EURASIP Journal on
Advances in Signal Processing
a SpringerOpen Journal

RESEARCH   **Open Access**

An active noise control algorithm with gain and power constraints on the adaptive filter

Walter J Kozacky and Tokunbo Ogunfunmi[*]

Abstract

This article develops a new adaptive filter algorithm intended for use in active noise control systems where it is required to place gain or power constraints on the filter output to prevent overdriving the transducer, or to maintain a specified system power budget. When the frequency-domain version of the least-mean-square algorithm is used for the adaptive filter, this limiting can be done directly in the frequency domain, allowing the adaptive filter response to be reduced in frequency regions of constraint violation, with minimal effect at other frequencies. We present the development of a new adaptive filter algorithm that uses a penalty function formulation to place multiple constraints on the filter directly in the frequency domain. The new algorithm performs better than existing ones in terms of improved convergence rate and frequency-selective limiting.

Keywords: Adaptive filtering, Adaptive signal processing, Discrete fourier transforms, FXLMS, Optimization

1. Introduction

Active noise control (ANC) systems can be used to remove interference by generating an anti-noise output that can be used in the system to destructively cancel the interference [1]. In some applications, it is required to limit the maximum output level to prevent overdriving the transducer, or to maintain a specified system power budget. In a frequency-domain implementation of the least-mean-square (LMS) algorithm, the limiting constraints can be placed directly in the frequency domain, allowing the adaptive filter response to be reduced in the frequency regions of constraint violation, with minimal effect at other frequencies [2]. Constraints can be placed on either the filter gain, or filter output power, as appropriate for the application.

A general block diagram of an ANC system is illustrated in Figure 1, with $H(z)$ representing the primary path or plant (e.g., an acoustic duct), $W(z)$ representing the adaptive filter, and $S(z)$ representing the secondary path (which may include the D/A converter, power output amplifier, and the transducer). The adaptive filter $W(z)$ typically uses the filtered-X LMS algorithm, where the input to the LMS algorithm is first filtered by an estimate of the secondary path [3]. The adaptive filter will need to simultaneously identify $H(z)$

and equalize $S(z)$, with the additional constraint of limiting the maximum level delivered to $S(z)$.

Applications of gain-constrained adaptive systems include systems that use a microphone for feedback, and due to the acoustic path to the microphone notches or peaks occur in the microphone frequency response (which may not be present in other locations). Adding gain constraints to the adaptive filter prevents distortion at those frequencies by limiting the peak magnitude of the filter coefficients [2]. Applications of power-constrained adaptive systems include requirements to limit the maximum power delivered to $S(z)$ to a predetermined constraint value to prevent overdriving the transducer, prevent output amplifier saturation, or prevent other nonlinear behavior [4]. The primary difference between these implementations is that the gain-constrained algorithm does not take the input power into account when determining the constraint violation.

Previous implementations of gain and output power limiting include output rescaling, the leaky LMS, and a class of algorithms termed constrained steepest descent (CSD) previously presented in [2]. We develop a new class of gain-constrained and power-constrained algorithms termed constrained minimal disturbance (CMD). The new CMD algorithms provide faster convergence compared to previous algorithms, and the ability to handle multiple constraints.

* Correspondence: togunfunmi@scu.edu
Santa Clara University, Santa Clara, CA, USA

Figure 1 Block diagram of the ANC system.
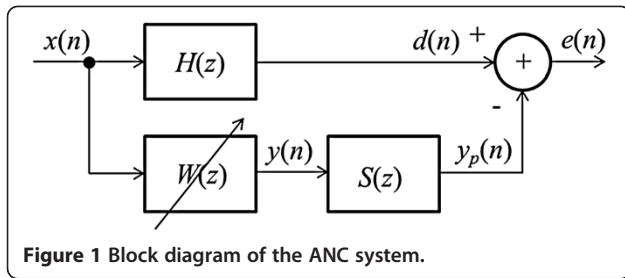
This article is organized as follows. Section 2 presents a review of prior work. Section 3 presents the CMD algorithm development. Section 4 presents a convergence analysis. Section 5 presents simulations with comparisons to other algorithms. Section 6 provides some concluding remarks.

2. Review of prior work

For comparison purposes, the following notation is used.

n Adaptive filter size and block size
N Sample number in the time domain
m Block number in the time or frequency domain
W Weight in the frequency domain
X Input in the frequency domain
E Error in the frequency domain
D Plant output in the frequency domain
Y Filter output in the frequency domain
C Gain or power constraint
S Secondary path

Lowercase w, x, e, d, and y are the time-domain representations of their respective frequency-domain counterparts. Vectors will be denoted in boldface, and the subscript k is used to denote an individual component of a vector. The superscript $*$ is used to denote complex conjugate, and the superscript T denotes vector transpose. The parameter μ is used as a convergence step-size coefficient, and the parameter γ is used as a leakage coefficient.

The first two methods of power limiting were described in detail in [5] and are briefly restated here. The first "output clipping" simply limits the output power to a maximum value. This is what would normally happen in a real system (e.g., the output amplifier would saturate). With the filter output y at iteration n denoted by $y(n)$ and the output constraint by C, the output clipping algorithm is given by

$$\text{if} \quad y(n+1) > C$$
$$y(n+1) = y(n+1)\frac{C}{|y(n+1)|}. \tag{1}$$

A potential problem in using output clipping for adaptive filtering applications is that the weight updates for

$\mathbf{w}(n)$ continue to occur while the filter output remains clipped, causing potential stability problems since the filter weight update is decoupled from the filter output. To prevent this, the "output re-scaling" algorithm can be used, which is given by

$$\text{if} \quad y(n+1) > C$$
$$y(n+1) = y(n+1)\frac{C}{|y(n+1)|} \tag{2}$$
$$\mathbf{w}(n+1) = \mathbf{w}(n+1)\frac{C}{|y(n+1)|}.$$

Here, in addition to the output being clipped, the adaptive filter weights are also rescaled; filter adaptation continues from the appropriate weight value corresponding to the actual output.

The next algorithm to be considered for gain or power limiting is the leaky LMS [6], which is given by

$$\mathbf{w}(n+1) = (1 - \mu\gamma)\mathbf{w}(n) + \mu e(n)\mathbf{x}(n). \tag{3}$$

The leaky LMS reduces the filter gain each iteration, with the leakage coefficient γ controlling the rate of reduction. The coefficient γ is determined experimentally according to the application, but gain reduction occurs at all frequencies, resulting in a larger steady-state convergence error. When the leakage is zero, this algorithm reduces to the standard LMS [7].

The algorithms described thus far are processed directly in the time domain. However, with large filter lengths the required convolutions become computationally expensive, and alternative methods can be more efficient. If the processing is done in block form and a fast Fourier transform (FFT) used, the required convolutions become multiplications. This also allows additional constraints to be added to limit the filter response directly in the frequency domain. For example, in [8], an ANC system using a loudspeaker with poor low-frequency response was stabilized in the FFT domain by zeroing out the low-frequency components, preventing adaptation at those frequencies. However, using block processing will result in a one block delay, which may be undesirable in some real-time applications. A delayless structure [9], with filtering in the time-domain and signal processing in the frequency-domain, can be used to mitigate this delay. A block diagram of the delayless frequency-domain LMS (FDLMS) is shown in Figure 2. In delayless ANC applications with a secondary path $S(z)$, the adaptive filter input vector $\mathbf{x}(m)$ is first filtered
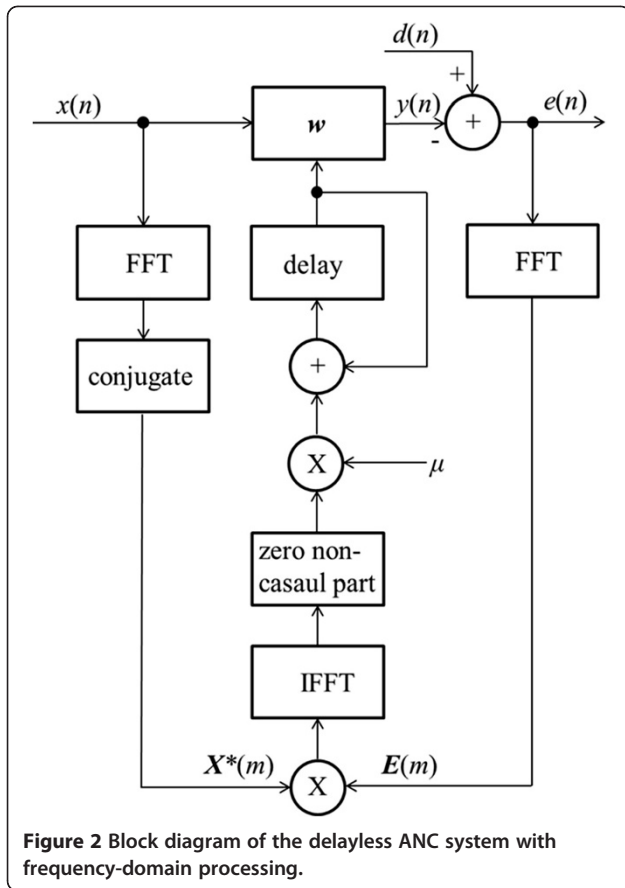
Figure 2 Block diagram of the delayless ANC system with frequency-domain processing.

by an estimate of the secondary path. The adaptive filter weight, input, and error vectors are defined as

$$\mathbf{w}(m) = [w_0(n)w_1(n)\ldots w_{N-1}(n)]^T$$

$$\mathbf{x}(m) = [x(n)x(n-1)\ldots x(n-N+1)]^T$$

$$\mathbf{e}(m) = [e(n)e(n-1)\ldots e(n-N+1)]^T \quad (4)$$

A block size of N is used for both the filter and each new set of data to maximize computational efficiency, with m representing the block iteration. To avoid circular convolution effects, each FFT uses blocks of size $2N$ [10]. The frequency-domain input and error vectors (size $2N$) are defined as

$$\mathbf{X}(m) = \text{FFT}\left\{[\mathbf{x}^T(n-N) \quad \mathbf{x}^T(n)]^T\right\}$$

$$\mathbf{E}(m) = \text{FFT}\left\{[\mathbf{0} \qquad \mathbf{e}^T(n)]^T\right\}, \qquad (5)$$

where $\mathbf{0}$ is the N-point zero vector.

The delayless FDLMS weight update equation at iteration m without a gain or power constraint is given by [9]

$$\mathbf{w}(m+1) = \mathbf{w}(m) + \mu\text{IFFT}\{\mathbf{X}^*(m)\mathbf{E}(m)\}_+, \qquad (6)$$

where the + subscript denotes the causal part of the IFFT (corresponding to the *gradient constraint* in [10]), and μ is the convergence coefficient. Adding a leakage factor to (6) results in a frequency-domain version of the leaky LMS which can be used to limit the adaptive filter output [2], and is given by

$$\mathbf{w}(m+1) = \gamma\mathbf{w}(m) + \mu\text{IFFT}\{\mathbf{X}^*(m)\mathbf{E}(m)\}_+ \qquad (7)$$

where γ is the leakage factor.

The next two weight update equations were developed in [2], which processes the constraints in the frequency domain using an algorithm based on the method of steepest descent. The delayless form of the gain-constrained version is given by

$$\mathbf{w}(m+1) = \mathbf{w}(m)$$

$$+\mu\text{IFFT}\left\{\mathbf{X}^*(m)\mathbf{E}(m) - 4\alpha N\left[\mathbf{W}(m)^2 - C)\right]_z\mathbf{W}(m)\right\}_+ \qquad (8)$$

where the z subscript sets the result in the brackets to 0 if the value in the brackets is less than 0 (the constraint is satisfied), or to the value of the difference (the constraint is violated). The constraint is individually applied to each frequency bin. Here, α controls the "tightness" of the penalty: a larger α places a stiffer penalty on constraint violation at the expense of a larger steady-state convergence error.

The delayless form of the power-constrained algorithm is given by

$$\mathbf{w}(m+1) = \mathbf{w}(m)$$

$$+\mu\text{IFFT}\left\{\mathbf{X}^*(m)\mathbf{E}(m) - 4\alpha[P(m) - C]_Z\mathbf{X}(m)^2\mathbf{W}(m)\right\}_+. \qquad (9)$$

The output power $P(m)$ is determined by the squared Euclidean norm of the filter output, which is required to be limited to a constraint value C, or equivalently

$$P(m) = \mathbf{y}(m)^2 < C. \qquad (10)$$

Note that in (10) there is only one constraint. When used for comparison purposes, we will denote (8) and (9) as constrained steepest descent (CSD) algorithms.

3. New algorithm development
The new CMD algorithm will be developed using the *principle of minimal disturbance*, which states that the

weight vector should be changed in a minimal manner from one iteration to the next [11]. A constraint is added for filter convergence, and a constraint is also added for either the filter gain (coefficient's magnitude in each frequency bin), or the filter output power, depending on which we intend to limit. The method of Lagrange multipliers [11,12] is then used to solve this constrained optimization problem [13].

3.1. Gain-constrained algorithm

At each block update m, the new algorithm will minimize the squared Euclidean norm of the frequency-domain weight change in each individual frequency bin k, where the weight change is given by

$$\delta W_k(m+1) = W_k(m+1) - W_k(m), \quad (11)$$

subject to the condition of *a posteriori* filter convergence in the frequency domain

$$D_k(m) = S_k(m)W_k(m+1)X_k(m). \quad (12)$$

In gain-constrained applications, the algorithm will additionally add a penalty based on the amount of magnitude violation above a maximum constraint value, requiring

$$|W_k(m+1)| \leq \sqrt{C_k} \quad (13)$$

or equivalently

$$W_k(m+1)^2 \leq C_k. \quad (14)$$

The three requirements given by (11), (12), and (14) are combined into a single cost function, written as

$$J(m+1) = \delta W_k(m+1)^2$$
$$+ \text{Re}\{\lambda^*[D_k(m) - S_k(m)W_k(m+1)X_k(m)]\}$$
$$+ \alpha_{\max,k}\left[\left(W_k(m+1)^2 - C_k\right)^2\right]_z \quad (15)$$

where the Lagrange multiplier λ controls the convergence requirement of (12), and the Lagrange multiplier $\alpha_{\max,k}$ with subscript k controls the individual frequency bin magnitude constraint; parameter $\alpha_{\max,k}$ controls the "tightness" of the penalty term, with a larger value placing more emphasis on meeting the constraint at the expense of increasing the convergence error [2]. The cost function (15) is differentiated with respect to each of the three variables and set to 0. For each frequency bin k

$$\frac{\partial J(m+1)}{\partial W_k(m+1)} = 2[W_k(m+1) - W_k(m)]^* - \lambda^* S_k(m)X_k(m)$$
$$+ 2\alpha_k W_k^*(m+1)\left[W_k(m+1)^2 - C_k\right]_z = 0 \quad (16)$$

$$\frac{\partial J(m+1)}{\partial \lambda^*} = D_k(m) - S_k(m)W_k(m+1)X_k(m) = 0 \quad (17)$$

$$\frac{\partial J(m+1)}{\partial \alpha_{\max,k}} = \left[\left(W_k(m+1)^2 - C_k\right)^2\right]_z = 0. \quad (18)$$

Rearranging (16) gives

$$\left(1 + 2\alpha_{\max,k}\left[W_k(m+1)^2 - C_k\right]_z\right)W_k^*(m+1)$$
$$= W_k^*(m) + \frac{1}{2}\lambda^* S_k(m)X_k(m). \quad (19)$$

We now propose the following interpretation of the gain-constraint term. In steady state (after convergence), we would expect the successive weight values to be approximately the same for a small convergence coefficient step size. Therefore, as long as the constraint of (14) was satisfied in the previous iteration, the penalty is set to 0. However, if the magnitude of the filter weight exceeds the constraint value, then the penalty is scaled in proportion to the constraint violation (similar to the method in [14], which initiates the penalty at 90% of the constraint). We define

$$\alpha_k = 2\alpha_{\max,k}\left[W_k(m)^2 - C_k\right]_z \quad (20)$$

where the z subscript term will force α_k to zero if the constraint of (14) is satisfied.

Substituting (20) into (19) at frequency bin k, conjugating both sides, and rearranging into a recursion results in

$$W_k(m+1) = \frac{1}{[1+\alpha_k]}\left[W_k(m) + \frac{1}{2}\lambda S_k^*(m)X_k^*(m)\right]. \quad (21)$$

Substituting (21) into (12) gives

$$D_k(m) - S_k\left\{\frac{1}{[1+\alpha_k]}\left[W_k(m) + \frac{1}{2}\lambda S_k^*(m)X_k^*(m)\right]\right\}X_k(m) = 0. \quad (22)$$

Rearranging (22) yields

$$[D_k(m) - S_k(m)W_k(m)X_k(m)]$$
$$+ \alpha_k D_k(m) - \frac{1}{2}\lambda\|S_k(m)\|^2\|X_k(m)\|^2 = 0. \quad (23)$$

The first term in brackets is the error at frequency bin k, $E_k(m)$. Solving for λ results in

$$\lambda = \frac{2[E_k(m) + \alpha_k D_k(m)]}{S_k(m)^2 X_k(m)^2}. \quad (24)$$

Rearranging (21) into a recursion, using (24), and introducing a convergence step size parameter μ to control the rate of adaptation yields

$$W_k(m+1)$$
$$= \frac{1}{1+\alpha_k}\left[W_k(m) + \mu\frac{[E_k(m) + \alpha_k D_k(m)]}{S_k(m)^2 X_k(m)}S_k^*(m)X_k^*(m)\right].$$
(25)

Noting that $D_k(m)$ in (25) can be written as $D_k(m) = E_k(m) + S_k W_k(m)X_k(m)$ results in

$$W_k(m+1) = \left(\frac{1+\mu\alpha_k}{1+\alpha_k}\right)W_k(m)$$
$$+ \frac{\mu}{S_k(m)^2 X_k(m)^2}S_k^*(m)X_k^*(m)E_k(m).$$
(26)

For small μ, (26) can be approximated as

$$W_k(m+1) = \frac{1}{1+\alpha_k}W_k(m)$$
$$+ \frac{\mu}{S_k(m)^2 X_k(m)^2}S_k^*(m)X_k^*(m)E_k(m).$$
(27)

Using the definitions

$$\gamma_k = \frac{\alpha_k}{\mu(1+\alpha_k)}$$
(28)

and

$$\mu_k = \frac{\mu}{S_k(m)^2 X_k(m)^2}$$
(29)

the weight update given by (27) can be written for each frequency bin as

$$W_k(m+1) = (1 - \mu\gamma_k)W_k(m)$$
(30)
$$+ \mu_k S_k^*(m)X_k^*(m)E_k(m).$$

Taking the IFFT of both sides and casting into a delayless structure results in the new CMD algorithm given by

$$\mathbf{w}(m+1) = \mathbf{w}(m)$$

$$+\mu\mathrm{IFFT}\left\{\frac{\mathbf{S}(m)\mathbf{X}^*(m)\mathbf{E}(m)}{\mathbf{S}(m)^2\mathbf{X}(m)^2} - \mathbf{\Gamma}(m)\mathbf{W}(m)\right\}_+$$
(31)

where

$$\mathbf{\Gamma}(m) = \mathrm{diag}\left[\gamma_0(m), \gamma_1(m), \ldots, \gamma_{2N-1}(m)\right]$$
(32)

is a diagonal matrix of variable leakage factors as determined by (28).

The $\|\mathbf{X}(m)\|^2$ term provides an estimate of the input power $P_{x,k}(m)$ in frequency bin k,

$$P_{x,k}(m) = E\{X_k(m)^2\}$$
(33)

which can be determined recursively by [15]

$$P_{x,k}(m) = \beta P_{x,k}(m-1)$$
(34)
$$+(1-\beta)\left[X_k(m)X_k^*(m)\right]$$

where β is a smoothing constant slightly less than 1. (*Note*: In equations such as (31) which use an estimated power value in the denominator, low power in a particular frequency bin may result in division by a very small number, potentially causing numerical instability. To guard against this, a small positive regularization parameter is added to the denominator to ensure numerical stability [11]).

The following observations can be made of the CMD algorithm given by (31), which is shown in Figure 3:

1. If the constraint of (14) is violated, the CMD algorithm will reduce the magnitude of the adaptive filter frequency response in proportion to the level of constraint violation.
2. The CMD algorithm normalizes the weight update in a manner similar to the normalized-LMS with leakage. The amount of leakage is dependent on the level of constraint violation.
3. The CMD algorithm scales the weight update by the inverse of secondary path frequency response, resulting in a faster convergence in regions corresponding to valleys (low magnitude response) in the secondary path.
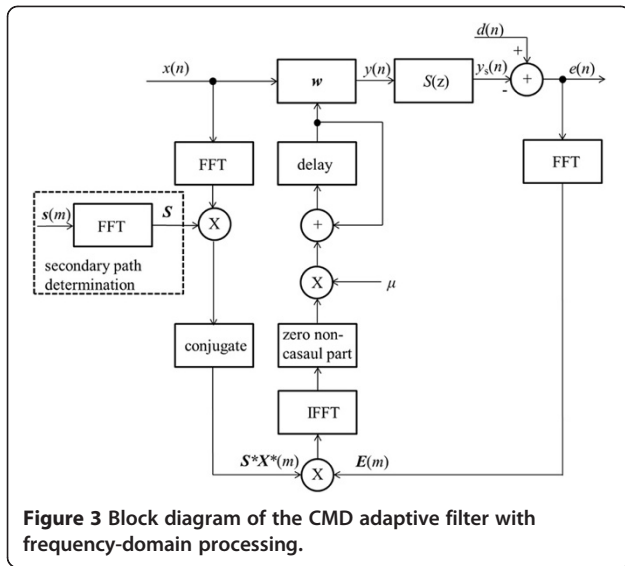
3.2. Power-constrained algorithm
In applications where the filter output power is to be limited, the gain coefficient constraint is replaced by an output power constraint. If total control effort is to be limited [2,6], a single output power constraint can be expressed as

$$P_y(m+1) \leq C$$
(35)

where

$$P_y(m+1) = \frac{1}{N}\sum_{k=0}^{N-1}W_k(m+1)^2 X_k(m)^2.$$
(36)

Figure 3 Block diagram of the CMD adaptive filter with frequency-domain processing.

The new power constrained cost function then becomes

$$J(m+1) = \delta W_k(m+1)^2$$

$$+ \text{Re}\left\{\lambda^*\left[D_k(m) - S_k(m)W_k(m+1)X_k(m)\right]\right\}$$

$$+ \alpha_{\max,k}\left[\left[P_y(m+1) - C\right]^2\right]_z. \tag{37}$$

Following the development of the gain-constrained algorithm, this cost function is differentiated with respect to each of the three variables and set to 0. The resulting equations are

$$\frac{\partial J(m+1)}{\partial W_k(m+1)} = 2[W(m+1) - W_k(m)]^* - \lambda^* S_k(m)X_k(m)$$

$$+ 2\alpha_k W^*(m+1)X_k(m)^2\left[P_y(m+1) - C\right]_z = 0 \tag{38}$$

$$\frac{\partial J(m+1)}{\partial \lambda^*} = D_k(m) - S_k(m)W_k(m+1)X_k(m) = 0 \quad (39)$$

$$\frac{\partial J(m+1)}{\partial \alpha_{\max,k}} = \left[\left[P_y(m+1) - C\right]^2\right]_z = 0. \tag{40}$$

Rearranging (38) yields

$$\left(2\alpha_{\max,k}X_k(m)^2\left[P_y(m+1) - C\right]_z\right)W_k^*(m+1)$$

$$+ W_k^*(m+1)$$

$$= W_k^*(m) + \frac{1}{2}\lambda^* S_k(m)X_k(m). \tag{41}$$

Using the same procedure previously described after (19), the term in (20) is replaced by

$$\alpha_k = 2\alpha_{\max,k}X_k(m)^2\left[P_{y,k}(m) - C_k\right]_z. \tag{42}$$

Following a development similar to the gain-constrained case results in the CMD algorithm given by (31) using a new diagonal matrix of leakage factors (32).

Better frequency performance can be achieved by estimating the power in each frequency bin, making the algorithm more selective in attenuating those frequencies in violation of the constraint. The output power in each frequency bin is determined by

$$P_{y,k}(m+1) = W_k(m+1)^2 P_{x,k}(m+1). \tag{43}$$

Using C_k as the power constraint, it is required that

$$P_{y,k}(m+1) \leq C_k. \tag{44}$$

The resulting cost function is given by

$$J(m+1) = \delta W_k(m+1)^2$$

$$+ \text{Re}\{\lambda^*[D_k(m) - S_k(m)W_k(m+1)X_k(m)]\}$$

$$+ \alpha_{\max,k}\left[\left[P_{y,k}(m+1) - C_k\right]^2\right]_z. \tag{45}$$

Following the development of the gain-constrained algorithm, this cost function is differentiated with respect to each of the three variables and set to 0. The resulting equations are

$$\frac{\partial J(m+1)}{\partial W_k(m+1)} = 2[W_k(m+1) - W_k(m)]^* - \lambda^* S_k(m)X_k(m)$$

$$+ 2\alpha_k W_k^*(m+1)X_k(m)^2\left[P_{y,k}(m+1) - C_k\right]_z = 0 \tag{46}$$

$$\frac{\partial J(m+1)}{\partial \lambda^*} = D_k(m) - S_k(m)W_k(m+1)X_k(m) = 0 \quad (47)$$

$$\frac{\partial J(m+1)}{\partial \alpha_{\max,k}} = \left[\left[P_{y,k}(m+1) - C_k\right]^2\right]_z = 0. \tag{48}$$

Rearranging (46) yields

$$\left(2\alpha_{\max,k}X_k(m)^2\left[P_{y,k}(m+1) - C_k\right]_z\right)W_k^*(m+1)$$

$$+ W_k^*(m+1)$$

$$= W_k^*(m) + \frac{1}{2}\lambda^* S_k(m)X_k(m). \tag{49}$$

Using the same procedure previously described after (19), the term in (20) is replaced by

$$\alpha_k = 2\alpha_{\mathrm{max},k} X_k(m)^2 \left[P_{y,k}(m) - C_k \right]_z.\tag{50}$$

Following a development similar to the gain-constrained case, and using a new diagonal matrix of leakage factors (32) results in the CMD algorithm, repeated below.

$$\mathbf{w}(m+1) = \mathbf{w}(m)$$
$$+ \mu \mathrm{IFFT} \left\{ \frac{\mathbf{S}^*(m)\mathbf{X}^*(m)\mathbf{E}(m)}{\mathbf{S}(m)^2 \mathbf{X}(m)^2} - \mathbf{\Gamma}(m)\mathbf{W}(m) \right\}_+ ,\tag{51}$$

where

$$\mathbf{\Gamma}(m) = \mathrm{diag}\left[\gamma_0(m), \gamma_1(m), \ldots, \gamma_{2N-1}(m) \right].\tag{52}$$

4. Convergence analysis

We assume that all signals are white, zero-mean, Gaussian wide-sense stationary, and employ the independence assumption [7] under a steady-state condition, where the constraint violation is constant and the transform-domain weights are mutually uncorrelated (which occurs as the filter size N grows large [16]). We will also use a normalized input power of unity in (34), which then allows the analysis to apply to both gain-constrained and power-constrained cases. Uncorrelated white measurement noise with a variance of σ_n^2 will be denoted by η_k.

4.1. Mean value

The weight update equation (30) can be written as

$$W_k(m+1) = \left[1 - \mu\gamma_k \right] W_k(m)$$
$$+ \mu_k [D_k(m) - S_k(m)W_k(m)X_k(m)] S_k^*(m)X_k^*(m)\tag{53}$$

or equivalently

$$W_k(m+1) = \left[1 - \mu(\gamma_k + 1) \right] W_k(m) + \mu W_{k,\mathrm{opt}},\tag{54}$$

where $W_{k,\mathrm{opt}}$ denotes the optimal Wiener solution [given as $S_k^{-1}(m)D_k(m)$]. Taking expectations of both sides, using the assumptions, and noting that the input power is normalized per (29) results in

$$E[W_k(m+1)] = \left[1 - \mu(\gamma_k + 1) \right] E[W_k(m)] + \mu W_{k,\mathrm{opt}}\tag{55}$$

By induction, this recursion can be written as

$$E[W_k(m)] = \left[1 - \mu(\gamma_k + 1) \right]^m E[W_k(0)]\tag{56}$$
$$+ \mu W_{k,\mathrm{opt}} \sum_{i=0}^{m-1} \left[1 - \mu(\gamma_k + 1) \right]^{m-1-i}$$

Convergence requirements on μ are given below. When these conditions are satisfied the result is

$$\lim_{m \to \infty} E[W_k(m)] = \mu W_{k,\mathrm{opt}} \lim_{m \to \infty} \sum_{i=0}^{m-1} \left[1 - \mu(\gamma_k + 1) \right]^{m-1-i},\tag{57}$$

which converges in the limit to the steady-state solution $W_{k,\mathrm{ss}}$.

$$W_{k,\mathrm{ss}} = \frac{W_{k,\mathrm{opt}}}{1 + \gamma_k}\tag{58}$$

4.2. Convergence in the mean

The deviation from the steady-state solution in bin k is defined by a weight error [17] given by

$$V_k(m) = W_k(m) - W_{k,\mathrm{ss}}\tag{59}$$

allowing the CMD algorithm to be expressed as

$$V_k(m+1) = \left[1 - \mu\gamma_k - \mu_k X_k^*(m)X_k(m) \right] V_k(m)\tag{60}$$
$$+ \mu_k \eta_k X_k(m) - \mu\gamma_k W_{k,\mathrm{ss}}$$

Taking expectations of both sides results in

$$E[V_k(m+1)] = \left[1 - \mu(\gamma_k + 1) \right] E[V_k(m)] - \mu\gamma_k W_{k,\mathrm{ss}}\tag{61}$$

By induction, this recursion can be written as

$$E[V_k(m)] = \left[1 - \mu(\gamma_k + 1) \right]^m E[V_k(0)]$$
$$- \mu\gamma_k W_{k,\mathrm{ss}} \sum_{i=0}^{m-1} \left[1 - \mu(\gamma_k + 1) \right]^{m-1-i}\tag{62}$$

For this to converge requires the exponential term to decay

$$\left| 1 - \mu(\gamma_k + 1) \right| < 1\tag{63}$$

resulting in

$$\mu < \frac{2}{1 + \gamma_k}\tag{64}$$

with the upper bound on μ occurring for maximum constraint violation, given by

$$\mu < \frac{2}{1 + \gamma_{k,\mathrm{max}}}\tag{65}$$

4.3. Convergence in the mean square

Both sides of (60) are first post-multiplied by their respective conjugate transposes, rearranged, and after taking expectations the result is

$$E\left[V_k(m+1)V_k^*(m+1)\right]$$

$$= E\left[(1-\mu\gamma_k-\mu_k)^2 V_k(m)V_k^*(m)\right] + \frac{\mu^2\sigma_n^2}{X_k(m)^2} + \mu^2\gamma_k^2 W_{k,\mathrm{ss}}W_{k,\mathrm{ss}}^*$$

$$-\mu\gamma_k E\left[(1-\mu\gamma_k-\mu_k)V_k(m)W_{k,\mathrm{ss}}^*\right] - \mu\gamma_k E\left[(1-\mu\gamma_k-\mu_k)V_k^*(m)W_{k,\mathrm{ss}}\right]$$

(66)

Rearranging and employing the assumptions [18] gives

$$E\left[V_k(m+1)V_k^*(m+1)\right]$$

$$= \left[(1-2\mu(\gamma_k+1)+\mu^2(\gamma_k^2+2\gamma_k+1))\right]$$

$$E\left[V_k(m)V_k^*(m)\right] + \frac{\mu^2\sigma_n^2}{X_k(m)^2} + \mu^2\gamma_k^2 W_{k,\mathrm{ss}}(m)^2$$

$$- 2\mu\gamma_k\left[1-\mu(\gamma_k+1)\right]W_{k,\mathrm{ss}}^* E[V_k(m)].$$

(67)

As the weight error variance update depends on the mean coefficient error vector, $V_k(m)$, a state-space model can be defined as

$$\mathbf{Z}_k(m) = \begin{bmatrix} E\left[V_k(m)V_k^*(m)\right] \\ E\left[V_k(m)\right] \end{bmatrix}$$

(68)

and the update defined as the real component of

$$\mathbf{Z}_k(m+1) = \mathbf{A}\mathbf{Z}_k(m) + \mathbf{B}$$

(69)

with

$$\mathbf{A} = \begin{bmatrix} A_{11} & A_{12} \\ 0 & A_{22} \end{bmatrix}$$

(70)

and

$$\mathbf{B} = \begin{bmatrix} B_1 \\ B_2 \end{bmatrix}$$

(71)

where

$$A_{11} = 1 - 2\mu(\gamma_k+1) + \mu^2(\gamma_k^2+2\gamma_k+1)$$

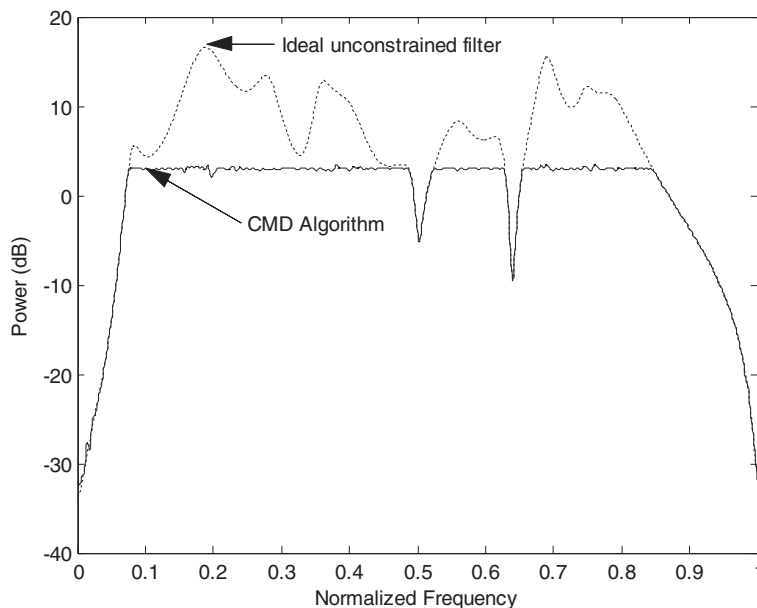$$A_{12} = -2\mu\gamma_k\left[1-\mu(\gamma_k+1)\right]W_{k,\mathrm{ss}}^*$$

$$A_{22} = 1 - \mu(\gamma_k+1)$$

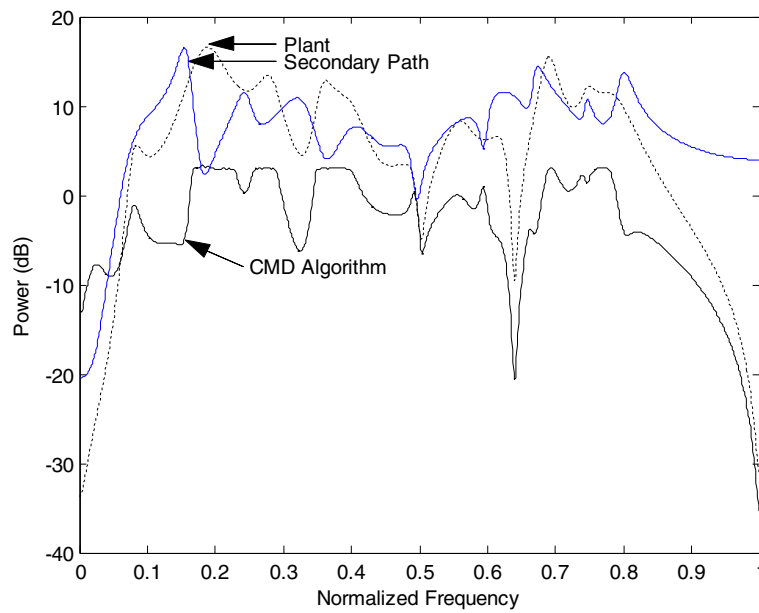$$B_1 = \mu^2\left(\frac{\sigma_n^2}{X_k(m)^2} + \gamma_k^2 W_{k,\mathrm{ss}}(m)^2\right)$$

$$B_2 = -\mu\gamma_k W_{k,\mathrm{ss}}.$$

(72)

For stability, it is required that the eigenvalues in the state transition matrix **A** have a magnitude less than 1

Figure 4 Frequency response of gain-constrained CMD algorithm.

Figure 5 Frequency response of gain-constrained CMD algorithm with secondary path.

[19], requiring matrix entry A_{11} in (70) to be bounded to magnitude less than 1, resulting in

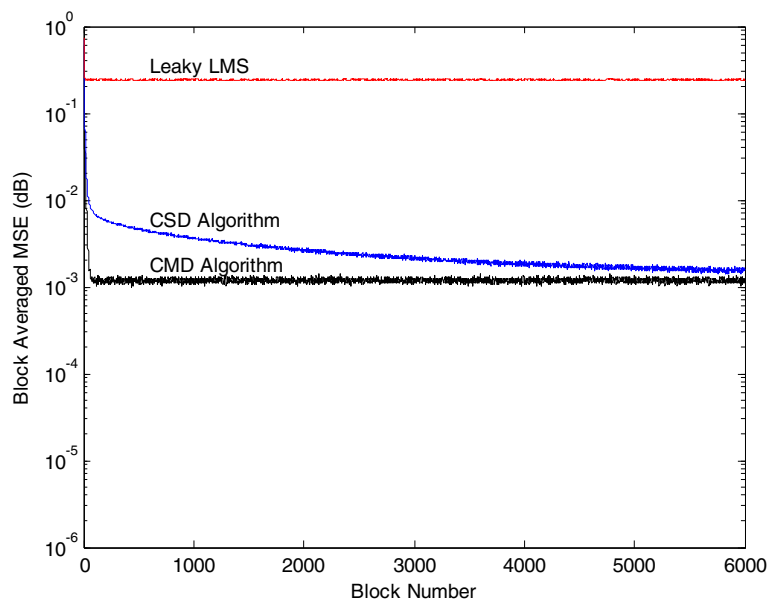$$\left| 1 - 2\mu(\gamma_k + 1) + \mu^2(\gamma_k^2 + 2\gamma_k + 1) \right| < 1 \qquad (73)$$

or

$$\mu < \frac{2}{1 + \gamma_k} \qquad (74)$$

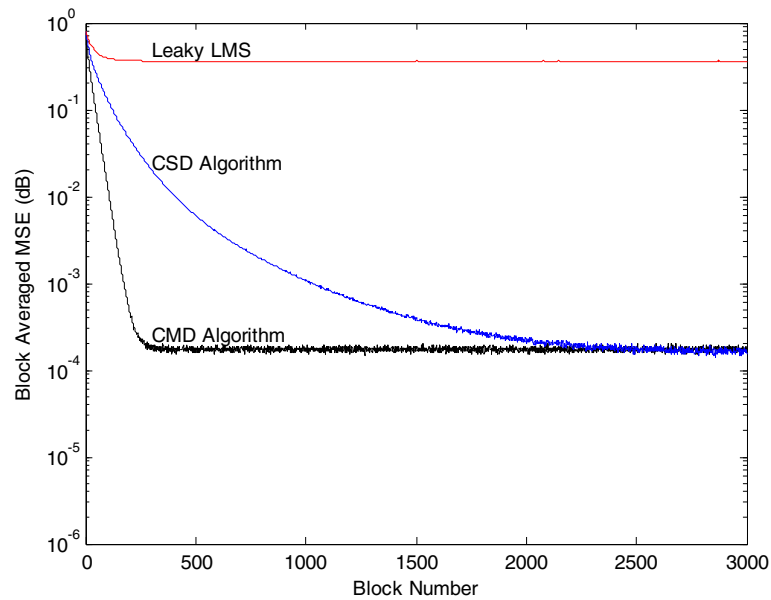with the upper bound on μ occurring for maximum constraint violation, given by

$$\mu < \frac{2}{1 + \gamma_{k,\max}} \qquad (75)$$

5. Simulations

In the simulations, the experimental data from [3] is used for the plant, modeled by a 512-term all-zero filter centered at $N/2$. The output rescaling algorithm (2) is

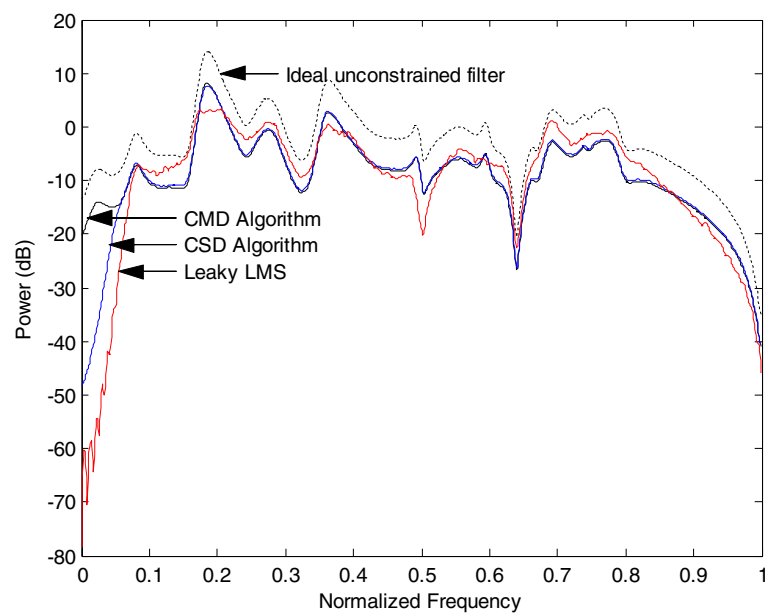**Figure 6 Convergence comparison, gain-constrained condition.**

Figure 7 Convergence comparison, gain-constrained condition, AR(1) colored noise input.

applied in the frequency-domain to determine the steady-state adaptive filter final coefficients. We demonstrate the improved convergence performance of the CMD algorithm as compared to the CSD algorithm and the leaky LMS in both gain-constrained and power-constrained applications. The values of constraint terms C, $\alpha_{\max k}$, and C_k are held constant in the simulations, but could be shaped over frequency for specific applications. External uncorrelated Gaussian white noise with a variance of 0.01
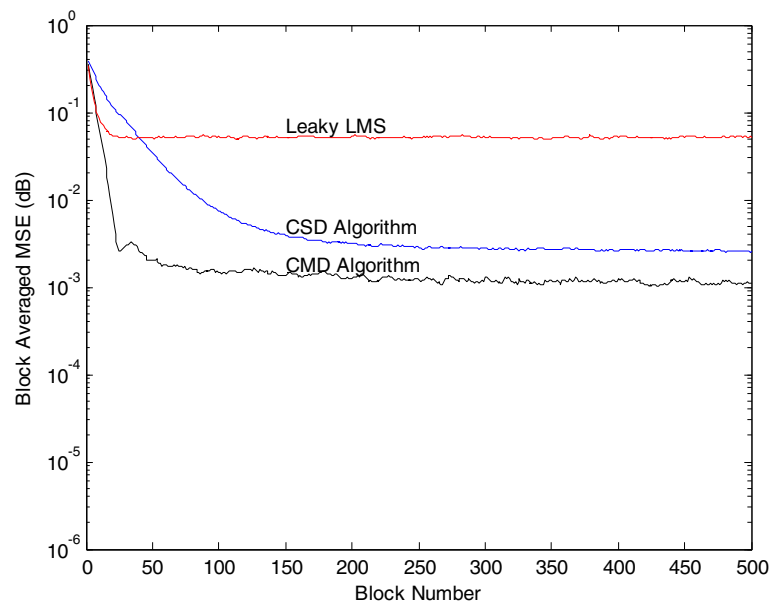
is added for the convergence comparisons, and an average of 100 runs is plotted. In the simulations, we are assuming prior knowledge of the secondary path transfer function; methods for on-line and off-line secondary path identification are presented in, e.g., [20,21].

5.1. Gain-constrained algorithm
Using a unity gain secondary path, a 3-dB coefficient gain constraint is imposed, and Figure 4 shows the plant

**Figure 8 Frequency response comparison, power-constrained condition.**

Figure 9 Convergence comparison, power-constrained condition.

frequency response and the response of the new CMD algorithm, illustrating the clipping effect of the algorithm.

Using the experimental data from [3] for the secondary path, the algorithms should converge to the filter in Figure 5, which shows the CMD algorithm response, the plant frequency response, and the secondary path frequency response. The adaptive filter in this case will need to simultaneously identify $H(z)$ and equalize $S(z)$, while still maintaining the gain constraints. The

convergence comparison for the three algorithms for the system in Figure 5 for a white noise input is displayed in Figure 6. The CMD algorithm has the fastest convergence performance. The CSD algorithm began converging in a similar manner, but was not able to fully achieve the relatively high 20 dB gain required at the lowest frequencies in Figure 5. However, other simulations without deep secondary path nulls showed that the two algorithms converge to similar final weight values, with the CMD having
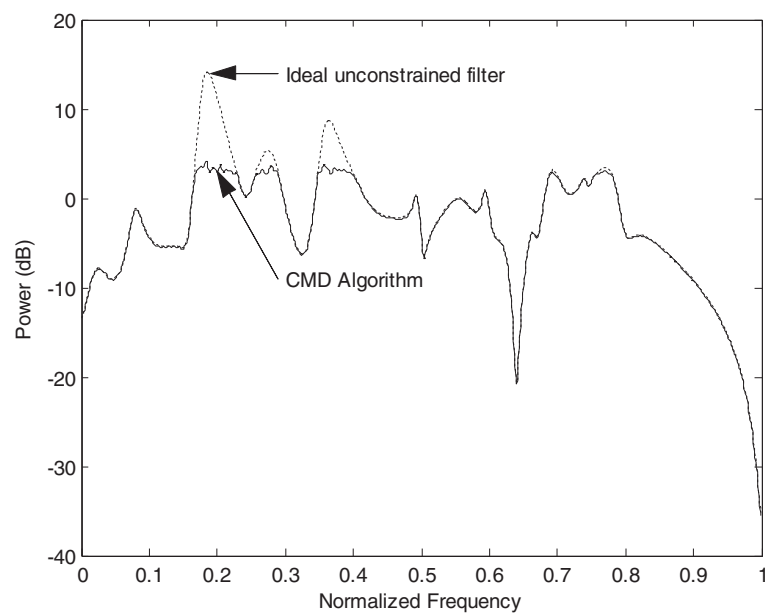
Figure 10 Frequency response, bin-power-constrained condition.

a faster convergence rate. The leaky LMS attenuates all frequencies (and not just those in violation of the constraint) and has the poorest convergence performance. (The leaky LMS appears smoother than the other two algorithms, but this is due to the logarithmic scale of the y-axis in the plots.)

Figure 7 compares the convergence of the three algorithms for colored noise input, created by filtering the input with a first order AR(1) low pass filter process with coefficients [1−0.95]. The CSD algorithm requires a significant reduction of μ in (8) to maintain stability, resulting in a slow response. However, the increased energy in the lower frequency regions due to the low pass input process improved the misadjustment for this case. The leaky LMS attenuates all frequencies (and not just those in violation of the constraint) and has the poorest convergence performance and highest excess misadjustment.

5.2. Power-constrained algorithm

The frequency response and convergence of the three algorithms is compared in Figures 8 and 9, respectively, using a single output power constraint of 25% of the unconstrained value (−6 dB). The CMD algorithm has the fastest convergence performance and maintains a 6-dB power reduction over frequency. The CSD displays similar performance, but again was not able to fully achieve the relatively high 20 dB gain required at the lowest frequencies. The leaky LMS has the poorest convergence performance, primarily due to its inability to track the lowest frequencies. Both the CMD and CSD algorithms allow the power constraint to be set explicitly, while the leaky LMS requires a trial and error approach to determine the parameters.

The CMD algorithm frequency response for the individual bin-constrained case using the constraint of (44) is shown in Figure 10 for a 3-dB power limit with a wideband white noise input. Comparing this to Figure 8 illustrates how the new CMD algorithm reduces the output in the frequencies of power-constraint violation, while minimizing the effect at other frequencies.

6. Conclusion

A new algorithm was presented, the CMD LMS, for gain-constrained and power-constrained adaptive filter applications. Analysis results were developed for the stability bounds in the mean and mean-square sense. The CMD algorithm was compared to the algorithm developed in [2] and the leaky LMS for filtered-X ANC applications. The new CMD algorithm provides faster convergence and improved frequency response performance, especially in colored noise environments. Additionally, the new CMD algorithm has the ability to handle multiple constraints in both gain-constrained and power-constrained applications.

References
1. PA Nelson, SJ Elliott, *Active Control of Sound* (Academic Press, London, 1992)
2. B Rafaely, S Elliot, A computationally efficient frequency-domain LMS algorithm with constraints on the adaptive filter. IEEE Trans. Signal Process. **48**(6), 1649–1655 (2000)
3. SM Kuo, DR Morgan, *Active Noise Control Systems: Algorithms and DSP Implementations* (Wiley, New York, 1996)
4. F Taringoo, J Poshtan, MH Kahaei, Analysis of effort constraint algorithm in active noise control systems. EURASIP J. Appl. Signal Process. **2006**, 1–9 (2006)
5. X Qiu, CH Hansen, A study of time-domain FXLMS algorithms with control output constraint. J. Acoust. Soc. Am. **1097**(6), 2815–2823 (2001)
6. P Darlington, Performance surfaces of minimum effort estimators and controllers. IEEE Trans. Signal Process. **43**(2), 536–539 (1995)
7. B Widrow, *SD Stearns: Adaptive Signal Processing* (Prentice-Hall, Upper Saddle River, NJ, 1985)
8. MP Nowak, BD Van Veen, A constrained transform-domain adaptive IIR filter structure for active noise control. IEEE Trans. Speech Audio Process. **5**(5), 334–347 (1997)
9. DR Morgan, JC Thi, A delayless subband adaptive filter architecture. IEEE Trans. Signal Process. **43**(8), 1819–1830 (1995)
10. JJ Shynk, Frequency-domain and multirate adaptive filtering. IEEE Signal Process. Mag. **9**, 337–339 (1993)
11. S Haykin, *Adaptive Filter Theory* (Prentice-Hall, Upper Saddle River, NJ, 2002)
12. R Fletcher, *Practical Methods of Optimization* (Wiley, New York, 1987)
13. WJ Kozacky, T Ogunfunmi, Convergence analysis of a frequency-domain adaptive filter with constraints on the output weights, in *Proceedings of the Asilomar Conference on Signals, Systems, and Computers* (Pacific Grove, USA, 2009), pp. 1350–1355
14. SJ Elliott, KH Beck, Effort constraints in adaptive feedforward control. IEEE Signal Process. Lett. **3**(1), 7–9 (1996)
15. PCW Sommen, PJ Van Gerwen, HJ Kotmans, JEM Janssen, Convergence analysis of a frequency-domain adaptive filter with exponential power averaging and generalized window function. IEEE Trans. Circuits Syst. **34**(7), 788–798 (1987)
16. B Farhang-Boroujeny, KS Chan, Analysis of the frequency-domain block LMS algorithm. IEEE Trans. Signal Process. **48**(8), 2332–2342 (2000)
17. K Mayyas, T Aboulnast, Leaky LMS algorithm: MSE analysis for Gaussian data. IEEE Trans. Signal Process. **45**(4), 927–934 (1997)
18. SC Douglas, Performance comparison of two implementations of the leaky LMS adaptive filter. IEEE Trans. Signal Process. **45**(8), 2125–2129 (1997)
19. K Mayyas, T Aboulnasr, Leaky LMS: a detailed analysis, in *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS)*. Seattle, USA **2**, 1255–1258 (1995)
20. SM Kuo, D Vijayan, A secondary path modeling technique for active noise control systems. IEEE Trans. Speech Audio Process. **5**(4), 374–377 (1997)
21. MT Akhtar, M Abe, M Kawamata, On active noise control systems with online acoustic feedback path modeling. IEEE Trans. Audio Speech Lang. Process. **15**(2), 593–600 (2007)