Inverse filtering of a loudspeaker and room acoustics using time-delay neural networks

Po-Rong Chang, C. G. Lin, and Bao-Fuh Yeh
Department of Communication Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan

(Received 30 July 1993; accepted for publication 28 January 1994)

This paper presents the design of a neural-based acoustic control used for the equalization of the response of a sound reproduction system. The system usually can be modeled as a composite system of a loudspeaker and an acoustic signal-transmission channel. Generally, an acoustic signal radiated inside a room is linearly distorted by wall reflections. However, in a loudspeaker, the nonlinearity in the suspension system produces a significant distortion at low frequencies and the inhomogeneity in the flux density causes a nonlinear distortion at large output signals. Both the linear and nonlinear distortions should be reduced so that high fidelity sound can be reproduced. However, the traditional adaptive equalizer which is only capable of dealing with linear systems or specific nonlinear systems cannot compensate these nonlinear distortions. The time-delay feedforward neural network (TDNN) which has the capability to learn arbitrary nonlinearity and process the temporal audio patterns are particularly recognized as the best nonlinear inverse filter of the composite system. The performance of a TDNN-based acoustic controller is verified by some simulation results.

PACS numbers: 43.60.Gk, 43.38.Ar, 43.55.Me

INTRODUCTION

The objective of the sound reproduction system has been assumed to be the "perfect" reproduction of the recorded signals at the listener's ears, i.e., the signals recorded at two points in the recorded space are reproduced exactly at points in the listening space. Generally, the sound reproduction system is used to achieve the perfect reproduction of the recorded audio signals at the listener's ears, i.e., the signals recorded at a point in the recording space are reproduced exactly at a point in the listening space. However, the original audio signals are imperfectly reproduced at the ears of a listener when these signals are replayed via loudspeakers in a listening room. The imperfections in the reproduction arise from two main sources: (i) the acoustic signals radiated inside a room are linearly distorted by wall reflections, and (ii) in a loudspeaker, the suspension nonlinearity produces a significant distortion at low frequencies and the imhomogeneity in the flux density causes a nonlinear distortion at large output signals. In order to eliminate the above two undesired factors, it is necessary to introduce inverse filters that act on the inputs to the loudspeakers used for reproduction which will compensate for both the loudspeaker response and the room response. Initial attempts to design such inverse filters has been considered in designing the filters used for the equalization of the response of the room acoustic signal-transmission channel. Neely and Allen[1] showed through computer simulations that the loudspeaker to microphone room impulse response is generally a nonminimum phase. This means that it is not possible to realize the exact inverse of an acoustic system that has nonminimum phases. Alternative approachs for the realization of the inverse[2,3] are on the basis of the conventional least-squares error (LSE) methods. However, this inverse is not an exact inverse but rather an approximate inverse of the acoustic system. The principal objective of such equalization schemes has been assumed to be the production of a "closest possible approximation" to the exact reproduction of a recorded audio signal at a single point in the listening space.

An account[4] of work aimed at producing widespread effectiveness of the equalization of low-frequency sound reproduction in automative interiors shows that such an approach may well be useful. Since the traditional equalization can only deal with the linear systems or specific nonlinear systems, the suspension nonlinearity of loudspeakers will significantly degrade the quality of reproduction at low frequencies by using such an equalization. For small input signals, the loudspeakers can be approximated as a linear system, and the transfer behavior is described by a linear transfer response. However, the nonlinear distortions, i.e., harmonics and intermodulation, increase rapidly when the input signal power becomes larger. This leads to the nonlinear inverse filters that can equalize the nonlinear distortions of the loudspeakers. Most of them are based on the Volterra series expansion.[5–7] The Volterra series is both a useful tool for analyzing weakly nonlinear systems and a basis for synthesizing nonlinear filters with desired parameters. Nevertheless, the realization of Volterra filters suffers from its cumbersome representation and computational inefficiency. The emerging feedforward neural networks[8–10] have the capability to learn arbitrary nonlinearity and show great potential for nonlinear filter application. Artificial neural networks are systems that use nonlinear computational elements to model the neural behavior of the biological nervous systems. The properties of neural networks include: massive parallelism, high computation rates, and ease for VLSI implementation. The neural-based inverse filters would be applied to the equalization of the

 0001-4966/94/95(6)/3400/9/$6.00

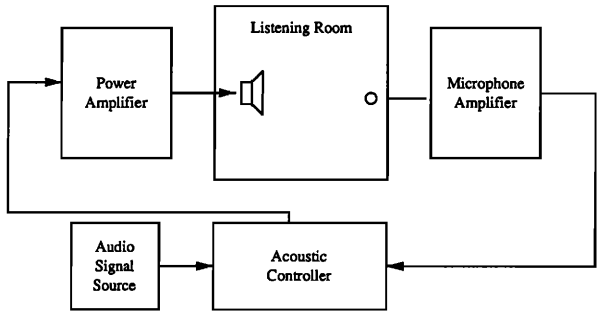


FIG. 1. A stereophonic sound reproduction system with acoustic controller.

response of a composite system of both loudspeakers and room acoustics.

I. MODEL DESCRIPTION FOR A COMPOSITE SYSTEM OF LOUDSPEAKERS AND ROOM ACOUSTICS

As alluded to earlier, the problem of loudspeaker–room interaction draws more attention in accurate sound reproduction. Bascially, a block diagram of a composite system consisting of loudspeakers and a room acoustic signal-transmission channel is illustrated in Fig. 1. However, a lot of researchers[1–3] did not consider the nonlinear distortion of loudspeakers which could degrade the quality of sound reproduction. In this section, we would like to review the mathematical models for the loudspeakers and room acoustics. A nonlinear inverse filter based on time-delay neural networks will be introduced in the next section.

A. The image models for room acoustics

Image methods are commonly used for the analysis of the acoustic properties of enclosures. Allen and Barkley[11] applied the image model to characterize the impulse response of the acoustic signal-transmission channel in a small rectangular room. Moreover, the room reverberation of any input audio signal can be obtained when the resulting impulse response is convolved with the input signal.

Usually, a loudspeaker in a room is modeled as a point source in a rectangular cavity. A signal frequency point source of acceleration in free-space emits a pressure wave of the form,

$$P(\omega,\mathbf{X},\mathbf{X}')=\frac{\exp[\,j\omega(R/c-t)\,]}{4\pi R}, \tag{1}$$

where P=pressure, ω=angular frequency, t=time, c=speed of sound, R=distance between X and X′, X=the vector that represents the loudspeaker's location (x,y,z), and X′=the vector that represents the microphone's location (x',y',z').

When a rigid wall is present, the rigid wall boundary condition may be satisfied by placing an image symmetrically on the far side of the wall. Since there are generally six walls that enclose a room, the situation becomes more complicated because each image is itself imaged. Allen and Berkley[11] showed that the pressure can be written,

$$P(\omega,\mathbf{X},\mathbf{X}')=\sum_{p=1}^{8}\sum_{r=-\infty}^{\infty}\frac{\exp[\,j(\omega/c)|R_p+R_r|\,]}{4\pi|R_p+R_r|}\times\exp(-j\omega t), \tag{2}$$

where R_p denotes the eight permutation vectors over the positive and negative signs,

$$R_p=(x\pm x',y\pm y',z\pm z'),\quad 1\leqslant p\leqslant 8 \tag{3}$$

and

$$R_r=2\mathbf{rL}=2(nL_x,lL_y,mL_z), \tag{4}$$

where $\mathbf{r}=(n,l,m)$ is an integer vector and $\mathbf{L}=(L_x,L_y,L_z)$ is a vector that represents the room dimensions.

Since Eq. (2) is the pressure frequency response, its corresponding time-domain impulse response can be obtained by taking the inverse Fourier transform,

$$p(t,\mathbf{X},\mathbf{X}')=\sum_{p=1}^{8}\sum_{r=-\infty}^{\infty}\frac{\delta[t-(|R_p+R_r|/c)]}{4\pi|R_p+R_r|}. \tag{5}$$

In practice, the room walls are not rigid. Allen and Berkley[11] showed that the nonrigid walls can be approximated by the above point image method with an angle-independent pressure wall reflection coefficient β. The wall reflection coefficients are greater than 0.7 over the frequency range of 100 Hz–4 kHz for the typical listening room geometries. Introducing the wall reflection coefficients into Eq. (5), the time-domain impulse response becomes

$$p(t,\mathbf{X},\mathbf{X}')=\sum_{p=0}^{1}\sum_{r=-\infty}^{\infty}\beta_{x_1}^{|n-p|}\beta_{x_2}^{|n|}\beta_{y_1}^{|l-i|}\beta_{y_2}^{|l|}\beta_{z_1}^{|m-k|}\beta_{z_2}^{|m|}\times\frac{\delta[t-(|R_p+R_r|/c)]}{4\pi|R_p+R_r|}, \tag{6}$$

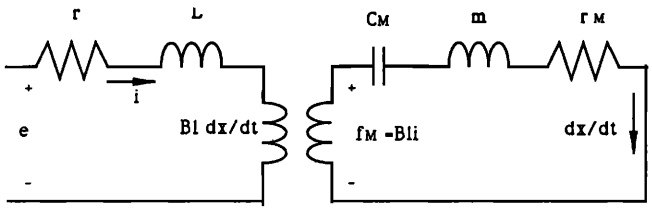
where R_p is now expressed in terms of the integer vector $\mathbf{p}=(q,i,k)$ as

$$R_p=(x-x'+2qx',y-y'+2iy',z-z'+2kz'). \tag{7}$$

The β's are the pressure reflection coefficients of the six walls with the subscript "1" referring to walls adjacent to the reference origin. Subscript 2 is the opposing wall. Here, R_p is identical to that of (5).

B. Equivalent electrical and mechanical circuit model for a loudspeaker

A loudspeaker is composed of an electrical part and a mechanical part. The electrical part is the voice coil. The mechanical part consists of the cone, the suspension, and the air load. The two parts interact through the magnetic field. The mechanical part can also be described by an equivalent electrical circuit. References 6 and 7 introduced an equivalent electrical circuit of a loudspeaker that is shown in Fig. 2. In the voice coil electrical circuit, e, i, r, L, and E represent the input voltage, the current in the voice coil, the electrical resistance of the voice coil, the inductance of the voice coil, and the voltage produced in the electrical circuit by the mechanical circuit, respectively. The voltage E is equal to $Bl\,dx/dt$, where B is the mag-



r : Total resistance of the generator.
L : Inductance of the voice coil.
B : Magnetic flux density in the air gap(nonlinear device).
l : Length of the voice coil conductance.
x : Coil displacement.
C_M: Compliance of the suspension (nonlinear device).
m : Total mass of the coil, the cone and the air load.
r_M : Total mechanical resistance.
f_M : Force generated in the voice coil.

FIG. 2. Equivalent electrical and mechanical circuit of a loudspeaker.

netic flux density in the air gap, l is the length of the voice coil conductor, and x is the cone displacement. In the mechanical circuit, m, r_M, C_M, and f_M denote the total mass of the coil and the air load, total mechanical resistance due to the dissipation in the air load and the suspension system, the compliance of the suspension, and the force generated in the voice coil, respectively. The force f_M is equal to Bli.

Generally, the mechanomotive force in the voice coil is a nonlinear function of the displacement x. The force deflection characteristics of the loudspeaker cone suspension system is approximated by

$$f_M=\alpha x+\beta x^2+\gamma x^3. \tag{8}$$

Thus the compliance of the suspension system is obtained

$$C_M=\frac{x}{f_M}=\frac{1}{\alpha+\beta x+\gamma x^2}. \tag{9}$$

Another source of harmonic distortion is nonuniform flux density B. The flux density is a function of the displacement x, which may also be approximated by a second-order polynomial,

$$B(x)=B_0+B_1x+B_2x^2. \tag{10}$$

Let the state variables $x_1=i$, $x_2=x$, and $x_3=dx_2/dt$. From the equivalent electrical and mechanical circuits, one can obtain the following state-space dynamical equation:

$$\begin{aligned}
\frac{dx_1}{dt}&=\frac{1}{L}(-rx_1-B_0lx_3+e-B_1lx_2x_3-B_2lx_2^2x_3),\\
\frac{dx_2}{dt}&=x_3,\\
\frac{dx_3}{dt}&=\frac{1}{m}(B_0lx_1-\alpha x_2-r_Mx_3-\beta x_2^2-\gamma x_2^3+lB_1x_1x_2\\
&\quad +lB_2x_1x_2^2),
\end{aligned} \tag{11}$$

and

$$y(t)=x_2(t), \tag{12}$$

where $y(t)$ is output signal of the loudspeaker.

II. NEURAL-BASED MODEL IDENTIFICATION

Neural networks have become a very fashionable area of research with a range of potential applications that spans artificial intelligence (AI), engineering, and science. All the applications are dependent upon training the network with illustrative examples and this involves adjusting the weights which define the strength of connection between the neurons in the network. This can often be interpreted as a system identification problem of estimating the system that transforms inputs into outputs given a set of examples of input–output pairs.

This section focuses on the feasibility of neural networks and their learning algorithms for training the networks to represent forward and inverse transform models of nonlinear acoustic systems. Training a neural network using input–output data from a nonlinear plant can be considered as a nonlinear functional approximation problem. Approximation theory is a classical field of mathematics. From the well-known Stone–Weierstrass theorem,[12] it shows that polynomials can approximate arbitrarily well a continuous function. Recently, the approximation capability of networks has been investigated[8–10] by using the similar concept based on the Stone–Weierstrass theorem. A number of results have been published showing that a feedforward network of the multilayer perceptron type can approximate arbitrarily well a continuous function.[8–10] To be specific, these research works prove that multilayer feedforward networks with as few as a single layer and an appropriately smooth hidden layer activation function are capable of arbitrarily accurate approximation to an arbitrary continuous function.

Before applying the feedforward neural networks to the model identification of loudspeaker-room system, it is important to establish their approximation capabilities to some arbitrary nonlinear real-vector-valued continuous mapping $\mathbf{y}=\mathbf{f}(\mathbf{x}):D\subseteq R^n\rightarrow R^m$ from input/output data pairs $\{\mathbf{x},\mathbf{y}\}$, where D is a compact set on R. Consider a feedforward network $F(\mathbf{x},\mathbf{w})$ with $\mathbf{x}$ as a vector representing inputs and $\mathbf{w}$ as a parameter-weighting vector that is updated by some learning rules. It is desired to train $F(\mathbf{x},\mathbf{w})$ to approximate the mapping $\mathbf{f}(\mathbf{x})$ as close as possible. The Stone–Weierstrass theorem[12] shows that for any continuous function $\mathbf{f}\in C^1(D)$ with respect to $\mathbf{x}$, a compact metric space, an $F(\mathbf{x},\mathbf{w})$ with appropriate weight vector $\mathbf{w}$ can be found such that $\|F(\mathbf{x},\mathbf{w})-\mathbf{f}(\mathbf{x})\|_{\mathbf{x}}<\epsilon$ for an arbitrary $\epsilon>0$, where $\|\mathbf{e}\|_{\mathbf{x}}$ is the norm defined by

$$\|\mathbf{e}\|_{\mathbf{x}}=\sup_{\mathbf{x}}\{\|\mathbf{e}(\mathbf{x})\|:\mathbf{x}\in D,\quad \|\cdot\| \text{ is the vector norm}\}. \tag{13}$$

For network approximators, key equations are how many layers of hidden units should be used, and how many units are required in each layer. Cybenko[9] and Homik *et al.*,[10] have shown that the feedforward network with a single hidden layer can uniformly approximate any continuous function to an arbitrary degree of exactness—providing that the hidden layer contains a sufficient number of units. However, it is not cost effective for the practical implementation. Nevertheless, Chester[13] gave a

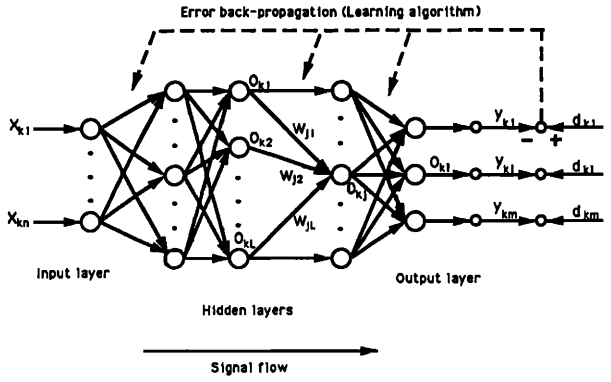


FIG. 3. Multilayer feedforward neural network.

theoretical support to the empirical observation that networks with two hidden layers appear to provide high accuracy and better generalization than a single hidden layer network, and at a lower cost (i.e., fewer total processing units). Since, in general, there is no prior knowledge about the number of hidden units needed, a common practice is to start with a large number of hidden units and then prune the network whenever possible. Additionally, Huang and Huang[14] gave the lower bounds on the number of hidden units which can be used to estimate its order.

A. Feedforward neural networks and their learning rules

A feedforward neural network shown in Fig. 3 is a layered network consisting of an input layer, an output layer, and at least one layer of nonlinear processing elements. The nonlinear processing elements, which sum incoming signals and generate output signals according to some predefined function, are called neurons. In this paper, the function used by nonlinear neurons is called the sigmoidal hyperbolic tangent function G, which is similar to a smoothed step function,

$$G(x)=\tanh(x). \tag{14}$$

The neurons are connected by terms with variable weights. The output of one neuron multiplied by a weight becomes the input of an adjacent neuron of the next layer.

In 1986, Rumelhart *et al.*[15] proposed a generalized delta rule known as backpropagation for training layered neural networks. For control engineers, it is appropriate to consider feedforward neural networks as a tool to solve function approximation problems rather than pattern recognition problems. In mathematical sense, the backpropagation learning rule is used to train the feedforward network $F(\mathbf{x},\mathbf{w})$ to approximate a function $\mathbf{f}(\mathbf{x})$ from compact subset D of n-dimensional Euclidean space to a bounded subset $\mathbf{f}(D)$ of m-dimensional Euclidean space. Let $\mathbf{x}_k$ which belongs to D be the kth pattern or sample and selected randomly as the input of the neural network, let $\mathbf{F}(\mathbf{x}_k,\mathbf{w})$ $(=\mathbf{o}_k)$ be the output of the neural network, and let $\mathbf{f}(\mathbf{x}_k)(=\mathbf{d}_k)$ which also belongs to $\mathbf{f}(D)$ be the desired output. This task is to adjust all the variable weights of the neural network such that the system error E can be reduced, where E is defined as

$$E=\sum_{k=1}^{N} E_k, \tag{15}$$

where E_k is the square error of the kth pattern,

$$\begin{aligned} E_k&=\tfrac{1}{2}\,\|F(\mathbf{x}_k,\mathbf{w})-\mathbf{f}(\mathbf{x}_k)\|^2 \\ &=\tfrac{1}{2}\,\|\mathbf{o}_k-\mathbf{d}_k\|^2 \\ &=\frac{1}{2}\sum_{j=1}^{m}(o_{kj}-d_{kj})^2, \end{aligned} \tag{16}$$

and N is the number of samples, o_{kj} and d_{kj} are the jth compenents of $\mathbf{o}_k$ and $\mathbf{d}_k$, respectively.

Here, we define the weighted sum of the output of the previous layer by the presentation of input pattern $\mathbf{x}_k$:

$$\text{net}_{kj}=\sum_i w_{ji}o_{ki}, \tag{17}$$

where w_{ji} is the weight that connects the output of the ith neuron in the previous layer with respect to the jth neuron, and o_{ki} is the output of the ith neuron. It should be noted that o_{ki} is equal to x_{ki} when the ith neuron is located in the input layer, where x_{ki} is the ith component of pattern x_k. Using (14), the output of neuron j is

$$o_{kj}=\begin{cases} x_{kj}, & \text{if the neuron } j \text{ belongs to the input layer} \\ G(\text{net}_{kj}), & \text{otherwise.} \end{cases} \tag{18}$$

As discussed above, the goal is to choose the network connection weights w_{ji}'s such that the system error E is reduced. The most popular technique for training neural networks and modifying those connection parameters is the backpropagation algorithm.[16] The algorithm computes iteratively the partial derivatives of the system error E with respect to each parameter and modify the parameter in order to achieve a gradient descent in E with a momentum term added to dampen oscillations. In particular, the weight w_{ji} is updated at the $t+1$st iteration, according the rule,

$$\Delta w_{ji}(t+1)=-(1-\alpha)\eta(t+1)\frac{\partial E}{\partial w_{ji}}+\alpha\Delta w_{ji}(t), \tag{19}$$

where $\Delta w_{ji}(t+1)$ is the weight increment for the $t+1$st iteration, $\eta(t+1)$ is the learning rate value corresponding to $\Delta w(t+1)$ at time $t+1$, and α is the momentum rate.

Since the expression of $\partial E/\partial w_{ij}$ could be in terms of $\partial E_k/\partial w_{ij}$'s, it is useful to see this partial derivative for pattern k, $\partial E_k/\partial w_{ji}$, as resulting from the product of two parts: one part reflecting the change in error to a function of the change in the network input to the neuron and one part representing the effect of changing a particular weight on the network input:

$$\frac{\partial E_k}{\partial w_{ji}}=\frac{\partial E_k}{\partial\,\text{net}_{kj}}\frac{\partial\,\text{net}_{kj}}{\partial w_{ji}}. \tag{20}$$

From (17), the second part becomes

$$\frac{\partial\,\text{net}_{kj}}{\partial w_{ji}}=\frac{\partial}{\partial w_{ji}}\left(\sum_i w_{ji}o_{ki}\right)=o_{ki}. \tag{21}$$

An error signal term δ called delta produced by the jth neuron is defined as follows:

$$\delta_{kj} \triangleq -\frac{\partial E_k}{\partial(\text{net}_{kj})}. \tag{22}$$

Note that E is a composite function of net_{kj}, it can be expressed as follows:

$$E_k = E_k(o_{k1}, o_{k2}, \ldots, o_{kL}) = E_k(G(\text{net}_{k1}), G(\text{net}_{k2}), \ldots, G(\text{net}_{kL})), \tag{23}$$

where L is the number of the neurons in the current layer. Thus we have from (22),

$$\eta_{kj} = -\frac{\partial E_k}{\partial o_{kj}} \frac{\partial o_{kj}}{\partial \text{net}_{kj}}. \tag{24}$$

Denoting the second term in (24) as a derivative of the activation function,

$$\frac{\partial o_{kj}}{\partial \text{net}_{kj}} = G'(\text{net}_{kj}). \tag{25}$$

However, to compute the first term, there are two cases. For the hidden-to-output connections, it follows the definition of E_k that

$$\frac{\partial E_k}{\partial o_{kj}} = -(d_{kj} - o_{kj}). \tag{26}$$

Substituting for the two terms in (24), we can get

$$\delta_{kj} = (d_{kj} - o_{kj}) G'(\text{net}_{kj}). \tag{27}$$

Second, for hidden (or input)-to-hidden connection, the chain rule is used to write

$$\frac{\partial E_k}{\partial o_{kj}} = \sum_l \frac{\partial E_k}{\partial \text{net}_{kl}} \frac{\partial \text{net}_{kl}}{\partial o_{kj}} = -\sum_l \delta_{kl} w_{lj}. \tag{28}$$

Substituting into (24), it yields

$$\delta_{kj} = G'(\text{net}_{kj}) \sum_l \delta_{kl} w_{lj}. \tag{29}$$

Equations (27) and (29) give a recursive procedure for computing the δ's for all neurons in the network. Once those error signal terms have been determined, the partial derivatives for the system error can be computed directly by

$$\frac{\partial E}{\partial w_{ji}} = \sum_{k=1}^{N} \frac{\partial E_k}{\partial w_{ji}} = \sum_{k=1}^{N} \frac{\partial E_k}{\partial \text{net}_{kj}} \frac{\partial \text{net}_{kj}}{\partial w_{ji}} = -\sum_{k=1}^{N} \delta_{kj} o_{ki}, \tag{30a}$$

where

$$\delta_{kj} = \begin{cases} (d_{kj} - o_{kj}) G'(\text{net}_{kj}), \\ \quad \text{if neuron } j \text{ belongs to the output layer,} \\ G'(\text{net}_{kj}) \sum_l \delta_{kl} w_{lj}, \quad \text{otherwise.} \end{cases} \tag{30b}$$

It should be mentioned that o_{ki} is equal to x_{ki} when neuron i belongs to the input layer. The expression of Eqs. (30a) and (30b) is also called the generalized delta learning rule.

Jacobs[16] showed that the momentum can cause the weight to be adjusted up the slope of the system error surface. This would decrease the performance of the learning algorithm. To overcome this difficulty, Jacobs[16] proposed a promising weight update algorithm based on the delta-bar-delta rule which consists of both a weight update rule and learning rate update rule. The weight update rule is the same as the steepest descent algorithm and is given by (19). The delta-bar-delta learning rate update rule is described as follows:

$$\Delta\eta(t+1) = \begin{cases} \kappa, & \text{if } \bar{\lambda}(t-1)\lambda(t) > 0, \\ -\phi\eta(t), & \text{if } \bar{\lambda}(t-1)\lambda(t) < 0, \\ 0, & \text{otherwise,} \end{cases} \tag{31a}$$

where

$$\lambda(t) = \frac{\partial E}{\partial w_{ij}} \tag{31b}$$

and

$$\bar{\lambda}(t) = (1-\theta)\lambda(t) + \theta\bar{\lambda}(t-1). \tag{31c}$$

In these equations, $\lambda(t)$ is the partial derivative of the system error with respect w_{ij} at the tth iteration and $\bar{\lambda}(t)$ is an exponential average of the current and past derivatives with θ as the base and index of iteration as the exponent. If the current derivative of a weight and the exponential average of the weight's previous derivatives possess the same sign, then the learning rate for that weight is incremented by a constant κ. The learning rate is decremented by a proportion ϕ of its current value when the current derivative of a weight and the exponential average of the weight's previous derivatives possess opposite signs.

From Eqs. (31a) and (31c), it can be found that the learning rates of the delta-bar-delta algorithm are incremented linearly in order to prevent them from becoming too large too fast. The algorithm also decrements the learning rates exponentially. This ensures that the rates are always positive and allows them to be decreased rapidly. Jacobs[16] showed that a combination of the delta-bar-delta rule and momentum heuristics can achieve both the good performance and faster rate of convergence.

B. Forward and inverse system model identification by feedforward network

In gereral, system identification is usually recognized as a process to train networks to represent nonlinear dynamical systems and their inverses. This would be distinctly helpful in achieving the desired output signal of the system. The issue of identification is perhaps of even greater importance in the field of adaptive control and signal processing systems.[17] Since the plant in an adaptive control varies in operation with time, the adaptive control must be adjusted to account for the plant variations.

The procedure of training a neural network to represent the forward dynamics will be referred to as forward model identification. The basic configuration for achieving this is shown schematically in Fig. 4. A feedforward neural network with a single hidden layer is placed in parallel

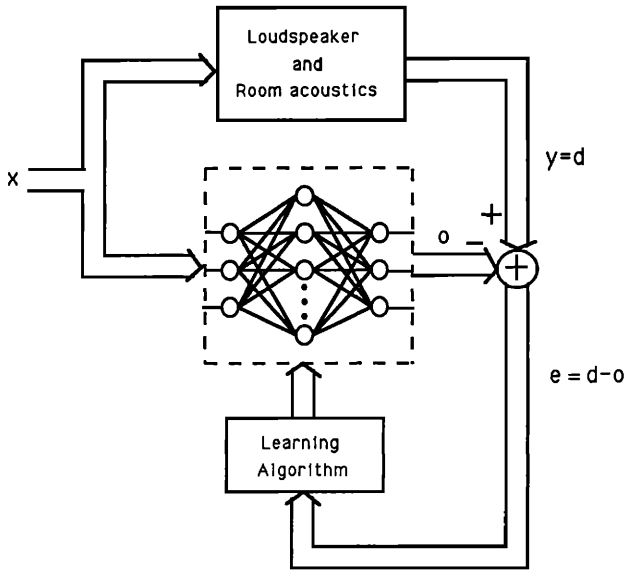


FIG. 4. Forward model identification.

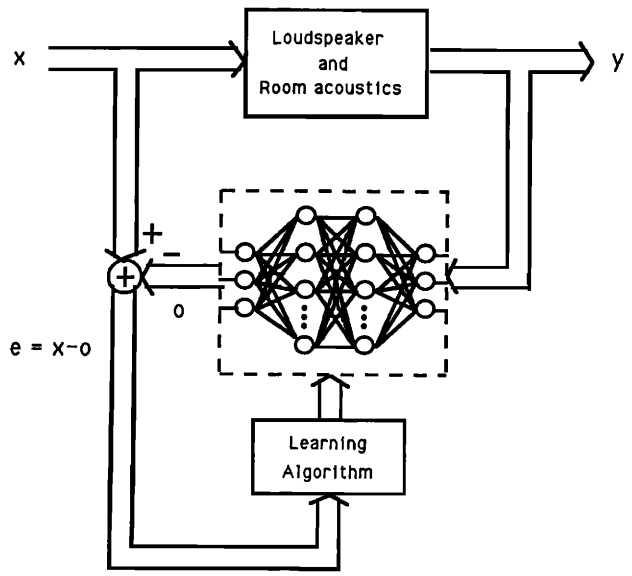


FIG. 5. Inverse model identification.

with the system and receives the same input **x** as the system. The system output provides the desired response **d** during training. The purpose of the identification is to find the appropriate weights $w_{ij}'s$ of the network with response **o** that matchs the response **y** of the system for a given set of inputs **x**. During the identification, the norm of the error vector, $\|\mathbf{e}\|=\|\mathbf{d}-\mathbf{o}\|$, is minimized through a number of weight adjustments by the delta-bar-delta learning rule. In our case, those weights are updated by minimizing the system error, $E=\Sigma_{k=1}^{N}\|\mathbf{e}_k\|^2$, by the same algorithm, where $\mathbf{e}_k=(\mathbf{d}_k-\mathbf{o}_k)$. Figure 4 shows the case for which the network attempts to model the mapping of system input to output, which both input and output measured at the same time. With sufficiently large number of hidden units, Stone–Weierstrass theorem shows that the neural network will be identical to the system in the domain of interest.

Figure 4 shows use of a feedforward neural network for direct modeling of an unknown system to obtain a close approximation to its responses. By changing the configuration, it is possible to use the feedforward network for inverse modeling to obtain the reciprocal of the unknown system's transfer function when the system is invertible. In contrast to forward system characteristics identification, the system output **o** is used as neural network input, as shown in Fig. 5. The unknown system's input **x** delayed by Δ time units is the desired response of the feedforward network. Thus the error vector of network training is computed as a $\mathbf{x}(t-\Delta)-\mathbf{o}(t)$. The system error to be minimized through learning is therefore $\tilde{E}=\Sigma_{k=1}^{N}\|\mathbf{x}(t_k-\Delta)-\mathbf{o}(t_k)\|^2$. The neural network trained by the delta-bar-delta algorithm will implement the mapping of the system inverse. Once the network has been successfully trained to mimic the delayed system inverse, it can be used directly for inverse feedforward control. In other words, the inverse model is cascaded with the controlled unknown system in order that the composed system results in an identity mapping with a time delay Δ between desired response (i.e., the network inputs) and the controlled system output. The output of the system follows the input signal delayed by Δ time samples.

As mentioned, it is assumed that the system is invertible. Then there exists an injective mapping which represents its inverse. If it is not true, a major problem with system inverse identification arises when the system inverse is not uniquely defined. A second approach to inverse modeling which aims to overcome these problems is known as specialized inverse learning.[18] As pointed out in Psaltis *et al.*,[18] the specialized method allows the training of the inverse network in a region in the expected operational range of the system. On the other hand, the generalized training procedure produces an inverse over the operating space which may be uniquely defined. Fortunately, the mapping of a loudspeaker-room system may have a unique inverse or its approximation. Thus we could apply the direct invesre method as illustrated in Fig. 5 to find the approximation of the inverse. In addition, since the nonlinearity of a system inverse is higher than that of forward modeling, a network with two hidden layers is considered in constructing the inverse modeling.

III. INVERSE FILTERING AND MODEL IDENTIFICATION OF A LOUDSPEAKER-ROOM SYSTEM BY THE TIME DELAY NEURAL NETWORK

As discussed in the above section, the feedforward neural network results in a static network which maps static input patterns to static output patterns. Any temporal patterns in the input data are not recognizable by such a network. A time delay neural network (TDNN) shown in Fig. 6 is extended to networks with delay elements in the connections can be trained to recognize specific spectral structures within a consecutive frame of audio signal.[19] Usually, these temporal audio patterns generated by loudspeaker-room system can be governed by a nonlinear discrete-time difference equation where the output has a finite temporal dependence on the input, that is,

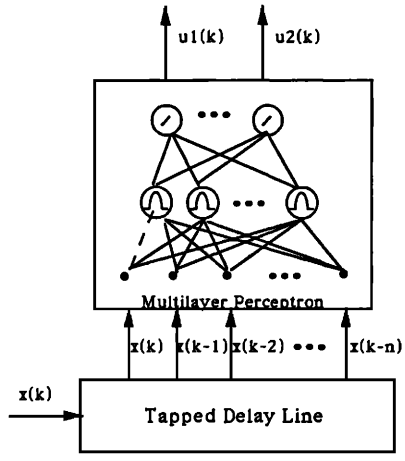


FIG. 6. Time-delay neural network.

$$y(t)=\mathbf{f}(\mathbf{x}(t-\Delta_f),\mathbf{x}(t-\Delta_f-1),\ldots,\mathbf{x}(t-\Delta_f-n_f)), \tag{32}$$

where Δ_f is the forward system time delay and (Δ_f+n_f) is the maximum lag in the input.

This architecture is equivalent to a linear finite impulse response (FIR) filter when the function $\mathbf{f}(\cdot)$ is a weighted linear sum. This would be identical to our method without including the nonlinearity of the loudspeakers. The transfer function of the acoustic signal-transmission channel between loudspeaker and microphone is denoted as $G(z)$, which is a FIR (finite impulse response) system, $G(z)$ represents the reflective sound as well as the direct sound between the loudspeaker and microphone.

To process the time series data generated by (32), it is possible to convert the temporal audio sequence into a static pattern by unfolding the sequence over time and then use this pattern to train a static network. From a practical point of view, it is suggested to unfold the sequence over a finite period of time. This can accomplished by feeding the input sequence into a tapped delay line of finite extent, then feeding the taps from the delay line into a static feedforward network. Because there is no feedback in this network, it can be trained using the standard backpropagation algorithm.

Since the input–output structure of a real acoustic system involves the loudspeaker's nonlinearity, it is quite difficult to describe clearly the dynamic behavior of the inverse of a nonlinear system. For simplicity, we would like to discuss the linear acoustic signal-transmission channel and its inverse. Generally, Refs. 1 and 2 showed that the transfer function of the acoustic channel $G(z)$ is considered to be a nonminimum phase system where $G(z)$ has one or more of its zeros outside the unit circle in the z plane. A reciprocal transfer function $D(z)(=1/G(z))$ would then have unstable poles. Usually, the reciprocal function $D(z)$ can be decomposed into two component subsystems, each of which has all of its poles either inside or outside the unit circle, that is,

$$D(z)=D_c(z)\,D_{ac}(z), \tag{33}$$

where $D_c(z)$ and $D_{ac}(z)$ have stable and unstable poles, respectively.

In other words, the system implements $D(z)$ by a cascade connection of the subsystems $D_c(z)$ and $D_{ac}(z)$. Because the poles of $D_c(z)$ are inside the unit circle, a stable causal recursive filter can implement $D_c(z)$. Since the poles of $D_{ac}(z)$ are outside the unit circle, no stable implementation of $D_{ac}(z)$ exist if the causality is required. However, by allowing their impulse responses to extend backward in time, the stable inverse filter does exist. To better understand this principle, it is necessary to review the properties of the bilateral z transform. A specific pole contributes either to the causal or the anticausal portion of the impulse response of $D_{ac}(z)$ depending on the associated region of convergence (ROC) of its z transform. Consider a circle in the z plane that is centered at the origin and that passes a pole; if the ROC associated with the pole lies outside that circle, then the time response extends forward in time. Conversely, if the ROC of the pole is inside that circle, the time response extends backward in time. For example, the same expression of $D_{ac}(z)=z/z-a$, $a>1$ yield two different impulse responses, that is, a stable anticausal impulse response $h_1[n]=-(a)^n u[-n-1]$ and an unstable causal impulse response $h_2[n]=(a)^n u[n]$, where $u[n]$ is a discrete-time unit step function.

The ROC of the entire system is the intersection of the ROCs of all the poles. This intersection must include the unit circle for the system to be stable. Therefore, there exists a stable inverse to $G(z)$ when the impulse response of $D_c[z]$ is strictly causal and the impulse response of $D_{ac}(z)$ is strictly anticausal. If the anticausal impulse response is finite in duration, noncausal filtering can be achieved exactly when the filtering introduces delay, effectively shifting the impulse response until it is strictly causal. Moreover, Widrow[17] showed that the anticausal impulse response could be approximated by a causal stable impulse response truncated and shifted in time. As a result, it can be shown that a causal FIR filter can approximate a delayed version of the system inverse $D(z)$. This argument is also true when the room acoustic system includes the nonlinearity of speakers. The causal FIR filter can be replaced by a nonlinear FIR filter which approximates the system inverser of (32) and given by

$$u(t)=g(x(t-\Delta_I),x(t-\Delta_I-1),\ldots,x(t-\Delta_I-n_I)), \tag{34}$$

where $x(t)$ is the input voltage, $u(t)$ is the driver voltage to the loudspeaker, Δ_I is the system inverse time delay, and (Δ_I+n_I) is the maximum lag in the input.

Similarly, the TDNN can implement the nonlinear FIR filter by inputting the temporal audio signal, i.e., $x(t)$ to a tapped delay line, then feeding the taps from the delay line into a static feedfoward network. Next the output of the static network, i.e., $u(t)$ acts as a drive voltage to the composite system of loudspeaker and room acoustic channel. From the previous section, the inverse of the composite system can be identified by minimizing the system error $E=\Sigma_{k=1}^{N}\|x(t_k-\Delta_I)-O(t_k)\|^2$ where $O(t_k)$ is the kth sample of the response of the composite system. The trained

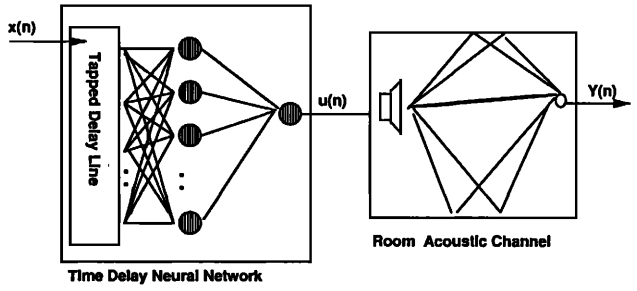


FIG. 7. TDNN-based plant inverse acoustic controller.

TDNN-based inverse model can be cascaded with the composite system and then preequalize its response. The properly trained TDNN acts as the inverse feedforward controller in the configuration as illustrated in Fig. 7.

IV. ILLUSTRATED EXAMPLES

To evaluate the performance of TDNN-based model identification, the simulation of the composite system of loudspeakers and room acoustics is performed by a fourth-order Runge–Kutta method with sampling period $=1/5$ kHz $=2\times10^{-4}$s. The dimensions of the rectangular listening room are $10\times15\times12.5$ ft^3 with equal wall reflection coefficients of $\beta_x=\beta_y=0.9$ and with floor and ceiling reflection coefficients of $\beta_z=0.7$. The loudspeaker is mounted on location (3.75, 12, and 5 ft). The location of microphone is (6.25, 1.25, and 7.5 ft). The simulated room impulse response can be calculated using the image method.[11] For the simulation of loudspeakers, it requires knowing the values of the associated parameters, that is, $(r/L)=1.1$, $(B_0l/L)=0.2$, $(B_0l/m)=0.6$, $(\alpha/m)=0.5$, $(r_M/m)=1.15$, $(B_1l/L)=0.04$, $(B_2l/L)=0.05$, $(r/m)=0.08$, $(lB_1/m)=0.01$, and $(lB_2/m)=0.02$ which are suggested by Ref. 6. Since β is very small in practice, β is chosen as zero. Notice that e of (11) is the input voltage to the dynamic system. Here, $y(t)$ of (12) is the resulting output of the loudspeaker and also acts as an input signal to the room acoustic signal transmission channel. As a result, the response of a composite system of loudspeaker and room acoustics is produced by convolving $y(t)$ with the calculated room impulse response.

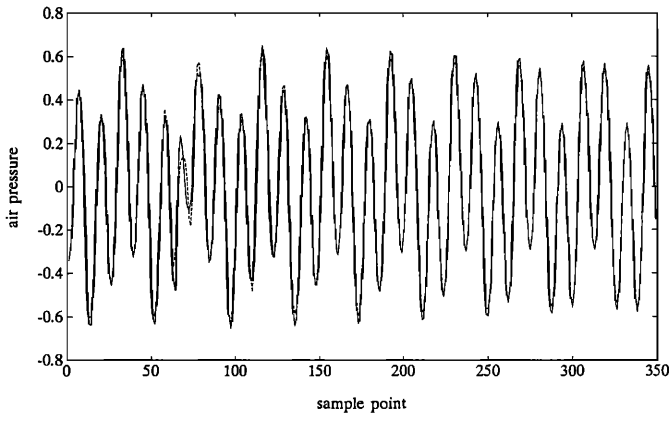


FIG. 8. Comparison of the desired response (—) from the actual system and the response (- - -) from the estimated forward model.

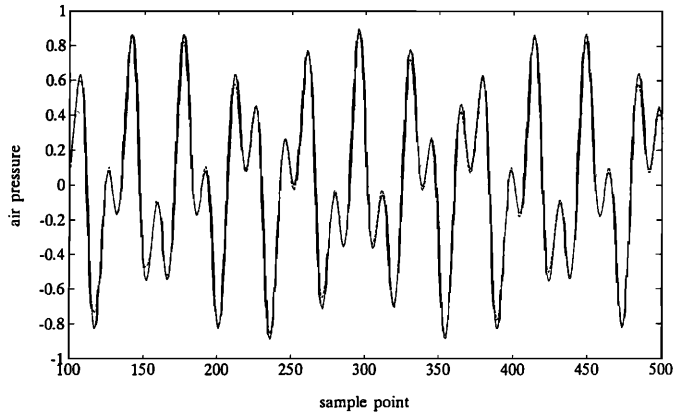


FIG. 9. Comparison of the desired response (—), the response without equalizer (···), and the response (- - -) with TDNN-based equalization.

Two TDNNs with one hidden layer and two hidden layers are designed to learn the forward and inverse models of the composite system, respectively. According to Huang and Huang's suggestions,[14] it is possible to estimate the lower bounds on the numbers of hidden units in both TDNNs. The numbers of hidden units for both the TDNNs associated with the forward and inverse models are chosen as 30 for each layer. For the TDNN associated with the forward model, the number of taps and Δ_f determined by the model validation test[20] are equal to 100 and 40 units, respectively. Similarly, the tap number and Δ_I of the TDNN for the inverse model are chosen as 100 and 90 units, respectively. By inputting 10 000 random sequence with unity maximum amplitude into the composite system and then performing the delta-bar-delta algorithm on the TDNN-based forward model, the training root-mean-square error (rms) can be found as 0.0031. Similarly, it can be found that the training error is 0.007 for TDNN-based inverse model.

Next, we would like to verify the performance of both the estimated TDNN-based forward and inverse models by a test signal $x(t)=0.3\sin(1.28\omega t)+0.5\cos(3.93\omega t)$, where $\omega=200\pi$. The resulting error for the TDNN-based forward and inverse model are 0.015 and 0.0267, respectively. Figure 8 illustrates a comparison of the responses of both the composite system and TDNN-based forward model. Figure 9 shows the tracking performance of the TDNN-based inverse feedforward control. The presented curves clearly show the performance improvement that is achieved by using the proposed inverse model. It should be noted that the error resulted from the composition system without including the TDNN-based inverse controller would become 0.2951. The TDNN-based inverse controller improves the performance of the sound reproduction by an order of magnitude.

V. CONCLUSION

The use of TDNN-based inverse filters for loudspeaker-room correction promises to bring a new level of

accuracy to sound reproduction systems. The inverse filter is simply cascaded with the controlled system in order that the composed system results in an identity mapping between desired response (i.e., the network inputs) and the controlled output. A model of combining the room acoustics and loudspeaker's system dynamics has been developed and studied which take into account a linear reverberant distortion and two principal sources of nonlinear loudspeaker distortion. Based on this, simulations of the proposed method have been performed. The results have shown that both linear and nonlinear distortions of the composite system can be reduced by an order of magnitude.

ACKNOWLEDGMENT

This work was supported by the National Science Council, Taiwan, under Contract NSC 81-0404-E-009-027.

[1] S. T. Neely and J. B. Allen, "Invertibility of a room impulse response," J. Acoust. Soc. Am. **66**, 165–169 (1979).
[2] Masato Miyoshi and Yutaka Kaneda "Inverse filtering of room acoustic," IEEE Trans. Acoust. Speech Signal Process. **36**, 145–152 (1988).
[3] P. A. Nelson, H. Hamada, and S. J. Elliott, "Adaptive inverse filters for stereophonic sound reproduction," IEEE Trans. Acoust. Speech Signal Process. **40**, 1621–1632 (1992).
[4] S. J. Elliott and P. A. Nelson, "Multiple point least squares equalization in a room using adaptive digital filters," J. Audio Eng. Soc. **37**, 899–907 (1988).
[5] R. L. Greiner and M. Schoessow, "Electronic equalization of closed-box loudspeakers," J. Audio Eng. Soc. **31**, 125–134 (1980).
[6] F. X. Y. Gao and W. M. Snelgrove, "Adaptive linearization of a loudspeaker," IEEE Int. Symp. Circ. Syst. 3589–3592 (1991).
[7] A. J. M. Kaizer, "Modeling of the nonlinear response of electrodynamic louspeaker by a Volterra series expansion," J. Audio Eng. Soc. **35**, 421–433 (1987).
[8] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," Neural Networks **2**, 359–366 (1989).
[9] G. Cybenko, "Approximations by superposition of a sigmoidal function," Math. Control Syst. Signals **2**, 303–314 (1989).
[10] K. Hornik, M. Stinchcombe, and H. White, "Universal approximation of an unknown mapping and its derivatives using multilayer feedforward networks," Neural Networks **3**, 551–560 (1990).
[11] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," J. Acoust. Soc. Am. **65**, 943–950 (1979).
[12] M. E. Cotter, "The Stone–Weierstrass theorem and its applications to neural nets," IEEE Trans. Neural Networks, **1**, 290–295 (1990).
[13] D. Chester, "Why two hidden layers are better than one," in *Proceedings of the International Joint Conference on Neural Networks* (IEEE, Washington, DC, 1989), pp. 613–618.
[14] S. C. Huang and Y. F. Huang, "Bounds on the number of hidden neurons in multilayer perceptrons," IEEE Trans. Neural Networks, **2**, 47–55 (1991).
[15] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing*, edited by D. E. Rumelhart and J. L. McCelland (M.I.T., Cambridge, MA, 1986), pp. 318–362.
[16] R. A. Jacobs, "Increased rates of convergence through learning rate adaptation," Neural Networks, 295–307 (1988).
[17] B. Widrow and R. Winter, "Neural nets for adaptive filtering and adaptive pattern recognition," IEEE Comput. Mag. **21**, 25–39 (1988).
[18] D. Psaltis, A. Sideris, and A. A. Yamamura, "A multilayer neural network controller," IEEE Control Syst. Mag. **8**, 17–21 (1988).
[19] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," IEEE Trans. Acoustic Speech Signal Process. **ASSP-37**, 328–339 (1988).
[20] S. A. Billings and W. S. F. Voon, "Correlation based model validity tests for non-linear models," Int. J. Control **44**, 235–244 (1986).