International Journal of Engineering and Technical Research (IJETR)
ISSN: 2321-0869 (O) 2454-4698 (P) Volume-7, Issue-7, July 2017

FIR Filter Design Using Mixed Algorithm

Vikash Kumar, Mr. Vaibhav Purwar

***Abstract*— A method for the design of non recursive digital low pass FIR filter is proposed using GA. The main focus of the paper is to describe the developed and dynamic method of designing finite impulse response filter with automatic rapid and less error by an efficient genetic and neural approach. GA and Neural are powerful global optimization algorithm introduced in combinational optimization problems. Here, FIR filter is designed using Genetic, Neural approach by efficient coding schemes. The response is studied and implemented by keeping values of fixed order, crossover probability and mutation probability. Some data group of coefficients is used to train the neural network designed using generalized regression algorithm and rest are used as test input to neural network. GA & ANN offers a quick, simple and automatic method of designing low pass FIR filters that are very close to optimum in terms of magnitude response, frequency response and in terms of phase.**

***Index Terms*— Terms—Genetic Algorithm, Artificial Neural Networks, Back propagation, FIR Filter, Optimization, DSP. FDA.**

I. INTRODUCTION

Filters constitute an essential part of DSP. Actually, their extraordinary performance is one of the main reasons which have made DSP so popular. Filter is essentially a system or network that improves the quality of a signal and/or extracts information from the signals or separates two or more signals which are previously combined Nowadays digital filters can be used to perform many filtering tasks are replacing the traditional role of analog filters in many applications.[7].

II. DIGITAL FILTER

Digital Filter is an important part of digital signal processing (DSP) system and it usually comes in two categories: Finite Impulse Response (FIR) and Infinite Impulse Response (IIR). FIR filter is an attractive choice because of the ease of design and stability. By designing the filter taps to be symmetrical about the centre tap position, a FIR filter can be guaranteed to have linear phase. Linear phase FIR filters are also required when time domain features are specified

A. Finite Impulse Response (FIR)

Digital filter is one whose impulse response is of finite duration [7]. The impulse response is "finite" because there is no feedback in the filter. If we put in an impulse (that is, a single "1" sample followed by many "0" samples), zeroes will eventually come out after the "1" sample has made its way in the delay line past all the coefficients. FIR (Finite Impulse Response) filters are implemented using a finite number "n" delay taps on a delay line and "n" computation coefficients to compute the algorithm (filter) function. The above structure is non-recursive, a repetitive delay-and-add format, and is most often used to produce FIR filters. This structure depends upon each sample of new and present value data. The number of taps (delays) and values of the computation coefficients () are selected to "weight" the data being shifted down the delay line to create the desired amplitude response of the filter. In this configuration, there are no feedback paths to cause instability. The calculation of coefficients is not constrained to particular values and can be used to implement filter functions that do not have a linear system equivalent. More taps increase the steepness of the filter roll-off while increasing calculation time (delay) and for high order filters, limiting bandwidth. This can be stated mathematically as:

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

where, y(n) = Response of Linear Time Invariant (LTI) system.
x(k) = Input signal
h(k) = Unit sample response
N = No. of signal samples
FIR filters are simple to design and they are guaranteed to be Bounded Input-Bounded Output (BIBO) stable. By designing the filter taps to be symmetrical about the centre tap position, an FIR filter can be guaranteed to have linear phase response. This is a desirable property for many applications such as music and video processing.

B. Infinite Impulse Response (IIR) Filter

IIR filter is one whose impulse response is infinite [2]. Impulse response is infinite because there is feedback in the filter.
This permits the approximation of many waveforms or transfer functions that can be expressed as an infinite recursive series. These implementations are referred to as Infinite Impulse Response (IIR) filters. The functions are infinite recursive because they use previously calculated values in future calculations to feedback in hardware systems. IIR filters can be mathematically represented as:
M is the number of feed-back taps in the IIR filter and N is the number of feed-forward taps. IIR Filters are useful for high-speed designs because they typically require a lower number of multiply compared to FIR filters. IIR filters have lower side lobes in stop band as compared to FIR filters. Unfortunately, IIR filters do not have linear phase and they can be unstable if not designed properly. IIR filters are very sensitive to filter coefficient quantization errors that occur due to use of a finite number of bits to represent the filter coefficients. One way to reduce this sensitivity is to use a cascaded design.

III. DESIGNING TECHNIQUES OF FIR FILTERS

There are essentially three well-known methods for FIR filter design namely:
(1) The window method
(2) The frequency sampling technique

Vikash Kumar, Department of Electronics & Communication, M.Tech Scholar, Kanpur Institute of Technology, Kanpur, India.
Mr. Vaibhav Purwar, Assistant Professor, Department of Electronics & Communication, Kanpur Institute of Technology, Kanpur, India

(3) Optimal filter design methods

A. *Kaiser window*

Kaiser window is a well known flexible window and widely used for FIR filter design and spectrum analysis, since it achieves close approximation to the discrete pro late spheroidal functions that have maximum energy concentration in the main lobe. With adjusting its two independent parameters, namely the window length and the shape parameter, it can control the spectral parameters main lobe width and ripple ratio for various applications. Side lobe roll-off ratio is another spectral parameter and important for some applications. For beam forming applications, the higher side lobe roll-off ratio means, that it can reject far end interferences better. For filter design applications, it can reduce the far end attenuation for stop band energy. And for speech processing, it reduces the energy leak from one band to another.

B. *Optimal Filter Design Methods*

Optimization is the act of obtaining the best results under given circumstances. Optimization can be defined as the process of finding the condition that gives the maximum or minimum value of the function. If x* corresponds the minimum value of function f(x), the same point also corresponds to maximum value of the function –f(x). Thus optimization can be taken to mean minimization since the maximum of the function can be found by seeking of the negative of the same number.

IV. GENETIC ALGORITHM

A Genetic algorithm (GA) is an optimization technique that is based on the evolution theory. Instead of searching for a solution to a problem in the "state space" (like the traditional search algorithms do), a GA works in the "solution space" and builds new, hopefully better solution based on existing ones. GA operates with a collection of chromosomes, called a population. The population is normally randomly initialized. The population includes

fitter and fitter solution, and eventually it converges, meaning that it is dominated by a single solution. The general idea behind GA is that it builds a better solution by somehow combining the "good" parts of other solutions (schemata theory), just like nature does by combining the DNA of living beings [10]. In GA, different operators are to generate new solutions from existing ones. These operators are based on reproductions, Reproduction operators are crossover and mutation. The size of each chromosome must remain the same for crossover to be applied. Fittest chromosomes are selected in each generation to produce offspring which replace the previous generation. The good individuals remain in the population and reproduce; while the bad individuals are eliminated from the population. Finally the population will consist only of the best individuals fulfilling the design specifications. The genetic algorithm is an artificial genetic system based on the process of natural selection and genetic operators. Genetic algorithm is a heuristic algorithm which tries to find the optimal results by decreasing the value of the objective function.

A. *Initialization*

In the initialization, the first thing to do is to decide the coding structure. Coding for a solution, termed a chromosome in GA, is usually described as a string of symbols from (0,1). These components of the chromosomes are then labeled as genes.

B. *Crossover*

The crossover operator is the most important operator of GA. In crossover, generally two chromosomes, called parents, are combined together to form new chromosomes, called offspring. The parents are selected among existing chromosomes in the population with preference towards fitness so that offspring is expected to inherit good. By one from two parent point crossover method, for a chromosome of length, l, a random number c between 1 and l is first generated. The first child chromosome is formed by appending the last l-c elements of the first parent chromosome to the first c elements of the second parent chromosome. The second child chromosome is formed by appending the last l-c elements of the second parent chromosome to the first c elements of the first parent chromosome. Probability of crossover ranges from 0.6 to6 to0.95

C. *Mutation*

Mutation is another important operator in CGA, though it is usually considered as a background operator. It operates independently on each individual by probabilistic perturbing each bit string. The mutation operator introduces random changes in to characteristic of chromosomes. Mutation is generally applied at the gene level. There is a chance that a gene of a child is changed randomly. Generally the chances of mutation are low. Therefore, the new chromosome produced by mutation will not be very different from the original one. Mutation is a unary operator that is usually applied with a low probability An usual way to mutate used in CGA is to generate a random number v between 1 and l and then make a random change in the vth element of the string with probability pm€(0, 1) Typically, the probability for bit mutation changes from 0.001 to 0.01

Parent#1 1 0 0 1 0 1 0 01 0 Child#1 0 1 0 1 0 1 0 0 1 0

Parent#2 0 1 0 1 1 1 1 0 1 1 Child#2 1 0 0 1 1 1 1 01 1

Parent 1 1 0 1 0 1 0 0 1 0 Child 1 1 0 1 0 1 0 1 1 0

Figure 1: One-point Crossover and Mutation operators

D. *Genetic Algorithm Procedure*

The genetic algorithm loops over an iteration process to make the population evolve [12]. It consist the following steps:

1. The first step consists in selecting individuals for reproduction. This selection is done randomly with a probability depending on the relative fitness of the individuals so that best ones are often chosen for reproduction than poor ones.

2. Reproduction: In the second step, offspring are bred by the selected individuals. For generating new chromosomes, the algorithm can use both recombination and mutation.

3. Evaluation: Then the fitness of the new chromosomes is evaluated.

4. Replacement: During the last step, individuals from the old population are killed and replaced by the new ones. The algorithm is stopped when the population converges towards the optimal solution.

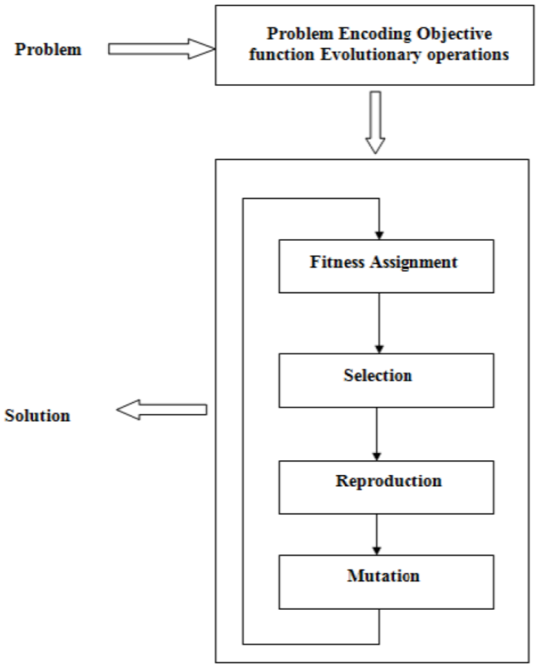

Figure 2 Flow Chart of GA

E. Application of Genetic Algorithm to FIR Filter Design

A digital FIR filter is characterized by the following transfer function,

$$H(z) = \sum_{n=0}^{N-1} h(n) z^{-n}$$

In the above expression, N is the order of the filter and h(n) represent the filter coefficients to be determined in the design process. Designing the FIR filters as minimum phase provides some important advantages. Minimum phase filters have two main advantages: Reduced filter length and Minimum group delay. Minimum phase filters can simultaneously meet delay and magnitude response constraints yet generally require fewer computations and less memory than linear phase. Recently, GA has been emerged into optimum filter designs. The characteristics of multi-objective, coded variables and natural selection make GA different from other optimization techniques. Filters designed by GA have the potential of obtaining near global optimum solution [13]. FIR digital filter has a finite number of nonzero entries of its impulse response such as h[n], n=0,1,…,N. Generally assume implicitly that h[n]≠ 0 , h[0] ≠ 0. The transfer function of the FIR filter is given in eq. (2) and the frequency response of form is:

$$H(e^{j\omega}) = \sum_{n=0}^{N-1} h(n) e^{-j\omega n}$$

Consider the ideal frequency response Hd(ejω) with the samples divided into equal frequency interval, Thus we can get,

$$H_d(e^{j\omega})|\omega = 2\pi k/N = H_d(k)$$

where, Hd(k) is regarded as the frequency response of the filter to design. Equation (4) can be rewritten as

$$H_d(k) = H_d(e^{j\omega})|\omega = 2\pi k/N, k=0,1,.....N-1$$

To design a linear phase FIR filter, we must minimize the error between actual and ideal output. There exist some forms of error function for the filter design. One of them is the least-squares method. We define the error function as the error between the desired magnitude and the actual amplitude at a certain frequency, that is

$$E(e^{j\omega}) = H_d(e^{j\omega}) - H(e^{j\omega})$$

Thus we can adopt the objective function for the minimization as total squared error across frequency domains as follows

$$E(e^{j\omega}) = \sum_{i=1}^{M} [\,|H_d(e^{j\omega_{\omega i}})| - |H(e^{j\omega_{\omega i}})|\,]^2$$

where, M is the number of frequency interval. From eq. (3) we can write the above equation as:

$$E(e^{j\omega}) = \sum_{i=1}^{M} [\,|H_d(e^{j\omega_{\omega i}})| - |\sum_{n=0}^{N-1} h(n) e^{-j\omega_{\omega i} n}|\,]^2$$

The problem is reduced to find out h (n) by minimizing the squared error E.

F. Coefficient Encoding

The filter impulse response coefficients, h (0) to h (N), are sufficient to represent a digital FIR filter. Thus, N+1 coefficients of the filter form the genome and the particle position in the GA and the PSO, respectively. Each coefficient is represented by a floating number in the range [-1, 1], inclusive. G. Fitness Function A fitness function is a particular type of objective function that is used to summarize, as a single figure of merit. Fitness function must be devised for each problem to be solved. Given a particular chromosome, the fitness function returns a single numerical fitness, "figure of merit," which is supposed to be proportional to the "utility" or "ability" of the individual which that chromosome represents. We use the total squared error as the fitness function of FIR digital filter, that is:

$$E(e^{j\omega}) = \sum_{i=1}^{M} [\,|H_d(e^{j\omega_{\omega i}})| - |\sum_{n=0}^{N-1} h(n) e^{-j\omega\omega_{in}}|\,]^2$$

V. ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network (ANN) also known as "Neural Network (NN)" is a computational model based on the structure and function of biological neural network [1]. In other words ANN is computing system which is made up of a number of simple processing elements (the computer equivalent of neurons, Nodes) that are highly interconnected to each other through synaptic weights. The number of nodes, their organization and synaptic weights of these connections determine the output of the network. ANN is an adaptive system that changes its structure/weights based on given set of inputs and target outputs during the training phase an produces final outputs accordingly. ANN is particularly effective for predicting events when the network have a large database of prior examples to draw. 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. 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 Figure 3.

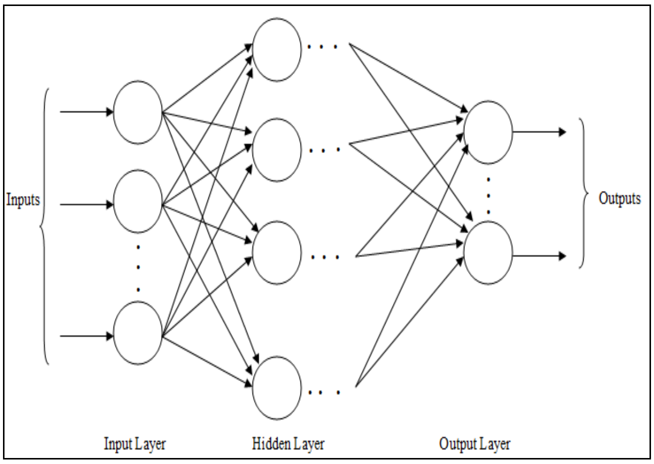

Figure 3: General Structure of Neural Network

There are some algorithms that can be used to train an ANN such as: Back Propagation, Radial-basis Function, an Support Vector learning, etc. The Back Propagation is the simplest but it has one disadvantage that it can take large number of iterations to converge to the desired solution [3]. In Radial Basis Function (RBF) network the hidden neurons compute radial basis functions of the inputs, which are similar to kernel functions in kernel regression. Speech has popularized kernel regressions, which he calls a General Regression Neural Network (GRNN) [3]. General Regression Neural Network (GRNN) is a variation of Radial Basis Function (RBF) network that is based on the Nadaraya – Watson kernel regression. The main features of GRNN are fast training time and it can also model nonlinear function. GRNN being firstly proposed by Sprecht in 1991 is a feed forward neural network model base on non linear regression theory. It approximates the function through activating neurons. In GRNN transfer function of hidden layer is radial basis function.

$$y'i = \frac{\sum_{i=1}^{n} yi * exp - D(x - xi)}{\sum_{i=1}^{n} exp - D(x - xi)}$$

$$D(x - xi) = \sum_{k=1}^{m} (\frac{xi - xik}{\sigma})^2$$

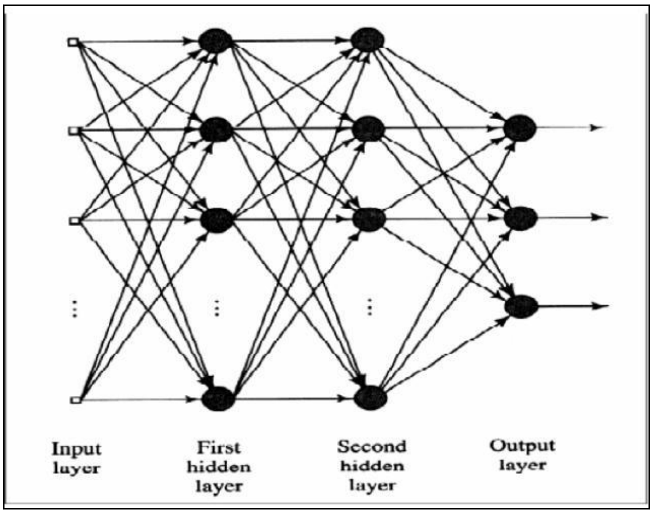

Figure 4: Feed forward back propagation neural network

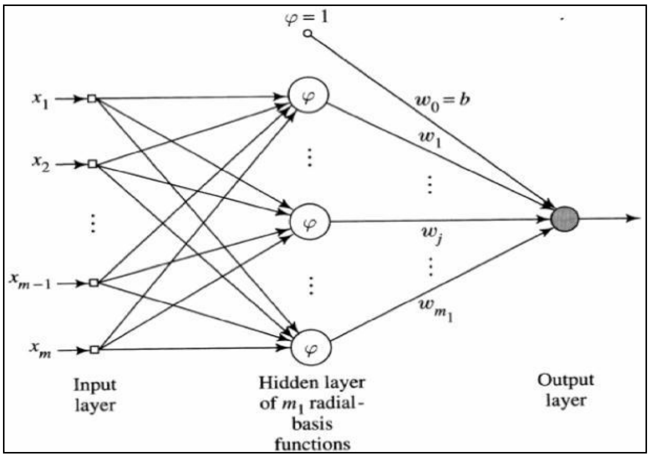

Figure 5: Radial basis function

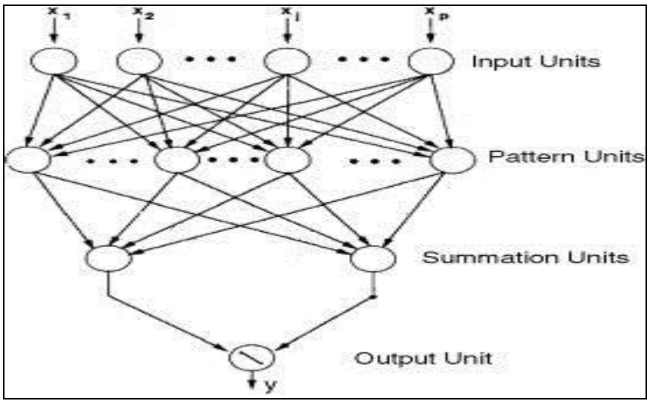

Figure 6: Generalized regression neural network

VI. PROPOSED WORK

There is a method for designing low pass Finite Impulse Response filter with ideal magnitude response, small phase variation, small pass band ripple, high attenuation in stop band and minimum transition bandwidth.

Filter Type	Low Pass Filter
Generation Number	200
Mutation Ratio	0.25
Pass Band Cut off Frequency	2500(HZ)
Stop Band Cut off Frequency	3000(HZ)
Pass Band Ripple	0.015
Stop Band Ripple	0.15
Sampling Frequency	12000(HZ)

Table 1: Initial conditions for designing low pass fir filter

Ideal Low pass filter passes all the signals that are below the cut off frequency and stop all others.

Here, there is a flat pass band below pass band frequency (ωP) =2500 Hz and flat attenuation band above stop band frequency (ωs) =3000 Hz. Here we have applied Genetic Algorithm with two parents and three parents separately on filter response which is obtained by using Kaiser Window. Then the results are studied and compared .When we are using only two parents, we get the magnitude response versus frequency curve as shown in Fig.7, Fig.9. But, when we are using three parents, we get a better magnitude response versus frequency curve as shown in Fig.8, Fig.10.

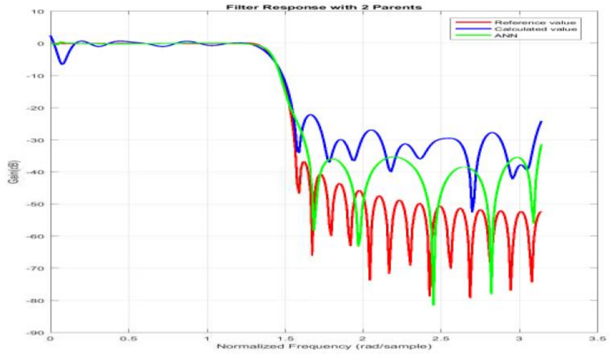

Figure 7: Magnitude Response of FIR Filter using two Parents at 200 generations with 3 attempts

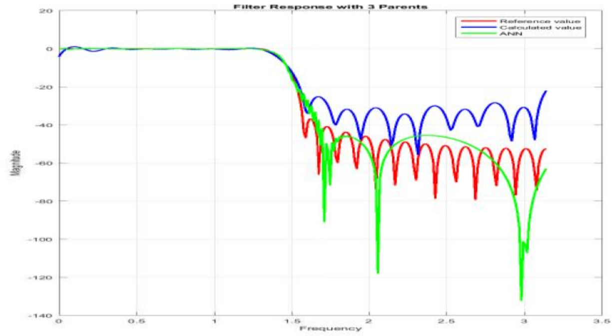

Figure 8: Magnitude Response of FIR Filter using three Parents at 200 generations with 3 attempts

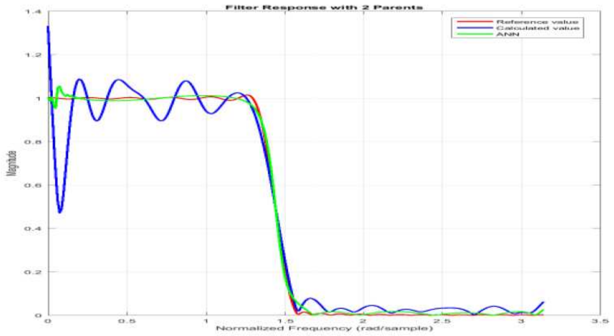

Figure 9:Magnitude Response of FIR Filter using two Parents at 200 generations with 3 attempts

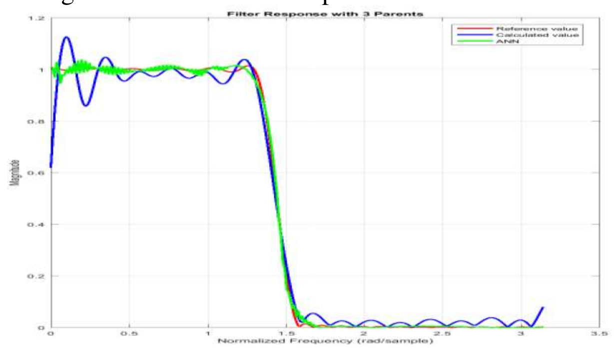

Figure 10: Magnitude Response of FIR Filter using three Parents at 200 generations with 3 attempts

VII. CONCLUSION

The proposed technique achieves the optimum number of coefficients required to get the desired frequency response with the optimum word length. In this present work, FIR filter is designed using Kaiser Window, GA and ANN. The response is studied by keeping values of fixed order, crossover probability and mutation probability. From the outputs obtained it is clear that GA offers a quick, simple and automatic method of designing low pass FIR filters that are very close to optimum in terms of magnitude response, frequency response and in terms of phase variation. From the outputs obtained it is clear that more better results in ANN offers a quick, simple and automatic method of designing low pass FIR filters that are very close and ripple free to optimum in terms of magnitude response, frequency response and in terms of phase variation.

A technique of using three parents using Kaiser Window, GA and ANN has been proposed and outputs are compared with the outputs obtained using two parents using Kaiser Window, GA and ANN. We have obtained various outputs by changing the generations and attempts. It has been observed that a better response is achieved when three parents are used instead of two. Best response is obtained in figure (7),(8),(9) and (10),where 200 generations are taken with three attempts. With the help of GA, the number of operations in design process is reduced and coefficient calculation is easily done.

REFERENCES

[1] S. Haykins, “Neural Networks –A comprehensive foundation”, Prentice –Hall of India Private Limited, New Delhi, (2003).

[2] Neha Aggarwal, Sheenu Thapar,Parmindar kaur “ design a low pass FIR low pass filter using particle swarm optimization Based ArtificialNeural network” (IJETTCS) vol 1 Issue 4 ,December 2012.

[3] Jasdeep singh,charanjit singh Design of Low pass FIR filter by using (GRNN) (IJARCSSE) vol 3 issue 10 October 2013

[4] Avci K., and Nacaroglu A., “A New Window Based on Exponential Function”, Proc. Of 7.th International Conference COMMUNICATIONS 2008, 63-66, Bucharest, Romania, 2008.

[5] Saramaki, T., “A class of window functions with nearly minimum side lobe energy for designing FIR filters”, inproc. IEEE Int. Symp.Circuits and systems (ISCAS’89).359-362, Portland, Ore, USA, 1989.

[6] Kaiser J.F., and Schafer, R.W., “On the use of the I0-sinhwindow for spectrum analysis”, IEEE Trans. Acoustic, Speech, and Signal Processing, 28(1), 105-107, 1980.

[7] Kaiser, J.F., “Non recursive digital filter design using I0-sinh window function”, in proc. IEEE Int. Symp. Circuits and systems (ISCAS’74). 20-23, San Francisco, Calif, USA, 1974.

[8] Bergen S. W. A., and Antoniou A. “Design of Ultra spherical Window Functions with Prescribed Spectral Characteristics” EURASIP Journal on Applied Signal Processing , no.13, pp.2053-2065, 2004.

[9] “Genetic Algorithms in search, optimization and Machine Learning” By David E. Goldberg.

[10] Teaching Genetic Algorithm using MATLAB, Y.J. Cao and Q.H. Wu. Int. J. Elect. Enging. Educ., Vol. 36, pp. 139–153. Manchester U.P.,1999. Printed in Great Britain

[11] Ajoy Kumar Dey, “A Method of Genetic Algorithm (GA) for FIR Filter Construction”, Vol. 1, pp. 87-90, Dec 2010

[12] Design of FIR Filter using Genetic Algorithm, Samrat Banerjee, Sriparna Dey, Supriya Dhabal, IJETR, Volume-3, Issue-6, June 2015

[13] T. Saramaki, “Finite impulse response filter design”, Handbook for Digital Signal Processing, S. K. Mitra and J. F. Kaiser, Eds.,Wiley, New York, NY, USA, (1993).

[14] D. F. Spechtand P. D. Shapiro, “Training speed comparison of probabilistic neural networks with back-propagation networks”, Proc. Int. Neural Network Conf.(Paris, France), vol. 1,(1990) July, pp. 440-443.

[15] P. Burrascano, “Learning vector quantization for the probabilistic neural network”, IEEE Trans. Neural Networks, vol. 2,(1991)July, pp. 458-461.

[16] H.B. Celikoglu,“Application of radial basis function and generalized regression neural networks in non-linear utility function specification for travel mode choice modelling”, Math Compute Model, vol. 44, (2006), pp. 640-58.

Vikash Kumar, Department of Electronics & Communication, M.Tech Scholar, Kanpur Institute of Technology, Kanpur, India.

Mr. Vaibhav Purwar, Assistant Professor, Department of Electronics & Communication, Kanpur Institute of Technology, Kanpur, India