34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Adding Real-Time Noise Suppression Capability to the Cochlear Implant PDA Research Platform*

Taher Mirzahasanloo, Vanishree Gopalakrishna, Nasser Kehtarnavaz, *Fellow, IEEE*, and Philipos Loizou, *Senior Member, IEEE*

Abstract— **This paper presents the real-time implementation of an environment-adaptive noise suppression algorithm on an FDA-approved PDA platform for cochlear implant studies. This added capability involves identifying the background noise environment in real-time and adapting a data-driven noise suppression approach to that noise environment on-the-fly. Various software optimization steps are taken in order to achieve a real-time throughput on the PDA platform involving both the speech decomposition and the adaptive noise suppression components. Real-time timing results and a quantitative measure of noise suppression are presented.**

I. Introduction

The number of cochlear implants (CI) patients has increased significantly during the last decade [1]. A personal digital assistant (PDA) interface research platform has been recently approved by FDA for clinical studies with Nucleus CI patients [2]. This platform allows the assessment of speech processing algorithms for CI studies. The real-time implementation of the CI speech processing pipeline on this platform was covered in our previous work [3, 4].

In this work, a noise suppression capability is added to this platform that is designed to run in real-time in conjunction with the speech processing pipeline. This capability is useful and necessary as it has been shown that in noisy environments the speech understanding of CI patients decreases significantly [5, 6]. This added component or path consists of a noise feature extractor, a noise classifier, and a noise suppression module. The classifier uses features from the noise signal to identify the background noise environment in order to switch to those parameters of the noise suppression module that are optimized for that particular noise environment. The thrust of this paper is on the optimization steps taken in order to allow real-time implementation of all the modules on the FDA-approved PDA research platform.

The paper is organized as follows. Section II gives an overview of the CI speech processing path together with the details of the adaptive noise suppression component. Section III discusses the optimization steps taken for the purpose of achieving its real-time implementation on the PDA platform together with the timing outcomes. Section IV includes the noise classification and suppression results that are obtained when running the entire speech processing pipeline in real-time. Finally, the conclusions are stated in section V.

*This work was supported by grant No. DC010494 from NIDCD/NIH.
T. Mirzahasanloo, V. Gopalakrishna, N. Kehtarnavaz, and P. Loizou are with the Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75080 USA (phone: 972-883-6838; fax: 972-883-2710; e-mail: kehtar@utdallas.edu).

II. Cochlear Implant Speech Processing Pipeline

Speech processing in a CI system includes decomposition of the input signal into different channels and extracting the channel envelope by summing up the power from all frequency bins falling within the channel bandwidth. In [3], we used a recursive wavelet-packet transform (WPT) to decompose the input speech into different channels. The extracted channel envelopes were compressed and an n-of-m strategy were then applied, where n denotes the maximum amplitude channels out of total of m channels that are selected at any time for generating CI stimulation pulses.

A noise suppressor along the speech processing path is added in this work, see Fig. 1, in order to track the noise spectra and apply an appropriate spectral weighting to suppress it [7-9]. The weighting function is derived using a log-MMSE estimator. To take into consideration the variability in the noise spectra for different types of background noise, several environment-specific noise suppressors have been proposed in the literature. In this work, we have considered a data-driven approach due to its computational efficiency. In order to change the noise suppressor parameters based on the background noise environment, a feature extractor and a classifier are deployed.

A. *Noise suppression*

In the spectral domain, a gain function is assumed to be applied on the magnitude spectrum of the input noisy speech signal providing an estimate of the associated clean spectrum. This gain is represented as a function of prior and posterior SNRs minimizing the mean squared error over a training set of noisy and clean sample pairs [9]. Decision-directed approach is the most commonly used method to estimate the prior SNR. However, as discussed in [10], this approach leads to biased and erroneous results as some SNR values cause underestimation or overestimation of noise spectra. In addition, the gain function solution obtained using the MMSE and log MMSE estimators in [9] assume specific distributions for the noise and speech spectra which may not necessarily be the best fitting distributions.

To account for such modeling and estimation shortcomings, a data-driven approach, as proposed in [10-13], is adopted here where the gain values are obtained via a minimization formulation. For non-stationary noise tracking, the tabular representation is considered to provide an estimation of noise spectrum. Then, this estimate is used in

the analytic gain suppression function to provide the enhanced magnitude spectrum, e.g. the log-MMSE estimator as used in [10]. The data-driven nature of this approach allows one to optimize the gain representation independently for each noise environment by considering the corresponding dataset.

Since this solution is optimal in the MMSE sense, it outperforms the conventional model-based methods as shown in [10]. Noting that different gain table parameters are optimized and used for different noise types, the overall performance improves over that of a fixed-noise suppression approach.

B. Background noise environment detector

The noise classification path is activated for frames which contain only noise. To determine if the incoming frame is noise only, a voice activity detector (VAD) using an adaptive threshold for subband power is used here. Subband power is computed using the wavelet coefficients which are already computed as part of the decomposition path, hence making the VAD computationally efficient.

To characterize the noise frames for classification, a 26-dimensional feature vector consisting of a combination of MFCCs (mel-frequency cepstral coefficients) with their first derivatives is utilized as it is found that such a feature vector provides high classification rates while not being computationally intensive. In our previous study [14], a support vector machine (SVM) classifier with a radial basis kernel was used. However, to perform multiclass noise classification, SVM becomes computationally very expensive and thus a Gaussian mixture model (GMM) with two Gaussian mixtures is employed here as it provides a balance between classification accuracy and computational complexity. The parameters of the GMM classifier are estimated using the k-means clustering and expectation maximization (EM) algorithms.

III. Real-Time Implementation on PDA Platform

The FDA-approved PDA platform consists of a 624 MHz clock rate ARM processor. The coding was done in C.

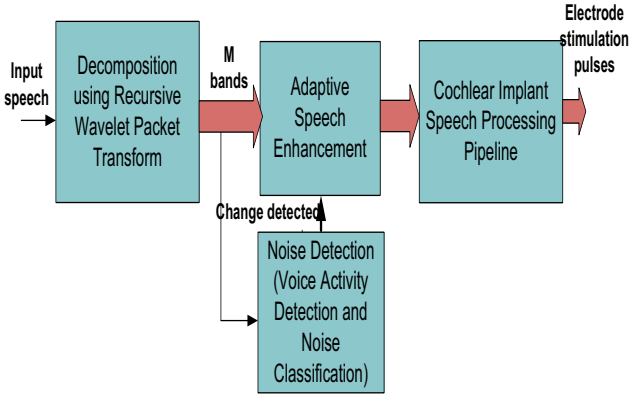

Figure 1. Environment-adaptive noise suppression component added to the FDA-approved PDA cochlear implant research platform.

Initially, the code was written in floating-point but it failed to run in real-time due to the processor being fixed-point. Hence, the code was rewritten in fixed-point using 32-bit word length in Q15 and Q24 formats depending on the required precision at different parts of the code. To further reduce the computational load, the envelope extraction and compression sections of the code were written using 16-bit word length as this sufficed the required precision for these sections. Table I reports the times required by different modules to process 11.6 ms duration frames (or 256 samples at 22050 Hz sampling rate).

As can be seen, neither the floating-point nor the fixed-point versions of the code could be run in real-time, i.e. the total time being greater than 11.6 ms of frame time. In what follows, the optimization steps taken to reduce the total processing time and thus to achieve the real-time implementation are mentioned:

1- **Look-up table (LUT)** – The sections of the code consisting of exponential integral, dB to linear and linear to dB conversion were implemented as LUTs to reduce the computational load. For a 32-bit fixed-point implementation, having an entry for each possible input led to a very large LUT with 2^{32} entries, hence the number of entries in the table had to be reduced. For a non-linear function as shown in Fig. 2(a), having table entries which were linearly spaced along the input range, marked with '+' along the input output curve, led to large quantization errors. To improve the design of LUTs of such functions, a table with the same number of entries was used that was linearly spaced along the output as shown in Fig. 2(b). This way there were no more entries in the table in the region where the output was dynamically changing for a small change in the input compared to the region where the output was slowly changing with the input.

However, since the key for the table entries was not linearly spaced, this added additional table search time for a given input. In our implementation, the LUT was divided into different regions where the input key was linearly spaced in each region; spacing in each region was made dependent on the dynamic nature of the output as shown in Fig. 2(c). Fig. 2 shows the error between the actual output and the output obtained by looking up the nearest entry in the LUT. The MSE for the three cases shown were 0.028, 0.004 and 0.003, respectively. The LUT was constructed as illustrated in Fig. 2(c) for an exponential integral function with the input in Q15 having a maximum of 2^{-15} deviation from the actual value. This reduced the total required memory to only 8% as compared to the LUT which had an entry for all possible inputs.

2- **Search optimization**- As mentioned above, in order to reduce the number of entries in the LUT and also maintain the accuracy, the LUT was designed in such a way that the key was not placed linearly along all possible input values. The LUT was divided into different regions where the input key to the table was linearly spaced along the input values, and different regions had different spacings between the keys. As a result, the LUT was arranged in a balanced tree structure with each branch corresponding to a different region with a different spacing. Such an arrangement made the search for an entry in the table possible with few binary

operations on the fixed-point input. A 3-level tree structure was constructed for this LUT in order to compute the exponential integral; hence searching the table took only 3 comparisons.

3- **Series expansion**- There were multiple places where linear to dB scale conversion and other log computations were needed. For such computations, the LUTs were approximated using series expansion. The number of terms used in the expansion depended on the accuracy required and the actual input value.

4- **Memory management**- Static memory allocations were used to avoid any dynamic memory allocation. The arrangement of the LUTs along a binary tree reduced the requirement of a contiguous memory space as different regions of the tree could be stored separately. Allocation of a contiguous memory block for several frequently used constants was made to avoid pages being missed during memory accesses.

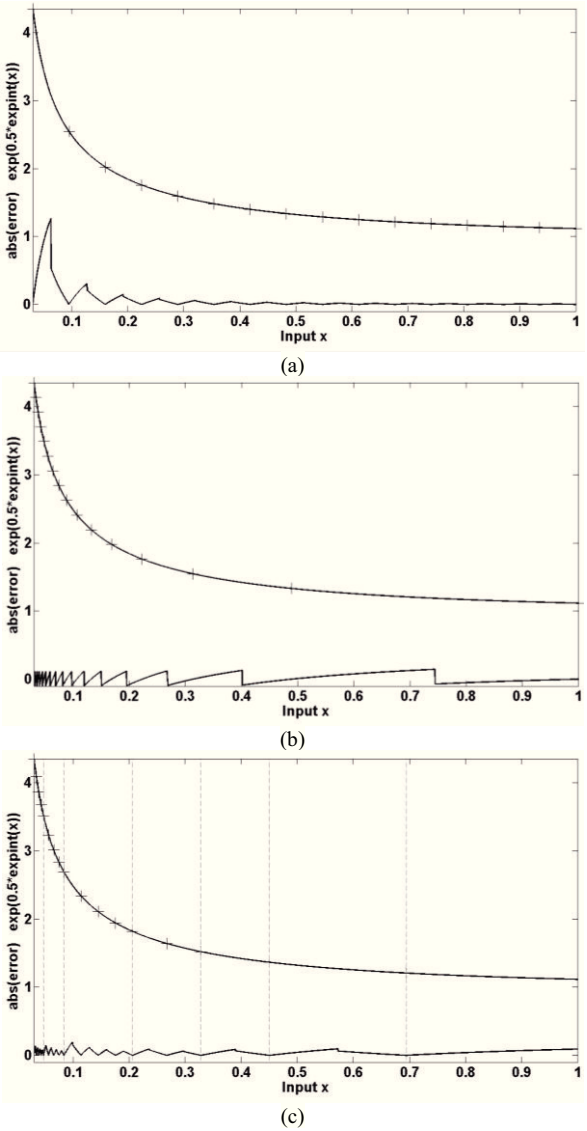

Figure 2. Look-up table entries and error between actual output and LUT output; LUT designed such that table entries are spaced (a) linearly along input, (b) linearly along output, (c) linearly along input with different spacing in different regions.

TABLE I. TIMING PROFILE IN MS OF THE ENTIRE CI SPEECH PROCESSING PIPELINE FOR 11.6 MS FRAMES.

	Total Time	**A**	**B**	**C**	**D**
Floating-Point	95.00	34.36	34.62	34.05	0.08
Fixed-Point (Non-optimized)	28.33	1.50	15.21	11.17	0.07
Fixed-Point (Optimized)	**7.77**	1.32	2.55	2.41	0.06

A: *Recursive WPT decomposition*, **B**: *Speech enhancement*, **C**: *Noise detection*, **D**: *Channel envelope computation*.

5- **Reducing classifier frame rate**- The rate at which the classifier was activated was reduced by alternating the 'noise only' frames. This reduction in classifier frame rate delayed the detection of a change in the noise environment by one frame (11.6 ms) while not causing any delay to the decomposition path.

Table I shows the processing times of different modules using different implementations of the complete CI system which include wavelet decomposition, noise suppression, channel envelope computation, lowpass filtering, envelope compression, VAD, feature extraction, and classification. As can be seen from row 3, after incorporating all the optimizations discussed above, the real-time implementation on the PDA platform was made possible, i.e. processing of 11.6 ms frames took only 7.7 ms.

IV. REAL-TIME SPEECH ENHANCEMENT RESULTS

The PDA was taken to four different most commonly encountered noise environments namely street, car, restaurant and mall to capture segments of noise for training. The recordings were collected using the BTE (Behind-The-Ear) microphone as worn by CI patients. For each environment, 5 sample files of 1-minute long duration were collected. In every recording, the integrated (average) sound pressure levels (SPLs) for the run periods of 1 minute were noted as 75.8 dBA for street, 66.4 dBA for car, 71.2 dBA for restaurant and 67.8 dBA for mall noise environments. The two environments of restaurant and mall noise included babble.

Two performance measures of classification rate and speech enhancement are reported here. The classifier was trained using 50% of the collected data, and the rest was used for testing, with no overlap between the training and testing samples. The average or overall correct classification

rate was found to be 91.5% noting that the misclassifications were not disruptive to the enhancement process. That is to say when misclassifications occurred, the gain table of the environment with the highest similarity with the actual environment was used.

To examine the adaptive noise suppression method, we used 20 sentences of approximately 2 seconds duration for training. Noisy speech files were created by adding the four noise types to the clean speech sentences at 9 different SNR levels. These noisy files were used to generate the optimized gain look-up table as discussed in section II-A. To evaluate how the trained gain table performed in suppressing noise, for each noise type, the noise files for testing and training were considered to be exclusive. The performance of each gain function was tested at 5 dB SNR and compared to the non-adaptive or fixed noise suppression approach of log-MMSE algorithm [9]. Perceptual evaluation of speech quality (PESQ), an ITU-T recommended standardized objective quality measure [15], was used here to assess the performance of noise reduction algorithms in terms of quality. This measure gives a score between 0.5 and 4.5 with higher numbers representing better quality. Fig. 3 illustrates the PESQ scores for the noisy and the speech enhanced outcomes using the fixed and adaptive suppression methods for the four noise types of street, car, restaurant and mall, averaged over 700 different speech sentences. As can be observed from this figure, higher (predicted) quality ratings were obtained after incorporating the above discussed adaptive noise-suppression techniques relative to the no-noise suppression and fixed-noise suppression conditions.

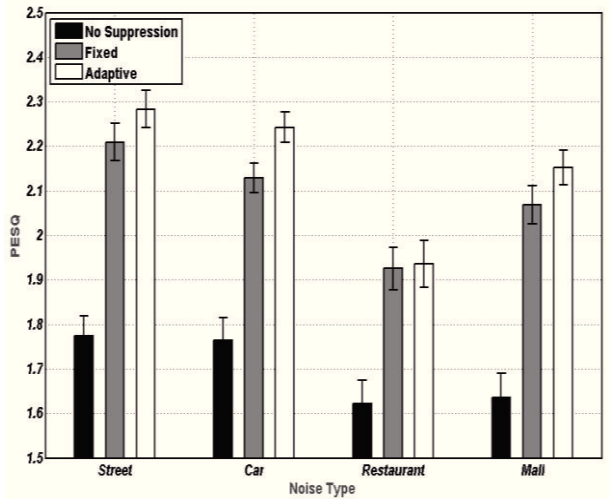

Figure 3. PESQ quality scores obtained with no noise suppression, fixed -noise suppression, and adaptive-noise suppression methods (scores on original noisy signals with no suppression are considered as baseline).

V. CONCLUSION

This paper has presented the optimization steps taken in order to add a real-time noise suppression capability to the FDA-approved PDA research platform for cochlear implant studies. These steps have included fixed-point techniques, efficient look-up table design, search optimization, series expansion, memory management, and classification frame rate reduction. This added capability is expected to benefit CI patients as the current PDA research platform does not include any noise suppression component.

REFERENCES

[1] National Institute on Deafness and Other Communication Disorders, "Cochlear implants," http://www.nidcd.nih.gov/health/hearing/coch.asp.

[2] P. Loizou, A. Lobo, D. Kim, N. Gunupudi, V. Gopalakrishna, N. Kehtarnavaz, H. Lee, S. Guo, and H. Ali, "Open architecture research interface for cochlear implants," *Final Report NIHN01DC60002*, 2011. http://utd.edu/~loizou/cimplants/.

[3] V. Gopalakrishna, N. Kehtarnavaz, and P. Loizou, "A recursive wavelet-based strategy for real-time cochlear implant speech processing on PDA platforms," *IEEE Trans. Biomed. Eng.* 57(8), pp. 2053-2063, 2010.

[4] V. Gopalakrishna, N. Kehtarnavaz, and P. Loizou, "Real-time implementation of wavelet-based advanced combination encoder on PDA platforms for cochlear implant studies," *IEEE Int. Conf. on Acoust., Speech, and Sign. Proces., ICASSP*, 2010.

[5] J. Remus, and L. Collins, "The effects of noise on speech recognition in cochlear implant subjects: Predictions and analysis using acoustic models," *Eurasip J. Appl. Sign. Proces.* (18), pp. 2979-2990, 2005.

[6] B. Fetterman, and E. Domico, "Speech recognition in background noise of cochlear implant patients," *Otolaryngol. Head. Neck. Surg.* 126(3), pp. 257-263. 2002.

[7] Y. Hu, P. Loizou, N. Li, and K. Kasturi, "Use of a sigmoidal-shaped function for noise attenuation in cochlear implants," *J. Acoust. Soc. Am.* 122(4), pp. EL128-EL134. 2007.

[8] P. Loizou, A. Lobo, and Y. Hu, "Subspace algorithms for noise reduction in cochlear implants," *J. Acoust. Soc. Am.* 118(5), pp. 2791-2793, 2005.

[9] Y. Ephraim, and D. Malah, "Speech enhancement using a minimum mean-square error-log-spectral amplitude estimator," *IEEE Trans. Acoust. Speech Sign. Proces.* 33(2), pp. 443-445, 1985.

[10] J. Erkelens, J. Jensen, and R. Heusdens, "A data-driven approach to optimizing spectral speech enhancement methods for various error criteria," *Speech Commun.* 49(7-8), pp. 530-541, 2007.

[11] J. Erkelens, and R. Heusdens, "Tracking of nonstationary noise based on data-driven recursive noise power estimation," *IEEE Trans. Audio Speech Lang. Proces.* 16(6), pp. 1112-1123, 2008.

[12] P. Loizou. "Speech enhancement based on perceptually motivated bayesian estimators of the magnitude spectrum," *IEEE Trans. Speech Audio Proces.* 13(5), pp. 857-869, 2005.

[13] T. Fingscheidt, S. Suhadi, and S. Stan, "Environment-optimized speech enhancement," *IEEE Trans. Audio Speech Lang. Proces.* 16(4), pp. 825-834, 2008.

[14] V. Gopalakrishna, N. Kehtarnavaz, P. Loizou, and I. Panahi. "Real-time automatic switching between noise suppression algorithms for deployment in cochlear implants," *IEEE Eng. Med. Biol. Soc.,* pp. 863-866, 2010.

[15] ITU-T, "Perceptual evaluation of speech quality (PESQ): An objective method for end-to-end speech quality assessment of narrowband telephone networks and speech codecs," *ITU-T Recommendation P. 862*, 2000.