Learning Algorithms for Audio Signal Enhancement Part 1: Neural Network Implementation for the Removal of Impulse Distortions*

ANDRZEJ CZYZEWSKI, *AES Member*

Technical University of Gdańsk, Sound Engineering Department, 80-952 Gdańsk, Poland

Learning algorithms were implemented for the elimination of impulse distortions found in old records and in transmitted audio signals. Neural network simulations were tested with regard to the detection of scratches and clicks. Some special algorithms for the interpolation of distorted signal fragments were studied. Applied methods, results of musical signal processing, and conclusions are presented.

0 INTRODUCTION

A variety of digital signal processing methods applicable for the removal of impulsive noise are known from the literature [1]–[3]. The algorithms are based on various threshold filtering operations, on some extrapolation techniques [4], on linear prediction implementations, on the orthogonal wavelets approach, or on adaptive filters [5]. Meanwhile some advanced and very well developed methods for analyzing and processing data are only occasionally applied to audio acoustics [6]–[9]. The methods mentioned are known in computer science as artificial intelligence approaches to computational problems.

A common feature of artificial intelligence methods is their capability to learn from examples. Learning is the way by which the knowledge base is built, allowing one to accurately recognize known situations, parameters, features, and dependencies. Consequently most qualified artificial intelligence algorithms are able to generalize the acquired knowledge in such a way that they may match inexact or noisy data, patterns misaligned in time, and data patterns really belonging to the same classes despite the fact that relations among them are hidden. Therefore, the collection of methods applied to the problem of audio restoration should be augmented with learning algorithms.

Learning algorithms were employed by the author for the detection of impulsive distortions and for the interpolation of gaps resulting from the removal of parasite impulses. Since typical learning algorithms are highly nonlinear, only those signal portions qualified as distorted should be processed by these algorithms and the rest of the music material should remain unaffected. Otherwise it would be difficult to avoid strong nonlinear distortions, changing the timbre of the music. In this kind of approach the intervention is thus limited to the portions of signal distorted by unwanted (parasite) impulses.

Neural network algorithms (perceptron models) for the detection and removal of clicks are presented in this part of the paper. The first neural network was trained to detect clicks and to build up a table of damaged intervals. Another neural network was trained in such a way that it is able to perform a kind of nonlinear prediction of the signal, providing a valuable tool for the retrieval of samples obscured by scratches and clicks.

Results of practical applications of neural networks for the removal of scratches and clicks are shown in this part of the paper. The second part of the paper will present a rough-set-based approach for the removal of hiss.

1 AUTOMATIC DETECTION OF PARASITE IMPULSES

1.1 Threshold Algorithms

The purpose of the detection procedure is to collect information concerning scratches, clicks, or other impulsive disturbances affecting audio recordings or audio transmission. Hence the restoration of signal affected by

* Based on papers presented at the 96th, 97th, and 99th Conventions of the Audio Engineering Society, 1994 and 1995; revised 1996 May 28 and 1997 May 7.

impulsive distortions is seen as a two-stage process:

1) Detection of clicks and identification of packet boundaries for the lost samples

2) Recovery of distorted fragments, ensuring the best possible approximation of the original signal characteristics.

In general, impulsive disturbances are characterized by the following features:

- In the time domain, concentration of energy within a short time interval
- In the frequency domain, spreading out of energy in a wide band.

The assumption made regarding typical spectrum-based algorithms concerns the additive character of impulse disturbances (clicks, scratches). Subsequently the linearity of an orthogonal transform is exploited, allowing one to subtract the impulse-related part of the spectrum from the whole spectral representation of the composed signal. An exemplary algorithm of this kind is presented in Fig. 1(a), and the process of extracting the parasite impulse is illustrated in Fig. 1(b)–(f). The wavelet transform is particularly applicable to the task mentioned because of its good resolution in the upper part of the spectral representation.

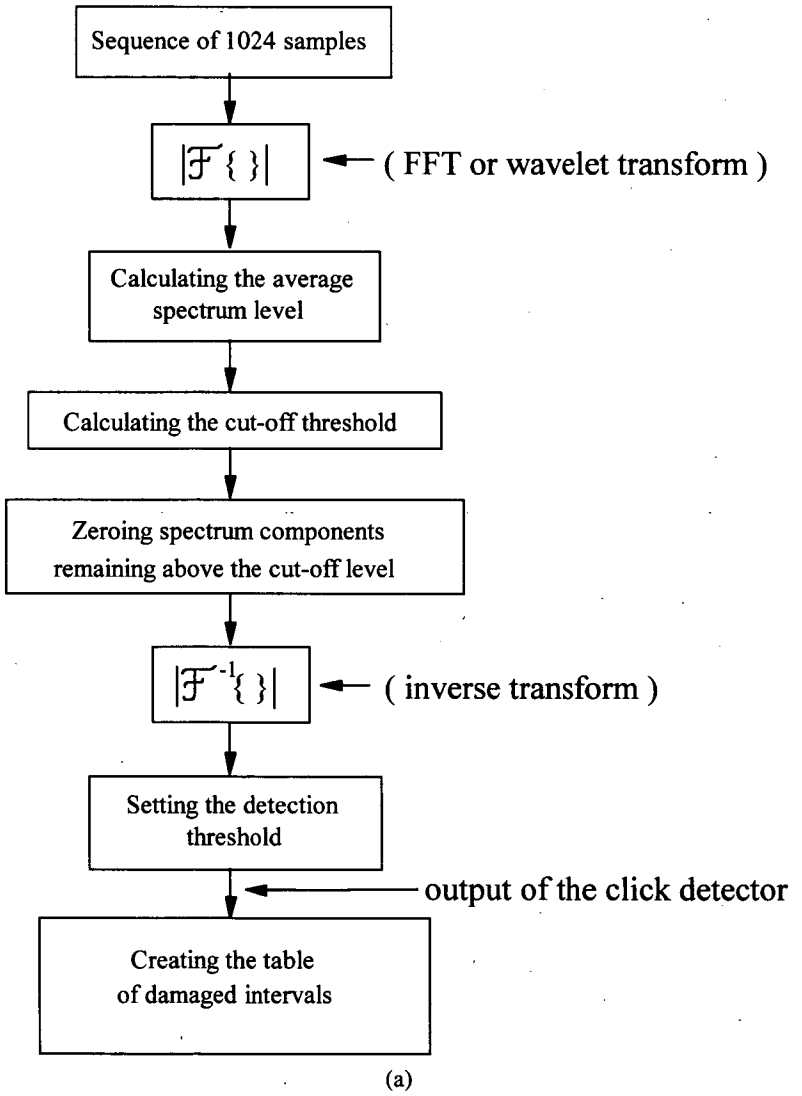
These methods have two main drawbacks:

1) Only short impulses produce spectral events that are easily discernible from the original signal.

2) Transients, percussive sounds, and other desirable impulsive effects may be treated as disturbances by the algorithm.

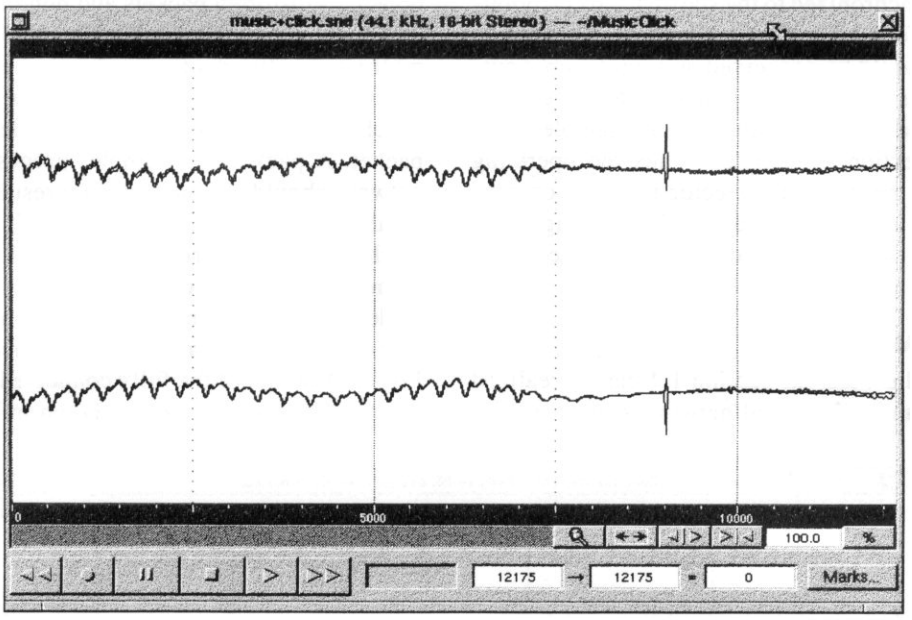
Consequently there is a need to employ learning algorithms for the processing of audio signal, allowing one to build a knowledge base of both impulsive disturbances and desirable signal properties.

In general the detection procedure should allow for the exact marking of the starting point and the end point of any time interval containing samples distorted by the undesired impulse. The detector output can be defined in such a way that the presence of the parasite impulse inside the observed sample packet results in the appearance of the value $+1$ at the output, and the nondistorted signal is represented by the value -1 at this output.

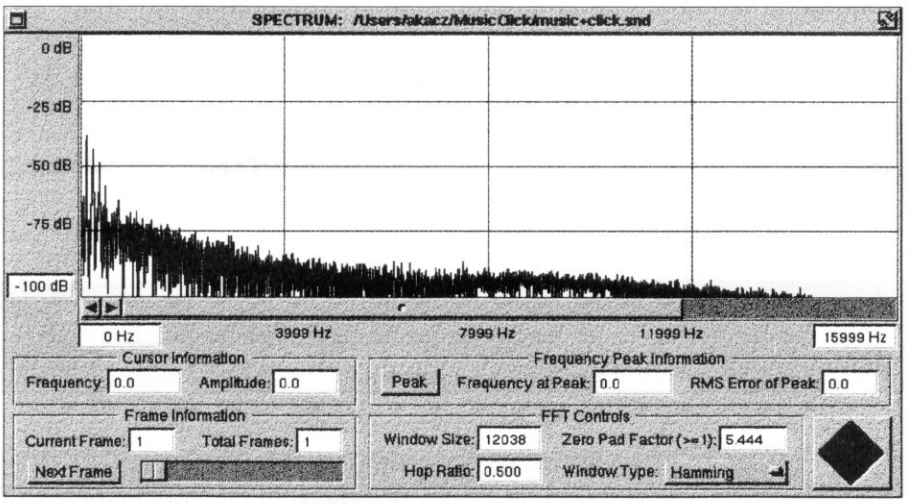


(a)

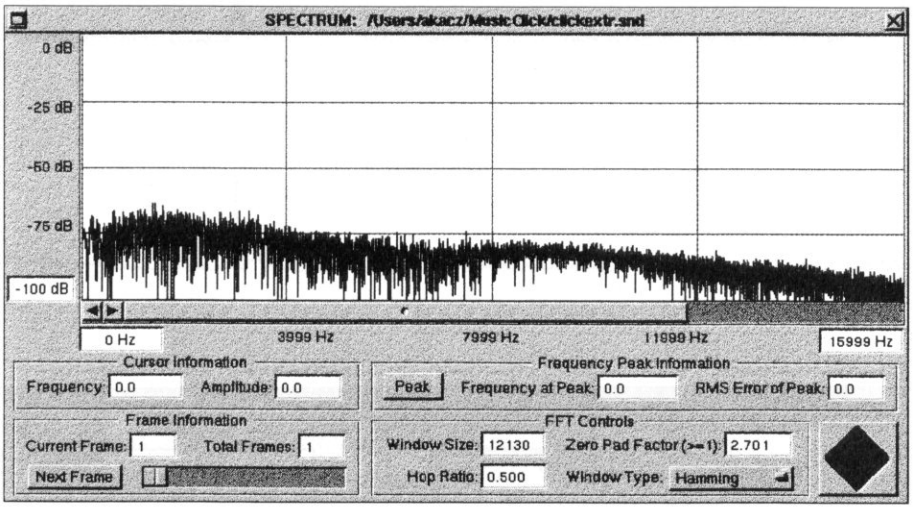
Fig. 1. Exemplary orthogonal transform-based filtration method for detection of clicks. (a) Threshold algorithm. (b) Stereo signal fragment with clicks. (c) Spectrum of distorted signal (both channels mixed). (d) Result of cut-off operation in spectrum domain. (e) Signal after inverse transform. (f) Signal in output of click detector.



(b)



(c)



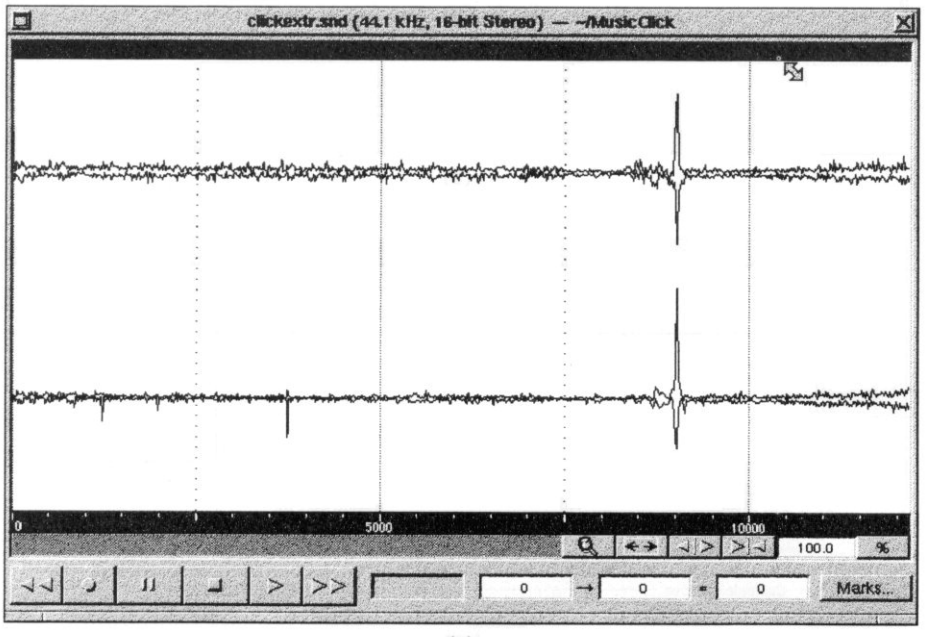
(d)

Fig. 1. continued

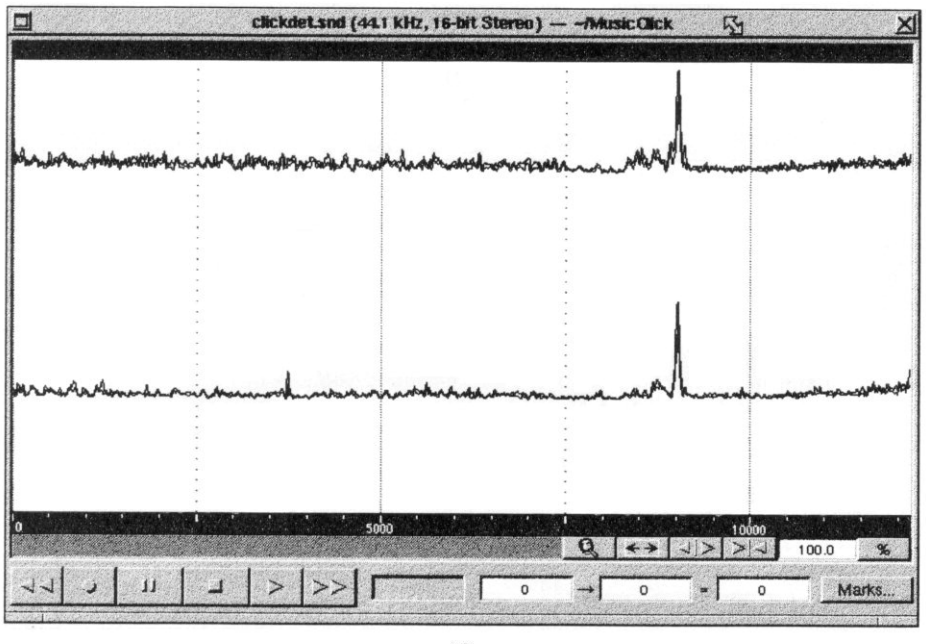
Consequently a square wave is generated by the detector, having slopes synchronized to the starting and end points of the packets affected by impulse distortions. The detector mentioned should be conceived as a multi-input system, because it is not possible to assess the presence of a click on the basis of the value of one sample or an insufficient collection of samples. Thus the operating principle of this multi-input detector is based on processing a packet of consecutive samples moving along its inputs. Practical details concerning this concept are presented in the next section.

1.2 Learning Algorithm

The detector discussed in Section 1.1 can be realized on the basis of a simple neural network structure (perceptron algorithm) trained to recognize two classes of objects—distorted packets and nondistorted ones. Consequently it is sufficient that the output layer of such a multilayer network consist of a single neuron providing the final decision element. The neural network output is synchronized to the central sample of the restored packet. A packet of n samples processed by the neural network should produce a single result consisting of the symbol -1 or $+1$. A series of k such packets processed by the neural network will produce a rectangular waveform at its output. Consequently the neurodetector may be defined as the operation $\mathrm{ND}\{x(n)\}$ implemented for the task of detection of impulse disturbances, as presented in Fig. 2. A layout of the fully connected feedforward neural network (perceptron) is shown in Fig. 3.



(e)



(f)

Fig. 1. continued

The hyperbolic tangent function was selected as the transfer function of the neurons. Thus the output state y of the jth neuron is determined by the relationship

$$y = \tanh(\Sigma_i w_i x_i + b_j) \quad (1)$$

where

x_i = input signal on ith synapse
w_i = weight of ith synapse
b_j = bias of jth neuron.

The network is characterized by the status vector defined as

$$S = [w_1, w_2, \ldots, w_s, b_1, b_2, \ldots, b_r] \quad (2)$$

where

S = status vector, describing state of network
$w_1, \ldots, w_s$ = consecutive synapse weights
$b_1, \ldots, b_r$ = neuron biases
s = number of synapses
r = number of neurons.

The neural network training procedure involves examples chosen from both classes—desired signal patterns and disturbing impulses. This assumption allows one to formulate the equation describing the mean-square error that is to be minimized during the training phase,

$$E = \frac{1}{m}\sum_{i=1}^{m}(y_i - z_1)^2 + \frac{1}{n}\sum_{j=1}^{n}(y_j - z_2)^2 \quad (3)$$

where

m = number of examples for class 1, representing nondistorted signal
n = number of examples for class 2, representing distorted signal
y_i = output of network for ith example of class 1
y_j = output of network for jth example of class 2
z_1 = expected state of network output for signal belonging to class 1 ($z_1 = -1$)
z_2 = expected state of network for signal belonging to class 2 ($z_2 = +101$).

As all examples representing both classes are presented to the network during the training, the status vector is known for each example. It is thus possible to introduce the following notation:

$$y_i = y_i(S), \quad y_j = y_j(S), \quad E = E(S). \quad (4)$$

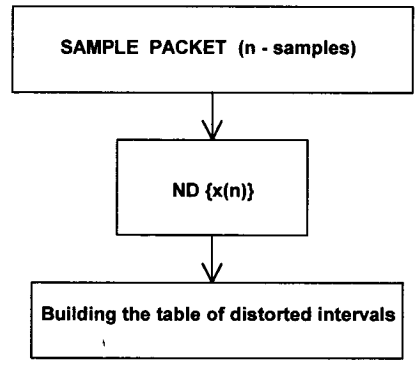
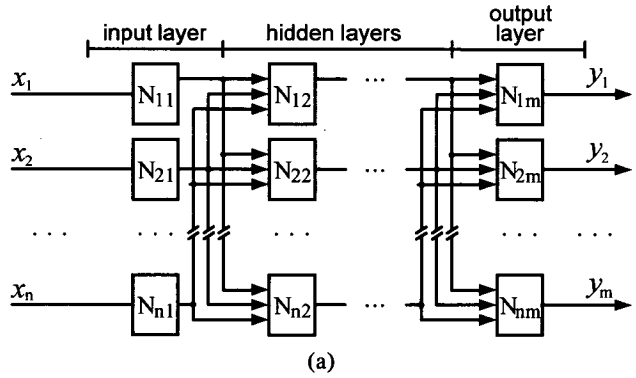
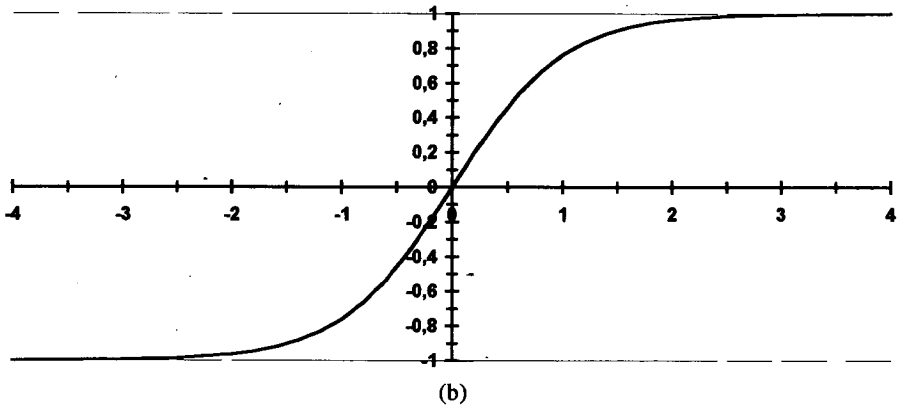


Fig. 2. Neural network as a detector of distorted samples.



(a)



(b)

Fig. 3. Neural network structure. (a) Layout of perceptron. (b) Shape of transfer function.

The procedure for minimizing the error defined by Eq. (3) is based on the modified error back-propagation method [9], ensuring the minimization of half of the norm of error E versus all weights and biases. Consequently synaptic weight values are to be corrected in consecutive iterations according to the error gradient minimization procedure. The gradient of the error is defined as

$$\nabla E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \ldots, \frac{\partial E}{\partial w_s}, \frac{\partial E}{\partial b_1}, \frac{\partial E}{\partial b_2}, \ldots, \frac{\partial E}{\partial b_r}\right]. \tag{5}$$

In practice, the following increments are to be computed instead of the derivatives:

$$\frac{\Delta E}{\Delta w_i}, \frac{\Delta E}{\Delta b_j}, \quad i = 1, \ldots, s; \quad j = 1, \ldots, r. \tag{6}$$

After initializing the procedure (step 0), values of Δw_i and Δb_j are set to a small value Δ_0 selected empirically,

$$\Delta w_1 = \Delta w_2 = \cdots = \Delta w_s = \Delta b_1 = \Delta b_2 = \cdots = \Delta b_r = \Delta_0. \tag{7}$$

In the steps that follow (step 1, step 2, . . . , step n) Δw_i and Δb_j are set to values Δ_n such that

$$\Delta_n = \beta \cdot \Delta_{n-1} \tag{8}$$

where $\beta < 1$ is a small number selected arbitrarily.

The increments diminish progressively due to being multiplied by the coefficient β. Hence the precision of the error minimization procedure grows after each iteration step. As a result of synaptic weight updates, a new status of the network will be reached that fulfills the following term:

$$E(S_n) < E(S_{n-1}) \tag{9}$$

where S_n is the network status vector for step n.

The new values of the status vectors are calculated on the basis of the rule

$$S_n = S_{n-1} - \eta \cdot \nabla E \tag{10}$$

where η is a coefficient that is calculated as follows:

$$\eta = k \cdot \Delta_{n-1} \tag{11}$$

k being the biggest integer fulfilling the following term:

$$E(S_{n-1} - k \cdot \Delta_{n-1} \cdot \nabla E) < E(S_{n-1} - (k-1) \cdot \Delta_{n-1} \cdot \nabla E). \tag{12}$$

The largest integer k fulfilling the relationship (12) is used for the calculation of the final status vector S_n for the nth iteration step on the basis of Eqs. (10) and (11). It was found experimentally by the author that the appropriate values of parameters Δ_0 and β allowing to optimize this back-propagation algorithm are $\Delta_0 = 10^{-4}$ and $\beta = 0.87$.

Another neural network may be applied to the detection of impulsive disturbances, namely, the predictive neural network, which is described in Section 2. The main purpose of this network is to recover missing samples.

2 RESTORATION OF LOST SAMPLES

Once the intervals containing distorted samples are detected and their boundaries marked, the main process of sound restoration is applied to replace these samples. Many methods of extrapolation and interpolation of irrevocably lost samples are known from the literature. Some of these techniques are

- Interpolation of zeroth order (this technique does not ensure continuity of the signal at the interpolated fragment boundaries), of first order (continuity of the first derivative is not ensured), and of higher orders
- Linear prediction (LP), that is, estimation of forward samples based on the linear combination of k preceding samples or the estimation of backward samples based on the linear combination of k subsequent samples
- Polynomial extrapolation of samples lost in the vicinity of gaps (operating in the frequency domain) [4]
- Others techniques [1], [3].

Most of the interpolation methods enable the recovery of missing samples only when the nondistorted neighbor samples exist at the boundaries of the distorted interval. For example, the LP-based extrapolation algorithm demands that k adjacent samples not be affected by impulsive disturbances, where k is the LP order. Meanwhile, because of the concept of the direct neural network implementation for the processing of signal samples, the possibility occurred to perform a kind of nonlinear prediction realized by the neural network. This concept is discussed next.

2.1 Neuropredictor

The process of recovering distorted samples can be realized with the use of a neural network algorithm. Actually in this case the perceptron algorithm provides a kind of nonlinear prediction of the signal. This type of algorithm performs the processing of consecutive samples in order to generate a subsequent sample in the case of forward prediction or to generate a preceding sample in the case of backward prediction of lost samples. The computed sample is joined with the existing signal samples. Then the algorithm does the next step and the procedure is repeated. In this case the neural network output is synchronized with the samples from the outside of the restored packet. The concept of such

an algorithm is illustrated in Fig. 4.

Denoting the neuropredictor function by $x_n = \mathrm{NP}^+\{x, r\}$, performing nonlinear prediction of samples x_i on the basis of r preceding samples $x_{i-1}, x_{i-2}, \ldots, x_{i-r}$, with r being the prediction order equal to the number of neural network inputs, one can predict the series of n forward samples on the basis of the following procedure:

$$x_n = \mathrm{NP}^+\{x, r\} \tag{13}$$

where $n = j, j + 1, j + 2, \ldots, k$, with j and k being indexes of distorted interval neighbor samples.

Similarly, for the case of the backward prediction, a series of samples y_n is produced,

$$y_n = \mathrm{NP}^-\{x, r\} . \tag{14}$$

The interpolation procedure employing the neural network may ensure the accuracy demanded, provided a sufficient number of examples extracted from the nondistorted signal was used for the training. The training pattern consists of a certain number of samples equal to the number of network inputs and of the actual value of the output sample to be estimated. This time, the collected examples are not divided into classes because this neural network is not supposed to make any classifications. Correspondingly, the number of classes may be considered as equal to 1. The error function E to be minimized at the training stage can be defined as

$$E = \sum_{i=1}^{n} (y_i - p_i)^2 \tag{15}$$

where

n = number of examples used
y_i = output state of neural network for ith example
p_i = value of training sample of ith example.

The procedure for the minimization of the error [Eq. (15)] is identical to that described in Section 1.2. This procedure allows building the knowledge of the signal in the network automatically. Consequently the signal estimation is expected to be of increasing accuracy after each training run.

The next problem to be solved is the proper selection of training patterns. However, this task may be eased because of the predictive capabilities of the neural network. Once this network serves as the (nonlinear) predictor, it is possible to generate the prediction error at its output. This prediction error signal is then observed while the nondistorted signal packets are processed by the neural network algorithm. Theoretically the prediction error for the network having full knowledge of the processed signal should be equal to zero. Consequently the process of the selection of examples for the training may be assisted by the observation of the error value. Hence examples of nondistorted signals causing possibly high prediction errors should be employed in the neural network training. Repetitions of this procedure normally cause decreasing error and at the same time increasing accuracy of the signal estimation.

As results indicate, the neural network realizing the nonlinear prediction of sample values may be applied for the interpolation of distorted signal portions.

2.2 Program Implementing Neural Networks

A computer model of a fully connected feedforward neural network was implemented in order to experiment with the removal of impulse distortions. As a result of experiments performed by the author, the hyperbolic tangent function [Fig. 3(b)] proved to be the most efficient transfer function of the neurons. The number of layers was selected optionally in the range of 3 to 5. The modified back-propagation algorithm was used for the learning procedure, based on the minimization of half of the norm of error E against all synaptic weights and neuron biases, as described in Section 1.2.

The program SigNet.app was prepared for a UNIX computer workstation (NeXTStep operation system, 33-

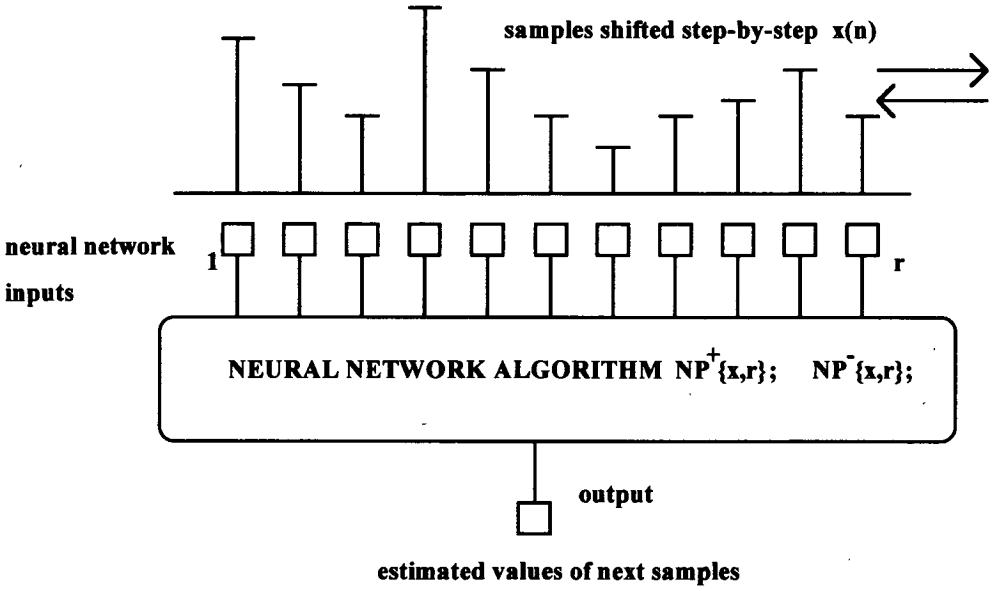


Fig. 4. Illustration of concept of neuropredictor.

MHz clock speed). The user interface panel designed for the training process is presented in Fig. 5, and the panel for the restoration process is shown in Fig. 6.

The panel Training (see Fig. 5) is used for loading and saving neural network configurations. The network may be designed for the recognition of clicks or for the restoration of distorted samples. Examples chosen for the network training are listed in the window Examples. Each example is designated as Clean or Click by the operator. The fields marked Error limit, Output error limit, and Steps limit enable the control of parameters related to the training procedure. The field Progress View shows the progress of the training on the basis of analyzing the neural network output error. In the center of the screen there are fields for monitoring the current behavior of the mean-square error. In the lower right corner of the screen the program log is displayed, presenting information about the progress of the training phase. The lower part of the window presents the plot of the error as a function of the iteration step number.

The panel Restoration (see Fig. 6) consists of two parts—the Neural Network field used for loading network structures (left-hand part of the window) and the Restoration operation field used for control and monitoring of the sound restoration process (right-hand part of the window). The first neural network that is loaded using this program interface (see upper left of Fig. 6) was trained to detect lost intervals, while the second one (lower left of Fig. 6) was prepared to predict missing samples.

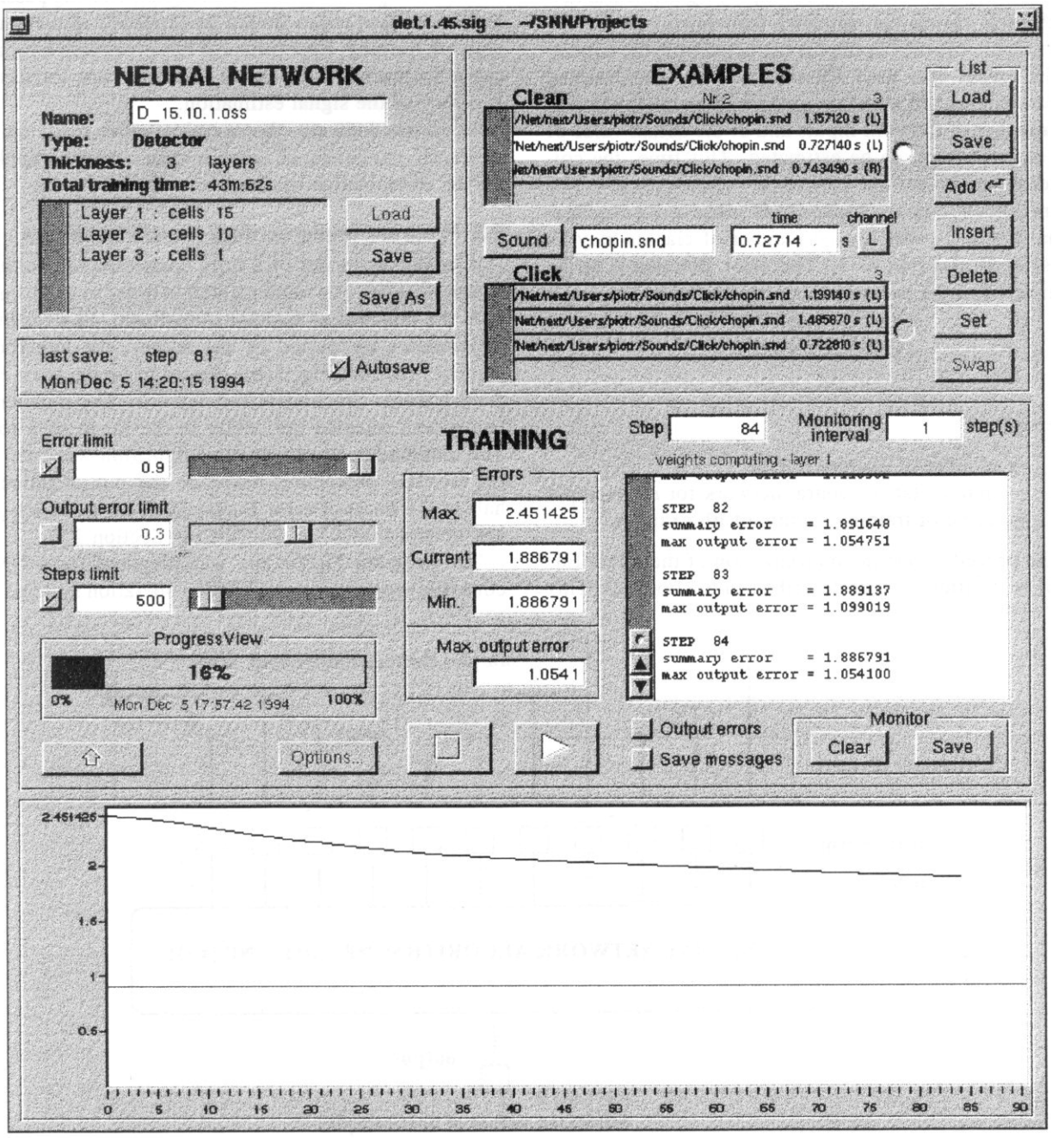


Fig. 5. Training window of program SigNet.app.

3 EXPERIMENTS WITH NEURAL NETWORKS

3.1 Training Neural Networks

Patterns used to train the learning algorithms were extracted from old records. In some recent experiments signals were also investigated which were affected by parasite impulses during transmission through various types of telecommunication channels. It was necessary to employ both examples of distortion-free sounds and examples of the undesired impulse disturbances. Patterns of clean sound teach the learning algorithms how to avoid misleading classifications in the case of musical signal events having an impulselike character.

A vinyl recording of Chopin's Mazurka performed by the Polish pianist Karol Malcuzyński was one of the subjects of the experiments. The record has a good quality of sound except for the presence of strong clicks caused by its frequent use. The patterns for training were sampled at 44.1 kHz with 16-bit resolution (monophonic).

The first task was to investigate the speed and efficiency of the training of various neural network structures. Consequently, the network structures were defined consisting of various numbers of neurons in the input layer and differentiated numbers of hidden layers. The resulting structures of type V, U, O, and A are shown in Fig. 7. All these structures were trained using the same set of examples. The plots in Fig. 8 illustrate the progress of the training of these structures.

The structures of type A often did not converge properly. The output error remained large or even grew during the training. As Fig. 8 shows, the neural network structures reveal the following character of error decrease:

- *Phase 1*: Large value of error (around 1). The number of iterations necessary to pass this phase may be different for various network structures (from 2 to as many as 66 000 iteration steps).
- *Phase 2*: Fast decrease of error (to values around 0.5). This phase is passed reasonably quickly (after 200–300 iteration steps).
- *Phase 3*: Again a slower decrease of error. In this phase the error function plot can be quite complex because the error may decrease nonsmoothly.
- *Phase 4*: Long and very slow decrease of error. The plot of the error function is approaching a certain value asymptotically. In this phase some local fluctuations can also occur, appearing in the form of "impulses" in some error function plots.

Several structures of neural networks were trained during the experiments. The structures investigated are listed in Table 1. In this table, neural network structures are denoted symbolically. For example, d5v2 denotes a detector structure of type V having an input layer consisting of 5 neurons and 2 hidden layers. The number of neurons in all layers is also shown in Table 1 for all

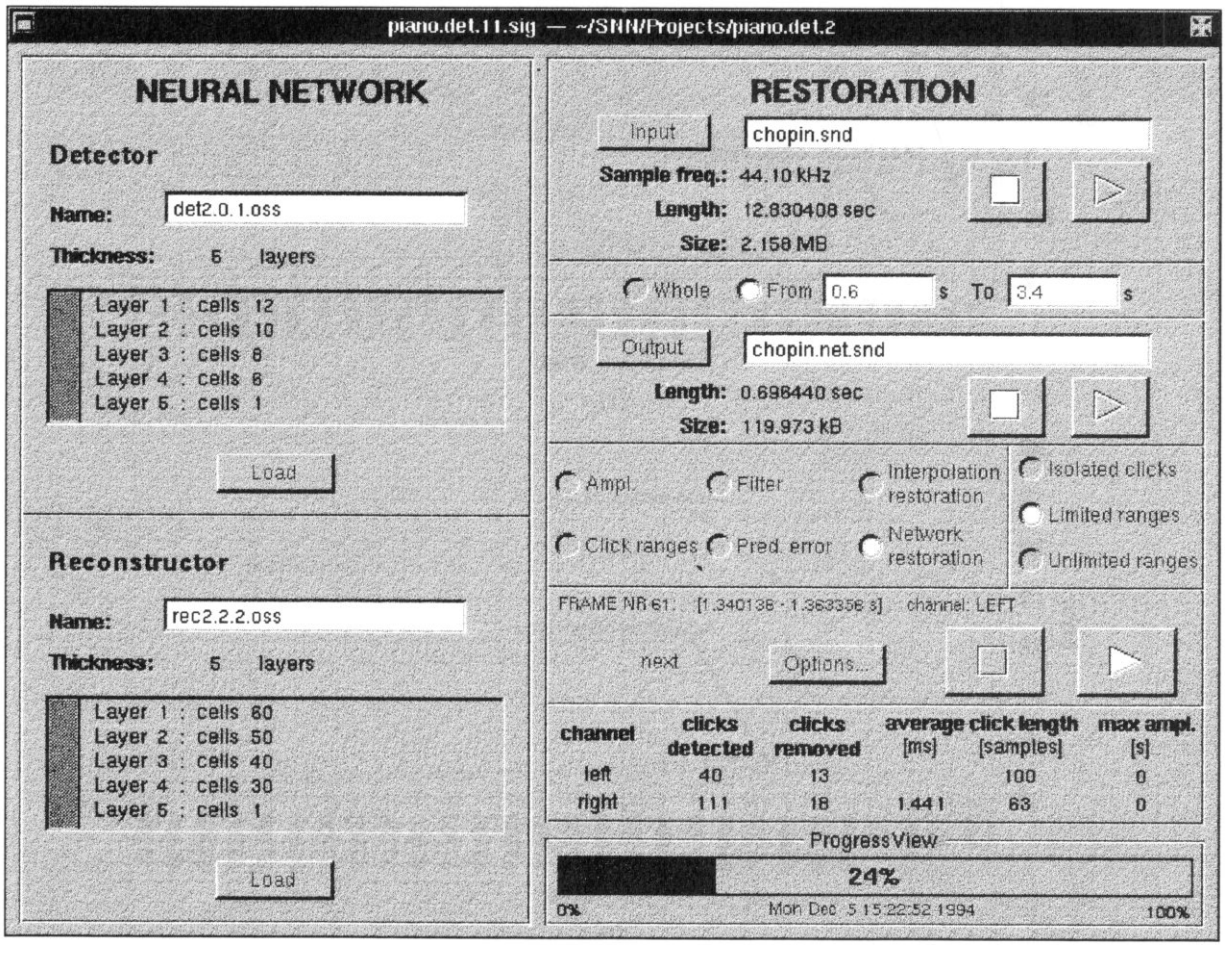


Fig. 6. Restoration window of program SigNet.app.

structures investigated. The task was to choose those structures that will converge most efficiently. The experiments were carried out in a systematic way. Initially 10 training patterns containing clicks and clean portions of some 5-s fragments of music were used. Plots of the error functions are shown in Fig. 9 in such a way that it allows one to compare directly various neural networks having the same number of inputs and hidden layers. Moreover, a comparison of the characteristics shown in the consecutive drawings of Fig. 9(a)–(d) permits studying the influence of the number of inputs and hidden layers on the efficiency of the training progress. In the analysis of the results such factors as the speed of the mean-square error decrease and the smoothness of the error function were considered. However, it must also be kept in mind that the larger the structure of the network, the more time is needed for computations related to each iteration. The speed of training for the various structures can be assessed on the basis of the histogram presented in Fig. 10.

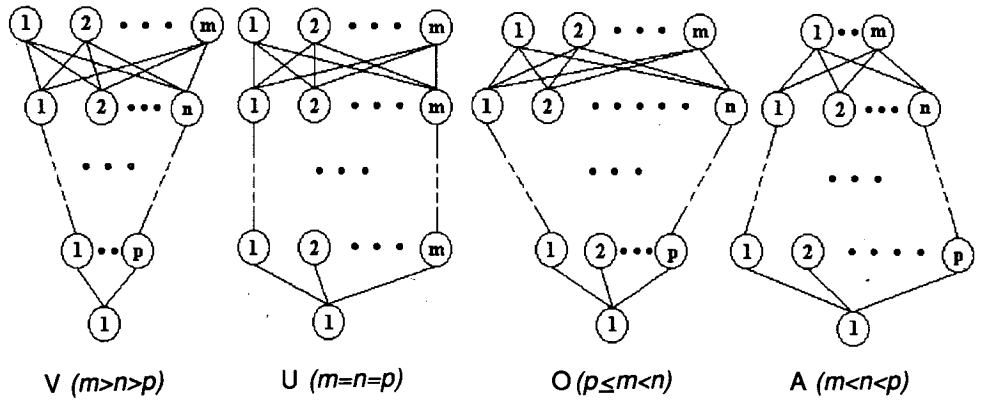
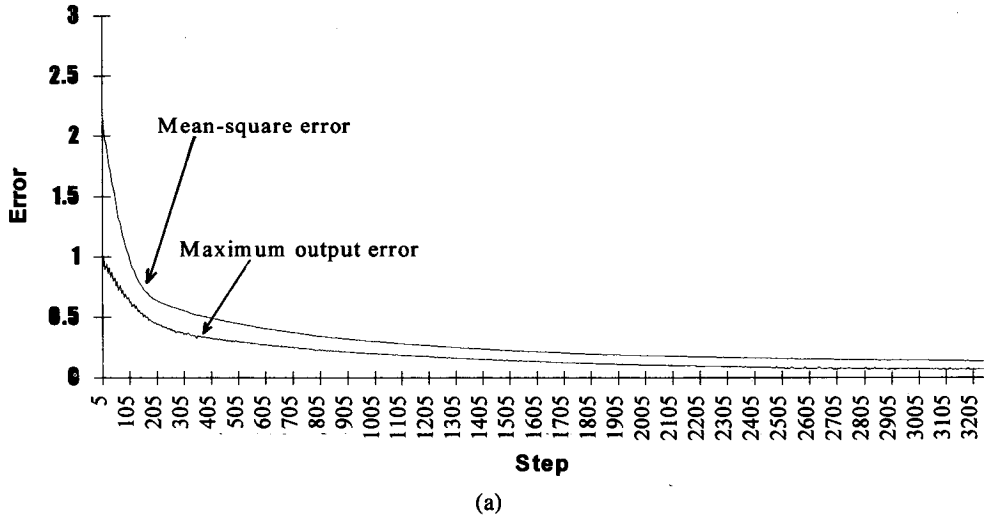
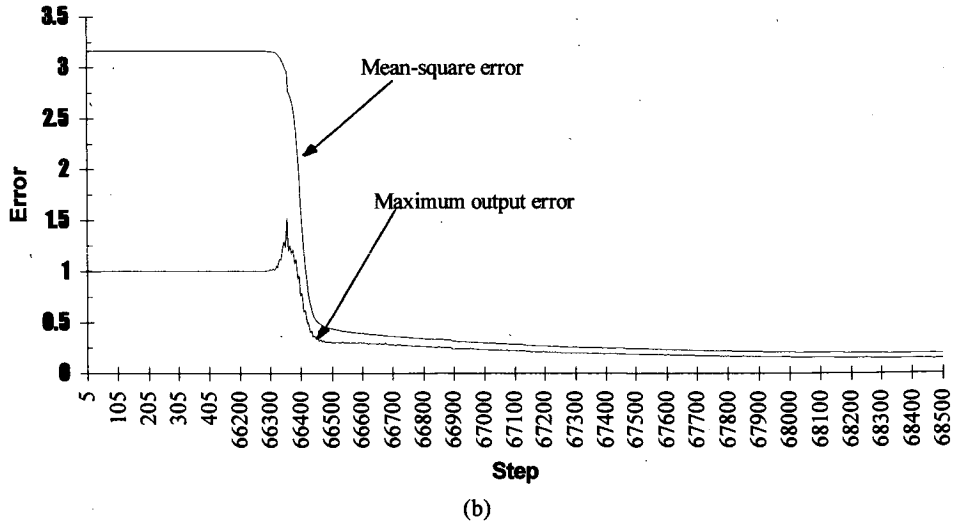


Fig. 7. Structures of neural networks investigated.



(a)



(b)

Fig. 8. Progress of training of exemplary neural networks. (a) Type U structure, 21 neurons. (b) Type A structure, 27 neurons.

From a detailed analysis of the results obtained it can be concluded that a V-type structure can be trained most efficiently. At the same time, the structure having two hidden layers is sufficiently complex to be able to identify clicks. The compromise between the speed of error decrease and the time necessary to complete each iteration leads to the selection of the structure d15v2 as one of the most appropriate to detect clicks in audio material.

In addition it was checked how the number of training patterns affects the speed of error decrease. As can be seen from Table 2, some experiments were organized using 10, 20, and 40 sound patterns. The data show that too small or too large a set of patterns can cause an increase in training time. It was also found that a too small a number of training patterns can cause a nonmonotonically decreasing error. In turn, in the case of too many patterns the network may not converge in a reasonable time.

Table 2 shows that the computation time using a single UNIX workstation is generally high. The process can be sped up by using computers connected via a network. This parallel processing feature is fully supported by the software described.

Following the training process, the entire recording was processed using the impulse detection neural network algorithm. Several hundred distorted blocks were found in the every minute of the recording. The impulse detection process is many times faster than the training procedures. Nevertheless, building the table of distorted intervals is not realized in real time on a typical workstation—it may require several times the duration of the original recording.

3.2 Restoring Lost Intervals

Procedures similar to those described were performed with a second kind of algorithm—the predictive neural network. On the basis of investigations similar to those described in Section 3.1 it was found empirically that the appropriate configuration of such a network is as follows: 60.50.40.30.1 (60 input neurons, 3 hidden layers containing 50, 40, and 30 neurons, and 1 neuron in the output layer).

The predictive algorithm was trained using only patterns of clean sound (not spoiled by parasite impulses). The concept of this algorithm is the same as that described in Section 2.1. The network learned how to predict the next sample on the basis of the 60 preceding samples currently present at its inputs. The algorithm is designed in such a way that each example was processed both forward and backward, so the network could learn how to predict missing samples on the basis of its left and right neighbor samples. Theoretically the network should process the whole nondistorted audio material in order to build a knowledge of all available signals. However, such a task would be much too arduous and even not desired, because a relatively small "overtrained" network may not be able to acquire sufficient knowledge or may lose its generalization capabilities. Consequently the program was conceived in such a way that the material for the training is automatically edited from the neighborhood of the distorted intervals detected earlier. This feature makes the process of training faster and ensures that the network is gaining knowledge of those portions of signal that are of similar character to the lost fragments to be restored on the basis of prediction. Obviously, the training material for the predictive algorithm should not contain parasite impulses. Therefore in the case of the occurrence of a series of short clicks, the interval to be restored should cover all short gaps. Otherwise it would be impossible to find a sufficient portion of samples between individual distorted intervals to feed the neural network inputs. This requires that automatic "gluing" of lost intervals be performed. This procedure is executed anytime, provided there are less than 60 nondistorted samples at the boundaries of the lost interval. (A 60-input predictive neural network

Table 1. Structures of neural networks examined.

5 Inputs	7 Inputs	9 Inputs	11 Inputs	15 Inputs	21 Inputs
d5v2 5/4/2/1	d7v2 7/5/3/1	d9v2 9/7/5/1	d11v2 11/8/5/1	d15v2 15/11/7/1	d21v2 21/17/11/1
d5u2 5/5/5/1	d7u2 7/7/7/1	d9u2 9/9/9/1	d11u2 11/11/11/1	d15u2 15/15/15/1	
d5o2 5/7/5/1	d7o2 7/10/7/1	d9o2 9/12/9/1	d11o2 11/15/11/1	d15o2 15/18/15/1	
d5a2 5/6/7/1	d7a2 7/9/11/1	d9a2 9/11/13/1	d11a2 11/13/15/1	d15a2 15/17/19/1	
d5v3 5/4/3/2/1	d7v3 7/5/4/3/1	d9v3 9/7/5/3/1	d11v3 11/9/7/5/1		
d5u3 5/5/5/5/1	d7u3 7/7/7/7/1	d9u3 9/9/9/9/1	d11u3 11/11/11/11/1		
d5o3 5/9/7/5/1	d7o3 7/11/9/7/1	d9o3 9/15/12/9/1	d11o3 11/17/14/11/1		
d5a3 5/7/9/11/1	d7a3 7/9/11/13/1	d9a3 9/11/13/15/1	d11a3 11/13/15/17/1		

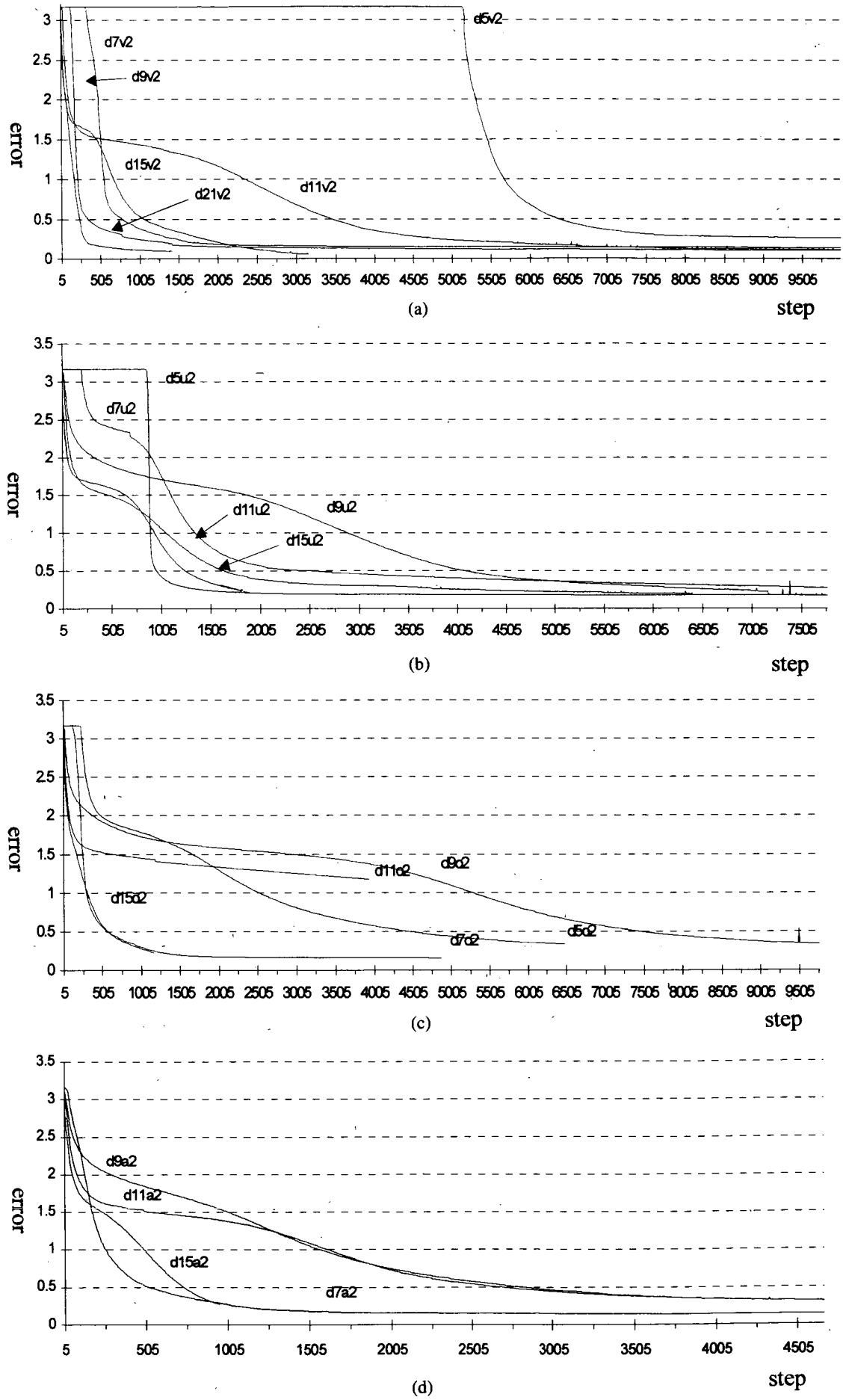


Fig. 9. Progress of training of some selected structures of neural networks (decrease of mean-square error). (a) Type V structures. (b) Type U structures. (c) Type O structures. (d) Type A structures.

was used.) Thus the series of distorted intervals may be restored by linking them into one longer block and applying the procedures to the whole block. Some results of the interpolation of a large block of missing samples are presented in the Section 4.

The program interface designed for the control of the restoration process was previously presented in Fig. 6. The restoration algorithm is executed in such a way that impulse distortions are automatically detected by the first neural network, the table of lost intervals is built up, the procedure of gluing a series of short intervals is executed, and subsequently the second neural network algorithm "slides" along the samples in the neighborhood of the lost intervals, producing forward and backward streams of predicted samples. Finally the overmixing procedures described in the next section complete the signal restoration task.

3.3 Overmixing Recovered Intervals

Independently of the method applied to forward and backward interpolation of the signal, it is essential to implement overmixing techniques which can help improve the subjective quality of the interpolated sound. First a linear weighted summation method was performed according to the rule

$$z_n = \frac{(k-n)x_n + (n-j)y_n}{k-j} \tag{16}$$

where

$\langle j, k\rangle$ = interval to be restored
x_n = series of forward estimated samples, $n = j, j+1, \ldots, k$
y_n = series of backward estimated samples
z_n = series of resulting samples.

The linear weighted summation of the forward and backward series of estimated samples does not ensure the continuity of the first derivative of the signal. Thus a second summation algorithm was performed according to the equation

$$z_n = 0.5\left\{\left[1 + \cos\left(\frac{n-j}{k-j}\pi\right)\right]x_n + \left[1 - \cos\left(\frac{n-j}{k-j}\pi\right)\right]y_n\right\} \tag{17}$$

where $n = j, j+1, j+2, \ldots, k$. The function is plotted in Fig. 11.

Finally, a third procedure for overmixing sample series, which may be called antisymmetrical gluing, was performed as follows:

$$x_n = 2x_{j-1} - x_{2j-n-2}$$
$$y_n = 2x_{k+1} - x_{2k-n+2} \qquad n = j, j+1, \ldots, k. \tag{18}$$

Table 2. Decrease in error versus number of patterns used for training for structure V, 15 inputs, 2 hidden layers.

Number of Patterns	Mean-Square Error	Maximum Output Error	Computation Time
10	0.25	0.24376	4 h 24 min
20	0.25	0.21217	3 h 47 min
40	0.9474	0.89450	10 h 00 min

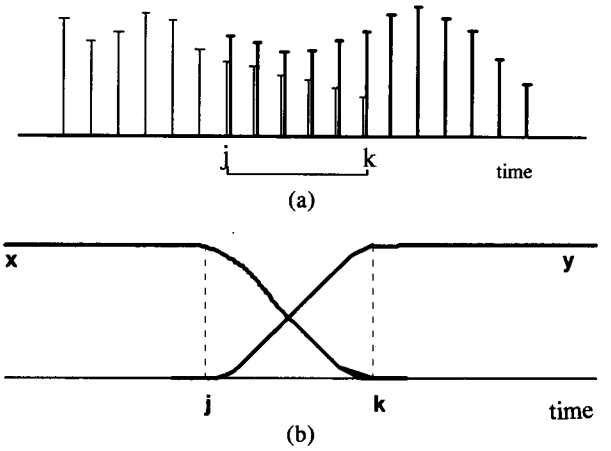


Fig. 11. Illustration of overmixing procedure linking samples extrapolated forward and backward. (a) Overmixed samples. (b) Nonlinear weighted summation according to Eq. (17).

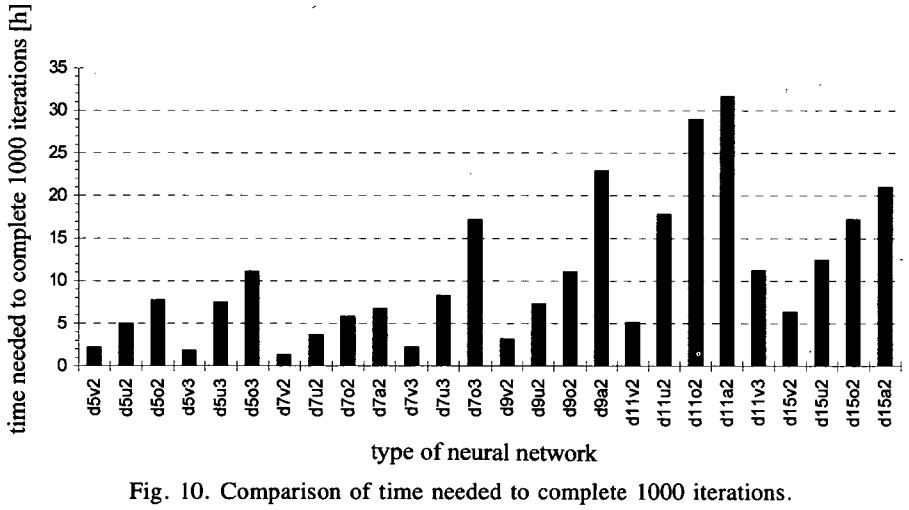


Fig. 10. Comparison of time needed to complete 1000 iterations.

Informal listening tests showed that the overmixing procedure according to Eq. (17) proved to be more effective than procedures executed according to Eqs. (16) and (18).

4 RESULTS

The detection of distorted intervals was performed by the detective neural network algorithm. As can be seen in Fig. 12(a)–(c), a properly trained neural detector is able to discern normal impulsive sounds from the parasite impulses, while the wavelet-based threshold procedure detects impulses independently of their origin and consequently incorrectly identifies several normal impulsive signals as artifacts to be removed. For example, consider the middle part of Fig. 12(a), which represents a strong transient. Since the neural network was trained using musical patterns containing such transient states, the network does not qualify this fragment as a parasite impulse [Fig. 12(c)], while the threshold algorithm [Fig.

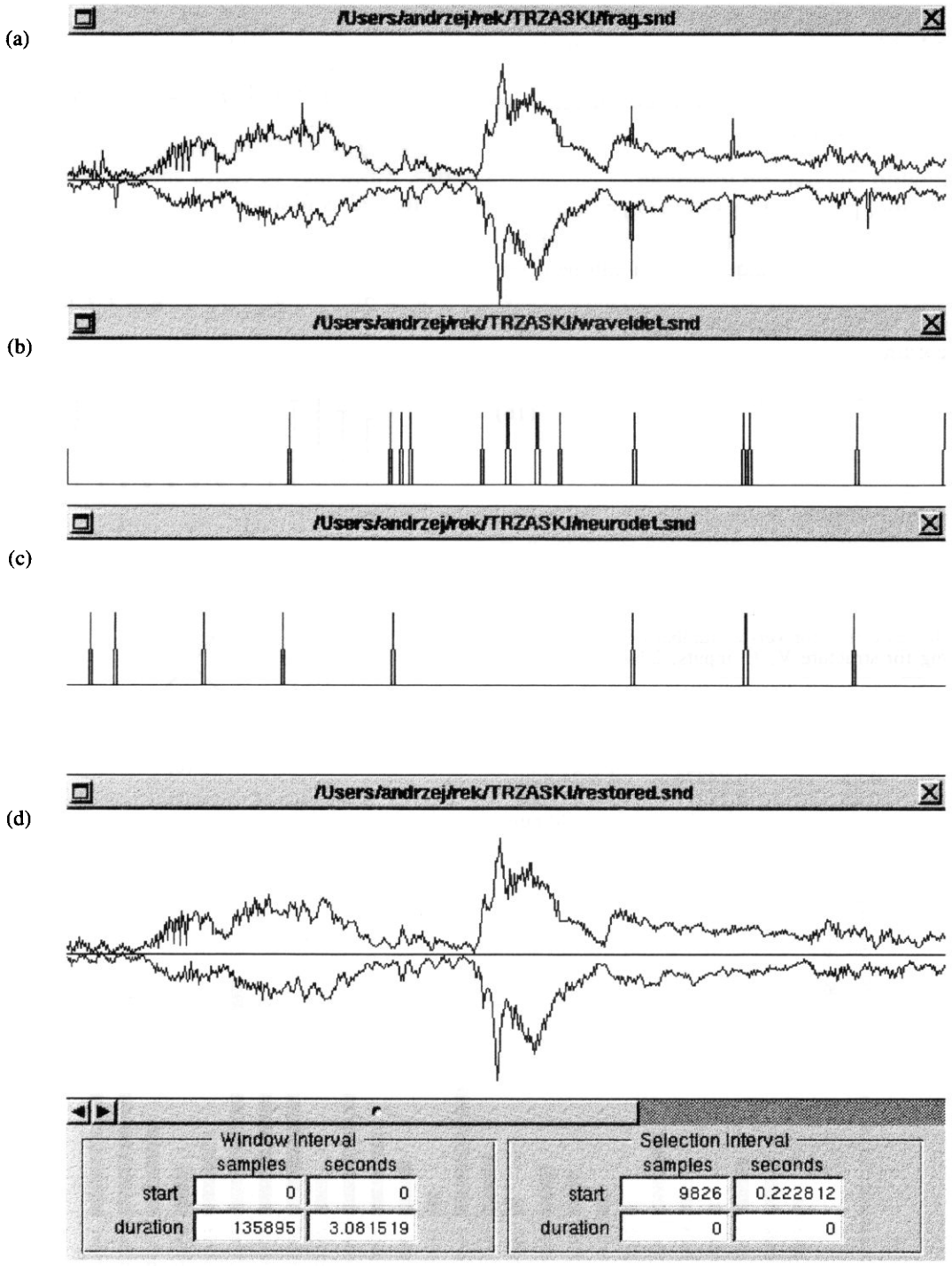


Fig. 12. Results of impulse detection and interpolation. (a) Signal affected by clicks. (b) Clicks detected by wavelet threshold procedure. (c) Clicks extracted by neurodetector. (d) Signal restored by neuropredictor algorithm; experiment described in Section 4.

12(b)] reacts to each impulsive event. Moreover, the neural detector is capable of finding some impulses hidden below the signal envelope which remain undetected by the threshold algorithm.

In order to evaluate the subjective effectiveness of several restoration schemes, signal reconstruction was performed based on the following methods:

- *Test 1*: Zero-order interpolation
- *Test 2*: Second-order interpolation
- *Test 3*: 100th-order forward and backward LP interpolation with nonlinear overmixing procedure [Eq. (17)]
- *Test 4*: Interpolation based on the wavelet threshold procedure (as in Fig. 1) and filling of the gap with copy of neighboring fragment
- *Test 5*: Interpolation based on predictive neural network algorithm with nonlinear overmixing [Eq. (17)].

Some results of processing the signal by means of these methods are illustrated in Figs. 12(d) and 13. The result in Fig. 12(d) was obtained using the neuropredictor algorithm with overmixing the forward and backward samples [Eq. (17)]. In order to compare the effectiveness of various interpolation methods, an artificial gap was intentionally made in the signal fragment plotted in Fig. 13(a). Subsequently the gap was interpolated using the various techniques, as shown in Fig. 13(b)–(f).

For the subjective assessment, a session of paired comparison tests was organized employing five subjects

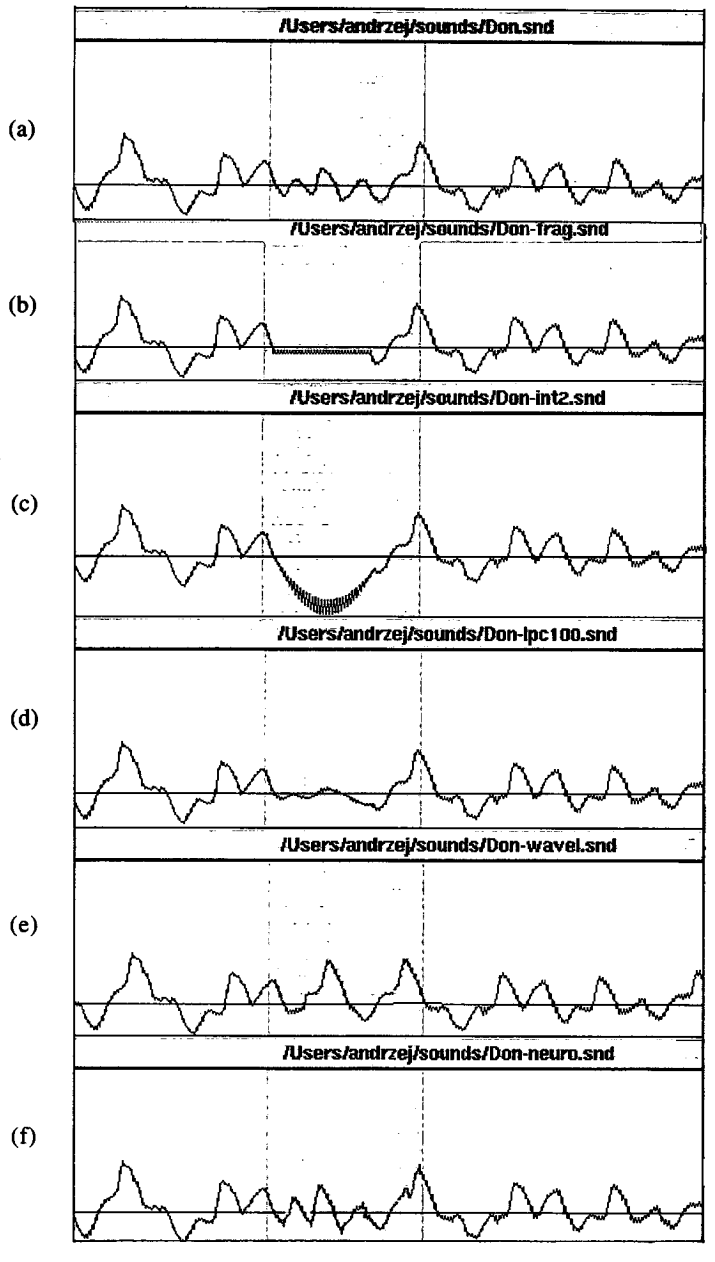


Fig. 13. Zoomed results of interpolation of missing samples utilizing various methods. (a) Original signal. (b) Intentionally made signal gap (interpolated with zeroth-order linear interpolation). (c) Interpolation of second order. (d) 100th-order overmixed backward and forward LP interpolation. (e) Wavelet detection of lost interval and result of automatic filling of gap with copy of neighboring fragment. (f) Interpolation with neural prediction and nonlinear overmixing of forward and backward interpolated samples.

recruited from the researchers of the Sound Engineering Department of the Technical University of Gdańsk. Sound patterns were edited and paired according to the rules of paired comparison tests [10]. The pairs of signals to be compared were constructed on the basis of five processed patterns from the same 10-s music portion. The need to compare five audio fragments leads to the creation of $\begin{bmatrix} 5 \\ 2 \end{bmatrix} = 10$ independent pairs. The total number of results of the assessments is equal to 10 pairs × 5 subjects = 50. A special computer program was developed to assist in the processing of the test results. This program allows checking the reliability of the results of the subjective tests by the χ^2 test (Pearson test) [10]. The results obtained revealed the supremacy of the proposed neural predictor method (test 5) over the other interpolation methods. The results of the subjective tests are plotted in Fig. 14.

5 CONCLUSIONS

The results of these studies demonstrate neural networks as an appropriate tool for both the detection of impulsive disturbances and their removal using nonlinear predictive interpolation of missing intervals. The learning detection algorithm is able to discern unwanted impulsive disturbances from eligible percussive or transient events. The learning algorithm is also applicable for the restoration of missing samples while processing audio. Unfortunately the knowledge base of signal and impulse distortions acquired by the neural networks during training cannot be considered universally applicable. Consequently the process of building the knowledge base inside the neural networks must be repeated if the character of the audio material or the impulse distortions changes.

The prolonged process of training neural networks, which limits their presumed value in practical applications, is not a concern with some other algorithms belonging to the domain of artificial intelligence. Consequently an alternative approach for building the knowledge base of signals and distortions will be presented in Part 2 of this paper. This alternative approach, derived from the rough-set method, is used for the detection and removal of noise found in recorded or transmitted audio signals.

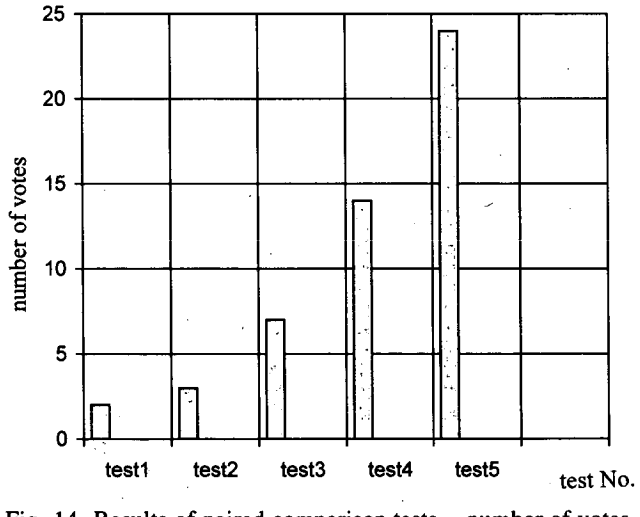


Fig. 14. Results of paired comparison tests—number of votes obtained for individual interpolation methods.

6 ACKNOWLEDGMENT

This research was sponsored by the Committee for Scientific Research, Warsaw, Poland (Projects 8 T11D 002 08 and 8 T11D 021 12). The author would also like to thank Laurent Mainard and Andrzej Kaczmarek for their helpful suggestions when he started the neural network project. He is indebted to his students for their assistance in the software preparation and neural network training.

7 REFERENCES

[1] R. Frayling-Cork and S. V. Vaseghi, "Restoration of Old Gramophone Recordings," *J. Audio Eng. Soc.*, vol. 40, pp. 791–801 (1992 Oct.).

[2] J. C. Valiere, S. Montresor, J. F. Allard, and M. Baudry, "The Restoration of Old Recordings with Digital Techniques," presented at the 88th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 38, p. 382 (1990 May), preprint 2915.

[3] S. Montresor, J. C. Valiere, J. F. Allard, and M. Baudry, "Evaluation of Two Interpolation Methods Applied to the Restoration of Old Recordings," presented at the 90th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 39, p. 382 (1991 May), preprint 3022.

[4] R. C. Maher, "A Method for Extrapolation of Missing Digital Audio Data," *J. Audio Eng. Soc. (Engineering Reports)*, vol. 42, pp. 350–357 (1994 May).

[5] A. Czyzewski and C. Supron, "Learning Algorithms for the Cancellation of Old Recordings Noise," presented at the 96th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 42, p. 407 (1994 May), preprint 3847.

[6] A. Czyzewski, "Artificial Intelligence-Based Processing of Old Audio Recordings," presented at the 97th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 42, p. 1056 (1994 Dec.), preprint 3885.

[7] L. Yin, J. Astola, and Y. Neuvo, "A New Class of Nonlinear Filters—Neural Filters," *IEEE Trans. Signal Process.*, vol. 41 (1993 Mar.).

[8] A. Czyzewski, "Some Methods for Detection and Interpolation of Impulsive Distortions in Old Audio Recordings," in *Proc. IEEE ASSP Workshop* (Mohonk Mountain, New Paltz, NY, 1995).

[9] J. M. Zurada, *Introduction to Artificial Neural Systems* (West Publishing, St. Paul, MN, 1992).

[10] B. Kostek, "Statistical versus Artificial Intelligence Based Processing of Subjective Test Results," presented at the 98th Convention of the Audio Engineering Society, *J. Audio Eng. Soc. (Abstracts)*, vol. 43, p. 403 (1995 May), preprint 4018.

THE AUTHOR

Andrzej Czyzewski was born in Gdańsk, Poland, in 1956. He received the M.S.Eng. degree in sound engineering in 1982 and Ph.D. degree in 1987 from the Technical University of Gdańsk. In 1992 he received the D.Sc. degree from the Academy of Mining and Metallurgy, Cracov.

In 1984 he joined the Sound Engineering Department at the Technical University of Gdańsk. As a research worker, he and his team of research engineers designed several digital devices, some of which were produced in Poland. The main projects under his guidance concerned a programmable digital reverberator, computerized telephone systems, a polyphonic organ synthesizer, signalization systems for visually impaired people, and electronic speech aids. In 1991 he published a monograph devoted to digital audio operations. Then his research interest focused on applications to audio engineering of some unconventional computational methods such as neural networks, fuzzy logic, and rough sets. He led a number of research projects sponsored by the Polish State Committee for Scientific Research concerning new methods of speech recognition, audio restoration, waveguide sound synthesis, and digital processing algorithms for hearing aids and cochlear implants among others. He has presented more than 100 scientific papers in journals and at conferences.

In 1991 Dr. Czyzewski helped establish the Polish Section of the AES. He was elected Secretary of this new section and since 1993 has been a committee member. He is also a member of the IEEE, the International Rough Set Society, and the International Fluency Association. He was elected a member of the Acoustical Committee of the Polish Academy of Science and a member of the Scientific Council of the Institute of Physiology and Pathology of Hearing in Warsaw. He serves as Head of the Sound Engineering Department and a supervisor of doctoral studies at the Faculty of Electronics, Telecommunications and Informatics, of the Technical University of Gdańsk. In 1996 he was promoted to the position of Associate Professor at this university.