from:

Proceedings IEEE workshop on neural networks for signal processing

Princeton, NJ, 1991

ADALINE WITH ADAPTIVE RECURSIVE MEMORY

Bert De Vries, Jose C. Principe, and Pedro Guedes de Oliveira*

Department of Electrical Engineering
University of Florida
Gainesville, FL 32611
principe@brain.ee.ufl.edu

*Departamento Eletronica/INESC
Universidade de Aveiro
Aveiro, Portugal

Abstract - We present a generalization of Widrow's adaptive linear combiner with an adaptive recursive memory. Expressions for memory depth and resolution are derived. The LMS procedure is extended to adapt the memory depth and resolution so as to match the signal characteristics.The particular memory structure, gamma memory, was originally developed as part of a neural net model for temporal processing.

INTRODUCTION

Although infinite impulse response (IIR) systems are more powerful for modeling, identification, and signal processing, the subclass of finite impulse response (FIR) filters is almost exclusively utilized in adaptive signal processing. The main reason is centered in the difficulty of ensuring stability during adaptation of IIR systems. Moreover, gradient descent update rules are not guaranteed to find global minima in the non-convex error surfaces of IIR systems.

Feedforward systems, on the other hand, are always stable. Yet the tapped delay line memory structure implies that the depth of memory is always of the same order as the number of adaptive weights in an FIR structure. For some real world signal environments, in particular biological signals, this is a severe modeling drawback. Thus, the choice between a recurrent or feedforward processing or modeling system can be a difficult one.

In this paper we present the gamma memory - a structure that is characterized by a restricted IIR architecture. The uncoupling of memory depth from filter order is inherited from recurrent systems. Yet, the stability condition for the gamma memory will prove to be trivial.

Consider the filter network defined by -

$$y(n) = \sum_{k=0}^{K} w_k \sum_{m=0}^{\infty} e(m) g_k(n-m)$$

$$= \sum_{k=0}^{K} w_k [g_k(n) \bullet e(n)] \qquad (1)$$

$$= \sum_{k=0}^{K} w_k x_k(n)$$

where *e(n)* is an input sequence, *y(n)* is an output sequence, the tap variables $x_k(n)$, $k = 0,..,K$, are the convolution of *e(n)* and a delay kernel $g_k(n)$, w_k is a set of adaptive parameters and $\bullet$ denotes the convolution operator. We assume that the kernels $g_k(n)$ are *causal* and *normalized*, that is, $g_k(n) = 0$ for $n < 0$ and $\sum_{n=0}^{\infty} g_k(n) = 1$. When additionally the kernels admit a *recurrence* relation of the form -

$$g_k(n) = g(n) \bullet g_{k-1}(n) \quad , k = 1,..,K, \qquad (2)$$

then it is possible to generate the tap variables $x_k(n)$ recursively by $x_k(n) = g(n) \bullet x_{k-1}(n)$ or in the *z*-domain by -

$$X_k(z) = G(z) X_{k-1}(z) \ . \qquad (3)$$

We refer to the structure described by (1) and (3) as the *generalized feedforward filter*. The tap-to-tap transfer function *G*(*z*) is the (*generalized*) *delay operator*. This structure is depicted in Figure 1.

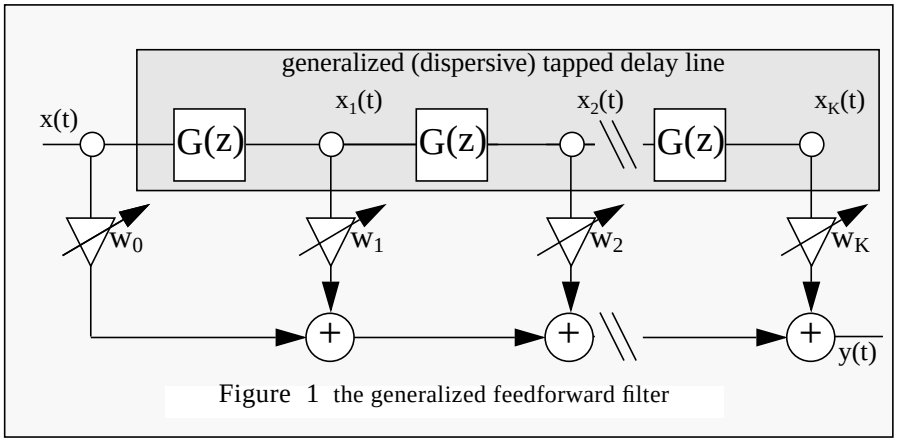

Figure 1 the generalized feedforward filter

Note that we can write the transfer function $H(z) \equiv \frac{Y(z)}{X(z)}$ of the generalized feedforward filter as follows -

$$H(z) = \sum_{k=0}^{K} w_k [G(z)]^k \quad . \qquad (4)$$

It follows from (4) that *H(z) is stable whenever G(z) is stable*.

For $g_k(n) = \delta(n-k)$, this structure reduces to Widrow's *Adaline* [1]. The past of $x(t)$ is represented in the tap variables $x_k(t)$. Although conventional digital signal processing structures are built around the tapped delay line ($G(z) = z^{-1}$), we have observed that alternative delay operators may lead to better filter performance. In general, the optimal memory structure $G(z)$ is a function of the input signal characteristics as well as the goal of the filter operation. This observation has led us to consider *adaptive* delay operators $G(z)$.

In this paper we consider *gamma delay kernels* as defined by -

$$g_k(n) = \binom{n-1}{k-1} \mu^k (1-\mu)^{n-k} U(n-k) \quad , k = 1,..,K, \qquad (5)$$

where $U(n)$ is the unit step function. The gamma delay kernels were originally developed in continuous time as part of a neural net model for temporal processing [2]. We showed that - by transformation $s = \frac{z-1}{T_s}$ - the impulse response of the continuous time gamma filter can be written as $h(t) = \sum_{k=0}^{K} w_k g_k(t)$, where $g_k(t) = \frac{\mu^k}{(k-1)!} t^{k-1} e^{-\mu t}$, $k = 1,...,K$, and $g_0(t) = \delta(t)$. The functions $g_k(t)$ are the integrands of the (normalized) gamma function. Hence the name gamma model for structures that utilize tap variables of type $x_k(t) = (g_k \bullet x)(t)$ to store the past of x(t).

Z-transformation of (5) yields -

$$G_k(z) = \left(\frac{\mu}{z-(1-\mu)}\right)^k \qquad (6)$$

We define the *gamma delay operator* as $G(z) = \frac{\mu}{z-(1-\mu)}$. The gamma delay operator can be interpreted as a leaky integrator with loop gain $1-\mu$. The parameter μ controls the memory depth and resolution. Note that stability is guaranteed when $0 < \mu < 2$.

In the following, we refer to an adaline structure with gamma memory as adaline(μ). Note that Widrow's adaline is equivalent to adaline(1). Adaline(μ) is shown in Figure 2.

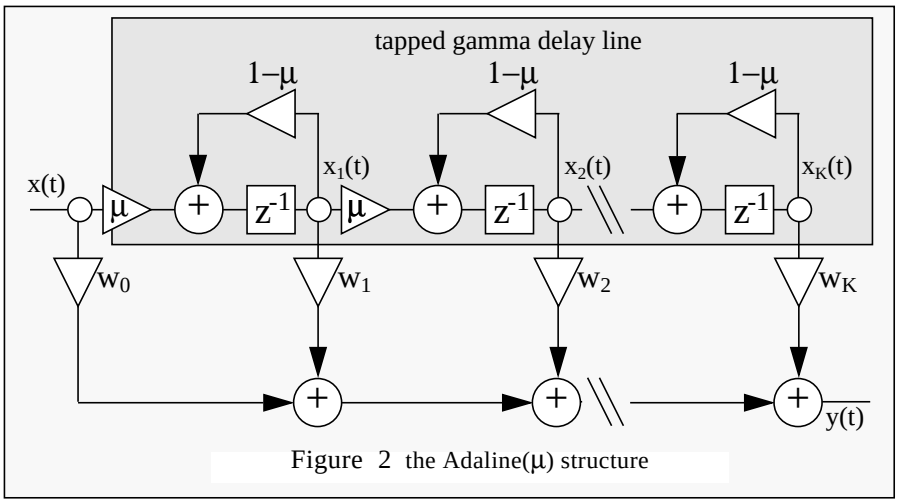

Figure 2 the Adaline(μ) structure

CHARACTERISTICS OF GAMMA MEMORY

An extensive account of the properties of the gamma memory structure has been presented in [3]. Here we summarize a discussion regarding the memory depth and we present an LMS learning procedure for adaline(μ).

Memory Depth versus Filter Order

Although the impulse response $g_k(n) = Z^{-1}\{G_k(z)\}$ of the *k*th tap of the gamma memory extends to infinite time for $0 < \mu < 1$, it is possible to formulate a mean memory depth for a given memory structure $g_k(n)$. Let us define the *mean sampling time* n_k for the *k*th tap as -

$$n_k = \sum_{n=0}^{\infty} n g_k(n) = Z\{n g_k(n)\}\Big|_{z=1} = -z\frac{dG_k(z)}{dz}\Big|_{z=1} = \frac{k}{\mu} \quad . \qquad (7)$$

We also define the *mean sampling period* Δn_k (at tap *k*) as $\Delta n_k = n_k - n_{k-1} = \frac{1}{\mu}$. The *mean memory depth* D_k for a gamma memory of order *k* then becomes -

$$D_k = \sum_{i=1}^{k} \Delta n_i = n_k - n_0 = \frac{k}{\mu} . \qquad (8)$$

If we define the *resolution* R_k as $R_k = \frac{1}{\Delta n_k} = \mu$, the following formula arises which is of fundamental importance for the characterization of the gamma memory structure[1] -

$$K = D \times R \qquad (9)$$

Equation (9) reflects the possible trade-off of resolution versus memory depth in a memory structure for fixed dimensionality *K*. Such a trade-off is not possible in a non-dispersive tapped delay line, since the fixed choice of $\mu = 1$ sets the depth and resolution to $D = K$ and $R = 1$ respectively. However, in the gamma memory, depth and resolution can be adapted by variation of μ. The choice $\mu = 1$ represents a memory structure with maximal resolution and minimal depth. For this case, the order *K* and depth *D* of the memory are equal. In the adaline structure, the number of adaptive parameters w_k equals o the number of taps (*K*+1). Thus, when $\mu = 1$, the number of weights equals the memory depth. Very often this coupling leads to overfitting of the data set (using parameters to model the noise). The parameter μ provides a means *to uncouple the memory order and depth*.

As an example, assume a signal whose dynamics are described by a system with 5 parameters and maximal delay 10, that is, $y(t) = f(x(t - n_i), w_i)$ where $i = 1,..,5$, and $max_i(n_i) = 10$. If we try to model this signal with an adaline structure, the choice $K = 10$ leads to overfitting while $K < 10$ leaves the network unable to incorporate the influence of $x(t\text{-}10)$. In an adaline with gamma memory network, the choice $K = 5$ and $\mu = 0.5$ leads to 5 free network parameters and mean memory depth of 10, obviously a better compromise.

LMS Adaptation

In this section we present the least mean square (LMS) adaptation update rules for the parameters w_k and μ. Let the performance of the system be measured by the *total error E*, defined as -

$$E \equiv \sum_{n=0}^{T} E_n = \sum_{n=0}^{T} \frac{1}{2}\varepsilon^2(n) = \sum_{n=0}^{T} \frac{1}{2}(d(n) - y(n))^2 \qquad (10)$$

where $d(n)$ is a *target signal*. We first expand for w_k, yielding-

$$\Delta w_k(n) = -\eta \frac{\partial E_n}{\partial w_k} = \eta \varepsilon(n) \frac{\partial y(n)}{\partial w_k} = \eta \varepsilon(n) x_k(n) \qquad (11)$$

Similarly, the update equation for μ evaluates to -

---

1. We dropped the subscript k when $k = K$.

$$\Delta\mu = -\eta\frac{\partial E_n}{\partial\mu} = \eta\varepsilon(n)\sum_k w_k\alpha_k(n) \qquad (12)$$

where $\alpha_k(n) \equiv \dfrac{\partial x_k(n)}{\partial\mu}$. The gradient signal $\alpha_k(n)$ can be computed on-line by differentiating (Eq.6) leading to -

$$\alpha_0(n) = 0$$

$$\alpha_k(n) = (1-\mu)\alpha_k(n-1) + \mu\alpha_{k-1}(n-1) + [x_{k-1}(n-1) - x_k(n-1)] \quad \text{, for } k = 1,..,K. \qquad (13)$$

A DSP INTERPRETATION OF ADALINE(μ)

In order to quantify the signal processing effects of the gamma kernels, we define an auxiliary variable γ as -

$$\gamma = \frac{z-(1-\mu)}{\mu} \qquad (14)$$

Substitution of (14) in the *z*-transform of $x(n)$ yields the following expression for the γ–transform -

$$X(\gamma) = \sum_{n=0}^{\infty} \mu^{-n} x[n]\left\{\gamma - \frac{\mu-1}{\mu}\right\}^{-n} , \qquad (15)$$

that is, the (one-sided) Laurent series of the sequence $\mu^{-n}x(n)$ evaluated at $\gamma_0 = \dfrac{\mu-1}{\mu}$. The relation between the *z*- and γ-plane is depicted in Figure 3. Note that adaline(μ) is a *feedforward* filter centered around the ideal delays γ^{-1} in the γ-domain.

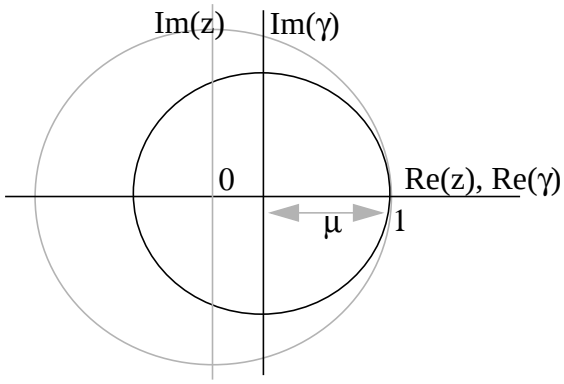

Figure 3 *relation between z- and γ-transform*

Therefore it is straightforward to analyze and design gamma memory structures in the γ-domain as FIR filters, and obtain the corresponding pole zero maps and impulse responses.

EXPERIMENTAL DATA

The performance of adaline(μ) versus adaline(1) has been tested for various experimental protocols. In this section we present data on a prediction and system identification experiment.

prediction/noise reduction of sinoids in noise

We constructed an input signal consisting of a sum of sinusoids, contaminated by gaussian noise. Specifically, e(*t*) was described by -

$$e(t) = sin(\pi(0.06t + 0.1)) + 3sin(\pi(0.12t + 0.45)) + 1.5sin(\pi(0.2t + 0.34)) + sin(\pi(0.4t + 0.67)) + N(0, 1) \quad (16)$$

where $N(0,1)$ denotes zero mean gaussian noise with unit variance. This signal is shown in Figure 4a. The desired output signal was the next sample of the sum of sinusoids. Hence, the processing problem involves a combination of prediction and noise reduction. The processing system was an adaline(μ). We trained the network using the LMS adaptation rules as derived in this paper. Both the training set and the validation set consisted of 300 samples. The parameters were only adapted during a run on the training data set. The tracks of the memory parameter μ are shown in Figure 4c. After each run on the training data, we ran the system on the validation data set and measured the normalized performance index $E = \frac{var[\varepsilon(n)]}{var[d(n)]}$. A run over the training data set followed by a run over the validation data is called an epoch. If the performance index for the validation data set increases for 4 consecutive epochs, we assume that the system parameters have converged and thus we stop training (Figure 4d). In Figure 4b we plot the normalized performance index after convergence as a function of memory parameter μ. μ was parametrized over the domain [0,1] using a step size $\Delta\mu = 0.1$. The system performance is measured for memory orders $K = 1$ through $K = 5$. Note that $\mu = 1$ refers to Widrow's adaline structure. The graph clearly shows that adaline(μ) outperforms adaline(1). Even for low order $K = 1$, adaline(0.1) performs better than a fifth order adaline(1).

Elliptic Filter System Identification

In this section we present numerical simulation results when adaline(μ) is used in a system identification configuration. The system to be identified is the third order elliptic low pass filter described by[1] -

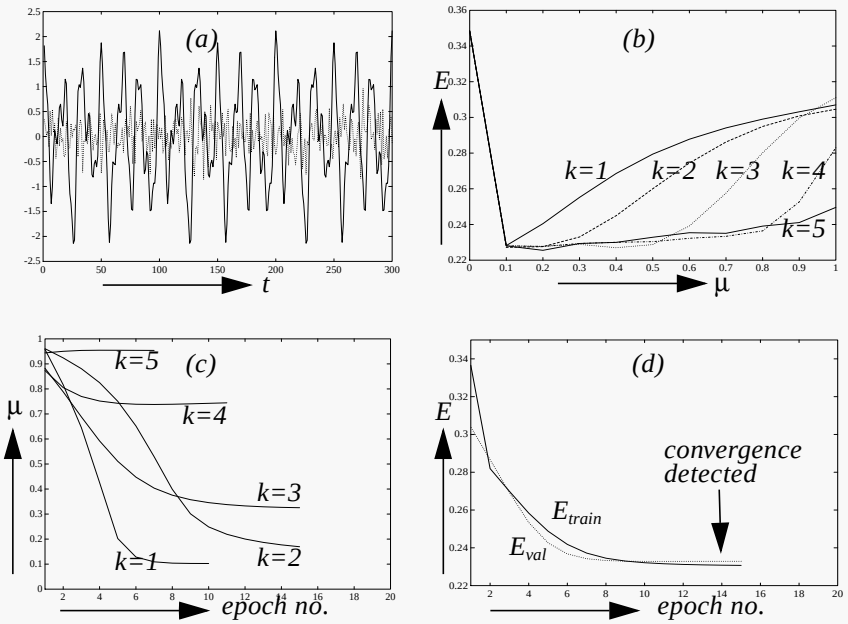

Figure 4 results for prediction/noise removal experiment. The processing system is adaline(μ). The number of prediction steps is 1. (a) input signal consists of a sum of four sinusoids contaminated with gaussian noise. SNR = 10 dB. (b) Normalized error criterion (mse) after convergence versus μ. (c) The tracks of μ during training. (d) The performance index over the training set (E_{train}) and the validation set (E_{val}) versus the epoch number.

$$H(z) = \frac{0.0563 - 0.0009z^{-1} - 0.0009z^{-2} + 0.0563z^{-3}}{1 - 2.1291z^{-1} + 1.7834z^{-2} - 0.5435z^{-3}} \quad . \qquad (17)$$

In Figure 5 we show the performance index after convergence. Note that the optimal memory depth $D_{opt} \equiv \frac{K}{\mu_{opt}} \approx 5$ is constant for different memory orders. The graph shows that it is irrelevant to use memory orders higher than K = 5 for this identification problem. In fact, when K = 5, adaline(1) performs as well as K = 3 for adaline(0.6). However, we still prefer K = 3, since this structure has 5 free parameters (K+1 weights w_k plus μ) whereas adaline uses 6 parameters.

1. This filter has been described in [4] on pg.226

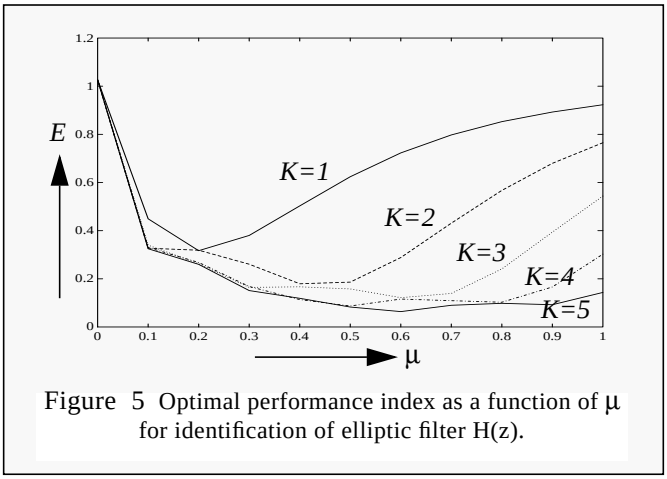

Figure 5 Optimal performance index as a function of μ for identification of elliptic filter H(z).

Parsimony in the number of free parameters provides adaline(0.6) with better modeling (generalization) characteristics.

We have experimented with several signals (sinusoids in noise, Feigenbaum map, electroencephalogram (EEG)) for various processing protocols (prediction, system identification, classification). Invariably the optimal memory structure[1] was obtained for $\mu < 1$.

CONCLUSIONS

We presented a new memory structure, the gamma memory, and applied it in the adaptive linear combiner. The gamma memory structure is characterized by two properties in comparison to fully recurrent networks (IIR filters). First, the loops in the gamma filter are local with respect to the taps. As a result, stability is easily controlled in the gamma memory. Secondly, the memory parameter μ is global with respect to the memory taps. This is advantageous in an adaptive environment, since a single parameter controls the trade-off between memory depth and resolution. It is worth mentioning that the application of gamma memory is not limited to the adaptive linear combiner. The same principle for short term memory has applications in various non-linear neural network configurations.

Acknowledgments

This work was partially supported by NSF grant ECS-8915218. The stay of the third author (P.G.O.) at the University of Florida has been partially supported by JNICT.

1. The optimal memory structure is defined as the structure of lowest dimensionality that minimizes the performance index *E*.

References

[1] B. Widrow and S.D. Stearns, Adaptive Signal Processing. Prentice Hall, Inc., Englewood Cliffs, NJ, 1985.

[2] B. De Vries and J.C. Principe, A theory for neural nets with time delays. in Proceedings of NIPS90, Denver, CO, 1991.

[3] J.C. Principe J.C. and B. De Vries, The Gamma Filter - A New Class of Adaptive IIR Filters with Restricted Feedback, Submitted to IEEE Transactions on Signal Processing, June 1991.

[4] A. Oppenheim and R.J. Schafer, Digital Signal Processing, Prentice Hall, Inc., Englewood Cliffs, NJ, 1975.