Kufa Journal of Engineering
Vol. 16, No. 3, July 2025, P.P. 485-498
Article history: Received 18 September 2024, last revised 27 December 2024, accepted 27 December 2024

A FAST ALGORITHM FOR COMPUTING SHORT AND LONG –LENGTH LINEAR AND CIRCULAR DISCRETE CONVOLUTION

Ali K. Nahar[*] and Hussein K. Khleaf [*]

[*] **Electrical Engineering Dept., University of Technology-Iraq.**

https://doi.org/10.30572/2018/KJE/160327

ABSTRACT

Convolution is a powerful operator that has applications in science, engineering, and mathematics. However, Convolution:-(Issues and Applications) is necessary for addressing many scientific and technical cases, including partial differential equations, signal processing, and image processing. Traditional problem resolution is complicated and has several drawbacks. This work presents a set of efficient algorithmic methods for both linear and circular computing. Depending on the duration of introduction and the treatment medium, three sloven methods can be applied based on the rapid table method. Convolution-based systems are very suitable for dealing with complex data that requires appropriate handling to achieve an ideal solution with increasing data length. Though the suggested techniques are geared toward fully parallel hardware implementation, they are contrasted with depending on length N and multiples, fully parallel hardware implementation using the proposed approach requires 40% to 65% less compared to the traditional approach. Since multipliers need a lot more space on the chip and energy than adders, the proposed algorithms are resource and power efficient when implemented on hardware.

KEYWORDS

Digital Signal Processor, linear convolution algorithms; circular convolution algorithms; Fast Table Methods, Discrete Fourier transform.

1. INTRODUCTION

Convolution is a key component for many scientific and engineering problems, such as signal processing, partial differential equations, and image processing, added to it are the correlation and discrete linear correlation methods,and their applications in many fields . Numerous scientific and engineering applications involve discrete convolution (Krishna, 2017; Bi et al. , 2004). Most importantly, it is essential to contemporary digital signal and image processing. It is basis filtering, multi resolution decomposition, and orthogonal computation optimization in digital signal processing transform (Wang et al. ,2018; Parhi, 2007). Convolution is a fundamental mechanism used in digital image processing for smoothing and noise removal. Blur, focus, edge detection, and so forth (Chan et al. ,1994). Discrete convolutions come in two flavors: linear and circular convolutions. Overarching guidelines for the creation of convolution algorithms (Vasilache, et al. ,2014) gives a description of them. The (Abdelkareem, 2017) implemented hardware considerations Based on convolution Digital Signal Processor (DSP) platform decryption is discussed. Specifically, the effect of code constraint length both memory management and clock cycles are taken into account.

In these works, computing has been the primary focus. Although circular convolution is computed in many applications related to digital signal and image processing Convolutions must be linear. Convolution has been extremely popular in Fast Table Method (FTM) in recent years. most frequently employed (Pratt, 2007). Since multiple accounts account for more than 67% of the processes in a typical implementation, linear convolutions are the easiest and most computationally dense processes in FTMs (Krizhevsky et al. ,2017). In a typical FTM, more than two thousand multiplications and additions are needed for just one convolution level. In the FTM, there are typically multiple of these levels. Because of this, designers of these kinds of networks look for effective techniques to execute linear convolution with the fewest number of arithmetic operations. Many algorithmic techniques have been developed to accelerate the computation of linear convolution. The most popular method for computing linear convolution efficiently involves dipping it into the space of a double-size cyclic convolution and then using the Fast Fourier transform (FFT) technique (Mathieu et al. , 2013; Lin et al. , 2018).

These techniques do not compute the genuine linear convolution since they merely compute two inner products of neighboring vectors produced from the current data stream by a sliding time frame of length N. Concurrently, there are several FTMs where full-size small-length linear convolutions must be computed. For example, in the context of the Discrete-time Fourier transform (DTFT). Furthermore, the interpolation-based signal frequency estimation algorithm is widely used in digital systems because of its ease of use. Interpolation techniques in analysis rely on the FFT due to its high-performance speed. This is to improve the accuracy of frequency estimation. This paper proposes a method for single-tone frequency estimation that uses DFT interpolation with Parzen window (Alrubei et al. , 2023). The difficulty of computing a one-dimensional convolution using its conversion into a multidimensional convolution arises in many applications of digital signal processing. The resulting algorithm is modular in nature, with each module computing a one-dimensional convolution of a brief length (Abtahi et al. , 2018; Lavin et al. , 2016). Sequences of lengths *3, 4, 5, 6, 7*, and *8* are the most prevalent types of twisted sequences. Nevertheless, there is no description of resource-efficient sequences of unlimited length in the known papers of the authors. Linear convolution algorithms for lengths greater than four (Ju et al. , 2019). Conversely, the solutions offered in the literature for $N = 4$, $N = 3$, and $N = \infty$ lack full inventiveness. Regarding how the computation of linear convolution is organized, given that its matching sign There are no flow charts visible anywhere.

Using the FTM by convolution multiplication is a method that is actually used (Hsu, 2011; Heba, et al., 2024). Whereas, a set of cases will be established in this work that have not been used previously. Moreover, only this method is used with specified inputs x,h lengths in a simplified manner. In this work, innovative FM method, comprehensive mathematical relationships, and very fast results will be achieved. Thus, this work achieves a new, unused method, using FTM for entries of infinite lengths, creating fast results and simplified calculations that lead to the greatest benefit achieved.

Image encryption, which uses logistical chaotic maps to satisfy the requirement for encrypted data communications during image acquisition, is one of the most significant applications of the computational approach suggested in the study. The suggested picture encryption technique is a safe and efficient way to encrypt and transfer photos, according to recent sources. The process of digital image encryption, which entails transforming source images into a format that is challenging to decode, has been studied using logistic chaotic maps (Roberts, 2012). In future research, cases and algorithms can be realized using FPGA, which is the latest class and takes its place in DSP applications have been implemented and demonstrated the ability to handle such tasks it supports the critical needs of scalability, speed, scale, cost and efficiency (mahmoud et al. , 2006; Naghmash et al. , 2016).

2. THEORY

Convolution is a mathematical operation on two functions (x and h) that results in a third function (y) that expresses how the form of one is changed by the shape of the other (Lavin et al. , 2016). The classic linear convolution equation:

$$y[n] = \sum_{k=0}^{n} x[k] * h[n-k] \qquad (1)$$

Where, $x(n), h(n)$; the input and the impulse response respectively.

Then, n= 1, 2, 3, ………, Nx+Nh-1.

This infinite sum says that a single value of n, call it y[n] may be found by performing the sum of all the multiplications of x[k] and h[n − k] at every value of k (mahmoud et al. , 2006; Naghmash et al. , 2016). Two DFTs (finite-length sequences, often of length N) cannot simply be multiplied together as in the convolution formula above, which is also known as linear convolution. Since the DFTs are periodic, their multiplication by n will also be nonzero for n ≥ N. This is because the DFTs have nonzero values for n≥N. A new kind of convolution operation must be defined in order for our convolved signal to be zero outside of the interval n = 0 , 1 , …, N−1. Using MATLAB y = conv (x, h) (Ghasemi et al. , 2017).

Periodic convolution, or the convolution of two periodic functions with the same period, is a special instance of circular convolution, also referred to as cyclic convolution. There is periodic convolution. Specifically, the periodic convolution of the TFTs of the separate sequences is the DTFT of the product of two discrete sequences. Additionally, every DTFT is a continuous Fourier transform function's periodic summation. While DTFTs are often continuous functions of frequency, discontinuous sequences of data can also directly benefit from the notions of periodic and circular convolution (Grinshpan,2017). Circular convolution is crucial in that situation for optimizing the effectiveness of a particular type of typical filtering operation. It was developed as a result of this concept.

Then; the circular convolution

$$y_c[n] = \sum_{k=0}^{Nx} \sum_{n=0}^{Nh} x[k] * h[n-k] \qquad (2)$$

Where the length Ny: of the circler is the Maximum length between lengths Nx and Nh.

3. NUMERICAL CALCULATION FOR FTM

The current structures rely heavily on Eq.1 and 2, and they can be referred to through references, which are traditional methods, such as the complex and long mathematical method, which must be solved using the two equations mentioned, and the method of graphics and linear projection onto the drawing (Krishna, 2017; Bi et al. , 2004; Wang et al. ,2018; Parhi, 2007). While, the matrix method uses multiplication and shift. All of these methods mentioned, the graphics, matrices and the equation method, do not deal with infinite lengths. In these methods, you need a specific length because they are specific equations or matrix, Therefore it is not compatible with parallel processing.

A reasonably recent method converts an optimal algorithm directly using index mapping. FTM then completes the brief convolutions along each dimension. There are no mathematical savings from the index mapping alone, in contrast to the DFT example. Effective short algorithms are responsible for all of the savings. The multiplications need to be nested together in the algorithm's core, just like in the case of index mapping with convolution. Convolution has no analogue for the Table structure. As the DFT was computed using row and column DFTs (Cariowa, 2020), the multidimensional convolution cannot be computed using row and column DFTs. Distributed arithmetic is an approach appealing for special purpose hardware. This method creates a system that performs convolution without the need for multiplications by using a table lookup of precomputed partial products (25Alkadhim, 2020).

Convolution is also calculated using number theoretic transforms, which calls for specialized hardware. These transformations are defined modulo exceptional numbers in arithmetic over finite fields or rings. Although these transforms are not very flexible, they are quite effective when applied.

Review the basic principles and steps followed for the FTM:

1- Distributing the entries on the first column and row. It makes no difference whether x is on the first row and h is on the first column, or vice versa, the row is for h and the first column is for x.

2- Multiply all opposite numbers and put the resulting number in the square Intersecting.

3- Taking an inclined path at an angle of 450, starting from the first square of the table to the end of the table, and collecting all the numbers that pass in each path to produce the output elements Y, as show in Fig.1.

4- If the chain starts from negative n to positive, passing through zero, it is possible to define n = 0 with a circle, and the path that passes through it will be n = 0 for output y, as in Fig.1.

5- To obtain circular convolution, it can be found in a simple and quick way that is easier than the well-known circle method (Li, et al., 2020) by finding the circular length of the output, which is the largest of the two input lengths, x, h, and then adding the number next to the length with the first number, and the number after it with the second, and so on, as appearance in Fig.2.

x[n] / *h[n]*	*x[-1]*	*x[0]*	*x[1]*	*x[2]*	…
h0	*x[-1]h[0]*	*x[0]h[0]*	*x[1]h[0]*	*x[2]h[0]*	…
h1	*x[-1]h[1]*	*x[0]h[1]*	*x[1]h[1]*	*x[2]h[1]*	…
h2	*x[-1]h[2]*	*x[0]h[2]*	*x[1]h[2]*	*x[2]h[2]*	…
⋮	⋮	⋮	⋮	⋮	

Fig. 1. Table convolution method

$$y[n] = [(x_{-1}h_0)\ldots(x_0h_0 + x_{-1}h_{-1})\ldots(x_1h_0 + x_0h_1 + x_{-1}h_2 + \ldots)] \quad (3)$$

Or y[n] = [y_{-1} y_0 y_1 y_2 y_3 y_4 y_5 . . .] if length of input x (N_x) equal 4 and length of impulse response N_h equal 3, can find the output length:

$$N_y = N_x + N_h - 1 \qquad (4)$$

Then, length of convulsion output N_y = 6

Fast Discrete Circular convolution based on the results of the linear convolution, whereNy: themaximumlengthbetweenN_x, N_h based an example N_y = 4, as appearance in Fig.2 ana results for Fast Discrete Circular convolution in Eq.5.

$$y_c[n] = [(y_{-1} + y_4)(y_0 + y_5)y_1y_2y_3] \qquad (5)$$

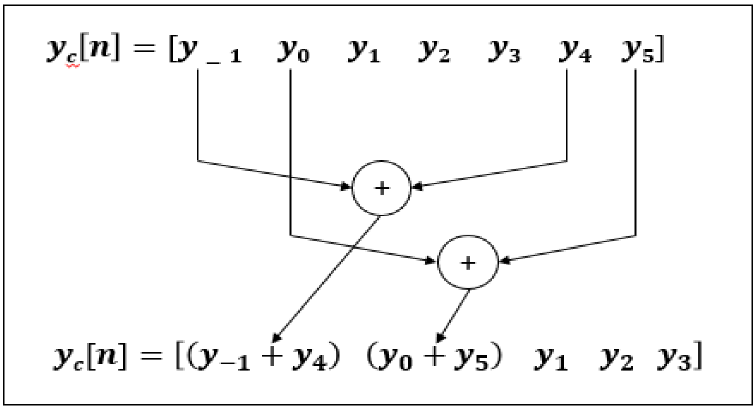

Fig. 2: Circular convolution yc[n] using output linear convolution

4. CASES AND RESULTS

In general, convolution problems are in three cases, depending on the length of the input Nx or the length of the impulse response Nh, where it is possible for each of the two inputs to be in the discrete state, either with a specific length and signal of a specific length, or to be infinite in length. Therefore, each case in details with an example of the algorithm are reviewed here.

4.1. First case

when the sequence for x[n] and h[n] are finite length.

The input sequence $x[n]$ and an impulse response $h[n]$ for a Linear Time Invariant (LTI) system as shown below:

$$x[n] = 2^{-n}(u[n+2] - u[n-3]) \quad ;h[n] = \left(-1^n(x[1-n])\right) \qquad (6)$$

Determine the (a) Linear Correlation; (b) Circular Correlation

First step finds x[n], that u[n+2] - u[n-3] this means the length for x[n] is finite length because n = -2 ,-1 , 0, 1, 2, as show in Fig.3.

The result of x[n] = [4 2 1 0.5 0.25], where n = -2, -1 ,0, 1, 2: Nx = 5 (Finite length)

Then $h[n] = \left(-1^n(x[1-n])\right)$ the results h(n) when n starting by n = -2 to n = 3 , equal x(3) = 0, x(2) = -0.25, x(1) = +0.5, x(0) = -1, x(-1) = +2, x(-2) = -4, x(-3) = 0

h[n] = [-0.25 0.5 -1 2 -4], where n = -1, 0, 1, 2, 3: Nh = 5 (Finite length)

Using Discrete convolution by Fast Table Method explain in Fig. 4 to Determine the y[n] after sum of product.

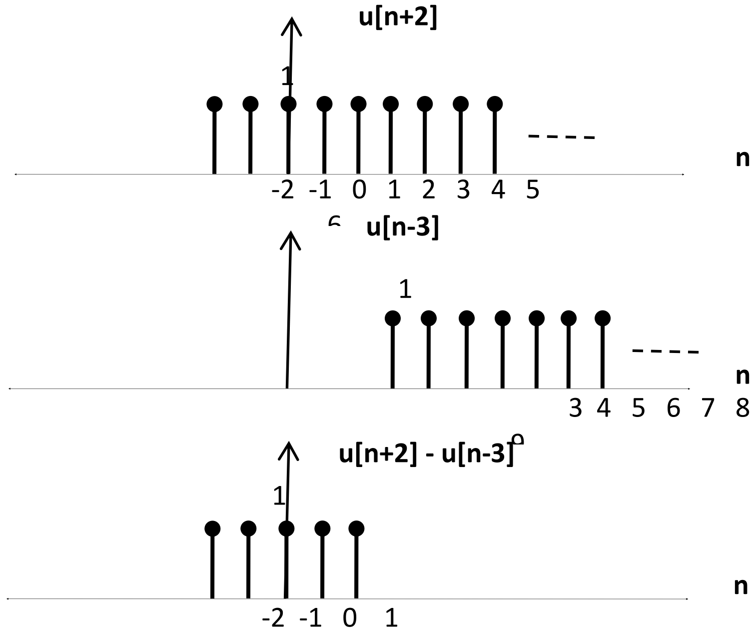

Fig. 3: An example for the finite length

x[n] / h[n]	4	2	1	0.5	0.25
-0.25	-1	-0.5	-0.25	-0.125	-0.0625
-0.5	2	1	0.5	0.25	0.125
-1	-4	-2	-1	-0.5	-0.25
2	8	4	2	1	0.5
-4	-16	-8	-4	-2	-1

Fig. 4. Discrete convolution by fast table method

$$y[n] = [\ -1 \quad 1.5 \quad -3.25 \quad 6.375 \quad -12.8125 \quad -6.375 \quad -3.25 \quad -1.5 \quad -1] \tag{7}$$

When n = -3, -2, -1, 0, 1, 2, 3, 4, 5. As well, Ny = Nx + Nh – 1 = 5+5 – 1 = 9; As show in Fig.5.

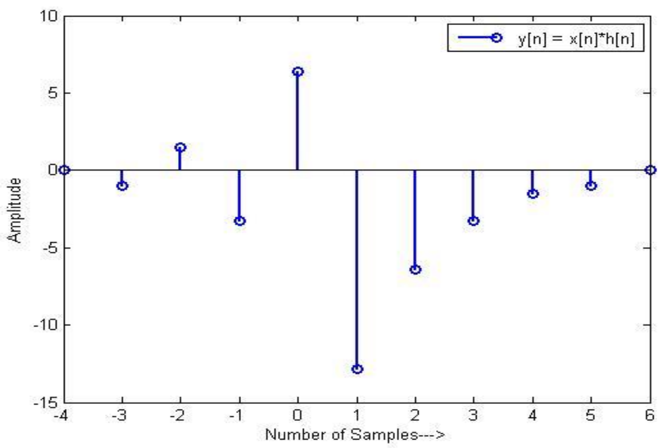

Fig. 5: The convolution output y[n]

Then, the Discrete circular convolution is computed by simple and fast method N_{yc} = Max between N_x and N_h while the length of x,and h; N_{yc} equal 5 as show in Fig. 6, to obtain the $y_c[n]$ in Eq.8.

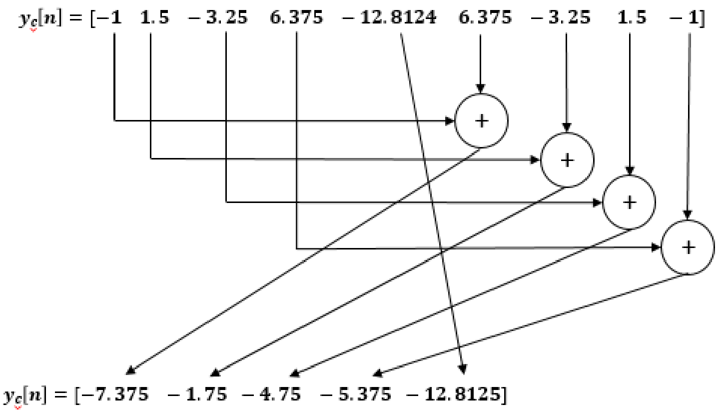

Fig. 6. Discrete circular convolution by Fast Table Method

$$yc[n] = [\ -7.375 \quad -1.75 \quad -4.75 \quad 5.375 \quad -12.8125\ ] \tag{8}$$

Then, more example in same case let x[n] = [1 2 3 4] and h[n] = [5 0 10] where the $N_x = 4$, $N_h = 3$.

$N_y = 4 + 3 - 1 = 6$; $N_{yc} = \text{Max}\ \{3,4\} = 4$ to computing in Fig.7 and Eq.9. The discrete linear correlation can be calculated in the same way as proposed, only the inverse sequence of one of the two inputs x or h as shown in Fig.8. Fig. 9 gives the same results from the second method, calculating the linear correlation R_{xh} in Eq.10. Furthermore, Discrete circular convolution by simple and fast as show in Fig.10, and Eq.11.

x[n] / **h[n]**	**1**	**2**	**3**	**4**
5	5	10	15	20
0	0	0	0	0
10	10	20	30	40

Fig. 7: Discrete linear convolution Nx = 4, Nh = 3, by FTM

x[n] / **h[n]**	**4**	**3**	**2**	**1**
5	20	15	10	5
0	0	0	0	0
10	40	30	20	10

Fig. 8: Discrete linear correlation Nx = 4 , Nh = 3, by FTM

x[n] / **h[n]**	**1**	**2**	**3**	**4**
5	5	10	15	20
0	0	0	0	0
10	10	20	30	40

Fig. 9: Discrete linear correlation another methods, by FTM

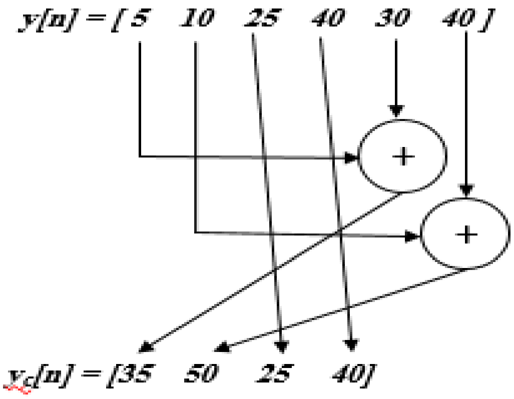

Fig. 10: Discrete circular convolution or correlation Nx = 4 , Nh = 3, by FTM

$$y[n] = [\,5 \quad 10 \quad 25 \quad 40 \quad 30 \quad 40\,] \tag{9}$$

$$Rxh = [20 \;\; 15 \;\; 50 \;\; 35 \;\; 20 \;\; 10] \tag{10}$$

$$yc[n] = [35 \;\; 50 \;\; 25 \;\; 40] \tag{11}$$

4.2. Second case

when the sequence for x[n] or h[n] are finite Length and anther sequacious infinite length

- Find the Linear convolution y[n], with
- Input sequence: $x[n] = \frac{1}{n}(u[n-1] - u[n-5])$ &
- Impulse response: $h(n) = \left(\frac{1}{2}\right)^n u(n).$
- x[n] = [1 ½ 1/3 ¼] ; Nx = 4 (finite length)
- h[n] = [1 ½ ¼ 1/8 1/16 1/32 . . .] (infinite length)

N_y also infinite length; we can the same Fast table methods but teacake 3 to 6 foe first sequence for h as shown the Fig.11.

h[n] \ x[n]	1	1/2	1/3	1/4
1	1	1/2	1/3	1/4
1/2	1/2	1/4	1/6	1/8
1/4	1/4	1/8	1/12	1/16
1/8	1/8	1/16	1/24	1/32
1/16	1/16	1/32	1/48	1/64
1/32	1/32	1/64	1/96	1/128
:	:	:	:	:
1/2n	1*1/2n	1/2*1/2n	1/3*1/2n	1/4*1/2n

Fig. 11. Discrete linear convolution Nx = 4, Nh = ∞ , by FTM

The first sequacious for output y are correct

y[n] = [1 1 5/6 4/6 2/6 1/6 0.5/6 . . .]

therefore, it can be written y[n]:

$$y[n] = \delta[n] + \delta[n-1] + 5/6\,\delta[n-2] + \frac{2^{5-n}}{6}u[n-3] \qquad (12)$$

A radical of notes must be mentioned here. The important thing is, firstly, a quick and easy way to find the linear convolution, which cannot be found in this case by law or by any other method. Secondly, the output y[n] depends significantly on the impulse response h[n] of infinite length, but what is more important is that the output y[n] will be sequential with a direct mathematical relationship depending on the length of the input x[n], which is determined by length n = 4 minus 1.

That is, it will have a length of n = 3 sums with a relationship that is from y = 3 to infinity, as shown in Equation 13. The next example will be more clear; where x[n] = 2n u[n] and h[n] = [1 2 3] using Fast Table Methods in Fig.12:

x[n] = [1 2 4 8 16 32 64 128 . . .] (13a)

h[n] = [1 2 3] (13b)

x[n] / h[n]	1	2	3
1	1	2	3
2	2	4	6
4	4	8	12
8	8	16	24
16	16	32	48
32	32	64	96
64	64	128	192
:	:	:	:
2n	1*2n	2*2n	3*2n

Fig. 12: Discrete linear convolution Nx = ∞, Nh = 3, by FTM

y[n] = [1 4 11 22 44 88 176 352 . . .] =

$$y[n] = \delta[n] + 4\delta[n-1] + 11 * 2^{n-2}u[n-2]. \qquad ...\ (14)$$

First of all, changing the inputs does not affect the output and the results will be the same. It will be also found in y[n] that after 1 and 4 there will be a mathematical equation (14) is $11 * 2^{n-2}u[n-2]$,that connects the numbers from n = 3 - 1 = 2 to the end of the infinite series. Essentially, it is feasible to swiftly and simply identify a mathematical equation for each situation.

4.3. Third case when the sequence for x[n] and h[n] are infinite length

- Find the Linear convolution *y[n],* with input sequence: *x[n] = 3ⁿ u[n]*; therefore

x[n] = [1 3 9 81 . . .] (infinite length); and

- Impulse response: $h[n] = \left(\frac{1}{2}\right)^n u[n]$; *h[n] = [1 ½ ¼ 1/8 1/16 1/32 . . .]* (infinite length).

- The result in this example can used Eq.1 only whit geometric series to find the discrete convolution.

$$y[n] = \sum_{k=0}^{n} x[k] * h[n-k] = \sum_{k=0}^{n} 3^{(k)} * 2^{(-n+k)}$$

$$= 2\text{-n} \sum_{k=0}^{n} 6^{(k)} = \left(\frac{1}{2}\right)^{n} . \left(\frac{1-6^{n}}{1-6}\right). = 0.2\ (3^{n} - 2^{-n}) \qquad (15)$$

But in a case other than the above, which is a number raised to the power of *n*, Equation 1 cannot be solved, and we use our FTM. It will satisfy the purpose and be general for a set of different, infinite entries, as in the following case in equation (16).

$$x[n] = n\,u[n] = [\,0\ \ 1\ \ 2\ \ 3\ \ 4\ \ 5\ \ 6\ \ \ldots\,] \qquad (16a)$$

$$\text{and} \quad h[n] = 2^{n}\,u[n];\ \ h[n] = [\,1\ \ 2\ \ 4\ \ 8\ \ 16\ \ 32\ \ \ldots\,] \qquad (16b)$$

Focusing on the case that the table will be infinited in length or width as Fig.13, and also that the output will be corrected at its beginning, progressing, and finding the missing sequence in Eq.17.

$$y(n) = [\,0\ \ 1\ \ 4\ \ 11\ \ 26\ \ 57\ \ 120\ \ \ldots\,]\ ,$$

$$\text{When } y[0] = 0\ ;\ \text{then }\ y[n+1] = y[n] + (2n+1 - 1) \qquad (17)$$

x[n] / **h[n]**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**. . .**	**n**
	0	0	0	0	0	0	0	. . .	0
1	1	2	3	4	5	6	7	. . .	n*1
2	2	4	6	8	10	12	14	. . .	n*2
4	4	8	12	16	20	24	28	. . .	n*4
8	8	16	24	32	40	48	56	. . .	n*8
16	16	32	48	64	80	96	112	. . .	n*16
32	32	64	96	128	160	192	224	. . .	n*32
64	64	128	192	256	320	384	448	. . .	n*64
:	:	:	⋮	⋮	⋮	⋮	⋮		⋮
2n	1*2n	2*2n	3*2n	4*2n	5*2n	6*2n	7*2n	. . .	n*2n

Fig. 13: FTM to found the discrete linear convolution Nx, Nh and Ny = ∞.

5. DISCUSSION AND IMPLEMENTATION COMPLEXITY

The first case is measured by speed and lack of complexity for both linear and circular discrete convolution, and it is possible to find the percentage of speed and smoothness relative to the traditional method at more than 70% as show in Fig.14. Through the second case, which can only be solved by the proposed Fast Table method, and noted that it concerns linear because the length is infinite and is equivalent between linear and circular. In addition, the method is fast, comprehensive, and reduces complexity. It is clear from this that the specified input length,n-1, is followed by a comprehensive mathematical equation in y used for any length, such as 10, 100, etc., as explain in Fig.10 and 11 and Eq.12 and 14 and achieves flexibility and high accuracy.

The third case does not differ much from the previous one, although any other traditional method will not be able to calculate the output as the table method calculates it, and with any different, non-standard input signal, Equation No. 1 may be able to solve it. The facilitation provided by our method is shown in Fig.12 and Eq.16. It is shown how it is possible to write a simplified mathematical relationship for this case, with different input signals, and achieve great flow.

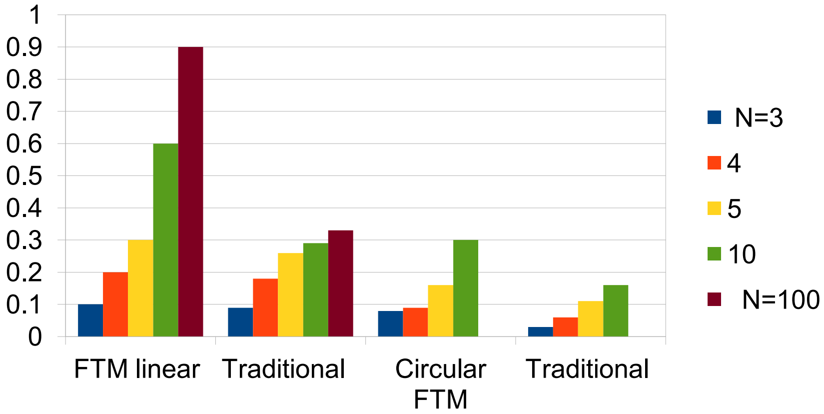

Fig. 14. Percentage reduction in complexity and time for the FTM relative to the traditional

6. CONCLUSIONS

In this paper, the potential was examined for lowering the exponential complexity of computing linear and circular convolution of small-length input sequences and developing a workable solution for large- and infinite-length input sequences. Also, it was assembled three cases to carry out these operations for $N = 3, 4, 5$, and infinite (∞). By using these algorithms, the hardware implementation of linear convolution is reduced in terms of computational complexity. The suggested algorithms have a pronounced parallel modular structure, which simplifies mapping algorithms and lowers complexity as well. As a result, computations are accelerated. This can also be accomplished during the execution of these algorithms because of parallelism in mathematical calculations.

7. REFERENCES

A. E. Abdelkareem, Convolutional Code Constraint Length over Rayleigh Fading Channel: Performance Evaluation and Hardware Aspects, Iraqi Journal of Computers, Communication and Control & System Engineering (IJCCCE), Vol. 17, No. 1, November 2017.

A. Lavin and S. Gray, "Fast Algorithms for Convolutional Neural Networks,", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 4013-4021.

A. T. Alrubei, Mohammed, and Pozdnyakov Alexander Dmitrievich. “AN APPROACH FOR SINGLE-TONE FREQUENCY ESTIMATION USING DFT INTERPOLATION WITH PARZEN WINDOWING”. Kufa Journal of Engineering, vol. 14, no. 3, Aug. 2023, pp. 93-104, doi:10.30572/2018/KJE/140307.

Abdul-Jaleel Al-Asady, Heba, et al. “AN IMAGE ENCRYPTION METHOD BASED ON LOGISTICAL CHAOTIC MAPS TO ENCRYPT COMMUNICATION DATA”. Kufa Journal of Engineering, vol. 15, no. 4, Nov. 2024, pp. 55-64, doi:10.30572/2018/KJE/150405.

Abtahi, T.; Shea, C.; Kulkarni, A.; Mohsenin, T. Accelerating convolutional neural network with fft on embedded hardware. IEEE Trans. Very Large Scale Integr. (VLSI) Syst. 2018, 26, 1737–1749.

Alkadhim, Saif Aldeen Saad, Digital Convolution with Digital Signal Processing (DSP) (July 9, 2020). Available at SSRN: https://ssrn.com/abstract=3647517.

Bi, G.; Zeng, Y. Transforms and Fast Algorithms for Signal Analysis and Representations; Springer Science & Business Media: New York, NY, USA, 2004.

C. Li, H. K. Kwan and X. Qin, "Revisiting Linear Convolution, Circular Convolution and Their Related Methods," 2020 13th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Chengdu, China, 2020, pp. 1124-1131.

Cariow, A.; Cariowa, G. Minimal Filtering Algorithms for Convolutional Neural Networks. ArXiv 2020, arXiv:2004.05607.

Chan, Y.H.; Siu, W.C. General approach for the realization of DCT/IDCT using convolutions. Signal Process.1994,p.p 37, 357–363.

Ghasemi, S. Hooman; Nowak, Andrzej S. , "Reliability Index for Non-normal Distributions of Limit State Functions", Structural Engineering and Mechanics, 62 (3): 365–372, 2017.

Grinshpan, A. Z. , "An inequality for multiple convolutions with respect to Dirichlet probability measure", Advances in Applied Mathematics, 82 (1): 102–119, 2017.

Hsu, H. (2011). Schaum’s Outline of Signals and Systems (2nd ed.). New York, NY: McGraw-Hill.

Ju, C.; Solomonik, E. Derivation and Analysis of Fast Bilinear Algorithms for Convolution. ArXiv 2019, arXiv:1910.13367.

Krishna, H. Digital Signal Processing Algorithms: Number Theory, Convolution, Fast Fourier Transforms, and Applications; Routledge: New York, NY, USA, 2017.

Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. Commun. ACM 2017, 60, 84–90.

Lavin, A.; Gray, S. Fast algorithms for convolutional neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 4013–4021.

Lin, S.; Liu, N.; Nazemi, M.; Li, H.; Ding, C.; Wang, Y.; Pedram, M. FFT-based deep learning deployment in embedded systems. In Proceedings of the 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 19–23 March 2018; pp. 1045–1050.

M. S. Naghmash, M. K. Wali, A. A. Abdulmajeed,High Level Implementation Methodologies of DSP Module using FPGA and System Generator, Eng. &Tech.Journal, Vol.34,Part (A), No.2, 2016.

Mathieu, M.; Henaff, M.; LeCun, Y. Fast training of convolutional networks through ffts. ArXiv 2013, arXiv:1312.5851.

N.Vasilache, J. Johnson, M. Mathieu, S. Chintala, S. Piantino, and Y. LeCun. Fast convolutional nets with fbfft: A GPU perfor-mance evaluation. CoRR, abs/1412.7580, 2014.

Parhi, K.K. VLSI Digital Signal Processing Systems: Design and Implementation; John Wiley & Sons: New York, NY, USA, 2007.

Pratt, W.K. Digital Image Processing; John Wiley & Sons: Hoboken, NJ, USA, 2007.

Roberts, M.J. (2012). Signals and Systems: Analysis Using Transform Methods and MATLAB® (2nd ed.). New York, NY: McGraw-Hill.

W.A. mahmoud; M. N.Hussainn, Wavelet and Wavelet Packet Transform Realization Using FPGA, Iraqi Journal of Computers, Communication and Control & System Engineering IJCCCE, Vol.6, No.2, 2006.

Wang, X.; Wang, C.; Zhou, X. Work-in-progress: WinoNN: Optimising FPGA-based neural network accelerators using fast winograd algorithm. In Proceedings of the 2018 International Conference on Hardware/Software Codesign and System Synthesis (CODES+ ISSS), Turin, Italy, 30 September–5 October 2018; pp. 1–2.