REAL-TIME SPEECH-IN-NOISE INTELLIGIBILITY ENHANCEMENT BASED ON SPECTRAL SHAPING AND DYNAMIC RANGE COMPRESSION

Vassilis Tsiaras[1], *Tudor-Cătălin Zorilă*[2], *Yannis Stylianou*[3], *Masami Akamine*[4]

[1]Technical University of Crete, Chania, Greece
[2]Computer Science Department, University of Crete, Heraklion, Crete, Greece
[3]Toshiba Cambridge Research Lab, UK
[4]Toshiba Research and Development Center, Kawasaki, Japan

{vass.tsiaras,ztudorc}@gmail.com,yannis.stylianou@crl.toshiba.co.uk,masa.akamine@toshiba.co.jp

ABSTRACT

We demonstrate a real-time implementation of a speech-in-noise intelligibility enhancement algorithm based on spectral shaping and dynamic range compression. The signal is enhanced before presented in a noisy environment, under the constraint of equal global signal power before and after modifications. The demonstrator modifies in real-time pre-recorded sentences as well as input from a microphone (live speech), while it runs in a standard laptop. The noise level and type can be controlled by the user using a Java-based graphical interface. The whole architecture produces low-delay and artifact-free signals and it can be deployed on various Java compatible devices.

Index Terms— real-time speech-in-noise intelligibility enhancement, spectral shaping, dynamic range compression

1. INTRODUCTION

The ability to detect speech in noise plays a significant role in our communication with others. However, speech produced under real conditions (not in a recording studio, nor in a quiet room) is not always intelligible due to the presence of background noise. In many real life situations, strong acoustical background noise (ABN) found at listener's side could severely degrade speech intelligibility (e.g., airports, train stations, sports arenas, traffic noise etc.). Therefore, it is particularly important to explore signal processing techniques aiming at artificially increasing speech intelligibility in communicative scenarios where the listener is located in noisy environments. This is called near end listening enhancement (NELE) and has direct implications for many speech-oriented technologies, such as speech synthesis, speech recognition or speech coding [7].

Various solutions have been suggested for NELE problem which involves different digital signal processing strategies. A naive solution would be to increase the power of speech (e.g., increase the volume). However, such approaches are dangerous for listener's hearing at severe noise levels and not always effective. More elaborate algorithms have been suggested to improve intelligibility of speech in noise which involve boosting speech audibility over noise at selected frequency ranges [7, 8] or/and enhancing those speech cues known to promote intelligibility in several studies using clear, Lombard and hearing-impaired-oriented stimuli. [1, 6, 4, 9, 3].

Although there exist many published intelligibility enhancement algorithms, most of them are too complex for being implemented in real-time (which is important for many real applications like announcements).

In this paper we present a software demonstrator capable of enhancing in real-time speech captured from a microphone towards much higher intelligible signals when subjected to various ABN. The speech modification algorithm used for this demo operates on clean (not-noisy) sounds and it is based on the work presented in [9], where both spectral shaping (SS) and dynamic range compression (DRC) were used to reallocate signal's energy in frequency and time domains, respectively, under the constraint of equal RMS before and after modification. This method (referred to as SSDRC) was shown to outperform recent state-of-the-art similar systems in terms of intelligibility gains for a broad range of noise conditions [2, 5]. Its light but effective DSP architecture makes it suitable for real-time implementation and thus useful from a practical point of view. The system will be demonstrated using recorded or live speech (from microphone). The conference environment provides an ideal and natural noisy set-up to demonstrate the power of the real-time SSDRC system.

The rest of the paper is organized as follows. Section 2 briefly presents baseline and real-time SSDRC (denoted (bs)SSDRC and (rt)SSDRC, respectively), Section 3 describes implementation details of (rt)SSDRC and it shows the design and usage of software demonstrator, while Section 4 concludes this work.

2. SCIENTIFIC AND TECHNICAL DESCRIPTION

2.1. Baseline SSDRC

Both baseline and real-time SSDRC algorithms consist of two sub-systems connected in cascade form [9]. The whole process is done under the constraint of equal RMS of signals before and after modifications. The first sub-system is a spectral shaper designed to re-adjust signal's spectral energy following results from clear and Lombard speech studies. Thus, short-term magnitude spectra of speech is enhanced by three filters, two of them being adapted to the voicing probability on a frame-by-frame basis. Voicing adaptivity is used to reduce audible artifacts caused by enhancing unvoiced segments of speech. Therefore, the first filter is an adaptive formant sharpener inspired from clear speech studies, the second one reduces spectral tilt (Lombard speech) by applying an adaptive pre-emphasis, while the last one boosts the components from the mid-high frequency range protecting them from low-pass operations which are usually used during playback (e.g., headphones, loudspeakers). A full description of all of these filters can be found in [9]. All previous operations are done in frequency domain and the modified speech signal is reconstructed by means of overlap-add (OLA), thus keeping the original phase spectra.

The second stage of SSDRC is intended to re-arrange the energy of speech waveform over time such that low energy segments (e.g., nasal, onsets and offsets) are amplified, while more energetic areas (sonorant sounds) are attenuated. Inspired by audio broadcasting and hearing-aid amplification techniques, the later waveform manipulation is important in promoting speech intelligibility because low energy segments of speech are most likely to be masked by noise. In effect, original waveform is re-scaled using time-varying gains computed from applying both dynamic and static stages of DRC to signal's temporal envelope. For this purpose, the temporal envelope of speech is firstly estimated by means of Hilbert transform, then the smoother version of the estimated envelope is dynamically compressed with 2ms release time and almost an instantaneous attack time constants (dynamic compression), while the resulted sequence is finally converted to dB and statically compressed using a pre-defined input/output envelope characteristic (IOEC) (static compression) [9].

2.2. Real-time SSDRC

Although (rt)SSDRC follows the same signal processing path as (bs)SSDRC, it includes several modifications to account for the practical issues that arise when operating in real-time. Two major types of problems are addressed in this context. The first one concerns speech segmentation, while the other one is related to non-causal operations used in the (bs)SSDRC algorithm.

2.3. Speech segmentation

The (rt)SSDRC runs in a loop, where in each iteration it reads and modifies a segment of the signal. The length of the segments is determined by two conflicting requirements. The larger the segment of the signal, better the approximation of (bs)SSDRC output by (rt)SSDRC. On the other hand, the length of the segment is proportional to the latency of the output. Also, the processing is simplified if the length of the segments is a multiple of the length of the frames, which depend on the sampling frequency (Fs) and the average fundamental frequency of the speaker (F_0) (although a fixed value for F_0 may be used as well).

The spectral shaper uses Short Time Fourier Transform (STFT) with a Hann window to calculate short-term magnitude spectra. The analysis frames overlap in order to achieve perfect reconstruction of the modified speech signal. However, the energy is attenuated at the two ends of the processed segment where the frames do not overlap. In (rt)SSDRC this can potentially cause audible artifacts, unless two successive segments overlap by at least one frame. Also, the overlapping between successive segments helps to avoid discontinuities in the output of both sub-systems of (rt)SSDRC. The prototype system that is used in the demonstration works with segments that consist of 4 frames (approx. 77 ms at $Fs = 16$ kHz and $F_0 = 130$ Hz) where two successive segments overlap by 2 frames.

2.4. Non-causal operations

For the real-time development of SSDRC, three non-causal operations used in the (bs)SSDRC should be modified accordingly. In (bs)SSDRC, these operations require the a priori knowledge of the whole signal. The first operation is the normalization of the local voicing value (defined by zero-crossing and energy criteria) by the maximum value of voicing detected in the signal. The second is the calculation of the maximum of the smoothed envelope estimated in the signal. That is used by the DRC part in order to rearrange the energy of speech waveform over time. Finally, the third part is the calculation of the global multiplication constant that is used to preserve the energy of signals after enhancement. All three parameters are computed from their statistics (e.g., mean values) using speech corpora like TIMIT. These values are then used as initial guess during the real-time process, which are then updated as more frames of input speech become available. It has been noticed that the first two parameters (probability normalizing constant and reference level) quickly attain values close to their optimal values (if the whole signal was known a priori). The global multiplication constant, after a short transient, reaches a value slightly above the corresponding parameter of (bs)SSDRC (it actually has low amplitude oscillations around the optimal value).

3. IMPLEMENTATION AND USE

The (rt)SSDRC library is written in C and is compiled into a shared library (.dll in Windows and .so in Linux). The demo program is written in Java and calls the (rt)SSDRC library through the Java Native Interface. The Java program starts a thread that handles the input/output operations using classes

from the javax.sound.sampled package and a thread that runs the (rt)SSDRC. The first thread, which runs in high priority, continuously reads 64 bytes from the microphone or from an input stream and writes these bytes in a circular buffer C1, then it reads 64 bytes from another circular buffer C2 and writes them to sound card. The second thread, continuously reads a segment of data from C1, modifies it, and writes the results to C2. The access to circular buffers is synchronized in order to avoid race conditions. Note that, the processing requirements of (rt)SSDRC are low and, apart from an initial delay, the input/output thread always finds data in buffer C2. The circular buffers offer three major benefits: a) they protect against delays caused by the scheduling of threads from the operating system, b) they allow (rt)SSDRC to use segments whose length does not depend on the hardware (microphone, sound card) audio buffers length, and c) they facilitate the processing of overlapping segments.

The block diagram of (rt)SSDRC demo and a screenshot of its graphical user interface are shown in Figs. 1 and 2, respectively.

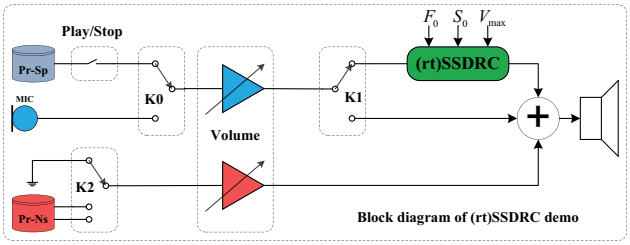

Fig. 1. Block diagram of suggested software demonstrator.

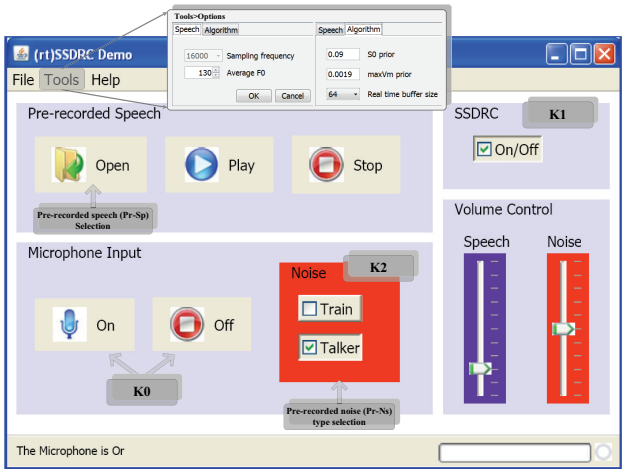

Fig. 2. The main form of (rt)SSDRC showcase.

The demo will demonstrate the performance of (rt)SSDRC with both live (captured from a microphone) or pre-recorded speech (switch K_0 in Fig. 1), in various noise conditions. Thus, three noise sources can be used (switch K_2). The first one is the noise from the environment where the listener is located, while the other two are stationary and fluctuating (competing talker) pre-recorded noise types. Additionally, the user can independently alter the volume of speech and noise signals. Finally, unmodified clean speech or enhanced by (rt)SSDRC is mixed with selected noise maskers and presented to the listener though loudspeakers (or headphones).

4. CONCLUSIONS AND FUTURE DEVELOPMENTS

In this paper we have presented a real-time implementation of a speech-in-noise intelligibility enhancement algorithm based on spectral shaping and dynamic range compression, SSDRC, which has recently shown to outperform state-of-the-art approaches in this field. First, a low-complexity (rt)SSDRC algorithm was implemented in C code, then a software demonstrator based on the Java platform was built. The demo applies online SSDRC enhancement on clean speech signals captured from microphone (or saved on disk) and it allows to combine modified voices with various noise maskers. It provides low-delay and artifact-free enhanced signals, while the Java framework makes the code portable to any compatible devices, such as mobile phones, tablets, electronic sound books readers etc.

As future work, we intend to evaluate SSDRC at enhancing noisy (not clean) speech signals captured from acoustically difficult environments where most everyday face to face (F2F) communicative scenarios take place.

5. REFERENCES

[1] T. Baer, B. Moore, and S. Gatehouse. Spectral contrast enhancement of speech in noise for listeners with sensorineural hearing impairment: effects on intelligibility, quality, and response times. *J. Rehab. Res. Dev.*, 30(1):49–72, 1993.

[2] M. Cooke, C. Mayo, C. Valentini-Botinhao, Y. Stylianou, B. Sauert, and Y. Tang. Evaluating the intelligibility benefit of speech modifications in known noise conditions. *Speech Communication*, (55):572–585, 2013.

[3] E. Godoy, M. Koutsogiannaki, and Y. Stylianou. Approaching speech intelligibility enhancement with inspiration from Lombard and clear speaking styles. *Computer Speech & Language*, 28(2):629–647, 2014.

[4] V. Hazan and R. Baker. Acoustic-phonetic characteristics of speech produced with communicative intent to counter adverse listening conditions. *J. Acoust. Soc. Am.*, 130:2139–2152, 2011.

[5] LISTA Showcase. *http://listening-talker.org/showcase*.

[6] B. Moore. Speech processing for the hearing-impaired: successes, failures, and implications for speech mechanisms. *Speech Communication*, 41:81–91, 2003.

[7] B. Sauert and P. Vary. Near end listening enhancement: Speech intelligibility improvement in noisy environments. In *Proc. ICASSP*, pages 493–496, 2006.

[8] Y. Tang and M. Cooke. Optimised spectral weightings for noise-dependent speech intelligibility enhancement. In *Proc. Interspeech*, 2012.

[9] T. Zorilă, V. Kandia, and Y. Stylianou. Speech-in-noise intelligibility improvement based on spectral shaping and dynamic range compression. In *Proc. Interspeech*, pages 635–638, 2012.