A robust, low-computational, near optimal convergence speed, multi-channel Filtered-X LMS algorithm

Martin Bouchard, Mechanical Engineering
Bruno Paillard, Electrical Engineering
University of Sherbrooke
2500 Boul. Université, Sherbrooke
J1K 2R1

1.0 Introduction

A widely used algorithm for real-time implementations of multi-channel active control systems is the multi-channel Filtered-X LMS, and a normalisation of the algorithm has been published [1] : the NLMS. The drawback of these algorithms is their slow broadband convergence speed for strongly correlated reference signals. In this paper, a normalised fast convergence algorithm that has nearly the same low computational load than the Filtered-X LMS will be presented.

2.0 A new fast convergence algorithm

To describe the algorithms, here are some definitions for the different elements of a feedforward FIR adaptive control system :

Nx : number of reference sensors
Ny : number of output actuators
Ne : number of error sensors
W(i,j,iter): adaptive filter between i^{th} input sensor and j^{th} output actuator, after « iter » optimisation iterations
ΔW(i,j,iter): modification to the adaptive filter between i^{th} input sensor and j^{th} output actuator, after « iter » optimisation iterations
H(j,m) : filter modeling the path between the j^{th} output actuator and the m^{th} error sensor
Lw : length of the adaptive filters W(i,j,iter)
Lh : length of the filters H(j,m)
X(i,k) : vector of the Lh last samples at time k from the i^{th} input sensor
e(m,k) : residual error associated to the m^{th} error sensor, at time k
d(m,k) : sample from the m^{th} error sensor, at time k
V(i,j,m,k): vector of the Lw last samples of the filtered reference signal, calculated by filtering X(i,k) with H(j,m)

$X(i,k)^T =$ [x(i,k-Lh+1) ... x(i,k)]
$H(j,m)^T =$ [h(j,m,Lh) ... h(j,m,1)]
$W(i,j,iter)^T =$ [w(i,j,iter,Lw) ... w(i,j,iter,1)]
$\Delta W(i,j,iter)^T =$ [Δw(i,j,iter,Lw) ... Δw(i,j,iter,1)]
$V(i,j,m,k)^T =$ [v(i,j,m,k-Lw+1) ... v(i,j,m,k)].

In practice, a Filtered-X LMS algorithm is often divided into two parts : the control part (real-time part) and the optimisation part. As shown in Figure 1, instead of optimizing directly the adaptive filters W(i,j,iter) with the signals from the error sensors, variation filters ΔW(i,j,iter) are optimized with the residual errors, i.e. the signals that have not been cancelled by both the W(i,j,iter) and ΔW(i,j,iter) filters. The real-time control task of the W(i,j,iter) filters is executed at every sample, but the optimisation of the ΔW(i,j,iter) filters is calculated during idle processor time (whenever the processor is not executing a real-time control task). After some number of optimisation iterations, the values of ΔW(i,j,iter) are added to W(i,j,iter) and then cleared to start a new optimisation cycle.

The basic equations of a Filtered-X LMS are then («*» denotes a scalar product):

$$v(i,j,m,k) = X(i,k)^T * H(j,m) \qquad \text{(eq. 1)}$$

$$e(m,k) = \sum_i \sum_j V(i,j,m,k)^T * \Delta W(i,j,iter) + d(m,k) \qquad \text{(eq. 2)}$$

$$\Delta W(i,j,iter+1) = \Delta W(i,j,iter) - u \sum_m V(i,j,m,k) e(m,k) \qquad \text{(eq. 3).}$$

Now to describe the NLMS, an interlaced notation that will make the global correlation matrix of the filtered reference signals appear « block Toeplitz » will be introduced. We shall make use of that block Toeplitz property later.

If the following matrices are defined :

V(k)=

$$\begin{bmatrix} \begin{bmatrix} v(1,1,1,k-Lw+1) \\ \vdots \\ v(Nx,Ny,1,k-Lw+1) \end{bmatrix} & & \begin{bmatrix} v(1,1,Ne,k-Lw+1) \\ \vdots \\ v(Nx,Ny,Ne,k-Lw+1) \end{bmatrix} \\ \vdots & \cdots & \vdots \\ \begin{bmatrix} v(1,1,1,k) \\ \vdots \\ v(Nx,Ny,1,k) \end{bmatrix} & & \begin{bmatrix} v(1,1,Ne,k) \\ \vdots \\ v(Nx,Ny,Ne,k) \end{bmatrix} \end{bmatrix}$$

$$\Delta W(iter) = \begin{bmatrix} \begin{bmatrix} w(1,1,iter,Lw-1) \\ \vdots \\ w(Nx,Ny,iter,Lw-1) \end{bmatrix} \\ \vdots \\ \begin{bmatrix} w(1,1,iter,1) \\ \vdots \\ w(Nx,Ny,iter,1) \end{bmatrix} \end{bmatrix}$$

$$E(k) = \begin{bmatrix} e(1,k) \\ \vdots \\ e(Ne,k) \end{bmatrix} \qquad D(k) = \begin{bmatrix} d(1,k) \\ \vdots \\ d(Ne,k) \end{bmatrix}$$

then equations 2 and 3 for the Filtered-X LMS algorithm can now be re-written in a compact way:

$$E(k) = V(k)^T \Delta W(iter) + D(k) \qquad \text{(eq. 4)}$$

$$\Delta W(iter+1) = \Delta W(iter) - uV(k)E(k) \qquad \text{(eq. 5).}$$

The NLMS algorithm recurrence equation can easily be written with that same notation (equations 1 and 4 are still valid for the NLMS):

$$\Delta W(iter+1) = \Delta W(iter) - uV(k)(V(k)^T V(k))^{-1} E(k) \qquad \text{(eq. 6).}$$

Introducing the inverse of the Ne×Ne matrix $V(k)^T V(k)$ in the recurrence equation 6 sets the range of u for which the algorithm is to converge (that range is independent of the filtered reference signals statistics) : $0 < u < 2$. At the opposite, the valid range of step sizes for equations 3 or 5 for the Filtered-X LMS depends on the filtered reference signals statistics and is the following (for $u \ll 1/(\text{any eigenvalue of } E[V(k)V(k)^T])$ [4]) :

$$0 < u < 2/\mathrm{Trace}(E[V(k)V(k)^T])$$

$E[V(k)V(k)^T]$ is the NxNyLw × NxNyLw global correlation matrix of all the filtered reference signals (not to be confused with the instantaneous $V(k)^T V(k)$ of dimension Ne × Ne in equation 6). For now on, the former will be called Rvv.This correlation matrix is block Toeplitz because of the interlaced notation chosen for V(k) and ΔW(iter).

The main drawback of the Filtered-X LMS is its low broadband convergence speed when the filtered reference signals are strongly correlated. To achieve optimal convergence speed, the Filtered-X Newton-LMS algorithm can be used [2]. The Filtered-X Newton-LMS algorithm recurrence equation is (again equations 1 and 4 are valid for the Filtered-X Newton-LMS) :

$$\Delta W(iter+1) = \Delta W(iter) - u\,Rvv^{-1}V(k)\,E(k) \qquad \text{(eq. 7).}$$

The matrix product with Rvv^{-1} ,requiring a lot of computational power, makes this algorithm unpractical for real-time systems with a large number of coefficients (Lw). Still, this algorithm achieves the optimal convergence speed and is the fastest of all stochastic gradient algorithms. The valid range of u for which the algorithm converges is (for u << 1): $0 < u < 2/NxNyLw$.

It is a well known fact in digital signal processing that the discrete cosine transform (DCT) can almost orthogonalize a strongly correlated signal. In [3], a fast convergence mono-channel Filtered-X LMS algorithm based on a transform domain optimisation (DCT transform) was presented. Orthogonalizing many filtered reference signals between each other is the same as diagonalizing the global correlation matrix Rvv. Let the new filtered reference matrix C(k) be defined as:

$$C(i,j,m,k)^T = [c(i,j,m,Lw) \ldots c(i,j,m,1)]$$
$$= \text{discrete cosine transform } (v(i,j,m,k-Lw+1) \ldots v(i,j,m,k))$$
$$= \text{discrete cosine transform } (V(i,j,m,k)^T) \qquad \text{(eq.8)}$$

$$\text{and } C(k) = \begin{bmatrix} \begin{bmatrix} c(1,1,1,Lw) \\ c(Nx,Ny,1,Lw) \end{bmatrix} & \cdots & \begin{bmatrix} c(1,1,Ne,Lw) \\ c(Nx,Ny,Ne,Lw) \end{bmatrix} \\ \begin{bmatrix} c(1,1,1,1) \\ c(Nx,Ny,1,1) \end{bmatrix} & \cdots & \begin{bmatrix} c(1,1,Ne,1) \\ c(Nx,Ny,Ne,1) \end{bmatrix} \end{bmatrix}$$

The discrete cosine transform on the different filtered reference signals can make Rcc ($=E[C(k)C(k)^T]$) approximately block diagonal, which means that the autocorrelation of the different filtered reference signals has been almost eliminated by the discrete cosine transform, and that the intercorrelation between the different filtered reference signals has also been almost eliminated except between components at the same frequency.

The matrix that has nul elements outside of the main block diagonal and that has the elements of Rcc in the main block diagonal is called RDcc (RDcc $\approx$ Rcc). The filtered reference matrix Cn(k) is then introduced:

$$Cn(k) = RDcc^{-1/2\,T}\,C(k) \qquad \text{(eq.9)}$$

and the new global correlation matrix will be:

$$\begin{aligned} Rcncn &= E[Cn(k)\,Cn(k)^T] \\ &= E[RDcc^{-1/2\,T}\,Rcc\,RDcc^{-1/2}] \\ &\approx E[RDcc^{-1/2\,T}\,RDcc\,RDcc^{-1/2}] \\ &= I \end{aligned} \qquad \text{(eq.10).}$$

So the filtered reference signals are now almost uncorrelated and the convergence behavior is almost optimal. Note that $RDcc^{-1/2\,T}$ in eq.9 requires only the product of the main block diagonal unlike Rvv^{-1} in eq.7 for the Newton-LMS (much less computations for a large Lw). The equations for the COS-NLMS algorithm are (equation 1 remains valid, $\Delta\Omega(iter)$ is the transform domain equivalent of the time domain $\Delta W(iter)$) :

$$E(k) = Cn(k)^T\,\Delta\Omega(iter) + D(k) \qquad \text{(eq.11)}$$

$$\Delta\Omega\,(iter+1) = \Delta\Omega(iter) - u Cn(k)\,(Cn(k)^T\,Cn(k))^{-1}\,E(k) \qquad \text{(eq.12)}$$

After an optimisation sequence is complete, an inverse discrete cosine transform is performed on each adaptive filter of $\Delta\Omega(iter)$ to calculate all the adaptive filters of $\Delta W(iter)$ (so that the real-time control computations can still be performed in the time domain):

$$\Delta W(i,j,iter)^T = [\Delta w(i,j,iter,lw) \ldots \Delta w(i,j,iter,1)]$$
$$= \text{inverse discrete cosine transform}(\Delta\omega(i,j,iter,lw)\ldots\Delta\omega(i,j,iter,1))$$
$$= \text{inverse discrete cosine transform } (\Delta\Omega(i,j,iter)^T) \qquad \text{(eq.13).}$$

Figure 2 shows an implementation of the COS-NLMS algorithm. Just like the NLMS, the range of u for which the algorithm converges is : $0 < u < 2$.

Some active control experiments in a duct have been performed to compare the NLMS and the COS-NLMS algorithms. Figure 3 shows some typical convergence curves for the two algorithms. As it is shown in that figure, the COS-NLMS has a faster convergence behavior than the NLMS.

3.0 Conclusion

A robust low computational approximation of the Newton-LMS algorithm was developped : the COS-NLMS algorithm. Real experiments of active control of noise in a duct have shown the effective gain of convergence speed provided by the COS-NLMS over the NLMS (or over the standard Filtered-X LMS).

Acknowledgments

Financial support for this work was provided by the « Institut de Recherche en Santé et Sécurité au Travail » (IRSST).

References

[1] DOUGLAS, S., OLKIN, J. (avril 1993) *Multiple-Input, Multiple Output, Multiple-Error Adaptive Feedforward Control Using the Filtered-X Normalised LMS Algorithm*, Proceedings of the second conference on recent advances in active control of sound and vibration, Blacksburgh(VA), 743-754.

[2] WIDROW, B., STEARNS, S.D. (1985) *Adaptive Signal Processing*, Englewood Cliffs, Prentice Hall, 99-116, 142-147.

[3] PAILLARD, B., BERRY, A., LE DINH, C.T., NICOLAS, J. (1995) *Accelerating The Convergence Of The Filtered-X LMS Algorithm Through Transform-Domain Optimization*, To be published in Mechanical Systems and Signal Processing.

[4] HAYKIN, S. (1991) *Adaptive Filter Theory*, Englewood Cliffs, Second edition, Prentice Hall

Figures

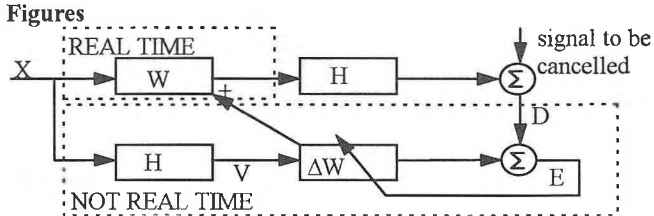

Figure 1: an implementation of the Filtered-X LMS algorithm.

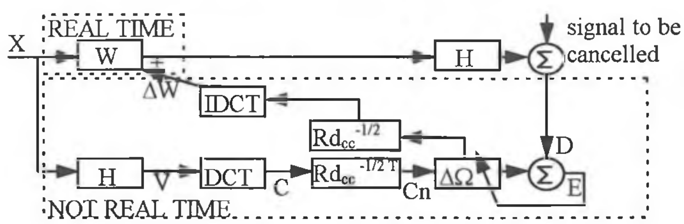

Figure 2: an implementation of the COS-NLMS algorithm.

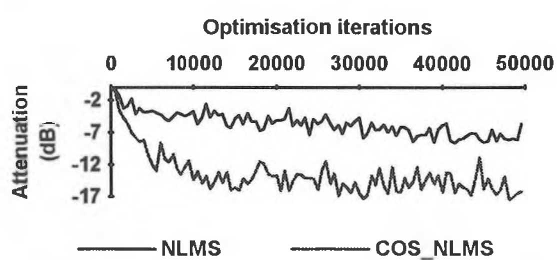

Figure 3 : results of active noise control in a duct