ISSN 2319 – 2518 www.ijeetc.com
Vol. 3, No. 2, April 2014

Research Paper

DESIGN AND ANALYSIS OF DIFFERENT CONVOLUTION TECHNIQUES

Kirandeep Kaur[1]*, Harmanpreet Kaur[1] and Rajesh Mehra[1]

*Corresponding Author: Kirandeep Kaur, ✉ kiran.saini90@gmail.com

In this paper the technique of convolution and its need in signal processing is discussed. The comparison of three basic convolution techniques like linear, circular convolution and Discrete Fourier Transform for general digital signal processing is analyzed. Their performance has been compared in terms of computational complexity, number of operations and number of outputs. A relationship has been drawn between the three techniques using MATLAB simulation and mathematical methods.

Keywords: Linear convolution, Circular convolution, Discrete fourier transform

INTRODUCTION

Convolution is considered to be heart of the digital signal processing. It is the mathematical way of combining two signals to obtain a third signal. Convolution helps to estimate the output of a system with arbitrary input, with knowledge of impulse response of the system. A linear system's characteristics are completely specified by the system's impulse response, as governed by the mathematics of convolution.

Convolution is an operation which takes two functions as input, and produces a single function output (much like addition or multiplication of functions). The method of combining these functions is defined as

$$f(t)*h(t)=\int_{0}^{t} f(^{2})\,h(-^{2})\,d^{2} \quad ...(1)$$

Need of Convolution

Let's consider the case of formation of ripples on the surface of a pond. If a stone is thrown in the standing water of the pond, then it will cause a ripple to travel outwards across the surface. Now on the next instant when we take a bigger stone, then a bigger ripple is expected. Similarly if lighter stone is thrown, small ripple is expected. All three of the ripple patterns above share the same general shape, but differ only in their magnitude. Now to calculate the impact of different stones, we look at the impacting stone as our input and the resulting

[1] Department of Electronics and Communication Engineering, National Institute of Technical' Training & Research Chandigarh, UT, India.

ripple as our output. We can express the dependence of the ripple's size on the weight of the stone by saying that the output scales linearly with the input. Let's consider a special stone, say of unit weight. If we know the ripple caused by this stone, we can then find the ripple caused by any stone (lighter or heavier) by just scaling by the proper amount.

Similarly this approach can be extended to various signals in signal processing. Let's consider a case where determining the frequency spectrum or frequency transfer function of a linear network provides one with the knowledge of how a network will respond to or alter an input signal. Conventional methods used to determine this entail the use of spectrum analyzers which use either sweep generators or variable-frequency oscillators to impress upon a network all possible frequencies of equal amplitude and equal phase. The response of a network to all frequencies can thus be determined. Any amplitude and phase variations at the output of a network are due to the network itself and as a result define the frequency transfer function. Another means of obtaining this same information would be to apply an impulse function to the input of a network and then analyze the network impulse-response for its spectral frequency content. Comparison of the network-frequency transfer function obtained by the two techniques would yield the same information. Since an impulse response provides information of a network frequency spectrum or transfer function, it additionally provides a means of determining the network response to any other time function input. Hence convolution can be used to find response of any signal just knowing its impulse response. Convolution relates the three signals of interest: the input signal, the output signal, and the impulse response. There are various types of techniques to perform convolution.

CONVOLUTION TECHNIQUES

There are different type of techniques used to implement convolution. They can be divided on the basis of time domain as well as frequency domain. In time domain we have linear convolution, circular convolution. On the other hand we can have Discrete Fourier transform and further its fast implementations to be categorised under frequency domain part.

Linear convolution is the simplest type of convolution. For two finite discrete sequences of length Nx and Nh, the linear or aperiodic convolution is defined by the form:

$$y(n) = \sum_{k=0}^{n} h(k)\, x(n-k) \quad ...(2)$$

where $h(k)$ and $x(n-k)$ are zero outside their appropriately defined intervals. For $N_x > N_h$, each summation need only be calculated for the $0 < k < N_h - 1$ terms. The output, $y(n)$, will have length $N_x + N_h - 1$. As we know that convolution forms the basis of all filtering processes. So if the linear convolution is used to implement tapped delay line FIR filter, it requires N_h multiplications and N_h additions for single output sample. This method is conducive for small impulse responses but its inadequate efficiency makes it undesirable. So we switch on to other techniques.

The circular convolution, also known as cyclic convolution, of two aperiodic functions occurs when one of them is convolved in the normal way with a periodic summation of the other function. Linear convolution can be obtained from its cyclic counterpart. Circular convolution is much more effective than linear convolution in a real-time signal processing system since it has successive inputs of data. By providing hardware for circular convolution, Digital Signal Processor (DSP) allows for such high speed convolution processing which happen frequently during the treatment of the time domain of signals (Ungseon Yun and Byeungwoo Jeon, 2010). In circular convolution instead of considering two sequences $x[n]$ and $h[n]$ of different length we consider them of same length N. Such that the output sequence $y[n]$ lies between 0 to $2N–1$. Now if we extend $x[n]$ and $h[n]$ periodically with period N (Chi-Tsong Chen, 1994). Then circular convolution is represented by the equation.

$$y_c[k] = \sum_{i=0}^{N-1} x[i]\, h[k-i] \quad \text{...(3)}$$

The basic motive behind implementing circular convolution is that it saves memory as well as improves the processing time.

In frequency domain the given time domain sequences are transformed to frequency range and are simply multiplied. The Discrete Fourier Transform (DFT) is purely discrete, discrete-time data sets are converted into a discrete-frequency representation.. For the discrete case, multiplication in the frequency domain translates circular convolution in the time domain (Rabiner and Gold, 1975). So to overcome the computational complexity with circular convolution we shift to frequency domain and uses DFT. The discrete Fourier transform changes an N point input signal into two $N/2+1$ point output signals. The input signal contains the signal being decomposed, while the two output signals contain the amplitudes of the component (Steven, 2002). The DFT for a signal with period N is:

$$X[k] = \sum_{n=0}^{N-1} x[n] e^{j\frac{2\pi kn}{N}} \quad \text{...(4)}$$

The IDFT for a signal can be calculated as:

$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} x[k] e^{j\frac{2\pi n}{N}} \quad \text{...(5)}$$

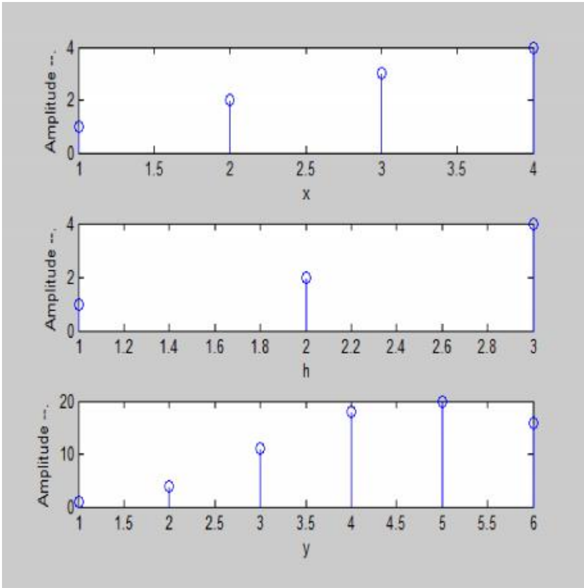
We can use DFT and IDFT to calculate the circular convolution using the equation:

$$y(n) = IDFT(Y(m)) = IDFT(X(m) \cdot H(m)) \quad \text{...(6)}$$

MATLAB RESULTS AND SIMULATION

Let's take an example by taking two sequences as $x(n)$ = [1 2 3 4] and $h(n)$ = [1 2

Figure 1: Linear Convolution of Two Sequences Using Matlab



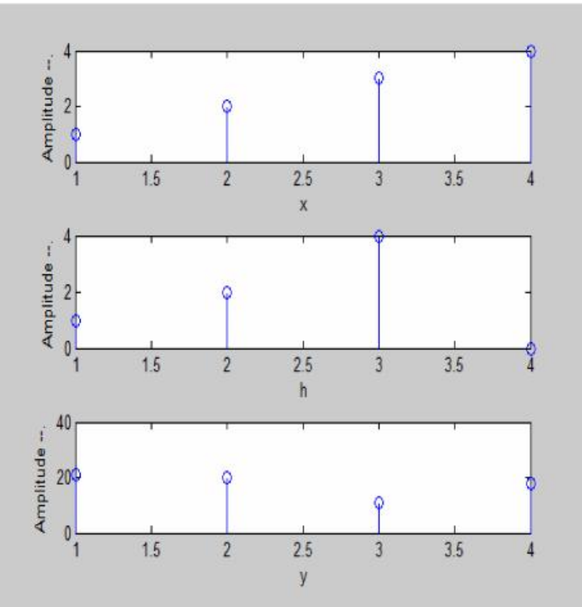
4]. Computing the linear convolution of above sequences using Matlab are as:

The above graphical results can be verified using an easy mathematical approach a novel method for calculating convolution sum of two finite length sequences (John Pierre, 1996).

x(*n*):			1	2	3	4
h(*n*):			*	1	2	4
			4	8	12	16
		2	4	6	8	
	1	2	3	4		
	1	**4**	**11**	**18**	**20**	**16**

Let's consider two sequence *x*[*n*] = [1 2 3 4] and *h*[*n*] = [1 2 4 0] of the same length and compute its circular convolution using Matlab code.

Figure 2: Circular Convolution of Two Finite Sequences



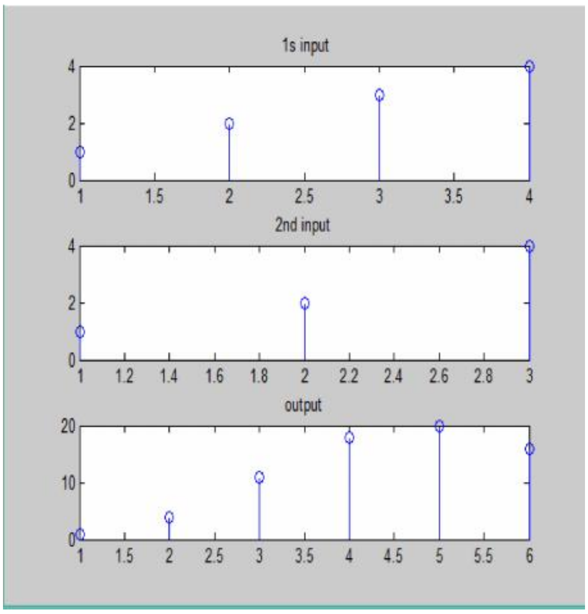
The above code computed result can be verified using the novel method for calculating convolution(John Pierre, 1996).

h[*n*]:		1	2	4	0
x[*n*]:	*	1	2	3	4
		4	8	16	0
		6	12	0	3
		8	0	2	4
		0	1	2	4
		18	21	20	11
		y(3)	y(0)	y(1)	y(2)

The far left value computed by this method corresponds to *y*(*N* – 1) [2] where *N* is the length of the sequence. Therefore value 18 corresponds to value *y*(3). The lengths of the sequences produce by discrete-time convolution and circular convolution are different. Specifically, the sequence that results from circular convolution is the same length, *N*, as the two input sequences. However, discrete-time convolution results in a sequence that has theoretically infinite length, but the interval over which the output is nonzero is [0, 2*N* – 2].

Discrete-time convolution can be thought of as a low-pass version of the original signal on the interval [0, *N* – 1] with a "tail" of filter decay placed onto the end interval [*N*, 2*N* – 2]. The output from circular convolution can be seen to be the same as that for discrete-time linear convolution except at the ends of the signal. Hence in the middle of the sequences, both circular convolution and discrete-time convolution give roughly the same output values. The length of a circular convolution is equal to the length of the longest sequence. The trailing samples that were left alone in linear convolution are now wrapped around, which leads to short and different results. The problem with convolution though is reduced to an extend but still in terms of number multiplications and additions required to calculate each sample persists.

Figure 3: Calculating Linear Convolution Using DFT and IDFT



The Matlab result for calculating circular convolution for sequence $x[n]$ = [1 2 3 4] and $h[n]$ = [1 2 4] using DFT and IDFT are as:

This method yields the desired linear convolution result only if $x(n)$ and $h(n)$ are padded with zeros prior to the DFT such that their respective lengths are $Nx + Nh - 1$, essentially zeroing out all circular artifacts.

The disadvantage of frequency domain method is Processing in the frequency domain introduces significant input to output latency, since the input must be initially buffered and transformed into the frequency domain. Conversely, the output must be inversely transformed into the time domain. The buffering imposes a minimum $2 \times Nx$ sample latency. The frequency transformation itself and the convolution process introduce further overhead (Jason, 1998).

Table 1. Comparison Between 3 Techniques

Parameters	Linear Convolution	Circular Convolution	DFT
No. of Outputs	6	4	6
No. of Adders	5	3	5
No. of Multipliers	5	4	6
No. of States	5	3	5
Multiplications per Input Sample	5	4	6
Additions Input Sample	5	3	per 5

The above table shows the quantity wise comparison between the three convolution techniques. The above table is based on the Matlab results obtained by designing filter on the given 3 techniques and analysing its cost implementation.

CONCLUSION

An effort has been made to compare conventional convolution techniques and to visualise a relationship between them. Linear convolution is computation wise more complex. Circular convolution gives reduced number of outputs by about 33% in comparison to linear type and hence memory requirement is less. DFT is far more easier in terms of mathematical calculations than other two techniques but it has more hardware latency and number of overheads. But these different methods of convolution provides approximately same value of outputs.

REFERENCES

1. Chi-Tsong Chen (1994), "System and Signal Analysis", *University of New York*

at Stony Brook, Vol. 2, pp. 410-413, Oxford University Press.

2. Jason R VandeKieft (1998), "Computational Improvement to Linear Convolution with Multirate Filtering Methods", Research Project, Coral Gables, Florida.
3. John W Pierre (1996), "A Novel Method for Calculating the Convolution Sum of Two Finite Length Sequences", *IEEE Transactions on Education*, Vol. 39, No. 1, pp. 77-80.
4. National Semiconductor Application Note 237, January 1980.
5. Steven W Smith (2002), "The Scientist and Engineer's Guide to Digital Signal Processing", Newnes Publication Edition.
6. Ungseon Yun and Byeungwoo Jeon (2010), "Implementation of Real-Time High Speed Convolution Using DSP", *New Trends in Information Science and Service Science (NISS)*, May, pp. 499-502.