An Adaptive Real-Time FPGA-Based Speech Enhancement Processing System

Andrés J. Cuevas-Romano, Víctor H. Díaz-Ramírez, and Andrés Calvillo-Téllez

Instituto Politécnico Nacional - CITEDI, Avenida Instituto Politecnico Nacional 1310, Mesa de Otay, Tijuana B.C. 22510, Mexico; email: \{acuevas, vhdiaz, calvillo\}@citedi.mx

Abstract—A real-time system for robust speech processing based on a Field Programable Gate Array (FPGA) is presented. The system estimates a clean signal from a noisy speech signal using a locally adaptive algorithm based on calculation of rank-order statistics from the input signal over a moving window. The algorithm is able to improve the quality of noisy speech with no perceivable musical noise. The processing system adapts the size and contents of the moving window as well as the local estimator used to recover the clean signal at each iteration. Results obtained with the proposed system in terms of quality and intelligibility objective measures are presented and discussed when testing the system in different non-stationary noise environments. The real-time performance of the proposed FPGA system is also evaluated and discussed.

Index Terms—Speech enhancement, noise reduction, audio processing, field programmable gate arrays, real-time.

I. INTRODUCTION

NOWADAYS, the demand for transmitting noise-free speech signals on mobile communication equipment has greatly increased the need for robust techniques for real-time speech enhancement. Two common techniques used for speech enhancement are given by the spectral subtraction and Wiener filtering algorithms [1]. The former approach calculates an estimate of the average noise spectrum which is subtracted from the spectrum of the noisy signal in such a manner that the average signal-to-noise ratio (SNR) is improved [2], [3]. On the other hand, the Wiener filtering is a linear filter optimized with respect to the mean-squared-error between the clean and processed signals. In this approach, an estimate of the clean speech power spectrum is estimated (locally) from the noisy speech signal [4]. Furthermore, when the noise can be described by a stationary random process then the use of wiener filtering is recommendable [5].

The spectral subtraction and the Wiener filtering algorithms are fast and reliable noise reduction techniques. However, because of both of these algorithms operate in the frequency domain, they commonly introduce spurious artifacts to processed speech resulting in annoying musical noise [1]. Furthermore, it is important to note that these algorithms assume that noise is stationary, however, in real life scenarios such as in mobile communications, noise is almost never stationary. For this reason, the use of a robust filtering is desirable. There are several successful nonlinear filters based on calculation of rank-order statistics that could be used for speech enhancement [6]. It is well known that these filters are robust and able to preserve fine signal structures. In speech enhancement, this feature can help to suppress the additive noise while preserving the intelligibility of speech. Speech processing systems can be broadly classified in single or multiple channel systems. Single channel systems use only one microphone to capture speech signals whereas multiple channel systems utilize a microphone array to capture speech from different locations.

In this work, we propose a single channel system for real-time speech enhancement which uses a nonlinear locally adaptive algorithm for robust speech processing based on a FPGA chip. The algorithm is carried out in the time-domain using a time variant moving window. The algorithm is able to improve the quality of noisy speech in terms of objective metrics and with no perceivable musical noise [7]. The processing system adapts the size and contents of the moving window as well as the local estimator used to recover the clean signal at each iteration. Therefore, a noise-free signal can be estimated using a time-variant estimator over a local adaptive neighborhood. A local neighborhood is a subset of signal elements of the sliding window, which are close in some sense to a given element [6]. The proposed algorithm is adaptive to non-stationary signal fragments and noise fluctuations.

Note that since the proposed method is implemented in the time domain then a priori estimation of the spectral distributions of signals and noise is not carried out for each frame. Thus, annoying artifacts introduced to processed speech are not perceivable [8]. A schematic diagram of the proposed system is shown in Fig. 1.

II. SPEECH ENHANCEMENT ALGORITHM

Adaptive filtering has proven to be a good choice in processing signals corrupted by non-stationary noise [9]. The proposed algorithm for speech enhancement takes advantage
of the calculation of rank-order statistics to locally modify the parameters of a robust estimator for recovering a speech signal with an improved quality and by preserving its intelligibility [7].

The flow diagram of the used speech enhancement algorithm is presented in Fig. 2 and consists of the following steps [7]:

**STEP 1:** Read an initial input speech segment $n_0$ with $S$ elements assuming speaker’s silence.

**STEP 2:** Start processing of speech segments ($j = 1$).

**STEP 3:** Read the $j$th input speech segment $f_j = s_j + r_j$ (with $N$ samples) to be processed. Note that the speech segment is given by superposition of the clean signal $s_j$ and additive noise $r_j$.

**STEP 4:** Create a sliding window $w_{j,i}$ around the noisy element $f_j(i)$, given by

$$w_{j,i} = \left[ w_{j,i}(n) = f_j(n) : |n - i| \leq \frac{S - 1}{2} \right]^T.$$  \hspace{1cm} (1)

**STEP 5:** Calculate a local estimate of the SNR as follows:

$$SNR_{j,i} = \frac{w_{j,i}^T w_{j,i}}{n_0^T n_0}.$$  \hspace{1cm} (2)

**STEP 6:** Calculate the parameter $\epsilon_v$ as follows:

$$\epsilon_{v,j,i} = k_1 \sigma_n \left[ 1 - \frac{1}{1 + (SNR_{j,i})^{-k_2}} \right],$$  \hspace{1cm} (3)

where $k_1$ and $k_2$ are constant parameters which are defined within the range of (0,1]. The parameters $k_1$ and $k_2$ help us to take into account a priori information about either the spread of the signal to be preserved or noise fluctuation to be suppressed [7]. Using these two parameters a trade-off between noise suppression and introduction of artifacts to the processed speech signal can be achieved.

**STEP 7:** Construct an EV-neighborhood $v_{j,i}$ from $w_{j,i}$ as follows:

$$v_{j,i} = \left[ w_{j,i}(n) : w_{j,i}(i) - \epsilon_v \leq w_{j,i}(n) \leq w_{j,i}(i) + \epsilon_v \right]^T.$$

**STEP 8:** Compute the output estimate as $y_j(i) = a^T v_{j,i}$.

**STEP 9:** Move the window ($i = i + 1$) and evaluate if ($i \leq N$). If the result is "true" go to STEP 4. Otherwise, go to STEP 3.

### III. REAL-TIME FPGA-BASED SYSTEM FOR SPEECH ENHANCEMENT

In recent years, FPGA devices have achieved a growing popularity because of their great flexibility. These devices offer the possibility to reconfigure their hardware according to the designer’s needs. An FPGA can be programmed to operate with a custom hardware architecture or can be programmed to operate as a customizable reduced instruction set computer (RISC) with particular specifications. In this work we propose a real-time robust speech enhancement system using the algorithm shown in Fig. 2 which is implemented on an FPGA configured as a high-speed custom RISC processor.

The development board used for the implementation is the Altera DE-115 which has a Cyclone IV E EP4CE115 chip. The sampling rate for this system was set at 8 Khz and the input speech was coded using signed 16 bit integer data. From Fig. 1 we can observe that the CODEC captures the input speech signal through the microphone (mic) and codes each data sample according to the set sample rate. Next, the FPGA processor reads out each data sample from the CODEC and arranges them into data frames of fixed length for subsequent signal processing. Each processed data frame is sent back to the CODEC one next to another for sound playback through a speaker (spk).

In order to obtain a real-time processing system, the processing block must be fast enough to maintain a constant data flow. To obtain the best performance with the FPGA chip, we use the Nios II/f (fast) configuration. The block diagram of the proposed speech processing system is shown in Fig. 3.

For real-time operation, we must take into account that the size of the data frames in the system is a hardware dependent parameter. Observe that if the size of the data frame is too large then the system’s latency will be high. On the other hand, if the size of the data frame is too small then the latency will be
short but there will be interruptions in the output audio stream [10]. So, the length of the data frame is a trade-off parameter among the system’s latency and the quality of playback stream. The size of the data frame used for the implementation is 204 samples, because the size of the sliding window is 51 samples. It is important to mention that the processing time for 204 samples captured with a sample rate of 8 Khz takes 25 ms.

IV. RESULTS

In this section we present the results obtained with the proposed real-time system for speech enhancement. We evaluate the performance of the system in terms of objective quality and intelligibility metrics. Also, we test the real-time performance of the FPGA-based system. To evaluate speech quality we use the following metrics: perceptual evaluation of speech quality (PESQ) [11], source to distortion ratio (SDR) and source to interference ratio (SIR) [12]. The SDR metric, characterizes the effective separation between speech and background noise. The SIR, is a metric commonly used to characterize introduction of artificial artifacts such as musical noise. Speech intelligibility was evaluated by the short-time objective intelligibility (STOI) metric [13].

We tested the proposed system for speech enhancement in four different noisy environments: train, car, street and restaurant. Speech samples were taken from the NOIZEUS database [14]. The database consists of 30 IEEE sentences [15], produced by different speakers. The speech samples were originally sampled at 25 kHz and downsamped to 8 kHz. All sentences from the database were processed with the proposed system in different noise environments. Fig. 4 shows an example of one speech file from the database after processing with the proposed FPGA-based system when speech is corrupted with 15 dB SNR car noise. Note that the system adapts well to the non-stationary characteristics of speech signal and noise (see the ± behavior). With 95% confidence the results obtained in terms of performance metrics are shown on tables 1-4 and Fig. 9-12. The results show a considerable quality improvement from the noisy recordings and show consistent results across the different noisy environments. The results obtained in terms of speech intelligibility (see STOI results) show a lower rating for higher noise SNR, however, this is normal behavior because it has shown [1] that all noise reduction algorithms either improve the speech quality or intelligibility. Also, note that the proposed approach yields a good performance in terms of noise reduction that is characterized by the SDR.

In Fig. 5 we see the spectrogram of the clean sentence of the database "He knew the skill of the great young actress". In Fig. 6, we see the spectrogram of the sentence used in Fig. 5 corrupted with 15 dB SNR car noise. The corrupted sentence used in Fig. 6 was processed with the proposed FPGA-based system. The spectrogram of the processed sentence is shown in Fig. 7. It can be noted that there is a considerable reduction of noise and the frequency distribution of the processed signal is very similar to the one for the clean signal shown in Fig. 5.

<table>
<thead>
<tr>
<th>Table I</th>
<th>PERFORMANCE OF THE PROPOSED ALGORITHM WITH 95% CONFIDENCE IN NON-STATIONARY TRAIN NOISE ENVIRONMENT.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>STOI</td>
<td>0.93 ± 0.03</td>
</tr>
<tr>
<td>PESQ</td>
<td>2.52 ± 0.10</td>
</tr>
<tr>
<td>SIR</td>
<td>17.18 ± 1.25</td>
</tr>
<tr>
<td>SDR</td>
<td>15.96 ± 0.55</td>
</tr>
</tbody>
</table>

Evaluation of the real-time performance of the system

For evaluation of the real-time performance of the system, we measure the processing time scheduling of the FPGA implementation in terms of the rate of data frame processing (latency). By performing these tests, we are interested in
Figure 5. Spectrogram of the clean speech sentence "He knew the skill of the great young actress".

Figure 6. Spectrogram of the speech sentence "He knew the skill of the great young actress" corrupted with 10dB SNR car noise.

Table II

<table>
<thead>
<tr>
<th></th>
<th>15 dB</th>
<th>10 dB</th>
<th>5 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STOI</strong></td>
<td>0.92 ± 0.03</td>
<td>0.84 ± 0.03</td>
<td>0.77 ± 0.02</td>
</tr>
<tr>
<td><strong>PESQ</strong></td>
<td>2.55 ± 0.10</td>
<td>2.22 ± 0.11</td>
<td>1.94 ± 0.12</td>
</tr>
<tr>
<td><strong>SIR</strong></td>
<td>17.74 ± 0.80</td>
<td>13.40 ± 0.99</td>
<td>8.45 ± 1.11</td>
</tr>
<tr>
<td><strong>SDR</strong></td>
<td>16.55 ± 0.55</td>
<td>11.09 ± 0.39</td>
<td>7.26 ± 0.64</td>
</tr>
</tbody>
</table>

Table III

<table>
<thead>
<tr>
<th></th>
<th>15 dB</th>
<th>10 dB</th>
<th>5 dB</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>STOI</strong></td>
<td>0.93 ± 0.03</td>
<td>0.87 ± 0.03</td>
<td>0.77 ± 0.02</td>
</tr>
<tr>
<td><strong>PESQ</strong></td>
<td>2.53 ± 0.10</td>
<td>2.27 ± 0.10</td>
<td>1.91 ± 0.12</td>
</tr>
<tr>
<td><strong>SIR</strong></td>
<td>16.87 ± 1.25</td>
<td>11.53 ± 0.99</td>
<td>6.51 ± 1.57</td>
</tr>
<tr>
<td><strong>SDR</strong></td>
<td>16.03 ± 0.55</td>
<td>10.78 ± 0.39</td>
<td>5.97 ± 0.65</td>
</tr>
</tbody>
</table>

Figure 7. Spectrogram of the processed speech sentence "He knew the skill of the great young actress" when is corrupted with 15dB SNR car noise.

Figure 8. Spectrogram of the processed speech sentence "He knew the skill of the great young actress" when is corrupted with 10dB SNR car noise.
showing that the proposed system can be efficiently used for real-time applications. Additionally, we want to show that the proposed system can improve the quality of noisy speech signals in real-time without sacrificing intelligibility and with no perceivable musical noise. To measure the latency of the FPGA implementation, a 600 Hz sinusoid tone was used as the input signal. In Fig. 13 we can see that the obtained output latency is 80 ms. When the speech enhancement algorithm is disabled, that is, when the input samples are copied from the input ADC to a memory buffer and then sent for playback, it took 64 ms (see Fig. 13). So by subtracting this latency to the latency when the algorithm is active we get 16 ms which is the latency of the processing operations.

V. CONCLUSIONS

A locally adaptive FPGA-based system for robust speech processing was presented. The implemented algorithm is able to recover an undistorted signal from a noisy speech employing a time-variant estimator over a locally adaptive neighborhood without introducing musical noise. Based on the conducted tests, it is concluded that the system is capable of enhancing speech signals corrupted by non-stationary noise in a real-time processing scenario.

REFERENCES

Figure 13. Measurement of system’s output latencies (top) Input, (center) Bypass Output and (bottom) Processing Output.


