Learning Algorithms for Adaptive Signal Processing and Control

Bernard Widrow Michael Lehr Françoise Beaufays Eric Wan Michel Bilello
Stanford University Department of Electrical Engineering, Stanford, CA 94305-4055

***Abstract*— Linear and nonlinear adaptive filtering algorithms are described, along with applications to signal processing and control problems such as prediction, modeling, inverse modeling, equalization, echo cancelling, noise cancelling, and inverse control.**

I. Introduction

The purpose of this paper is to present an overview of the learning algorithms that are used in both linear and nonlinear adaptive filters, and to describe a number of significant applications for these filters. The basic building block of adaptive filters is the adaptive linear combiner shown in Fig. 1. A set of input signals, represented at time k by the vector $\mathbf{X}_k \triangleq [x_{0k}, x_{1k}, x_{2k}, \ldots, x_{nk}]^T$, are multiplied by variable coefficients or weights, represented by the vector $\mathbf{W}_k \triangleq [w_{0k}, w_{1k}, w_{2k}, \ldots, w_{nk}]^T$, to form the output signal $y_k = \mathbf{X}_k^T \mathbf{W}_k = \mathbf{W}_k^T \mathbf{X}_k$. The output signal is compared with a "desired response" signal d_k which is supplied during the training process. The error signal is defined as the difference between the desired output and the actual output, $\epsilon_k \triangleq d_k - y_k$. When the $\mathbf{X}$-input and the corresponding desired-response input are applied to the linear combiner during training, the weights are adjusted to minimize mean-square-error. Thus, the linear combiner learns to produce an output which is the best linear least squares estimate of the desired response. These ideas are fundamental to learning in an adaptive filter. They are also fundamental to learning in neural networks.

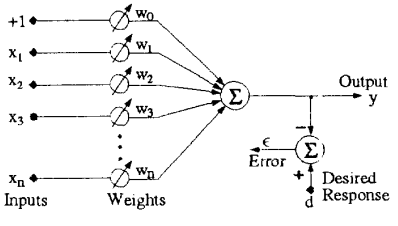


Fig. 1: An adaptive linear combiner.

II. Adaptive LMS Filtering

An adaptive linear filter is shown in Fig. 2. The input signal is discrete in time and is applied to a "tapped delay line," a string of unit delays (represented by z^{-1}). The output signal is a weighted linear combination of present and past input samples, which at time k is given by the convolution sum $y_k = \sum_{i=1}^{n} w_i x_{k-i+1}$. For training, the desired response signal is supplied at the same time as the input signal. The filter learns to produce an output signal which is a best linear least squares estimate of the desired response.

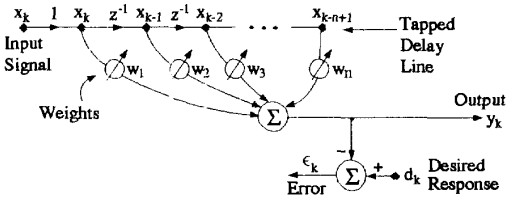


Fig. 2: An adaptive linear filter.

It can easily be shown [1] that when the input and the desired response of the adaptive filter of Fig. 2 are stochastic and stationary, the mean-square-error (MSE) is a quadratic function of the weight values. Fig. 3 shows a typical MSE function which in this 2-weight case is a paraboloid. With many weights, the MSE function is a hyperparaboloid. To adjust the weights to minimize MSE, the method of steepest descent is the simplest and most commonly used approach. Fig. 3 shows a gradient search, starting with an initial guess, and following the gradient with a series of steps, finally arriving at the "bottom of the bowl," where the MSE is minimized.

In 1959, an algorithm to perform steepest descent was developed by Widrow and Hoff [2]. This algorithm, called LMS (Least Mean Square), uses an instantaneous gradient and is so simple and effective that today it remains the most widely used learning algorithm for adaptive signal processing. The algorithm is described by a simple recursive formula:

$$\begin{aligned} \mathbf{W}_{k+1} &= \mathbf{W}_k + 2\mu\epsilon_k\mathbf{X}_k \\ \epsilon_k &= d_k - \mathbf{X}_k^T\mathbf{W}_k \end{aligned} \qquad (1)$$

The present weight vector is $\mathbf{W}_k$, and the next weight vector is $\mathbf{W}_{k+1}$. The present error is ϵ_k. The present

0-7803-0999-5/93/$03.00 ©1993 IEEE

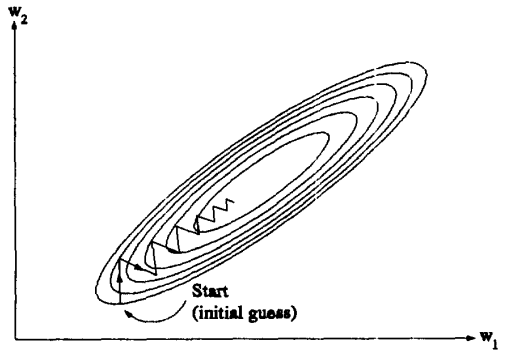


Fig. 3: A quadratic MSE function explored by steepest descent.

desired response is d_k. The constant μ is a design parameter that determines stability, rate of convergence, and the amount of noise in the weights resulting from using noisy, instantaneous gradients, each based on a single sample of data.

III. Adaptive Filtering with DCT/LMS

There are many newer algorithms different from LMS that have been described in the literature [3, 4], all claiming to converge faster. In some cases, they really do. Such methods, however, are invariably more complicated than LMS, and are usually more prone to numerical problems. Steepest descent could be very slow when the MSE hyperparaboloid is highly elliptical in cross section, a condition caused by correlation between components of the input **X**-vectors. Decorrelating these signals leads to a circular MSE function, which is easy to extremize. The newer algorithms in most cases decorrelate these signals. In place of tapped delay lines, one might use adaptive lattice structures [3, 4] to accomplish this purpose. Other methods based on Recursive Least Squares (RLS) [3] have been used, as well as methods based on Gram-Schmidt orthogonalization [5].

An orthogonalization technique based on the Discrete Fourier Transform (DFT) has been proposed by Shankar Narayan [6]. A similar technique based on the Discrete Cosine Transform (DCT) is now under investigation in our laboratory. The algorithm combines the DCT with LMS. An exhaustive treatment of the DCT can be found in a book by Rao and Yip [7]. An adaptive filter based on DCT/LMS is illustrated in Fig. 4. The signals from the tapped delay line are generally mutually correlated. Going through the DCT tends to decorrelate these signals, since the DCT outputs consist of signal components separated into individual frequency bands. Decorrelation with the DCT is not perfect, however, because there is always some "leakage" of signal components from one frequency bin to another. It is important to note that by using a tapped delay line prior to the DCT, we force each DCT input to have equal average power. In applications not using a tapped delay line, the DCT will be ineffective unless measures are taken to ensure its inputs are of equal power.

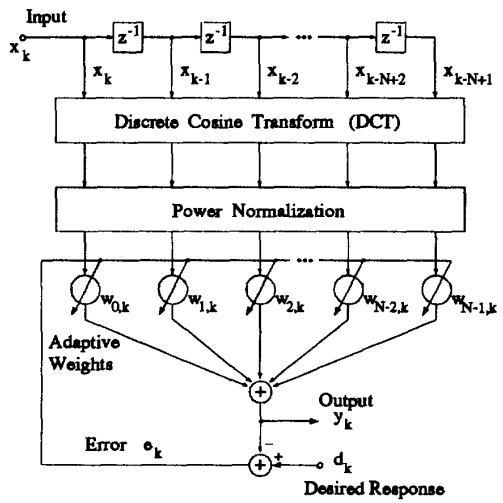
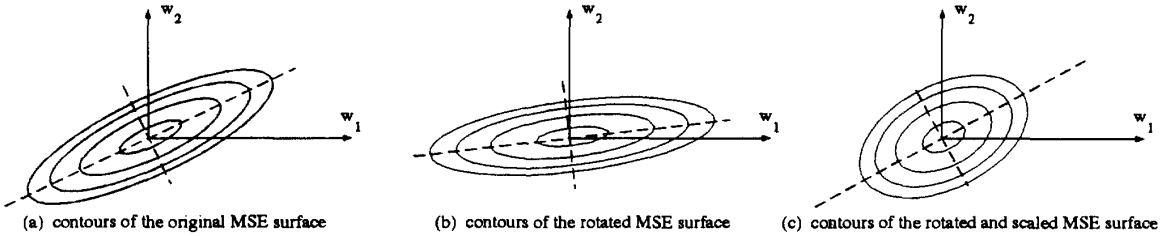


Fig. 4: The DCT/LMS Adaptive Filter.

After the DCT, Fig. 4 shows a power normalization stage. This is needed to make the MSE bowl approach a circular form. The decorrelation process rotates the hyperelipsoidal bowl so that its principal axes (the directions of the eigenvectors), approximately align with the axes of the space, i.e., the components of the weight vector. An illustration of this effect in two dimensions (2 weights) is shown in Fig. 5. Power normalization takes the rotated surface and scales the axes to make the bowl effectively circular, to make the eigenvalues of the hyperellipsoid approximately equal. Once this bowl is circular or approximately so, the LMS algorithm can be used with high effectiveness.

The DCT in Fig. 4 is a sliding window DCT requiring only $O(N)$ computations per iteration, where N is the number of taps in the filter. LMS and the power normalization stage are also $O(N)$, so the DCT does not represent a major computational burden.

We will compare the LMS adaptive filter with the DCT/LMS adaptive filter when both are performing the task of "plant identification," a task which is very important in control systems. The system to be controlled is called the plant. To control it, we first model it by feeding the same test input signal to the plant and to an adaptive filter, as illustrated in Fig. 6. The output of the plant is used as the desired response for the adaptive filter. The error is the difference between the plant output and the adaptive filter output. Adapting to minimize MSE causes the adaptive filter dynamics to become a best least



(a) contours of the original MSE surface (b) contours of the rotated MSE surface (c) contours of the rotated and scaled MSE surface

Fig. 5: Rotation and scaling of MSE bowl by action of the DCT followed by power normalization.

squares approximation to the plant's dynamics.

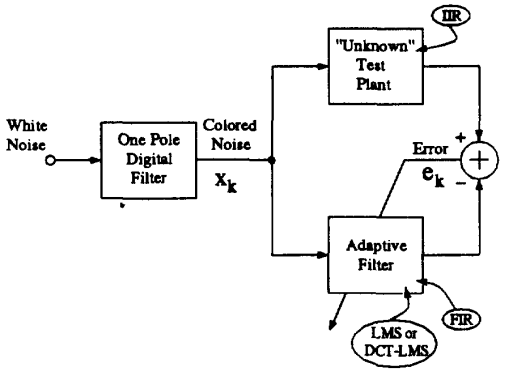


Fig. 6: Using adaptive filters for plant modeling.

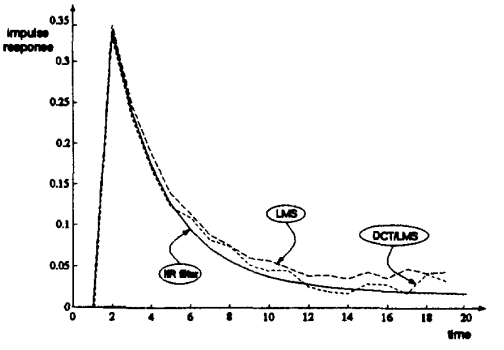


Fig. 7: Impulse response of the plant, of the LMS filter, and of the DCT/LMS filter.

The impulse response of the plant is plotted in Fig. 7. Since the plant has poles and zeros, its impulse response has infinite length, i.e. the plant is IIR (Infinite Impulse Response). Superposed are the converged impulse responses of the LMS filter and the DCT/LMS filter. The adaptive impulse responses of these two filters are of finite length, i.e. they are FIR (Finite Impulse Response) filters. The number of weights chosen for each of these filters was 20. The convergence speed of DCT/LMS, which explores the MSE bowl on a trajectory loosely approximating that of a stochastic Newton-type method, usually depends only weakly on the position of the starting point on a hyperellipsoid of constant MSE. In contrast, the convergence speed of LMS, which explores the MSE bowl by the method of steepest descent, can depend strongly on the position of the starting point on the constant MSE hyperellipsoid. In most cases, LMS will be faster converging than DCT/LMS from some starting points, and slower converging from other starting points.

The DCT/LMS algorithm for adaptive filters depends upon three very robust calculations: the DCT, power normalization, and the LMS algorithm. In a typical situation where the test input in Fig. 6 was a random first-order Markov process, the eigenvalue spread for the MSE function of the LMS filter was about 1000 to 1. For the same input, the eigenvalue spread for the MSE function of the DCT/LMS filter was about 2 to 1. Under best-case initial conditions, DCT/LMS will converge several hundred times faster than LMS. A plot of MSE vs. number of training cycles for LMS and for DCT/LMS from typical initial conditions is illustrated in Fig. 8. The DCT/LMS filter is a general purpose algorithm and can be used effectively in all the applications that follow.

Efficiency of steepest descent algorithms and Newton-type algorithms have been compared in [8]. The DCT/LMS algorithm has not been used very much in practice. We expect this will change as the algorithm's advantages become better appreciated.

IV. Adaptive Noise Cancelling

Separating a signal from additive noise is a common problem in signal processing. Fig. 9a shows a classical approach to this problem using optimal Wiener or Kalman filtering [9] The purpose of the optimal filter is to pass the signal s without distortion while stopping the noise n_0. In general, this cannot be done perfectly. Even with the best filter, the signal is distorted, and some noise goes through

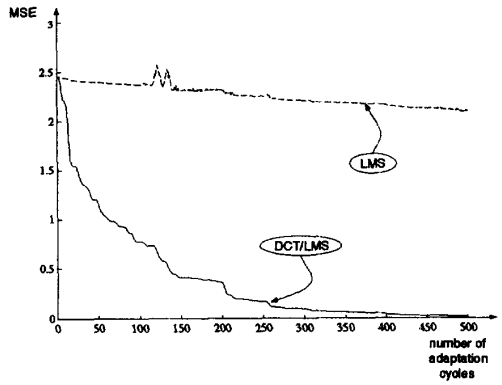


Fig. 8: Learning curves for the LMS filter and the DCT/LMS filter.

to the output.

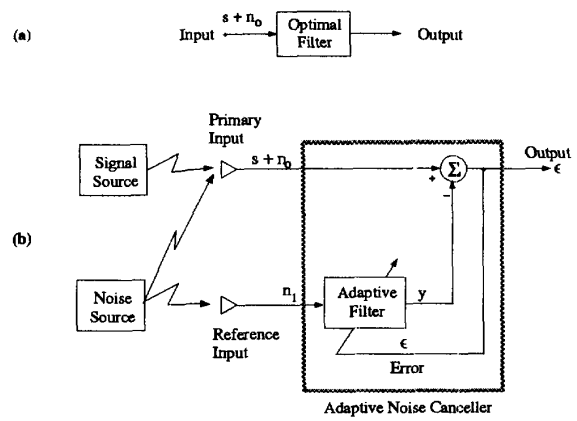


Fig. 9: Separation of signal and noise: (a) classical approach; (b) adaptive noise cancelling approach.

Fig. 9b shows another approach to the problem using adaptive filtering. This approach is viable only when an additional "reference input" is available containing noise n_1, which is correlated with the original corrupting noise n_0. In Fig. 9b, the adaptive filter receives the reference noise, filters it, and subtracts the result from the noisy "primary input," $s+n_0$. For this adaptive filter, the noisy input $s + n_0$ acts as the desired response. The "system output" acts as the error for the adaptive filter.

One might think that some prior knowledge of the signal s or of the noises n_0 and n_1 would be necessary before the filter could adapt to produce the noise-cancelling signal y. A simple argument will show, however, that little or no prior knowledge of s, n_0, or n_1 or of their interrelationships is required.

Assume that s, n_0, n_1 and y are statistically stationary and have zero means. Assume that s is uncorrelated with n_0 and n_1 and suppose that n_1 is correlated with n_0. The output is

$$\epsilon = s + n_0 - y \tag{2}$$

Squaring, one obtains

$$\epsilon^2 = s^2 + (n_0 - y)^2 + 2s(n_0 - y) \tag{3}$$

Taking expectations of both sides of Equation 2, and realizing that s is uncorrelated with n_0 and with y, yields

$$\begin{aligned} E[\epsilon^2] &= E[s^2] + E[(n_0 - y)^2] + 2E[s(n_0 - y)] \\ &= E[s^2] + E[(n_0 - y)^2] \end{aligned} \tag{4}$$

Adapting the filter to minimize $E[\epsilon^2]$ will not affect the signal power $E[s^2]$. Accordingly, the minimum output power is

$$E_{min}[\epsilon^2] = E[s^2] + E_{min}[(n_0 - y)^2] \tag{5}$$

When the filter is adjusted so that $E[\epsilon^2]$ is minimized, $E[(n_0-y)^2]$ is therefore also minimized. The filter output y is then a best least-squares estimate of the primary noise n_0. Moreover, when $E[(n_0-y)^2]$ is minimized, $E[(\epsilon-y)^2]$ is also minimized, since, from Equation 1,

$$(\epsilon - s) = (n_0 - y) \tag{6}$$

Adjusting or adapting the filter to minimize the total output power is tantamount to causing the output ϵ to be a best least-squares estimate of the signal s for the given structure and adjustability of the adaptive filter and for the given reference input.

A. Fetal Electrocardiography

There are many practical applications for adaptive noise cancelling techniques. One involves cancelling interference from the mother's heart when attempting to record clear fetal electrocardiograms. Fig. 10 shows the location of the fetal and maternal hearts and the placement of the input leads. The abdominal leads provide the primary input (containing fetal ECG and interfering maternal ECG signals), and the chest leads provide the reference input (containing pure interference, the maternal ECG). Fig. 11 shows the results. The maternal ECG from the chest leads was adaptively filtered and subtracted from the abdominal signal, leaving the fetal ECG. This was an interesting problem since the fetal and maternal ECG signals had spectral overlap. The two hearts were electrically isolated and worked independently, but the second harmonic frequency of the maternal ECG was close to the fundamental of the fetal ECG. Ordinary filtering techniques would have great difficulty with this problem.

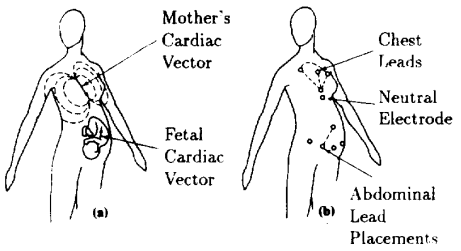


Fig. 10: Cancelling maternal heartbeat in fetal electrocardiography: (a) cardiac electric field vectors of mother and fetus; (b) placement of leads.

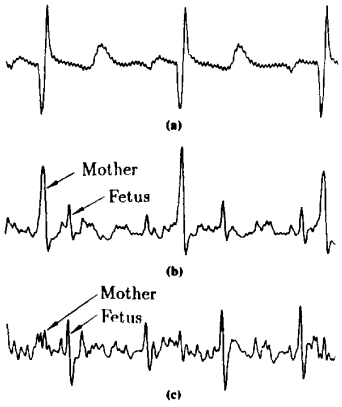


Fig. 11: Result of fetal ECG experiment (bandwidth, 3-35 Hz; sampling rate, 256 Hz): (a) reference input (chest lead); (b) primary input (abdominal lead); (c) noise canceller output.

B. Adaptive Echo Cancelling

Echo is a natural phenomenon in long-distance telephone circuits because of amplification in both directions and series coupling of telephone transmitters and receivers at each end of the circuit. Echo suppressors, used to break the feedback, give one-way communication to the party speaking first. To avoid switching effects and to permit simultaneous two-way transmission of voice and data, adaptive echo cancellers are replacing echo suppressors worldwide (see Fig. 12).

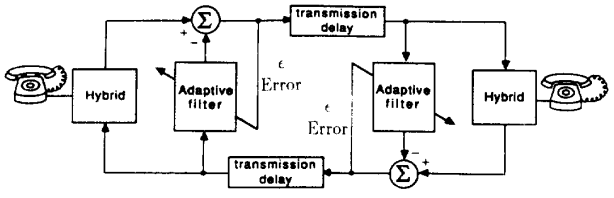


Fig. 12: Long-distance system with adaptive echo cancellation.

In Fig. 12, the delay boxes represent transmission delays in the long distance line. Note that separate circuits are normally used in each direction because the repeater amplifiers used to overcome transmission loss are one-way devices. Hybrid transformers prevent incoming signals from coupling through the telephone set and passing as outgoing signals. Hybrids are balanced to do this by designing them for the average, local telephone circuit. Since each local circuit has its own length and electrical characteristics, the hybrid cannot do its job perfectly. Using an adaptive filter to cancel any incoming signal that might leak through the hybrid eliminates the possibility of echo. The circuit of Fig. 12, used in MODEMs, works well allowing simultaneous two-way communication without echo.

V. Inverse Plant Modeling

Fig. 6 showed use of an adaptive filter for direct modeling of an unknown plant to obtain a close approximation to its impulse and frequency responses. By changing the configuration, it is possible to use the adaptive filter for inverse modeling to obtain the reciprocal of the unknown system's transfer function. The idea is illustrated in Fig. 13. The output of the unknown system is the input to the adaptive filter. The unknown system's input delayed by Δ time units is the desired response of the adaptive filter. For simplicity, assume that Δ is set to zero delay. To make the error small, the cascade of the unknown system and the adaptive filter needs a unity transfer function. Therefore, when using an adaptive algorithm to make the error small, the adaptive filter develops a transfer function which is the inverse of the unknown system's transfer function.

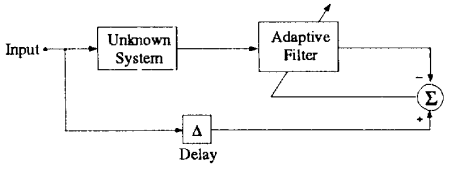


Fig. 13: Inverse modeling.

The plant can be assumed to have poles and zeros. An inverse would need to have zeros where the plant had poles, and poles where the plant had zeros. Making an inverse would be straightforward except for the case of a nonminimum-phase plant. It would seem that the inverse would need to have unstable poles. This would be true if the inverse were causal. If the inverse could have both causal and noncausal components however, then a two-sided stable inverse would exist for all linear time-invariant plants in accord with the theory of two-sided

Laplace transforms. A causal FIR filter can approximate a delayed version of the two-sided plant inverse, and an adaptive FIR filter can self-adjust to this function. To achieve this objective with a nonminimum phase plant, the delay Δ needs to be chosen appropriately, although the choice is generally not critical.

Applications for inverse modeling exist in the field of adaptive control, in geophysical signal processing, where it is called "deconvolution," and in telecommunications for channel equalization.

VI. Adaptive Inverse Control

An application of the inverse modeling idea to adaptive inverse control [10] is illustrated with the system of Fig. 14. The weights of the inverse model of the plant are copied into an adjustable controller. The command input to the control system is fed to this controller, whose output is used to drive the plant. After convergence of the inverse filter, the output of the plant follows the input signal delayed by Δ time samples. Experimental confirmation is given by the time plots in Fig. 15.

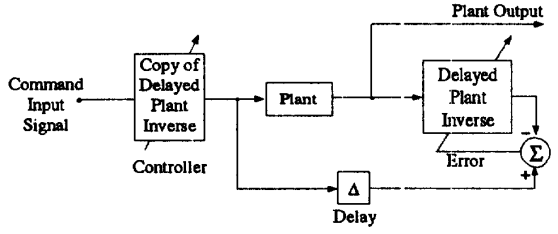


Fig. 14: An adaptive inverse control system.

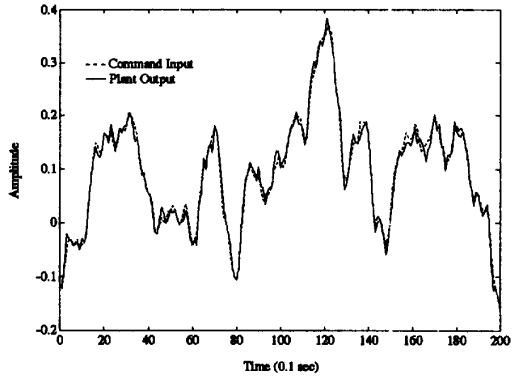


Fig. 15: Tracking between command input and plant output.

The plant in Fig. 14 is single-input, single-output, and linear. The inverse is linear when it is converged. The plant and the controller are commutable. The system error, the difference between the two plots of Fig. 15, is identical to the error of the adaptive inverse modeling process. Feedback in the LMS adaptive process therefore serves to keep the system error low. This is a completely different approach from the use of feedback to control error in a servo.

VII. Adaptive Equalization

Telephone channels, radio channels, and even fiber optic channels can have non-flat frequency responses and nonlinear phase responses in the signal passband. Sending digital data at high speed through these channels often results in a phenomenon called "intersymbol interference," caused by signal pulse smearing in the dispersive medium. Equalization in data MODEMs combats this phenomenon by filtering incoming signals. A MODEM's adaptive filter, by adapting itself to become a channel inverse, can compensate for the irregularities in channel magnitude and phase response.

The adaptive equalizer in Fig. 16 consists of a tapped delay line with an adaptive linear combiner connected to the taps. Deconvolved signal pulses appear at the weighted sum, which is quantized to provide a binary output corresponding to the original binary data transmitted through the channel. The adaptive linear combiner and its quantizer comprise a single adaptive neuron. Any least-squares algorithm can adapt the weights, but the telecommunications industry uses the LMS algorithm almost exclusively.

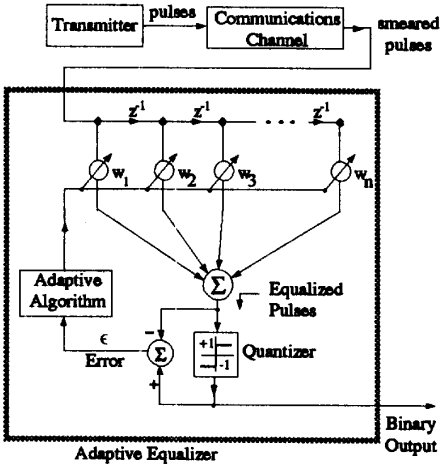


Fig. 16: Adaptive channel equalizer with decision-directed learning

In operation, the weight at a central tap is generally fixed at unit value. Initially, all other weights are set to zero so that the equalizer has a flat frequency response and a linear phase response. Without equalization, tele-

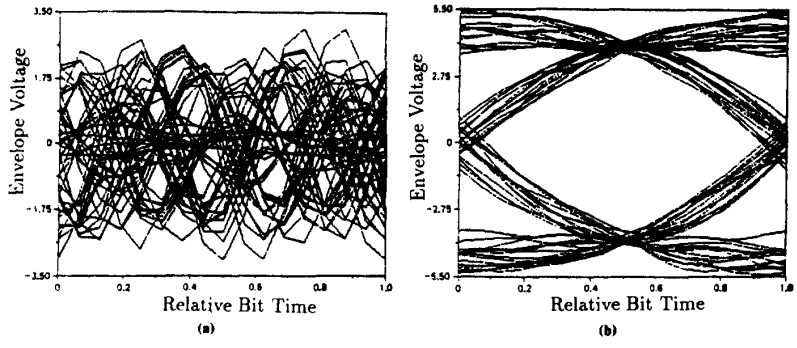


Fig. 17: Eye patterns produced by overlaying cycles of the received waveform: (a) before equalization; (b) after equalization.

phone channels can provide quantized binary outputs that reproduce the transmitted data stream with error rates of 10^{-1} or less. As such, the quantized binary output can be used as the desired response to train the neuron. It is a noisy desired response initially. Sporadic errors cause adaptation in the wrong direction, but on average, adaptation proceeds correctly. As the neuron learns, noise in the desired response diminishes. Once the adaptive equalizer converges, the error rate will typically be 10^{-6} or less. The method, called "decision- directed" learning, was invented by Robert W. Lucky of AT&T Bell Labs [11].

Fig. 17a shows the analog response of a telephone channel carrying high-speed binary pulse data. Fig. 17b shows an "eye" pattern, which is the same signal after going through a converged adaptive equalizer. Equalization opens the eye and allows clear separation of $+1$ and -1 pulses. Using a MODEM with an adaptive equalizer enables transmitting several times as much data through the same channel with the same reliability as without equalization.

Integrated services digital network (ISDN), a new concept now in development and deployment, makes high-speed digital communication possible through ordinary local copper telephone circuits. ISDN requires both adaptive equalization and adaptive echo cancelling at each line termination. The number of adaptive filters used in the world's telecommunications plant is massive.

VIII. Adaptive Linear Prediction

One can estimate future values of time-correlated discrete-time signals from present and past input samples. Wiener [9] has developed optimal linear least-squares filtering techniques for signal prediction. When the signal's autocorrelation function is known, Wiener's theory yields the impulse response of the optimal filter. More often than not, however, the autocorrelation function is unknown and may be time-variable. One could use a correlator to measure the autocorrelation function and plug this into Wiener theory to get the optimal impulse response, or one could get the optimal prediction filter directly by adaptive filtering. Fig. 18 illustrates the latter approach.

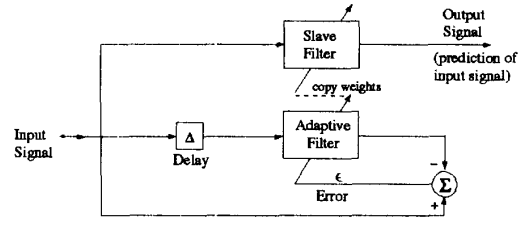


Fig. 18: An adaptive linear statistical predictor.

In this figure, the input signal delayed by Δ time units is fed to an adaptive filter. The undelayed input serves as the desired response for this adaptive filter. The filter weights adapt and converge to produce a best least-squares estimate of the present input signal, given an input that is this very signal delayed by Δ. The optimal weights are copied into a "slave filter" whose input is undelayed and whose output therefore is a best least-squares prediction of the input Δ time units into the future.

IX. Nonlinear Filtering and Prediction

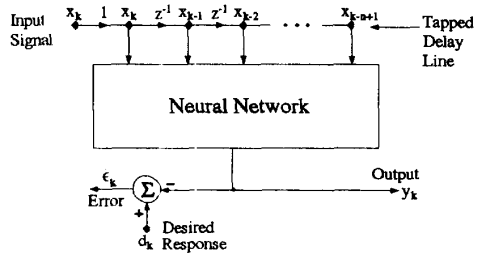


Fig. 19: Adaptive nonlinear filter.

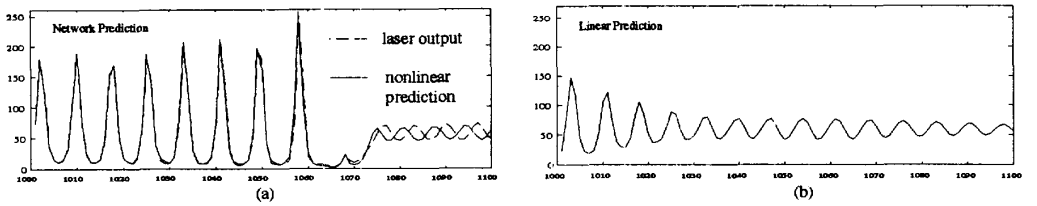


Fig. 21: Time series predictions: (a) Closed loop neural network prediction does very well. (b) Iterated 25th order linear predictor does poorly.

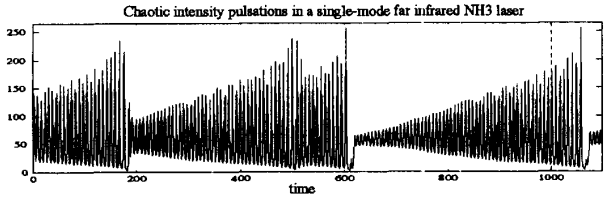


Fig. 20: Chaotic light intensity pulsations on an NH_3 laser.

So far, we have focused on linear filters that can be represented by the convolution $y_k = \sum_{i=1}^{n} w_i x_{k-i+1}$. More recently, neural networks have been used to form nonlinear mappings of past states to the output: $y_k = \mathcal{N}[\mathbf{W}, x_k, x_{k-1}, ...x_{k-n+1}]$. $\mathcal{N}$ represents a standard feedforward network with weight values $\mathbf{W}$ [12]. The inputs to the network are fed from a tapped delay line as illustrated in Fig. 19. This structure forms an adaptive nonlinear filter and may replace the adaptive linear filter in each of the applications previously discussed. A variant of the standard feedforward network is the Finite Impulse Response Network [13] which replaces the single weight for each synaptic connection in the network by an adaptive linear FIR filter. The adaptive linear filter discussed in this paper is thus used to provide dynamic interconnectivity between all processing units in the network. We have found this FIR network to be advantageous for filtering temporal data.

To illustrate the potential of a nonlinear filter, we provide an example in nonlinear prediction. The series to predict is shown in Fig. 20 and corresponds to chaotic intensity pulsations on an NH_3 laser measured in a laboratory experiment. This data was distributed as part of the *Santa Fe Institute Time Series Prediction and Analysis Competition* [14], a international competition. 1000 samples of the sequence were provided. The goal was to predict the next 100 samples.

An FIR network was trained on the 1000 points to form single step predictions using a configuration similar to Fig. 19. Single step predictions were then fed back as input to the network to form a closed loop dynamic system which could be iterated forward in time to achieve the desired 100 prediction points. The most accurate 100-step prediction is shown in Fig. 21a, along with the actual series continuation. A prediction using a 25th order linear autoregression is shown in Fig. 21b. The correspondence between the nonlinear prediction and the actual laser output is impressive.

References

[1] B. Widrow. Adaptive filters. In R. Kalman and N. DeClaris, editors, *Aspects of Network and System Theory*, pages 563–587. Holt, Rinehart, and Winston, New York, 1971.

[2] B. Widrow and M. Hoff. Adaptive switching circuits. In *1960 IRE WESCON Convention Record*, volume 4, pages 96–104. IRE, New York, 1960.

[3] S. Haykin. *Adaptive Filter Theory*. Prentice-Hall, Englewood Cliffs, NJ, 2 edition, 1991.

[4] B. Widrow and S. D. Stearns. *Adaptive Signal Processing*. Prentice-Hall, Englewood Cliffs, NJ, 1985.

[5] F. Ling. Efficient least-squares lattice algorithms based on givens rotation with systolic array implementations. In *Proc. ICASSP-89*, pages 1290–2193, Glasgow, Scotland, 1989.

[6] S. S. Narayan and A. M. Peterson. Frequency domain least-mean-square algorithm. *Proc. IEEE*, 69(1):124–126, January 1981.

[7] K. Rao and P. Yip. *Discrete Cosine Transform*. Academic Press, San Diego, 1990.

[8] B. Widrow and E. Walach. On the statistical efficiency of the LMS algorithm with nonstationary inputs. *IEEE Transactions on Information Theory*, IT-30(2):211–222, March 1984.

[9] T. Kailath. *Lectures on Wiener and Kalman Filtering*. Springer Verlag, New York, 1981.

[10] B. Widrow. Adaptive inverse control. In *Proceedings of the 2nd International Federation of Automatic Control Workshop*, pages 1–5, Lund, Sweden, July 1-3 1986.

[11] R. W. Lucky. Automatic equalization for digital communication. *Bell Syst. Tech. J.*, 44:547–588, April 1965.

[12] D. E. Rumelhart and J. L. McClelland, editors. *Parallel Distributed Processing*, volume 1 and 2. The MIT Press, Cambridge, MA, 1986.

[13] E. A. Wan. Temporal backpropagation for FIR neural networks. In *International Joint Conference on Neural Networks*, volume I, pages 575–580, San Diego, CA, June 1990.

[14] A. Weigend and N. Gershenfeld, editors. *Proceedings of the NATO Advanced Workshop on Time Series Prediction and Analysis (Sante Fe, NM, MAY 14-17 1992)*. Addison-Wesley, 1993.