Advanced Research in Electrical and Electronic Engineering
p-ISSN: 2349-5804; e-ISSN: 2349-5812 Volume 3, Issue 3, April-June, 2016, pp. 219-222

http://www.krishisanskriti.org/Publication.html

Low Pass Raised Cosine FIR Filter Design using Artificial Neural Network

Vipin Chaturvedi[1] and Vandana Vikas Thakare[2]

[1,2]Department of Electronics & Communication, MITS GWALIOR
E-mail: [1]chaturvedivipin1990@gmail.com, [2]vandanavt_19@rediffmail.com

Abstract—*This paper is intended to introduce a novel approach for designing of FIR filter design by using various neural network models. In this paper we designed raised-cosine FIR filter using hamming window technique and three different neural network models. We used Feed forward back propagation (FFBP), Non Linear Auto Regressive (NARX), Layer Recurrent (LR) neural network models. The coefficients of filter are obtained by varying the normalized cutoff frequency. The information of filter coefficients is used to train and test the neural network whereas the information of cutoff frequencies is used as target. Here, we compared the performance of three neural networks models and picked the model having minimum mean square error (MSE). The results indicate that layer recurrent neural network gave the more improved performance.*

1. INTRODUCTION

Digital filters design is basic but dynamic topic of research in the field of digital signal processing (DSP). For two decades, many researchers are working on the optimal designing of digital filters.

Filters are the wave shaping electronic circuits or systems that respond to impulses, processing signals by boosting certain characteristics and/or reducing certain characteristics. A digital filter is algorithm implemented in both hardware and software that operates on digital input signal to produce digital output signal. Depending on the output of the digital filters, they are mainly classified into two types: Finite impulse response (FIR) filter is one in which the response goes to zero in finite amount of time. Whereas, an Infinite impulse response (IIR) filter is one whose output is not going to zero in finite interval of time.

The FIR filter is also known as non-recursive filter as the feedback is not used by them i.e. these filters use current sample as input. The IIR filter is known as recursive filter as the internal feedback is used by them i.e. these filters use previous output as input to next stage.

In this paper we preferred FIR filter over IIR filter because FIR filter has linear phase characteristics in the pass-band, constant phase and group delay.

The conventional method of designing FIR filter is: Window method, frequency sampling method, weighted least square method (WLS) [1-2]. Now days we have numerous techniques for designing of digital filters such as: Artificial neural network (ANN) [3-4], Genetic algorithm [5-6], Particle swarm optimization [7] etc. This paper presents the approach for designing of FIR filter using nonlinear auto regressive (NARX), feed forward back propagation (FFBP) and layer recurrent neural network. The hamming window technique is used to compute the filter coefficients. The hamming window function expressed as:

$$w(n) = 0.54 - 0.46\cos\left(\frac{2\pi n}{N}\right),\ 0 \le n \le N$$

In section II, Raised cosine FIR filter is defined. Then in section III, The introduction of artificial neual network and three different neural networks are described. The problem formulation are explained in section IV, the simulation results are displayed in section V and finally the conclusion is stated in section VI.

2. RAISED-COSINE FIR FILTER

In digital signal processing, raised cosine filter is widely used as pulse shaping filter due to its capability to reduce intersymbol interference(ISI) to great extent. The two important parameters of raised cosine filter are: Roll-off factor (β) and symbol rate($\frac{1}{Ts}$) .

The impulse response of raised cosine fir filter is expressed as:

$$h(t)=\begin{cases} \frac{1}{\sqrt{Ts}}\left(1-\beta+4\frac{\beta}{\pi}\right), t=0 \\ \frac{\beta}{\sqrt{2Ts}}\left[\left(1+\frac{2}{\pi}\right)\sin\left(\frac{\pi}{4\beta}\right)+\left(1-\frac{2}{\pi}\right)\cos\left(\frac{\pi}{4\beta}\right)\right], t=\pm\frac{Ts}{4\beta} \\ \frac{1}{\sqrt{Ts}}\frac{\sin\left[\pi\frac{t}{Ts}(1-\beta)\right]+4\beta\frac{t}{Ts}\cos\left[\pi\frac{t}{Ts}(1+\beta)\right]}{\pi\frac{t}{Ts}\left[1-\left(4\beta\frac{t}{Ts}\right)\hat{}2\right]}, \text{otherwise} \end{cases}$$

3. ARTIFICIAL NEURAL NETWORK

An artificial neural network (ANN) is massively connected parallel network of nonlinear computational elements known as neurons. The whole neural network is classified into three layers: input layer, hidden layer and output layer. Input layer is to accept the inputs for the network. Output layer creates the output from the network. Hidden layer states are not available for user and it computes the intermediary functions. In neural network the neurons are represented by nodes whereas the connection between neurons is symbolized by weighted edges.

Artificial neural network is an robust system that can modify its structure or weights depend on predetermined set of input and target outputs at the time of training phase on creates final output. The neural network is especially adequate for forecast events when the network has huge database of earlier examples.

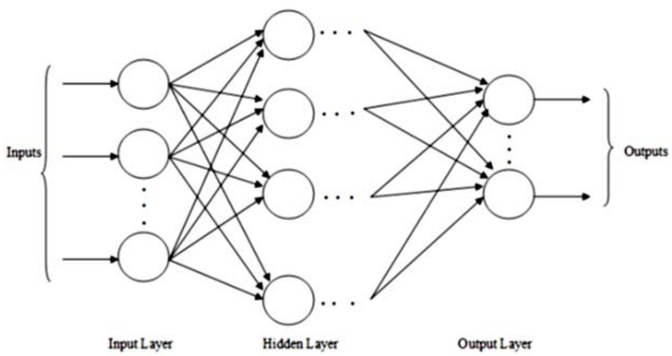

Fig. 1: General Structure of Neural Network

To train the neural network there are various algorithms available. Feed forward back propagation (FFBP), nonlinear auto regressive (NARX) and layer recurrent (LR) are some of them.

3.1. Feed Forward Back Propagation Neural Network

The feed forward back propagation neural network is simplest type of neural network model. In this model input flow only in forward direction through hidden layer to output layer, here the back propagation suggests that the errors propagate in backward direction. The output of each neuron is the combination of the numbers of neurons of the preceding level multiplied by its equivalent weights. The input signals are converted into output signals by calculating the weighted sum of inputs [8].

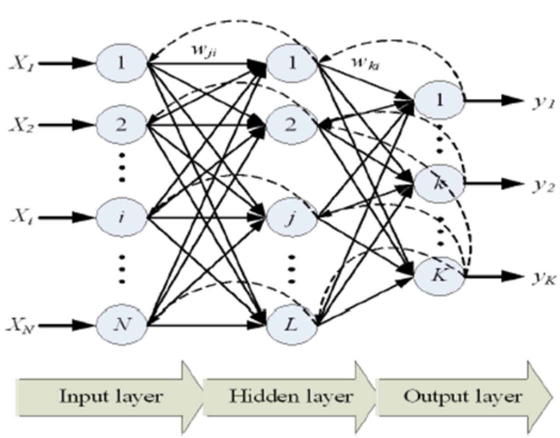

Fig. 2: Feed Forward Back Propagation Neural Network

3.2. Nonlinear Auto Regressive Neural Network

The nonlinear auto regressive neural network is dynamic network with feedback connections including various layers. It has sole input and single output, with the delay line on input and the output is applied back to the input by other delay line. This neural network is very effective for time series prediction [9]. The equation for NARX neural network is:

$$y(t)=f(y(t-1),y(t-2),..,y(t-n_y),x(t-1),x(t-2),..,x(t-n_x))$$

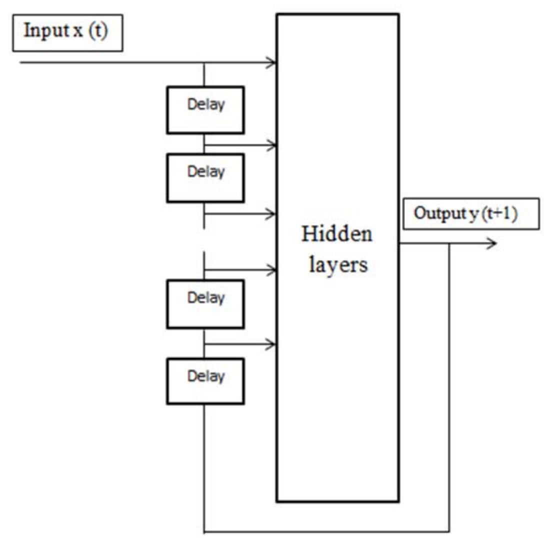

Fig. 3: Nonlinear Auto Regressive Neural Network

3.3. Layer Recurrent Neural Network

The layer recurrent neural network comprises at least one feed-back connection with single delay around every layer of network except the last layer, so the weighted sum of inputs can flow round in a loop that are used to enables the network for sequences learning and temporal processing. Layer recurrent neural network maps the all previous inputs to every output.

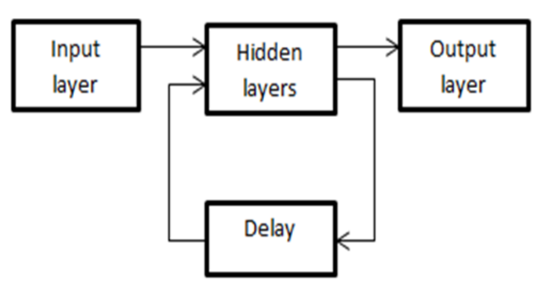

Fig. 4.Layer Recurrent Neural Network

4. DESIGN METHODOLOGY

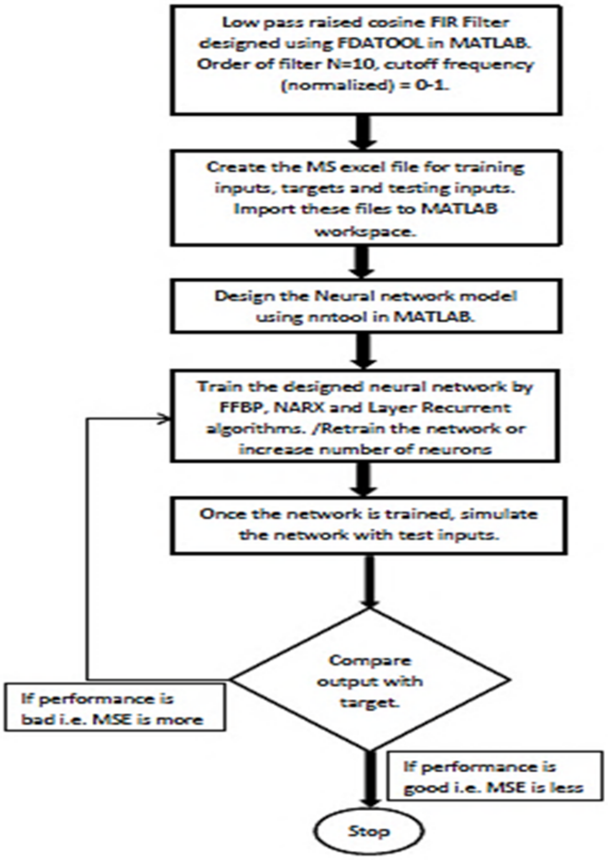

Fig. 5: Flow Chart for Design Methodology

5. SIMULATION RESULTS

After training the network is gone through testing using ten values of filter coefficients out of 42 values obtained from raised cosine FIR filter design using FDATOOL in MATLAB using hamming window.

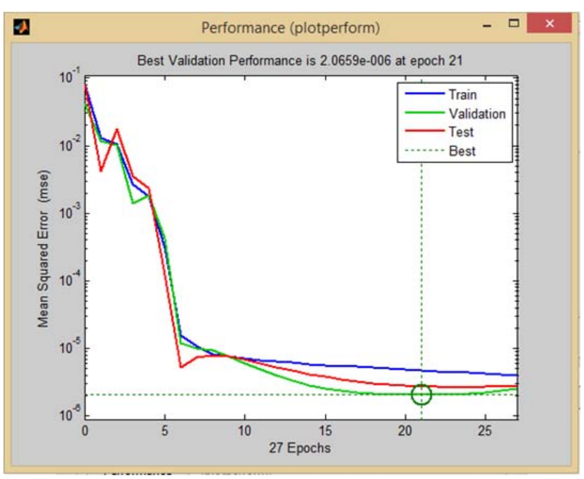

Fig. 6: Performance Plot of Layer Recurrent Neural Network

Fig. 6 displays the performance of layer recurrent neural network using NNTOOL in MATLAB and Fig. 7 shows the simulation result of layer recurrent neural network.

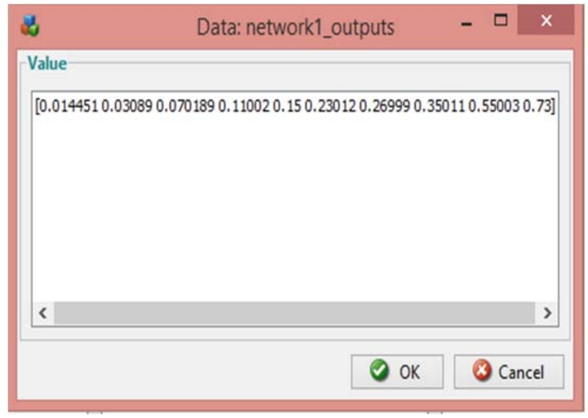

Fig. 7: Simulation Results of Layer Recurrent Neural Network

Table 1: Filter Coefficient Using FDA Tool and NN Tool

filter coefficients h(n)	Hamming window	Artificial Neural Network		
		FFBP	NARX	LAYER RECURRENT
h(0)	0.01	0.02099	0.02130	0.01445
h(1)	0.03	0.02857	0.03059	0.03089
h(2)	0.07	0.07154	0.07022	0.07018
h(3)	0.11	0.10938	0.10994	0.11002
h(4)	0.15	0.15012	0.14993	0.15
h(5)	0.23	0.22903	0.22998	0.23012
h(6)	0.27	0.27051	0.26995	0.26999
h(7)	0.35	0.35038	0.34992	0.35011
h(8)	0.55	0.55009	0.55004	0.55003
h(9)	0.73	0.73047	0.72999	0.73

Table 2: Square Error

Square Error		
FFBP	NARX	LAYER RECURRENT
0.00012	0.000127	0.0000198
0.00000202	0.000000348	0.000000792
0.00000237	0.0000051	0.0000000357
0.000000384	0.0000000036	0.0000000004
0.0000000144	0.0000000049	0.00
0.00000094	0.0000000004	0.0000000144
0.00000026	0.0000000025	0.0000000001
0.000000144	0.0000000064	0.0000000121
0.0000000081	0.0000000016	0.0000000009
0.00000022	0.0000000001	0.00

Table 1 shows the filter coefficients using FDA tool and three different types of neural network models using NN tool. Whereas table 2 displays the square error of three neural network models which is calculated using square error = $(\text{target-output})^2$. Where target is filter coefficients using hamming window and output is the output of different neural network models. The mean square error for three different neural network models is:

Table 3: Mean Square Error

Mean Square Error	
FFBP	1.26×10^{-5}
NARX	1.32×10^{-5}
LAYER RECURRENT	2.06×10^{-6}

From table 3 it is clear that the mean square error for layer recurrent neural network is less as compared to feed forward back propagation neural network and nonlinear auto regressive neural network.

6. CONCLUSION

In this paper, the comparative study for designing of linear phase raised cosine fir filter using three different types of artificial neural network namely is proposed. Through comparison we come to know that performance of layer recurrent neural network is better than nonlinear auto regressive neural network and feed forward back propagation. The outputof layer recurrent neural network is close to the target values i.e. filter coefficients using hamming window [10].

REFERENCES

[1]V.R. Alagzi, and M. Suk, "On The Frequency Weight Least Squares Design of Finite Duration Filters", IEEE trans. Circuits Syst., vol.22, no.12, pp.943-953, Dec. 1975.

[2] Y.C. Lim, J.H. Lee, C.K. Chen, and R.H. Yang, "A Weight Least Squares Algorithm for Quasi-Equiripple FIR and IIR Digital Filter Design", IEEE Trans. Signal Processing, vol. 40, no. 3, pp. 551-558, Mar. 1992.

[3] P. Sharma and R. P. Narwaria, "Comparison of FIR Filters using Neural Network and FDA Tool",International Journal of Advanced and Innovative Research (IJAIR), vol. 2, no. 8, pp. 234-239, Aug. 2013.

[4] HarpreetKaur and BalwinderDhaliwal,"Design of Low Pass FIR Filter using Artificial Neural Network",International Journal of Information and Electronics Engineering (IJIEE), vol. 3, no. 2, Mar. 2013.

[5] D. Suckley, "Genetic algorithm in the design of FIR filters", IEE Proceedings Circuits, Devices and Systems, vol. 138, pp. 234-238, Apr. 1991.

[6] S. Thapar, "A low pass FIR filter design using genetic algorithm based artificial neural network",International Journal of Computer Technology and Electronics Engineering (IJCTEE), vol. 2, no. 4, Aug. 2012.

[7] AmanpreetKaur, "Design of FIR Filter Using Particle Swarm Optimization Algorithm for Audio Processing", International Journal of Computer Science and Network (IJCSN), vol. 1, no. 4, Aug. 2012.

[8] Zhen-GuoChe, Tzu-An Chiang, and Zhen-HuaChe, "Feed-Forward Neural Networks Training: A Comparison between Genetic Algorithm and Back-Propagation Learning Algorithm", International Journal of Innovative Computing, Information and Control, vol. 7, no. 10, Oct. 2011.

[9] Tsungnan Lin, Bill G. Horne, Peter Tino, and C. Lee Giles, "Learning long-term dependencies in NARX recurrent neural networks", IEEE Trans. on Neural Networks, Vol.7,No.6, 1996, pp. 1329-1351.

[10] M. A. Singh, and V. B. V. Thakare, " Artificial Neural Network Use for Design Low Pass FIR Filter a Comparison", International Journal of Electronics and Electrical Engineering Vol. 3, No. 3, June 2015.

[11] H. Zhao, and J. Yu, "A Novel Neural Network-Based Approach for Designing Digital Filters", in Proc. IEEE Int. Symp. Circuits Syst., 1997, pp. 2272–2275.

[12] J. Larsen, "Design of Neural Network Filters", Ph.D. Thesis, Electronics Institute, Technical University of Denmark, 1993.

[13] H. Demuth, and M. Beale, "Neural Network Toolbox for Use with MATLAB", User's Guide, Version 3.0.