LETTER Communicated by Shihab Shamma

A Novel Model-Based Hearing Compensation Design Using a Gradient-Free Optimization Method

Zhe Chen
zhechen@soma.ece.mcmaster.ca
Department of Electrical and Computer Engineering, McMaster University
Hamilton, Ontario L85 4k1, Canada

Suzanna Becker
becker@mcmaster.ca
Department of Psychology, McMaster University
Hamilton, Ontario L85 4k1, Canada

Jeff Bondy
jeff@soma.ece.mcmaster.ca
Department of Electrical and Computer Engineering, McMaster University
Hamilton, Ontario L85 4k1, Canada

Ian C. Bruce
ibruce@ieee.org
Department of Electrical and Computer Engineering, McMaster University
Hamilton, Ontario L85 4k1, Canada

Simon Haykin
haykin@mcmaster.ca
Department of Electrical and Computer Engineering, McMaster University
Hamilton, Ontario L85 4k1, Canada

We propose a novel model-based hearing compensation strategy and gradient-free optimization procedure for a learning-based hearing aid design. Motivated by physiological data and normal and impaired auditory nerve models, a hearing compensation strategy is cast as a neural coding problem, and a Neurocompensator is designed to compensate for the hearing loss and enhance the speech. With the goal of learning the Neurocompensator parameters, we use a gradient-free optimization procedure, an improved version of the ALOPEX that we have developed (Haykin, Chen, & Becker, 2004), to learn the unknown parameters of the Neurocompensator. We present our methodology, learning procedure, and experimental results in detail; discussion is also given regarding the unsupervised learning and optimization methods.

1 Introduction

Current fitting strategies for hearing aids set the amplification in each frequency channel based on the hearing-impaired person's audiogram, which measures pure tone thresholds for each of a small set of frequencies. However, it is well known that the detection of a sound can be strongly masked in the presence of background noise or competing speech, for example. It is therefore not surprising that many people with hearing loss end up not wearing their hearing aids. The devices are unhelpful and may even worsen the wearer's ability to hear sounds under noisy listening conditions. Directional microphones and other generic signal processing strategies for noise reduction have resulted in modest benefits in some contexts, but not dramatic improvement. Instead, the approach we take here is to treat hearing aid design as a neural coding problem. We start with detailed models of the normal auditory nerve as well as that of a hearing-impaired person. We then search for a signal transformation that, when applied to the input to the impaired model, will result in a neural code that is close to that of the intact model. We refer to this strategy as neural compensation (Becker & Bruce, 2002). The signal transformation is highly nonlinear and dynamic and calculates the gain in each frequency channel by combining information across multiple channels rather than using a static set of channel-specific gains. The Neurocompensator should therefore be capable of approximating the contrast enhancement function of the normal ear.

Neural compensation (Becker & Bruce, 2002) was motivated by the design of adaptive hearing aid devices for hearing-impaired persons. The goal of the Neurocompensator is to restore near-normal firing patterns in the auditory nerve in spite of the hair cell damage in the inner ear. A schematic diagram of normal and impaired hearing systems, as well as the neural compensation, is illustrated in Figure 1. Ideally, the Neurocompensator attempts to compensate the hearing impairment in the auditory system and match the output of the compensated system, as closely as possible, to the output of the normal hearing system. In other words, by regarding the outputs of the normal and impaired hearing systems as the neural codes generated by the brain, we attempt to maximize the similarity of the neural codes generated from models H and $\hat{H}$ in Figure 1.

The early development of the Neurocompensator was described in Bondy, Becker, Bruce, Trainor, and Haykin (2004). In this initial work, we compared the output of the normal and damaged models directly at the level of the raw spike trains. However, auditory nerves have high spontaneous firing rates, and when driven by auditory input, they convey predominantly steady-state information, whereas the transient information is most critical to speech perception. In our previous work, we tested the algorithm on vowel sounds, which are relatively steady state. Here, we apply a transient detection procedure to the auditory nerve spike trains to simulate higher levels of auditory processing, and we train and test the model on continuous

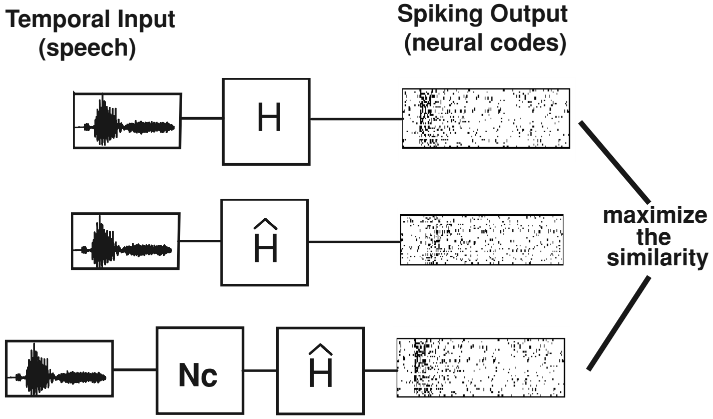

Figure 1: A schematic diagram of Neurocompensation. (Top) Normal hearing system. (Middle) Impaired hearing system. (Bottom) Neurocompensator (Nc) followed by the Impaired hearing system. The hearing systems map the temporal speech signal input to a spike trains map (neural codes) output; H and $\hat{H}$ denote the input-output mappings of the normal and impaired ear models, respectively. The Neurocompensator acts as a preprocessor before the impaired ear model in order to produce the similar neural codes as the normal neural codes from the normal ear model.

speech containing both voiced and unvoiced components. Also, in our previous work, an ad hoc perturbation-like optimization procedure was used to learn the Neurocompensator parameters with a simple error metric. Moreover, it does not provide a probabilistic measure of how well the Neurocompensator compensates for the hearing loss; neither does it present an informative comparative metric between the compensated and the normal hearing systems. It is our goal in this article to formulate a principled methodology and improve the optimization efficiency. In the work reported here, we incorporate four major advances in the development of the Neurocompensator algorithm. (1) We apply an onset-detection procedure to the auditory nerve model outputs (Bondy, Bruce, Dong, Becker, & Haykin, 2003) and adapt the model to continuous speech signals. (2) We develop a probabilistic metric to characterize the discrepancy between the onset spike train maps. (3) We incorporate an improved ALOPEX (ALgorithm Of Pattern EXtraction) procedure (Haykin, Chen, & Becker, 2004) for gradient-free optimization. (4) We present a major improvement in the design of the Neurocompensator that combines a fixed linear frequency-specific gain calculated by a standard widely used hearing aid algorithm (NAL-RP; Byrne, Parkinson, & Newall, 1990) with a context-dependent divisive normalization term whose coefficients are optimized using the ALOPEX.

The letter is organized as follows. In section 2, we present the model-based hearing compensation strategy and detail the probabilistic modeling of neural spike trains data. Section 3 describes the methodology for learning the Neurocompensator, including the architecture and the optimization procedure. We present some experimental results in section 4, followed by summary and discussion in sections 5.

2 Model-Based Hearing Compensation Strategy

2.1 An Overview of the System. Given the Neurocompensator diagram illustrated in Figure 1, the learning of the adaptive hearing system is shown in Figure 2. First, the time domain audio (speech or natural sound) signal is converted into frequency domain through short-time Fourier transform. The role of the Neurocompensator, which is modeled through frequency-dependent gain coefficients for different bands (described later in this section), is to conduct spectral enhancement in the frequency domain. Given the normal (H) and impaired ($\hat{H}$) auditory models, the feedback error is calculated via a probabilistic metric by comparing the spike train images between the normal and compensated hearing systems (detailed in section 3.1). Furthermore, a gradient-free optimization procedure (detailed in section 3.2) uses the error for updating the Neurocompensator's

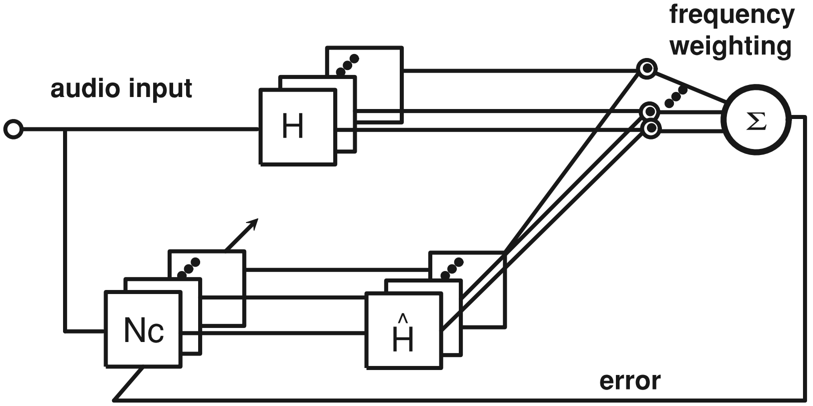

Figure 2: Block diagram of training the Neurocompensator (Nc). The normal (H) and impaired ($\hat{H}$) auditory models' output is a set of the spike trains at different best frequencies, which are then subjected to an onset-detection process (see text), while the Neurocompensator is represented as a preprocessor that calculates gains for each of the different frequencies. The error is actually the Kullback-Leibler (KL) divergence between the probability distributions of the two models' outputs (see text).

Table 1: Selected Speech Samples Used in the Experiments.

Speech Sample	Content
TIMIT-1	/The emperor had a mean temper./
TIMIT-2	/His scalp was blistered by today's hot sun./
TIMIT-3	/Would a tomboy often play outdoor?/
TIMIT-4	/Almost all of the colleges are now coeducational./
TIDIGITS-1	/one/
TIDIGITS-2	/one, two/
TIDIGITS-3	/nine, five, one/
TIDIGITS-4	/eight, one, o, nine, one/

parameters to minimize the discrepancy between the neural codes generated from the normal and impaired hearing models.

2.2 Experimental Data. The audio data presented to the ear models can be speech or any other natural sound. In our experiments, the speech data are selected from the TIMIT and the TIDIGITS databases. From the TIMIT database, 10 spoken sentences by different male and female speakers are used for the simulations reported here; the sample frequency of the speech data is 16 kHz. In the TIDIGITS database, the data consist of English spoken digits (in the form of isolated digits or multiple-digit sequences) recorded in a quiet environment, with sample frequency 8 kHz. All of the experimental data were subjected to resampling preprocessing (to 16 kHz if applicable) prior to being presented to the auditory models. Some of the speech samples used in the experiments are listed in Table 1. Ideally, all of the speech samples are truncated to within the same length.

2.3 Auditory Models. The auditory peripheral model used here is based on the earlier work of Bruce and colleagues (Bruce, Sachs, & Young, 2003). In particular, the model consists of a middle-ear filter, time-varying narrow- and wide-band filters, inner hair cell, outer hair cell, synapse model, and spike generator, describing the auditory periphery path from the middle ear to the auditory nerve. More recently, a new middle ear model and a new saturated exponential synapse gain control have been incorporated into that model.[1] The hearing-impaired version of the model described in detail in Bondy et al. (2004) simulates a typical steeply sloped high-frequency hearing loss.

With the normal or impaired auditory models (Bruce et al., 2003), the spike train maps can be generated via feeding the temporal audio (speech

[1] For further information on the auditory peripheral models, see Ian C. Bruce's web site: http://www.ece.mcmaster.ca/~ibruce/.

or natural sound) signal to the system.[2] We further process the auditory representation generated by the auditory nerve models by applying an onset detection procedure (Bondy et al., 2003), consisting of a derivative mask with rectification and thresholding (see the appendix). This removes much of the noisy spontaneous spiking and high degree of steady-state information in the signal-driven spike trains. The resultant spike trains onset map is used here as the basis for comparing the neural codes generated by the normal and impaired models.

2.4 Probabilistic Modeling. In order to compare the neural codes of the normal and impaired models, we characterized the spike trains onset time-frequency map, which contains a number of two-dimensional data points (represented as black dots in the output image), by its probability density function. To overcome the inherent noisiness of the spike-generating and onset-detection processes, we chose a two-dimensional mixture of gaussians to characterize this distribution, given its spatial smoothing property across the spectral-temporal plane. Suppose that $D_1 \equiv \{\mathbf{x}_i\}_{i=1}^{\ell}$ and $D_2 \equiv \{\mathbf{z}_i\}_{i=1}^{\ell'}$ denote the two-dimensional neural codes (i.e., the onset spike train binary images) that are calculated from the normal and impaired hearing models (Bruce et al., 2003), respectively.[3] Assume that $p(D_1|M)$ is a probabilistic model that characterizes the data D_1, when M here is represented by a gaussian mixture model—$M \equiv \{c_j, \boldsymbol{\mu}_j, \Sigma_j\}_{j=1}^{K}$.

Note that $\{\mathbf{x}_i\} \in D_1$ are the data points calculated from the normal ear model (with input-output mapping H) given the audio (speech) data. Suppose the data $\{\mathbf{x}_i\} \in \mathbb{R}^d$ are drawn from a two-dimensional ($d = 2$) mixture of gaussian density:

$$\begin{aligned} p(\mathbf{x}) &= \sum_{j=1}^{K} p(j)p(\mathbf{x}|j) \\ &= \sum_{j=1}^{K} c_j \frac{1}{\sqrt{(2\pi)^d |\Sigma_j|}} \exp\left(-\frac{1}{2} |\mathbf{x} - \boldsymbol{\mu}_j|^T \Sigma_j^{-1} |\mathbf{x} - \boldsymbol{\mu}_j| \right), \end{aligned} \tag{2.1}$$

where c_j is the prior probability for the jth gaussian component, with mean $\boldsymbol{\mu}_j$ and covariance matrix Σ_j. Given a total of ℓ data points in the time-frequency spike trains onset map, we can calculate the joint likelihood of the data given the mixture model M:

$$p(D_1|M) = \prod_{i=1}^{\ell} p(\mathbf{x}_i). \tag{2.2}$$

[2] The C++ codes written for generating the neural spike trains are available from Ian C. Bruce upon request.

[3] Note that in general, $\ell \neq \ell'$, where ℓ and ℓ' denote the total number of points in D_1 and D_2, respectively.

Alternatively, we can calculate the log likelihood

$$\mathcal{L} = \log p(D_1|M) = \sum_{i=1}^{\ell} \log p(\mathbf{x}_i), \tag{2.3}$$

and the associated average log likelihood $\mathcal{L}_{av} = \mathcal{L}/\ell$. In this article, we have not used any model selection procedure for gaussian mixture modeling. Nevertheless, it is straightforward to use the penalized maximum likelihood that incorporates a complexity metric, such as the Bayesian information criterion (BIC),[4] for model selection. More discussion of the model selection issue will be given later.

The clustering is fitted via a mixture of elliptical gaussians using the expectation-maximization (EM) algorithm (e.g., Duda, Hart, & Stork, 2001). It is known that the EM algorithm is guaranteed only to converge monotonically to a local minimum or saddle point. In our early investigations (Gupta, 2004), several empirical findings were observed. First, it is necessary to rescale the time and frequency ranges for better gaussian mixture fitting; an optimal scale ratio (time versus frequency) of 0.25 applied to the normalized time-frequency coordinate is suggested; namely, the time axis is constrained within the region [0,1], whereas the frequency axis is within the region [0,0.25]. This is tantamount to scaling the variance of the coordinates and compressing the data in terms of their distance, which is advantageous for probabilistic fitting. Second, for the spike trains onset map, a total of 20 to 30 mixtures of elliptical gaussians is sufficient to characterize the data distribution (see Figure 3), although the optimal number of mixtures varies from one data set to another. For simplicity, a fixed number of mixtures determined empirically is assumed throughout our experiments, though this is not a principled solution. In addition, gaussian mixture fitting via the EM algorithm is well known to be sensitive to the initialized (mean and covariance) parameters (see Figure 4 for an illustration) for both the convergence speed and log likelihood performance. With a better initialization scheme compared to Gupta (2004), we use the K-means clustering method (e.g., Duda et al., 2001) to initialize the mean parameters to accelerate the convergence. We found that 10 to 20 iterations of the batch EM algorithm produce reasonable-fitting results for all data used thus far.

2.5 Spectral Enhancement. Spectral enhancement is achieved through the Neurocompensator. The principle of the Neurocompensator is to control the spectral contrast via the gain coefficients using the idea of divisive

[4] For a K-mixture of gaussians model, the BIC is defined as $BIC(K) = \sum_{i=1}^{\ell} \log p(\mathbf{x}_i|\boldsymbol{\theta}) - \frac{\ell_K}{2} \log \ell$, where $\ell_K = K\big(1 + d + \frac{d(d+1)}{2}\big)$ represents the total number of free parameters in the model.

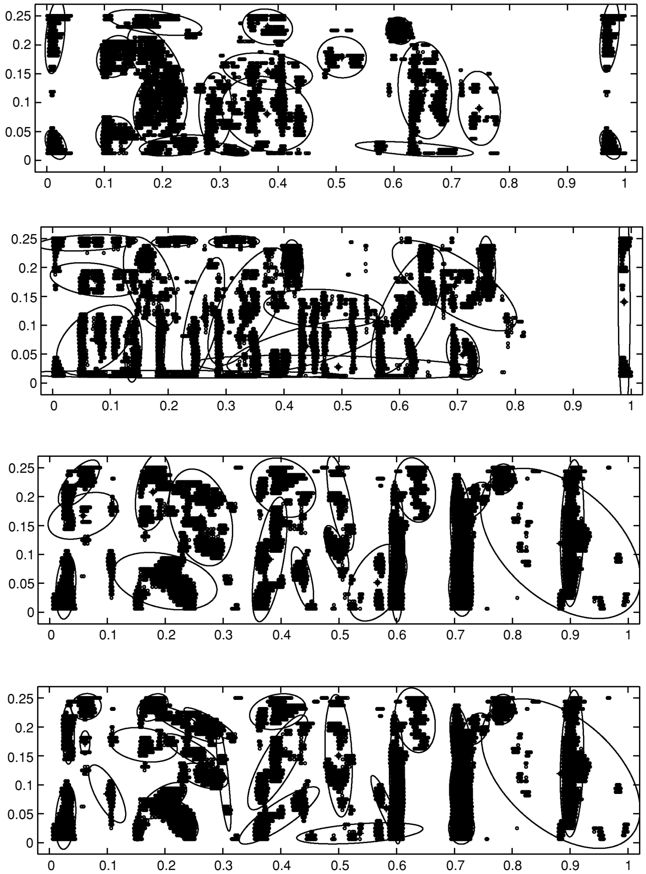

Figure 3: Three selected sets of spike trains data calculated from the normal hearing model and their probabilistic fittings using 20 (the first three plots) or 30 (the fourth plot) gaussian mixtures. In these four plots, the horizontal axis represents scaled time; and the vertical axis represents scaled frequency, with a frequency versus timescale ratio of 0.25. For the third plot, $\mathcal{L} = 22009$, $\mathcal{L}_{av} = 1.97$, and $BIC(20) = 20891$; for the fourth plot, $\mathcal{L} = 23942$, $\mathcal{L}_{av} = 2.14$, and $BIC(30) = 22264$. It is evident that the fourth plot is a better fit than the third one.

normalization (Schwartz & Simoncelli, 2001). In particular, the frequency-dependent gain coefficient, G_i, at the ith frequency band, is calculated as

$$G_i = \frac{\| f_i \|^2}{\sum_j v_{ji} \| f_j \|^2 + \sigma}, \tag{2.4}$$

where i and j represent the indices of the frequency bands; v_{ji} denotes the cross-frequency-effect coefficient; G_i is a nonlinear function of the weighted

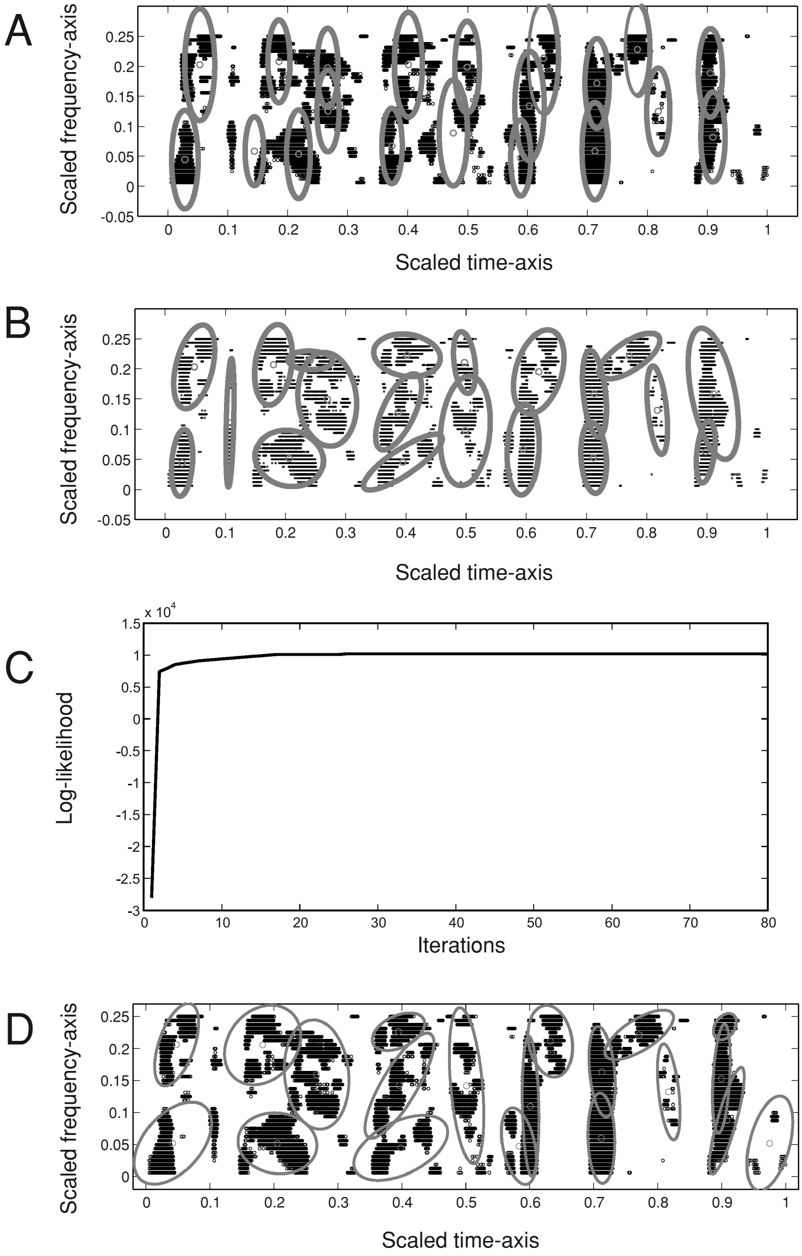

Figure 4: (A) The initialized 20 gaussian mixtures via *K*-means clustering. (B) The gaussian mixture fitting after 80 iterations of the EM algorithm. (C) The log-likelihood convergence curve. (D) Another fitting result obtained from a different initial condition.

input (frequency) power; $\| f_i \|^2$, divided by the weighted sum of all the frequencies' power; and σ is a regularization constant that ensures that the gain coefficient G_i does not go to infinity. The design of gain coefficient function is the essence of a Neurocompensator. Applying gain coefficients to frequency bands is tantamount to implementing a bank of nonlinear filters, the motivation of which is to mimic the inner hair cells' frequency response. The divisive normalization was originally aimed at suppressing the statistical dependency between the filters' responses (Schwartz & Simoncelli, 2001). Here, we employ a similar functional form, but rather than adapting the normalization coefficients to optimize information transmission, we adapt the parameters to optimize a measure of the similarity between the neural codes generated by the two models (see section 3).

For the present purpose, we propose a slightly different version of equation 2.4, as follows:

$$G_i = h\left(\frac{w_i \| f_i \|^2}{\sum_j v_{ji} \| f_j \|^2 + \sigma} \right), \quad \text{where} \quad w_i \propto G_i^{NAL-RP}, \tag{2.5}$$

where G_i^{NAL-RP} represents a positive coefficient based on NAL-RP (National Acoustics Lab–Revised Profound), a standard hearing aid fitting protocol (Byrne et al., 1990) that can be calculated from the ith frequency band (see Bondy et al., 2004); and $h(\cdot)$ is a continuous, smooth (e.g., sigmoid) function that constrains the range of the gains as well as ensures that the gains will vary smoothly in time. When $h(\cdot)$ is linear and $G_i^{NAL-RP} = 1$, equation 2.5 reduces to 2.4. When all $v_{ji} = 0$ and $h(\cdot)$ is linear, equation 2.5 reduces to the standard, fixed linear gain NAL-RP algorithm. We have chosen w_i to be proportional (in value) to the G_i^{NAL-RP} that is given by the standard NAL-RP algorithm for calculation of the gains, while ensuring that w_i will not be so large or small as to push the sigmoid function into the saturated region where derivatives would be near zero; w_i will be fixed after appropriate scaling. For the hearing aid application, it is appropriate to constrain $G_i \geq 0$.[5] Now, the goal of the learning procedure is to find the optimal parameters $\{v_{ji}\}$ that compensate the hearing impairment or intelligibility according to a certain performance metric. Because these normalization parameters are adapted to compensate for impaired auditory peripheral processing, we expect them to mimic the true neurobiological filter that they are substituting for. For example, for a fixed-frequency channel j, v_{ji} might evolve toward an "on-center off-surround"–shape filter. Since the Neurocompensator attempts to substitute the role of a real neurobiological filter, it is reasonable to impose biologically realistic constraints on the compensator parameters: the gain coefficients G_i should be nonnegative, bounded, and varying smoothly over a short period of time. It is important to note that unlike the traditional hearing aid algorithms, the parameters to be optimized are not independent,

[5] The case $G_i < 0$ has an effect of phase reversal to the frequency domain.

in the sense that the cross-frequency interference may cause modifying one parameter to indirectly affect the optimality of the others. All of these issues make the learning of the Neurocompensator a hard optimization problem, and the solution might not be unique. In our early investigations (Bondy et al., 2004), the optimization procedure and the error metric used were quite ad hoc, and certain instability during the optimization was also observed. One of our major goals here is to recast this optimization problem in a more principled way.

3 Training the Neurocompensator

3.1 Optimization Problem Formulation. Let $\boldsymbol{\theta} \equiv \{v_{ji}\}$ denote the vector that contains all of the parameters to be estimated in the Neurocompensator. Let $D_2 = \{\mathbf{z}_i\}$ denote the data calculated from the deficient ear model (with input-output mapping $\hat{\mathrm{H}}$), after preprocessing the audio (speech) with the Neurocompensator parameterized by $\boldsymbol{\theta}$. Let $p(D_2|M, \boldsymbol{\theta})$ be the marginal likelihood of the impaired model's spike trains having been generated by a normal model; then the associated log likelihood can be written as

$$\mathcal{L}'_{av} = \frac{1}{\ell'} \log p(D_2|M, \boldsymbol{\theta}) = \frac{1}{\ell'} \log \left(\prod_{i=1}^{\ell'} \sum_{k=1}^{K} c_k \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k; \mathbf{z}_i) \right)$$

$$= \frac{1}{\ell'} \sum_{i=1}^{\ell'} \log \left(\sum_{k=1}^{K} c_k \mathcal{N}(\boldsymbol{\mu}_k, \Sigma_k; \mathbf{z}_i) \right),$$

where M is a gaussian mixture model fitted to the normal hearing model's output, D_1, by maximizing $\log p(D_1|M)$, which can be optimized off-line as a preprocessing step. One way of optimizing the Neurocompensator would be to maximize $\mathcal{L}'_{av}$ with respect to $\boldsymbol{\theta}$; however, directly maximizing it may cause a "saturation," since the number of points in D_2, ℓ', might grow over ℓ.[6] A better objective function that does not suffer this pitfall is the Kullback-Leibler (KL) divergence between the probability of observing the impaired model's output under the normal versus impaired density function. Unfortunately, calculating the latter is much more costly because it must be done repeatedly, interleaved with optimization of the Neurocompensator parameters $\boldsymbol{\theta}$. We therefore consider a discrete sampling approach to estimate this density, which is computationally simpler than fitting a gaussian mixture model.

Specifically, we quantize or discretize evenly the spike trains onset map into a number of bins, where each bin contains zero or more of the spikes.

[6] This has been confirmed in our experiments. The worst case of the saturation effect will be that $\{\mathbf{z}_i\}$ are uniformly distributed across the whole spike trains map.

To quantitatively measure the discrepancy between the normal spike trains and reconstructed spike trains maps, we calculate the probability of each bin that covers the spikes; this can be easily done by counting the number of the spikes in the bin and further normalizing by the total number of the spikes in the whole spike trains map. In particular, the objective function to be minimized is a quantized form of the KL divergence,

$$E \equiv \mathrm{KL}(D_2 \| D_1) = \sum_{i}^{\#bins} p(bin_i | D_2) \log \frac{p(bin_i | D_2)}{p(bin_i | D_1)}, \tag{3.1}$$

where $p(bin_i|D_1)$ and $p(bin_i|D_2)$ represent the probabilities of the ith bin that contains the spikes in the normal and reconstructed spike trains maps, respectively. Note that $p(bin_i|D_1)$ can be calculated (only once) in the preprocessing step. In our experiment, we quantize evenly the spike trains map into a (40-time)×(10-frequency) mesh grid (see Figure 5A), with a total of 400 bins.

However, equation 3.1 suffers from two drawbacks. (1) For some bins, the denominator $p(bin_i|D_1)$ can be zero, thereby causing a numerical problem, and (2), there is no smoothing between two discrete maps, hence it will suffer from the noise in the spiking or onset detection processes. Fortunately, since we have the gaussian mixture probabilistic fitting for D_1 at hand, this can provide a spatial smoothing across the neighboring (time and frequency) bins, thereby counteracting the noise effect. To overcome the above two problems, we therefore substitute $p(bin_i|D_1)$ (quantized version) with $p(bin_i|M)$ (continuous version), where $p(bin_i|M)$ is calculated by fitting the center point in the ith bin with the gaussian mixture model M, divided by a normalization factor: $\sum_j p(bin_j|M)$ (see Figure 5B).[7] To do so, we modify 3.1 to obtain our final objective function:

$$E \equiv \mathrm{KL}(D_2 \| M) = \sum_{i}^{\#bins} p(bin_i | D_2) \log \frac{p(bin_i | D_2)}{p(bin_i | M)}. \tag{3.2}$$

Note that $p(bin_i|M)$ is usually a nonzero value due to the overlapping gaussian covering, although it can be very small.[8] As before, $p(bin_i|M)$ can be calculated in the preprocessing step. When $p(bin_i|D_2) = p(bin_i|M)$, it follows that $E = 0$; otherwise, E is a nonnegative value given $0 \leq p(bin_i|D_2) < 1, 0 \leq p(bin_i|M) < 1$. Since the probability $p(bin_i|D_2)$ can be zero, we have assumed that $0 \log 0 = 0$.

[7] To see how close the approximation is, we calculate the KL divergence in the example of Figure 5: $\sum_{i=1}^{400} p(bin_i|D_1) \log \frac{p(bin_i|D_1)}{p(bin_i|M)} = 0.1888$.

[8] To avoid the numerical problem in practice, we add a very small value (10^{-16}) to the denominator to prevent overflowing.

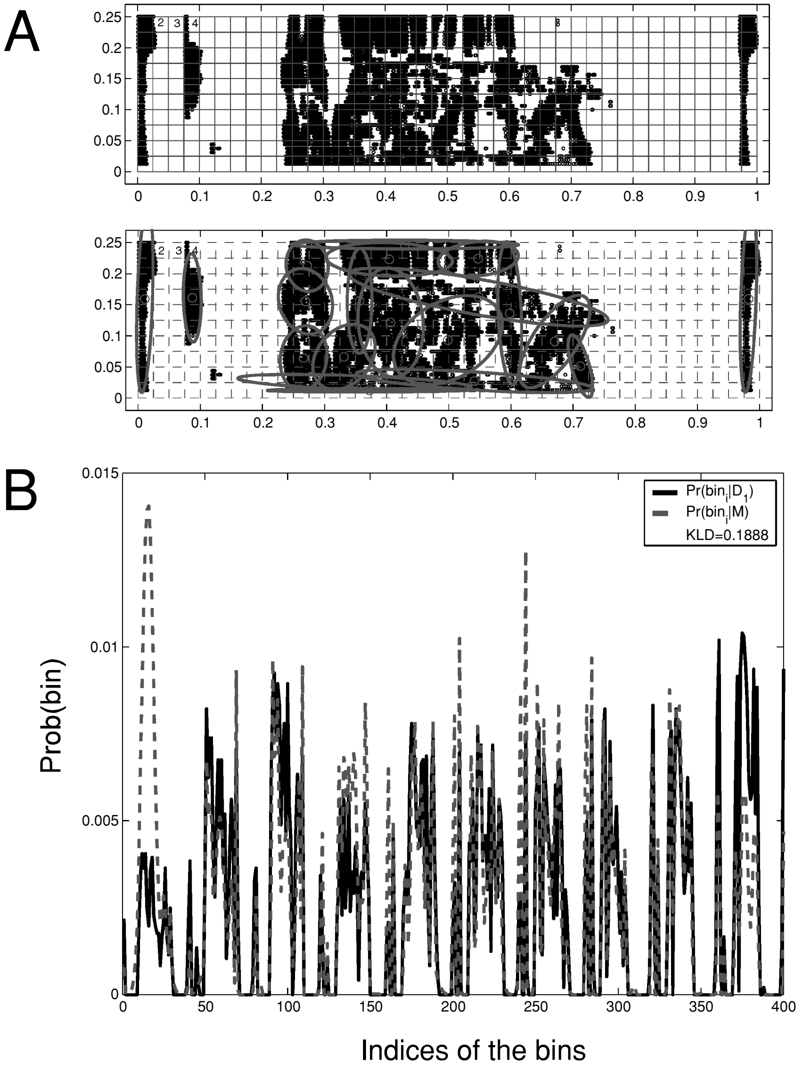

Figure 5: (A) A grid quantization compared with a gaussian mixture fitting (middle panel) on the spike trains map. Each map contains 40 × 10 = 400 bins; the arabic numerals inside the bins indicate their respective indices. (B) The approximation comparison between $p_1 = p(bin_i|D_1)$ and $p_2 = p(bin_i|M)$ ($i = 1, \ldots, 400$), $\mathrm{KL}(p_1 \| p_2) = 0.1888$.

It is noted that direct calculation of the gradient $\frac{\partial E}{\partial \boldsymbol{\theta}}$ in either equation 3.1 or 3.2 is inaccessible due to the characteristics of the ear model as well as the form of the objective function; hence, we can only resort to gradient-free optimization, which we discuss below. During the training phase, the gain coefficients are adapted to minimize the discrepancy between the "Neuro-compensated" and the original spike trains (see Figure 2).

3.2 Gradient-Free Optimization: ALOPEX. The ALOPEX algorithm was originally developed in vision research for optimizing the neurons' response in terms of number of spikes (Harth & Tzanakou, 1974; Tzanakou, Michalak, & Harth, 1979; Harth, Unnikrishnan, & Pandya, 1987). Since then, many versions of the ALOPEX have been developed (Unnikrishnan & Venugopal, 1994; Tzanakou, 2000; Bia, 2001; Sastry, Magesh, & Unnikrishnan, 2002; Chen, Haykin, & Becker, 2003). As a generic optimization framework, the ALOPEX-type algorithms have certain appealing advantages: gradient free, network architecture independent, and synchronous learning. These appealing features make the ALOPEX a useful tool for nonconvex optimization and many machine learning problems.

In Chen et al. (2003) and Haykin et al. (2004), we proposed a modified version of the ALOPEX algorithm, which aims to maintain the improved convergence speed of the ALOPEX-B algorithm (Bia, 2001) over the original ALOPEX, while improving the susceptibility of ALOPEX-B to local minima that we found in our earlier investigations. Specifically, let $\boldsymbol{\theta}$ denote a vector of some unknown parameters, and assume that the objective function, $E \equiv E(\boldsymbol{\theta})$, is to be minimized; our proposed algorithm proceeds as follows:

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta \Delta\boldsymbol{\theta}(t) \Delta E(t) + \gamma \boldsymbol{\xi}(t), \tag{3.3}$$

where η and γ are the step-size parameters, and $\Delta\boldsymbol{\theta}(t) = \boldsymbol{\theta}(t) - \boldsymbol{\theta}(t-1)$, $\Delta E(t) = E(t) - E(t-1)$. The vector $\boldsymbol{\xi}(t)$ is a random vector with its jth entry determined element-wise by

$$\xi_j(t) = \operatorname{sgn}(u_j - p_j(t)), \quad u_j \sim \mathcal{U}(0,1), \tag{3.4}$$

$$p_j(t) = \phi(C_j(t)), \tag{3.5}$$

$$C_j(t) = \frac{\operatorname{sgn}(\Delta\theta_j(t))\Delta E(t)}{\sum_{k=2}^{t} \lambda(\lambda-1)^{t-k} |\Delta E(k-1)|}, \tag{3.6}$$

where u is a uniformly distributed random variable drawn from the region $(0,1)$, $\operatorname{sgn}(\cdot)$ is the signum function, and $\phi(\cdot)$ is the logistic sigmoid function. The scalar $0 < \lambda < 1$ is a forgetting parameter. An optimal forgetting parameter is often problem dependent; a common value is often chosen within the range $[0.35, 0.7]$. The parameter setup used in our experiments is $\eta = 0.01$, $\gamma = 0.05$, and $\lambda = 0.5$.

The algorithm starts with a randomly initialized parameter $\boldsymbol{\theta}(0)$ and stops when the cost function $E(t)$ is sufficiently small or a predefined maximal step is reached. The stochastic component $\boldsymbol{\xi}(t)$, being a random force with certain acceptance probability, is included to help (but with no guarantee) the algorithm escape from local minima.

It is noteworthy to make several remarks here regarding the optimization algorithm:

- Being a stochastic correlative learning algorithm, the modified ALOPEX-B algorithm incorporates two types of correlation. The first kind of correlation takes a form of instantaneous cross-correlation described by the product term $\Delta\boldsymbol{\theta}(t)\Delta E(t)$. The second kind of correlation appears in the computation of $\boldsymbol{\xi}(t)$ as in equations 3.4 through 3.6, which determines the acceptance probability of random perturbation force $\boldsymbol{\xi}(t)$. (See Haykin et al., 2004, for further discussion.)
- It is straightforward to apply a more sophisticated version of the ALOPEX algorithm for optimization, for instance, the Monte Carlo sampling-based ALOPEX algorithms developed in Chen et al. (2003) and Haykin et al. (2004). Nevertheless, we caution that the error metric should be carefully bounded and scaled for calculating the posterior probability $p(\boldsymbol{\theta}) \propto \exp(-E(\boldsymbol{\theta}))$.

The entire learning procedure is summarized as follows:

1. Initialize the parameters: $\{v_{ji}\} \in \mathcal{U}(-0.5, 0.5)$, $\sigma = 0.001$. Randomly select one speech sample.
2. Load the selected speech data, the associated spike trains fitting mixture parameters $M \equiv \{c_i, \boldsymbol{\mu}_i, \Sigma_i\}$, and the probability $p(bin_i|M)$, the latter two of which are precalculated off-line.
3. Apply the short-term Fourier transform (STFT) to the speech data (128-point FFT with a 64-point overlapping Hamming window).[9] The results of time-frequency analysis then provide the temporal-spectral information across frequency bands.[10]
4. Apply the gain coefficients $\boldsymbol{\theta}$ to the frequency bands according to equation 2.5. Perform inverse Fourier transform to reconstruct the time domain waveform.
5. Present the reconstructed waveform to the hearing-impaired ear model; produce a "Neurocompensated" spike trains map.

[9] For 16 kHz sampling frequency, it corresponds to a duration of 8 ms.

[10] Depending on the frequency resolution requirement, the number of frequency bands can vary from 20 to 40. We use 20 frequency bands in the experiments.

6. Using the quantized approximation to the hearing-impaired data probability density and the precalculated gaussian mixture model. Calculate the objective function 3.2.
7. Apply the improved ALOPEX algorithm (see equations 3.3 through 3.6) to optimize $\boldsymbol{\theta}$.
8. Repeat steps 3 through 7 for a fixed number (say 100 to 200) of iterations.
9. Select another speech sample, and repeat steps 2 through 8. Repeat the whole procedure until the convergence criterion is satisfied.

As far as step 7 in the optimization procedure is concerned, two kinds of optimization schemes can be considered:

- Synchronous optimization. All of the gain coefficients are treated with no difference; all of the parameters are updated in parallel across different frequency bands. This scheme is simple, but due to the cross-frequency interdependence of the coefficients, it can be very slow given a poor parameter initialization.
- Asynchronous optimization. The gain coefficients in different frequency bands are treated differently and optimized sequentially with different priority. Starting with the highest-frequency band, all the other parameters associated with the lower-frequency bands are set as zeros; update only the parameters associated with the high-frequency band. Then freeze these parameters, switch to a lower-frequency band (i.e., the second highest) repeat the optimization, and so on. For each frequency band, the optimization stopping criterion is empirically set as repeating 10 to 15 iterations. This sequential optimization can be justified by the fact that in a hearing-impaired system, it is the lower frequencies that tend to interfere with the detection of higher frequencies, not the converse.

4 Experimental Results

To reduce the computational burden, we have consistently used a fixed number ($K = 20$) of gaussian mixtures for fitting all of spike trains data. We present results here based on the training speech samples listed in Table 1, totaling about 14.1 seconds of continuous speech.

We apply the improved version of the ALOPEX-B algorithm for optimization, where the objective function to be minimized is equation 3.2. Figure 6 shows the performance metric curve using the synchronous optimization scheme. We have not extensively investigated the asynchronous optimization scheme, but it was observed in an empirical test that inappropriate initialization may cause unstable performance. For this reason, we have restricted ourselves here to the synchronous optimization scheme.

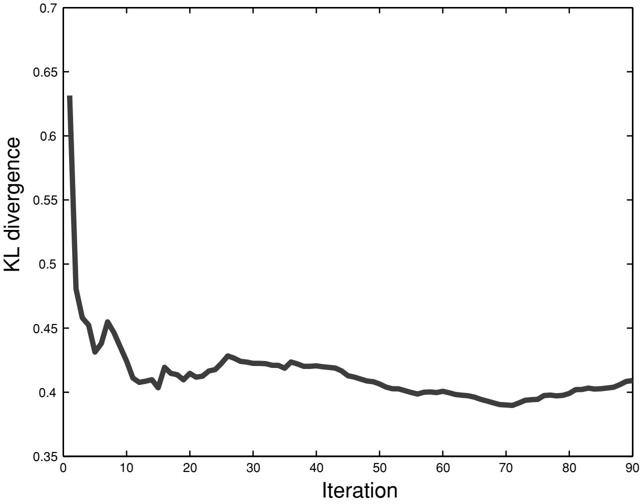

Figure 6: Learning curve of one speech sample using synchronous optimization. The KL divergence starts with 0.63 and stays around 0.4 after 90 iterations.

Note that finding the optimal $\boldsymbol{\theta}$ from normal spike trains is an ill-posed inverse problem; hence, it is impossible to build a perfect inverse model. However, it is hoped that the reconstructed spike trains image from the compensated hearing-impaired model is close to the one from the normal hearing model after the learning the Neurocompensator. Figure 7 shows the comparison between the normal, deficient, and Neurocompensated spike trains maps of the training speech sample.

Upon completion of the training process, we freeze $\boldsymbol{\theta}$ and further test the Neurocompensator on some unseen speech samples. The training and testing KL divergence results of the experimental data are summarized in Table 2.

Two sets of testing results on two spoken speech signals are shown in Figure 8. It is seen that the Neurocompensated spike trains maps are reasonably close to the normal ones, though not perfect. This is quite encouraging given the fact that we have used only about 3.7 seconds of speech for training here. Ideally, given sufficient computational power, we should use as many speech samples as possible for training. It is hoped that by averaging across more speech samples (with different contexts, speakers, spoken speeds, and so forth), the learning process can yield a more accurate and robust solution.

5 Summary and Discussion

We have described a novel methodology for learning a Neurocompensator, an ingredient of a learning-based, intelligent hearing aid device. The

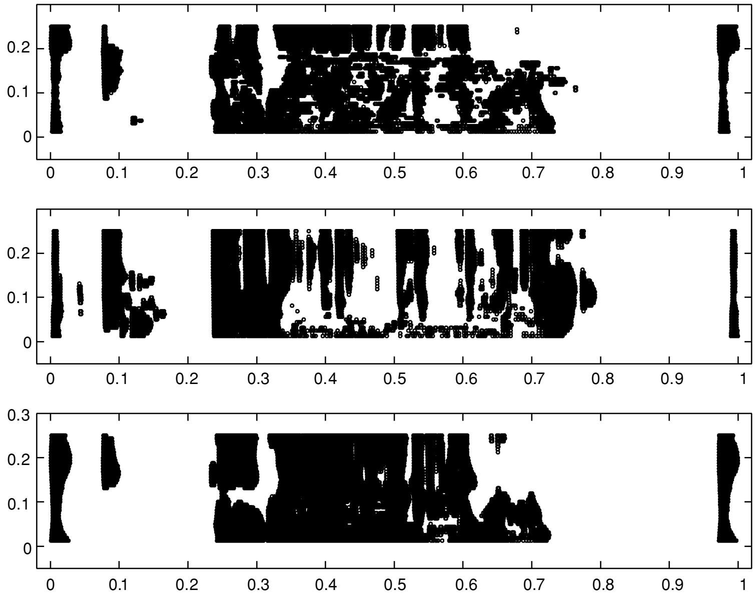

Figure 7: Comparisons of normal, deficient, and Neurocompensated (respectively, from top to bottom panels) spike trains onset maps. The deficient spike trains map is generated using the hearing-impaired model applied to the deficient waveform (which is produced by preprocessing the signal through the standard NAL-RP algorithm, with all gains set to $G_i \equiv G_i^{NAL-RP}$ for the 20 time-frequency bands and then reconstructing the signal by inverse FFT). The KL divergence between the deficient and normal spike trains is 0.664 before the learning, as opposed to 0.42 between the Neurocompensated and normal spike trains after the learning.

learning is achieved by probabilistic modeling of auditory nerve model spike trains and a gradient-free optimization procedure for parameter update. Based on our empirical experiments, it has been shown that the Neurocompensator provides a promising approach to adaptive compensation for reducing perceptual distortion due to hearing loss.

We have observed some problems with our current approach. In particular, we have found in the experiments that the optimization solution is nonunique. As seen from Figure 7, there are still obvious differences between the normal and Neurocompensated spike trains maps. We suspect that constraining the solution space and incorporating prior knowledge might somewhat alleviate this issue. Second, we found that the parameters are somewhat training data dependent. In other words, one set of Neurocompensator parameters good for one speech sample does not necessarily

Table 2: Training and Testing Results of the Experimental Data in Table 1.

Speech Sample	$\text{KL}_{init}(D_2 \| M)$	$\text{KL}_{final}(D_2 \| M)$	$\text{KL}_{final}(D_2 \| D_1)$	$\text{KL}(D_1 \| M)$
TIMIT-1	1.2058	**0.4462**	1.2828	0.1885
TIMIT-2	0.6152	0.4697	1.9255	0.2493
TIMIT-3	0.6692	0.6105	1.7367	0.2741
TIMIT-4	0.6477	0.4666	1.8329	0.2743
TIDIGITS-1	1.0626	**0.1798**	0.5591	0.0547
TIDIGITS-2	1.0234	0.4345	1.5918	0.1634
TIDIGITS-3	0.4913	0.2013	0.5759	0.0871
TIDIGITS-4	0.6346	**0.2599**	0.3757	0.1888

Notes: The right-most column $\text{KL}(D_1 \| M)$ indicates the approximation accuracy between the quantized pmf and continuous gaussian mixture pdf on the neural codes obtained from the normal hearing system. It can be roughly viewed as a lower bound for the values in the third and fourth columns, which are the final values of $\text{KL}(D_2 \| M)$ and $\text{KL}(D_2 \| D_1)$ for the training or testing data after the learning is terminated. The second and third columns show the values of $\text{KL}(D_2 \| M)$ (objective function 3.2) before and after employing the Neurocompensator. The numbers in boldface indicate the training results.

produce a similarly good performance for another one (see Table 2). This problem should be somewhat alleviated by averaging across more training samples. Another solution to this problem may be to train a mixture of Neurocompensator modules adapted to different input statistics, such as different talkers under varying listening conditions. One could then use a trained classifier to select the best Neurocompensator for the current context.

One obvious weakness here is to use a fixed number of mixtures for different spike trains image data. In order to alleviate the computational burden of our procedure and focus on the optimization part, we have neglected to consider model selection in our probabilistic modeling. In the literature, however, there are some principled ways, such as Bayesian approaches (Roberts, Husmeier, Rezek, & Penny, 1998; Attias, 2000), the merging-splitting approach (Ueda, Nakano, Ghahramani, & Hinton, 2000), or the greedy approach (Verbeek, Vlassis, & Kröse, 2003), to tackle this issue.

Another important area for future investigation is the design of the gain function 2.5. We have found that the form of the gain function (e.g., the range and the shape of $h(\cdot)$ function) has a crucial effect on the optimization performance, particularly on the speed of convergence. The possibility of incorporating prior knowledge or adding constraints to the gain function might also accelerate the convergence speed of optimization. How to design an optimal form of the gain function remains an unsolved problem.

In the simulations reported here, we have used a generic hearing-impaired model with a classic "ski-slope" loss profile, with a sharp linear falloff in perceptibility at high frequencies (Bondy et al., 2003). However, using the same auditory nerve model, it is possible to create an extremely

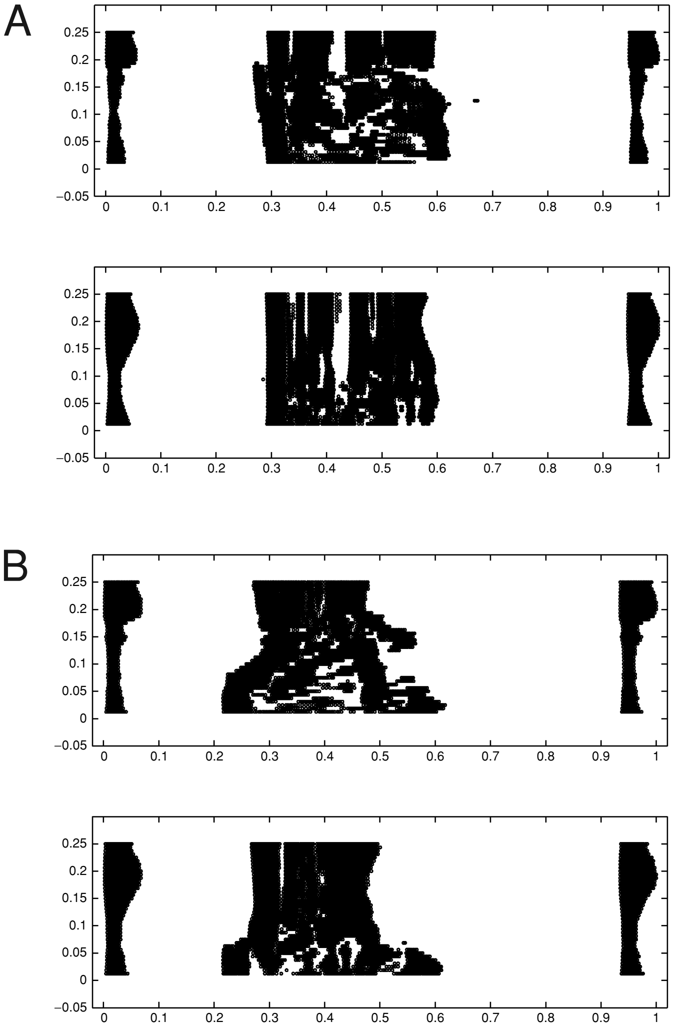

Figure 8: Testing results on two untrained continuous speech samples. Comparison is made between the normal and Neurocompensated spike trains onset maps. The KL divergence of equation 3.1 is 0.2013 between the top two maps (A) and 0.5591 between the bottom two maps (B).

detailed and accurate model of an individual's hearing loss profile, and then learn appropriate compensation parameters; In other words, the Neurocompensator can be designed to be person specific. This requires separate estimation of the impairment of inner and outer hair cells at a wide range of frequencies. Although such measurements go well beyond the standard audiogram, psychophysical tests have been developed for this purpose (Shera, Guinan, & Oxenham, 2002; Plack & Oxenham, 2000; Moore, Huss, Vickers, Glasberg, & Alcantara, 2000; Moore, Vickers, Plack, & Oxenham, 1999). It is particularly important to map out "holes in hearing"—any dead region of the cochlea where inner hair cells, the primary auditory sensory receptors, have died off. Although nearby hair cells will fire in response to the frequencies normally transmitted by the dead region, a simple amplification of frequencies in a dead region, as would be done by standard hearing aids, may result in severe perceptual distortions. Unlike other hearing aid strategies, the Neurocompensator should be able to correct for such distortions. However, one limitation of this approach is that it neglects the normal listener's ability to perform auditory sound localization and stream segregation, and the use of top-down expectations to focus attention. Future development of this work could incorporate more sophisticated auditory models to train the Neurocompensator.

After further development of our algorithm, the ultimate test of its efficacy will be to conduct human hearing tests. The hearing-impaired person(s) will listen to the reconstructed speech waveform yielded from the hearing aid device (i.e. Neurocompensator) and compare the intelligibility quality with and without the hearing compensation. Once the training is accomplished, the hearing test requires no additional computational effort and is easily performed. Furthermore, once the Neurocompensator parameters are optimized, the algorithm represented by equation 2.5 could be straightforwardly and efficiently implemented in a digital hearing aid circuit.

Appendix: Onset Spike Trains Map Generation

The onset of energetic amplitude modulation (AM) components of the stimuli coded in the spike trains map is used in our experiments for perceptual grouping. In what follows, we briefly describe the motivation, representation of the spike trains map, and the onset map generation procedure (Bondy et al., 2003).

The goal of transforming the spike trains into the AM onset map is to provide a more parsimonious representation of the important acoustic events. Auditory research has showed that the AM feature extraction plays a critical role, being biologically viable and psychophysically justified. The slow AM fluctuations that are highlighted by our transformation mapping are based on the important AM found in speech (Drullman, Festen, & Plomp, 1994). In addition, the study in auditory periphery (Wang & Shamma, 1995) demonstrated that the spectral-temporal response fields (STRFs) at many points in the auditory brain show strong AM responses; Fishbach, Yeshurun, and Nelken (2003) also proposed their auditory function model based on the AM feature extractors.

Here we use a single AM extractor per frequency channel that passes the psychophysically important modulations. The instantaneous neural spike trains are computed for a set of 20 logarithmically spaced central frequencies (CFs) using the auditory model developed in Bruce et al. (2003). The representation is then a finite resolution of time-frequency map (with horizontal axis representing time and vertical axis representing frequency).

Onset of AM in each frequency band is calculated with a difference of exponential filters, $h_1[n]$, in each frequency band:

$$h_1[n] = \frac{n}{\alpha_1^2} \exp(-n/\alpha_1) - \frac{n}{\alpha_2^2} \exp(-n/\alpha_2).$$

The input to h_1 is the instantaneous discharge rate over time for each channel. The values α_1 and α_2 are selected to pass the psychophysically important frequencies from 4 to 32 Hz. These frequencies contribute most to intelligibility, with a signal's fine temporal structure adding only a small amount to the intelligibility. This is a little wider than the data in Drullman et al. (1994) because of the difficulty in making a very sharp filter.

The onset data are then integrated over a typical acoustic event time window, $h_2[n]$, which has a 6dB cutoff at 125 Hz. This integrator is defined as

$$h_2[n] = \frac{n}{\alpha_3^2} \exp(-n/\alpha_1).$$

For a sample rate of 11,025 Hz, the parameters are chosen to be $\alpha_1 = 0.06, \alpha_2 = 0.10$ and $\alpha_3 = 0.001$. The values of α_1 and α_2 turned out to be very similar to those chosen in the feedback architecture in Nelson & Carney (2004) that explored the linear AM response. Thus, applying h_1 corresponds to employing an AM extractor for each frequency channel, which can be thought of as the basic excitatory and inhibitory interplay between the auditory neurons. The data from the AM feature detector are then integrated with h_2 over the typical syllabic rate.

An adaptive threshold and refraction operation is then applied, which mimics the neural firing patterns to produce AM "events" over a certain length. The thresholding was selected to produce a suitable sparsity for grouping. For instance, the threshold value is selected to produce some percentage (0.1 to 0.5%) of active events in the discretized time-frequency spike trains map when the refractory period is set as 1 ms. The greater the threshold value, the sparser are the spikes in the onset map; on the other hand, increasing the refractory period would thin out the continuous blocks in the onset map. In our experiments, active event probabilities from 0.01 to 5 percent were tried before settling on 0.2 percent for the Neurocompensator simulations. Typical threshold value is within the region $[1, 1.7]$.

Acknowledgments

The work reported here was supported by the Natural Sciences and Engineering Research Council of Canada and is associated with the Blind Source Separation and Application project. We thank S. Gupta for the early input and the assistance of implementation. We also thank an anonymous reviewer for valuable comments that improved the presentation of the article.

References

Attias, H. (2000). A variational Bayesian framework for graphical models. In S. A. Solla, T. K. Leen, & K. Müller (Eds.), *Advances in neural information processing systems, 12* (pp. 201–215). Cambridge, MA: MIT Press.

Becker, S., & Bruce, I. C. (2002). Neural coding in the auditory periphery: Insights from physiology and modeling lead to a novel hearing compensation algorithm. In *Workshop in Neural Information Coding*, Les Houches, France.

Bia, A. (2001). Alopex-B: A new, simple, but yet faster version of the Alopex training algorithm. *International Journal of Neural Systems, 11*(6), 497–507.

Bondy, J., Becker, S., Bruce, I., Trainor, L., & Haykin, S. (2004). A novel signal-processing strategy for hearing-aid design: Neurocompensation. *Signal Processing, 84*, 1239–1253.

Bondy, J., Bruce I., Dong, R., Becker, S., & Haykin, S. (2003). Modeling intelligibility of hearing-aid compression circuits. In *Proc. 37th Asilomar Conf. Signals, Systems, and Computers* (pp. 720–724).

Bruce, I. C., Sachs, M. B., & Young, E. (2003). An auditory-periphery model of the effects of acoustic trauma on auditory nerve responses. *Journal of the Acoustical Society of America, 113*, 369–388.

Byrne, W., Parkinson, A., & Newall, P. (1990). Hearing aid gain and frequency response requirements for the severely/profoundly hearing impaired. *Ear and Hearing, 11*, 40–49.

Chen, Z., Haykin, S., & Becker, S. (2003). *Sampling-based ALOPEX algorithms for neural networks and optimization* (Tech. Rep.). Hamilton, Ontario: Adaptive Systems Lab, McMaster University. Available online at http://soma.crl.mcmaster.ca/~zhechen/download/alopex.ps.

Drullman, R., Festen, J. M., & Plomp, R. (1994). Effect of reducing slow temporal modulations on speech reception. *Journal of the Acoustical Society of America, 95*(5), 2670–2680.

Duda, R. O, Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York: Wiley.

Fishbach, A., Yeshurun, Y., & Nelken, I. (2003). Neural model for physiological responses to frequency and amplitude transitions uncovers topographical order in the auditory cortex. *Journal of Neurophysiology, 90*, 3663–3678.

Gupta, S. (2004). *Efficient testing of the Neurocompensator through the development of an unsupervised learning clustering algorithm and an adaptive psychometric function*. Bachelor's thesis, McMaster University.

Harth, E., & Tzanakou, E. (1974). Alopex: A stochastic method for determining visual receptive fields. *Vision Research, 14*, 1475–1482.

Harth, E., Unnikrishnan, K. P., & Pandya, A. S. (1987). The inversion of sensory processing by feedback pathways: A model of visual cognitive functions. *Science, 237*, 184–187.

Haykin, S., Chen, Z., & Becker, S. (2004). Stochastic correlative learning algorithms. *IEEE Transactions on Signal Processing, 52*(8), 2200–2209.

Moore, B. C., Huss, M., Vickers, D. A., Glasberg, B. R., & Alcantara, J. I. (2000). A test for the diagnosis of dead regions in the cochlea. *British Journal of Audiology, 34*(4), 205–224.

Moore, B. C., Vickers, D. A., Plack, C. J., & Oxenham, A. J. (1999). Interrelationship between different psychoacoustic measures assumed to be related to the cochlear active mechanism. *Journal of the Acoustical Society of America, 106*(6), 2761–2778.

Nelson, P. C., & Carney, L. H. (2004). A phenomenological model of peripheral and central neural responses to amplitude modulated tones. *Journal of the Acoustical Society of America, 116*(4), 2173–2186.

Plack, C. J., & Oxenham, A. J. (2000). Basilar-membrane nonlinearity estimated by pulsation threshold. *Journal of the Acoustical Society of America, 107*(1), 501–507.

Roberts, S. J., Husmeier, D., Rezek, I., & Penny, W. (1998). Bayesian approaches to gaussian mixture modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 20*(11), 1133–1144.

Sastry, P. S., Magesh, M., & Unnikrishnan, K. P. (2002). Two timescale analysis of the Alopex algorithm for optimization. *Neural Computation, 14*, 2729–2750.

Schwartz, O., & Simoncelli, E. (2001). Natural sound statistics and divisive normalization in the auditory system. In T. Leen, T. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems, 13* (pp. 166–172). Cambridge, MA: MIT Press.

Shera, C. A., Guinan, J. J., & Oxenham, A. J. (2002). Revised estimates of human cochlear tuning from otoacoustic and behavior measurements. *Proceedings of National Academy of Sciences, USA, 99*(5), 3318–3323.

Tzanakou, E. (2000). *Supervised and unsupervised pattern recognition: Feature extraction and computational intelligence.* Boca Raton, FL: CRC Press.

Tzanakou, E., Michalak, R., & Harth, E. (1979). The Alopex process: Visual receptive fields by response feedback. *Biological Cybernetics, 35*, 161–174.

Ueda, N., Nakano, R., Ghahramani, Z., & Hinton, G. (2000). SMEM algorithm for mixture models. *Neural Computation, 12*, 2109–2128.

Unnikrishnan, K. P., & Venugopal, K. P. (1994). Alopex: A correlation-based learning algorithm for feedforward and recurrent neural networks. *Neural Computation, 6*, 469–490.

Verbeek, J. J., Vlassis, N., & Kröse, B. (2003). Efficient greedy learning of gaussian mixture models. *Neural Computation, 15*, 469–485.

Wang, K., & Shamma, S. A. (1995). Spectral shape analysis in the central auditory system. *IEEE Transactions on Speech and Audio Processing, 3*(5), 382–395.

Received September 3, 2004; accepted April 19, 2005.