Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2009, Article ID 465189, 10 pages
doi:10.1155/2009/465189

Research Article

Influence of Acoustic Feedback on the Learning Strategies of Neural Network-Based Sound Classifiers in Digital Hearing Aids

**Lucas Cuadra (EURASIP Member), Enrique Alexandre,
Roberto Gil-Pita (EURASIP Member), Raúl Vicen-Bueno, and Lorena Álvarez**

*Departamento de Teoria de la Señal y Comunicaciones, Escuela Politecnica Superior, Universidad de Alcala,
28805 Alcala de Henares, Spain*

Correspondence should be addressed to Lucas Cuadra, lucas.cuadra@uah.es

Received 1 December 2008; Revised 4 May 2009; Accepted 9 September 2009

Recommended by Hugo Fastl

Sound classifiers embedded in digital hearing aids are usually designed by using sound databases that do not include the distortions associated to the feedback that often occurs when these devices have to work at high gain and low gain margin to oscillation. The consequence is that the classifier learns inappropriate sound patterns. In this paper we explore the feasibility of using different sound databases (generated according to 18 configurations of real patients), and a variety of learning strategies for neural networks in the effort of reducing the probability of erroneous classification. The experimental work basically points out that the proposed methods assist the neural network-based classifier in reducing its error probability in more than 18%. This helps enhance the elderly user's comfort: the hearing aid automatically selects, with higher success probability, the program that is best adapted to the changing acoustic environment the user is facing.

1. Introduction

Acoustic feedback appears when part of the conveniently amplified output signal produced by a digital hearing aid returns through the auditory canal, and enters again this device, being thus anew amplified [1–6]. Sometimes feedback may cause the hearing aid to become unstable, producing very unpleasant and irritating howls. Preventing the hearing aid from such instability enforces designers to limit the maximum gain that can be used to compensate the patient's acoustic loss. In this regard, along with *noise reduction* [1, 7, 8], the topic of *controlling acoustic feedback* plays a key role in the design of hearing devices [1, 3–5, 9–17]. In particular, a very extensive and clear review by A. Spriet et al. on the topic of *adaptive feedback cancellation* in hearing aids may be found in [3–5] for further details.

However, even without reaching the limit of instability, feedback often affects negatively the performance of those hearing aids that operate with high levels of gain, causing, for instance, distortions [1, 3–5]. In this situation, a relevant application—whose performance may be presumably affected, and on which this paper focuses—is the one in which the hearing aid *itself* classifies [1, 18–23] the acoustic environment that surrounds the user, and automatically selects the amplification "program" that is best adapted to such environment ("self-adaptation") [20–23].

Within the more general and highly relevant topic of sound classification in hearing aids [1, 18, 19], self-adaptation is currently deemed very appreciated by hearing aid users, specially by the elderly, because the "manual" approach (in which the user has to identify the acoustic surroundings and chooses the more adequate program) is extremely uncomfortable, and very often exceeds the abilities of many hearing aid users [24, 25]. Only about 25% of hearing aid recipients (a scarce 20% of those that could benefit from hearing aids) wear it because of the unpleasant whistles and/or other amplified noises the hearing instrument often produces, and in particular, when moving from one acoustic environment (e.g., speech-in-quiet) to another different one (say, for instance, a crowded restaurant) for which the active program is not suitable (the user thus hears a sudden, uncomfortable amplified noise).

More figures confirming these facts are, for instance, that about one-third of Americans between the ages of 65 and 74 suffer from hearing loss, and about half the people who are 85 and older have hearing loss [26]. Or that about sixteen percent of adult Europeans have hearing problems strong enough to adversely affect their daily life. The Royal National Institute for Deaf People (RNID) has reported that there are 8.7 million deaf and hard of hearing people in the UK, and that just one in four hearing-impaired Britons has received a hearing aid [27]. Furthermore, the number of people with hearing loss is increasing at an alarming rate not only because of the aging of the world's population, but also because of the growing exposure to noise in daily life. These facts illustrate the necessity for hearing aids to automatically classify the acoustic environment the user is in.

In addition, most elderly people regrettably suffer from a pernicious presbyacusis with deep loss at some frequencies. The device has then to work with *high level of gain*, and usually with a *very short margin gain to oscillation* [3]. As a result, the sound processed in the device not only is a "clean" version of the "external" acoustic environment but also contains irritating distortions. Thus, the sound classifier will not work properly if the classification algorithm designed were based simply on clean (feedback-free) sounds.

In this paper we explore whether it is worthwhile or not to include the effects of this distortion (caused by feedback) in the training process of a neural network-based classifier. The way the feedback should be included in the learning process and to what extent this has an effect on the classifier efficiency constitute the *general purpose* of this paper. Regarding this, it is worth noting that the training process of the classifier is always performed *off-line* on a desktop computer, and *never* on the hearing aid itself. Only once the neural network-based classifier has been properly trained, validated, and tested in the laboratory, this classifier will be uploaded onto the hearing aid!

Note that a question related to the general topic proposed in this paper could be as follows: what is the effect of feedback reduction algorithms on the classification process? These algorithms may increase the overall gain significantly before feedback effects is occurr. Would the classifier implemented on the device have to select different feature sets based on the presence or absence of feedback? Obviously, addressing this complex question in details would require a new study, which is clearly out of the main scope of this paper.

After having a look at some key design aspects in Section 2, we will focus on two tasks, which are intimately linked, that will guide our research. The first one consists in creating a variety of adequate sound databases that should incorporate the effects of feedback. The "original" database, from which the others will be derived, consists of samples of different environmental sounds (Section 3), and will be modified according to various configurations of real patients (the hearing loss and his/her type of hearing instrument). We will focus on 18 real patients (study cases) whose hearing loss is specially problematic (since they require high gain at some frequencies, and consequently, a low gain margin to oscillation) (Section 4.1). The databases including sounds with feedback, as will be explained in Section 4.2, will be created by filtering "normal sounds" (without feedback) by means of systems [10, 28] that simulate the dissimilar interactions (feedback paths) between the hearing aid and the user's auditory system for each of the real study cases. The second step—based on making different uses of the sogenerated databases containing sounds with feedback—consists in exploring a variety of *learning-and-test strategies* inspired by a leave-one-out strategy, and determining which is the most appropriate (Section 5).

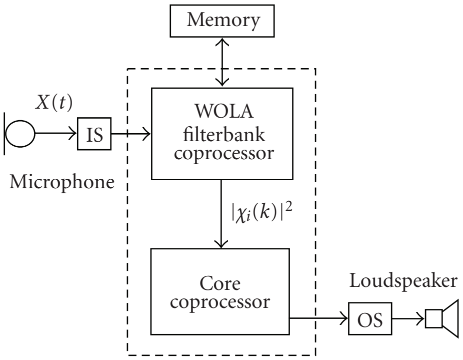

Figure 1: Simplified architecture diagram. For the sake of clarity, it only shows the two coprocessors that operate concurrently. The WOLA filterbank coprocessor computes $|\chi_i[k]|^2$, $\chi_i[k]$ are being the kth frequency bin of the spectrum of a given sound frame $X_i(t)$, while the "core processor" deals with the remaining signal processing tasks. $X(t)$ labels the sound signal entering the microphone, while IS and OS represent, respectively, the input and output stages.

2. Framework, Design Limitations, and Problem Statement

Implementing complex algorithms on digital hearing aids is limited by some design restrictions. These devices basically consist of a microphone, a loudspeaker, a digital signal processing (DSP), and a battery. Among the constraints, battery life still remains certainly a big problem. Although memory and computational power (number of instructions) are currently less critical, the optimization of resources remains a key issue for more demanding algorithms such as, for instance, neural network (NN) classifiers, noise reduction, or dereverberation systems [1, 3].

In particular, the DSP [29, 30] used by our platform to carry out the experiments is basically composed of two coprocessors operating concurrently, as schematically illustrated in Figure 1. The first one is the weighted overlapp-add (WOLA) filterbank coprocessor, which performs the time/frequency decomposition with $N_B = 64$ frequency bands: it provides $|\chi_i[k]|^2$, $\chi_i[k]$ being the kth frequency bin of the spectrum of a given sound frame $X_i(t)$. The second coprocessor is the "core processor" dealing with the remaining tasks, such as, for instance, compensating the hearing loss, reducing noise, classifying the acoustic environment, and so on.

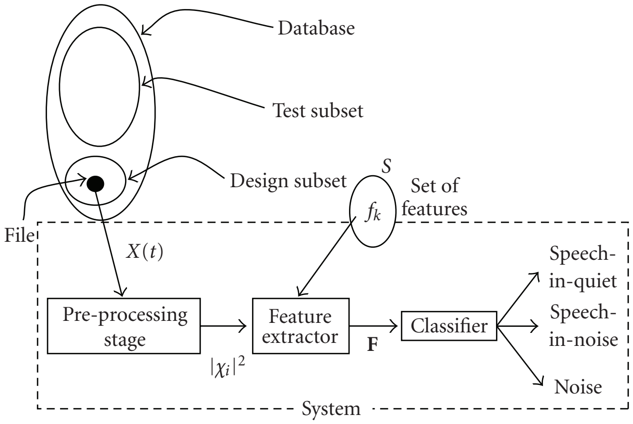

Figure 2: Conceptual description of the way the complete classifying system works. This consists of a preprocessing stage, a feature extraction stage (which computes a number of features arranged in the vector **F**), and a classifying algorithm, which categorizes input sounds from the database into three classes (speech, music, and noise). $X(t)$ is the sound signal contained in each file in the database. $\chi_i[k]$ is the spectrum of any frame, $X_i(t)$, into which the signal is segmented. S is the set of available features.

Regarding the sound classifier, as shown in Figure 2, it basically involves the following (1) a preprocessing block (Section 2.1); (2) a *feature extraction* stage, which computes the main characteristics of the sound (Section 2.2); and (3) some type of *classifying algorithm*, a neural network, that should be presumably able to learn from sounds with feedback (Section 2.3).

2.1. The Preprocessing Stage. Each of the input sounds to be classified, $X(t)$, assumed to be a stochastic process, is segmented into frames, $X_i(t)$, $i = 1,\ldots,r$, r being the number of frames into which the signal is divided. To compute the features, used to characterize any frame, $X_i(t)$, it is necessary to sample the frame: $X_i(t_k)$, $k = 1,\ldots,p$, p being the number of samples per frame. Since each frame, $X_i(t)$, is a windowed stochastic process, any one of its samples, $X_i(t_k)$, is a random variable that, for simplicity, is labeled X_{ik}. Thus, for each audio frame, $X_i(t)$, the following random vector is obtained: $\mathbf{X_i} = [X_{i1},\ldots,X_{ip}]$. As sketched in Figure 1, the WOLA filterbank coprocessor computes its discrete Fourier transform (DFT), leading to $\overline{\chi}_i = [\chi_{i1},\ldots,\chi_{iN_B}]$. This is just the initial information the second stage in Figure 2 uses to calculate all the features aiming at describing frame $X_i(t)$.

The experiments that will be described below have been carried out by using frames of 20 ms length.

2.2. The Feature Extraction Stage. This functional block plays the role of processing the signal in order to extract some kind of valuable information that characterizes it and helps the second stage (the NN-based classifier) to work properly. The features, which must be computed on the DSP, are based on a set S of well-selected, widely-used, sound-describing features that exhibit good performance along with low-medium computational load. They have been selected deliberately simple in order to emphasize the design of a more complex neural network-based classifier (Section 2.3) that is capable of learning sounds with feedback (Section 4.2). The features $f_k \in S$, which take as initial information the matrix $\overline{\chi}_i = [\chi_{i1},\ldots,\chi_{iN_B}]$ computed by the WOLA coprocessor for any sound frame, $X_i(t)$, have been found to be as follows.

(1) The spectral centroid of the sound frame $X_i(t)$, which can be associated with the measure of the sound brightness, is

$$\mathrm{SC}_i = \frac{\sum_{k=1}^{N_B} \left|\chi_{ik}\right| \cdot k}{\sum_{k=1}^{N_B} \left|\chi_{ik}\right|}, \tag{1}$$

where χ_{ik} represents the kth frequency bin of the Fourier transform at frame i, and N_B is the number of frequency bands ($N_B = 64$ in this particular platform). Please note that χ_{ik} merely represents each of the elements in the matrix $\overline{\chi}_i = [\chi_{i1},\ldots,\chi_{iN_B}]$ computed by the WOLA coprocessor for the sound frame $X_i(t)$.

(2) The "Voice2White" ratio of frame $X_i(t)$, proposed in [31], is a measure of the energy inside the typical speech band (300–4000 Hz) in respect to the whole energy of frame $X_i(t)$. It is computed by using the following:

$$\mathrm{V2W}_i = \frac{\sum_{k=M_1}^{M_2} \left|\chi_{ik}\right|^2}{\sum_{k=1}^{N_B} \left|\chi_{ik}\right|^2}, \tag{2}$$

with M_1 and M_2 being the indexes that limit the speech band (300–4000 Hz).

(3) The "Spectral Flux" of frame $X_i(t)$ is associated with the amount of spectral local changes when comparing this frame with the previous one:

$$\mathrm{SF}_i = \sum_{k=1}^{N_B} \left(\left|\chi_{ik}\right| - \left|\chi_{i-1k}\right|\right)^2. \tag{3}$$

(4) The short time energy of frame $X_i(t)$ is defined as

$$\mathrm{STE}_i = \frac{1}{N_B} \sum_{k=1}^{N_B} \left|\chi_{ik}\right|^2. \tag{4}$$

Note that since $\overline{\chi}_i = [\chi_{i1},\ldots,\chi_{iN_B}]$ has been found to be a random-variable vector, then any feature $f_k \in S$ applied to it, $f_k(\overline{\chi}_i)$, is thus a function of N_B random variables, $f_k(\chi_{i1},\ldots,\chi_{iN_B})$, and, consequently, a random variable [32]. In order to simplify the notation, the random variable $f_k(\chi_{i1},\ldots,\chi_{iN_B})$ will be labeled f_{ki}. Finally, to complete the characterization of the audio input signal, the above-mentioned sequence of processes has to be applied to all the r frames into which the entering sound has been segmented.

For the sake of simplicity, we have labeled $[f_{k1},\ldots,f_{kr}] \equiv \mathbf{F_k}$ as the feature vector that contains those elements obtained when feature f_k is applied to each of the r frames into which the sound signal has been segmented. We have completed the statistical characterization of the random vector $\mathbf{F_k}$ by estimating its mean value, $\widehat{E}[\mathbf{F_k}]$, and its variance, $\mathrm{Var}[\mathbf{F_k}]$ [32].

Finally, this characterization must be done for *all* the available features $f_k \in S$. The feature *extraction* algorithm ends in generating the following feature vector: $\mathbf{F} = [\widehat{E}[\mathbf{F}_1], \mathrm{Var}[\mathbf{F}_1], \ldots, \widehat{E}[\mathbf{F}_{n_f}], \mathrm{Var}[\mathbf{F}_{n_f}]]$, its dimension being $\dim(\mathbf{F}) = 2n_f = N_F$. This is just the signal-describing vector that feeds the classifier. For the sake of clarity, it is written formally as $\mathbf{F} = [F_1, \ldots, F_{N_F}]$, which is the vector represented in Figure 2 as entering the classifying algorithm.

2.3. The Classifying Algorithm: The Neural Network Approach. In order to make the sound classifier work better with sounds that contain feedback, we have chosen, among a variety of previously explored algorithms, a particular kind of neural network (NN) [33–35], which should be properly designed for being constrained to the hardware requirements of the DSP the hearing aid is based on (see [20] for further details). The key reason that has compelled us to choose the NN approach is that neural networks are able to learn from appropriate training pattern sets, and properly classify other patterns that have never been found before [20, 22, 25]. This ultimately leads to very good results in terms of smaller error probability when compared to those from other popular algorithms such as the rule-based classifier [35], the Fisher linear discriminant [23, 34, 36], the minimum distance classifier, or the k-nearest neighbour algorithm [35]. Despite of its *presumably* high computational cost, its implementation has been proven to be feasible on digital hearing aids: it requires some tradeoffs involving a balance between reducing the computational demands (that is the number of neurons) and not degrading the quality perceived by the user [20].

The basic architecture of the "original" NN, which is the cornerstone of the learning schemes to be explored in Section 5, consists of three layers of neurons (input, hidden, and output layers), interconnected by links with *adjustable weights* [33, 35]. As will be better understood later on, we have named it "original neural network" in the effort of emphasizing that, depending on the learning strategy adopted, a *number of different neural network configurations* will be finally obtained. That is, although with different details that will be explained in Section 5, all the classifiers have been evolved from a basic configuration whose main aspects are as follows.

In this original NN architecture, the number of input neurons corresponds to that of the features used to characterize the sound, the number of output neurons is related to the three classes we are interested in, and finally, the number of hidden neurons depends on the adjustment of the complexity of the network [35]. We have explored how many hidden neurons are required by means of batches of experiments that have ranged from 1 to 40 neurons. A higher number of hidden neurons have been found to be unfeasible because of the greater associated computational cost. Any of these experiments, which aim at estimating the number of hidden neurons, has been repeated 10 times. In this design process, the "proper" NN configuration is precisely the one that exhibits the lowest error probability computed over the *validation set*. The Levenberg-Marquardt algorithm [33, 37] with Bayesian regularization [35, 38] has been found to be the most appropriate method for training the neural network. The main advantage of using regularization techniques relies on the fact that the generalization capabilities of the NN ends in being improved, making it capable of reaching good results even with smaller networks, since the regularization algorithm itself prunes those neurons that are not strictly necessary [20].

Please note that, as mentioned, in our platform, the features are computed every 20 ms, and, at the end, the classifier makes a decision every 2.5 seconds, based on the mean value and the variance of the features during that time.

3. The Original "Feedback-Free" Sound Database

As mentioned in Section 1, our research aims to take the distortions due to feedback into account when training the neural network-based classification algorithm. This requires making use of a number of databases that should include different feedback effects. These databases, as will be explained in Section 4.2, will be created by filtering "normal sounds" (with no feedback) using systems that simulate the dissimilar interaction between a hearing aid and its user's auditory system [10]. We have grouped such normal, feedback-free sounds in a database labeled $\mathcal{D}_0$ that, for the sake of clarity, we have called the "original database" or the "unfiltered" database.

This "feedback-free" database $\mathcal{D}_0$ contains a total of 7340 seconds of audio, including speech-in-quiet, speech in noise, speech-in-music, vocal music, instrumental music, and noise. The database has been manually labeled, leading to a total of 1272.5 seconds of "speech-in-quiet", 3637.5 seconds of "speech-in-noise", and 2430 seconds of "noise". These classes have been considered by our patients as the most practical ones in their daily life. Note that, within the context of the application at hand, music files (both vocal and instrumental) have also been categorized as "noise" sources, since emphasis is placed here on improving speech intelligibility, the first priority for the patients. All sound files are monophonic, have been sampled at $f_S = 16$ kHz, and have been coded with 16 bits per sample. Both speech and music files were provided by D. Ellis, and were recorded by Scheirer and Slaney [39]. This database [40] has already been used by different authors in a number of works [39, 41–43].

The designers made a strong attempt to collect a data set which represented as much of the breadth of available input signals as possible as follows.

(i) Speech was recorded by digitally sampling FM radio stations, using a variety of stations, content styles, and *levels*. This variety of sounds allows to test the robustness of the classification system as a function of different sound input levels. Additionally, the speech sounds were recorded from a uniformly distributed set of male and female speakers in the aim of making classification as robust as possible.

(ii) Music sounds include samples of jazz, pop, country, salsa, reggae, classical, various non-Western styles, various sorts of rock, and new age music, both with and without vocals.

(iii) Finally, noise samples contain sounds from the following diverse environments: aircraft, bus, cafe, car, kindergarten, living room, nature, school, shop, sports, traffic, train, and train station. These sources of noise have been artificially mixed with those of speech files (with varying grades of reverberation) at different signal-to-noise ratios (SNRs) ranging from 0 to 10 dB. In a number of experiments, these values have been found to be representative enough of the following fact related to perceptual criteria: lower SNRs could be treated by the hearing aid as noise, and higher SNRs could be considered as clean speech.

The sound files exhibit different input levels, with a range of 30 dB between the lowest and the highest.

For proper training, validation, and testing, it is necessary for the database $\mathcal{D}_0$ to be divided into three different subsets. We formally write this as $\mathcal{D}_0 = \mathcal{P}_0 \cup \mathcal{V}_0 \cup \mathcal{T}_0$, where $\mathcal{P}_0$, $\mathcal{V}_0$, and $\mathcal{T}_0$ represent, respectively, the "training", "validation", and "test" subsets. These sets contain 2685 seconds ($\approx$36%) for training, 1012.5 seconds ($\approx$14%) for validation, and 3642.5 seconds ($\approx$50%) for testing. This division has been done randomly, ensuring that the relative proportion of files of each category is preserved for each set. Since the number of patterns is high enough, no leave-m-out process has been used, and only one repetition has been made. However, and as will be shown when designing the modified databases of sounds with the feedback of the different study cases (Section 4), a number of strategies, based on the leave-one-out principle, have been adopted in the effort of enhancing the ability of generalization of the corresponding neural networks (Section 5).

4. Study Cases: Patients and Modified Databases

4.1. The Real Patients. Figure 3 illustrates the problem of acoustic feedback that often occurs in the system formed by a hearing aid and the outer ear of its user. G represents the effective gain corresponding to the *forward path* of the "normal" (no-feedback) signal processing of the hearing aid. On the other hand, F labels the equivalent *feedback path* between the loudspeaker and the microphone [2, 3, 10]. The closed-loop system soformed is stable if and only if the open-loop gain fulfills $|G(\omega)F(\omega)| < 1$, $\forall \omega \in [0, \pi]$ with positive feedback, that is, with phase-lag $\varphi[(G(\omega)F(\omega))] = n2\pi$, $\forall n \in \mathbb{Z}$ [3]. If the gain is increased beyond this limit, the system begins to oscillate. Furthermore, at low gain margin to oscillation, $\mathrm{GMO[dB]} = -20\log|G(\omega)F(\omega)|$, acoustic feedback degrades the quality of sound by producing howling or ringing. In the effort of avoiding significant distortion, a gain margin of at least 6 dB is recommended [3].

Thus the distortion caused by feedback often occurs in hearing instruments that operate with high levels of amplification, and, as a result, with low GMO. To what extent the sounds are affected by feedback depends basically on the following: (1) the gain to compensate the loss at each frequency, (2) the *type* of hearing aid—in the ear (ITE), in the canal (ITC), and behind the ear (BTE)—, and (3) the way this is coupled with the user's outer ear (which, at the end, determines the GMO). This is just the case of our 18 study examples: Table 1 lists, for any patient $\{P_1, \ldots, P_{18}\}$ (and the corresponding hearing aid, $\{\mathrm{HA}_1, \ldots, \mathrm{HA}_{18}\}$), the GMO(dB) as a function of frequency (Hz). A more detailed description of the real patients from whom the data were collected can be found in [10].

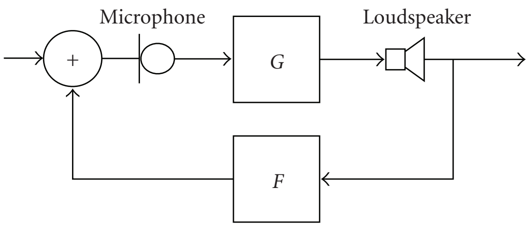

Figure 3: Representation of the problem of acoustic feedback in hearing aids. G represents the effective gain corresponding to the *forward path* of the "normal" (no-feedback) signal processing of the hearing aid. F labels the equivalent *feedback path* between the loudspeaker and the microphone.

4.2. Creating Feedback-Affected Training Pattern Sets. The more realistic sounds (with feedback) that each patient P_i hears have been generated as explained in [10]. The equivalent feedback path, F, which contains the combination of all the feedback factors, has been modeled from a variety of empirical studies [2]. The proper design of the gain curves for each patient (and his/her corresponding hearing aids) has been carried out in order to fulfill the gain requirements specified by the FIG6 prescription method [6]. For further details, reference [10] contains an extensive description of the compression and the gain curves. Without going into details, and aiming at explaining it as simply as possible, the simulator works as a "filter" that takes "feedback-free" sounds and generates "feedback-affected" sounds like those the user P_i usually hears. As illustrated in Figure 4, for any patient P_i, the generation of his/her particular feedback-affected soundpatterns is qualified to filter any of the sound samples that belong to the the original database, $\mathcal{D}_0$, by means of the filter $\mathcal{F}_i$. It provides, at the end, the modified database with feedback, $\mathcal{D}_i$, subscript i meaning that it contains the feedback distortions corresponding to patient P_i. Bear in mind that any modified database $\mathcal{D}_i$ may be formally written as $\mathcal{D}_i = \mathcal{P}_i \cup \mathcal{V}_i \cup \mathcal{T}_i$, where $\mathcal{P}_i$, $\mathcal{V}_i$, and $\mathcal{T}_i$ are, respectively, the *modified* training, validation, and test subsets.

Once the initial, "unfiltered" database has been conveniently modified, the next step should consist in training and validating the NN-based classifier by using the subsets $\mathcal{P}_i$ and $\mathcal{V}_i$. From a conceptual point of view, it is very important to remark that, when the original classifier is trained by using a variety of different feedback-affected training sets, $\mathcal{P}_i$, it is possible to finally obtain *dissimilar* classifiers. This is because they have learnt different patterns (since the training sets $\mathcal{P}_i$ are different from each other, $i = 1, \ldots, N_P$) and, finally, they could work in a different manner.

Table 1: Gain margin to oscillation (GMO[dB]) as a function of frequency (Hz) for any of the 18 patients ($\{P_1, \ldots, P_{18}\}$) and the corresponding hearing aid ($\{HA_1, \ldots, HA_{18}\}$). ITE, ITC, and BTE mean, respectively, "in the ear", "in the canal", and "behind the ear". Those columns labeled "125, . . . , 8 k" list the GMO[dB] in any of these bands (from 125 Hz to 8 KHz).

		GMO(dB) as a function of frequency (Hz)								
P_i	HA$_i$	125	250	500	1 k	2 k	3 k	4 k	6 k	8 k
P_1	ITE	46	45	44	41	31	24	16	10	6
P_2	ITE	43	41	40	39	31	22	16	12	7
P_3	ITE	33	32	38	37	34	22	9	10	6
P_4	ITC	43	42	42	39	26	18	13	6	6
P_5	ITC	41	41	39	39	34	21	15	6	6
P_6	ITC	43	42	41	41	36	28	23	6	7
P_7	ITE	45	44	44	41	31	29	13	11	6
P_8	BTE	46	45	40	27	24	24	12	12	14
P_9	BTE	48	46	46	40	40	38	26	30	26
P_{10}	BTE	50	49	47	29	29	24	21	30	27
P_{11}	ITC	42	42	40	39	30	21	18	6	11
P_{12}	ITC	41	39	40	40	20	15	12	6	7
P_{13}	ITE	41	39	41	41	37	33	22	16	12
P_{14}	ITE	44	43	43	42	36	24	9	9	6
P_{15}	ITE	43	41	38	35	25	13	9	11	6
P_{16}	ITE	41	39	43	39	17	12	9	11	7
P_{17}	ITC	43	43	41	34	30	23	18	6	10
P_{18}	ITC	43	42	41	40	25	15	13	6	8

5. Learning-and-Testing Strategies

All the strategies that will be explained in this section, although conceptually different, will have their results statistically characterized by using the *mean value* (μ) and the *standard deviation* (σ)—or equivalently, the variance (σ^2)—of their *error probability* (P_e) computed over properly designed *test* sets. For the sake of clarity, and not repeating unnecessarily the same concept particularized to the different approaches, the error probability estimated for each method will be assumed to be a random variable. Thus, if X labels any of these random variables that, for instance, take M values, $x_i, i = 1, \ldots, M$, its mean value and variance can be computed by using the estimators [33] that follow:

$$E[X] \equiv \mu = \frac{1}{M}\sum_{i=1}^{M} x_i,$$
$$\text{Var}[X] \equiv \sigma^2 = \frac{1}{M}\sum_{i=1}^{M} (x_i - \mu)^2, \tag{5}$$

where $E[\cdot]$ means the *mathematical expectation* of X, while Var$[\cdot]$, means the variance.

Once we have the tools for statistically characterizing the *learning-and-testing strategies*, we can proceed further in studying them more deeply. In this respect, although with some details that will be explained later on, the methods we have explored can be conceptually categorized into two different groups of learning strategies: training without feedback (the "conventional approach", which will be outlined in Section 5.1) and the novel feedback-including learning strategies this paper centers on ("Training with feedback", which will be described in Section 5.2).

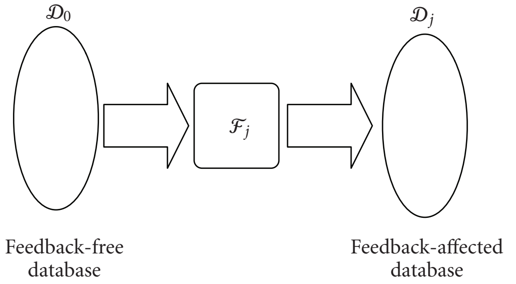

Figure 4: Representation of the way any feedback-affected database ($\mathcal{D}_j, j = 1, \ldots, N_P$) is created from the original one, $\mathcal{D}_0$.

5.1. "Conventional Approach": Training without Feedback. The purpose of exploring this procedure is to clearly discern and quantify to what extent the learning strategies to be explored, which aim at properly introducing the feedback effects, are effective or not. The "conventional approach" refers to the one, often used, in which the NN is designed by using the *unfiltered* (feedback-free) training and validation subsets, $\mathcal{P}_0, \mathcal{V}_0 \subset \mathcal{D}_0$. The NN-based classifier so-trained—the one that minimizes the error probability over the *validation* set $\mathcal{V}_0$—has been labeled NN$_0$, where subscript "0" means "feedback-free". Since $\mathcal{D}_0$ has been modified according to 18 configurations of real patients, then the classifier can be now tested by using the generated test sets $\mathcal{T}_j, j = 1, \ldots, 18$, corresponding to these study cases. Using the estimators (5), the mean value and the standard deviation of the error probabilities have been found to be, respectively, $\mu_{\text{conv}}[P_e] = 8.16\%$ and $\sigma_{\text{conv}}[P_e] = 0.31\%$, subscript "conv" labeling "conventional approach".

5.2. Novel Approaches: Learning by Using Patterns with Feedback. Please note that, as pointed out in Section 4, and as occurs in other biomedical problems, the drawback here consists in the fact that there are a large number of data over a relatively small number of samples (patients). This "high-dimensional problem" causes difficulties to most of the long-established classifiers. This is precisely one of the motivations that has compelled us to explore novel strategies in the aim of improving the classifier ability to generalize.

These approaches are inspired by a leave-one-out strategy that aims to enhance the ability to generalize of the sodesigned NN. Of course, a more general m-fold cross-validation technique could potentially be used by randomly dividing the corresponding data into m disjoint sets of *equal size* n/m, n being the total number of sound patterns [35]. The problem is that splitting it into m *equal* parts is very difficult because of the low number of patients. (Note that the number of patients is much smaller than the number of sound samples in the database). On the other hand, a *bootstrap* set [35] could be created by randomly selecting n points from the training set with replacement. This process could be independently repeated B times to yield B bootstrap sets, which may be treated as independent sets. If $B < n$, it will result in some saving in computation time. But, as mentioned, we do *not* need to save these computational resources because the design phase is carried out *off-line* on a desktop computer.

Therefore, the learning strategies we have explored, and which aim to include sound with feedback based on leave-one-out strategy, can be categorized into the three strategies that follow.

5.2.1. Training with an Average Feedback-Affected Set. This strategy works as follows.

(1) Take the training set $\mathcal{T}_j$, and leave it out for being ulteriorly used for testing.

(2) Create the "average" design set involving feedback-affected data from the remaining $N_P - 1$ patients ($P_i, i = 1,\ldots,N_P; i \neq j$). For doing so, the following is required: (a) computing the feedback-modified design sets ($\mathcal{P}_i$, $\mathcal{V}_i$, $i = 1,\ldots,N_P; i \neq j$), (b) estimating the *features* that properly describe the sounds belonging to such modified sets as explained in Section 2.2, and(c) statistically characterizing these features by estimating the mean value and the standard deviation.

(3) Train the NN-based classifier with the "average" features, leading to a NN that we have labeled $\mathrm{NN}_{\mathrm{av},j}$. In this notation, subscript "av" stands for "average", while j means that its performance will be tested with the feedback-modified test set, $\mathcal{T}_j$, corresponding to the patient P_j that has been left out during the design stage.

(4) Estimate the error probability over $\mathcal{T}_j$.

(5) Repeat step 1 in order for index j to sweep all possible values ($j = 1,\ldots,N_P$).

Please note that the aforementioned approach exhibits two interesting aspects as follows.

(a) This strategy creates N_P different NN-based classifiers:

$$\{\mathrm{NN}_{\mathrm{av},1}, \mathrm{NN}_{\mathrm{av},2}, \ldots, \mathrm{NN}_{\mathrm{av},N_P}\}. \tag{6}$$

(b) According to the leave-one-out strategy adopted, any of the learning-machines $\mathrm{NN}_{\mathrm{av},j}$ is tested with the feedback-affected data that have been kept out during the design process, $\mathcal{T}_j$. The use of estimators (5) leads to $\mu_{\mathrm{av}}[P_e] = 7.63\%$, and $\sigma_{\mathrm{av}}[P_e] = 0.79\%$.

5.2.2. Training with "All" (Massive Database). What motivates this strategy is to answer the question whether or not the creation of a large training set with information involving the different feedback phenomena for all the N_P patients could enrich the learning process. This creation process can be summarized as follows.

(1) Take a representative patient P_j and leave his/her test set ($\mathcal{T}_j$) out for being used in the subsequent test.

(2) In the double effort of (a) designing a database characterized by containing a very rich information encompassing all the feedback phenomena, and (b) preventing the NN from overfitting, we have created the *training set* $\bigcup_{\forall i \neq j}^{N_P} \mathcal{P}_i$ that we have labeled $\mathcal{P}_{\forall i \neq j}$ because it contains information from all patients' databases except that from the one (P_j) that has been left out for testing.

(3) Create in a similar way the *validation set* $\mathcal{V}_{\forall i \neq j} = \bigcup_{\forall i \neq j}^{N_P} \mathcal{V}_i$.

(4) Train and validate with $\mathcal{P}_{\forall i \neq j}$ and $\mathcal{V}_{\forall i \neq j}$, respectively.

(5) The NN so designed, $\mathrm{NN}_{\forall i \neq j,j}$, is the one that minimizes the error probability over the validation set $\mathcal{V}_{\forall i \neq j}$.

(6) Test with the feedback-affected test set that we have left out for this purpose, $\mathcal{T}_j$, thus obtaining $p_{\forall i \neq j,j}$

(7) Repeat steps 1–6 for all remaining the patients.

By making use of arguments in the same line of reasoning as those in the previous methods, the error probability achieved by the classifier trained and tested with this approach can be characterized by its mean value, which has been found to be $\mu_{\mathrm{all}}[P_e] = 7.88\%$, and its standard deviation, $\sigma_{\mathrm{all}}[P_e] = 0.40\%$.

As probably noted by the reader, the feedback-affected learning methods explored till reduces the mean error probability when compared to that achieved by the conventional approach (8.16%), however, some of them, for instance, the "average" training reduces the mean value at the expense of increasing the variance. The method described below has been designed in the effort of reducing even more such error probability, both in mean value and standard deviation.

5.2.3. Improved Training Based on Selecting the Best NN Among All the Patients. In this approach, inspired again by a leave-one strategy, we have performed the following sequence of operations.

(1) Take one of the case study patients, let us say, P_j, for instance, and leave his/her test set $\mathcal{T}_j$ out for being ulteriorly used as a test subject.

(2) Train a number of $N_P - 1$ different NN's by making use of the remaining training and validation sets, $\mathcal{T}_i$, and $\mathcal{V}_i$, $i = 1, \ldots, N_P$, $i \neq j$.

(3) Among these NN's ($\{\mathrm{NN}_{\mathrm{best},i}, i \neq j\}$), select the *best* one in terms of validation error.

(4) Such NN, labeled now, $\mathrm{NN}_{\mathrm{best},i \neq j}$, is tested with the feedback-affected sounds belonging to the test set $\mathcal{T}_j$ that has been kept out during the design stage. Let the socalculated error probability be labeled $p_{\mathrm{best},i \neq j}$.

(5) Repeat this process for all the remaining patients.

When extending this method for all the patients in Table 1, the process finally ends in providing N_P values, $\{p_{\mathrm{best},1}, p_{\mathrm{best},2}, \ldots, p_{\mathrm{best},N_P}\}$, whose mean value and standard deviation have been found to be, respectively, $\mu_{\mathrm{best}}[P_e] = 6.77\%$, and $\sigma_{\mathrm{best}}[P_e] = 0.18\%$.

6. Discussion

Figure 5 shows in a comprehensive way the results we have obtained in the effort of validating the proposed learning-and-testing techniques. We have chosen a representation that illustrates the results as a function of the mean value and the standard deviation of the error probability estimated for each of the different procedures. The key points to note in Figure 5 are as follows.

(1) The conventional method performs worse than the novel strategies: the neural network-based classifier designed with the conventional approach (NN_0) has a mean error probability ($\mu_{\mathrm{conv}}[P_e] = 8.16\%$), which is higher than those achieved by the classifiers trained with the novel methods we have proposed ($\mu_{\mathrm{all}}[P_e] = 7.7\%$, $\mu_{\mathrm{av}}[P_e] = 7.63\%$, $\mu_{\mathrm{best}}[P_e] = 6.77\%$). This is because NN_0 has learnt patterns that are *not representative enough* of sounds with feedback.

(2) Although the novel approaches labeled "all" and "average" assist the different classifiers in reducing the *mean* error probability, they achieve this at the expense of increasing the standard deviation when compared to that of the conventional approach.

(3) On the contrary, the novel strategy called "best" makes the corresponding sound classifier to achieve the best results since it helps reduce the mean error probability ($\mu_{\mathrm{best}}[P_e] = 6.77\%$) while maintaining the standard deviation within a low value ≈0.018%.

(4) One important question that could arise when comparing the mean error probabilities estimated by using the different approaches is whether these different values statistically significant? To answer this question, it is important to analyze the relative error rate associated with the results.

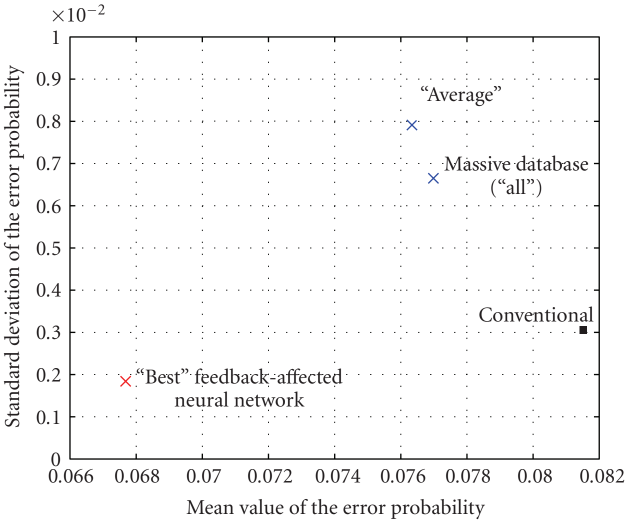

FIGURE 5: Two-dimensional plot illustrating the mean value and the standard deviation of the error probability corresponding to the different *learning-and testing strategies* explored in this work.

The relative error $\varepsilon_{\hat{P}_e}$ of the error probability estimator, $\hat{P}_e$, with a given confidence interval α, is given by [44]:

$$\varepsilon^2_{\hat{P}_e} \leq \frac{(1 - P_e) \cdot \left[Q^{-1}(\alpha/2)\right]^2}{M \cdot P_e}, \tag{7}$$

where P_e is the probability to be calculated, M is the number of elements in the test set, and $Q^{-1}(x)$ is the complementary error function defined as

$$Q(x) = \int_x^{\infty} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{t^2}{2}\right) dt. \tag{8}$$

For our application, and considering a confidence interval of $\alpha = 0.99$, the relative error for the estimation has been found to fulfill $\varepsilon_{\hat{P}_e} < 0.0095$. That is, those differences among error probabilities greater that this value *must be considered as significant*.

The reason why the learning-and-testing strategy called "best" reaches the lowest error probability in both mean and standard deviation ($\mu_{\mathrm{best}}[P_e] = 6.77\%$, and $\sigma_{\mathrm{best}}[P_e] = 0.18\%$) is because, in this approach, the classifier finally created is the neural network that results in being selected among the neural networks that, as pointed out in Section 5.2.3, have been trained with feedback-affected sets in a leave-one-out strategy that prevents them from overfitting. Although not clear at first sight, the key point to understand this is that, as emphasized in Section 5.2.3, for any patient P_j kept out for ulteriorly test, the algorithm trains $N_P - 1$ different NN's (by making use of the remaining training and validation sets, $\mathcal{T}_i$, and $\mathcal{V}_i$, $i = 1, \ldots, N_P$, $i \neq j$), and selects the *best* network in terms of validation error. By repeating this procedure for the remaining $N_P - 1$ patients (study cases), this process allows to finally include the information from all the patients, and to pick out the "best" neural network, that is, the one that has learnt better:

the lowest validation error is equivalent to say that the neural network has acquired the knowledge of the realistic sounds the patients usually hear, and that it is not excessively specialized (overfitted) since it is able to properly classify novel sounds that it has never heard before.

This feedback-including learning strategy helps the embedded classifier improve its performance because it works more robustly in the sense it reaches the lowest error probability, what in turns leads to enhance the user's comfort: the hearing aid itself selects (with higher success probability) the program best adapted to the varying acoustic environment in which the hearing aid user is listening.

7. Conclusions

Feedback appears when part of the amplified output signal produced by a digital hearing aid returns through the auditory canal and enters again this device. Sometimes feedback may cause the hearing aid to become unstable, producing irritating howls. Avoiding instability leads to limit the maximum gain that can be used to compensate the patient's acoustic loss, and the use of algorithms for *controlling acoustic feedback*.

However, even without reaching the limit of instability, feedback often affects negatively the performance of hearing aids that operate with high levels of gain (and as a result, with low gain margin to oscillation), so that the sound processed in the device is not only a "clean" version of the external acoustic environment but also contains annoying distortions. As a result, the sound classifier does not work properly if the classification algorithm is designed (as usual) by using clean (feedback-free) sounds. Our research aims to take these distortions caused by feedback into account when training the classification algorithm. The original sound database, which consist of samples of different environmental sounds, has been modified according to various configurations ($N_P = 18$) of real patients (hearing loss and hearing instrument). Such properly feedback-affected databases, $\mathcal{D}_i = \mathcal{P}_i \cup \mathcal{V}_i \cup \mathcal{T}_i$—composed of training, validation and test sets, $\mathcal{P}_i, \mathcal{V}_i, \mathcal{T}_i$, respectively—have been created by filtering "normal sounds" (without feedback) by means of systems that model the dissimilar interaction between the hearing aid and the user's auditory system.

Making use of these modified databases $\{\mathcal{D}_i, i = 1, \ldots, 18\}$, we have explored three feedback-enriched *learning-and-testing strategies*.

The first one has evaluated whether or not training-and-validating with a set that contains an "average" feedback would lead to a significant reduction in the classification error probability.

The second one has centered on answering the question whether or not the creation of a large training set (with information involving the different feedback phenomena of all the real patients) could enrich the learning process.

In the third method, for any test set $\mathcal{P}_j$ reserved for future test (in a leave-one-out strategy aiming at preventing overfitting), the algorithm has trained $N_P - 1$ different neural networks (by making use of the remaining training and validation sets, $\mathcal{T}_i$, and $\mathcal{V}_i$, $i = 1, \ldots, N_P$, $i \neq j$), and selected the *best* one in terms of validation error. By repeating this for the remaining patients, this method has been found to be able to include the information from all the patients, and to pick out the "best" neural network. "Best" means here the one that has learnt better: it has acquired the knowledge of the realistic sounds the patients hear usually, and it is not excessively specialized (overfitted) in learning such sounds so that it would not be able to properly classify "novel" sounds that it has never heard before. This is the reason why this approach assists the classifier in reducing its mean error probability from 8.17% (in the conventional approach) down to 6.77%. This finally leads to enhance the user's comfort: the hearing aid itself selects (with higher success probability) the program best adapted to the varying acoustic environment in which the patient is listening.

Acknowledgments

The authors gratefully acknowledge the valuable comments kindly suggested by the anonymous reviewers and the editor, which undoubtedly have greatly improved the quality of this paper. This work has been partially funded by the Comunidad de Madrid/Universidad de Alcalá (projects CCG08-UAH/TIC-4270, CCG07-UAH/TIC-1572, and CCG06-UAH/TIC-0378), the Spanish Ministry of Education and Science (TEC2006-13883-C04-04/TCM), and the Ministry of Science and Innovation (TEC2009-14414-C03-03).

References

[1] J. Kates, *Digital Hearing Aids*, Plural Publishing, San Diego, Calif, USA, 2008.

[2] J. Hellgren, T. Lunner, and S. Arlinger, "System identification of feedback in hearing aids," *The Journal of the Acoustical Society of America*, vol. 105, no. 6, pp. 3481–3496, 1999.

[3] A. Spriet, *Adaptive filtering techniques for noise reduction and acoustic feedback cancellation in hearing aids*, Ph.D. thesis, Katholieke Universiteit Leuven, Leuven, Belgium, 2004.

[4] A. Spriet, G. Rombouts, M. Moonen, and J. Wouters, "Adaptive feedback cancellation in hearing aids," *Journal of the Franklin Institute*, vol. 343, pp. 545–573, 2007.

[5] A. Spriet, K. Eneman, M. Moonen, and J. Wouters, "Objective measures for real-time evaluation of adaptive feedback cancellation algorithms in hearing aids," in *Proceedings of the 16th European Signal Processing Conference (EUSIPCO '08)*, Lausanne, Switzerland, August 2008.

[6] H. Dillon, *Hearing Aids*, Boomerang Press, Turramurra, Australia, 2001.

[7] T. Fillon and J. Prado, "Evaluation of an erb frequency scale noise reduction for hearing aids: a comparative study," *Speech Communication*, vol. 39, no. 1-2, pp. 23–32, 2003.

[8] J. Alcantara, B. Moore, V. Kuhnel, and S. Launer, "Evaluation of the noise reduction system in a commercial digital hearing aid," *International Journal of Audiology*, vol. 42, no. 1, pp. 34–42, 2003.

[9] J. Kates, "Feedback cancellation in hearing aids: results from a computer simulation," *IEEE Transactions on Signal Processing*, vol. 39, no. 3, pp. 553–562, 1991.

[10] R. Vicen-Bueno, R. Gil-Pita, M. Utrilla-Manso, and L. Alvarez-Perez, "A hearing aid simulator to test adaptive signal processing algorithms," in *Proceedings of the IEEE International Symposium on Intelligent Signal Processing*, pp. 1–6, Xiamen, China, December 2007.

[11] R. Vicen-Bueno, A. Martinez-Leira, R. Gil-Pita, and M. Rosa-Zurera, "Acoustic feedback reduction based on filtered-x lms and normalized filtered-x lms algorithms in digital hearing aids based on wola filterbank," in *Proceedings of the IEEE International Symposium on Intelligent Signal Processing (WISP '07)*, pp. 1–6, Xiamen, China, December 2007.

[12] J. Kates, "A computer simulation of hearing aid response and the effects of ear canal size," *Acoustic Society of America*, vol. 83, no. 5, pp. 1952–1963, 1988.

[13] J. Maxwell and P. Zurek, "Reducing acoustic feedback in hearing aids," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 4, pp. 304–313, 1995.

[14] P. Z. J. E. Greenberg and M. Brantley, "Evaluation of feedback reduction algorithms for hearing aids," *The Journal of the Acoustical Society of America*, vol. 108, no. 5, pp. 2366–2376, 2000.

[15] N. Abinwi and B. Rafaely, "Unbiased adaptive feedback cancellation in hearing aids by closedloop identification," *IEEE Transactions on Speech and Audio Processing*, vol. 14, no. 2, pp. 658–665, 2006.

[16] D. Freed and S. Soli, "An objective procedure for evaluation of adaptive antifeedback algorithms in hearing aids," *Ear and Hearing*, vol. 27, no. 4, pp. 382–398, 2006.

[17] D. Freed, "Adaptive feedback cancellation in hearing aids with clipping in the feedback path," *The Journal of the Acoustical Society of America*, vol. 123, no. 3, pp. 1618–1626, 2008.

[18] P. Nordqvist and A. Leijon, "An effcient robust sound classification algorithm for hearing aids," *Acoustic Society of America*, vol. 115, no. 6, pp. 3033–3041, 2004.

[19] P. Nordqvist, *Sound classification in hearing instruments*, Ph.D. thesis, Royal Institute of Technology (KTH), Stockholm, Sweden, 2004.

[20] E. Alexandre, L. Cuadra, M. Rosa, and F. López, "Speech/non-speech classification in hearing aids driven by tailored neural networks," in *Speech, Audio, Image and Biomedical Signal Processing Using Neural Networks*, B. Prasad and S. M. Prassana, Eds., pp. 145–167, Springer, Berlin, Germany, 2008.

[21] E. Alexandre, L. Alvarez, L. Cuadra, M. Rosa, and F. Lopez, "Automatic sound classification algorithm for hearing aids using a hierarchical taxonomy," in *New Trends in Audio and Video*, A. Dobrucki, A. Petrovsky, and W. Skarbek, Eds., vol. 1, Politechnika Bialostocka, 2006.

[22] E. Alexandre, L. Cuadra, L. Perez, M. Rosa-Zurera, and F. Lopez-Ferreras, "Automatic sound classification for improving speech intelligibility in hearing aids using a layered structure," Lecture Notes in Computer Science, Springer, Berlin, Germany, 2006.

[23] E. Alexandre, M. Rosa, L. Cuadra, and R. Gil-Pita, "Application of fisher linear discriminant analysis to speech/music classification," in *Proceedings of the 120th Audio Engineering Society Convention*, Paris, France, May 2006.

[24] M. Büchler, S. Allegro, S. Launer, and N. Dillier, "Sound classification in hearing aids inspired by auditory scene analysis," *EURASIP Journal on Applied Signal Processing*, vol. 2005, no. 18, pp. 2991–3002, 2005.

[25] M. Büchler, *Algorithms for sound classification in hearing instruments*, Ph.D. dissertation, Swiss Federal Institute of Technology, Zurich, Switzerland, 2002.

[26] http://www.nia.nih.gov.

[27] http://www.rnid.org.uk.

[28] R. Vicen-Bueno, A. Martinez-Leira, R. Gil-Pita, and M. Rosa-Zurera, "Modified LMS-based feedback-reduction subsystems in digital hearing aids based on WOLA filter bank," *IEEE Transactions on Instrumentation and Measurement*, vol. 58, no. 9, pp. 3177–3190, 2009.

[29] Dspfactory, *Toccata Plus EDK 2.3.1 Getting Started Guide*, Dspfactory, 2002.

[30] Dspfactory, *Toccata Plus Evaluation and Development Board Manual*, Dspfactory, 2002.

[31] E. Guaus and E. Batlle, "A non-linear rhythm-based style classification for broadcast speechmusic discrimination," in *Proceedings of the 116th Audio Engineering Society Convention*, Berlin, Germany, May 2004.

[32] A. Papoulis, *Probability, Random Variables, and Sthocastic Processes*, McGraw-Hill, New York, NY, USA, 3rd edition, 1991.

[33] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, 1995.

[34] S. Mika, G. Rätsch, J. Weston, B. Schölkpf, and K. Müller, "Fisher discriminant analysis with kernels," in *Proceedings of the IEEE Signal Processing Society Workshop on Neural Networks for Signal Processing*, pp. 41–48, Madison, Wis, USA, August 1999.

[35] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience, New York, NY, USA, 2001.

[36] S. A. Billings and K. L. Lee, "Nonlinear fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm," *Neural Networks*, vol. 15, no. 2, pp. 263–270, 2002.

[37] M. Hagan and M. Menhaj, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 989–993, 1994.

[38] F. D. Foresee and M. T. Hagan, "Gauss-Newton approximation to bayesian learning," in *Proceedings of the International Joint Conference on Neural Networks*, vol. 1, pp. 1930–1935, Houston, Tex, USA, 1997.

[39] E. Scheirer and M. Slaney, "Construction and evaluation of a robust multifeature speech/music discriminator," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '97)*, pp. 1331–1334, Munich, Germany, September 1997.

[40] http://labrosa.ee.columbia.edu/sounds/musp/scheislan.html.

[41] B. Thoshkahna, V. Sudha, and K. R. Ramakrishnan, "A speech-music discriminator using hiln model based features," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '06)*, pp. 425–428, Toulouse, France, May 2006.

[42] A. L. Berenzweig and D. P. W. Ellis, "Locating singing voice segments within music signals," in *Proceedings of the IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA '01)*, pp. 31–34, New Paltz, NY, USA, October 2001.

[43] G. Williams and D. P. W. Ellis, "Speech/music discrimination based on posteriori probability features," in *Proceedings of the 6th European Conference on Speech Communication and Technology (Eurospeech '99)*, Budapest, Hungary, September 1999.

[44] K. Shanmugam and P. Balaban, "A modified Monte Carlo simulation technique for the evaluation of error rate in digital communication systems," *IEEE Transactions on Communications*, vol. 28, no. 11, pp. 1961–1924, 1980.