DISTORTIONS IN SPEECH ENHANCEMENT DUE TO BLOCK PROCESSING

J.I. Marin-Hurtado, D.V. Anderson

School of Electrical and Computer Engineering
Georgia Institute of Technology, Atlanta, GA 30332
jimh3@gatech.edu, dva@ece.gatech.edu

ABSTRACT
In most speech enhancement algorithms, a frequency response is computed per each block, and a multiplication in the frequency domain is used to apply this frequency response. Typically, in these algorithms, the updating procedure cannot ensure the fulfillment of the condition to perform a linear convolution instead of a circular convolution. As a result, artifacts and distortions may be present. Standard methods to deal with these distortions use overlapping and windowing. This work presents a detailed analysis of these distortions and proposes a new method to perform a artifact-free and distortion-free convolution in the FFT domain. Index

Terms— blocking artifacts, speech enhancement, fast convolution

1. INTRODUCTION
Multiplication in the frequency-domain is an important part of most speech enhancement algorithms [1, 2, 3, 4, 5, 6, 7, 8, 9]. Multiplication of the FFT bins by a gain vector is equivalent to a circular convolution in the time domain, and to avoid this circular convolution, the constraint \( N \geq L + M - 1 \) must be imposed over the lengths for the FFT, \( N \), input block \( x_r(n) \), \( L \), and impulse response \( h_r(n) \), \( M \), at the block index \( r \) [10]. When these lengths are known, satisfying the above constraint is an easy task. In such a case, \( x_r(n) \) and \( h_r(n) \) are zero-padded to form vectors of length \( N \). However, in most speech enhancement algorithms, a frequency response \( H_r(k) \) of length \( N \) rather than \( h_r(n) \) of length \( M \) is estimated for the input block \( x_r(n) \) using a non-linear function of the \( N \)-point FFT of \( x_r(n) \) (Fig. 1a). In such a case, the inverse FFT of \( H_r(k) \), i.e. \( h_r(n) \), will have a length \( N \), even if the original FFT was taken of a zero-padded signal. As a result, the above constraint can be easily broken, and artifacts and distortions may appear.

To deal with artifacts and distortions resulting from the invalidation of the linear convolution condition, overlapping and windowing of input blocks have been commonly used [1]. A naive approach to satisfy this constraint is shown in Fig 1b. In such a case, \( H_r \) is truncated in the time-domain to ensure an impulse response of length \( M \), but this method modifies the desired frequency response. Alternative methods [11, 12] can minimize artifacts and distortions more successfully than windowing or truncation, but additional artifacts may be introduced. These artifacts and distortions have been evaluated by means of a perceptual characterization [13]. This paper provides a different approach attempting to understand the source of distortions and artifacts in standard FFT (or fast) convolution when the impulse response has a length \( N \) and is time-variant, and to answer how efficient windowing is to minimize these distortions and artifacts.

Different methods to update \( h_r(n) \) have been proposed for frequency-domain adaptive filtering, such as sliding DFT [14] and subband adaptive filtering [15, 16]. To implement most speech enhancement algorithms, FFTs are required, thus, a FFT-based method is preferable over a subband approach. For this reason, this paper proposes a method to perform distortion-free FFT convolution (Fig 1c).

2. MATHEMATICAL FRAMEWORK
In an attempt to satisfy the condition \( N \geq L + M - 1 \), many practical speech enhancement implementations assume \( N = 2L, M = L \), with \( L \) the length of the input block before zero-padding [1]. To reduce artifacts and distortions, overlapping of input blocks by \( R \)-samples and windowing are used. Although any overlap value is possible, overlap by 50\% (\( R = L/2 = N/4 \)) is commonly used [1]. For the rest of the paper, an FFT convolution using overlap-add and an overlap by 50\% is assumed, which will be referred as standard windowed FFT convolution (SWFC).

A data-flow graph for SWFC is shown in Figure 2. With this overlap, the input signal can be considered as partitioned in non-overlapped blocks of length \( L/2 \). Then, two contiguous blocks of length \( L/2 \) are concatenated, windowed and passed through a basic set of operations to produce an output block \( y_r \) of length \( N \), where \( r \) stands for the block index. Finally, all individual outputs \( y_r \) are merged together as shown in Figure 2b. The output \( y_r \) can be described by
\[ y_{\text{std}} = \left( \begin{bmatrix} \mathbf{L}_{\text{top}}^r \\ \mathbf{L}_{\text{bot}}^r \end{bmatrix} + \begin{bmatrix} \mathbf{U}_{\text{top}}^r \\ \mathbf{U}_{\text{bot}}^r \end{bmatrix} \right) \begin{bmatrix} \mathbf{W}_0^r x_r \\ \mathbf{W}_1^r x_{r+1} \end{bmatrix} \]  

(1)

where \( h_{(r)}^r \) denotes the impulse response elements, \( x_r \) and \( x_{r+1} \) are two consecutive non-overlapped input segments of length \( L/2 \), \( \mathbf{W}_0 \) and \( \mathbf{W}_1 \) are diagonal matrices holding the lower-half and upper-half parts of the window, \( \mathbf{L}_r \) matrices hold only the lower-half elements \( h_0, ..., h_{N/2-1} \), \( \mathbf{U}_r \) matrices hold the upper-half elements \( h_{N/2}, ..., h_{N-1} \), and the superscript \( \text{std} \) stands for SWFC.

For an output segment of length \( L/2 \) (Figure 2b), the combination of the output blocks \( y_r \) can be expressed as

\[ z_{\text{std}} = \frac{1}{L/2} \sum_{L/2}^n \left( y_{\text{std}}^r \right) + ... + \frac{1}{L/2} \left( y_{\text{std}}^{r-3} \right) \]  

(2)

\[ z_{\text{std}} = \mathbf{L}_0^r \mathbf{W}_0 x_r + \mathbf{L}_1^r \mathbf{W}_0 x_{r-1} + \mathbf{L}_2^r \mathbf{W}_1 x_r + \mathbf{L}_3^r \mathbf{W}_1 x_{r-1} + \mathbf{L}_4^r \mathbf{W}_1 x_{r-2} + \mathbf{L}_5^r \mathbf{W}_1 x_{r-3} + \mathbf{L}_6^r \mathbf{W}_1 x_{r-4} \]  

(3)

To compare the given output to the output expected by a linear convolution algorithm, \( \mathbf{U}_r \) is modified to be

\[ y_{\text{lin}} = \left( \begin{bmatrix} \mathbf{L}_{\text{top}}^r \\ \mathbf{L}_{\text{bot}}^r \end{bmatrix} + \begin{bmatrix} \mathbf{U}_{\text{top}}^r \\ \mathbf{U}_{\text{bot}}^r \end{bmatrix} \right) \begin{bmatrix} \mathbf{W}_0^r x_r \\ \mathbf{W}_1^r x_{r+1} \end{bmatrix} \]

then, the output of a linear convolution becomes

\[ z_{\text{lin}} = \mathbf{L}_0^r \mathbf{W}_0 x_r + \mathbf{L}_1^r \mathbf{W}_0 x_{r-1} + \mathbf{L}_2^r \mathbf{W}_1 x_r + \mathbf{L}_3^r \mathbf{W}_1 x_{r-1} + \mathbf{L}_4^r \mathbf{W}_1 x_{r-2} + \mathbf{L}_5^r \mathbf{W}_1 x_{r-3} + \mathbf{L}_6^r \mathbf{W}_1 x_{r-4} \]  

(4)

The above modification is based in the fact that an input block of length \( N/2 \) and an impulse response of length \( N \) will produce an output block of length \( 3N/2 = 3L \).

3. ANALYSIS OF DISTORTIONS

A comparison of (3) and (4) shows that the last three terms differ in the block indices \( r \). Since the wrong terms depend exclusively on the upper-half part of the impulse response \( \mathbf{U}_r \), a well-designed impulse response to be used in SWFC must have zeros in the range \( n = [N/2, N-1] \). Thus, a minimization of distortions can be achieved by forcing the smallest values of the impulse response to be into \( n = [N/2, N-1] \). This procedure can be performed by shifting the impulse response.

Simulations with different fixed impulse responses have been shown that if the 98% of energy of the impulse response is allocated in the lower-half part, distortions and artifacts might be negligible. An alternative solution may be a truncation of the impulse response to nullify the effect of the \( \mathbf{U}_r \) elements (Fig. 1b). This alternative is simple but modifies the desired frequency response, and adds an extra computational cost.

To analyze the impact of windowing on the minimization of distortions, suppose that for a short period time the speech enhancement algorithm produces similar frequency responses. Then, dropping the block indices \( r \) for \( \mathbf{U} \) and \( \mathbf{L} \) from (3) and (4), the difference between the expected and the given outputs, \( \Delta z_r = z_{\text{lin}} - z_{\text{std}} \), is given by

\[ \Delta z_r = U_{01}^0 W_0 (x_{r-4} - x_r) + U_{01}^1 W_1 (x_{r-3} - x_{r+1}) \]

with \( W_{01} = \mathbf{W}_0 + \mathbf{W}_1 \). This equation shows that artifacts cannot be reduced dramatically by windowing without compromising the behavior of the entire algorithm. For example, a tapered window could be designed to remove the effect of \( U_{01}^1 \), but to be a tapered-window \( \mathbf{W}_0 + \mathbf{W}_1 = \text{const} \times \mathbf{I} \), which implies that distortions caused by \( U_{00}^0 \) could not be removed. Relaxing the tapered-window constraint may allow distortions to be removed, but additional blocking effects might appear because of a bad-averaging of the output blocks.

To estimate quantitatively the efficiency of windowing to reduce distortions, a metric based on \( \| \Delta z_r \|^2 \) was used. This metric relies on the idea that FFT convolution should provide an output similar to that of linear convolution. Since \( \Delta z_r \) depends on the input signal, an ideal band-pass signal of central frequency \( \omega_c \) and bandwidth \( B \) was assumed for this analysis. In other words, the distortion \( \| \Delta z_r \|^2 \) is estimated at each \( \omega_c \). This metric was evaluated for time invariant filters under different typical windows. Fig. 3 shows \( \| \Delta z_r \|^2 \) for an ideal low-pass filter with a cutoff frequency 0.4\( \pi \). This frequency response was designed in the frequency domain by...
For example, the samples related to the range \( l = 0, \ldots, n \) are computed from averaging the impulse responses at the current frame, \( h(r) \), weighted by \( W_0 \), and the previous frame, \( h(r-1) \), weighted by \( W_1 \). The next range, \( l = [n, n + N/4] \) comes from the averaging of \( h(r-1) \) and \( h(r-2) \), and so on. This averaging will be considered as the ideal averaging. Note that as \( l \) increases, older impulse responses are used to estimate a given impulse response sample.

For SWFC (Fig. 4b), a bad averaging is present for the impulse response samples in the range \( l = [n + N/2, N - 1] \). For example, in the range \( l = [n + N/2, n + 3N/4] \), frames \( r - 3 \) and \( r \) are used instead of \( r - 3 \) and \( r - 4 \). As a consequence, additional artifacts may be present in time-variant filtering performed through SWFC. Again, the upper-half part of \( h(r), \ldots, h(r-3) \) is responsible for the bad time-averaging to produce the impulse response. If these samples are small enough, artifacts and distortions in a time-variant filtering may be negligible. However, in general, this is not the case, and when these artifacts and distortions are present, they are usually perceived as audible clicks [13].

4. DISTORTION-FREE FFT CONVOLUTION

Time-domain zero-padding is a typical way to avoid circular convolution when \( L \) and \( M \) are known. Assuming that \( L = M = N \), a distortion-free FFT convolution should use 2\( N \) FFTs, and the frequency response estimated by the speech enhancement algorithm must have a length \( N \). However, in the given scenario, \( N \) frequency samples rather than \( N \) time samples are known, therefore, this time-domain zero-padding should be performed by means of a frequency extension (FEXT) (Fig. 1c). This frequency extension requires an \( N \)-IFFT of \( H \) to get \( h \), zero-padding \( h \) to get \( \hat{h} \), and finally a 2\( N \)-FFT to get \( \hat{H} \). Exploiting the structure of the extended spectrum \( \hat{H} \), a more efficient algorithm is proposed as follows. Let the extended impulse response be \( \hat{h} = [h \, 0] \), then, the extended frequency response is given by

\[
\hat{H}(2k) = \sum_{n=0}^{N-1} h(n) F_\frac{2k}{N} = H(k)
\]

for \( 0 \leq k \leq N - 1 \). Thus, above equations suggest that even samples of the extended frequency response are just the same samples as the original frequency response \( H(k) \) of length \( N \), and the odd samples can be computed by an FFT-like transform that uses a twiddle factor \( F_\frac{2k}{N} \) instead of \( F_{kN} \), i.e. an FFT algorithm can be modified to compute \( H(k + \frac{1}{2}) \) in \( N \log_2 N \)-time. A sketch for FEXT is shown in Figure 5. In this diagram, the input block has a length \( N = 2L \) and output blocks are added assuming a shift by \( N/2 \). The output \( X_r(k) \) of length \( N \) is intended to be used in the update of \( H_r \).

FEXT processes the input signal at a rate twice higher than SWFC. Whereas FEXT uses input blocks of length \( N \), SWFC uses blocks of length \( L = N/2 \), although frequency response is still \( N \). This fact allows the reduction of computational load. Counting the number of real multipliers required to create \( N/4 \) output samples: SWFC (Fig. 1a), 4\( N \log_2 N + 9N/2 \), SWFC with truncation (Fig. 1b), 8\( N \log_2 N + 9N/2 \), and FEXT (Fig. 5), 6\( N \log_2 N + 13N/2 \). These results show that FEXT demands approximately 1.5 times more multipli-

Fig. 3. Distortion metric for SWFC using different windows. Cutoff frequency indicated by dashed vertical lines.

Fig. 4. Averaging of previous impulse responses to produce the impulse response at the time \( rR + n \).

In general, for any speech enhancement algorithm, frequency responses at each block index \( r \) are different. Thus, distortions induced by the upper-half part of the impulse responses may produce a scenario that is worse than that described above. Ideally, in a time-variant case, any output sample \( z(rR + n), n = 0, \ldots, L/2 - 1 \) is expected to be produced by a filter with a specific impulse response \( h^{(r+n)}(l), l = 0, \ldots, N - 1 \). This impulse response comes from a weighted time-averaging of impulse responses for the previous blocks \( h^{(r)}, \ldots, h^{(r-\max \log_2 L)} \). For SWFC and linear convolution, this averaging can be obtained exposing the temporal operations from (3) and (4), i.e. obtaining an expression to denote the vector elements \( z_{\text{std}}(n) \) and \( z_{R,n}(n) \) for \( 0 \leq n \leq L/2 - 1 \). The mathematical derivation is not included, but a graphical representation of these equations is shown in Fig. 4.

If linear convolutions were used (4), \( h^{(r+n)}(l) \) is the result of averaging the impulse responses of the previous 6 blocks, \( h^{(r)}(n), \ldots, h^{(r-5)}(n) \) (Fig. 4a). This averaging depends on \( n \), and each sample of the averaged impulse response involves only two neighbor impulse response frames.
ers than SWFC, whereas SWFC with truncation involves 2 times more multipliers than SWFC. Therefore, FEXT shows a better performance and would be preferable for most applications.

Artifacts regarding time-averaging of impulse responses are completely avoided in FEXT (Fig. 4c). Impulse response averaging in FEXT is correct because impulse response samples into the range $k = [n + N/2 + 1, N - 1]$ are the result of averaging the older impulse responses.

5. CONCLUSIONS

For many types of processing, a gain vector is calculated and applied in the frequency domain. In this case, the impulse response associated with the gain vector is not guaranteed to meet the requirements for linear convolution. This paper proved the existence of artifacts and distortions inherent to the standard FFT convolution even if the impulse response is time-invariant. To minimize these artifacts and distortions in SWFC, the most relevant samples of the impulse response have to be placed in the lower-half part of the impulse response vector of length $N$. Windowing may reduce distortions and artifacts, but its impact is not as high enough as that of impulse response shifting. In addition, a computationally efficient artifact-free and distortion-free strategy to perform FFT convolution, FEXT, was described. FEXT is based on the extension of the frequency response $H(k)$ of length $N$ to a new frequency response of length $2N$, assuming a zero-padding in the time-domain.

6. REFERENCES


