

### ABSTRACT

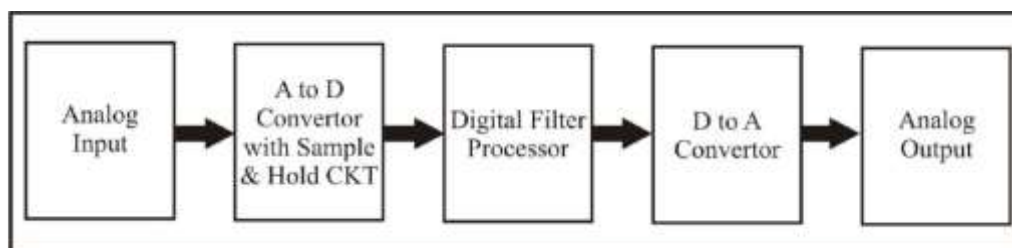
In this paper, we have designed the digital filter based on rectangular Window function using ANN. Firstly we have designed the 10th order digital filter based on rectangular Window function then calculate the coefficient of designed filter. We have normalized the frequency and designed filter at frequency ranges 0.03 to 0.99 Hz. Then we calculated the coefficients of filter at these different frequencies. Some data group of coefficients is used to train the neural network, designed using generalized regression algorithm and rest data are used to test the input to neural network. The output corresponding to test input is approximate equal to frequency corresponding to calculated coefficients using FDA tool. The optimization of this neural network model has been done using generalized regression algorithm.

**KEYWORDS:** Finite Impulse Response (FIR), Low-pass filter (LPF), Cutoff frequency, Multilayer Perceptron (MLP) ANN, GRNN, FDA TOOL.

### INTRODUCTION

A filter is essentially a system or network that selectively changes the wave shape amplitude - frequency and or phase - frequency characteristics of a signal in a desired manner[6]. Common filtering objectives are to improve the quality of a signal, to extract information from signals or to separate two or more signals[4]. The implementation of FIR filter having based on three terms: the cutoff frequency  $\omega_c$ , the filter order (N) and the window size. The order of filter determine the width of transition. Higher order of filter gives the sharp cutoff in frequency response[1]. A digital filter is a mathematical algorithm implementation in hardware and/or software that operates on a digital input signal to produce a digital output signal for the purpose of achieving a filtering objective. Filter designing is the process of transformation of input sequence to obtain desired output sequence. Filter designing process can be described as an optimization problem where each requirement terms to an error function which should be minimized. Digital filter are of two types, finite impulse response (FIR) filter and infinite impulse response (IIR) filter. In this paper the concept of Artificial Neural Network is to find the recursive coefficients that define the filter transfer function. In this research work, DSP and neural networks were combined to produce an excellent algorithm for digital filter design. In the previous work, the FIR filter was designed by least square method and frequency sampling method.

A simplified block diagram of a real-time digital filter, with analog input and output signal is as shown in Figure 1



**Figure. 1. Block Diagram of a Digital Filter**

## ARTIFICIAL NEURAL NETWORKS

Artificial neural networks (ANNs) are non-linear data driven self-adaptive approach as opposed to the traditional model based methods. They are powerful tools for modelling, especially when the underlying data relationship is unknown. ANNs can identify and learn correlated patterns between input data sets and corresponding target values. After training, ANNs can be used to predict the outcome of new independent input data. The common implementation of ANN has multiple inputs, weight associated with each input, a threshold that determine if the neuron should fire, an activation function that determine the output and mode of operation.

**Multilayer Perceptron** The most popular form of neural network architecture is the multilayer perceptron (MLP). A multilayer perceptron has any number of inputs, one or more hidden layers with any number of units. It uses linear combination functions in the input layers, generally sigmoid activation functions in the hidden layers. The Multilayer Perceptron allows establishing decision regions which are much more complex than the two semi planes generated by the Perceptron. The general structure of a neural network has three types of layers that are interconnected input layer, one or more hidden layers and output layer as shown in Figure2.

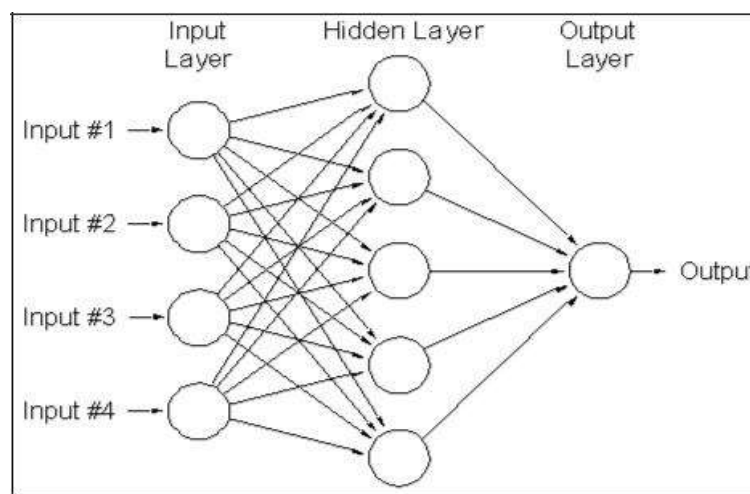


Figure 2. Architecture of MLP

## METHODOLOGY

### DESIGN FIR FILTER USING WINDOW METHOD

To design a filter means to select the coefficients such that the system has specific characteristics. The required characteristics are stated in filter specifications. Most of the time filter specifications refer to the frequency response of the filter. The three most popular design methods are linear phase method, least square method and window method. FIR filters can also be designed from a frequency response specification. The equivalent sampled impulse response, which determines the coefficients of the FIR filter can then be found by inverse (discrete) Fourier transformation. The windowing method requires minimum amount of computational effort; so window method is simple to implement. For the given window, the maximum amplitude of ripple in the filter response is fixed. Thus the stop band attenuation is fixed in the given window, but there are some drawbacks also of this method. The design of fir filter is not flexible. The frequency response of fir filter shows the convolution of spectrum of window function & desired frequency response because of this; the pass band & stop band edge frequency cannot be precisely specified. In this work rectangular window method is used.

### FILTER DESIGN USING FDA TOOLBOX IN MATLAB

In fdatool, first select the type of filter either low pass, high pass, band pass or band stop filter. Then select the filter on the basis of impulse response either IIR filter or FIR filter. Then decide the method for filter designing either equiripple, least square or window method. Then select the window and the order of filter. In frequency specification decide the unit and its value. Then after design the filter. First it shows the magnitude response graph and in analysis it change with many other different parameter. The order of filter for all windows is set to any value. Here the normalized frequency (0 to 1) is used as input and filter coefficients  $h(n)$  as output. Starting input from 0.03 to 0.99, design the filter using fdatool and find the filter coefficients. Due to this range of input the

input output dataset are 33. Export filter coefficients  $h(n)$  to MATLAB workspace. These are dataset of input and output.

**Table 2: Result from fdatool for train input**

| h(n)<br>(Train input) |       |       |       |       |      |       |       |       |       |       | Cutoff<br>frequency<br>(Target) |
|-----------------------|-------|-------|-------|-------|------|-------|-------|-------|-------|-------|---------------------------------|
| h(0)                  | h(1)  | h(2)  | h(3)  | h(4)  | h(5) | h(6)  | h(7)  | h(8)  | h(9)  | h(10) | Fc                              |
| 0.025                 | 0.026 | 0.027 | 0.028 | 0.029 | 0.03 | 0.029 | 0.028 | 0.027 | 0.026 | 0.025 | 0.03                            |
| 0.055                 | 0.056 | 0.057 | 0.058 | 0.059 | 0.06 | 0.059 | 0.058 | 0.057 | 0.056 | 0.055 | 0.06                            |
| 0.085                 | 0.086 | 0.087 | 0.088 | 0.089 | 0.09 | 0.089 | 0.088 | 0.087 | 0.086 | 0.085 | 0.09                            |
| 0.115                 | 0.116 | 0.117 | 0.118 | 0.119 | 0.12 | 0.119 | 0.118 | 0.117 | 0.116 | 0.115 | 0.12                            |
| 0.145                 | 0.146 | 0.147 | 0.148 | 0.149 | 0.15 | 0.149 | 0.148 | 0.147 | 0.146 | 0.145 | 0.15                            |
| 0.175                 | 0.176 | 0.177 | 0.178 | 0.179 | 0.18 | 0.179 | 0.178 | 0.177 | 0.176 | 0.175 | 0.18                            |
| 0.205                 | 0.206 | 0.207 | 0.208 | 0.209 | 0.21 | 0.209 | 0.208 | 0.207 | 0.206 | 0.205 | 0.21                            |
| 0.235                 | 0.236 | 0.237 | 0.238 | 0.239 | 0.24 | 0.238 | 0.238 | 0.237 | 0.236 | 0.235 | 0.24                            |
| 0.265                 | 0.266 | 0.267 | 0.268 | 0.269 | 0.27 | 0.269 | 0.268 | 0.267 | 0.266 | 0.265 | 0.27                            |
| 0.295                 | 0.296 | 0.297 | 0.298 | 0.299 | 0.30 | 0.299 | 0.298 | 0.297 | 0.296 | 0.295 | 0.30                            |
| 0.325                 | 0.326 | 0.327 | 0.328 | 0.329 | 0.33 | 0.329 | 0.328 | 0.327 | 0.326 | 0.325 | 0.33                            |
| 0.355                 | 0.356 | 0.357 | 0.358 | 0.359 | 0.36 | 0.359 | 0.358 | 0.357 | 0.356 | 0.355 | 0.36                            |
| 0.385                 | 0.386 | 0.387 | 0.388 | 0.389 | 0.39 | 0.389 | 0.388 | 0.387 | 0.386 | 0.385 | 0.39                            |
| 0.415                 | 0.416 | 0.417 | 0.418 | 0.419 | 0.42 | 0.419 | 0.418 | 0.417 | 0.416 | 0.415 | 0.42                            |
| 0.445                 | 0.446 | 0.447 | 0.448 | 0.449 | 0.45 | 0.449 | 0.448 | 0.447 | 0.446 | 0.445 | 0.45                            |
| 0.475                 | 0.476 | 0.477 | 0.478 | 0.479 | 0.48 | 0.479 | 0.478 | 0.477 | 0.476 | 0.475 | 0.48                            |
| 0.505                 | 0.506 | 0.507 | 0.508 | 0.509 | 0.51 | 0.509 | 0.508 | 0.507 | 0.506 | 0.505 | 0.51                            |
| 0.535                 | 0.536 | 0.537 | 0.538 | 0.539 | 0.54 | 0.539 | 0.538 | 0.537 | 0.536 | 0.535 | 0.54                            |
| 0.565                 | 0.566 | 0.567 | 0.568 | 0.569 | 0.57 | 0.569 | 0.568 | 0.567 | 0.566 | 0.565 | 0.57                            |
| 0.595                 | 0.596 | 0.597 | 0.598 | 0.599 | 0.60 | 0.599 | 0.598 | 0.597 | 0.596 | 0.595 | 0.60                            |
| 0.625                 | 0.626 | 0.627 | 0.628 | 0.629 | 0.63 | 0.629 | 0.628 | 0.627 | 0.626 | 0.625 | 0.63                            |
| 0.655                 | 0.656 | 0.657 | 0.658 | 0.659 | 0.66 | 0.659 | 0.658 | 0.657 | 0.656 | 0.655 | 0.66                            |
| 0.685                 | 0.686 | 0.687 | 0.688 | 0.689 | 0.69 | 0.689 | 0.688 | 0.687 | 0.686 | 0.685 | 0.69                            |
| 0.715                 | 0.716 | 0.717 | 0.718 | 0.719 | 0.72 | 0.719 | 0.718 | 0.717 | 0.716 | 0.715 | 0.72                            |
| 0.745                 | 0.746 | 0.747 | 0.748 | 0.749 | 0.75 | 0.749 | 0.748 | 0.747 | 0.746 | 0.745 | 0.75                            |
| 0.775                 | 0.776 | 0.777 | 0.778 | 0.779 | 0.78 | 0.779 | 0.778 | 0.777 | 0.776 | 0.775 | 0.78                            |
| 0.805                 | 0.806 | 0.807 | 0.808 | 0.809 | 0.81 | 0.809 | 0.808 | 0.807 | 0.806 | 0.805 | 0.81                            |
| 0.835                 | 0.836 | 0.837 | 0.838 | 0.839 | 0.84 | 0.839 | 0.838 | 0.837 | 0.836 | 0.835 | 0.84                            |
| 0.865                 | 0.866 | 0.867 | 0.868 | 0.869 | 0.87 | 0.869 | 0.868 | 0.867 | 0.866 | 0.865 | 0.87                            |
| 0.895                 | 0.896 | 0.897 | 0.898 | 0.899 | 0.90 | 0.899 | 0.898 | 0.897 | 0.896 | 0.895 | 0.90                            |
| 0.925                 | 0.926 | 0.927 | 0.928 | 0.929 | 0.93 | 0.929 | 0.928 | 0.927 | 0.926 | 0.925 | 0.93                            |
| 0.955                 | 0.956 | 0.957 | 0.958 | 0.959 | 0.96 | 0.959 | 0.958 | 0.957 | 0.956 | 0.955 | 0.96                            |
| 0.985                 | 0.986 | 0.987 | 0.988 | 0.989 | 0.99 | 0.989 | 0.988 | 0.987 | 0.986 | 0.985 | 0.99                            |

**Table 2: Result from fdatool for test input**

| h(n)<br>(Train input) |       |       |       |       |      |       |       |       |       |       | Cutoff<br>frequency<br>(Target) |
|-----------------------|-------|-------|-------|-------|------|-------|-------|-------|-------|-------|---------------------------------|
| h(0)                  | h(1)  | h(2)  | h(3)  | h(4)  | h(5) | h(6)  | h(7)  | h(8)  | h(9)  | h(10) | Fc                              |
| 0.325                 | 0.326 | 0.327 | 0.328 | 0.329 | 0.33 | 0.329 | 0.328 | 0.327 | 0.326 | 0.325 | 0.33                            |
| 0.655                 | 0.656 | 0.657 | 0.658 | 0.659 | 0.66 | 0.659 | 0.658 | 0.657 | 0.656 | 0.655 | 0.66                            |

**Table 5.3: Simulation result from nntool for test input**

| h(n)<br>(Train input) |       |       |       |       |      |       |       |       |       |       | Cutoff<br>frequency<br>(Target) |
|-----------------------|-------|-------|-------|-------|------|-------|-------|-------|-------|-------|---------------------------------|
| h(0)                  | h(1)  | h(2)  | h(3)  | h(4)  | h(5) | h(6)  | h(7)  | h(8)  | h(9)  | h(10) | Fc                              |
| 0.325                 | 0.326 | 0.327 | 0.328 | 0.329 | 0.33 | 0.329 | 0.328 | 0.327 | 0.326 | 0.325 | 0.3464                          |
| 0.655                 | 0.656 | 0.657 | 0.658 | 0.659 | 0.66 | 0.659 | 0.658 | 0.657 | 0.656 | 0.655 | 0.6672                          |

In table1, there is input output dataset for training the neural network. These dataset are derived from fdatool when designing the FIR filter. There are total 31 dataset in table 5.1. In the left hand side filter coefficients as a input data and in right hand side there is cut off frequency as in fdatool data. In table 2, there is input output dataset for testing the neural network. These dataset are also derived from fdatool. In table 3, there is simulation result from nntool which is closed to the result from fdatool.

## RESULT

According to the table 1, there are 33 samples of input output dataset. These dataset are derived from *fdatool*. In the neural network, this input output dataset working as the training dataset. According the Table 2, (Result from *fdatool*) For the test input  $h(n)$ , the output  $fc$  is 0.33 & 0.66. This result is deduced from filter designing tool. According to the table 3: (Simulation result from *nntool*) the output for the same test input  $h(n)$  is 0.3464 & 0.6672. This result is taken from neural network tool. So the results from *nntool* and *fdatool* are nearly same, and by this process it can easily estimate the cut off frequency of any fir filter. In this experiment the accuracy is about 98.9%.

## CONCLUSION & FUTURE WORK

In this work, the design of 1-D FIR filter using different ANN model. The same idea can also be implemented for 2-D FIR filter. In this work, we used rectangular window technique, we can also use to different type of window technique is used to denoise the ECG signal. The maximum accuracy in our work is about 99%. So for further research the accuracy can quite improve.

## REFERENCES

- [1] Haykins S. *Neural Networks - A comprehensive foundation*. New Delhi: Prentice -Hall of India Private Limited; 2003.
- [2] Aggarwal N., Thapar S., Kaur P. Design a low pass FIR low pass filter using particle swarm optimization Based Artificial Neural network, *IJETTC*. 2012; 1(4).
- [3] Singh J., Singh C. Design of low pass FIR filter by using (GRNN), *IJARCSSE*. 2013; 3(10). a. Proakis J.G., Manolakis D.G. *Digital Signal Processing – Principle, Algorithms and Applications*. New Delhi: Prentice Hall of India; 2000.
- [4] S. Moein “FIR Cutoff Frequency Calculating for ECG Signal Noise Removing Using Artificial NeuralNetwork” Verlag Berlin Heidelberg, pp.124-131( 2010)
- [5] Mohan Das Choudhary and Ravindra Pratap Narwaria “ suppression of noise in ECG signal low pass IIRfilters” *IJECSE*, Vol.1, No. 4 2012, pp 2238-2243
- [6] Ferdjallah M, Barr RE, “Frequency-domain digital filtering techniques for the removal of powerline noise with application to the electrocardiogram” *Comput Biomed Res*. 1990 Oct; 23(5):473-89.
- [7] Navneet Gupta & Ravindra Pratap Narwaria, “FIR filter design using artificial neural network” published in *Research & Reviews: A Journal of Neuroscience (STM Journal)* (2013), ISSN: 2277-6427 pp. 29-34.
- [8] S. Bittoliya & R. P. Narwaria, “Intelligent ECG signal noise removal with moving median filter using neural network” *IMPACT: IJRET* ISSN 2321-8843 Vol. 1, Issue 3, Aug 2013, 77-8
- [9] Aggarwal N., Thapar S., Kaur P. “Design a low pass FIR low pass filter using particle swarm optimization Based Artificial Neural network”, *IJETTC*. 2012; 1(4)
- [10] L. Badri, Development of Neural Networks for Noise Reduction *The International Arab Journal of Information Technology*, Vol. 7, No. 3, July 2010
- [11] S Moein. “Hepatitis Diagnosis by Training A MLP Artificial Neural Network” In: *Worldcomp2008 Conference*, Las Vegas, USA (2008)
- [12] Moein, S.T.: *Advances in Computational Biology: A MLP Neural Network for ECG Noise Removal Based on Kalman Filter*. In: Arabnia, H.R. (ed.). Springer, Heidelberg (2010)
- [13] R.D. Wagh et al., “Noise Removal from Electrocardiogram (ECG) a Comparison Approaches” *IJARCET*) Volume 3 Issue 1, January 2014
- [14] S.Kasar et al, “Performance of Digital filters for noise removal from ECG signals in Time domain” *IJIREICE Vol. 2, Issue 4, April 2014*
- [15] S.Moein “Intelligent ECG Signal Noise Removal using PSONN” *IJCA(0975 – 8887)* Vol.45 No.6, May 2012
- [16] K.D. Chinchkhede “On the implementation of FIR Filter with various window for enhancement of ECG signal”, *IJEST* Vol.3 No.3 March 2011
- [17] Mahesh Chavan, et al. “FIR Equiripple digital filter for reduction of power line interference in the ECG Signal”, *ISPRA* 2008