Computing Cross and Auto-Correlation Sequences of Output WSS Real Random Process in Linear Time

Ejaz A. Ansari*
Department of Electrical Engineering, COMSATS Institute of Information Technology
1.5 km Defense Road, off Raiwind Road, 53700, Lahore, Pakistan
***corresponding author's email: {dransari@ciitlahore.edu.pk}**

ABSTRACT: This paper proposes a ***new, simple*** and ***efficient*** technique in discrete-time domain which does the processing of auto correlation of the input discrete-time wide sense stationary (*WSS*) process through ***exponential, rectangular, triangular*** and ***trapezoidal*** shaped discrete time windows in ***linear*** time. The proposed algorithm also performs the ***Low Pass Filtering*** (*LPF*) of the input auto correlation in ***linear*** time since the frequency responses of the above mentioned discrete-time windows very much ***resemble*** with the frequency response of low pass filters. First, we develop the proposed algorithm for a discrete-time exponential shaped window having '***M***' samples and show that it takes ***O***(***L***) arithmetic operations (i.e., both *+s* and **s*) in processing the input auto correlation through it, where '***L***' is the size of input auto correlation sequence. We then, make use of this result in processing the input auto correlation through a rectangular shaped (moving average) window by incorporating a ***slight*** modification in it. Subsequently, we process the input auto correlation through triangular and trapezoidal shaped discrete-time windows using the modified result of the proposed algorithm. We thus show that our proposed algorithm ***only*** requires ***O***(***L***) ***additions*** to perform the low pass filtering of the input auto-correlation through the ***last three*** windows mentioned above. We also show that the proposed algorithm ***just*** takes ***2 x*** (***L - 1***) ***additions*** to perform the *LPF* of the input auto correlation through a rectangular shaped window and it remains ***independent*** with respect to the change in the size of the window, i.e., ***M***. Finally, we also show that our proposed algorithm ***outperforms*** all the existing techniques in ***both*** time and frequency domains with regards to ***exact*** number of arithmetic operations and its ***worst case*** time complexity grows ***linearly*** with the length of the input auto correlation sequence for the case when ***L >> M***.

Key Words: Auto correlation, Cross correlation, Efficient Filtering, Finite shaped Windows, Time Complexity

1 INTRODUCTION

Performing filtering operation on the input data sequence formulates an important part of Digital Signal Processing (DSP). There are plenty of application examples that require this operation in certain kind, for example in the area of medical image and video processing, radar and sonar signal processing, statistical signal processing and modern communication systems employing coherent receivers [1], [2]. Convolution of input data sequence through an *FIR* linear shift invariant system (*LSI*) basically implements both filtering and smoothing operations on the input data. Although, convolution in time domain has no processing delays but the required computational cost using convolution sum makes it highly impractical. Thus, it is implemented in frequency domain using *FFT/IFFT* based overlap-add and overlap-save methods. However, efficiency is achieved at the expense of the delay introduced equal to the filter length [3], [4].

The processing of discrete-time *WSS* random processes through *FIR LSI*-systems plays a very useful role in studying the behavior of many wireless communication systems and networks. Filtering operation does not bring about any change in the stationarity status of the input process and thus both input and output processes become jointly stationary [5], [6]. Frequency responses of all discrete-time windows discussed in this paper correspond to the frequency response of low pass filters. It is a well known fact that low pass filtering of the *WSS* input processes brings smoothness, removes noise buried in the uncorrupted input data and blurness present in the underlying image respectively, etc [6]-[8]. To the best of our knowledge, there exists no algorithm so far in the literature either in time or frequency domain that performs low pass filtering of the input data sequence through FIR shaped windows in linear time. We have already discussed the basic idea of the proposed algorithm in analog domain in [5]. We already know that computation of auto- and cross-correlation of discrete-time output process becomes very important for further analysis of the system. In most situations, transform techniques are normally employed for such computations but the analysis becomes difficult and challenging once the input process does not remain a stationary white noise as demonstrated in [5]. We, therefore, need to look for some alternate simple and efficient technique in time domain that could provide simpler solution in a much faster way.

In this paper, we, thus present a very simple and efficient method in time domain for computing cross and auto-correlation of the discrete-time output process through *FIR LSI*-windows. Different discrete-time *FIR* windows discussed in this paper are exponential, rectangular (moving average), triangular and trapezoidal shaped ones.

This paper is organized as follows*: Section 2 presents the* *equivalent relations of computing cross and auto correlations of the output WSS process in discrete time domain. Section 3 describes the development of the proposed algorithm for computation of cross and auto correlations of the output process in case of an exponential window. Time complexity of the proposed algorithm is discussed in section 3. Section 4 describes how the proposed algorithm with slight modification can be used for processing of WSS input process through other three types of windows. We also discuss the time complexity of our modified proposed algorithm in this section. Comparison of the proposed technique with other existing algorithms is discussed in section 5. Finally, we present our conclusions in section 6.*

2 Equivalent relations for Cross and Auto Correlations in Discrete-Time Domain

- ***For two deterministic discrete signals***

We have already derived the relations for cross and auto correlations of continuous-time *WSS* output process in [5]. Using the same approach and knowing the fact about summation operation as alternate of integral operation, we may express the equivalent relations of [eq.(***1***) – (***6***), 5] in discrete-time domain as

$$R_{xy}(k) = \sum_{n=-\infty}^{\infty} x(k+n)\,y(n) = \boxed{R_{yx}(-k)} \qquad (\mathbf{1})$$

Equation (***1***) is a familiar expression and may thus be expressed as discrete convolution of $x(-k)$ with $y(k)$. Hence,

$$R_{xy}(k) = x(-k) * y(k) = y(-k) * x(k) \qquad (\mathbf{2})$$

where $*$ denote discrete convolution operation between $x(k)$ and $y(-k)$, Similarly the autocorrelation of the input signal, $x(k)$ can be obtained by replacing $y(k) = x(k)$ in eq. (***2***) as $R_{xx}(k) = x(k) * x(-k) = x(-k) * x(k) = R_{xx}(-k)$, which follows from eq. (***1***). This shows that $R_{xx}(k)$ is an even and real function of time whose Fourier transform is also an even and real function of frequency , [6]-[7]. When an input signal $x(n)$ is passed through an *LTI*-system having $h(n)$ as its impulse response, the output, $y(n)$ generated by it may be computed [7] as $y(n) = h(n) * x(n) = x(n) * h(n)$. We thus, substitute this result in eq. (***2***) and express the cross correlation between $x(n)$ and $y(n)$ as

$$\begin{aligned} R_{xy}(k) &= x(k) * y(-k) = x(k) * \{x(-k) * h(-k)\} \\ &= \{x(k) * x(-k)\} * h(-k) = R_{xx}(k) * h(-k) = h(-k) * R_{xx}(k) \end{aligned} \qquad (\mathbf{3})$$

Likewise, we may also compute autocorrelation of the output *WSS* process in discrete domain using eq. (***2***) as

$$\begin{aligned} R_{yy}(k) &= y(k) * y(-k) = \{h(k) * x(k)\} * y(-k) = h(k) * \{x(k) * y(-k)\} \\ &= h(k) * R_{xy}(k) = \{h(k) * h(-k)\} * R_{xx}(k) = R_{hh}(k) * R_{xx}(k) \end{aligned} \qquad (\mathbf{4})$$

Where $R_{hh}(k)$ denote autocorrelation of an *LTI*- discrete system with $h(n)$ as its impulse response.

- ***For two jointly WSS real Discrete Random Processes***

The auto correlation of a *WSS* real input random process, $\boldsymbol{X}(n)$ can be defined as [9], $R_{XX}(k) = E[X(n+k)X(n)]$. Similarly, cross correlation between the input, $\boldsymbol{X}(n)$ and the output $\boldsymbol{Y}(n)$ of two real *WSS* processes can be defined as $R_{XY}(k)=E[X(n+k)Y(n)] = E[Y(n)X(n+k)]$. The output process, $\boldsymbol{Y}(n)$ obtained after filtering the input pr0cess, $\boldsymbol{X}(n)$ as

$$\boldsymbol{Y}(n) = h(n) * \boldsymbol{X}(n) = \sum_{k=-\infty}^{\infty} h(k)\,\boldsymbol{X}(n-k) \qquad (\mathbf{5})$$

Using this definition of convolution sum and cross correlation, we may express $R_{XY}(k)$ after following some simplification and manipulation steps as convolution of $h(-k)$ and $R_{XX}(k)$ like eq. (***3***). Similarly, we may use eq.(***5***) in the definition of auto correlation of the output process, $\boldsymbol{Y}(n)$ and after doing some simplification and manipulation steps, we can also express $R_{YY}(k)$ as convolution of $R_{hh}(k)$ and $R_{XX}(k)$ respectively like eq. (***4***). This shows that auto correlation of the input process, $\boldsymbol{X}(n)$ plays a vital role in determining the cross and auto correlations of the output process, $\boldsymbol{Y}(n)$.

- ***Transformation of relations derived in eqs.(3)-(4) to frequency domain***

If the Fourier transforms of auto and cross correlation functions are represented by $S_{XX}(e^{jw})$ and $S_{XY}(e^{jw})$, then we may consider $S_{XX}(e^{jw})$, $S_{YY}(e^{jw})$, $S_{hh}(e^{jw})$ and $H(e^{jw})$ as energy spectral densities present in the input process, output process and the filter and frequency response of the filter respectively. Knowing the fact that discrete convolution operation in time domain transforms to simple multiplication operation in the frequency domain, the time domain relations described in eqs. (***3***)-(***4***) can easily be transformed to frequency domain as

$$S_{XY}(e^{jw}) = H(e^{-jw})\,S_{XX}(e^{jw}) \text{ and } S_{YY}(e^{jw}) = S_{hh}(e^{jw})\,S_{XX}(e^{jw}) \qquad (\mathbf{6})$$

where $S_{hh}(e^{jw}) = H(e^{jw}) \times H(e^{-jw}) = |H(e^{jw})|^2$ as $H(e^{-jw}) = H^*(e^{jw})$ because $h(n)$ is real.

3 DEVELOPMENT OF THE PROPOSED ALGORITHM FOR EXPONENTIAL SHAPED WINDOWS

Consider $R_{XX}(k)$ has *'L'* samples where $0 \le k \le L-1$ and is symmetric at $k = L/2 - 1$. An exponential shaped window, $e(k)$ having M samples is defined as, $e(k) = (a)^k$; $0 \le k \le M-1$, where $|a| \le 1$. The processing of auto correlation, $R_{XX}(k)$ through $e[k]$ provides us cross correlation of the output *WSS* discrete-time process, $\boldsymbol{Y}(k)$ through it. We derive the desired relation of cross correlation, $R_{YXexp}(k)$ of the output process with input process using eq.(***4***) and eq.(***5***) as

$$\begin{aligned} \boldsymbol{R}_{YXexp}(k) &= e(k) * \boldsymbol{R}_{XX}(k) = \sum_{n=-\infty}^{\infty} \boldsymbol{R}_{XX}(n)\,e(k-n) \\ &= \sum_{n=0}^{k} \boldsymbol{R}_{XX}(n)\,e(k-n) = \sum_{n=0}^{k-1} \boldsymbol{R}_{XX}(n)\,e(k-n) + \boldsymbol{R}_{XX}(k) \end{aligned} \qquad (\mathbf{7})$$

Equation (***7***) is valid for $k \ge 0$ and $R_{YXexp}(k)$ is zero when $k < 0$. We recognize that the first term on the *RHS* of eq.(***7***) can be written in terms of cross correlation of the output process with the input process as $R_{YXexp}(k-1)$ using the definition of $e(k)$ in the same equation. Thus, we express eq.(***7***) in a more attractive (recursive) form as

$$\boldsymbol{R}_{\boldsymbol{YXexp}}\left(k\right)=a\boldsymbol{R}_{\boldsymbol{YXexp}}\left(k-1\right)+\boldsymbol{R}_{\boldsymbol{XX}}\left(k\right) \qquad (\boldsymbol{8})$$

Equation (***8***) tells us how to compute $\boldsymbol{R}_{YXexp}(k)$ *recursively* from the knowledge of its previous value for all values of k. $\boldsymbol{R}_{YXexp}(k) = \boldsymbol{R}_{XX}(k)$ for $k = 0$. It is $\boldsymbol{R}_{YXexp}(k) = a\boldsymbol{R}_{YXexp}(k-1) + \boldsymbol{R}_{XX}(k)$ for $1 \leq k \leq L-1$ and finally it is, $\boldsymbol{R}_{YXexp}(k) = a\boldsymbol{R}_{YXexp}(k)$ for $L \leq k \leq N-1$, where N represent the no. of samples in the output cross correlation sequence computed from [8] as ($N = L + M - 1$). For all other values of k, i.e., ($k \geq N$ and $k < 0$), $\boldsymbol{R}_{YXexp}(k) = 0$.

As $\boldsymbol{R}_{XYexp}(k) = \boldsymbol{R}_{YXexp}(-k)$, just flip the output cross correlation sequence computed from eq. (***8***) about $k = 0$ axis in order to get $\boldsymbol{R}_{XYexp}(k)$. This operation provides the sequence values of $\boldsymbol{R}_{XYexp}(k)$ for $-N+1 \leq k \leq 0$. Similarly, we can derive another recursive relation of auto correlation of the output sequence, $\boldsymbol{R}_{YYexp}(k)$ similar to eq. (***8***) using eq. (***4***) and (***5***) following the same process as

$$\boldsymbol{R}_{\boldsymbol{YYexp}}\left(k\right)=a\boldsymbol{R}_{\boldsymbol{YYexp}}\left(k-1\right)+\boldsymbol{R}_{\boldsymbol{XYexp}}\left(k\right) \qquad (\boldsymbol{9})$$

The no. of samples in the output auto correlation sequence computed from [8] as ($N_1 = N + M - 1$). However, in this case, $\boldsymbol{R}_{YYexp}(k) = \boldsymbol{R}_{XY}(k)$ for $k = (-N+1)$. It is, $\boldsymbol{R}_{YYexp}(k) = a\boldsymbol{R}_{YYexp}(k-1) + \boldsymbol{R}_{XY}(k)$ for $-N+2 \leq k \leq 0$ and *finally* it is, $\boldsymbol{R}_{YYexp}(k) = a\boldsymbol{R}_{YYexp}(k)$ for $1 \leq k \leq M-1$. For all other values of k, *i.e.*, ($k \geq M$ and $k \leq -N$), $\boldsymbol{R}_{YYexp}(k) = 0$. It is important to note here that contrary to the sample sequence values of $\boldsymbol{R}_{YXexp}(k)$, which exist only for positive values of k, we do get the sample values of the sequence, $\boldsymbol{R}_{YYexp}(k)$ present both for negative and positive values of k. This makes the sequence *non-causal* and it happened due to the fact that we have flipped the sequence, $\boldsymbol{R}_{YXexp}(k)$ about $k = 0$ in order to derive eq. (***9***). Hence, we need to introduce a delay equal to $k = N-1$ in order to shift the sample sequence, $\boldsymbol{R}_{YYexp}(k)$ for only positive values of k (*i.e.*, in order to make it causal).

4 DETERMINING THE TIME COMPLEXITY OF THE PROPOSED ALGORITHM

We can easily compute the ***total exact*** number of arithmetic operations (i.e., $+s$ and $*s$) taken by our proposed algorithm and its ***worst case*** total time complexity using eqs. (8) and (9). As demonstrated in section 3, we require ($\boldsymbol{L-1}$) multiplications and ($\boldsymbol{L-1}$) additions in order to compute the sample values of $\boldsymbol{R}_{YXexp}(k)$, for $1 \leq k \leq L-1$. In addition, we further require ($\boldsymbol{N-L}$) multiplications in order to compute $\boldsymbol{R}_{YXexp}(k)$, for $L \leq k \leq N-1$. Thus, in order to compute all sample values of $\boldsymbol{R}_{YXexp}(k)$, we thus, require ***exactly*** ($\boldsymbol{L-1}$) additions and $\{(L-1) + (N-L) = (N-1) = (\boldsymbol{L+M-2})\}$ multiplications in evaluating $\boldsymbol{R}_{YXexp}(k)$ using eq. (***8***) recursively.

Similarly, we require ($\boldsymbol{N-1}$) multiplications and ($\boldsymbol{N-1}$) additions in order to compute the sample values of $\boldsymbol{R}_{YYexp}(k)$, for $-N+2 \leq k \leq 0$. In addition, we further require ($\boldsymbol{M-1}$) multiplications in order to compute $\boldsymbol{R}_{YXexp}(k)$, for $1 \leq k \leq M-1$. Thus, in order to compute all sample values of $\boldsymbol{R}_{YYexp}(k)$, we thus, require ***exactly*** ($\boldsymbol{N-1 = L+M-2}$) additions and $\{(N-1) + (M-1) = (N+M-2) = (\boldsymbol{L+2M-3})\}$ multiplications in evaluating $\boldsymbol{R}_{YYexp}(k)$ using eq. (***9***) recursively. We also note that, in Matlab, a delay of certain length in the sequence is usually implemented by appending a vector of zeros in front of the sequence ***without*** incurring any arithmetic operations.

In total, we exactly require $\{(L-1) + (N-1) = (\boldsymbol{2L+M-3})\}$ additions and $\{(N-1) + (N+M-2) = (2N+M-3) = (\boldsymbol{2L+3M-5})\}$ multiplications to compute all sample values of $\boldsymbol{R}_{YXexp}(k)$ and $\boldsymbol{R}_{YYexp}(k)$ altogether respectively for our proposed algorithm in case of an exponential window. These exact arithmetic operations (i.e., $+s$ and $*s$) are required to do low pass filtering of input auto correlation, $R_{XX}(k)$ having length, L through an exponential shaped window of size, M. We also notice here that under the assumption, when the length of input auto correlation sequence becomes very large than the size of an exponential window, (i.e., when $\boldsymbol{L >> M}$), the ***worst case*** total time complexity of our proposed algorithm according to [10], just reduces to $\boldsymbol{O(L)}$ which shows that it ***linearly*** grows with the length of input auto correlation sequence, $R_{XX}(k)$.

5 PROCESSING (LOW PASS FILTERING) OF $R_{XX}(k)$ THROUGH OTHER LTI- WINDOWS

A. *Rectangular (Moving Average) Shaped Window*

We consider *'M'* samples of the rectangular window, $r(n)$ and it is defined as, $r(k) = 1/M$; for $0 \leq k \leq M-1$ and zero otherwise. We can express it as a linear combination of unit step sequence and its shifted version of sequence as, $r(k) = 1/M\ x\{u(k) - u(k-M)\}$. Comparison of the definition of an exponential window with a rectangular window hence, allows us to write it more suitable form as,

$$r\left(k\right)=M^{-1}\lim_{a\to 1}\left\{a^{k}u\left(k\right)-a^{k-M}u\left(k-M\right)\right\}=M^{-1}\lim_{a\to 1}\left\{e\left(k\right)-e\left(k-M\right)\right\} \quad (\boldsymbol{10})$$

Substitution of eq. (***10***) in eq. (***8***) and eq. (***9***) respectively and further utilizing ***LSI*** property of the system, we thus succeed in performing the low pass filtering of the input auto correlation of ***WSS*** process through a rectangular shaped window, $r(k)$ of size M. The compulsory steps of the proposed algorithm are already mentioned in [6] with minor modifications in step *v)* and *vi)* of [section 4 (b), 6]. Take the limit of the result when 'a' tends to 1 instead of being zero and scale down the sample values by M instead of T. This modification immediately reveals that we in fact require ***no*** multiplication operation (as a is tending to 1in case of $r(k)$) in performing low pass filtering of $R_{XX}(k)$ through $e(k)$ using eq. (***8***) and (***9***) respectively. This will generate $\boldsymbol{R}_{YXrect}(k)$ and $\boldsymbol{R}_{YYrect}(k)$ without incurring any multiplication operation.

However, we need only to shift the results of output cross and auto correlation sequences by *'M'* samples and need to subtract the delayed/shifted sequences from the results of original/un-delayed output cross and auto correlation sequences according to eq. (***10***). Thus, it incurs extra subtraction operations in addition to total exact number of addition operation taken by our proposed algorithm.

These subtraction operations are also usually implemented via addition operations using gates. The additional burden of these extra subtraction operations used by our proposed algorithm can easily be computed from the knowledge of the lengths of output cross and auto correlation sequences already mentioned in section 3. Thus, we require extra $(N - M) = (L - 1)$ addition operation in order to compute all sample values of $\boldsymbol{R_{YXrect}}(k)$ using eqs. (***8***) and (***10***). In total, we exactly require, $\{(L-1) + (L-1) = 2 \times (\boldsymbol{L-1})\}$, addition operations only to compute $\boldsymbol{R_{YXrect}}(k)$ which are independent to the size, M of $r(k)$. Similarly, we require extra $(N_1 - M) = (N - 1)$ addition operation in order to compute all sample values of $\boldsymbol{R_{YYrect}}(k)$ using eqs. (***9***) and (***10***). In total, we exactly require, $\{(N-1) + (N-1) = 2 \times (\boldsymbol{N-1})\}$, addition operations only to compute $\boldsymbol{R_{YYrect}}(k)$. Thus, we notice that in order to perform low pass filtering and to compute all sample values of output cross and auto correlation sequences, we on the whole, exactly require $2 \times (L-1) + 2 \times (N-1) = 2 \times (L + N - 2) = 2 \times (\boldsymbol{2L + M - 3})$ addition operations. Hence, we conclude that, in total, our proposed algorithm requires exactly *twice* addition operations only to perform low pass filtering of $R_{XX}(k)$ through $r(k)$ as compared to $e(k)$. No multiplication operations are required for this purpose.

B. Triangular Shaped Window with length = 2M+1

We know from [5] that a triangular shaped window, $\Delta(k)$ of length, $2M + 1$ in discrete-time domain is defined as: $\Delta(k) = k$; for $0 \le k \le M - 1$ and $\Delta(k) = 2M - k$; for $M \le k \le 2M$ and zero otherwise. A triangle shaped window in discrete-time domain can easily be generated by performing discrete convolution of rectangular shaped window, $r(k)$ with itself. But, this operation will generate a sample of value 1 at $k = 0$ instead of zero. We can circumvent this issue by introducing a ***delay*** of just one sample in the output result. Thus, by incorporating *LTI* property of the system, we can express mathematically $\Delta(k)$ in terms of $r(k)$ as, $\Delta(k) = r(k) * r(k - 1)$. The processing of $\boldsymbol{R_{XX}}(k)$ through this window will provide us cross and auto correlations of $\boldsymbol{Y(k)}$ using eq. (***3***) as

$$
\begin{aligned}
\boldsymbol{R_{YX\Delta}}(k) &= \Delta(k) * R_{XX}(k); \\
&= \{r(k-1) * r(k)\} * R_{XX}(k) \\
&= r(k-1) * \underbrace{\left(r(k) * R_{XX}(k)\right)}_{R_{YXrect}(k)} = \boxed{\left(r(k-1) * R_{YXrect}(k)\right)} \\
\boldsymbol{R_{YY\Delta}}(k) &= \Delta(k) * R_{XYrect}(k); \qquad \boxed{R_{XYrect}(k) = R_{YXrect}(-k)} \\
&= \{r(k-1) * r(k)\} * R_{XYrect}(k) \\
&= r(k-1) * \underbrace{\left(r(k) * R_{XYrect}(k)\right)}_{R_{YYrect}(k)} = \boxed{\left(r(k-1) * R_{YYrect}(k)\right)}
\end{aligned} \quad (\boldsymbol{11})
$$

Equation (***11***) shows that cross and auto correlation sequences of the output process, $\boldsymbol{Y(k)}$ can easily be obtained by applying the proposed algorithm twice in processing the input auto correlation through triangular shaped window. The exact no. of additions taken by proposed algorithm can easily be computed using the results of section 5 (A). We require exactly $2 \times (L-1) + 2 \times (N-1) = 2 \times (L + N - 2) = 2 \times (\boldsymbol{2L + M - 3})$ addition operations in generating the sequence, $\boldsymbol{R_{YX\Delta}}(k)$. In addition, we also require $2 \times (N-1) + 2 \times (N_1 - 1) = 2 \times (N + N_1 - 2) = 2 \times (\boldsymbol{2N + M - 3}) = 2 \times (\boldsymbol{2L + 3M - 5})$ more addition operation to generate the sequence, $\boldsymbol{R_{YY\Delta}}(k)$. On the whole, we require exactly $8 \times (\boldsymbol{L + M - 2})$ addition operations in performing low pass filtering of $\boldsymbol{R_{XX}}(k)$ through $\boldsymbol{\Delta}(k)$.

C. Trapezoidal shaped Window with width, $W = M_1 + M_2$

Likewise, a trapezoidal shaped window, *trap*(k) of size $(W + 1)$ with maximum value of the sample equal to M_2 is defined as, *trap*$(k) = k$ for $0 \le k \le M_2$; for the interval $M_2 + 1 \le k \le M_1$, it is equal to M_2; and finally it is equal to $(W - k)$ for $M_1 + 1 \le k \le W$ and zero elsewhere, where $W = M_1 + M_2$ and $M_1 > M_2$. Actually M_1 and M_2 represent the sizes of two rectangular shaped windows, $r_1(k)$ and $r_2(k)$ used to generate trapezoidal shaped window of length W. We express trap(k) as convolution of $r_1(k)$ and $r_2(k)$ in mathematical form as trap(k) = $r_1(k) * r_2(k - 1)$, where $r_i(k) = u(k) - u(k - M_i)$; for $i \varepsilon (1,2)$.

$$
\begin{aligned}
\boldsymbol{R_{YXtrap}}(k) &= trap(k) * R_{XX}(k); \\
&= \{r_2(k-1) * r_1(k)\} * R_{XX}(k) \\
&= r_2(k-1) * \underbrace{\left(r_1(k) * R_{XX}(k)\right)}_{R_{YXrect}(k)} = \boxed{\left(r_2(k-1) * R_{YXrect}(k)\right)} \\
\boldsymbol{R_{YYtrap}}(k) &= trap(k) * R_{XYrect}(k); \qquad \boxed{R_{XYrect}(k) = R_{YXrect}(-k)} \\
&= \{r_2(k-1) * r_1(k)\} * R_{XYrect}(k) \\
&= r_2(k-1) * \underbrace{\left(r_1(k) * R_{XYrect}(k)\right)}_{R_{YYrect}(k)} = \boxed{\left(r_2(k-1) * R_{YYrect}(k)\right)}
\end{aligned} \quad (\boldsymbol{12})
$$

Here, again we note from eq. (***11***) that cross and auto correlation sequences of the output process, $\boldsymbol{Y(k)}$ can easily be obtained by applying the proposed algorithm twice in processing the input auto correlation through trapezoidal shaped window. The exact no. of additions taken by our proposed algorithm can easily be computed using the results of section 5 (B). We thus require, exactly $2 \times (\boldsymbol{2L + M_1 - 3})$ addition operations in generating the sequence, $\boldsymbol{R_{YXtrap}}(k)$. In addition, we also require $2 \times (N + N_1 - 2) = 2 \times (\boldsymbol{2N + M_2 - 3}) = 2 \times (\boldsymbol{2L + 2M_1 + M_2 - 5})$ more addition operation to generate the sequence, $\boldsymbol{R_{YYtrap}}(k)$. On the whole, we require exactly $2 \times (\boldsymbol{4L + 3M_1 + M_2 - 8})$ addition operations in performing low pass filtering of $\boldsymbol{R_{XX}}(k)$ through ***trap***(k). We note here that these exact addition operations reduce to the case of triangular shaped window for $\boldsymbol{M_1 = M_2}$. Thus, processing and/or low pass filtering of $R_{XX}(k)$ through all types of windows except exponential one discussed in the paper actually requires ***no*** multiplication at all in getting the job done. We also conclude from the discussion of section 5 {(A) – (C)} and [10] that under the valid and logical assumption for ***L >>M***, low pass filtering of the input auto correlation can be achieved in linear time, i.e., ***O***(***L***) in all types of windows discussed in the paper. This grows linearly with the length, *L* of the input auto correlation sequence, $R_{XX}(k)$.

6 COMPARISON OF PROPOSED ALGORITHM WITH EXISTING ONES

If we use convolution sum expressed in eq. (***5***) to perform low pass filtering of input auto correlation sequence having length, ***L*** through an exponential shaped window of size ***M***, we then exactly require (***LM***) real multiplications & (***L – 1) x (M – 1***) real additions. The ***worst*** case time complexity of this algorithm in time domain under the assumption, ***L >>M*** is ***O***(***LM***). On the other hand, if we use fast Fourier transform (*FFT*) radix-2 algorithm to implement low pass filtering of the input auto correlation sequence through the same exponential window in frequency domain, then we require, $\boldsymbol{0.5N\log_2(N)}$ ***complex*** multiplications and $\boldsymbol{N\log_2(N)}$ ***complex*** additions to do so, where $\boldsymbol{N \geq L + M - 1}$ represent the size of ***DFT*** and ***IDFT*** and must be a power of 2. Further utilizing the fact ***four*** real multiplications and ***two*** real additions are required to implement ***one*** complex multiplication. Similarly, ***two*** real additions are required to implement ***one*** complex addition. Thus, we realize that in order to implement low pass filtering of the input auto correlation sequence in frequency domain, we actually require $\boldsymbol{2N\log_2(N)}$ real multiplications plus $\boldsymbol{3N\log_2(N)}$ real additions,. The ***worst*** case time complexity of this algorithm under the assumption, ***L>>M*** is $\boldsymbol{O(L\log_2 L)}$. We immediately infer from [10] that $\boldsymbol{O(L) < O(L\log_2 L) < O(LM)}$. We thus conclude from the above discussion that our proposed algorithm performs low pass filtering of the input auto correlation sequence through different ***FIR LSI*** windows much better than both existing time and frequency domain algorithms in terms of both ***worst*** case time complexity and ***exact*** no. of arithmetic operations.

7 CONCULUSIONS

In this paper, we presented a very ***simple*** and ***efficient*** technique in discrete-time domain for performing low pass filtering (*LPF*) of the auto correlation of the input process through various types of ***FIR*** windows. ***Recursive*** form of our proposed algorithm provided ***cross*** and ***auto*** correlation sequences of the output process in ***linear*** time. The ***worst*** case time complexity of our proposed algorithm grows linearly with the length of the input auto correlation sequence. We showed in this paper that except ***exponential*** shaped window, our proposed algorithm required ***only addition*** operations in performing ***LPF*** of the input auto correlation sequence through other three types of FIR shaped windows. This indeed, is really a great achievement of our proposed algorithm. We also showed that our proposed algorithm computed cross correlation sequence of the output process in quick time which does not depend on the size of the window, ***M***. Finally, we also showed in this paper that our proposed algorithm outperformed all the existing methods either in time or frequency domains both in terms of ***worst*** case time complexity and ***exact*** number of arithmetic operations (i.e., ***+s*** and ****s***).

ACKNOWLEDGMENT

We are extremely grateful to the Department of Electrical Engineering of COMSATS Institute of Information Technology (CIIT) for carrying out this work. Moreover, we are also thankful to the anonymous reviewers for their valuable suggestions which really helped us in improving the quality of the paper.

REFERENCES

1. *Wang Zongbo, Z., Meiguo, Gao, Xiongjun, Fu and Changyong, Jiang, Design of Real-time Convolution Processor and its Application in Radar Echo Signal Simulator. International Conference on Computer and Information Technology, pp. 163-166, 2008.*
2. *Bhattacharya, B. and Patel, Gopal, A fast implementation of Correlation of long data sequences for Coherent Receivers. IEEE Transactions on Circuits and Systems II: Analog and Digital Processing, vol. 49, pp. 430-433, June, 2002.*
3. *Jung Kap Kuk, Z. and Ik Cho, Nam, Block Convolution with Arbitrary Delays using Fast Fourier Transform. Proceedings of 2005 International Symposium on Intelligent Signal Processing and Communication Systems, pp. 814-816, 2005.*
4. *Suresh, K. and Sreeniva, T. V., Block Convolution using Discrete Trigonometric Transforms and Discrete Fourier Transform. IEEE Signal Processing Letters, vol. 15, pp. 430-433, 2008.*
5. *Ejaz A. Ansari, Saleem Akhtar, Muhammad Nadeem and Tariq Humayun, Processing of Auto and Cross Correlations Two jointly WSS Random Processes through various types of Windows. Accepted in 4th issue, 2013 of Science International Journal, Lahore*.
6. *A. V. Oppenheim, R. W. Schafer and J. R. Buck, Discrete-time Signal Processing, Pearson edition, 2nd edition, 1994.*
7. *J. G. Proakis and M. Dimitris, Digital Signal Processing, (Principles, Algorithms and Applications), Pearson son edition, 4th edition, 2011.*
8. *S. K. Mitra, Digital Signal Processing, Tata McGraw-Hill, 3rd edition, 2006.*
9. *A. Papoulis and S. U. Pillai, Probability, Random Variables and Stochastic Processes. 4th Edition, Tata McGraw Hill, India, 2002.*
10. *T. H. Cormen, C. E. Leiserson and R. L. Rives, Introduction to Algorithms. MIT Press, Cambridge, MA, 1990.*