

# Design of Digital Recursive Filter Using Artificial Neural Network

Amir A. Bature, and Sunusi S. Adamu

**Abstract-** Digital Signal Processing (DSP) is one of the most powerful technologies that will shape science and engineering in the twenty-first century. One of the key areas among DSP applications is digital filters, whose performances are in most cases better than equivalent analog filters. Various methods of filter design and analysis have been developed over the past three decades. In this work the concept of Artificial Neural Network (ANN), which is also a wide area of research, is used to design one class of digital filters termed Infinite Impulse Response Filters (IIR filters). The main aim of the design is to find the recursive coefficients that define the filters transfer function. Therefore, using the iteration procedure of ANN development, a software using Visual Basic.net is developed for designing IIR filters of any desired response, especially custom filters whose design is usually based on trial and error even in the present days. In this research work, DSP algorithms and neural networks were combined to produce an excellent algorithm for digital filter design. The work recommends further work in this area, which include extending the package to handle any number of filter orders; ability to allow frequency response specifications, and stability analysis routines to provide stability studies to be conducted.

**Key words:** DSP, ANN, IIR, FIR

## I. INTRODUCTION

Digital filters are very important part of *Digital Signal Processing* (DSP). In fact, their extraordinary performance is one of the key reasons that DSP has become so popular. Filter design is the process of developing an algorithm or other transformation of input sequences to obtain desired output sequence (in the sense in which the term is used in signal processing, statistics, and applied mathematics). Filter design process can be described as an optimization problem where each requirement contributes with a term to an error function which should be minimized. Digital filters are of two types, Finite Impulse Response (FIR) filters and Infinite Impulse Response (IIR) filters. *Infinite impulse response*, or *IIR*, filters are the digital counterpart to analog filters. Such a filter contains internal state, and the output and the next internal state are determined by a linear combination of the previous inputs and outputs (in other words, they use feedback, whereas the FIR filters do not). In theory, the impulse response of such a filter never dies out completely, though in practice, this is not true given the finite resolution of computer arithmetic.

Manuscript received October 3, 2011; revised November 29, 2011 Amir Bature is with Bayero University, Kano, PMB 3011 Kano, Nigeria, as an Assistant Lecturer, mobile: +2348037314616; [amirbature@gmail.com](mailto:amirbature@gmail.com). Sunusi Sani Adamu is also with Bayero University, Kano, PMB 3011 Kano, Nigeria; [adamus1664@buk.edu.ng](mailto:adamus1664@buk.edu.ng) as a senior lecturer.

This work uses ANN to implement a recursive filter design program. The ANN results are obtained using steepest decent algorithm. Visual Basic.net will be used to design user friendly windows application software where the desired frequency response of the custom filter will be entered and the program will use neural network algorithm to calculate the coefficients and sketch the filters frequency response as well as the desired frequency response.

## II. ANALYSIS AND DESIGN

The general formula of convolving any filter kernel ( $h(n)$ ) and any input signal for a general filters is

$$y[n] = \sum_{k=0}^{M-1} h[k]x[n-k] \dots 1$$

Whereas for IIR filters since they incorporate feedback is given by

$$y[n] = \sum_{k=0}^{M-1} h[k]x[n-k] = \sum_{k=0}^{M-1} a[k]x[n-k] - \sum_{k=1}^M b[k]y[n-k] \dots 2$$

The  $a[k]$ 's and  $b[k]$ 's defines the filters response. It can be rearrange into;

$$h[n] = \frac{y[n]}{x[n]} = \frac{\sum_{k=0}^{M-1} a[k]x[n-k]}{\sum_{k=1}^M b[k]y[n-k]} \dots \dots 3$$

The filter may also be represented as given in equation 4.

$$X(z) = \sum_{n=0}^{\infty} x[n]z^{-n} \dots \dots \dots 4$$

Where 'z' is a complex variable.

The transfer function of an IIR filter involves the ratio of two polynomials.

Therefore taking the Z-transform of equation 3 yields

$$H(Z) = \frac{Y(Z)}{X(Z)} = \frac{\sum_{m=0}^M a[m]z^{-m}}{\sum_{n=1}^N b[n]z^{-n}} \dots 5$$

This is the general equation of IIR filter. The power  $n$  represents the number of poles of the filter. Therefore the  $a[k]$ 's and  $b[k]$ 's will be from 0-7. The program uses zero base indexes of arrays.

### A. Neural Network Algorithm

The weights of the neural network are assigned to be the filters coefficients. Initially all the coefficients are assigned the value of zero except for  $a(0)$  which is assign a value of one, to make that response identity response, i.e  $\delta(n)$ . If the program is allowed to choose random numbers for first iteration, then the filter can become unstable. Therefore as the iteration procedure is carried out, the weights are changed accordingly by the *steepest decent* algorithm and each time the response is compared with the desired response.

The steepest descent procedure, which is a gradient search procedure, is as follows:

Denoting a weights vector setting after the  $w$ 'th iteration (the  $m$ 'th training set) as  $w(m)$ , then:

$$w(m+1)=w(m)+\Delta w(m) \dots\dots\dots(6)$$

Where  $\Delta w$  is the change (variation) in  $w(m)$ , this variation being given by:

$$\Delta w(m) = \mu \nabla J_{w(m)} \dots\dots\dots (7)$$

$\mu$  is the rate parameter; and

$$\nabla J = \left[ \frac{\partial J}{\partial w_1} \dots \frac{\partial J}{\partial w_n} \right]^T \dots\dots\dots(8)$$

The steepest descent procedure to update  $w(m)$  of Eq. (1.4) follows the following steps:

Step 1: Apply input vector  $x_m$  and the desired output  $d_m$  for the  $m$ th training set.

Step 2: Determine  $e_m^2$ , where

$$e_m^2 = [d_m - w(m)x(m)]^2$$

$$= d_m^2 - 2d_m w^T(m)x(m) + w^T(m)x(m)x^T(m)w(m) \dots\dots(9)$$

Step 3: Evaluate:

$$\nabla J = \frac{\partial e_m^2}{\partial w(m)} = 2x(m)w^T(m)x(m) - 2d_m x(m)$$

$$= -2[d(m) - w^T(m)x(m)]x(m) = -2e_m x(m) \dots\dots(10)$$

thus obtaining an approximation to  $\nabla J$  by using  $e_m^2$  as the approximate to  $J$ , namely

$$\nabla J \cong -2e_m x(m) \dots\dots\dots(11)$$

Step 4: Update  $w(m+1)$  via Eqs. (6), (7), namely

$$w(m+1)=w(m)-2 \mu e_m x(m) \dots\dots\dots(12)$$

This is called the Delta Rule of ANN [4].

The error is then summed and use in the algorithm to mold the coefficients in to desired response. The Pseudocode for the program as follows.

- Input desired response in array form
- Make all coefficients zero, except  $a(0)$  which is equal to one.
- Compute the FFT of the identity system
- Convert the values to magnitude form
- Find the difference with the desired response as the error
- Sum up the error and calculate the mean square error.
- Calculate the next iteration step size based on the error
- Repeat the iteration procedure again until the error is negligible.
- Print coefficients and plot response.

### Flowchart

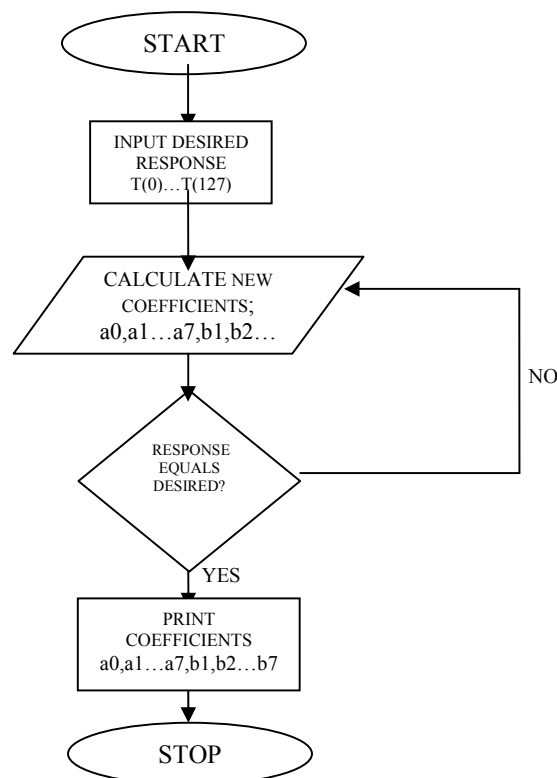


Fig 1. Flowchart Of IIR Filter Design Software

### III. IMPLEMENTATION

IIR filter software was designed using Visual Basic.Net. It's a windows based software using only one form and some basic windows forms tools.

The software accepts desired frequency response in an array of numbers containing the amplitude of the desired response from 0 to 0.5 which is the normalize frequency from 0 to the maximum frequency of the input signal (Nyquist frequency). The input is through a datagrid which is connected to Microsoft Access database where pre-define standard frequency responses (low pass, high pass, band pass, and band reject and custom filter) are stored. Microsoft Access database is connected to the program using the Visual Basic.net OLEDB data adapter and connection string. The copy of the data is then stored in Data Set object for easy retrieval. The Microsoft access file is saved in the folder where the executable of the project is saved.

Any change based on the design is updated and stored in the array after editing and pressing the Update Response button. The program runs through the codes and using steepest decent algorithm to compute the coefficient needed to achieve the desired frequency response.

By clicking the *Design* button in the form, designing the filter code is triggered as shown in fig 2

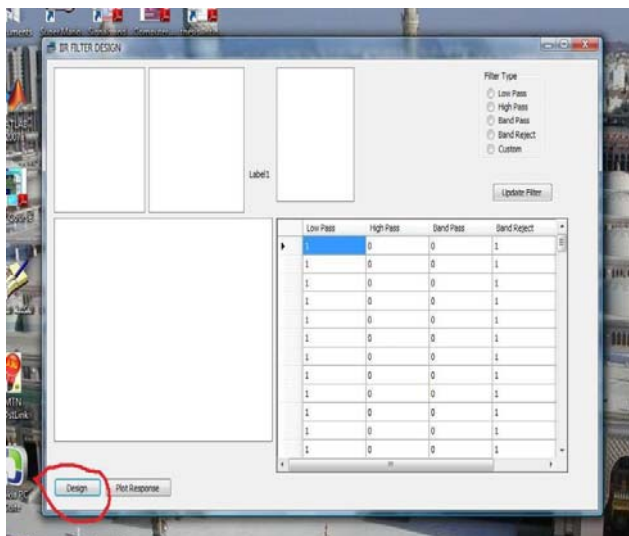


Fig 2 Design View Of IIR Filter Software Environment

#### IV. TESTING AND RESULT ANALYSIS

The screenshot of the result of an 8 pole low pass IIR filter is shown in figure 3

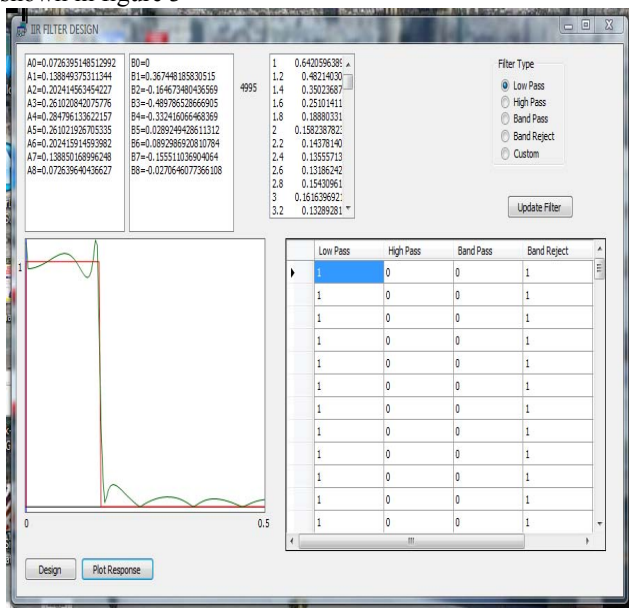


Fig 3 Low Pass IIR Filter Design Screenshot

The coefficients for the low pass IIR filter, which were found using neural network, are listed in table 1.

Table 1 Low Pass IIR Filter Coefficients

A COEFFICIENTS	B COEFFICIENTS
A(0) = 0.072639	B(0) = 0
A(1) = 0.138849	B(1) = 0.36744
A(2) = 0.202414	B(2) = -0.16467
A(3) = 0.26102	B(3) = -0.48978
A(4) = 0.28479	B(4) = -0.33241
A(5) = 0.26102	B(5) = 0.02892
A(6) = 0.20241	B(6) = 0.08929
A(7) = 0.138849	B(7) = -0.15551
A(8) = 0.072639	B(8) = -0.02706

The coefficients above are substituted, in equation 5 to get the transfer function of 8-pole IIR digital low pass filter, as shown in equation 13

$$H(z) = \frac{0.072639 + 0.138849z^{-1} + 0.2024z^{-2} + 0.261z^{-3} + 0.2847z^{-4} + \dots}{1 + 0.3674z^{-1} - 0.1646z^{-2} - 0.4897z^{-3} - 0.3324z^{-4} + \dots} \dots 13$$

The pole-zero plots of the transfer function of equation 13 is given in figure 5. The filter is stable since all the poles lies within the unit circle.

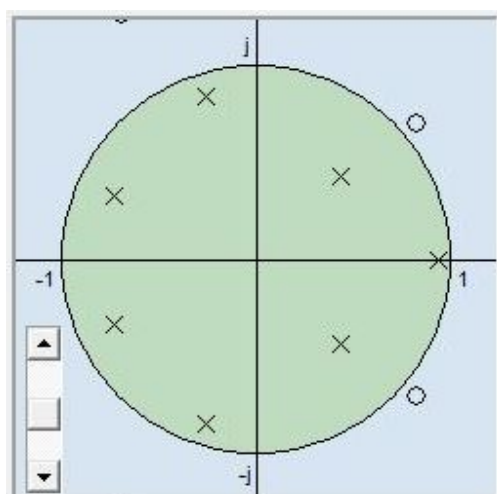


Fig 5. Pole-Zero Plot of Low Pass Filter

#### Custom Filter

This is the beauty of this project, using ANN to design a custom filter, whose response is totally defined by the user. Arbitrary values are selected as shown in figure 4

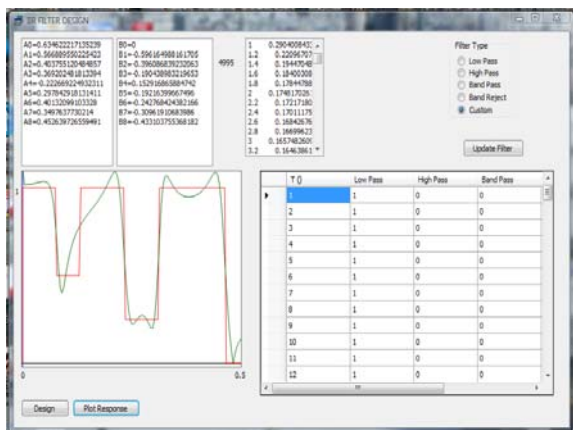


Fig 4. Custom Filter Design Screenshot

The coefficients for the custom IIR filter, which were found using neural network, are listed in table 2.

Coefficients are listed in table 2 below.

Table 2 Custom Filter Coefficients

A COEFFICIENTS	B COEFFICIENTS
A(0) = 0.6346	B(0) = 0
A(1) = 0.5668	B(1) = 0.5961
A(2) = 0.4037	B(2) = -0.3960
A(3) = 0.3692	B(3) = -0.1904
A(4) = -0.2226	B(4) = 0.1529
A(5) = 0.2978	B(5) = -0.1921
A(6) = 0.4013	B(6) = -0.2427
A(7) = 0.3497	B(7) = -0.3096
A(8) = 0.4526	B(8) = -0.4331

The coefficients above are substituted in equation 5 to get the transfer function of 8-pole IIR digital custom filter, as given in equation 14

$$H(z) = \frac{0.6346 + 0.5668z^{-1} + 0.4037z^{-2} + 0.3692z^{-3} - 0.2226z^{-4} + \dots}{1 + 0.5961z^{-1} - 0.396z^{-2} - 0.1904z^{-3} + 0.1529z^{-4} + \dots} \dots 14$$

The pole-zero plots of the transfer function of equation 14 is given in figure 6. The filter is stable since all the poles lies within the unit circle.

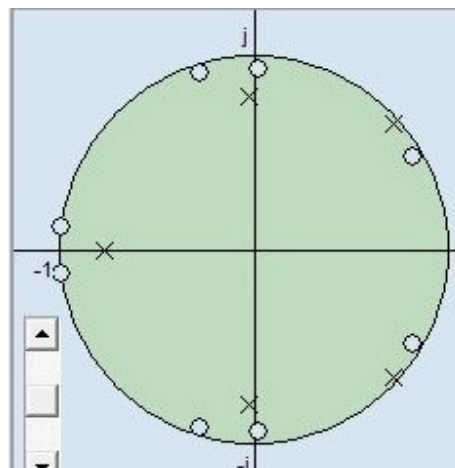


Fig 6. Pole-Zero Plot of Custom Filter

## V. CONCLUSION

Using the Artificial Neural Network algorithm, an IIR filter software was developed for designing any response especially custom filters. Again the concept of neural networks in DSP has solved the existing problem of designing custom filters by trial and error methods. Custom filters are use for *deconvolution* that is restoring signals that undergo unwanted convolution. It is also used in *optimal filtering* which is use in separating signals with overlapping frequency spectra. [1]

## REFERENCES

- [1] Steven W. Smith "The Scientist and Engineer's Guide to Digital Signal Processing" California Technical Publishing, 1999.
- [2] [http://en.wikipedia.org/wiki/Filter\\_design](http://en.wikipedia.org/wiki/Filter_design) "Filter Design" April, 2009
- [3] Farooq Azam "Biologically Inspired Modular Neural Networks" Phd Thesis, Blacksburg, Virginia May, 2000.
- [4] Daniel Graupe "Principles Of Artificial Neural Networks" Chicago Illinois. World Scientific, January 1997.
- [5] [http://en.wikipedia.org/wiki/Digital\\_Filter](http://en.wikipedia.org/wiki/Digital_Filter) "Digital Filter" April, 2009
- [6] Patrick Gaydecki "Digital Signal Processing" University of Manchester 2009.