SIMPLE ALGORITHMS FOR FAST ADAPTIVE FILTERING

Françoise Beaufays and Bernard Widrow *
Department of Electrical Engineering
Stanford University

ABSTRACT

The LMS algorithm invented by Widrow and Hoff in 1959 is the simplest, most robust, and one of the most widely used algorithms for adaptive filtering. Unfortunately, its convergence rate is highly dependent upon the conditioning of the autocorrelation matrix of its inputs: the higher the input eigenvalue spread, the slower the convergence of the adaptive weights.

This problem can be overcome by preprocessing the inputs to the LMS filter with a fixed data-independent transformation that, at least partially, decorrelates the inputs. Typically, the preprocessing consists of a DFT or a DCT transformation followed by a power normalization stage. The resulting algorithms are called DFT-LMS and DCT-LMS. This technique is to be contrasted with more traditional approaches such as recursive least squares algorithms, where an estimate of the inverse input autocorrelation matrix is used to improve the filter convergence speed.

After placing DFT-LMS and DCT-LMS into context, we propose three different approaches to explain the algorithms both intuitively and analytically. We discuss the convergence speed improvement brought by these algorithms over conventional LMS, and we make a short analysis of their computational cost.

INTRODUCTION

It is well know from the theory of LMS (Widrow, 1985) that the mean square error of an adaptive filter trained with the LMS algorithm decreases over time as a sum of exponentials whose time constants are inversely proportional to the eigenvalues of the autocorrelation matrix of the inputs to the filter. This means that small eigenvalues create slow convergence modes in the MSE function. Large eigenvalues, on the other hand, put a limit on the maximum learning rate that can be chosen without encountering stability problems (Widrow,1985). It results from these two counteracting factors that the best convergence properties are obtained when all the eigenvalues are equal, that is when the input autocorrelation matrix is proportional to the identity matrix. In this case, the inputs are perfectly uncorrelated and have equal power; in other words, they are samples of a white noise process. As the eigenvalue spread of the input autocorrelation matrix increases, the convergence speed of LMS deteriorates.

DFT-LMS and DCT-LMS offer a solution to this problem. By preprocessing the input data with a well-chosen but fixed transformation that does not depend on the inputs, and with a simple power normalization stage, they cause the input eigenvalues of the LMS filter to cluster around one, and speed up the convergence of the adaptive weights.

Recursive least squares algorithms also decorrelate the inputs by preprocessing them, but they use to that effect an estimate of the inverse autocorrelation matrix, which thus depends on the actual inputs.

The performance of the algorithms based on data-independent transformations clearly depends on the orthogonalizing capabilities of the transform used. No general proof exists that demonstrates the superiority of one transform over the others. DFT-LMS first introduced by Narayan (1983) is the simplest algorithm of this family, both because of the exponential nature of the DFT and because scientists have developed a strong intuition for the Fourier transform. It is our experience though that in most practical situations DCT-LMS performs much better than DFT-LMS. In addition, it has the advantage over DFT-LMS to be real valued.

In this paper, we first compare the general philosophies of DFT/DCT-LMS and recursive least squares (RLS) algorithms. We then explain, through three different approaches, the mechanisms of DFT-LMS and DCT-LMS. We present new results on convergence speed, and conclude with a short analysis of computational cost.

*This research was sponsored by EPRI under contract RP8010-13, by NSF under grant NSF IRI 91-12531, and by ONR under contrct N00014-92-J-1787.

DFT/DCT-LMS VS. RLS

By iteratively calculating the inverse autocorrelation matrix of the input data and using it to compute the current filter weights, RLS implements an exact least squares solution (Franklin, 1990; Haykin, 1991). The major advantages of RLS over LMS are its relatively low sensitivity to input eigenvalue spread, its fast convergence, and the fact that, at least for stationary inputs, the quality of its steady-state solution keeps on improving over time. On the other hand, RLS suffers from poor tracking capabilities in nonstationary environments (Bershad, 1989), from high computational cost, and from lack of robustness under certain input conditions.

The computational cost and robustness issues have been addressed by researchers in developing other exact least squares algorithms, the most famous of them being the recursive lattice filter algorithms. Lattice filters typically require less computations per iteration than RLS, but even their most robust forms can present stability problems (North, 1993). In addition, they are long and complicated to implement.

LMS is intrinsically slow because it does not decorrelate its inputs prior to adaptive filtering, but preprocessing the inputs with an estimate of the inverse input autocorrelation matrix in the fashion of RLS leads to the problems cited above. The solution we propose in the next section consists of preprocessing the inputs to the LMS filter with a fixed transformation that *does not* depend on the actual input data. The decorrelation will only be approximative, but the computational cost will remain very low, and the robustness and tracking ability of LMS will not be affected.

DFT-LMS AND DCT-LMS

The DFT-LMS and DCT-LMS algorithms are composed of three simple stages (see Fig. 1). First, the tap-delayed inputs are preprocessed by a discrete Fourier or cosine transform. The transformed signals are then normalized by the square root of their power. The resulting equal power signals are inputted to an adaptive linear combiner whose weights are adjusted using the LMS algorithm. With these two algorithms, the orthogonalizing step is data independent; only the power normalization step is data dependent (i.e. the power levels used to normalize the signals are estimated from the actual data).

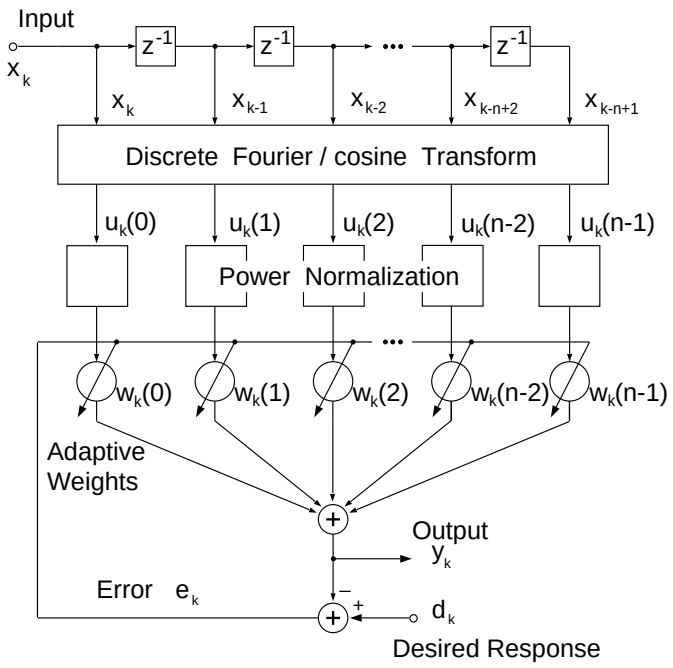

Figure 1: DFT-LMS and DCT-LMS block diagram.

A filtering approach

The n-point discrete Fourier/cosine transform can be seen as a $n \times n$ linear transformation from inputs $\mathbf{x}_k = (x_k, x_{k-1}, ..., x_{k-n+1})^t$ to outputs $\mathbf{u}_k = (u_k(0)), u_k(1), ..., u_k(n-1))^t$, where $u_k(i)$ is the i^{th} output of the DFT/DCT at time k (see Fig. 1). Each output $u_k(i)$ can be expressed as a weighted sum of the inputs x_{k-l}, for $l = 0..n-1$, that is as the convolution of x_k with some discrete impulse response h_i. In the case of the DFT,

$$h_i(l) = \sqrt{\frac{1}{n}}\, e^{j\frac{2\pi i l}{n}} \qquad \forall\, l = 0\,..\,n-1$$

The associated transfer function,

$$H_i(\omega) = \sqrt{\frac{1}{n}}\, \frac{1 - e^{-j\omega n}}{1 - e^{-j\omega} e^{j\frac{2\pi i}{n}}},$$

represents a bandpass filter of central frequency $2\pi i/n$. The DFT can thus be seen as a bank of bandpass filters whose central frequencies span the interval $[0\ 2\pi]$. Figure 2 shows the magnitude of the transfer functions $H_i(\omega)$ of an 8×8 DFT.

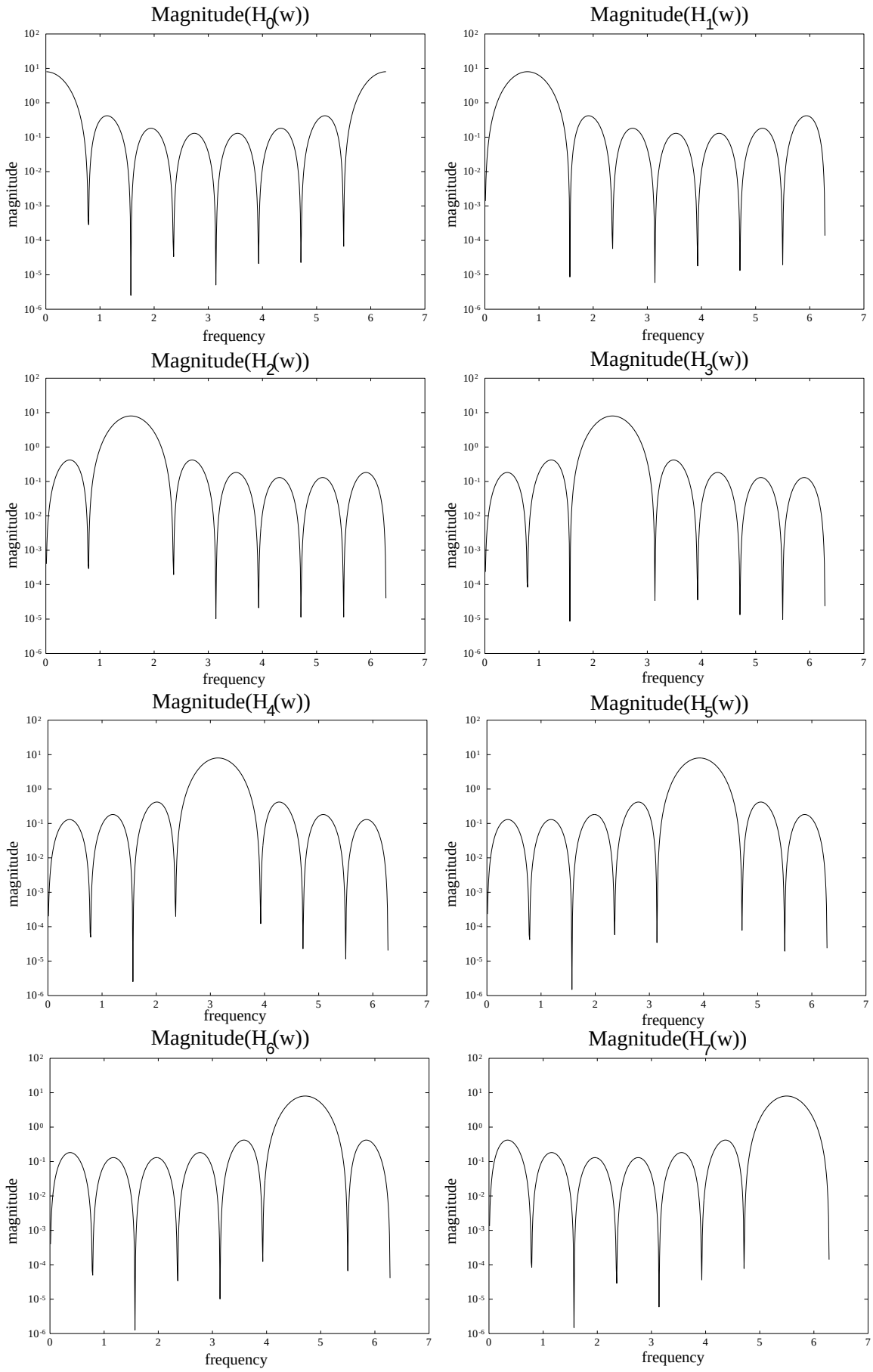

Figure 2: 8×8 DFT: magnitudes of the transfer functions of a bank of bandpass filters.

At each time k, the input signal $\mathbf{x}_k$ is decomposed into n signals lying in different frequency bins. If the bandpass filters were perfect, the outputs of the DFT would be perfectly uncorrelated, but due to the presence of side lobes (see Fig.2) there is some leakage from each frequency bin to the others, and thus some correlation between the output signals.

In the case of the DCT, the i^{th} impulse response $h_i(l)$ is given by

$$h_i(l) = \sqrt{\frac{2}{n}}\, K_i\, cos(\frac{i(l+1/2)\pi}{n}) \qquad \forall\, l = 0\,..\,n-1$$

where $K_i = 1/\sqrt{2}$ for $i = 0$ and 1 for $i = 1\,..\,n-1$. The corresponding transfer functions are given by

$$H_i(\omega) = \sqrt{\frac{2}{n}}\, K_i\, cos(\frac{i\pi}{2n})\, \frac{(1-e^{-j\,\omega})(1-(-1)^i e^{-j\,\omega n})}{1-2cos(\frac{i\pi}{n})e^{-j\,\omega}+e^{-2j\,\omega}}.$$

They still represent a bank of bandpass filters but with different central frequencies, different main lobe and side lobes, and different leakage properties.

A geometrical approach

The DFT-LMS and DCT-LMS algorithms can also be explained and illustrated geometrically. The DFT and DCT matrices defined by

$$\begin{aligned} \mathbf{F}(k,l) &= \sqrt{\frac{1}{n}}\, e^{j\,\frac{2\pi kl}{n}}, \\ \mathbf{C}(k,l) &= \sqrt{\frac{2}{n}}\, K_k\, cos(\frac{k(l+1/2)\pi}{n}), \end{aligned}$$

for $k, l = 0\,..\,n-1$, are unitary matrices (i.e. their rows are orthogonal to one another and have euclidian norm one). Unitary transformations perform only rotations and symmetries, they do not modify the shape of the object they transform.

The mean square error of LMS is a quadratic function of the weights (Widrow, 1985). Writing the MSE as a function of the weights and fixing it to some constant value, we get an implicit quadratic function of the weights that represents a hyperellipsoid in the n-dimensional weight space. A unitary transformation of the inputs rotates the hyperellipsoid and brings it into approximate alignment with the coordinate axes. The slight imperfection in alignment is primarily due to leakage in the transform, DCT or DFT. The idea is illustrated for a simple 2-weight case in Fig. 3. Figure 3(a) shows the original MSE ellipse, Fig. 3(b) shows it after transformation by a 2×2 DCT matrix. The shape of the ellipse is unchanged and so are the eigenvalues of the autocorrelation matrix.

The power normalization stage (cf. Fig. 1) can be viewed geometrically as a transformation that, while preserving the elliptical nature of the MSE, forces it to cross all the coordinate axes at the same distance from the center. This operation is not unitary and it does modify the eigenvalue spread. It almost always improves it. The better the alignment of the hyperellipsoid with the coordinate axes, the more efficient the power normalization will be (a hyperellipsoid perfectly aligned being transformed in a hypersphere). Figure 3 shows the result of power normalization for our example. The new ellipse is more round-shaped and has lower eigenvalue spread. This is very typical.

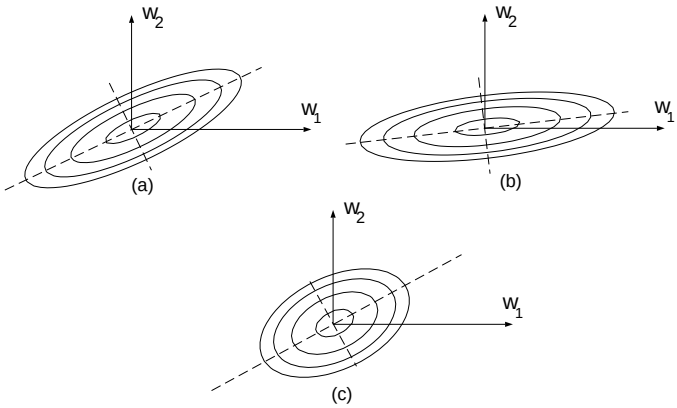

Figure 3: MSE hyperellipsoid (a) before transformation, (b) after DCT, (C) after power normalization.

An analytical approach

In order to find precise information on how well a given transform decorrelates certain classes of input signals, one must set the problem in a more mathematical framework. Transforming a signal $\mathbf{X}$ by a matrix $\mathbf{T}$ (the DFT or the DCT matrix), transforms its autocorrelation matrix $\mathbf{R} = E(\mathbf{X}\mathbf{X}^t)$ into $\mathbf{B} = E(\mathbf{T}\mathbf{X}\mathbf{X}^t\mathbf{T}^t) = \mathbf{T}\mathbf{R}\mathbf{T}^t$. The power of $\mathbf{T}\mathbf{X}$ can be found on the main diagonal of $\mathbf{B}$. Power normalizing $\mathbf{T}\mathbf{X}$ transforms its elements $\mathbf{T}\mathbf{X}_i$ into $\mathbf{T}\mathbf{X}_i/\sqrt{\text{Power of } (\mathbf{T}\mathbf{X}_i)}$, having the autocorrelation matrix

$$\mathbf{S} = (\text{diag}\mathbf{B})^{-1/2}\, \mathbf{B}\, (\text{diag}\mathbf{B})^{-1/2}.$$

If **T** decorrelated **X** exactly, **B** would be diagonal and **S** would be the identity matrix **I** and would have all its eigenvalues equal to one; but since the DFT and the DCT are not perfect decorrelators, this does not work out exactly. Some theory has been developed in the past about the decorrelating ability of the DFT and the DCT (see for example Grenander, 1984; Gray, 1977; Rao, 1990) but the results presented in the literature are in general too weak to allow us to infer anything about the magnitude of the individual eigenvalues of **S**, which is our main interest. For example, it has been proven that the autocorrelation matrix **B** obtained after the DFT or the DCT "asymptotically converges" to a diagonal matrix: "asymptotically" meaning as n, the size of **B**, tends to infinity, and "converges" being understood in a weak norm sense[1]. From this result, we can deduce that **S** will asymptotically converge to identity as n tends to infinity. However, we can not conclude anything about the possible convergence of the individual eigenvalues of **S** to one, everything depends on *how* and *how fast* **S** converges to **I**. To obtain stronger results, further assumptions are necessary, for example regarding the class of input signals to be considered.

Eigenvalues and eigenvalue spread for Markov-1 inputs

First order Markov signals are a very general, practical, and yet simple class of signals. They result from white noise passing through a single pole lowpass filter. Such a filter has an impulse response that decreases geometrically with a rate ρ given by the filter pole. A Markov-1 input signal $\mathbf{X}_k = (x_k, x_{k-1}, ..., x_{k-n+1})^t$ of parameter $\rho \in [0\ 1]$ has an autocorrelation matrix **R** equal to

$$\mathbf{R} = \begin{pmatrix} 1 & \rho & \rho^2 & \dots & \rho^{n-1} \\ \rho & 1 & \rho & \dots & \rho^{n-2} \\ \rho^2 & \rho & 1 & & \\ \vdots & \vdots & & \ddots & \vdots \\ \rho^{n-1} & \rho^{n-2} & & \dots & 1 \end{pmatrix}$$

For n large (theoretically for n tending to infinity), the minimum and maximum eigenvalues of an autocorrelation matrix **R** are given by the minimum and maximum of the power spectrum of the signal that generated this autocorrelation (Grenander, 1984; Gray, 1977). This result is a direct consequence of the fact that **R** is Toeplitz. It can easily be checked that in our case the power spectrum of x_k is given by

$$P(\omega) = \sum_{l=-\infty}^{+\infty} \rho^l e^{-j\,\omega l} = \frac{1}{1 - 2\rho cos(\omega) + \rho^2}.$$

Its maximum and minimum are respectively $1/(1-\rho)^2$ and $1/(1+\rho)^2$. The eigenvalue spread of **R** thus tends to

$$\text{Eigenvalue spread before transformation} = (\frac{1+\rho}{1-\rho})^2.$$

This eigenvalue spread can be extremely large for highly correlated signals (ρ close to 1).

The autocorrelation **S** of the signals obtained after transformation by the DFT or the DCT and after power normalization is not Toeplitz anymore, and the previous theory can not be applied. The analysis is further complicated by the fact that only asymptotically do the eigenvalues stabilize to fixed magnitudes independent of n, and that power normalization is a nonlinear operation. Successive matrix manipulations and passages to the limit allowed us to prove the following asymptotic results (see Beaufays, 1993, 1994) for more details):

$$\begin{aligned} \text{Eigenvalue spread after DFT} &= \frac{1+\rho}{1-\rho}, \\ \text{Eigenvalue spread after DCT} &= 1+\rho. \end{aligned}$$

Note that with the DCT, the asymptotic eigenvalue spread is never higher than 2!

As a numerical example, let the correlation ρ be equal to 0.95. The eigenvalue spread before transformation is 1521, after the DFT 39, after the DCT 1.95. In this case, using the DCT-LMS instead of LMS would speed up the filter weight convergence by a factor roughly equal to 750.

These results confirm, for a simple but very practical class of signals, the high quality of the DCT as a signal decorrelator.

Computational cost of DFT-LMS and DCT-LMS

In addition to their fast convergence and robustness, DFT-LMS and DCT-LMS have the advantage of a very low computational cost. The inputs $x_k, x_{k-1}, ..., x_{k-n+1}$ being delayed samples of the same signal, the DFT-DCT can be computed in $\mathcal{O}(n)$ operations. For the DFT,

$$u_k(i) = \sum_{l=0}^{n-1} e^{j\,\frac{2\pi i l}{n}} x_{k-l}$$

[1] Two matrices converge to one another in a weak norm sense when their weak norms converge to one another. The weak norm of a matrix is defined as the square root of the arithmetic average of its eigenvalues.

$$= x_k + \sum_{l=1}^{n} e^{j\frac{2\pi il}{n}} x_{k-l} - e^{j\frac{2\pi in}{n}} x_{k-n}$$
$$= e^{j\frac{2\pi i}{n}} u_{k-1}(i) + x_k - x_{k-n}.$$

The $u_k(i)$'s can thus be found by an $O(n)$ recursion from the $u_{k-1}(i)$'s. This type of DFT is sometimes called *sliding DFT*. A similar $\mathcal{O}(n)$ recursion can be derived with more algebra for the DCT:

$$u_k^{DCT}(i) = u_{k-1}^{DCT}(i)cos(\frac{\pi i}{n}) - u_k^{DST}(i)sin(\frac{\pi i}{n}) + \sqrt{\frac{2}{n}}cos(\frac{\pi i}{2n})(x_k - (-1)^i x_{k-n}),$$
$$u_k^{DST}(i) = u_{k-1}^{DST}(i)cos(\frac{\pi i}{n}) + u_k^{DCT}(i)sin(\frac{\pi i}{n}) + \sqrt{\frac{2}{n}}sin(\frac{\pi i}{2n})(x_k - (-1)^i x_{k-n}).$$

$u_k^{DCT}(i)$ is the i^{th} output of the DCT, $u_k^{DST}(i)$ is the i^{th} output of a DST (discrete sine transform) defined exactly like the DCT but replacing "cos" by "sin" (interlacing two recursions is necessary and comes basically from the fact that $cos(a+b) = cos(a)cos(b) - sin(a)sin(b)$).

The power levels of the $u_k(i)$'s can also be computed by a simple $\mathcal{O}(n)$ recursion:

$$P_k(i) = \beta P_{k-1}(i) + u_k^2(i),$$

where $P_k(i) = (\text{power of } u_k(i))/(1-\beta)$, $P_{-1}(i)$ is initialized to zero, and $\beta \in [0\ 1]$ is a forgetting factor.

Finally, the last step, the LMS adaptation of the variable weights, is $\mathcal{O}(n)$. The overall algorithm is thus $\mathcal{O}(n)$.

CONCLUSION

For the most part, the DCT-LMS algorithm is superior to the DFT-LMS algorithm. Both are robust algorithms, containing three robust steps: transformation, power normalization (like automatic gain control in a radio or TV), and LMS adaptive filtering. These algorithms are easy to program and to understand. They use a minimum of computation, only slightly more than LMS alone. They work almost as well as RLS but don't have robustness problems. The lattice forms of RLS are more robust than RLS, but they are much more difficult to program and to understand. All in all, the DFT-LMS and DCT-LMS algorithms should find increased use in practical real-time applications.

REFERENCES

Beaufays, F. and Widrow, B. 1993. Transform domain adaptive filters: an analytical approach. Submitted to *IEEE Trans. on Signal Proc.*

Beaufays, F. 1994. Ph.D. Thesis, in preparation. Information Systems Lab., Stanford University, Stanford, CA.

Bershad, N. and Macchi, O. 1989. Comparison of RLS and LMS algorithms for tracking a chirped signal. *Proc. ICASSP*. Glasgow, Scotland: 896-899.

Franklin, G. F. et al. 1990. *Digital Control of Dynamic Systems*. Second edition. Addison-Wesley, Reading, MA.

Gray, R. M. 1977. *Toeplitz and Circulant Matrices: II.* Tech. report. 6504-1. Information Systems Lab., Stanford University.

Grenander, U. and Szegö, G. 1984. *Toeplitz forms and their applications*. Second edition. Chelsea Publishing Company, New York.

Haykin, S. 1991. *Adaptive Filter Theory*. Second edition. Prentice Hall, Englewood Cliffs, NJ.

North, R. C. et al. 1993. A floating-point arithmetic error analysis of direct and indirect coefficient updating techniques for adaptive lattice filters. *IEEE Trans. on Signal Proc.* 41, no.5(May):1809-1823.

Rao, K. R. and Yip, P. 1990. *Discrete cosine transform*. Academic Press, Inc, San Diego, CA.

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

Françoise Beaufays received the bachelor degree in mechanical and electrical engineering in 1988 from Université Libre de Bruxelles, Brussels, Belgium, and the MSEE degree in 1989 from Stanford University. She is currently a PhD student and research assistant at Stanford University, working under the supervision of Prof. B. Widrow. Her interests include linear adaptive filtering, signal processing, and nonlinear neural networks.