isvr Institute of Sound and Vibration Research

Algorithm of Broadband Active Noise Controllers Using Frequency-Domain Methods

T. Kosaka, C.C. Boucher and S.J. Elliott

ISVR Technical Memorandum 816

March 1997

University of Southampton

SCIENTIFIC PUBLICATIONS BY THE ISVR

Technical Reports are published to promote timely dissemination of research results by ISVR personnel. This medium permits more detailed presentation than is usually acceptable for scientific journals. Responsibility for both the content and any opinions expressed rests entirely with the author(s).

Technical Memoranda are produced to enable the early or preliminary release of information by ISVR personnel where such release is deemed to the appropriate. Information contained in these memoranda may be incomplete, or form part of a continuing programme; this should be borne in mind when using or quoting from these documents.

Contract Reports are produced to record the results of scientific work carried out for sponsors, under contract. The ISVR treats these reports as confidential to sponsors and does not make them available for general circulation. Individual sponsors may, however, authorize subsequent release of the material.

UNIVERSITY OF SOUTHAMPTON
INSTITUTE OF SOUND AND VIBRATION RESEARCH
SIGNAL PROCESSING & CONTROL GROUP

Algorithm of broadband active noise controllers using frequency-domain methods

by

T.Kosaka, C.C.Boucher, and S.J.Elliott

ISVR Technical Memorandum No. 816

March 1997

1. Introduction

The Filtered-X LMS algorithm is widely used in active noise control.[1] In this report, frequency domain methods for the adaptation of active control system are considered. A Frequency Domain implementation of the LMS algorithm using the FFT has two advantages. One is reducing computing complexity for long control filter lengths. The other is the rapid convergence ability when the normalized LMS is employed. The Frequency Domain LMS involves the estimation of a cross spectral density. There is a problem in choosing the correct window function, that is applied before FFT calculation, to obtain an unbiased estimation and to eliminate wrap-around effects.

This report shows the effect of the window functions in estimating the impulse response of a primary path. The effect appears both as a modifier of the estimation and as eliminator of wrap-around effects. Both are expressed as equations. Then some examples of window functions are shown in figures. This makes clear what window function is suitable for estimation of impulse response using cross spectral density. It is also suggested what window function is suitable for estimation of cross spectral density.

The application of this method to active control system is then described. Frequency Domain LMS Algorithms suggested previously for active control have implemented the control filtering and the adaptation in the frequency domain [2, 3], which causes a one block processing delay in the controller.

In the Frequency Domain Filtered-X LMS Algorithm used in this paper, the updating of the control filter is performed in the frequency domain as a background task, while control filtering is performed in the time domain, thus minimizing the delay in the controller.

Time and Frequency Domain Filtered-X LMS algorithms for active noise reduction of a single channel system are simulated, which illustrate the importance of using the correct window function and of zeropadding the data vector. A multi-channel system, which uses one reference two secondary sources and two error sensors, is also described.

2. Estimation of Impulse Response with Cross Spectral Density

Although the frequency domain LMS uses the estimate of a cross spectral density, it is convenient for comparing window functions to estimate the primary path impulse response with various window functions. We compared algorithms using various window functions on estimating such an impulse response by block averaging method as shown in Fig. 2.1.

In Fig. 2.1 the estimation of the primary path impulse response is obtained by:

$$\hat{w}(i) = IFFT(W(k)) \qquad (i = 0,1,..,N-1,\ k = 0,1,...,N-1) \tag{2.1}$$

where:

$$W(k) = \frac{S_{xy}(k)}{S_{xx}(k)} \tag{2.2}$$

$$S_{xy}(k) = \frac{1}{L}\sum_{L} (Conj(X(k))\ Y(k)) \tag{2.3}$$

$$S_{xx}(k) = \frac{1}{L}\sum_{L} (Conj(X(k))\ X(k)) \tag{2.4}$$

X(k) and Y(k) are the spectra in the frequency domain of the input signal and the output signal respectively. N is the number of samples in one block, and L is the number of sample blocks. The correct estimation is expected in the first half of the block length, i.e., $0 \le i < N/2$.

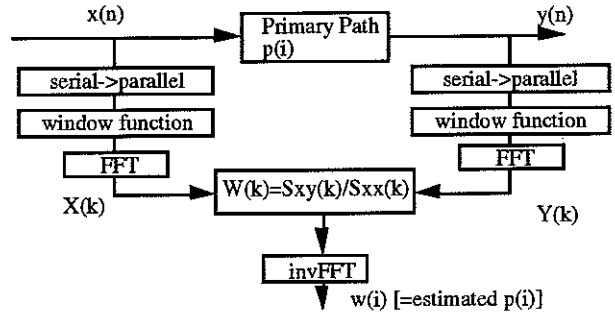


Fig. 2.1 Estimation of Impulse Response with Cross Spectral Density

3. Effect of the Window Function on the Estimated impulse response

Suppose we estimate the primary path impulse response using a window function shown in Fig. 3.1. The input signal to the primary path is white noise. We assume the primary path is a simple delay, i.e.,:

$$p(i) = \delta(i - i_1) \qquad (i = 0,1,...,N-1,\ \ 0 \le i_1 < N) \tag{3.1}$$

The impulse response to be estimated should be causal.

The window functions are:

$$f_x(i),\ f_y(i) \qquad (i = 0,1,...,N-1) \tag{3.2}$$

The time domain signals passed through windows functions are:

$$x_1(i) = f_x(i)\, x(i),\ \ y_1(i) = f_y(i)\, y(i) \qquad (i = 0,1,...,N-1) \tag{3.3}$$

In circular convolution the cross-correlation is:

$$R'_{xy}(j) = \frac{1}{N}\sum_{i=0}^{N-j-1} x_1(i)\, y_1(i+j) + \frac{1}{N}\sum_{i=N-j}^{N-1} x_1(i)\, y_1(i+j-N) \tag{3.4}$$

The first term involves valid data, while the second term consists of only invalid data. If block averaging is processed, it changes into:

$$E\{R'_{xy}(j)\} = R_{xy}(j)\,\frac{1}{N}\sum_{i=0}^{N-j-1} f_x(i)\, f_y(i+j) \tag{3.5}$$

The auto-correlation in circular convolution at j=0 is:

$$R'_{xx}(0) = \frac{1}{N}\sum_{i=0}^{N-1} x_1(i)\, x_1(i) \tag{3.6}$$

If block averaging is processed, it changes into:

$$E\{R'_{xx}(0)\} = R_{xx}(0)\,\frac{1}{N}\sum_{i=0}^{N-1} f_x^2(i) \tag{3.7}$$

As the primary path is a simple delay, the estimation of the primary path is easily obtained. The estimation of i1th element will be:

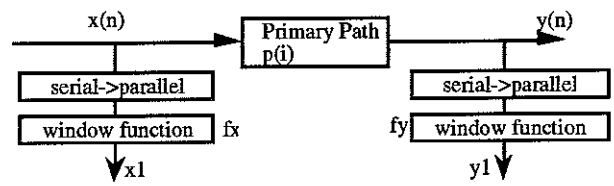


Fig. 3.1 Window Functions and Signals

$$\hat{w}(i_1) = \frac{E\{R_{xy}(i_1)\}}{E\{R_{xx}(0)\}} = w(i_1)\, m(i_1) \tag{3.8}$$

where

$$w(i_1) = \frac{R_{xy}(i_1)}{R_{xx}(0)} \tag{3.9}$$

is true estimation, and:

$$m(i_1) = \frac{q(i_1)}{q_1} \tag{3.10}$$

is scale function, that diminishes the amplitude. Where q(i1) and q1 are:

$$q(i_1) = \frac{1}{N} \sum_{i=0}^{N-i_1-1} f_x(i)\, f_y(i+i_1) \tag{3.11}$$

$$q_1 = \frac{1}{N} \sum_{i=0}^{N-1} f_x^2(i) \tag{3.12}$$

The value of m(i1) shows how window function scales down an estimated simple delay. It means that the simple delay defined in equation (3.1) is estimated as:

$$\hat{w}(i) = m(i_1)\, \delta(i - i_1) \quad (i = 0,1,...,N-1) \tag{3.13}$$

This result is genaralized to any primary path impulse responses.

$$\hat{w}(i) = m(i)\, p(i) \quad (i = 0,1,...,N-1) \tag{3.14}$$

This means m(i) distorts the impulse response. The m(i) can be calculated about various window functions. Now we can examine what window function is suitable for impulse response estimation.

Example 1 Rectangular Window

Assume that fx and fy are rectangular windows, i.e., :

$$f_x(i) = f_y(i) = \begin{cases} 1 & 0 \le i < N \\ 0 & otherwise \end{cases} \tag{3.15}$$

About these functions q1 and q(i) are:

$$q_1 = \frac{1}{2} \tag{3.16}$$

$$q(i) = \frac{N-i}{N} \tag{3.17}$$

Then the scale function m(i) is obtained as:

$$m(i) = \frac{N-i}{N} \tag{3.18}$$

Example 2 Zero padding in output signal

Assume that fx is rectangular window and zeros are padded in first half of output signal y, i.e.,:

$$f_x(i) = \begin{cases} 1 & 0 \le i < N \\ 0 & otherwise \end{cases}, \quad f_y(i) = \begin{cases} 0 & 0 \le i < N/2 \\ 1 & otherwise \end{cases} \tag{3.19}$$

About these functions q1 and q(i) are:

$$q_1 = 1 \tag{3.20}$$

$$q(i) = \begin{cases} \frac{1}{2} & 0 \le i < N/2 \\ \frac{N-i}{N} & otherwise \end{cases} \tag{3.21}$$

Then the scale function m(i) is obtained as:

$$m(i) = \begin{cases} \frac{1}{2} & 0 \le i < N/2 \\ \frac{N-i}{N} & otherwise \end{cases} \tag{3.22}$$

Example 3 Zero padding in input signal

Assume that fy is rectangular window and zeroes are padded in second half of input signal x, i.e.,:

$$f_x(i) = \begin{cases} 1 & 0 \le i < N/2 \\ 0 & otherwise \end{cases}, \quad f_y(i) = \begin{cases} 1 & 0 \le i < N \\ 0 & otherwise \end{cases} \tag{3.23}$$

About these functions q1 and q(i) are:

$$q_1 = \frac{1}{2} \tag{3.24}$$

$$q(i) = \begin{cases} \frac{1}{2} & 0 \le i < N/2 \\ \frac{N-i}{N} & otherwise \end{cases} \tag{3.25}$$

Then the scale function m(i) is obtained as:

$$m(i) = \begin{cases} 1 & 0 \le i < N/2 \\ \frac{2(N-i)}{N} & otherwise \end{cases} \tag{3.26}$$

Example 4 Hanning window

Assume that both fx and fy are Hanning windows, i.e.,:

$$f_x(i) = f_y(i) = \frac{1 - \cos\left(2\pi\frac{i}{N}\right)}{2} \quad 0 \le i < N \tag{3.27}$$

About these functions q1 and q(i) are:

$$q_1 = \frac{1}{N}\sum_{i=0}^{N-1}\left(\frac{1}{2}(1 - \cos 2\pi\frac{i}{N})\right)^2 = \frac{3}{8} \tag{3.28}$$

$$q(j) = \frac{1}{N}\sum_{i=0}^{N-j-1}\frac{1}{4}(1 - \cos 2\pi\frac{i}{N})(1 - \cos 2\pi\frac{i+j}{N}) \tag{3.29}$$

Then:

$$q(i) = \frac{1}{4}(1 - \frac{i}{N})(1 + \frac{1}{2}\cos 2\pi\frac{i}{N}) + \frac{1}{8N}C\sin 2\pi\frac{i}{N} \tag{3.30}$$

$$C = \frac{2\sin^2 2\pi\frac{1}{N} - \cos 2\pi\frac{1}{N} + \cos^2 2\pi\frac{1}{N}}{(1 - \cos 2\pi\frac{1}{N})\sin 2\pi\frac{1}{N}} \tag{3.31}$$

Then the scale function m(i) is obtained as:

$$m(i) = \frac{2}{3}(1 - \frac{i}{N})(1 + \frac{1}{2}\cos 2\pi\frac{i}{N}) + \frac{1}{3N}C\sin 2\pi\frac{i}{N} \tag{3.32}$$

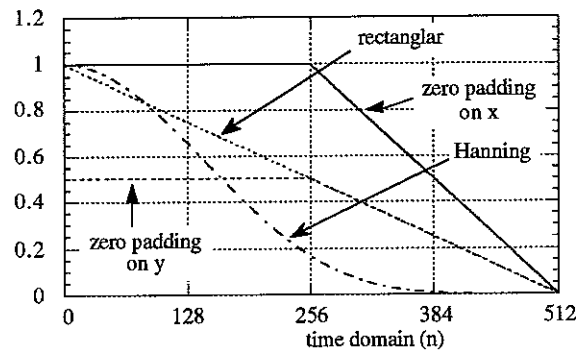
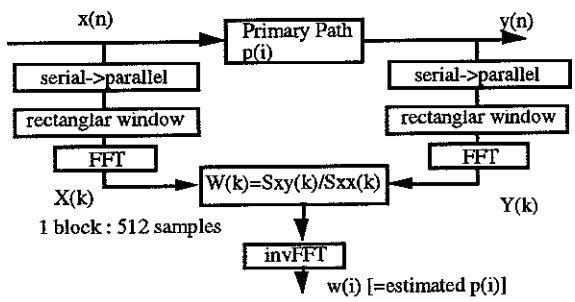


Fig. 3.2 the Scale Functions in Estimating Impulse Response depending on the Window Function

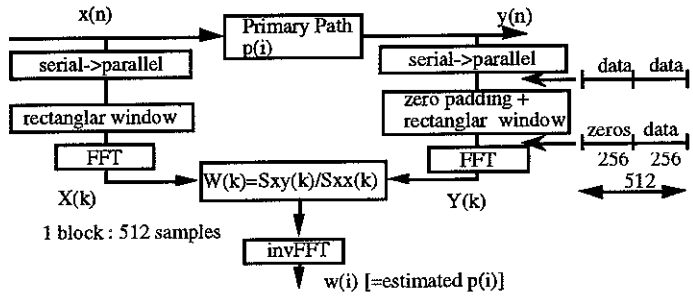
The results of the examples are shown in Fig. 3.2. It is found that both rectangular window and Hanning window distort estimation of impulse responses. Zero padding on x successfully estimates impulse response in the first half of it. As zero padding on y provides constant half scaling in the first half, it can be also used for this purpose.

4. Comparison of Window Functions in Simulation

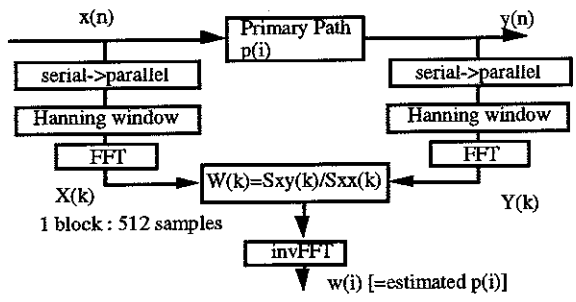
To ascertain above, we compared algorithms using the rectangular window, Hanning window and zero padding + rectangular window on estimating an impulse response by block averaging method as shown in Fig. 4.1. In the estimation the number of sample points of one block was N=512, and the number of averaging blocks was L=1024. The input signal was white noise. The impulse response of the primary path used here is shown in Fig. 4.2.



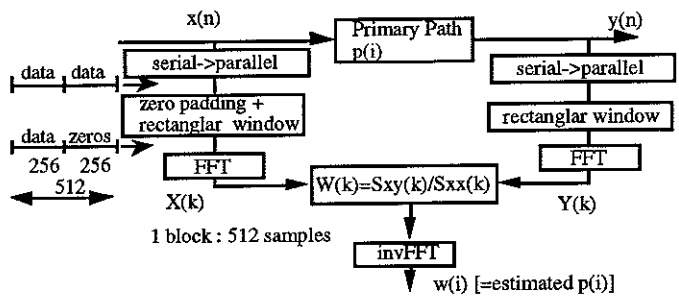
(a) Rectangular window



(b) Rectangular window + Zero padding



(c) Hanning window



(d) Zero padding + Rectangular window

Fig. 4.1 Comparison of Window Functions

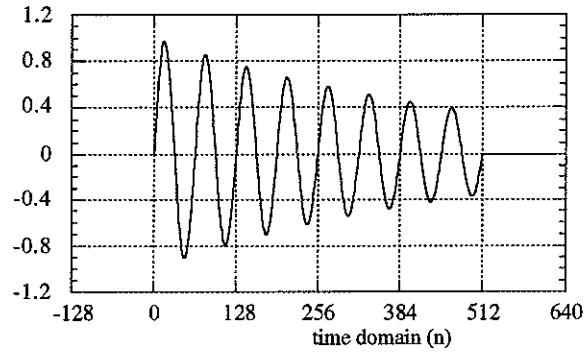
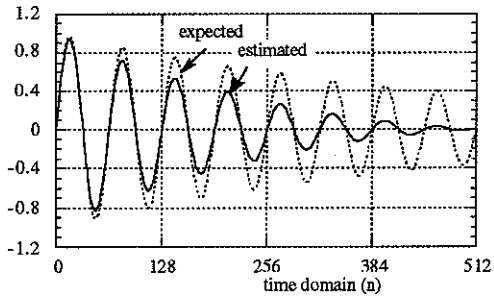


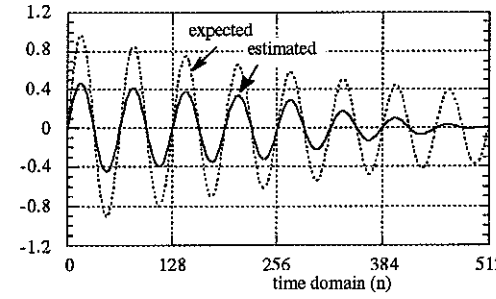
Fig. 4.2 the Impulse Response of the Primary Path

The results of simulation are shown in Fig. 4.3, in which the scale function calculated in previous section can be seen.

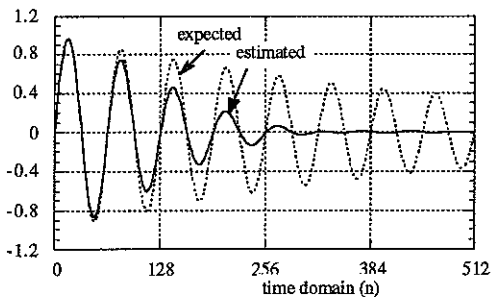
The rectangular window is not suitable for this estimation as shown in Fig. 4.3(a). Rectangular window + Zero padding in y provides the estimation whose amplitude is just half of the expected one as shown in Fig. 4.3(b). Previous frequency domain adaptive algorithms (2) have used the Hanning window, that provides a wrong estimation as shown in Fig. 4.3(c). However Zero padding in x + rectangular window is shown to provide a better estimation of the true response of the primary path, as shown in Fig. 4.3(d).



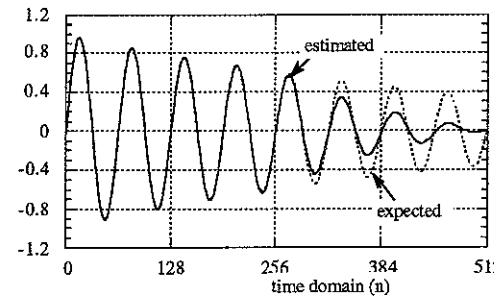
(a) Rectangular window



(b) Rectangular window + Zero padding on y



(c) Hanning window



(d) Zero padding on x + Rectangular window

an unbiased estimation is expected in the range of $0 \leq n < 256$

Fig. 4.3 Estimation of a Primary Path by Block Averaging Method

5. Effect on Acausal Path

In estimation of the impulse response of the primary path with circular convolution, if there is delay in sensing input signal as shown in Fig. 5.1, the estimated impulse response has some non-real acausal part by wrap-around effects. This situation usually occurs in active noise control. In this section we consider the wrap-around effects when window functions are applied.

Assume that primary path is a simple delay expressed in equation (3.1), and the delay in sensing input signal is expressed as:

$$d(i) = \delta(i - i_2) \qquad (i = 0,1,...,N-1,\ i_1 < i_2) \tag{5.1}$$

In this case, estimated impulse response obtained should be zero, there will, however, be some non-real acausal part. Because there exists only acausal part, the cross-correlation in circular convolution is obtained as:

$$R_{xy}^{'}(-j) = \frac{1}{N}\sum_{i=0}^{j-1} x_1(i)\, y_1(i-j+N) + \frac{1}{N}\sum_{i=j}^{N-1} x_1(i)\, y_1(i-j) \tag{5.2}$$

The second term involves valid data, while the first term consists of only invalid data. If block averaging is processed, it changes into:

$$E\{R_{xy}^{'}(-j)\} = R_{xy}(-j)\,\frac{1}{N}\sum_{i=j}^{N-1} f_x(i)\, f_y(i-j) \tag{5.3}$$

The auto-correlation in circular convolution at j=0 is:

$$R_{xx}^{'}(0) = \frac{1}{N}\sum_{i=0}^{N-1} x_1(i)\, x_1(i) \tag{5.4}$$

If block averaging is processed, it changes into:

$$E\{R_{xx}^{'}(0)\} = R_{xx}(0)\,\frac{1}{N}\sum_{i=0}^{N-1} f_x^2(i) \tag{5.5}$$

As the primary path is a simple delay, the estimation of the primary path is easily obtained. The non-real impulse will be appear at (i1-i2+N)th element, i.e.,:

$$\hat{w}(i_1 - i_2 + N) = \hat{w}(i_1 - i_2) = \frac{E\{R_{xy}^{'}(i_1 - i_2)\}}{E\{R_{xx}^{'}(0)\}} = w(i_1 - i_2)\, m(i_1 - i_2) \tag{5.6}$$

where

$$w(i_1 - i_2) = \frac{R_{xy}(i_1 - i_2)}{R_{xx}(0)} \tag{5.7}$$

is true estimation, and:

$$m(i_1 - i_2 + N) = m(i_1 - i_2) = \frac{q(i_1 - i_2)}{q_1} \tag{5.8}$$

is considered as leaked impulse amplitude, that provides non-real image of an acausal impulse. Where q(i1-i2) and q1 are:

$$q(i_1 - i_2) = \frac{1}{N}\sum_{i=i_2-i_1}^{N-1} f_x(i)\, f_y(i+i_1-i_2) \tag{5.9}$$

$$q_1 = \frac{1}{N}\sum_{i=0}^{N-1} f_x^2(i) \tag{5.10}$$

Replacing i1-i2+N with i, m(i) is considered as a leak function, which expresses how wrap-around effect disturbs the correct true impulse response at every ith element. Now we can calculate this function about several window functions.

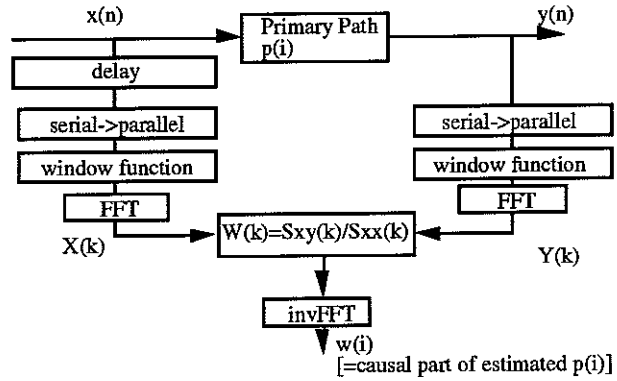


Fig. 5.1 Estimation of Impulse Response of Primary Path under exist of Delay in Input Signal

Example 1 Rectangular Window

The leak function m(i) for equation (3.15) is obtained as:

$$m(i) = \frac{i}{N} \tag{5.11}$$

Example 2 Zero padding in output signal

The leak function m(i) for equation (3.19) is obtained as:

$$m(i) = \begin{cases} 0 & 0 \le i < N/2 \\ \dfrac{2i-N}{2N} & otherwise \end{cases} \tag{5.12}$$

Example 3 Zero padding in input signal

The leak function m(i) for equation (3.23) is obtained as:

$$m(i) = \begin{cases} 0 & 0 \le i < N/2 \\ \dfrac{2i-N}{N} & otherwise \end{cases} \tag{5.13}$$

Example 4 Hanning window

The leak function m(i) for equation (3.27) is obtained as:

$$m(i) = \frac{2}{3}\frac{i}{N}\left(1 + \frac{1}{2}\cos 2\pi\frac{i}{N}\right) - \frac{1}{3N} C \sin 2\pi\frac{i}{N} \tag{5.14}$$

$$C = \frac{2\sin^2 2\pi\frac{1}{N} - \cos 2\pi\frac{1}{N} + \cos^2 2\pi\frac{1}{N}}{\left(1 - \cos 2\pi\frac{1}{N}\right)\sin 2\pi\frac{1}{N}} \tag{5.15}$$

These results of examples are shown in Fig. 5.2. It is found that both rectangular window and Hanning window contaminate first half of true impulse response. Zero padding on x doesn't affect the first half of the true impulse response. As zero padding on y doesn't affect in first half of true impulse response either, it can be used in this purpose.

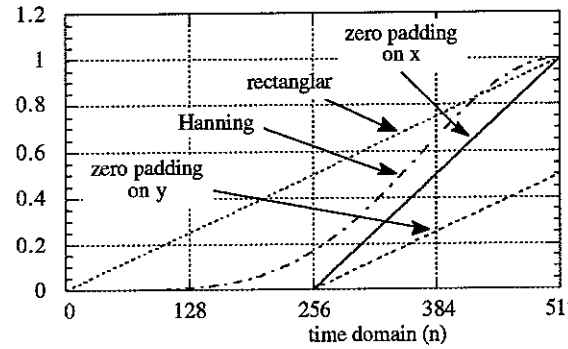
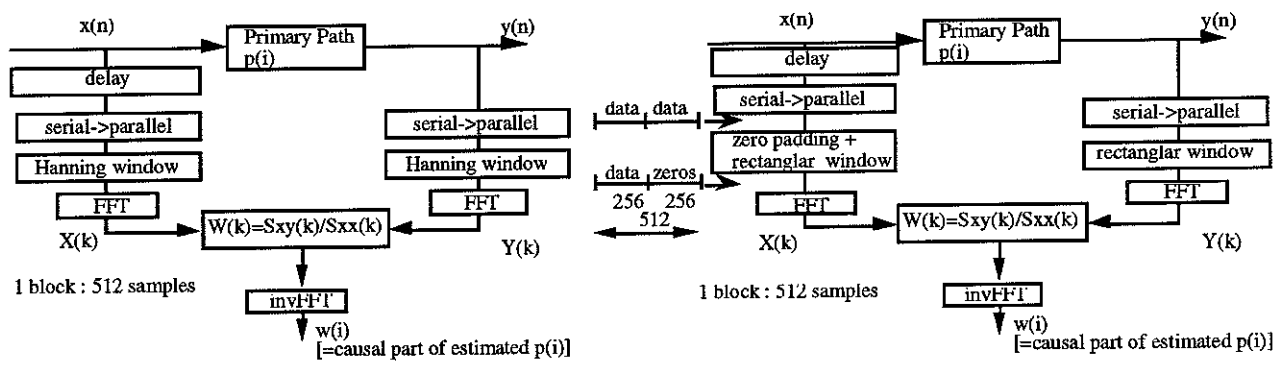


Fig. 5.2 the Leak Functions in Estimating Impulse Response depending on the Window Function

6. Comparison of Window Functions

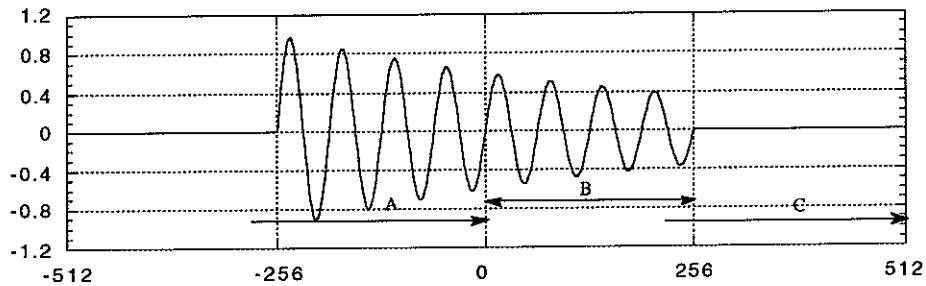
To ascertain above, we compared algorithms using Hanning window and zero padding + rectangular window on estimating an impulse response by block averaging method as shown in Fig. 6.1. In the estimation the number of sample points of one block was N=512, and the number of averaging blocks was L=1024 and 8192. The impulse response of the primary path used here is shown in Fig. 4.2. The input signal was white noise. The lengths of delay located at input signal, that examined here, were 256 and 512.

Fig. 6.2 shows estimated impulse response of a primary path by block averaging method with Hanning window and zero padding. The length of delay located at input signal was 256. The number of averaging was L=1024. An unbiased estimation is expected in the range of $0 \le n < 256$. Acausal non-real images are seen in the range of $256 \le n < 512$. Unbiased estimation is obtained in first half part, when zero padding was applied, while Hanning window provided wrong estimation in first half part. This result is the same as shown in Fig. 4.3. Both of window functions provided acausal non-real image in the second half by wrap-around effects, however the shapes were distorted following the leak function.

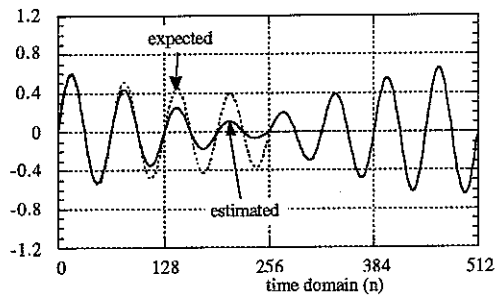


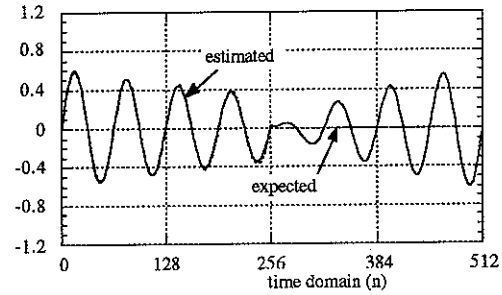
(a) Hanning window (b) zero padding + rectangular window

Fig. 6.1 Comparison of Window Functions



The section B is expected to be unbiased estimation.
The section A may appear in the section C as an acausal non-real image.
(c) theoritical impulse response to be obtained





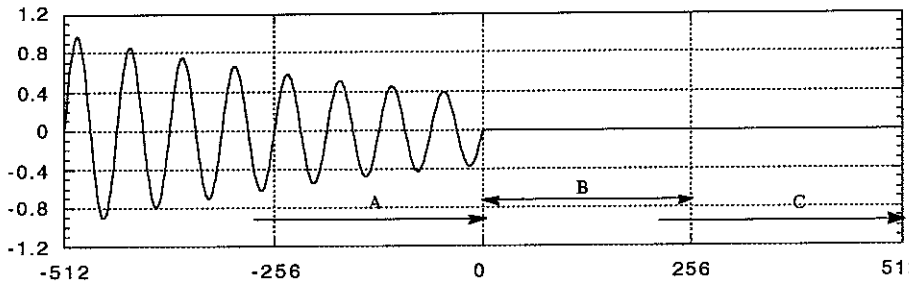
(a) Hanning window (b) Zero padding on x + Rectangular window

The length of delay located at input signal was 256. The number of averaging was L=1024. Unbiased estimation is expected in the range of $0 \le n < 256$. Acausal non-real images appear in the range of $256 \le n < 512$.

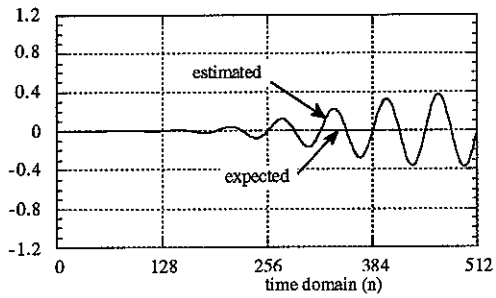
Fig. 6.2 Estimation of a Primary Path by Block Averaging Method

Fig. 6.3 also shows estimated impulse response of a primary path by block averaging method with Hanning window and zero padding. The length of delay located at input signal was 512. The number of averaging was L=8192. An unbiased estimation, that is zero, is expected in the range of $0 \le n < 256$. Acausal non-real images are seen in the range of $256 \le n < 512$. Unbiased estimation is obtained in first half part, when zero padding was applied, however, there is noise, that will be decrease if more averaging blocks are applied. Hanning windows provided the wrong estimation in first half part. This result is the same as shown in Fig. 4.3. Both of them provided acausal non-real images by wrap-around effects, however the shapes were distorted following the leak function.

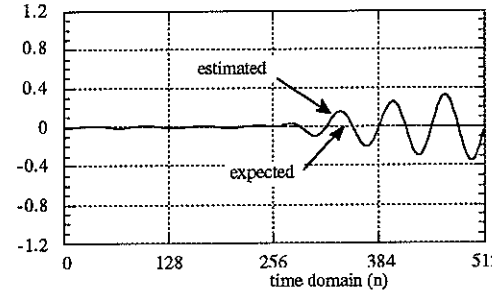
These results illustrate the danger of using the Hanning window or the rectangular one used in previous work[4].



The section B is expected to be unbiased estimation.
The section A may appear in the section C as an acausal non-real image.
(c) theoretical impulse response to be obtained



(a) Hanning window



(b) Zero padding on x + Rectangular window

The length of delay located at input signal was 512. The number of averaging was L=8192. Unbiased estimation, that is zero, is expected in the range of $0 \leq n < 256$. Acausal non-real images appear in the range of $256 \leq n < 512$.

Fig. 6.3 Estimation of a Primary Path by Block Averaging Method

7. Application to SISO active control system

7.1 frequency domain filtered-X LMS algorithm

It was proved that using the zeropadding + rectangular window offers unbiased cross spectra density. The zeropadding + rectangular window was applied to the single input single output active control system in Fig. 7.1.

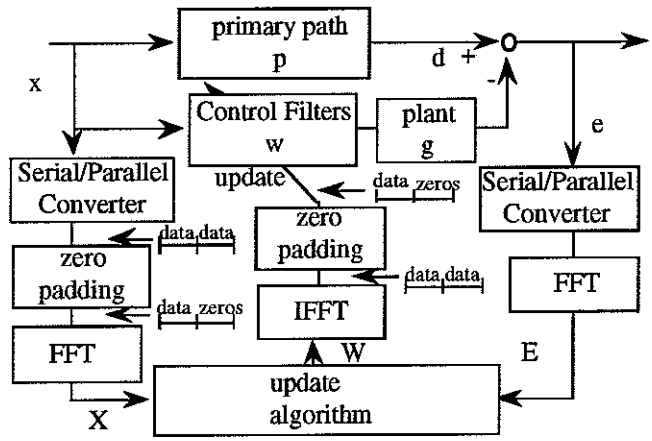


Fig. 7.1 the single input single output active control system

In this system, updating control filter is performed in frequency domain as a background task, while control filtering is performed in time domain.

In Fig.7.1 x, d and e are the input signal, the desired signal and the error signal in time domain respectively. X, D and E are their frequency domain notation.

In each frequency bin, the error and the cost function are expressed as:

$$E = D - GWX \tag{7.1}$$

$$J = \mathrm{E}\{E^*E\} \tag{7.2}$$

where G is the plant transfer function, W is the transfer function of the control filter. The descent of the cost function is :

$$\frac{\partial J}{\partial W_R} + j\frac{\partial J}{\partial W_I} = -2G^*\mathrm{E}\{X^*E\} \tag{7.3}$$

So the frequency domain filtered-X LMS algorithm is:

$$W_{new} = W_{old} + \alpha\, G^*X^*E \tag{7.4}$$

The convergence coefficient for each frequency bin can be normalized so that each bin converges at the same rate by defining

$$\alpha_{normalized} = \alpha\,[\,|G|^2\,\mathrm{E}\{X^*X\}]^{-1}, \tag{7.5}$$

where α is an overall convergence factor in equation (7.4).

Therefore renewing α, the normalized steepest descent algorithm is expressed as

$$W_{new} = W_{old} + \frac{\alpha}{\mathrm{E}\{X^*X\}}\frac{G^*}{G^*G}X^*E, \quad (0 < \alpha < 1) \tag{7.6}$$

The Newton's algorithm in frequency domain also becomes the equation (7.6).

$\mathrm{E}\{X(k)^*X(k)\}$ is estimated by using

$$\mathrm{E}\{X^*X\}] = P + \gamma, \tag{7.7}$$

$$P_{new} = P_{old}\,\beta + |X|^2\,(1-\beta), \quad 0 < \beta < 1, \tag{7.8}$$

where P is the power estimation, and the factor β is introduced to assure an unbiased estimation, and γ is a small positive constant, that prevents $E\{X^*X\}$ being zero. $X(k)^*X(k)$ is calculated for each block of data.

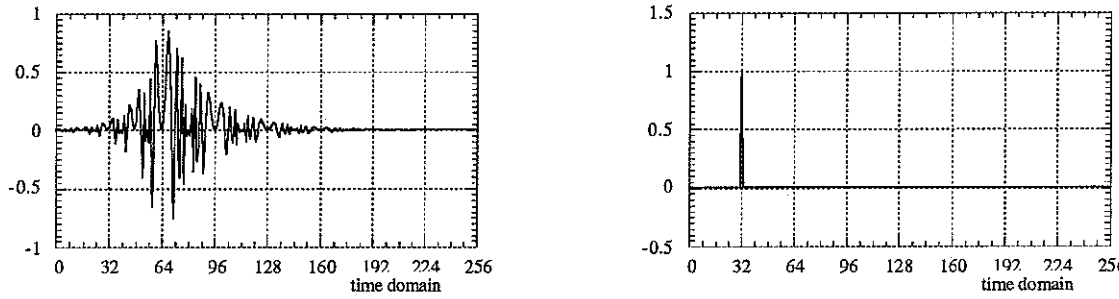
7.2 simulations

The frequency domain and time domain filtered-X algorithms are simulated in two situations. One has simple delay plant g, the other has simple delay + resonance plant g. In each situation, both white noise reference and coloured(pink) noise reference are supplied.

Simulation 1

The primary path is assumed as shown Fig. 7.2(a), that has several resonances. The plant path is assumed as shown Fig. 7.2(b), that is a simple delay. The length of control filter coefficients is 256, therefore in frequency domain the block size of samples is 512. The reference signals supplied in the simulation are white one and coloured one as shown in Fig. 7.3.

The simulation result is shown in Fig. 7.4. In the simulation with white noise reference, the convergence curves of both the time domain and frequency domain algorithms are almost same. But in the simulation with coloured(pink) noise reference, the frequency domain algorithm shows better convergence than the time domain one. This is because the convergence of the time domain filtered-X algorithm is limited by the eigenvalue spread of the autocorrelation matrix of the reference signal.



(a) assumed primary path impulse response (b) assumed plant impulse response

Fig. 7.2 impulse responses in the simulation

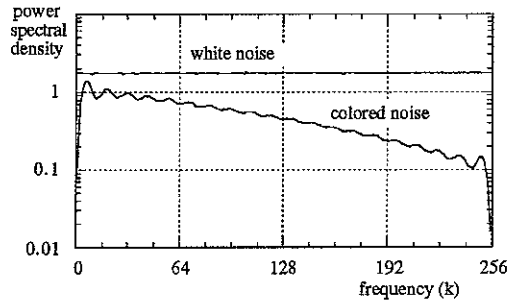
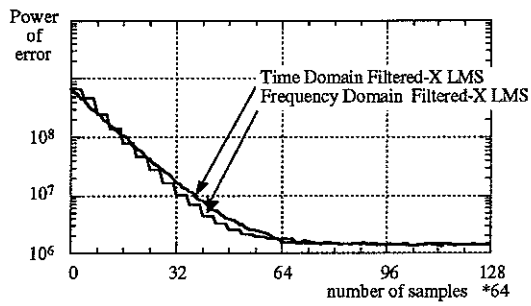
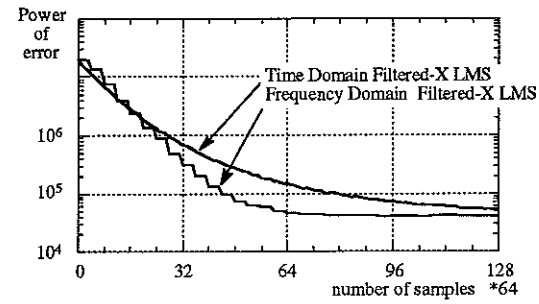


Fig. 7.3 white reference signal and coloured reference signal





Ensemble Average of 32 Learning Curves (a)white noise reference

Ensemble Average of 32 Learning Curves (b)coloured noise reference

Fig. 7.4 Power of Error in Frequency and Time Domain Filtered-X LMS Algorithms, in the case when plant g is simple delay

Simulation 2

The primary path is assumed as shown Fig. 7.2(a), that has several resonances. The plant path is assumed as shown Fig. 7.5, that has single resonance. The length of control filter coefficients is 256, therefore in frequency domain the block size of samples is 512. The reference signals supplied in the simulation are white one and coloured one as shown in Fig. 7.3.

The simulation result is shown in Fig. 7.6. In the both simulations with white or coloured(pink) noise reference, the frequency domain algorithm shows quick convergence than the time domain one. This is because the convergence of the time domain filtered-X algorithm is limited by the eigenvalue spread of the autocorrelation matrix of the reference signal filtered by the plant, which is resonant, rather than that of the reference signal itself.

The frequency domain algorithm shows less reduction when it converged. This is because as in some frequency bins $1/(G^*G)$ has large value, an apparent convergence constant α looks like a large value at that frequency.

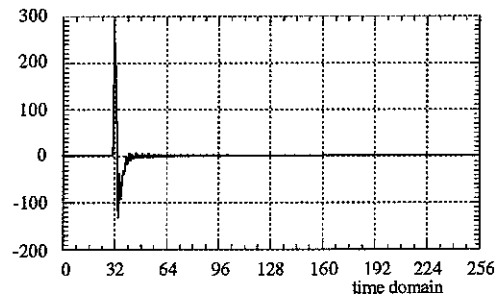
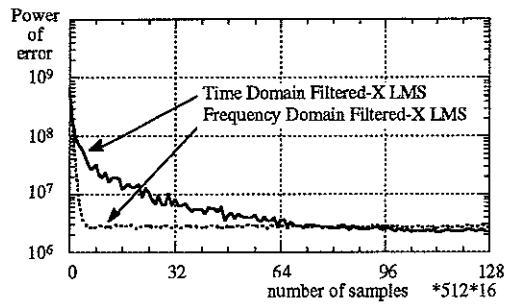
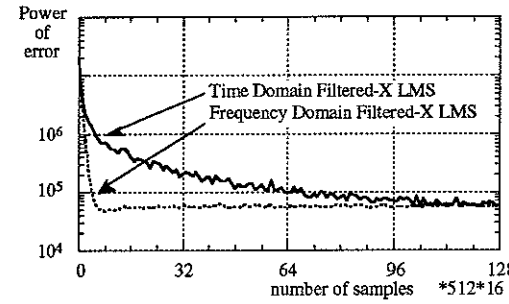


Fig. 7.5 assumed plant impulse response





Ensemble Average of 8 Learning Curves (a)white noise reference

Ensemble Average of 8 Learning Curves (b)coloured noise reference

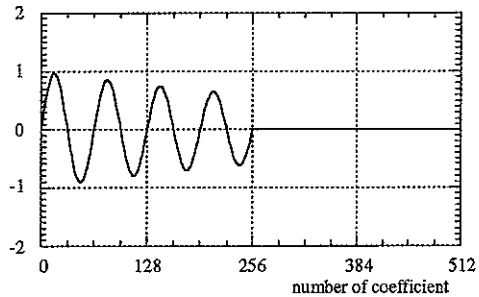
Fig. 7.6 Power of Error in Frequency and Time Domain Filtered-X LMS Algorithms, in the case when plant g has resonance

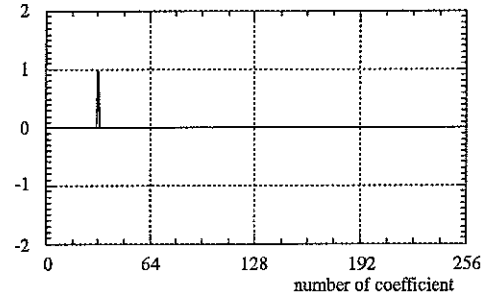
In these two simulations, when the reference is white noise and the plant is a simple delay, both the time domain algorithm and frequency domain algorithm converge with the same convergence curve. If either the reference is coloured noise or the plant is not a simple delay but has resonances, there is the eigenvalue spread of the autocorrelation matrix of the reference signal filtered by the plant, which is resonant, or/and that of the reference signal itself, and the frequency domain algorithm converges much faster than the time domain one.

7.3 converged control filter

To clarify if the frequency domain algorithm lets the control filter converges to the optimum one, a simple simulation was taken place. The impulse responses of the primary path and plant used here is shown in Fig. 7.7, and the causal part of unconstrained filter suitable to this situation that is not always an optimum filter, however, is also shown in Fig 7.8.

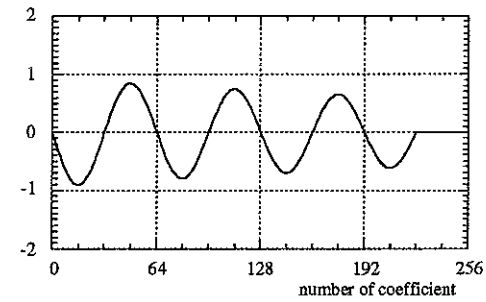
In the case when the reference is white noise, both the time domain and frequency domain LMS algorithm converges to the causal part of unconstrained filter that is the optimum one in this case. In the case when the reference is pink noise, the time domain LMS algorithm converges to the optimum filter, and the frequency domain LMS algorithm described in equation 7.4 also converges to the optimum filter, but the frequency domain LMS with Newton's algorithm described in equation 7.6 doesn't converge to either the optimum filter or the causal part of unconstrained filter, as shown in Fig. 7.9.



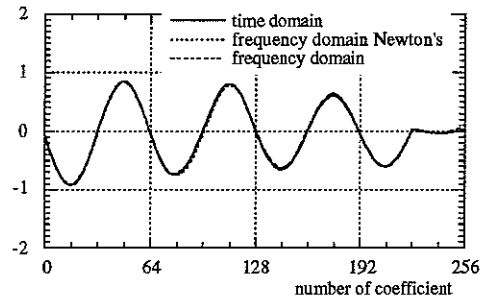


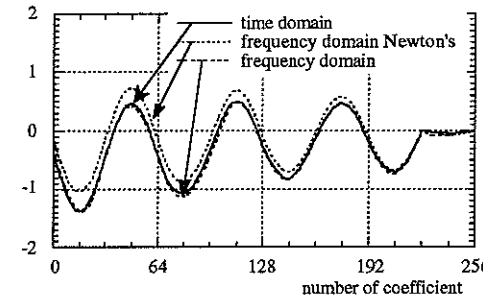
(a) primary path (b) plant

Fig.7.7 The impulse responses of the simulation model



In the case when the reference is white noise this is optimum
Fig. 7.8 the causal part of the unconstrained filter





(a) reference is white noise (b) reference is pink noise
"frequency domain Newton's" is the case of equation 7.6.
"frequency domain" is the case of equation 7.4.
Fig.7.9 coefficients of the converged control filters

8. Application to MIMO active controle system

8.1 frequency domain filtered-X LMS algorithm

It was proved that using the zeropadding + rectangular window offers unbiased cross spectra density. The zeropadding + rectangular window was applied to the multi-input multi-output active control system in Fig. 8.1.

In this system, updating control filter is performed in frequency domain as a background task, while control filtering is performed in time domain as well as SISO system.

In Fig.8.1 $\boldsymbol{x}$, $\boldsymbol{d}$ and $\boldsymbol{e}$ are the input signal vector, the desired signal vector and the error signal vector in time domain respectively. $\underline{\boldsymbol{x}}$, $\underline{\boldsymbol{d}}$ and $\underline{\boldsymbol{e}}$ are their frequency domain notation.

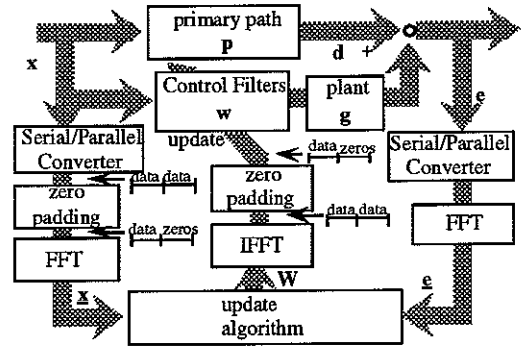


Fig. 8.1 the multi-input multi-output active control system

$$\underline{e} = (E_1, E_2, \cdots, E_L)^T \quad , \quad \underline{d} = (D_1, D_2, \cdots, D_L)^T \quad ,$$

$$G = \begin{pmatrix} G_{11}, G_{12}, \cdots, G_{1M} \\ G_{21}, G_{22}, \cdots, G_{2M} \\ \vdots \\ G_{L1}, G_{L2}, \cdots, G_{LM} \end{pmatrix} \quad W = \begin{pmatrix} W_{11}, W_{12}, \cdots, W_{1K} \\ W_{21}, W_{22}, \cdots, W_{2K} \\ \vdots \\ W_{M1}, W_{M2}, \cdots, W_{MK} \end{pmatrix} \quad ,$$

$$\underline{x} = (X_1, X_2, \cdots, X_K)^T \tag{8.6}$$

In each frequency bin, the error and the cost function are expressed as:

$$\underline{e} = \underline{d} - GW\underline{x} \tag{8.7}$$

$$J = \mathrm{E}\{G^H G\} \tag{8.8}$$

where G is the plant transfer function, W is the transfer function of the control filter. The descent of the cost function is :

$$\frac{\partial J}{\partial W_R} + j\frac{\partial J}{\partial W_I} = 2\,\mathrm{E}\{G^H \underline{e}\ \underline{x}^H\} \tag{8.9}$$

So the frequency domain filtered-X LMS algorithm is :

$$W_{new} = W_{old} + \alpha G^H \underline{e}\ \underline{x}^H \tag{8.10}$$

A normalized steepest descent algorithm is implemented as:

$$W_{new} = W_{old} + \alpha \frac{1}{\sum_{l=1}^{L}\sum_{m=1}^{M} |G_{lm}|^2} G^H \underline{e}\ \underline{x}^H \mathrm{E}[\underline{x}\ \underline{x}^H]^{-1} \tag{8.11}$$

, which is for comparing work.

The Newton's algorithm is[6]:

$$W_{new} = W_{old} + \alpha [G^H G]^{-1} G^H \underline{e}\ \underline{x}^H \mathrm{E}[\underline{x}\ \underline{x}^H]^{-1} \quad (0 < \alpha < 1) \tag{8.12}$$

To eliminate divergence of control filter, the Newton's algorithm is modified as:

$$W_{new} = W_{old} + \alpha [G^H G + \gamma I]^{-1} G^H \underline{e}\ \underline{x}^H \mathrm{E}[\underline{x}\ \underline{x}^H]^{-1} \quad (0 < \alpha < 1) \tag{8.13}$$

where

$$\gamma = \begin{cases} 0 & \lambda_1 / \lambda_2 > r \\ \lambda_2 r & \lambda_1 / \lambda_2 \le r \end{cases} \tag{8.14}$$

where λ_1, λ_2 ($\lambda_1 < \lambda_2$) are min and max eigenvalues of $G^H G$, and r is a threshold value. In the case when $\gamma = 0$, equation (8.13) is the pure Newton's Algorithm, and when γ is large the equation is similar to steepest descent, therefore equation (8.13) is considered as an intermediate form between the pure Newton's and the Steepest Descent algorithms[7].

In these simulations, $\mathrm{E}[\underline{x}\ \underline{x}^H]^{-1}$ was calculated in the same way as single channel simulation.

As $[G^H G+\gamma I]^{-1} G^H$ is calculated once at the beginning of the adaptation, the real time computation requirements of the two algorithms are comparable.

The effects of the plant coupling and the autocorrelation structure of the reference signals can be accounted for in a frequency domain version of Newton's algorithm.

8.2 simulation of one reference two sources two error sensors system

The frequency domain and time domain filtered-X algorithms are simulated in two arrangements with two kinds of transfer function g_{lm}. One arrangement has an ill conditioned frequency bin, while the other has no ill conditioned frequency bins, shown in Fig. 8.2. The transfer functions employed in one configuration are simple delays, while ones in the other configration have resonances, whose impulse response is shown in Fig. 7.5.

The error in the one reference-two sources-two error sensors system is expressed as;

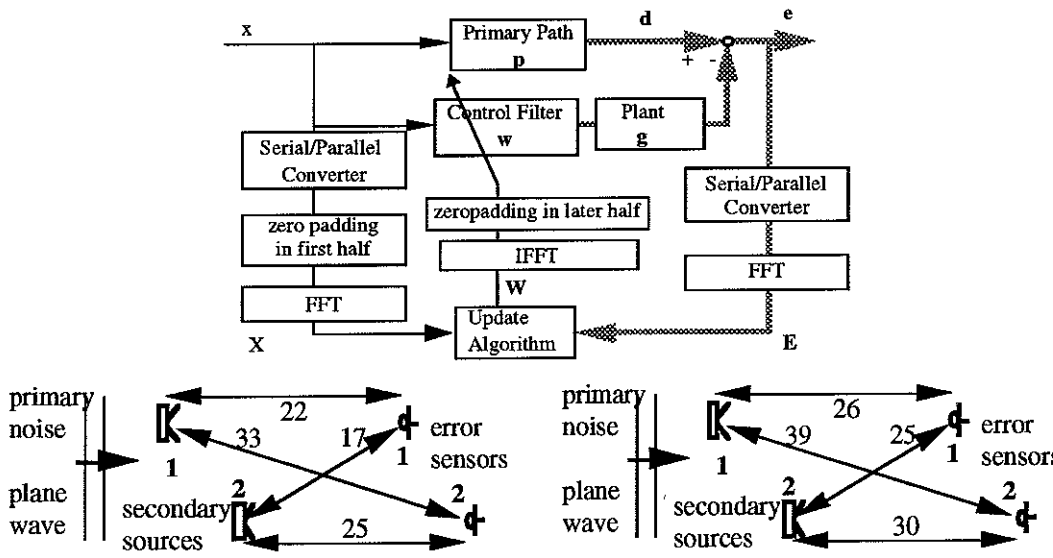
$$(\underline{e}_n - \underline{e}_{opt})^H(\underline{e}_n - \underline{e}_{opt}) = (1-\alpha\lambda_1)^{2n}|a_1|^2\lambda_1 + (1-\alpha\lambda_2)^{2n}|a_2|^2\lambda_2 \tag{8.15}$$

$$G^H G\ \underline{q}_m = \lambda_m \underline{q}_m \tag{8.16}$$

$$W_{opt}\underline{x} = \sum_{m=1}^{M} a_m \underline{q}_m \tag{8.17}$$

where λ and q are the eigenvalue and eigenvector of $G^H G$ respectively.

This means that power of error is decomposed into modes $|a_m|^2\lambda_m$, and the power in each mode decreases with its descent $(1-\alpha\lambda_m)^{2n}$. If the eigenvalue of the mode is small, the power of the mode decrease slowly.



arrangement A
with an ill conditioned frequency bin

arrangement B
without ill conditioned frequency bins

Fig. 8.2 arrangements of one reference two sources two error sensors systems

8.3 results of the simulation

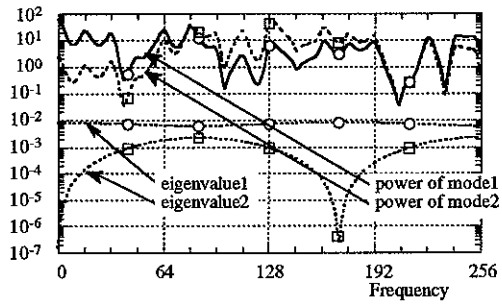
The primary paths are the same as shown in Fig. 7.2(a), but have different delays. In the case when plant transfer function g_{lm} has resonance, the impulse response of the plant is shown in Fig. 7.5, but has different delay. The reference signals are the same as shown in Fig. 7.3. The step gain α=0.001 and the threshold r=0.1 are used in each simulation.

The results are shown in Fig. 8.3 - 8.6. These reults show followings.

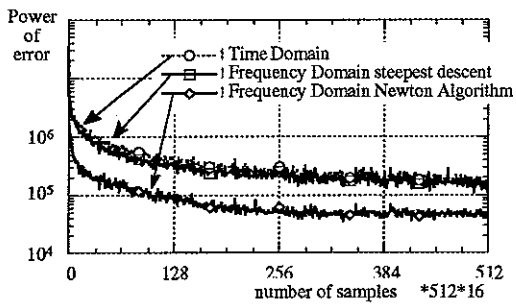
In the case when the reference signal is white or plants g are simple delays, both the time domain filtered-X LMS and the frequency domain filtered-X LMS algorithm using the steepest descent algorithm work comparably, and the frequency domain filtered-X LMS using Newton's Algorithm works much better than them, even if arrangement of secondary sources and error sensors has an ill conditioned frequency bin or the plants g have resonances.

In the case when the reference signal is coloured and plants g have resonances, the frequency domain filtered-X LMS algorithm using the steepest descent algorithm works better than the time domain one, and the frequency domain filtered-X LMS using Newton's Algorithm works much better than them, even if arrangement of secondary sources and error sensors has an ill conditioned frequency bin or the plants g have resonances.

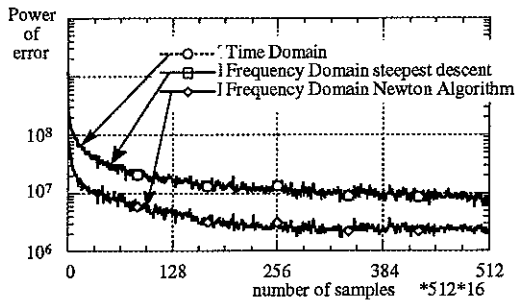
This means that the frequency domain filtered-X LMS using Newton's Algorithm overcomes the convergence difficulty caused by the eigenvalue spread of the autocorrelation matrix of the reference signal filtered by the plant, which is resonant, or/and that of the reference signal itself and by the potentially slow convergence mode due to the spatial arrangement of secondary sources and error sensors.



(a) eigenvalues G^HG and powers of modes

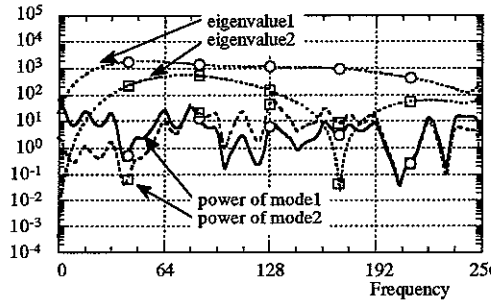


(b) white noise reference

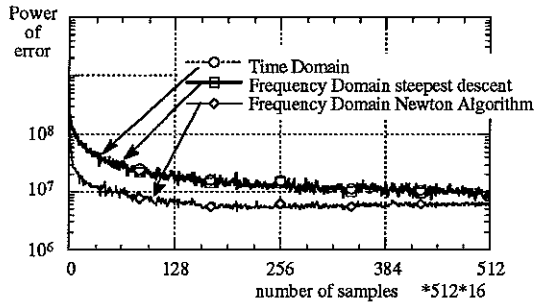


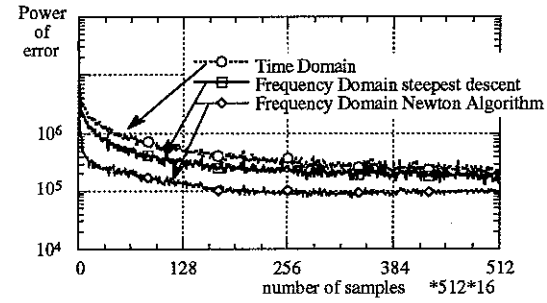
(c) pink noise reference

Fig. 8.3 Simulation result in the case when an arrangement of secondary sources and error sensors causes an ill conditioned frequency bin, and transfer functions g_{lm}s are simple delays.



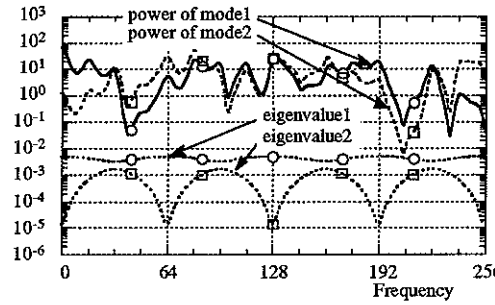
(a) eigenvalues G^HG and powers of modes



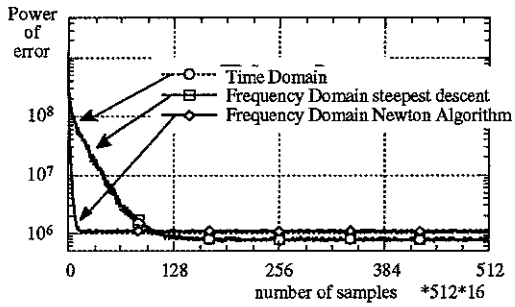


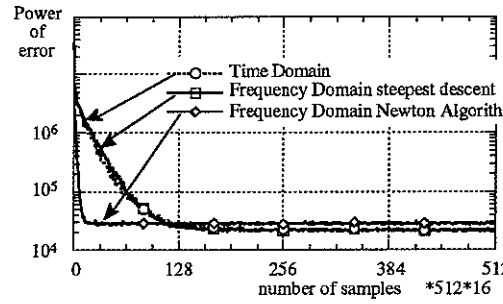
(b) white noise reference (c) pink noise reference

Fig. 8.4 Simulation result in the case when an arrangement of secondary sources and error sensors causes an ill conditioned frequency bin, and transfer functions g_{lm}s have resonances.



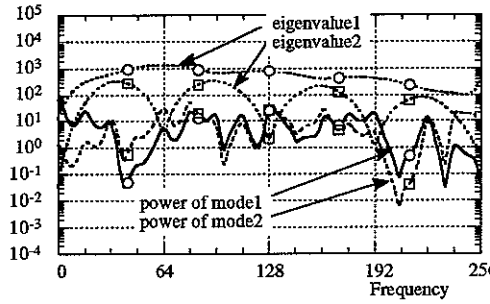
(a)eigenvalues G^HG and powers of modes



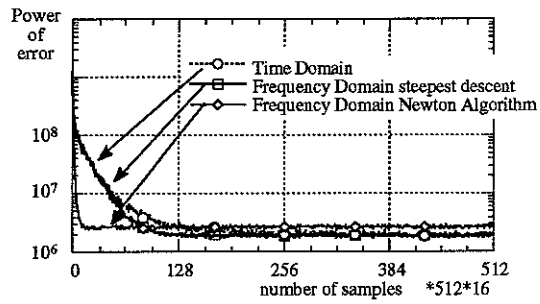


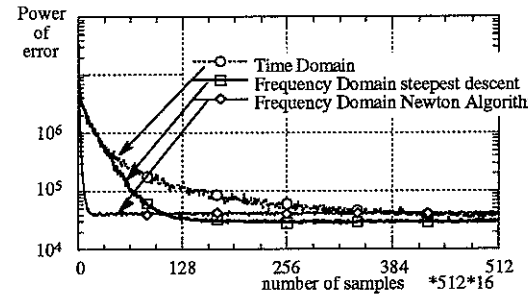
(b) white noise reference (c) pink noise reference

Fig. 8.5 Simulation result in the case when an arrangement of secondary sources and error sensors causes no ill conditioned frequency bins, and transfer functions g_{lm}s are simple delays.



(a) eigenvalues G^HG and powers of modes





(b) white noise reference (c) pink noise reference

Fig. 8.6 Simulation result in the case when an arrangement of secondary sources and error sensors causes no ill conditioned frequency bins, and transfer functions g_{lm}s have resonances.

8.4. convergence of the frequency domain filtered-X LMS using Newton's Algorithm in MIMO system

The time domain filtered-X LMS converges to the optimal solution. The frequency domain filtered-X LMS with steepest descent algorithm converges close to the optimal solution. The frequency domain filtered-X LMS with pure Newton's algorithm does not converges to optimal solution, because it converges to optimal solution in the case when all of signals and transfer functions are expressed in frequency domain and causality is ignored. This causes divergence of filter coefficients when the eigenvalues of G^HG spread in ill conditioned frequency bin. To avoid this a small positive constant is added to the diagonal elements of G^HG before inversion to give $[G^HG]^{-1}$ in the frequency bins which have the eigenvalue spread. The threshold to introduce this small positive constant effects the converged error shown in Fig. 8.7. Higher threshold offers small residual error and slow convergence speed, while lower threshold offers big residual error and quick convergence speed.

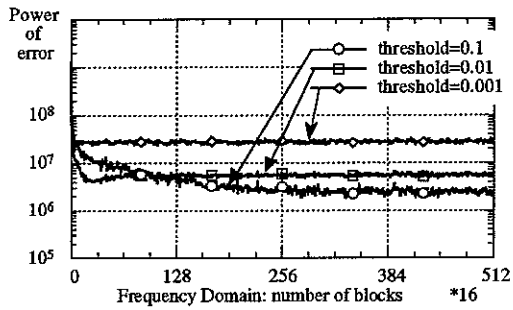


Fig. 8.7 effect of thresholds on convergence speed and residual error

9. Conclusions

The Frequency Domain LMS involves the estimation of a cross spectral density. There is a problem in choosing the correct window function, that is applied before FFT calculation, to obtain an unbiased estimation and to eliminate wrap-around effects.

This report shows the effect of the window functions in estimating the impulse response of a primary path. The effect appeared both as a modifier of the estimation and as eliminator of wrap-around effects. Both are obtained as the equations. Then some effects of window functions, that are the rectangular window, Hanning window and zero padding in output signal + rectangular window and zero padding in input signal + rectangular window, were shown in figures. These figures show the efficiency of zero padding in input signal + rectangular window, in the estimation of the impulse response of a primary path. This made clear that zero padding in input signal + rectangular window is suitable for estimation of impulse response using cross spectral density. It will be also suggested rectangular window + zero padding in output signal is suitable for estimation of cross spectral density.

Using zero padding in input signal + rectangular window the simulation of frequency domain filtered-X algorithm was examined. In the Frequency Domain Filtered-X LMS Algorithm used in this report, the updating of the control filter is performed in the frequency domain as a background task, while control filtering is performed in the time domain, thus minimizing the delay in the controller.

In single input single output system, the frequency domain filtered-X algorithm is compared with the time domain filtered-X algorithm. As the convergence of the time domain filtered-X algorithm is limited by the eigenvalue spread of the autocorrelation matrix of the reference signal filtered by the plant, which is resonant, rather than that of the reference signal itself, the frequency domain filtered-X algorithm converged more rapidly than a time domain LMS implementation even for a white noise reference signal.

In single input multi output system, the frequency domain filtered-X algorithm is compared with the time domain filtered-X algorithm. As the convergence of the time domain filtered-X algorithm is limited by the eigenvalue spread of the autocorrelation matrix of the reference signal filtered by the plant, which is resonant, rather than that of the reference signal itself and the frequency domain filtered-X with Newton's algorithm overcomes this problem as well as the case of SISO system. The frequency domain filtered-X with Newton's algorithm also overcomes the eigenvalue spread due to the spatial location of the transducers, the frequency domain filtered-X algorithm converged more rapidly than a time domain LMS implementation even for a white noise reference signal.

References:

[1]P.A. Nelson & S.J. Elliott, "Active Control of Sound" Academic Press, 1992

[2]Q.Shen, A.Spanias, "Frequency-domain adaptive algorithms for multi-channel active sound

control" Proc. Recent Advances in the Active Control of Sound & Vibration, 755-765, 1993
[3]Sen M. Kuo & Dennis R. Morgan Active Noise Control Systems,Wiley-Interscience Publication, 1996
[4]A.Roure, "Self adaptive broadband active sound control system" J of Sound and Vibration 101, 429-441, 1985
[5]S.J. Elliott, I.M. Stothers and P.A. Nelson, "A multiple error LMS algorithm and its application to the active control of sound and vibration" IEEE Trans. ASSP-35, 1423-1434, 1987
[6]S.J.Elliott, Transform-Domain Adaptation of Digital Control Filters, ISVR Technical Memorandum No.781, University of Southampton 1996
[7]C.C.Boucher, The behavior of multiple channel active control systems, PhD Thesis, University of Southampton 1992