This research has been supported in part by European Commission
FP6 IYTE-Wireless Project (Contract No: 017442)

Conjugate-gradient-based decision feedback equalization with structured channel estimation for digital television

Michael D. Zoltowski[a], William J. Hillery[a], Serdar Özen[a], and Mark Fimoff[b]

[a]Purdue University, School of Elect. and Comp. Eng., West Lafayette, IN 47907-1285
[b]Zenith Electronics Corp., 2000 Millbrook Drive, Lincolnshire, IL 60069

ABSTRACT

In this paper, we show how the convergence time of equalizers for 8-VSB based on the conjugate gradient (CG) algorithm can be considerably improved through initialization based on a channel estimate. We derive real and complex minimum mean-square error (MMSE) equalizers and implement them adaptively using the conjugate gradient, recursive least squares (RLS), and least mean squares (LMS) algorithms. We show that both CG and RLS have similar convergence times — both are much faster than LMS. Since the CG algorithm is easily initialized, we compare several methods of initialization to determine how each affects convergence and then apply the best methods to initialize equalizers using channel estimates. We find that initializing the correlation matrices and filling the feedback taps with training symbols greatly speeds convergence of the CG adaptive equalizer, potentially approaching the rate of convergence when running the algorithm on the matrix equations using the actual channel.

Keywords: Adaptive filtering, conjugate gradient method, decision feedback equalization (DFE), digital television, vestigial sideband (VSB)

1. INTRODUCTION

The terrestrial broadcast standard for digital television in the United States is based on 8-level vestigial sideband modulation (8-VSB).[1] According to the standard, a field sync segment is transmitted every 24.2ms containing 704 known symbols — four segment sync symbols, a length 511 pseudo-noise (PN) sequence, and three length 63 PN sequences. These symbols may be used to train an adaptive equalizer. Since the broadcast environment is characterized by long channel impulse responses,[2,3] long equalizers are required, which makes it more difficult for an adaptive equalizer to converge within the available training period.

Previous work on equalization for 8-VSB has focused on blind and semi-blind techniques.[3–6] However, when the training sequence is used (as in a semi-blind technique), a correlation must be performed to find the start of the training sequence. This correlation operation may be used to estimate the multipath channel and the channel estimate may then be used to aid the convergence of a training-based adaptive equalizer. More sophisticated techniques may also be applied to improve the channel estimate, as demonstrated by Özen et al.[7]

Recent work on reduced-rank filtering[8] has linked the multi-stage nested Wiener filter to the conjugate gradient (CG) algorithm. Chang and Willson[9] have analyzed adaptive implementations of the CG algorithm. Chowdhury and Zoltowski[10] have successfully applied the CG algorithm to DS-CDMA feedforward equalizers and found that it converges at about the same rate as the recursive least squares (RLS) algorithm with approximately the same computational complexity. In this paper, we show how the convergence time of decision feedback equalizers (DFE's) for 8-VSB based on the CG algorithm can be considerably improved through initialization based on a channel estimate. We begin by describing the 8-VSB system model and deriving expressions for real and complex minimum mean-square error (MMSE) equalizers. These equalizers are then implemented adaptively using the CG, RLS, and least mean squares (LMS) algorithms to compare their uninitialized convergence times. We then compare several methods of initialization to determine how each affects convergence and then apply the best methods to initialize equalizers using channel estimates. Finally, we present our conclusions.

Further author information: (Send correspondence to: M.D.Z.)
M.D.Z.: E-mail: mikedz@ecn.purdue.edu, Telephone: 765-494-3512, Fax: 765-494-0880
W.J.H.: E-mail: hilleryw@purdue.edu, S.O.: ozen@ecn.purdue.edu, M.F.: E-mail: Mark.Fimoff@zenith.com

2. SYSTEM MODEL

The baseband equivalent discrete-time system model is shown in Fig. 1 where symbol-rate sampling is assumed. The

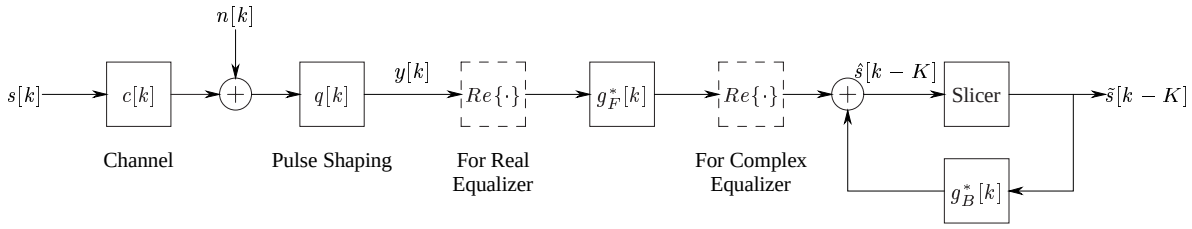

Figure 1: System model.

transmitted symbol sequence is represented by $s[k]$ where, in this study, we assume that the symbols are independent and equally probable. Symbols in the 8-VSB constellation are taken from the set $\{-7,-5,-3,-1,1,3,5,7\}$. The continuous-time channel contains M paths with delays τ_i and complex gains α_i. We assume that the path with maximum gain magnitude occurs at a delay of zero. When convolved with the complex transmitted pulse $q(t)$, the channel $c(t)$ becomes

$$c(t) = \sum_{i=1}^{M} \alpha_i q(t-\tau_i). \tag{1}$$

To match the specification in the ATSC standard,[1] the pulse $q(t)$ is a square-root raised cosine pulse designed for one-half the symbol rate whose spectrum has been shifted up in frequency by one-fourth the symbol rate. In other words, if $p_{RRC}(t)$ is a square-root raised cosine pulse with zero crossings at non-zero multiples of $2T_s$, where T_s is the symbol period, then

$$q(t) = e^{j\pi\frac{F_s}{2}t} p_{RRC}(t), \tag{2}$$

where $F_s = \frac{1}{T_s}$. The discrete-time channel is found by sampling $c(t)$ at the symbol rate: $c[k] = c(kT_s)$. The noise sequence $n[k]$ is complex additive white Gaussian noise with a power spectral density of N_0. The receiver filter $q[k]$ is matched to the transmitter pulse shaping filter so that $q[k] = q^*(-kT_s) = q(kT_s)$, where the asterisk superscript (*) denotes complex conjugation. The composite channel at the input to the equalizer is defined to be $h[k] = c[k] * q[k]$ (here, the large asterisk denotes convolution) and therefore

$$y[k] = \sum_m s[k-m]h[m] + \nu[k], \tag{3}$$

where $\nu[k] = n[k] * q[k]$ is the noise at the input to the equalizer.

3. EQUALIZER DERIVATIONS

As shown in Fig. 1, $y[k]$ may be equalized using either a complex or real feedforward section $g_F[k]$. For the complex equalizer, the real part of the signal is taken after the feedforward section. The real part of the signal is taken before the feedforward section for the real equalizer. The composite channel $h[k]$ is assumed to have length $L_h + 1$ with L_{ha} taps prior to the peak of the maximum magnitude path and L_{hc} taps afterward; thus, $L_h = L_{ha} + L_{hc}$. We represent the channel by a vector

$$\boldsymbol{h} = [\, h[-L_{ha}], \ldots, h[-1], h[0], h[1], \ldots, h[L_{hc}]\,]^T. \tag{4}$$

We also use a finite length representation of the receiver matched filter $q[k]$ with length $2L_q + 1$:

$$\boldsymbol{q} = [\, q[-L_q], \ldots, q[-1], q[0], q[1], \ldots, q[L_q]\,]^T. \tag{5}$$

We begin with the derivation of the complex equalizer. The equalizer is assumed to have $N_F + 1$ feedforward taps and N_B feedback taps. Assembling $N_F + 1$ consecutive samples of $y[k]$ into a vector $\boldsymbol{y}$ yields

$$\boldsymbol{y}[k] = \boldsymbol{H}\,\boldsymbol{s}[k] + \boldsymbol{Q}\boldsymbol{n}[k], \tag{6}$$

where $\boldsymbol{n}[k] = [\, n[k + L_q] \,\ldots\, n[k - L_q - N_F] \,]^T$, $\boldsymbol{s}[k] = [\, s[k + L_{ha}] \,\ldots\, s[k - L_{hc} - N_F] \,]^T$,

$$\boldsymbol{H} = \begin{bmatrix} \boldsymbol{h}^T & 0 & \cdots & 0 \\ 0 & \boldsymbol{h}^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{h}^T \end{bmatrix}, \qquad \text{and} \qquad \boldsymbol{Q} = \begin{bmatrix} \boldsymbol{q}^T & 0 & \cdots & 0 \\ 0 & \boldsymbol{q}^T & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{q}^T \end{bmatrix}. \tag{7}$$

The symbol estimate $\hat{s}[k - K]$ is given by

$$\hat{s}[k - K] \;=\; \mathrm{Re}\{{\boldsymbol{g}_F}^H \boldsymbol{y}[k]\} + {\boldsymbol{g}_B}^T \boldsymbol{s}_B[k - K - 1], \tag{8}$$

where K is the cursor location (defined below), $\boldsymbol{g}_F = [\, g_F[0], \ldots, g_F[N_F] \,]^T$, $\boldsymbol{g}_B = [\, g_B[1], \ldots, g_B[N_B] \,]^T$, $\boldsymbol{s}_B[k] = [\, s[k], \ldots, s[k + 1 - N_B] \,]^T$, and the superscript H denotes conjugate transpose. In this analysis we assume that all decisions are correct. If we adopt the convention that the subscripts R and I indicate the real and imaginary parts of quantities, then

$$\begin{aligned} \hat{s}[k - K] &= {\boldsymbol{g}_{FR}}^T \boldsymbol{y}_R[k] + {\boldsymbol{g}_{FI}}^T \boldsymbol{y}_I[k] + {\boldsymbol{g}_B}^T \boldsymbol{s}_B[k - K - 1] \\ &= {\boldsymbol{g}_{FC}}^T \boldsymbol{y}_C[k] + {\boldsymbol{g}_B}^T \boldsymbol{s}_B[k - K - 1], \end{aligned} \tag{9}$$

where $\boldsymbol{g}_{FC} = \left[\, {\boldsymbol{g}_{FR}}^T \; {\boldsymbol{g}_{FI}}^T \,\right]^T$ and $\boldsymbol{y}_C[k] = \left[\, {\boldsymbol{y}_R}^T[k] \; {\boldsymbol{y}_I}^T[k] \,\right]^T$. We may now write

$$\boldsymbol{y}_C[k] = \boldsymbol{H}_C \boldsymbol{s}[k] + \boldsymbol{Q}_C \boldsymbol{n}_C[k], \tag{10}$$

where $\boldsymbol{H}_C = \left[\, {\boldsymbol{H}_R}^T \; {\boldsymbol{H}_I}^T \,\right]^T$, $\boldsymbol{Q}_C = \begin{bmatrix} \boldsymbol{Q}_R & -\boldsymbol{Q}_I \\ \boldsymbol{Q}_I & \boldsymbol{Q}_R \end{bmatrix}$, and $\boldsymbol{n}_C[k] = \left[\, {\boldsymbol{n}_R}^T[k] \; {\boldsymbol{n}_I}^T[k] \,\right]^T$.

In the decision feedback equalizer, we use the cursor to define exactly which symbol is being estimated. To motivate the cursor definition, we use the real equalizer and consider a channel containing a single path with real gain at delay zero. In this case, $h_R[k]$ is a delta function at $k = 0$ and the sample $y_R[k]$ corresponds to the symbol $s[k]$ (since $y_R[k] = s[k] + \nu_R[k]$). Then the feedforward term in the symbol estimate is

$$\begin{aligned} {\boldsymbol{g}_F}^T \boldsymbol{y}_R[k] &= \sum_{n=0}^{N_F} g_F[n] y_R[k - n] \\ &= \sum_{n=0}^{N_F} g_F[n] \left(s[k - n] + \nu_R[k - n] \right), \end{aligned} \tag{11}$$

where we recall that $\boldsymbol{g}_F$ is real for this discussion. That is, the feedforward term in the equalizer output only depends on the symbols $s[k], s[k-1], \ldots, s[k - N_F]$. When there is multipath interference, the feedforward term will depend on symbols covering a wider time-span, but since we may encounter a channel where the multipath is negligible, we may only consider $s[k], s[k-1], \ldots, s[k - N_F]$ as candidates for the symbol to estimate. Therefore, we estimate the symbol $s[k - K]$, where $0 \le K \le N_F$, and call this symbol the *cursor*. Since there is a one-to-one correspondence between the candidate symbols and the taps in the feedforward section, we often identify the cursor by the corresponding feedforward tap. This definition of the cursor is consistent with standard DFE theory and practice.

We now find the equalizer by minimizing the mean-square error (MSE), where

$$\begin{aligned} \mathrm{MSE} &= \mathrm{E}\{(s[k - K] - \hat{s}[k - K])^2\} \\ &= \mathcal{E}_s - 2{\boldsymbol{g}_{FC}}^T \boldsymbol{r}_{s y_C} + 2{\boldsymbol{g}_{FC}}^T \boldsymbol{R}_{y_C s_B} \boldsymbol{g}_B + {\boldsymbol{g}_{FC}}^T \boldsymbol{R}_{y_C y_C} \boldsymbol{g}_{FC} + \mathcal{E}_s {\boldsymbol{g}_B}^T \boldsymbol{g}_B \end{aligned} \tag{12}$$

with $\mathcal{E}_s = \mathrm{E}\{(s[k])^2\}$, $\boldsymbol{r}_{sy_C} = \mathrm{E}\{s[k-K]\boldsymbol{y}_C[k]\}$, $\boldsymbol{R}_{y_C s_B} = \mathrm{E}\{\boldsymbol{y}_C[k]\boldsymbol{s}_B{}^T[k-K-1]\}$, and $\boldsymbol{R}_{y_C y_C} = \mathrm{E}\{\boldsymbol{y}_C[k]\boldsymbol{y}_C^T[k]\}$. Here we have made use of the expressions $\boldsymbol{R}_{s_B s_B} = \mathrm{E}\{\boldsymbol{s}_B[k]\boldsymbol{s}_B{}^T[k]\} = \mathcal{E}_s\boldsymbol{I}_{N_B}$ and $\boldsymbol{r}_{ss_B} = \mathrm{E}\{s[k-K]\boldsymbol{s}_B[k]\} = \boldsymbol{0}$. Minimization of this expression yields[11]

$$\boldsymbol{g}_{FC} = \left(\boldsymbol{R}_{y_C y_C} - \frac{1}{\mathcal{E}_s}\boldsymbol{R}_{y_C s_B}\boldsymbol{R}_{y_C s_B}{}^T\right)^{-1}\boldsymbol{r}_{sy_C} \tag{13}$$

$$\boldsymbol{g}_B = -\frac{1}{\mathcal{E}_s}\boldsymbol{R}_{y_C s_B}{}^T\boldsymbol{g}_{FC}, \tag{14}$$

where

$$\boldsymbol{r}_{sy_C} = \mathcal{E}_s\boldsymbol{H}_C\boldsymbol{\delta}_{K+L_{ha}} \tag{15}$$

$$\boldsymbol{R}_{y_C y_C} = \mathcal{E}_s\boldsymbol{H}_C\boldsymbol{H}_C{}^T + N_0\boldsymbol{Q}_C\boldsymbol{Q}_C{}^T \tag{16}$$

$$\boldsymbol{R}_{y_C s_B} = \mathcal{E}_s\boldsymbol{H}_C\boldsymbol{\Delta}_K. \tag{17}$$

The vector $\boldsymbol{\delta}_{K+L_{ha}}$ contains all zeros except for a one in element $K+L_{ha}+1$. The matrix $\boldsymbol{\Delta}_K$ is defined by

$$\boldsymbol{\Delta}_K = \begin{bmatrix} \boldsymbol{0}_{(L_{ha}+K+1)\times N_B} \\ \boldsymbol{I}_{N_B} \\ \boldsymbol{0}_{(N_F+L_{hc}-K-N_B)\times N_B} \end{bmatrix} \tag{18}$$

when $0 \le K \le L_{hc}+N_F-N_B$ and

$$\boldsymbol{\Delta}_K = \begin{bmatrix} \boldsymbol{0}_{(L_{ha}+K+1)\times(N_F+L_{hc}-K)} & \boldsymbol{0}_{(L_{ha}+K+1)\times(K+N_B-N_F-L_{hc})} \\ \boldsymbol{I}_{N_F+L_{hc}-K} & \boldsymbol{0}_{(N_F+L_{hc}-K)\times(K+N_B-N_F-L_{hc})} \end{bmatrix} \tag{19}$$

when $L_{hc}+N_F-N_B < K \le L_{hc}+N_F-1$. Here, $\boldsymbol{0}_{m\times n}$ is a zero matrix of size $m\times n$ and $\boldsymbol{I}_n$ is the identity matrix of size $n\times n$. An alternative expression for the complex equalizer which we will find useful is

$$\begin{bmatrix} \boldsymbol{R}_{y_C y_C} & \boldsymbol{R}_{y_C s_B} \\ \boldsymbol{R}_{y_C s_B}{}^T & \boldsymbol{R}_{s_B s_B} \end{bmatrix}\begin{bmatrix} \boldsymbol{g}_{FC} \\ \boldsymbol{g}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{sy_C} \\ \boldsymbol{r}_{ss_B} \end{bmatrix}, \tag{20}$$

where $\boldsymbol{R}_{s_B s_B} = \mathcal{E}_s\boldsymbol{I}_{N_B}$ and $\boldsymbol{r}_{ss_B} = \boldsymbol{0}$.

The derivation of the real equalizer is similar with the result that

$$\boldsymbol{g}_F = \left(\boldsymbol{R}_{y_R y_R} - \frac{1}{\mathcal{E}_s}\boldsymbol{R}_{y_R s_B}\boldsymbol{R}_{y_R s_B}{}^T\right)^{-1}\boldsymbol{r}_{sy_R} \tag{21}$$

$$\boldsymbol{g}_B = -\frac{1}{\mathcal{E}_s}\boldsymbol{R}_{y_R s_B}{}^T\boldsymbol{g}_{FR}, \tag{22}$$

where

$$\boldsymbol{r}_{sy_R} = \mathcal{E}_s\boldsymbol{H}_R\boldsymbol{\delta}_{K+L_{ha}} \tag{23}$$

$$\begin{aligned} \boldsymbol{R}_{y_R y_R} &= \mathcal{E}_s\boldsymbol{H}_R\boldsymbol{H}_R{}^T + N_0\left(\boldsymbol{Q}_R\boldsymbol{Q}_R{}^T + \boldsymbol{Q}_I\boldsymbol{Q}_I{}^T\right) \\ &= \mathcal{E}_s\boldsymbol{H}_R\boldsymbol{H}_R{}^T + N_0\boldsymbol{I}_{N_F+1} \end{aligned} \tag{24}$$

$$\boldsymbol{R}_{y_R s_B} = \mathcal{E}_s\boldsymbol{H}_R\boldsymbol{\Delta}_K. \tag{25}$$

The alternative expression for the real equalizer is

$$\begin{bmatrix} \boldsymbol{R}_{y_R y_R} & \boldsymbol{R}_{y_R s_B} \\ \boldsymbol{R}_{y_R s_B}{}^T & \boldsymbol{R}_{s_B s_B} \end{bmatrix}\begin{bmatrix} \boldsymbol{g}_F \\ \boldsymbol{g}_B \end{bmatrix} = \begin{bmatrix} \boldsymbol{r}_{sy_R} \\ \boldsymbol{r}_{ss_B} \end{bmatrix}. \tag{26}$$

4. ADAPTIVE ALGORITHMS

4.1. Conjugate Gradient Algorithm

Since the conjugate gradient algorithm solves equations of the form $\boldsymbol{R}\boldsymbol{g} = \boldsymbol{r}$, the algorithm requires estimates of $\boldsymbol{R}$ and $\boldsymbol{r}$. In this study, we estimate the components of $\boldsymbol{R}$ and $\boldsymbol{r}$ in Eqs. 20 and 26 using a running average. That is, to estimate $\boldsymbol{R}_{uv} = \mathrm{E}\{\boldsymbol{u}[n]\boldsymbol{v}^T[n]\}$ at time k, we use the expression

$$\boldsymbol{R}_{uv}[k] = \frac{1}{k+1}\left(k\boldsymbol{R}_{uv}[k-1] + \boldsymbol{u}[k]\boldsymbol{v}^T[k]\right). \tag{27}$$

This estimate has infinite memory and is only appropriate in a situation where the channel is static.

The adaptive conjugate gradient algorithm used here is identical to that described by Chang and Willson[9] with two exceptions. The first is our use of infinite memory matrix averaging as described above. The second difference is our use of the Hestenes-Stiefel formula for the calculation of the conjugate direction step size β_k.[12] The adaptive CG algorithm used is listed in Algorithm 1 where the conjugate direction vector $\boldsymbol{b}[k]$ is reset to the negative of the estimated

Algorithm 1 Adaptive Conjugate Gradient Algorithm

$\boldsymbol{R}_{y_C y_C}[0] = \boldsymbol{R}_{y_C y_C 0}$, $\boldsymbol{R}_{y_C s_B}[0] = \boldsymbol{R}_{y_C s_B 0}$, $\boldsymbol{r}_{s y_C}[0] = \boldsymbol{r}_{s y_C 0}$,
$\boldsymbol{R}_{s_B s_B}[0] = \mathcal{E}_s \boldsymbol{I}_{N_B}$, $\boldsymbol{r}_{s s_B}[0] = \boldsymbol{0}$
$\boldsymbol{g}[0] = \boldsymbol{g}_0$, $\boldsymbol{t}[0] = \boldsymbol{0}$
for $k = 1, \ldots$ **do**
  Update $\boldsymbol{R}_{y_C y_C}[k]$, $\boldsymbol{R}_{y_C s_B}[k]$, $\boldsymbol{R}_{s_B s_B}[k]$, $\boldsymbol{r}_{s y_C}[0]$, and $\boldsymbol{r}_{s s_B}[k]$
  $\boldsymbol{t}[k] = \boldsymbol{R}[k]\boldsymbol{g}[k-1] - \boldsymbol{r}[k]$
  if $k = 1$ or $k \bmod k_0 = 1$ **then**
    $\boldsymbol{b}[k] = -\boldsymbol{t}[k]$
  else
    $\beta_k = \dfrac{(\boldsymbol{t}[k] - \boldsymbol{t}[k-1])^T \boldsymbol{t}[k]}{(\boldsymbol{t}[k] - \boldsymbol{t}[k-1])^T \boldsymbol{p}[k-1]}$
    $\boldsymbol{b}[k] = -\boldsymbol{t}[k] + \beta_k \boldsymbol{b}[k-1]$
  end if
  $\alpha_k = \dfrac{\boldsymbol{b}^T[k]\boldsymbol{t}[k]}{\boldsymbol{b}^T[k]\boldsymbol{R}[k]\boldsymbol{b}[k]}$
  $\boldsymbol{g}[k] = \boldsymbol{g}[k-1] + \alpha_k \boldsymbol{b}[k]$
end for

gradient $\boldsymbol{t}[k]$ after every k_0 steps. Algorithm 1 has been written for the complex equalizer, where $\boldsymbol{g} = \left[\boldsymbol{g}_{FC}{}^T \boldsymbol{g}_B{}^T\right]^T$, $\boldsymbol{r} = \left[\boldsymbol{r}_{s y_C}{}^T \boldsymbol{r}_{s s_B}{}^T\right]^T$, and

$$\boldsymbol{R} = \begin{bmatrix} \boldsymbol{R}_{y_C y_C} & \boldsymbol{R}_{y_C s_B} \\ \boldsymbol{R}_{y_C s_B}{}^T & \boldsymbol{R}_{s_B s_B} \end{bmatrix}, \tag{28}$$

but it is also applied to the real equalizer with similar definitions based on Eq. 26.

The initial values $\boldsymbol{R}_{y_C y_C 0}$, $\boldsymbol{R}_{y_C s_B 0}$, and $\boldsymbol{r}_{s y_C 0}$ (or $\boldsymbol{R}_{y_R y_R 0}$, $\boldsymbol{R}_{y_R s_B 0}$, and $\boldsymbol{r}_{s y_R 0}$ for the real equalizer) in the first step of the algorithm may be zero matrices. However, if a channel estimate is available, Eqs. 15-17 (23-25) suggest that convergence time may be reduced by initializing using an estimate of the channel $\boldsymbol{h}$. With the infinite memory averaging in Eq. 27, the contribution of the initial value to the matrix average decreases immediately as $1/k$. In a more practical forgetting factor implementation as described by Chang and Willson,[9] the contribution of the initial value will decrease more slowly. Using the finite data windowing approach of the same paper, the contribution of the initial value can be subtracted at a linear rate until there is no contribution left. To obtain a similar effect using the running average approach in Eq. 27, we use the modified averaging

$$\boldsymbol{R}_{uv}[k] = \frac{1}{k + k_w + 1}\left((k + k_w)\boldsymbol{R}_{uv}[k-1] + \boldsymbol{u}[k]\boldsymbol{v}^T[k]\right), \tag{29}$$

where k_w is the initial weighting of $\boldsymbol{R}_{uv0}$.

4.2. Recursive Least Squares Algorithm

The RLS algorithm solves equations of the form $\boldsymbol{R}\boldsymbol{g} = \boldsymbol{r}$ by updating the inverse of the matrix $\boldsymbol{R}$. The algorithm used[13] is a standard RLS algorithm and is given in Algorithm 2, where $\boldsymbol{y}[k] = \left[{\boldsymbol{y}_C}^T[k]\ {\boldsymbol{s}_B}^T[k-K-1]\right]^T$ ($\boldsymbol{y}_R[k]$ instead of

Algorithm 2 Adaptive RLS Algorithm

$\boldsymbol{P}[0] = \delta\boldsymbol{I}, \boldsymbol{g}[0] = \boldsymbol{g}_0$
for $k = 1, \ldots$ **do**
$\boldsymbol{b}[k] = \boldsymbol{P}[k-1]\boldsymbol{y}[k]$
$\boldsymbol{t}[k] = \dfrac{\boldsymbol{b}[k]}{\lambda + \boldsymbol{y}^T[k]\boldsymbol{b}[k]}$
$\epsilon_k = \tilde{s}[k-K] - \boldsymbol{g}^T[k-1]\boldsymbol{y}[k]$
$\boldsymbol{g}[k] = \boldsymbol{g}[k-1] + \epsilon_k\boldsymbol{t}[k]$
$\boldsymbol{P}[k] = \frac{1}{\lambda}\left(\boldsymbol{P}[k-1] - \boldsymbol{t}^T[k]\boldsymbol{y}[k]\boldsymbol{P}[k-1]\right)$
end for

$\boldsymbol{y}_C[k]$ for the real equalizer), λ is the forgetting factor, and δ is the initialization factor for $\boldsymbol{P}$, the inverse of $\boldsymbol{R}$. The symbol $\tilde{s}[k-K]$ used in calculating the error ϵ_k is a known, correct symbol when training the equalizer and a sliced decision when in decision-directed mode.

Since the RLS algorithm updates the inverse of $\boldsymbol{R}$, there is no obvious way to initialize $\boldsymbol{P}$ using a channel estimate. Initialization of the equalizer $\boldsymbol{g}$ is possible, but this provides no advantage over the CG algorithm since the same can be done there.

4.3. Least Mean Square Algorithm

We use a standard LMS algorithm[13] with step size μ which is shown as Algorithm 3. As in the RLS algorithm, the symbol

Algorithm 3 Adaptive LMS Algorithm

$\boldsymbol{g}[0] = \boldsymbol{g}_0$
for $k = 1, \ldots$ **do**
$\hat{s}[k-K] = \boldsymbol{g}^T[k-1]\boldsymbol{y}[k]$
$\boldsymbol{g}[k] = \boldsymbol{g}[k-1] + \mu(\tilde{s}[k-K] - \hat{s}[k-K])\boldsymbol{y}[k]$
end for

$\tilde{s}[k-K]$ used in calculating the error ϵ_k is a known, correct symbol when training the equalizer and a sliced decision when in decision-directed mode.

5. SIMULATIONS

The equalizers were simulated using four channels with $M = 4$ paths in each channel. The gains of each path were randomly generated from uncorrelated complex Gaussian random variables with mean zero and variance 2. The maximum gain was normalized to one and located at zero delay. Three random delays were generated via independent uniform random variables across a delay spread of -30 to 250 symbols, relative to zero delay, and the remaining normalized gains were assigned to these paths. Table 1 shows the delays, relative gains, and phases of the paths in the four channels simulated. The complex pulse shape at the transmitter and receiver had an excess bandwidth of 0.1152[1] and was modeled using 61 samples centered at the main tap (i.e., $L_q = 30$). The equalizers had 101 feedforward taps and 300 feedback taps and the cursor was positioned at tap 76 in the feedforward section.

All simulations were performed at a received SNR of $30\mathrm{dB}$, where the SNR is defined at the input to the receiver matched filter. The SNR is calculated as

$$\mathrm{SNR} = \frac{\mathcal{E}_s \left\|\boldsymbol{c}\right\|^2}{N_0}, \tag{30}$$

Table 1: Simulated channel delays in symbol periods, relative gains in dB, and phases in degrees.

Chan	Main Path			Path 2			Path 3			Path 4		
	Delay	Gain	Phase	Delay	Gain	Phase	Delay	Gain	Phase	Delay	Gain	Phase
1	0	0	0	19.4	-6.45	291.2	176.7	-0.97	303.5	228.1	-0.28	245.0
2	0	0	0	-13.8	-7.98	146.8	84.9	-2.39	285.2	220.2	-5.59	342.8
3	0	0	0	-27.2	-13.86	91.5	68.8	-4.97	289.0	197.7	-4.67	182.5
4	0	0	0	8.9	-8.33	328.3	25.6	-4.99	299.1	26.8	-1.67	0.8

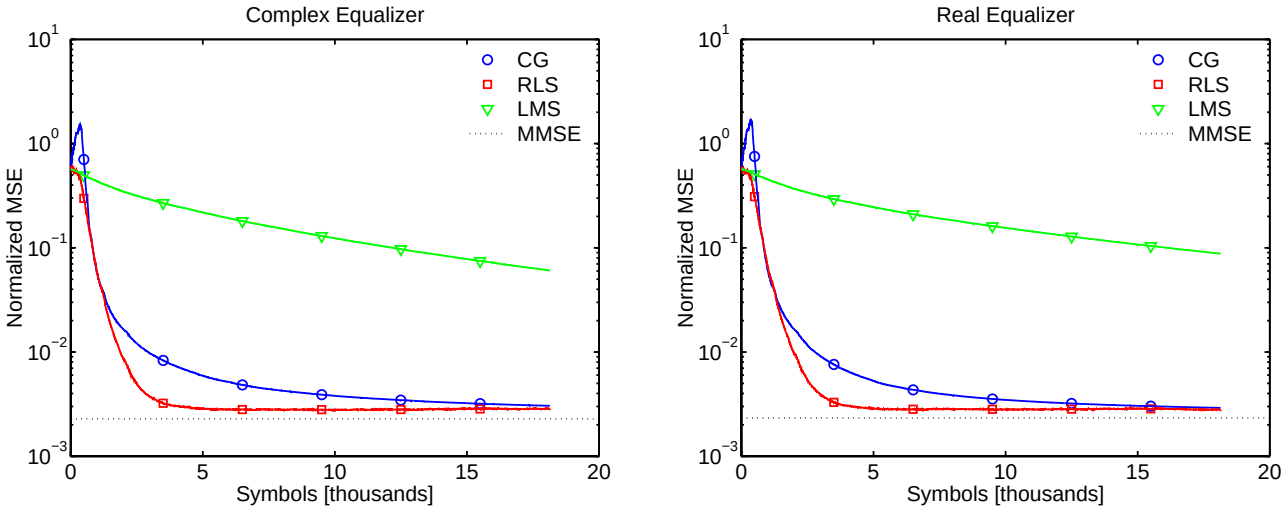

Figure 2. Complex (left) and real (right) equalizer convergence curves for CG, RLS, and LMS averaged across four channels at a received SNR of 30dB.

where $\mathcal{E}_s = 21$, $\boldsymbol{c} = [\, c[-L_{ha} + L_q], \ldots, c[L_{hc} - L_q]\,]^T$, and the noise power is calculated using the effective system bandwidth.

The LMS and RLS parameters were chosen by simulating various parameter combinations and choosing the parameters which yielded the fastest stable convergence. The LMS step size was $\mu = 10^{-5}$, and the RLS parameters were $\lambda = 0.999$ and $\delta = 10^{-3}$. In the CG algorithm, the conjugate direction was re-initialized after $k_0 = 502$ steps for the complex equalizer and $k_0 = 401$ steps for the real equalizer. In both cases, k_0 is equal to the dimension of the $\boldsymbol{R}$ matrix, as suggested in Chong and Zak.[12] For all algorithms, the mean-square error was calculated using the actual channel and noise parameters according to Eq. 12 and then normalized to the constellation variance $\mathcal{E}_s$.

In the first set of results, we compare the convergence rates of CG, RLS, and LMS without any special initialization (other than the diagonal loading of $\boldsymbol{P}[0]$ for RLS as described above). The equalizer weights are initialized to zeros except for a one at the cursor location (tap 76) in the feedforward section. The feedback tap delay line contains zeros when training begins. For the CG algorithm, $\boldsymbol{R}_{y_C y_C 0}$, $\boldsymbol{R}_{y_C s_B 0}$, and $\boldsymbol{r}_{s y_C 0}$ ($\boldsymbol{R}_{y_R y_R 0}$, $\boldsymbol{R}_{y_R s_B 0}$, and $\boldsymbol{r}_{s y_R 0}$) are zero matrices; $\boldsymbol{R}_{s_B s_B}[0]$ and $\boldsymbol{r}_{s s_B}[0]$ are initialized as noted in Algorithm 1. The results for the complex and real equalizers are shown in Fig. 2 for a received SNR of 30dB. The results have been averaged over all four channels. Both the RLS and CG algorithms converge much faster than the LMS algorithm. The CG and RLS algorithms have similar early descents except for the initial rise in MSE for the CG algorithm. The CG algorithm has slower convergence after the initial fall in MSE. Note, however, than none of the three algorithms converges within the 704 training symbols available in a VSB data field.

Next, we compare several methods of initializing the CG algorithm. It is possible to initialize the equalizer tap weights, the correlation matrices and vectors, and we may initialize the feedback tap contents to zero or fill the taps with training

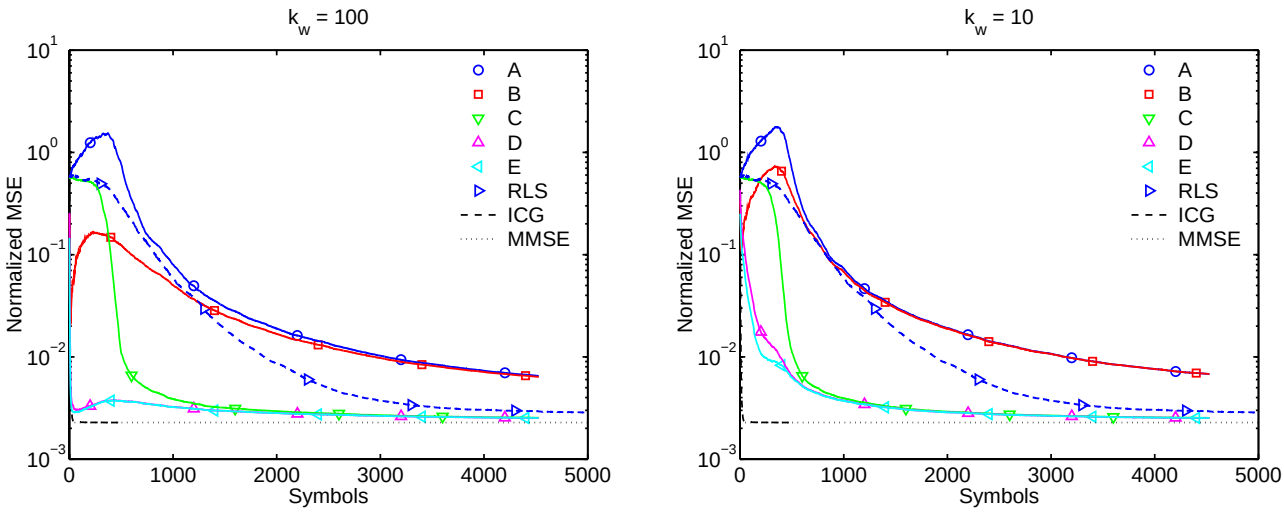

Figure 3. Convergence of complex CG equalizers initialized using actual channels at a received SNR of 30dB. The plot on the left is for an initial weight of $k_w = 100$. The plot on the right is for an initial weight of $k_w = 10$. The letters refer to the initialization descriptions in the text.

symbols. When we fill the feedback taps with training symbols, we use the first 300 training symbols, so these symbols are no longer available to train the equalizer. We simulated five different initializations:

A. Minimal initialization. This is identical to the first set of results above.

B. Matrix initialization. In this case, we initialize the correlation matrices using the actual channel $\boldsymbol{h}$ and noise variance N_0.

C. Feedback tap initialization. Here, we fill the feedback taps with training symbols prior to beginning adaptation. The correlation matrices are not initialized using the channel.

D. Matrix and feedback tap initialization. This is a combination of cases B and C. We initialize the matrices using the actual channel and noise variance, and we fill the feedback taps with training symbols.

E. Matrix and feedback tap initialization with equalizer tap weight initialization. This case is identical to case D except, in addition, we provide a simple initialization of the equalizer tap weights using the negative of the post-cursor portion of the channel.

In addition to these initializations, we also simulated the CG solution[12] of $\boldsymbol{Rg} = \boldsymbol{r}$ and calculated the MSE as a function of the iteration. We call this solution indirect CG (ICG) since it could be implemented as an indirect adaptation of the equalizer taps by tracking the channel estimate and calculating the tap weights by forming $\boldsymbol{R}$ and $\boldsymbol{r}$ from the channel estimate. Finally, we reiterate that performing matrix initialization of the RLS algorithm similar to that described in B would be difficult because the matrix updated by the RLS algorithm is the inverse of the correlation matrix.

The results of this simulation for the complex equalizer are shown in Fig. 3 along with the RLS results from the previous simulation. Results for the real equalizer are similar and are not shown here. The matrix initialization alone (B) does not yield a large improvement in the convergence time. The MSE initially drops steeply but then rises before falling again. This can be attributed to the zeros initially in the feedback taps which initially direct the matrix averages away from their initial values. Feedback tap initialization (C) does yield a significant improvement in convergence time, but a flat region remains in the MSE. This region can be attributed to the time it takes for the matrix averages to stabilize. As a result, the combination of B and C, shown as D, removes the flat region found in feedback tap initialization and a vast improvement is seen over minimal initialization (A). With an initial weighting of $k_w = 100$, the initial convergence trajectory approaches that of ICG. Even with $k_w = 10$, the normalized MSE quickly reaches 10^{-2}. When the equalizer

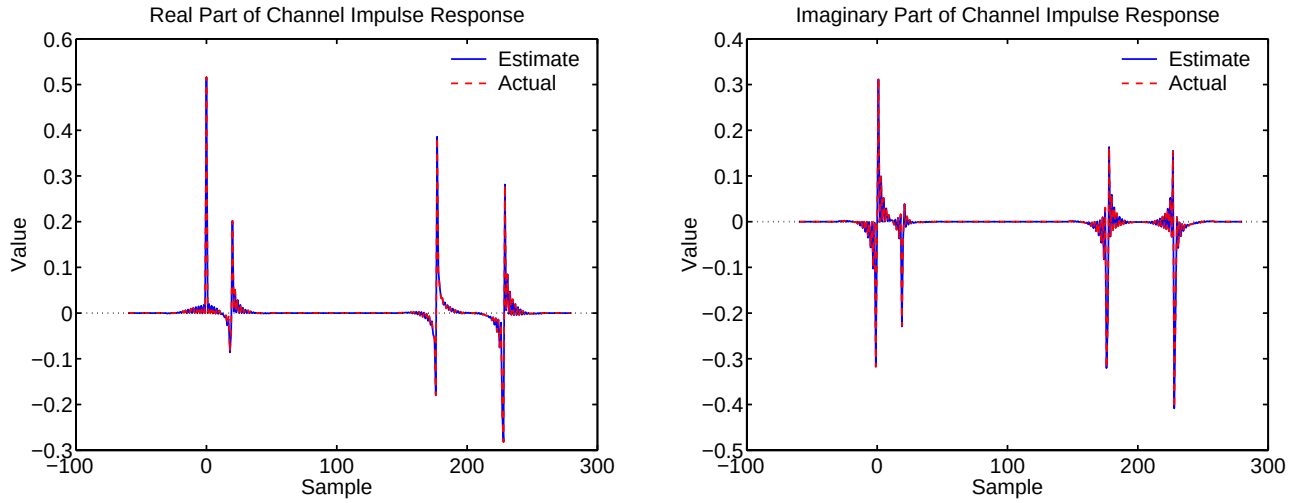

Figure 4. Comparison of estimated and actual channel impulse responses. The estimated responses were obtained at a received SNR of $30\mathrm{dB}$.

Table 2: Comparison of actual and estimated noise variances at $\mathrm{SNR} = 30\mathrm{dB}$.

Channel	Noise Variance	
	Actual	Estimated
1	0.03105	0.02825
2	0.02117	0.02150
3	0.01787	0.01944
4	0.02642	0.02543

taps are also initialized (E), there is a slight further improvement in convergence time. This is most easily seen when $k_w = 10$.

Finally, we test our initialization scheme using an estimated channel. We simulated the equalizer using the same four channels, but estimated the channel using the scheme described by Özen, et al.[7] The channel estimation is performed in two steps using symbol-spaced received samples prior to the receiver matched filter. In the first step, the received samples are correlated with the stored training sequence and thresholding is applied in order to determine the locations of the multipath delays. The purpose of the second step is to incorporate the transmitted pulse shape $q(t)$ into the channel impulse response. To do this, we locate three copies of $q(t)$ shifted by one-half of a symbol period around each multipath location and estimate complex scaling factors using a modified least squares approach. Further details may be found in the reference. Figure 4 compares the actual channel impulse response with the estimated impulse response for channel 1. The estimated impulse response matches the actual response very closely. These results are representative of the results for all four channels.

Using the channel estimate, an estimate of the noise variance is also found. This is done by reconstructing the received signal using the channel estimate and the training sequence. Only samples which depend on the training sequence and no other symbols are reconstructed so that the reconstructed samples are essentially a noiseless estimate of the received signal. The reconstructed signal is then subtracted from the actual received signal, and the squared magnitudes of the differences are averaged and divided by two to yield the estimate of the noise variance N_0. Table 2 compares the actual and estimated noise variance for all four channels at a received SNR of $30\mathrm{dB}$. Note that the noise variance changes with channel conditions since, in the received SNR, the signal includes the contributions from all four paths.

The channel estimate is used to initialize the correlation matrices in the equalizer prior to adaptation and, for case E, to initialize the feedback tap weights. We simulated initialization cases D and E for the complex equalizer and the

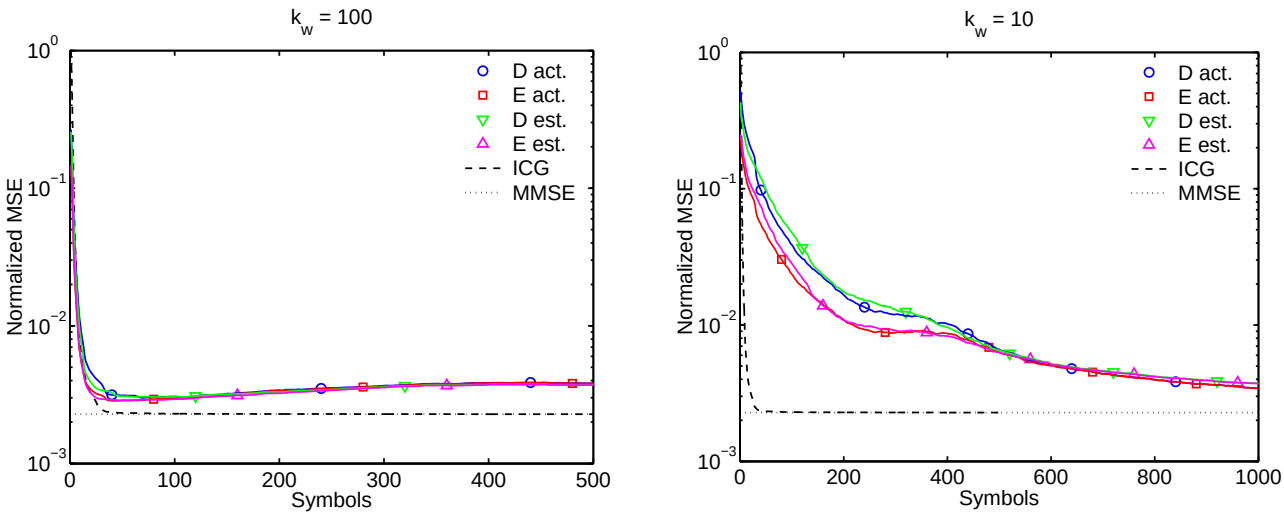

Figure 5. Convergence of complex CG equalizers initialized using channel estimates at a received SNR of $30\mathrm{dB}$. The convergence curves for initialization with the actual channels and the estimated channels are compared.

results are compared with the previous results using the actual channel in Fig. 5. For both $k_w = 10$ and $k_w = 100$, there is little difference between the convergence rates when the equalizers are initialized with the estimated channels and the convergence rates using the actual channels. With $k_w = 100$, the equalizer is able to converge within the available number of training symbols in a single VSB synchronization segment without resorting to blind adaptation techniques.

6. CONCLUSIONS

We have shown that the adaptive conjugate gradient and RLS algorithms have similar convergence rates without initialization in an 8-VSB system. Since the conjugate gradient algorithm is easily initialized when a channel estimate is available, we explored several methods for initialization. We found that initializing the correlation matrices while ensuring that the feedback taps contained known training symbols yielded a significant improvement in convergence time — much more than either technique yielded on its own. In addition, initializing the equalizer feedback tap weights using the post-cursor portion of the channel estimate provided an additional small improvement in convergence time. The improved convergence times are an advantage for the conjugate gradient algorithm since convergence is possible within the available training symbols in an 8-VSB field. Correlation matrix initialization of the RLS algorithm is difficult because the algorithm updates the inverse of the correlation matrix.

Future work will center on implementing the initialization scheme presented here using finite length data windowing and forgetting factor correlation matrix averaging. In addition, we will work to reduce the computational complexity of the conjugate gradient algorithm.

ACKNOWLEDGMENTS

This research was funded by the National Science Foundation under grant number CCR-0118842 and by a research gift from the Zenith Electronics Corporation.

REFERENCES

1. Advanced Television Systems Committee, “ATSC standard: Digital television standard, revision B.” Document A/53B, 7 August 2001.
2. Advanced Television Technology Center, “Evaluation of ATSC 8-VSB receiver performance in the presence of simulated multipath and noise.” Document 99-04A, 13 September 1999.

3. J. Oh, Y. Chang, K. Ha, S. Jung, and J. Kim, "A single VSB/QAM/QPSK IC for ATSC and OpenCable™ digital terminals," *IEEE Transaction on Consumer Electronics* **47**, pp. 443–449, August 2001.
4. N. Deshpande, "Fast recovery equalization techniques for DTV signals," *IEEE Transaction on Broadcasting* **43**, pp. 370–377, December 1997.
5. D. J. Kim, S. W. Park, Y. H. Kim, and W.-S. Yoon, "Timing-offset independent equalization techniques for robust indoor reception of ATSC DTV receivers," *IEEE Transaction on Consumer Electronics* **46**, pp. 442–448, August 2000.
6. M. Ghosh, "Blind decision feedback equalization for terrestrial television receivers," *Proceedings of the IEEE* **86**, pp. 2070–2081, October 1998.
7. S. Özen, M. D. Zoltowski, and M. Fimoff, "A novel channel estimation method: Blending correlation and least-squares based approaches," in *ICASSP 2002*, (Orlando, FL), May 2002. To be published.
8. G. Dietl, M. D. Zoltowski, and M. Joham, "Reduced-rank equalization for edge via conjugate gradient implementation of multi-stage nested wiener filter," in *2001 Vehicular Technology Conference (Fall)*, **3**, pp. 1912–1916, (Atlantic City, New Jersey), 7–11 October 2001.
9. P. S. Chang and A. N. Willson, Jr., "Analysis of conjugate gradient algorithms for adaptive filtering," *IEEE Transactions on Signal Processing* **48**, pp. 409–418, February 2000.
10. S. Chowdhury and M. D. Zoltowski, "Application of conjugate gradient methods in MMSE equalization for the forward link of DS-CDMA," in *2001 Vehicular Technology Conference (Fall)*, **4**, pp. 2434–2438, (Atlantic City, New Jersey), 7–11 October 2001.
11. T. A. Thomas, *Space-Time Processing for Interference Cancellation and Equalization in Narrowband Digital Communications*. PhD thesis, Purdue University, December 1997.
12. E. K. P. Chong and S. Żak, *An Introduction to Optimization*, John Wiley & Sons, New York, NY, 1996.
13. S. S. Haykin, *Adaptive Filter Theory*, Prentice-Hall, Upper Saddle River, NJ, 2002.