation time and possibly inaccurate results due to the designing process timing out while a lower order filter will not filter out the noise effectively enough.

**Power Spectral Density :**

The Power Spectral Density (PSD) of a signal shows us the spectral energy distribution per unit time over a large enough time period. This can be used to estimate the frequencies at which a signal is most powerful. In essence, one can identify the frequencies to filter out in case one wants to eliminate noise signals.

To estimate the power spectral density(P.S.D) of a signal in MATLAB, there are functions like pwelch( ) and dspdata.psd( ).

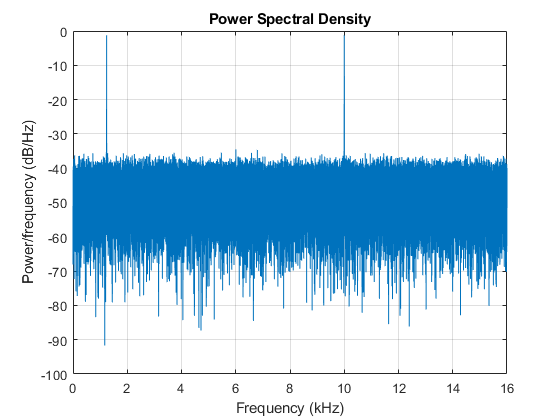
* Pwelch( ) returns the Welch’s power spectral density estimate.
* pxx = pwelch(x)
* pxx = pwelch(x,window)
* pxx = pwelch(x,window,noverlap)
* pxx = pwelch(x,window,noverlap,nfft)

pxx = pwelch(x) returns the power spectral density (PSD) estimate, pxx, of the input signal, x, found using Welch's overlapped segment averaging estimator. When x is a vector, it is treated as a single channel. When x is a matrix, the PSD is computed independently for each column and stored in the corresponding column of pxx. If x is real-valued, pxx is a one-sided PSD estimate. If x is complex-valued, pxx is a two-sided PSD estimate. By default, x is divided into the longest possible segments to obtain as close to but not exceed 8 segments with 50% overlap. Each segment is windowed with a Hamming window. The modified periodograms are averaged to obtain the PSD estimate. If you cannot divide the length of x exactly into an integer number of segments with 50% overlap, x is truncated accordingly.

pxx = pwelch(x,window) uses the input vector or integer, window, to divide the signal into segments. If window is a vector, pwelch divides the signal into segments equal in length to the length of window. The modified periodograms are computed using the signal segments multiplied by the vector, window. If window is an integer, the signal is divided into segments of length window. The modified periodograms are computed using a Hamming window of length window.

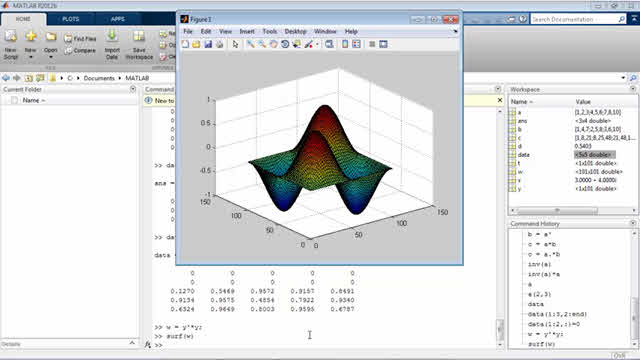
Whereas dspdata.psd() is intended for continuous spectra.

* Hpsd = dspdata.psd(Data)
* Hpsd = dspdata.psd(Data,Frequencies)
* Hpsd = dspdata.psd(Data) uses the power spectral density data contained in Data, which can be in the form of a vector or a matrix, where each column is a separate set of data.
* Hpsd = dspdata.psd(Data,Frequencies) uses the power spectral density estimation data contained in Data and Frequencies vectors



# TOOLS Required:

**MATLAB :**



MATLAB is a programming language developed by MathWorks. It is based on a matrix programming language which made linear algebra programming simple. It can be run both under interactive sessions and as a batch job. It is a fourth-gen high-level programming language and IDE for numerical computation, data visualization and programming. It allows matrix manipulations, plotting of functions and data, implementation of algorithms to solve problem statements, creation of UI , interfacing with programs written in other languages, including C, C++, Java and other high level languages, data analytics, develop algorithms in machine learning.

MATLAB is widely used in electronics engineering as it requires vast computations which can be made easier using it. Given below are few range of applications:

* 1. Signal Processing and Communications
  2. Image and Video Processing
  3. Control Systems
  4. Test and Measurement
  5. Computational Finance
  6. Computational Biology

Digital filters are central to almost every signal processing system. Filters eliminate unwanted artifacts from signals to enhance their quality and prepare them for further processing. Digital filters are used in a variety of signal processing tasks including outlier and noise removal, waveform shaping, signal smoothing, and signal recovery.

MATLAB provides extensive tools to design and realize digital filters.

Tools like FDAtool and SPtool are two significant tools used for filter design.

**FDA Tool:**

The Filter Design and Analysis Tool (FDATool) is a powerful graphical user interface(G.U.I) in the signal processing toolbox for designing and analyzing filters quickly.

FDATool enables us to design digital FIR or IIR filters by setting filter specifications, by importing filters from your MATLAB workspace, or by adding, moving or deleting poles and zeros.

FDATool also provides tools for analyzing filters, such as magnitude and phase response and pole-zero plots.

To open FDAtool type >>***fdatool in command window*** and the application will open.

There one can select FIR or IIR filter, order of filter and cutoff frequency of a filter (either HPF, LPF or BPF).

That code will automatically generate .m file for you. Record your voice. Analyze the effect of lowpass filtering on speech signal. Use the Filter Design and Analysis Tool (FDATool) of MATLAB for the purpose of designing LPF filter. Consider separately FIR and IIR filter.

After typing fdatool in the command window, a GUI pops up with deafault filter.

The GUI has three main regions:

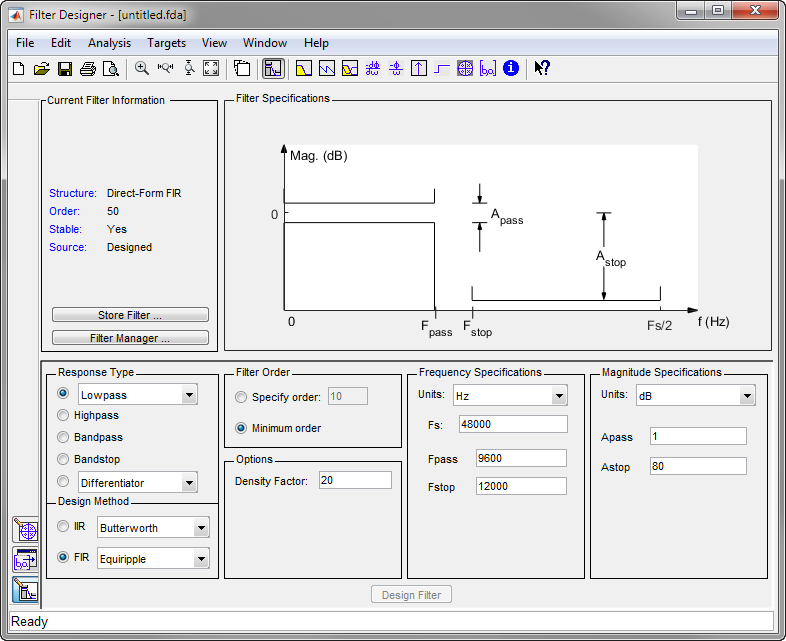
1. The Current Filter Information region

2. The Filter Display region and

3. The Design panel

The upper half of the GUI displays information on filter specifications and responses for the current filter.

The Current Filter Information region, in the upper left, displays filter properties, namely the filter structure, order, number of sections used and whether the filter is stable or not. It also provides access to the Filter manager for working with multiple filters.



The Filter Display region, in the upper right, displays various filter responses, such as, magnitude response, group delay and filter coefficients.

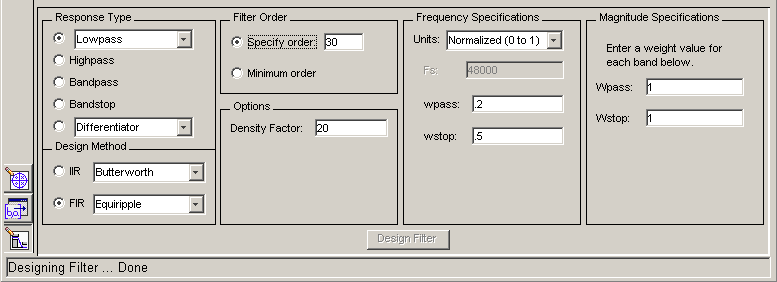
The lower half of the GUI is the interactive portion of FDATool. The Design Panel, in the lower half is where the programmer defines the filter specifications. It controls what is displayed in the other two upper regions. Other panels can be displayed in the lower half by using the sidebar buttons.

**Designing a Filter:**

We will design a low pass filter that passes all frequencies less than or equal to 20% of the Nyquist frequency (half the sampling frequency) and attenuates frequencies greater than or equal to 50% of the Nyquist frequency. We will use an FIR Equiripple filter with these specifications:

* Passband attenuation 1 dB
* Stopband attenuation 80 dB
* A passband frequency 0.2 [Normalized (0 to 1)]
* A stopband frequency 0.5 [Normalized (0 to 1)]

To implement this design, we will use the following specifications :



1. Select **Lowpass** from the dropdown menu under **Response Type** and **Equiripple** under **FIR Design Method**. In general, when you change the Response Type or Design Method, the filter parameters and Filter Display region update automatically.

2. Select **Specify order** in the **Filter Order** area and enter **30**.

3. The FIR Equiripple filter has a **Density Factor** option which controls the density of the frequency grid. Increasing the value creates a filter which more closely approximates an ideal equiripple filter, but more time is required as the computation increases. Leave this value at 20.

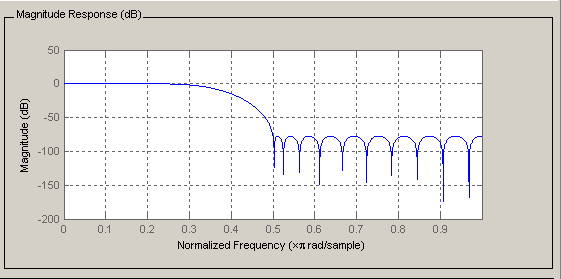
4. Select **Normalized (0 to 1)** in the Units pull down menu in the **Frequency Specifications** area.

5. Enter **0.2** for **wpass** and **0.5** for **wstop** in the **Frequency Specifications** area.

6. **Wpass** and **Wstop**, in the **Magnitude Specifications** area are positive weights, one per band, used during optimization in the FIR Equiripple filter. Leave these values at 1.

7. After setting the design specifications, click the **Design Filter** button at the bottom of the GUI to design the filter.

The magnitude response of the filter is displayed in the Filter Analysis area after the coefficients are computed.

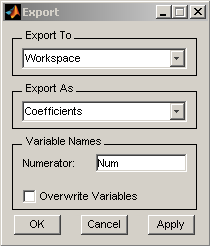


**Exporting the Filter:**

Once you are satisfied with your design, you can export your filter to the following destinations:

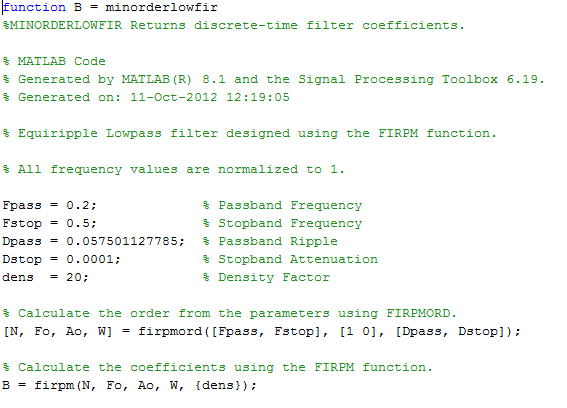
* MATLAB workspace
* MAT-file
* Text-file

Select **Export** from the **File** menu.



When we choose to export to the MATLAB workspace or to a MAT-file, we can export the filter as coefficients. If a DSP System Toolbox is available you can also export your filter as a System object.

Generating an M-File FDATool allows you to generate M-code to re-create your filter. This enables you to embed your design into existing code or automate the creation of your filters in a script. Select Generate M-file from the File menu and specify the filename in the Generate M-file dialog box.



**Objectives:**

1. To study different filters and to design low pass, band pass, band stop and high pass filters.

2. To develop static FIR filters to filter noise from audio signal.

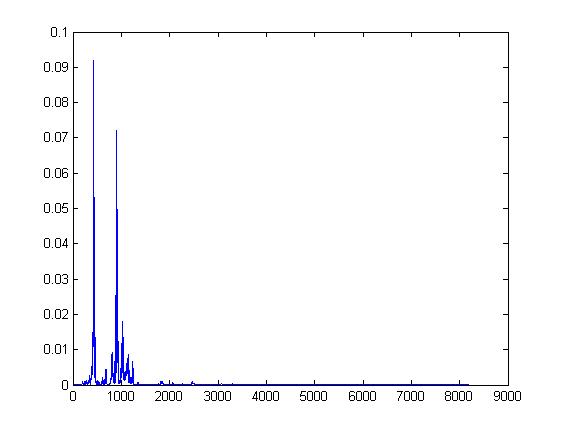
3. To equalize an audio signal channel taken at a particular sampling frequency which demonstrates the entire method using FDAtool.

**Process:**

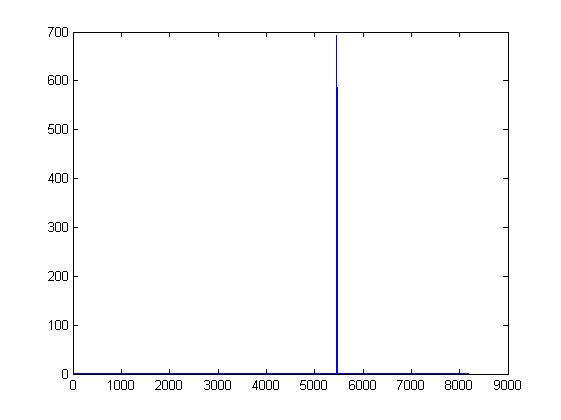
1. Take voice input using audiorecorder function.
2. Obtain array data from recorded signal and store in a variable **x**.
3. Declare noise signal **y**. This can be either sinusoidal with a fixed frequency, or other signals whose frequencies are not yet determined.
4. Using pwelch, plot and determine frequencies at which noise signals are most potent. Note the frequency range.
5. Mix signals **x** and **y**. Thus, we obtain the input, **r**, to the filter comprising of both voice input and noise signals.
6. Using FDAtool, design the desired filter by entering the requisite parameters. The frequency pass parameter will be set at 4 kHz in case of human voice signal.
7. In order to filter out the noise, enter the frequencies observed in the pwelch plot in the stop frequency parameter of the filter. This will ensure that the noise gets filtered out.
8. The sampling frequency is set at a value at least greater than two times the maximum human hearing frequency i.e. 40 kHz in accordance to the sampling theorem.
9. On designing the filter , import the coefficients to the workspace under the desired variable name, say, **a**.
10. Convolute the input signal **r** and the filter coefficients **a**. Observe the output using the sound function. You shall obtain a noise free signal.

**Input:**

1. Voice signal taken through microphone.
2. User defined input noise signal.



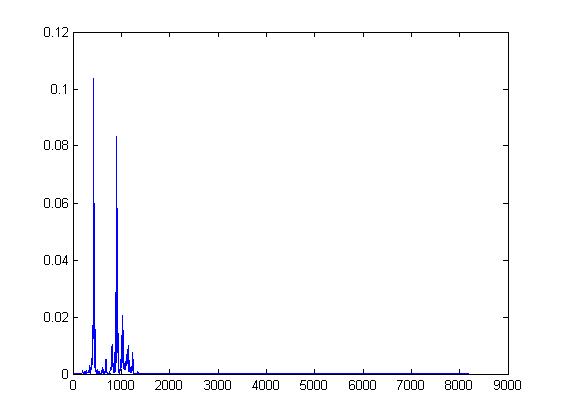
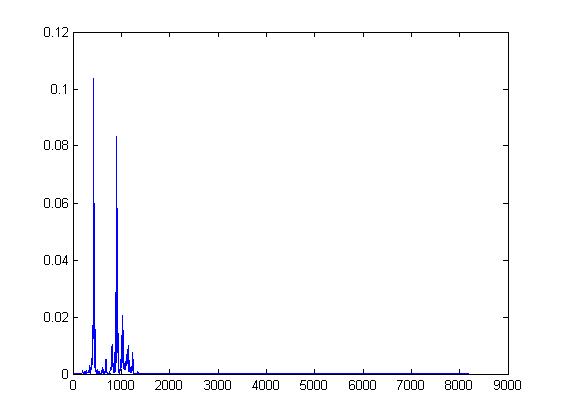
PSD of input voice signal

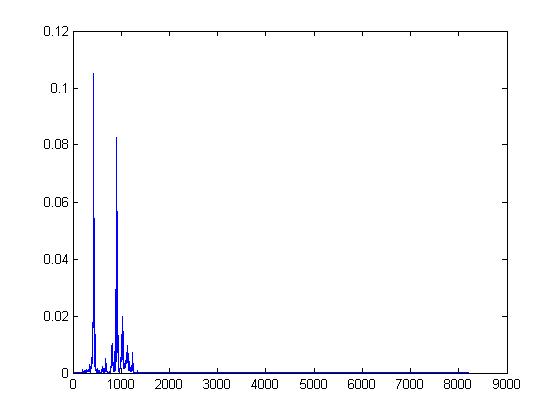
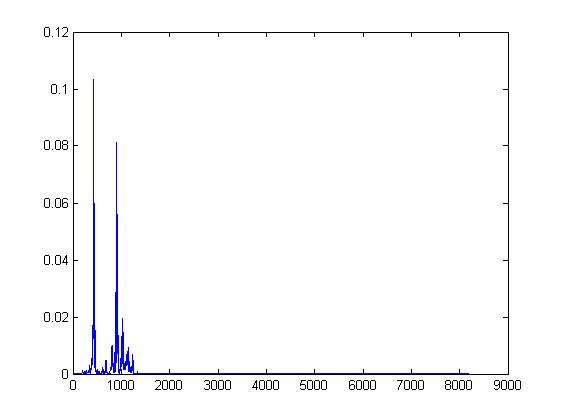


PSD of noise signal

**Output:**

1. Signal comprising of voice without noise.



****

PSD of outputs with different functions (from top left clockwise)  
a) Equiripple b) Least square method c) Window d) Constraint Least Square method

**Problems encountered/Precautions:**

1. Mixing the input voice signal and noise may lead to matrix dimension mismatch errors. This can be fixed by adjusting the noise signal to match the dimensions of the voice signal by limiting the number of array elements required to store it followed by transposition.
2. While designing the filter using FDA tool, one can come across errors when the stop frequency is lesser than the pass frequency. Ensure that there is no overlapping between the two.
3. A filter having a very high order might fail to get designed successfully. This might occur due to a very large amount of iterations required for the given frequency parameters. Ensure that the number of iterations at least exceeds 3 for accurate results and that the filter is designed successfully before importing the coefficients. Reducing the order helps solve this problem.
4. Ensure that the noise signal has a frequency of less than 20 kHz. This is the limit of human hearing so any signal with frequency above that cannot be perceived by our ears and hence no tangible observation will be made.

**Future Objectives of the Project:**

1. Extension of realised objectives to obtain adaptive filters using algorithms such as least mean squares (LMS) and recursive least squares (RLS) in order to filter out noise with changing frequency.
2. Realisation of channel equaliser using adaptive filters both digitally as well as physically using LABView, Code Compare Studio (CCS-5) and DSK6713.

**References:**

[1] S. Haykin. (1996). Filter Theory. (3rd edition). Prentice Hall.

[2] Gibbs, A.J. 1969 “An Introduction to Digital Filters” Aust. Telecomun. Res, Vol. 3 pp. 3-14, November.

[3] Gibbs, A.J. 1970 “The Design of Digital Filters” Aust. Telecomun. Res, Vol. 4 pp. 29-34, May.

[4] Constantinides, A. G. 1967. ‘’Frequency Transformations for Digital Filters,’’ Electron. Lett., vol. 3 pp.487-489, November

[5] John G. Proakis ‘’Digital Signal Processing”. (2017). Filter Theory. (21st edition). Prentice Hall.

[6] Oppenheim, Alan V., Willsky, Alan S., and Young, Ian T.,1983: Signals and Systems, p. 256 (Englewood Cliffs, New Jersey: Prentice-Hall, Inc.) ISBN 0-13-809731-3

[7]  Rabiner, Lawrence R., and Gold, Bernard, 1975: Theory and Application of Digital Signal Processing (Englewood Cliffs, New Jersey: Prentice-Hall, Inc.)

[8] Miroslav D. Lutovac, Dejan V. Tošić, Brian Lawrence Evans, *Filter Design for Signal Processing Using MATLAB and Mathematica*, Miroslav Lutovac, 2001.

[9] L. D. Paarmann, *Design and Analysis of Digital Filters: A Signal Processing Perspective*, Springer, 2001

[10] Leland B. Jackson, *Digital Filters and Signal Processing*, Springer, 1996.

[11] Design of Digital Filters using FDA tool in MATLAB <https://in.mathworks.com/help/signal/ug/getting-started-with-filter-designer.html>

[12] Waters, W,, and Jarrett , B. 1982. “Bandpass Signal Sampling.” IEEE Trans. On Aerospace and Electronic Systems. Vol. (18)4, pp. 802-816.