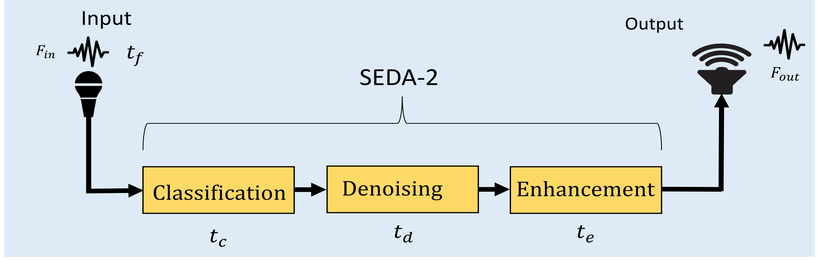
A REAL-TIME ANDROID APP FOR MULTI-TALKER BABBLE NOISE REDUCTION.

NYU School of Medicine
NYU LANGONE MEDICAL CENTER

Roozbeh Soleymani, Ivan W. Selesnick, David M. Landsberger

NYU TANDON SCHOOL OF ENGINEERING

Overview: Cochlear Implant users usually do not perform well in the presence of the background noise. Several single-channel de-noising algorithms have been previously designed to address this problem. Nevertheless, designing a de-noising algorithm which is capable of performing well for non-stationary noise (e.g. Multi-talker babble) still remains a difficult task. The problem becomes more challenging if functioning in real-time and having a low latency are added to the list of the algorithm's desired properties. We have designed a low latency, real-time babble noise reduction (SEDA2) which maintain these properties using devices with limited processing power such as a smart phone. The algorithm has been tested on both CI users and NH subjects yielding promising results. The algorithm consists of three main stages: 1- Classification 2-De-noising 3-Enhancement. We also have developed a prototype app (Bab-El) for Android cell phones based on a slightly modified version of the SEDA2. The Bab-El app can perform real time denoising on an ordinary android device with relatively low latency.



1-Classification: The first stage classifies very short frames of noisy speech into either noise or speech dominated classes. The classifier employs a number of novel features which maintain their robustness even for very short audio frames. Using weighted PCA the features are de-correlated and then using EM (Expectation maximization) algorithm a GMM (Gaussian Mixture Model) is created for the classification.

1-1 Feature Selection: Four features sensitive to changes of SNR in short frames of target speech mixed with multi-talker babble noise, were selected.

Feature	Parameters	Formula	
Entropy	Histogram bin width and number	$f_i^{(1)} = -\sum_{k=1}^{N} P(k) log_{10}(P(k)) = -\sum_{k=1}^{N} \frac{h(k)}{L} log_{10}(\frac{h(k)}{L})$	h: Amplitude histogram of F_i , $P(k)$: Probability of the kth bin, N: Number of bins, L: Frame length.
Post to pre thresholding RMS ratio	Thresh. level	$\tau(F_i) = \frac{1}{L} K \|F_i\|_1$, $f_i^{(2)} = \frac{rms(F_i^{th})}{rms(F_i)}$	$\|F_i\|_1$: l_1 norm of the frame F_i.
Envelope Mean-Crossing	Moving avg. window length	$f_i^{(3)} = \mathrm{var}\left(\frac{1}{\max(e_i)L_w} \sum_{k=-\frac{L_w}{2}}^{\frac{L_w}{2}} \|F_i(k+nh)\| w(k)\right)$	e_i: Frame's envelope, L_w: Window (w) length, h: Hop size.
Envelope Variance		$f_i^{(4)} = \frac{1}{2N_w} \sum_{k=2}^{N_w} \|\mathrm{sign}(\hat{e}_i(k) - \mu_{\hat{e}_i}) - \mathrm{sign}(\hat{e}_i(k-1) - \mu_{\hat{e}_i})\|$	N_w : Total number of windows in a frame , $\hat{e}_i$: Frame's normalized envelope

1-2 Feature Optimization: To optimize the quality of features, its Fischer score (S) was numerically maximized and the suitable values for feature parameters were selected.

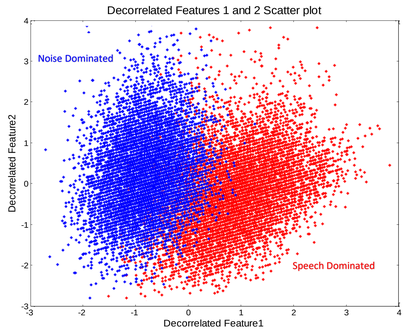
$S = \frac{\sum_{j=1}^{N_c} n_j(\mu_j - \mu)^2}{\sum_{j=1}^{N_c} n_j \sigma_j^2}$, N_c: Number of classes (i.e., $N_c = 2$), μ_j: Mean of the feature in class j, μ: Overall mean of the feature, ρ_j: Variance of the feature in class j, n_j: Number of samples in class j.

1-3 Feature Decorrelation with Weighted PCA (Principle Component Analysis):

To reduce the correlation (redundancy) between the features, we use PCA to generate a new smaller set of uncorrelated features. To take the quality of each feature into account we give a relative weight to each feature based on its Fischer quality score.

$\mathcal{F}_0 = \mathcal{F} - M$, $\mathcal{F}_d = T\mathcal{F}_0$, $C_d = \frac{1}{N}\mathcal{F}_d \mathcal{F}_d{}^T = \frac{1}{N}[T\mathcal{F}_0][T\mathcal{F}_0]^T = T[\frac{1}{N}\mathcal{F}_0\mathcal{F}_0{}^T]T^T = TC_0T^T$, $C_d = V^TC_0V \Rightarrow T = V^T$, $C_0 = \frac{1}{N}W\mathcal{F}_0\mathcal{F}_0{}^T W^T$.

$\mathcal{F}$: Feature matrix , M: Mean matrix of features, T: Transformation matrix, C_0 : Covariance matrix of F_0, F_d : De-correlated feature matrix, , C_d: Diagonal rank-ordered covariance matrix of F_d, W: Weighting matrix



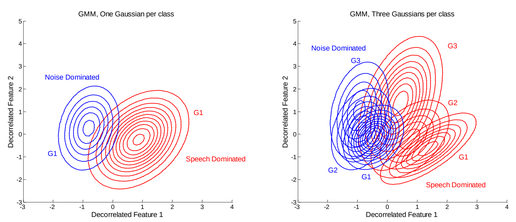
1-4 Training with GMM and EM:

For the classifier, we use the two dimensional Gaussian Mixture Model (GMM) where each class is modeled as the sum of a n Gaussian distributions . In order to train our model, we use the iterative Expectation-Maximization (EM) algorithm

GMM : $G(\mathcal{F}_d|\mu, w, C) = -\sum_{i=1}^{n} \frac{w_i}{(2\pi)^{\frac{d}{2}}\sqrt{|C_i|}} e^{\{-\frac{1}{2}[\mathcal{F}_d - \mu_i]^T C_i^{-1}[\mathcal{F}_d - \mu_i]\}}$

EM: maximization $log\{p(\mathcal{F}|\mu, C, w)\} = \sum_{k=1}^{N_F} log\{\sum_{i=1}^{N_g} w_i \mathcal{N}(f^k|\mu_i, C_i)\}$

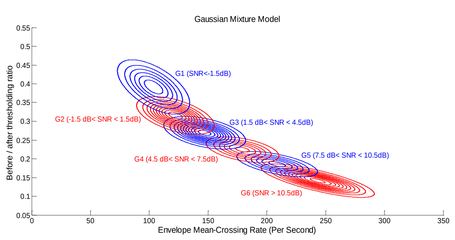
$\mu_i^{new} = \frac{\sum_k p_i^k \mathcal{F}^k}{\sum_k p_i^k}$ $\omega_i^{new} = \frac{\sum_k p_i^k}{N_F}$ $C_i^{new} = \frac{\sum_k p_n^k (\mathcal{F}^k - \mu_i^{new})(\mathcal{F}^k - \mu_i^{new})^T}{\sum_k p_n^k}$



1-5 Classification using MAP (Maximum a posteriori estimation)

Probability of each test feature set $\mathcal{F}$ belonging to a class X : $argmax_X[P(\mathcal{F}|class_X)P(class_X)]$, $P(\mathcal{F}|class_X) = \sum_{i=1}^{n} w_i \mathcal{N}(\mathcal{F}|\mu_i, C_i)$.

μ_i , C_i and w_i are obtained from the GMM. The values of $P(class_N)$ and $P(class_S)$ change as a function of the overall (long term) SNR. In the case of fast varying noisy condition) we can assume $P(class_N) = P(class_S) = 0.5$. In the case of slowly varying overall SNR, we can estimate more accurate values for $P(class_N)$ and $P(class_S)$ by roughly estimating the global SNR. To estimate the global SNR we suggest a very simple classifier which uses only two features (RMS ratio and envelope mean crossing) calculated over the long frames of the noisy speech without de-correlating the features with PCA. We use GMM with a single Gaussian per class for training the overall SNR classifier .

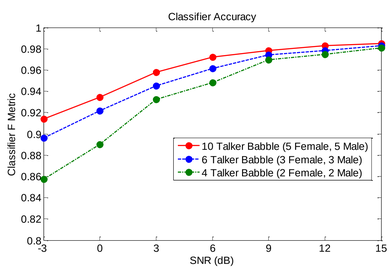


1-6 Performance evaluation:

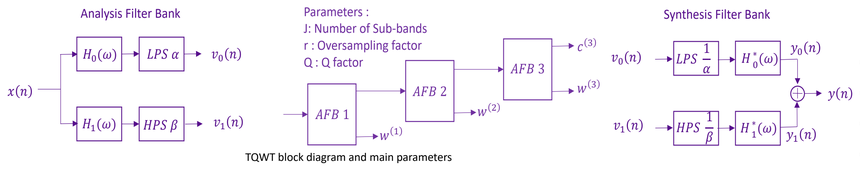
A database of 2,100 sentences, including 720 male speaker and 720 female speaker IEEE standard sentences, 260 male speaker HINT sentences and 400 male speaker SPIN sentences was used to create babble and speech samples. To create each babble sample, the number and gender of talkers were randomly selected. The number of talkers varied from 5 to 10. The performance of the classifier was evaluated using two-fold cross validation. First, the classifier was trained with noisy speech samples randomly created from half of the sentence database. Then the resulting classifier was evaluated using test samples created from the second half of the sentence data base. Then we replaced the testing and training database and repeated the same process. The average value of F accuracy metric was measured :

$P = \frac{C}{C+f^+}$ $R_N = \frac{C}{C+f^-}$ $F = \frac{2PR}{P+R}$

where C, $f^+$and f^- are correct, false positive and false negative detection, respectively.



2-Denoising and Enhancement: The representation of the clean speech samples in an oversampled Tunable Q-factor Wavelet Transform (TQWT) exhibits some degree of group sparsity which does not exist in babble samples. Moreover, the distribution of the center frequencies of the sub-bands and the shape of the frequency responses of the TQWT resemble Mel-scale and Gammatone filter banks that are designed to reflect the human auditory system



TQWT block diagram and main parameters

2-1 Updating the threshold level: Threshold levels in each sub-band depend on the average noise level over the last few noise dominated frames. μ_i: Estimated noise level for sub-band i, obtained by averaging $l_1$norm of that sub-band over the last M noise dominated frames, $F_n^{(k)}$: Last kth noise dominated frame , $w_i^{(k)}$: ith sub-band of $F_n^{(k)}$ in TQWT domain and J : total number of levels in TQWT (denoted with φ).

$\mu_i = \frac{1}{M}\sum_{k=1}^{M} \|w_i^{(k)}\|_1$ $w^{(k)} = \{w_1^{(k)}, w_2^{(k)}, \dots, w_{J+1}^{(k)}\} = \varphi(F_n^{(k)})$ $\mu_i^{new} = \frac{(M-1)\mu_i^{old} + \|w_i^{(M+1)}\|_1}{M}$

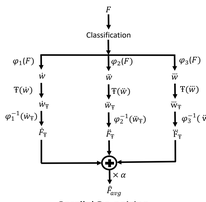
2-2 Adaptive Group Thresholding :

For noisy speech frame F frame: $w = \varphi(F)$ where $w = \{w_1, w_2, \dots\dots, w_{J+1}\}$, $w_i = \{c_1, c_2, \dots\dots, c_{n_i}\}$ $c_1$to c_{n_i} are coefficient-groups of w_i. For each coefficient-group c_k of sub-band w_i we define:

$r_k^{(i)} = n_i \frac{\|c_k\|_1}{\|w_i\|_1}$ $\hat{c}_k = \begin{cases} H_T(c_k), & r_k^{(i)} \le \gamma \\ S_{\epsilon T}(c_k), & r_k^{(i)} > \gamma \end{cases}$, $T = \frac{\rho\tau\mu_i}{L_i}$

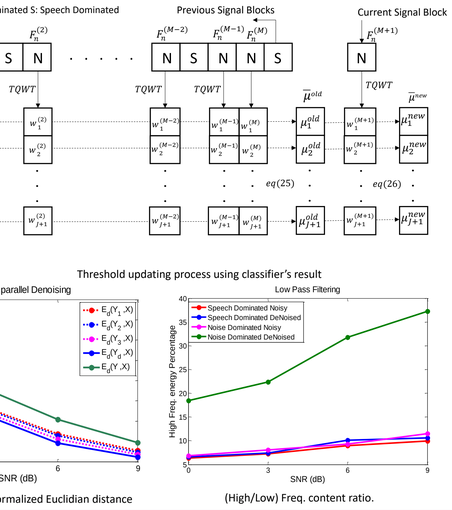
L_i: Length of sub-band i, τ: controls the thresholding aggressiveness based on the frame's class. ρ: determines our desired overall denoising aggressiveness, ϵ : Reduction factor for soft thresholding , γ: should always be greater than 1.

2-3 Parallel De-noising: First we create three distinct representations of the signal in the wavelet domain using three TQWTs with different settings. Then we apply adaptive group thresholding to each representation and create three slightly different de-noised versions of the same signal. Three resulting de-noised signals will eventually be averaged. To increase the de-noising performance, three TQWTs should have low, medium and high Q factors respectively. This will assure three different representations in the wavelet domain. The redundancy and number of levels in each TQWT should be selected so that the signal's energy is distributed over many sub-bands.



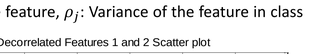
Parallel De-noising

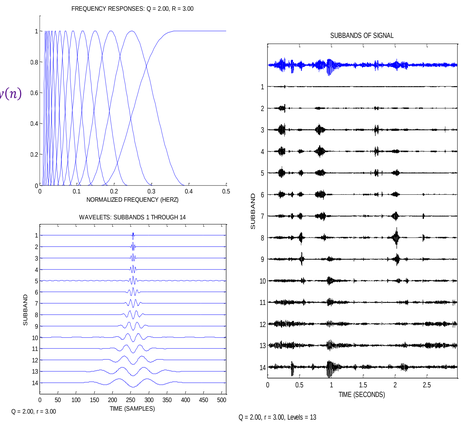
2-4 Enhancement: Adaptive group thresholding is adjusted based on the noise level. Hence it significantly alters the babble structure and reduces it to sporadic and isolated coefficients with high frequency content. To investigate this, we measured the high frequency content of speech and noise dominated frames, after and before denoising. The energy of high frequency components remains nearly constant in speech dominated frames, after and before parallel denoising whereas it drastically increases in noise dominated frames. To exploit the above mentioned property, after parallel denoising we apply a suitable low-pass filter only to the noise dominated frames, to remove the high frequency residual components resulting from the previous denoising steps and further enhance the speech quality. In SEDA we used a 6th order Butterworth low pass filter with cut-off frequency of 4000 Hz.



Threshold updating process using classifier's result

Clean to Denoised Normalized Euclidian distance (High/Low) Freq. content ratio.

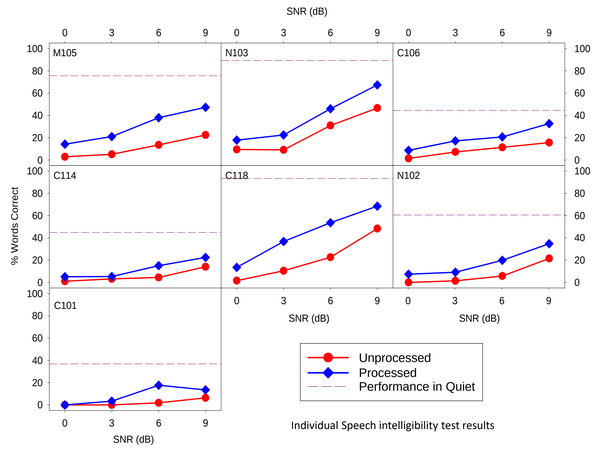




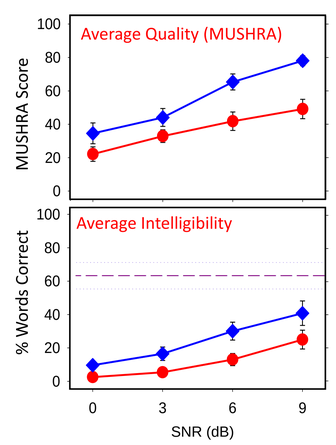
TQWT sub-bands, frequency response and wavelets.

3-Testing:

- 7 Cochlear Implant users
- IEEE standard sentences
  - with and without SEDA processing
  - SNRs of 0, 3, 6, and 9 dB
  - Stimuli: 4 randomly selected IEEE sentence lists for each condition (without replacement).
  - Noise: Randomly created 10-talker (5 male and 5 female) babble (see 1.6)
  - **Average improvement between 7.19% to 17.19% depending on SNR.**
- Sound quality measured with MUSHRA (Multiple Stimuli with Hidden Reference and Anchor)
  - **SEDA was rated as sounding better at all SNR levels.**

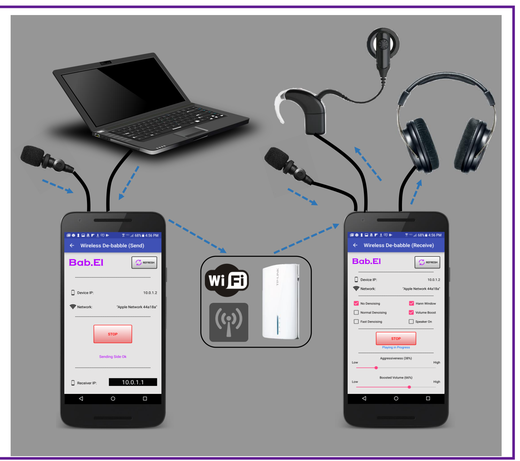


Individual Speech intelligibility test results



4-Bab-El Android app: We have designed a prototype cell-phone app. (Bab-EL) designed for Android devices based on SEDA2 algorithm. The current version of the Bab-El can run in real-time on most Android devices and introduces a latency as low as 20 milliseconds. Note that this latency will be added to the relatively high latency inherent in the Android phones. Some of the Bab-El features are briefly described as follows:

- **Calibration:** To eliminate the effect of variations in the microphone's frequency response and sensitivity in different devices, we should calibrate the app before using it for the first time. In this mode the user is asked to speak normally for 10 seconds in a quiet environment. The app analyses the audio and adjust the SEDA classifier based on the result. This mode also checks the speed of the phone and makes suggestions for the optimal de-noising settings.
- **Settings:** The app has 12 predefined settings which can be selected based on the noise type and phone performance. The settings mainly differ in frame length, window type and wavelet and classifier parameters. The user also can create new settings by choosing the wavelet and classifier parameters.
- **Save and De-babble:** This mode performs the de-noising on the recorded audio samples and saves the de-noised signal.
- **Real-Time De-babble:** This mode performs real time de-noising on the noisy signal received by the cell phone or an external microphone connected to the phone.
- **Wireless De-babble:** This mode performs real time de-noising on the noisy signal received from a remote cell-phone. For this mode we need two cell phones (one sender and one receiver). Currently the connection between the cell-phones is via Wi-Fi but we will introduce Bluetooth connection in the next version.
- **Analysis :** The app provides the user with real time basic information about the received audio signal including the frequency content, noise level and loudness.



References:

1969. IEEE Recommended Practice for Speech Quality Measurements. IEEE No 297-1969, 1-24.
Bishop C.M. 2007. Pattern Recognition and Machine Learning. Springer.
Duda R.O., Hart P.E., Stork D.G. 2001. Pattern classification. 2nd ed. Wiley, New York.
Gu Q., Li Z., Han J. 2012. Generalized Fisher Score for Feature Selection. arXiv preprint arXiv:1202.3725.
Krishnamurthy N., Hansen J.H.L. 2009. Babble Noise: Modeling, Analysis, and Applications. IEEE Transactions on Audio, Speech, and Language Processing 17, 1394-1407.
Selesnick I.W. 2011b. Wavelet Transform With Tunable Q-Factor. IEEE Transactions on Signal Processing 59, 3560-3575.
Shlens J. 2003. A TUTORIAL ON PRINCIPAL COMPONENT ANALYSIS.
Tang J., Aleyani S.,Liu H. 2014. Feature Selection for Classification: A Review, Data Classification: Algorithms and Applications. CRC Press.
Vincent E. 2005. MUSHRAM: A MATLAB interface for MUSHRA listening tests [Online] https://members.loria.fr/EVincent/software-and-data/ (verified August 2nd, 2016).
Yue H.H., Tomoyasu M. 2004. Weighted principal component analysis and its applications to improve FDC performance, Decision and Control, 2004. CDC. 43rd IEEE Conference on, Vol. 4. pp. 4262-4267 Vol.4.

Research funded by NIDCD grant R01-DC12152