DIGITAL FILTER DESIGN USING ARTIFICIAL NEURAL NETWORK

Adewole, Adetunji Philip*
Department of Computer Science, University of Agriculture, Abeokuta
philipwole@yahoo.com
&
Agwuegbo , Samuel Obi-Nnamdi
Department of Statistics, University of Agriculture, Abeokuta
E-mail: agwuegbo_son@yahoo.com

ABSTRACT

In this paper, Feed Forward Multi-Layer Perceptron neural network was adapted as a digital filtering tool in modeling communication systems that were corrupted by noise or interference.

Discrete-Fourier Transform was used to reduce error in transmission. The input and target output data from the study were generated using ionosphere data radar, and this proved to be essential and necessary for training and testing the network.

The network was trained using MATLAB R2008a and the training resulted to the minimization of the error. The result of digit filtration shows a near error-free output.

In conclusion, the forward-feed multilayered neural network can be used to build a functional digital filter.

Keywords: Analog signal, Data Communication, Digital Signal, Neural Networks, Signal Processing.

1.0 INTRODUCTION

Digital filter design emerged in the middle of sixties as a discipline in its own right with some well formulated techniques for analysis and design. Windowing techniques and the use of bilinear transform for recursive-filter design were discussed in [1] and [2]. At about the same time, faster and less costly digital computers were developed and considerable strides were made in integrated circuitry. It became apparent that digital filters were becoming competitive with analog filters, even for real-time signal processing [1]. Digital signal processing has found wide application in speech analysis and synthesis [3], and is playing an increasing role in communications.

In a communication system, a message or signal containing information is transferred from one place to another through a communication channel such as the atmosphere, a transmission line or cable, or a wave guide or optical fiber. Generally, these channels are subject to noise, interference, and losses and the signals received, differs from the signal transmitted. Research works have shown that digitized form of data transmission is better compared to the analog form because they are easy to read and manipulate in discrete form [4]. Enormous demand for communication equipment, real time data acquisition and processing and quest for ever improving voice and video compression algorithm has fuelled research and development in the field of digital signal processing (DSP) hardware and software [5].

This new generation filters rely on mathematical model to quickly and precisely filter the signal and extrapolates requirements of future data. The digital filter has been used in many areas where analog filter may have found it difficult to or impossible to apply.

Those areas include phone, compact disk player, electronics voice mail. The combination of this various areas of application makes it possible for any one individual to master all of the (DSP) technology that has been developed. Input signals such as audio signals, come in an analog form; they therefore need to be converted to digital signals. When using or transmitting data in an analog or digital form from one source to another over a communication channel, transmission impairments are bound to occur.

Transmission impairments are factors that cause signal distortion, delay distortion and noise. In this study, Feed forward Multilayer Neural Network is used in designing digital filter that is geared towards solving the problem of signal distortion in data transmission.

2.0 RELATED WORK

2.1 Numerical Method of Digital Filter

Digital filters evolved from simulation of analog filters on the early digital computers of the forties. One intuitive method of developing digital filter is by numerically simulating its analogue counterpart. Figure 1 shows a resistance capacitance (RC) low pass filter circuit. The basis of numerical simulation lies in converting the continuous differential equation that described the behaviour of a circuit into the difference equation. The general procedure for doing this is similar to differential calculus but in reverses. In calculus, you begin with a small but finite D and move toward an infinitesimal d. To turn differential equation into difference equation you have to translate the problem from the dt of calculus to Dt of numerical analysis.

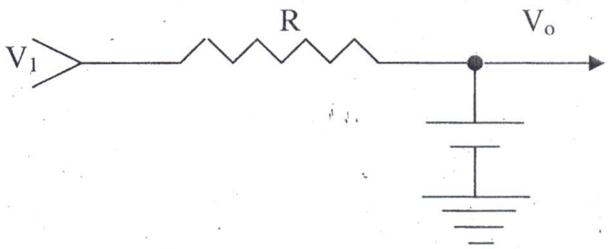

Figure 1. Low Pass Filter Circuit.

$$\frac{\delta V_o}{\delta t} = \frac{1}{RC}(V_1 - V_0) \qquad (1)$$

$$V_o(n+1) = V_o(n) + \frac{1}{RC}(V_1 - V_o(n))\Delta t \quad (2)$$

The differential equation relating output voltage y(t) to input voltage x(t) is

$$\frac{\delta y(t)}{\delta t} = \frac{1}{RC}[x(t) - y(t)] \quad (3)$$

Note that x and y are now just sequence of numbers of the sequence indicated by subscripts as opposed to continuous function.

A bit of rearrangement and sleight of hand gives an equation that is useful for predicting the next value of y from both its present value and the present value of the input x.

$$Y_{n+1} = Y_n + \Delta Y \quad (4)$$

$$\Delta Y = \frac{1}{RC}(X_n - Y_n)\Delta t \quad (5)$$

$$Y_{n+1} = Y_n + \frac{1}{RC}(X_n - Y_n)\Delta t \quad (6)$$

A difference equation in this form is easily expressed for computer solution.

Basically, the equation might be written as:

$$Y_{new} = Y_{old} + \frac{1}{RC}(X_{new} - Y_{old})\Delta t \quad (7)$$

2.2 Discrete Fourier Transform

To analyze a signal-processing problem that requires a filter, there is a need to leave the time domain and find a tool for translating our filter models into the frequency domain, because most of the signals that will be analyzed cannot be completely understood in the time domain. For example, most signals consist not only of a fundamental frequency, but also harmonics that must be considered, or they consist of many discrete frequency components that must be accounted for by the filters being designed. There are many tools that can be used to help understand the frequency-domain nature of signals. One of the most commonly used is the Fourier series. It has been shown that any periodic signal can be modeled as an infinite series of sines and cosines. The transform which converts the time domain information into frequency domain is called the Discrete Fourier Transform (DFT). One of the main reasons for utilizing the DFT in many applications is the existence of a fast algorithm to compute DFT. This fast algorithm is known as the Fast Fourier Transform (FFT). In the following, the mathematical Transform (IDFT) of a discrete sequence is $\{ x_n \}_{n=0}^{M-1}$ i.e. $\{x_0, x_1, x_2, \ldots, x_{m-1}\}$.

DFT: $$X(k) = \sum_{n=0}^{M-1} x(n)\, W_M^{kn} \quad \ldots\text{for } 0 \le k \le (M-1) \ldots\ldots\ldots\ldots(8)$$

IDFT: $$x(n) = \frac{1}{M}\sum_{k=0}^{M-1} X(k)\, W_M^{-kn} \quad \ldots\text{for } 0 \le n \le (M-1) \ldots\ldots\ldots(9)$$

Where

$$W_M = e^{-j(\frac{2\pi}{M})} \quad \text{...(10)}$$

3.0 METHODOLOGY

3.1 Training the Network

Consider a single layer of conventional perceptrons. Let the sequence of input vectors be $\{ Y_1, Y_2, ..., Y_L \}$. The following supervised learning procedure is utilized to classify the signal:

- Apply the DFT to the successive input training sample vectors resulting in the vectors $\{ Z_1, Z_2, ..., Z_L \}$.
- Train a single layer Perceptrons using the transformed sample vectors.

In conventional model of neuron, weighted contribution (weights being the synaptic weights) of *current* input values is taken and a suitable activation function (Signum or Sigmoid or hyperbolic tangent) is applied. A biologically more probable model takes the following facts into account

- The output of a neuron depends *not only* on the current input value, but all the input values over a finite horizon. Thus inputs to neurons are defined over a finite horizon (rather than a single time point).
- Synapses are treated as distributed *elements* rather than lumped elements. Thus synaptic weights are functions defined on a finite support.

For the sake of convenience, let the input as well as synaptic weight functions be defined on the support [0, T].

3.2 Mathematical Model of Neuron

Let the synaptic weights be $w_i(t)$, $1 \le i \le M$ i.e time functions defined on the support [0,T]. Also, let the inputs be given by $a_i(t), 1 \le i \le M$.

Thus, the output of the neuron is given by:

$$y(t) = Sign\left(\sum_{j=1}^{M} a_j(t)\, w_j(t) \right) \qquad (11)$$

More general activation functions (sigmoid, hyperbolic tangent, etc.) could be used. The successive input functions are defined over the interval [0,T]. They are fed as inputs to the continuous time neurons at successive SLOTS. For the sake of notational convenience, we call such a neuron, a continuous time perceptron.

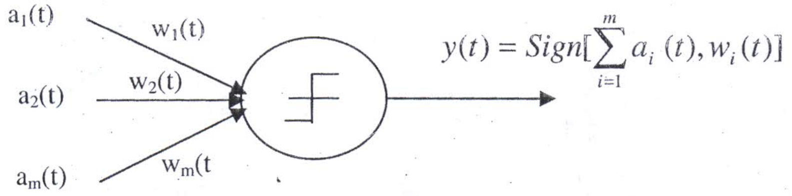

Figure 1: A Novel Model of Continuous Time Neuron

3.3 Continuous Time Perceptron Learning Law

As in the case of "conventional perceptron", a continuous time perceptron learning law is given by::

$$W_i^{(n+1)}(t) = W_i^{(n)}(t) + \eta(S(t)-g(t))\,a_i(t) \quad (3.2.2)$$

where S(t) is the target output for the current training example, g(t) is the output generated by the continuous time perceptron and η is a positive constant called the learning rate. The proof of convergence of conventional perceptron learning law, also guarantees the point wise convergence (not necessarily uni-form convergence) of synaptic weight functions.

Using sigmoid function as the activation function and the continuous prceptron as the model of neuron, it is straightforward to arrive at a continuous time Multi-Layer Perceptron. The conventional back propagation algorithm is generalized to such a feed forward network.

3.4 Modulation Theory: Feed Forward Neural Networks:

Suppose the synaptic weight functions are chosen as sinusoids i.e.

$w_i(t) = \cos \upsilon_i t$ *or* $\sin \upsilon_i t$ (where $\upsilon_i = 2\pi f_i$ and f_i's are frequencies of the sinusoids). The weighted contribution at each neuron actually corresponds to Amplitude Modulation (where the synaptic weight functions are the carrier frequencies and the inputs are the base band signals).

The input and target output data proved to be the most essential parameter necessary for training and testing network. Therefore, the input and targeted output for this study was generated using ionosphere data radar. The network was trained using data already stored in notepad package. After loading the data, the number of epoch was specified and the network was trained.

4.0 IMPLEMENTATION AND DISCUSSION OF RESULTS

4.1 Original Signal

The original signal is the pure signal from the source which has not been affected by any impairment. The Matlab code was used to generate it. The implementation of the code gives the graph as the output, which is the original signal that has not been corrupted.

4.2 Corrupted Signal

The corrupted signal occurs as a result of original signal passing through a medium and has been affected by the impairments such as attenuation, distortion, and noise. The graph of the signal is shown in Figure 2.

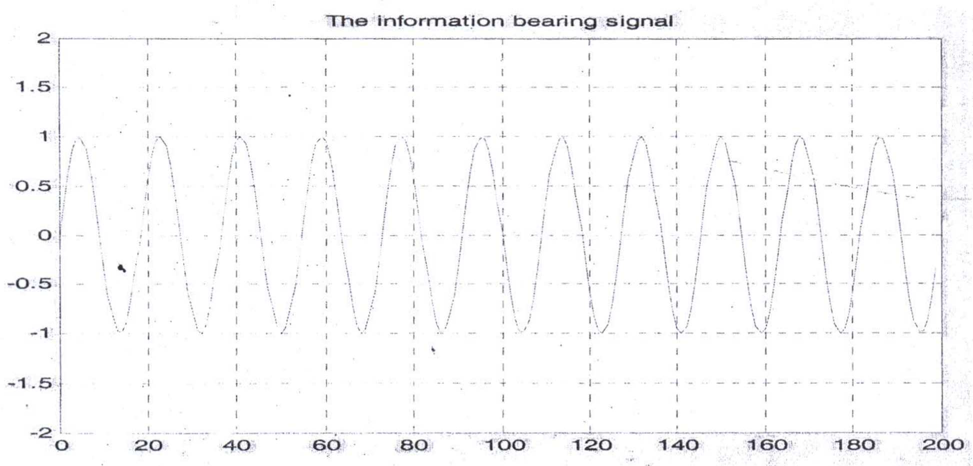

Fig 2: Original signal graph

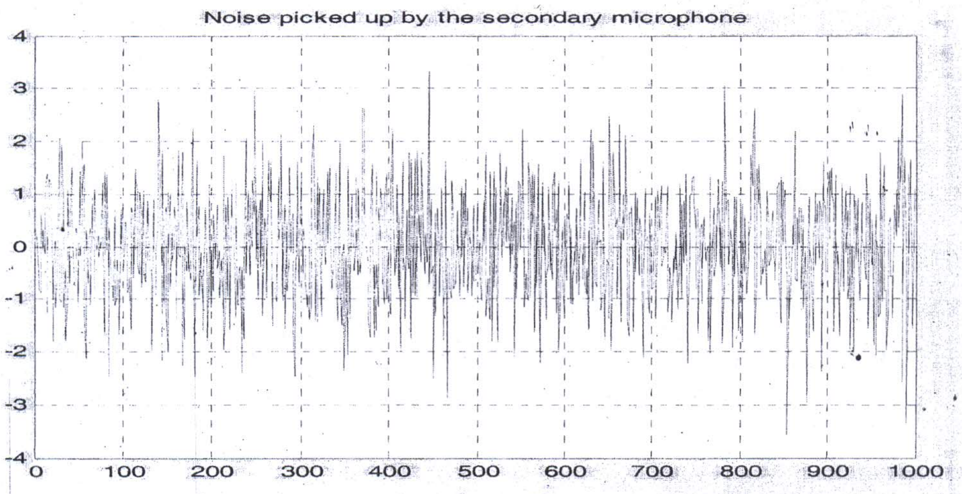

Fig 3: Corrupted Signal Graph

4.3 The Filtered Signal

The graph below is the output of the filtered signal.

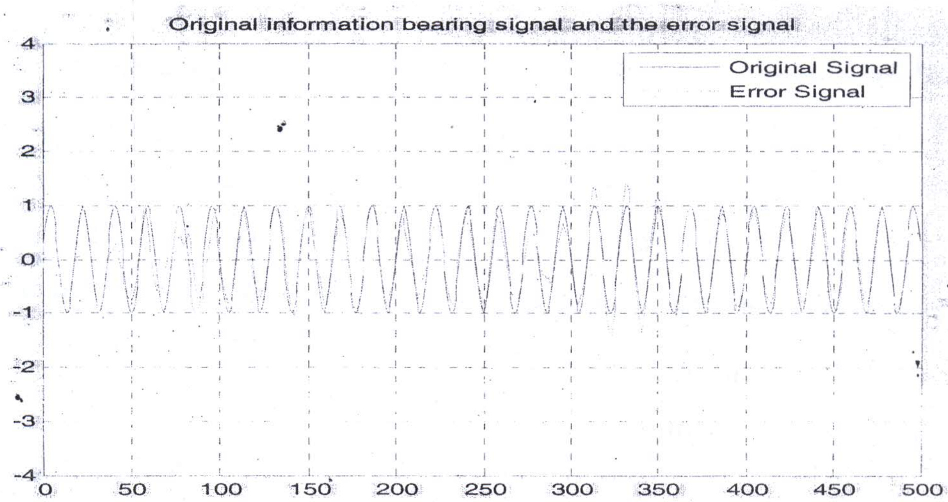

Fig.4: The output of the digital filtering

4.4 Discussion of Results

Figure 2 represents the original signal graph. As can be seen from this graph the signal has not been disturbed by any impairment. The graph is a pure sign wave graph with the range of 1 and -1 in the amplitude on the same axis. The next figure which is figure .3 is a corrupted signal. This graph is non regular shaped graph caused by impairments and it is totally different from the original signal graph. The range in the amplitude of this is between 3.3 and –3.5. Figure 4 is a filtered signal graph after the training of about 320 data set and it can be seen that there is no much difference in original signal and the recovered signal after the digital filtering has been performed. The range in the amplitude of this is between 1.3 and –1.3.

5.0 CONCLUSION

The study proposed digital digital filter utilizing FeedForward multilayered neural network. The filter significantly reduces random noises superimposed on signals. The high performance of this neural filter is demonstrated in computer simulation depicted in the study. When the noise power is small, the performance of the neural filter is almost the same as that of the ε-filter, which corresponds to a simplified neural filter; however, when the noise power is large, the effectiveness of the neural filter is clearly demonstrated.

6.0 REFERENCES

[1] Cunningham, E. P. (1995): "Digital Filtering: An introduction". John Wiley and Son, Inc. New York.

[2] Kaiser, J.F (1966): Digital Filters Chap 7 of system Analysis by Digital Computer ed Kuo, F.F and Kaiser, J.F, Wiley, New York.

[3] Cappelinni, V.; Constantinides,A.G; and Emiliani, P (1978): Digital Filters and their Application, Academic Press, London

[4] Hosson L.N. (1997). Neural Network Based Dynamic Reconstruction filter for digital audio signals, Australia

[5] Alexander and Lazarevic (2000): "Modeling of physical system using multidimensional serial wave digital", retrieved 2000,www.digitalsystem group.