COMPLEX PROPORTIONATE-TYPE NORMALIZED LEAST MEAN SQUARE ALGORITHMS

Kevin T. Wagner

Naval Research Laboratory
Radar Division
Washington, DC 20375, USA

Miloš I. Doroslovački

The George Washington University
Department of Electrical and Computer Engineering
Washington, DC 20052, USA

ABSTRACT

A complex proportionate-type normalized least mean square algorithm is derived by minimizing the second norm of the weighted difference between the current estimate of the impulse response and the estimate at the next time step under the constraint that the adaptive filter a posteriori output is equal to the measured output. The weighting function is assumed positive but otherwise arbitrary and it is directly related to the update gains. No assumptions regarding the input signal are made during the derivation. Different weights (i.e., gains) are used for real and imaginary parts of the estimated impulse response. After additional assumptions special cases of the algorithm are obtained: the algorithm with one gain per impulse response coefficient and the algorithm with lower computational complexity. The learning curves of the algorithms are compared for several standard gain assignment laws for white and colored input. It was demonstrated that, in general, the algorithms with separate gains for real and imaginary parts have faster convergence.

***Index Terms*—** Adaptive filtering, convergence, least mean square algorithms.

1. MOTIVATION

The complex least mean square (CLMS) adaptive filter [1] was originally proposed to extend the least mean square algorithm from real-valued signals to complex-valued signals. To date there exists no equivalent extension of proportionate-type normalized least mean square algorithms (PtNLMS) from real-valued signals to complex-valued signals. PtNLMS algorithms are a family of adaptive algorithms which update the estimate of the impulse response in a manner which is proportional to the magnitude of the current estimate of the impulse response. Examples of PtNLMS algorithms include the proportionate normalized least mean square (PNLMS) [2] algorithm, and μ-PNLMS (MPNLMS) [3] algorithm.

In this paper, the PtNLMS algorithm is extended from real-valued signals to complex-valued signals. The resulting algorithm is named the complex PtNLMS (CPtNLMS) algorithm. The CPtNLMS algorithm updates the real and imaginary parts of the unknown impulse response by applying separate real-valued gains. Additionally, several simplifications of the CPtNLMS will be proposed.

This paper is organized in the following manner. First, notation and the adaptive filter framework will be introduced. Next the derivation of the CPtNLMS algorithm will be presented as well as several simplifications and interpretations. After that, examples of CPtNLMS algorithms and their implementations will be discussed. Next simulation results for various CPtNLMS algorithms will be presented and finally a conclusion summarizing the results found in this paper will be given.

2. COMPLEX ADAPTIVE FILTER ALGORITHM INTRODUCTION AND NOTATION

2.1. Notation

Let $\mathbf{A}$ denote an arbitrary complex-valued matrix. The conjugate, transpose, and conjugate transpose of the matrix $\mathbf{A}$ are denoted by $\mathbf{A}^*$, $\mathbf{A}^T$, and $\mathbf{A}^H$; respectively. Additionally, Re$(\mathbf{A})$ and Im$(\mathbf{A})$ represent the real and imaginary parts of the complex matrix $\mathbf{A}$. These operators have the same meanings when employed on vectors and scalars where applicable. Let the i^{th} component of any vector $\mathbf{a}$ be denoted as a_i and the $(i,j)^{\text{th}}$ entry of any matrix $\mathbf{A}$ as $[\mathbf{A}]_{ij}$ throughout this work. Next, for vector $\mathbf{a}$ with length L we define the function $\mathbf{Diag}(\mathbf{a})$ as an $L \times L$ matrix whose diagonal entries are the L elements of $\mathbf{a}$ and all other entries are zero. Finally, we define $j = \sqrt{-1}$.

2.2. Complex Adaptive Filter Framework

All signals are complex throughout this work. Let us assume there is some complex input signal denoted as $\mathbf{x}(k)$ for time k that excites an unknown system with complex impulse response $\mathbf{w}$. Let the output of the system be $y(k) = \mathbf{x}^H(k)\mathbf{w}(k)$ where $\mathbf{x}(k) = [x(k), x(k-1), \ldots, x(k-L+1)]^T$ and L is the length of the filter. The measured output of the system, $d(k)$, contains complex-valued measurement noise $v(k)$ and is equal to the sum of $y(k)$ and $v(k)$. The impulse response of the system is estimated with the adaptive filter coefficient vector, $\hat{\mathbf{w}}(k)$, which has length L also. The output of the adaptive filter is given by $\hat{y}(k) = \mathbf{x}^H(k)\hat{\mathbf{w}}(k)$. The error signal $e(k)$ is equal to difference of the measured output, $d(k)$ and the output of the adaptive filter $\hat{y}(k)$.

For notational convenience we will suppress the time-indexing notation and instead use the following convention. For an arbitrary vector $\mathbf{a}$ we denote $\mathbf{a}(k+1) = \mathbf{a}^+$ and $\mathbf{a}(k) = \mathbf{a}$.

3. COMPLEX PROPORTIONATE-TYPE NORMALIZED LEAST MEAN SQUARE ALGORITHM DERIVATION

3.1. Complex PtNLMS Algorithm Using Separate Gains for Coefficient Imaginary and Real Parts

The estimated weight vector can be written in terms of real and imaginary parts $\hat{\mathbf{w}} = \hat{\mathbf{w}}_R + j\hat{\mathbf{w}}_I$, where $\hat{\mathbf{w}}_R$ and $\hat{\mathbf{w}}_I$ are vectors of length L consisting of the real and imaginary components of $\hat{\mathbf{w}}$, respectively. Motivated by the derivation of the normalized least mean

978-1-4673-0046-9/12/$26.00 ©2012 IEEE

square (NLMS) algorithm [4] let us consider the following minimization problem

$$\begin{aligned}\min_{\hat{\mathbf{w}}^+} \quad & (\hat{\mathbf{w}}_R^+ - \hat{\mathbf{w}}_R)^T \mathbf{M}_R^{-1} (\hat{\mathbf{w}}_R^+ - \hat{\mathbf{w}}_R) \\ & +(\hat{\mathbf{w}}_I^+ - \hat{\mathbf{w}}_I)^T \mathbf{M}_I^{-1} (\hat{\mathbf{w}}_I^+ - \hat{\mathbf{w}}_I) \\ \text{such that} \quad & d = \mathbf{x}^H \hat{\mathbf{w}}^+ \end{aligned} \quad (1)$$

where $\mathbf{M}_R$ and $\mathbf{M}_I$ are real-valued, nonnegative, diagonal matrices, which have the property that $\text{Tr}[\mathbf{M}_R] = \text{Tr}[\mathbf{M}_I] = L$.

The method of Lagrange multipliers will be used to cast this constrained minimization problem into one of unconstrained minimization. Prior to performing this step the following definitions are introduced to aide in the derivation. Let

$$\hat{\boldsymbol{\omega}} = \begin{bmatrix} \hat{\mathbf{w}}_R \\ \hat{\mathbf{w}}_I \end{bmatrix} \quad \mathbf{M} = \begin{bmatrix} \mathbf{M}_R & \mathbf{0} \\ \mathbf{0} & \mathbf{M}_I \end{bmatrix}. \quad (2)$$

The following form of $\mathbf{x}^H \hat{\mathbf{w}}$ will also be employed

$$\begin{aligned}\mathbf{x}^H \hat{\mathbf{w}} &= \mathbf{x}_R^T \hat{\mathbf{w}}_R + \mathbf{x}_I^T \hat{\mathbf{w}}_I + j\mathbf{x}_R^T \hat{\mathbf{w}}_I - j\mathbf{x}_I^T \hat{\mathbf{w}}_R \\ &= [\mathbf{x}_R^T,\ \mathbf{x}_I^T] \begin{bmatrix} \hat{\mathbf{w}}_R \\ \hat{\mathbf{w}}_I \end{bmatrix} + j[-\mathbf{x}_I^T,\ \mathbf{x}_R^T] \begin{bmatrix} \hat{\mathbf{w}}_R \\ \hat{\mathbf{w}}_I \end{bmatrix} \\ &= [\mathbf{x}^H,\ j\mathbf{x}^H] \begin{bmatrix} \hat{\mathbf{w}}_R \\ \hat{\mathbf{w}}_I \end{bmatrix} \\ &= [\mathbf{x}^H,\ j\mathbf{x}^H] \hat{\boldsymbol{\omega}}. \end{aligned} \quad (3)$$

Using these definitions the minimization problem can be rewritten as

$$\begin{aligned}\min_{\hat{\boldsymbol{\omega}}^+} J(\hat{\boldsymbol{\omega}}^+) &= (\hat{\boldsymbol{\omega}}^+ - \hat{\boldsymbol{\omega}})^T \mathbf{M}^{-1} (\hat{\boldsymbol{\omega}}^+ - \hat{\boldsymbol{\omega}}) \\ &+ \lambda \left(d - [\mathbf{x}^H, j\mathbf{x}^H] \hat{\boldsymbol{\omega}} \right) \\ &+ \lambda^* \left(d^* - [\mathbf{x}^T, -j\mathbf{x}^T] \hat{\boldsymbol{\omega}} \right). \end{aligned} \quad (4)$$

Next taking the derivative of $J(\hat{\boldsymbol{\omega}}^+)$ with respect to $\hat{\boldsymbol{\omega}}^+$ and setting the result to zero yields

$$\begin{aligned}\frac{\partial J(\hat{\boldsymbol{\omega}}^+)}{\partial \hat{\boldsymbol{\omega}}^+} &= 2\mathbf{M}^{-1}(\hat{\boldsymbol{\omega}}^+ - \hat{\boldsymbol{\omega}}) - \lambda \begin{bmatrix} \mathbf{x}^* \\ j\mathbf{x}^* \end{bmatrix} - \lambda^* \begin{bmatrix} \mathbf{x} \\ -j\mathbf{x} \end{bmatrix} \\ &= 0. \end{aligned} \quad (5)$$

Multiplying (5) from the left by $\mathbf{M}$ and rearranging terms allows us to write

$$2(\hat{\boldsymbol{\omega}}^+ - \hat{\boldsymbol{\omega}}) = \lambda \mathbf{M} \begin{bmatrix} \mathbf{x}^* \\ j\mathbf{x}^* \end{bmatrix} + \lambda^* \mathbf{M} \begin{bmatrix} \mathbf{x} \\ -j\mathbf{x} \end{bmatrix}. \quad (6)$$

Next we employ the relationships given by $[\mathbf{x}^H,\ j\mathbf{x}^H](\hat{\boldsymbol{\omega}}^+ - \hat{\boldsymbol{\omega}}) = e$ and $[\mathbf{x}^T,\ -j\mathbf{x}^T](\hat{\boldsymbol{\omega}}^+ - \hat{\boldsymbol{\omega}}) = e^*$ to form the following linear system of equations

$$2 \begin{bmatrix} e^* \\ e \end{bmatrix} = \begin{bmatrix} \mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x} & \mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x} \\ \mathbf{x}^H(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}^* & \mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x} \end{bmatrix} \begin{bmatrix} \lambda \\ \lambda^* \end{bmatrix}. \quad (7)$$

It is straightforward to invert the matrix given in (7) which allows us to find

$$\lambda = 2 \frac{\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}e^* - \mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}e}{[\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}]^2 - |\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}|^2}. \quad (8)$$

Returning our attention to (6) this equation can be rewritten as

$$2(\hat{\boldsymbol{\omega}}^+ - \hat{\boldsymbol{\omega}}) = \lambda \begin{bmatrix} \mathbf{M}_R \mathbf{x}^* \\ j\mathbf{M}_I \mathbf{x}^* \end{bmatrix} + \lambda^* \begin{bmatrix} \mathbf{M}_R \mathbf{x} \\ -j\mathbf{M}_I \mathbf{x} \end{bmatrix}. \quad (9)$$

Using the definition in (2) allows us to write

$$\begin{aligned} 2(\hat{\mathbf{w}}_R^+ - \hat{\mathbf{w}}_R) &= \lambda \mathbf{M}_R \mathbf{x}^* + \lambda^* \mathbf{M}_R \mathbf{x} \\ 2(\hat{\mathbf{w}}_I^+ - \hat{\mathbf{w}}_I) &= j\lambda \mathbf{M}_I \mathbf{x}^* - j\lambda^* \mathbf{M}_I \mathbf{x}. \end{aligned} \quad (10)$$

Next using (10) and (8) we form

$$\begin{aligned} &\hat{\mathbf{w}}^+ - \hat{\mathbf{w}} = (\hat{\mathbf{w}}_R^+ - \hat{\mathbf{w}}_R) + j(\hat{\mathbf{w}}_I^+ - \hat{\mathbf{w}}_I) \\ &= \frac{\lambda}{2}\mathbf{M}_R\mathbf{x}^* + \frac{\lambda^*}{2}\mathbf{M}_R\mathbf{x} - \frac{\lambda}{2}\mathbf{M}_I\mathbf{x}^* + \frac{\lambda^*}{2}\mathbf{M}_I\mathbf{x} \\ &= \frac{\lambda}{2}(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}^* + \frac{\lambda^*}{2}(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x} \\ &= \frac{\left[\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}e^* - \mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}e\right]}{[\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}]^2 - |\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}|^2}(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}^* \\ &+ \frac{\left[\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}e - \mathbf{x}^H(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}^*e^*\right]}{[\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}]^2 - |\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}|^2}(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}. \end{aligned} \quad (11)$$

Next the step-size parameter β is introduced to allow control over the update. The resulting CPtNLMS algorithm with separate gains for real and imaginary coefficients is given by

$$\begin{aligned} &\hat{\mathbf{w}}^+ = \hat{\mathbf{w}} \\ &+\beta \frac{\left[\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}e^* - \mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}e\right]}{[\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}]^2 - |\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}|^2}(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}^* \\ &+\beta \frac{\left[\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}e - \mathbf{x}^H(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}^*e^*\right]}{[\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}]^2 - |\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}|^2}(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}. \end{aligned} \quad (12)$$

3.2. CPtNLMS Algorithm Simplifications

The CPtNLMS algorithm given in (12) can be simplified under certain conditions. For instance it is straightforward to show that by setting $\mathbf{M}_R = \mathbf{M}_I = \mathbf{G}$ in (12) the algorithm is reduced to

$$\hat{\mathbf{w}}^+ = \hat{\mathbf{w}} + \beta \frac{\mathbf{G}\mathbf{x}e}{\mathbf{x}^H \mathbf{G} \mathbf{x}}. \quad (13)$$

As it turns out (13) is the solution to the following minimization

$$\begin{aligned}\min_{\hat{\mathbf{w}}^+} \quad & (\hat{\mathbf{w}}^+ - \hat{\mathbf{w}})^H \mathbf{G}^{-1} (\hat{\mathbf{w}}^+ - \hat{\mathbf{w}}) \\ \text{such that} \quad & d = \mathbf{x}^H \hat{\mathbf{w}}^+ \end{aligned} \quad (14)$$

where $\mathbf{G}$ is the stepsize control matrix. This version of the CPtNLMS algorithm uses one real-valued gain to simultaneously update both the real and imaginary parts of the estimated coefficient.

A separate simplification to (12) can be made if it is assumed that

$$\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x} \approx 0. \quad (15)$$

Using this approximation (12) is reduced to

$$\hat{\mathbf{w}}^+ = \hat{\mathbf{w}} + \beta \frac{\mathbf{M}_R(\mathbf{x}e + \mathbf{x}^*e^*) + \mathbf{M}_I(\mathbf{x}e - \mathbf{x}^*e^*)}{\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}}. \quad (16)$$

This version of the CPtNLMS algorithm reduces the computational complexity of (12) and will be referred to as the simplified CPtNLMS algorithm.

Table I		
Stepsize Control Matrix for CPtNLMS Algorithm Using One Real-valued Gain per Coefficient		
$F[\|\hat{w}_l(k)\|, k]$	=	Specified by the user
$\gamma_{min}(k)$	=	$\rho \max\{\delta_p, F[\|\hat{w}_1(k)\|, k], \dots, F[\|\hat{w}_L(k)\|, k]\}$
$\gamma_l(k)$	=	$\max\{\gamma_{min}(k), F[\|\hat{w}_l(k)\|, k]\}$
$g_l(k)$	=	$\frac{\gamma_l(k)}{\frac{1}{L}\sum_{i=1}^{L}\gamma_i(k)}$
$\mathbf{G}(k)$	=	$\mathbf{Diag}\{g_1(k), \dots, g_L(k)\}$

3.3. Alternative Representations of the CPtNLMS Algorithm

There are alternatives ways to represent (11). For instance one could rearrange terms such that

$$\begin{aligned}\hat{\mathbf{w}}^+ - \hat{\mathbf{w}} &= (\hat{\mathbf{w}}_R^+ - \hat{\mathbf{w}}_R) + j(\hat{\mathbf{w}}_I^+ - \hat{\mathbf{w}}_I) \\ &= \mathbf{M}_R \frac{\lambda^* \mathbf{x} + \lambda \mathbf{x}^*}{2} + \mathbf{M}_I \frac{\lambda^* \mathbf{x} - \lambda \mathbf{x}^*}{2}.\end{aligned} \quad (17)$$

Using (8) it is easy to show that

$$\begin{aligned}\frac{\lambda^* \mathbf{x} + \lambda \mathbf{x}^*}{2} &= 2\frac{\mathrm{Re}\,[e\mathbf{x}] - \mathrm{Re}\left[\frac{\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}}{\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}} e\mathbf{x}^*\right]}{\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x} - \frac{|\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}|^2}{\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}}} \\ \frac{\lambda^* \mathbf{x} - \lambda \mathbf{x}^*}{2} &= 2j\frac{\mathrm{Im}\,[e\mathbf{x}] + \mathrm{Im}\left[\frac{\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}}{\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}} e\mathbf{x}^*\right]}{\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x} - \frac{|\mathbf{x}^T(\mathbf{M}_R - \mathbf{M}_I)\mathbf{x}|^2}{\mathbf{x}^H(\mathbf{M}_R + \mathbf{M}_I)\mathbf{x}}}.\end{aligned} \quad (18)$$

Combining (17) and (18), and introducing the stepsize parameter β will result in an alternate form of the CPtNLMS algorithm. In this version of the CPtNLMS algorithm, the estimated impulse response coefficients are updated by one purely real term multiplying $\mathbf{M}_R$ and another purely imaginary term multiplying $\mathbf{M}_I$.

4. CPtNLMS ALGORITHM AND IMPLEMENTATIONS

In this section several examples of CPtNLMS algorithms are presented. The stepsize control matrix for the CPtNLMS algorithm using one real-valued gain per coefficient is shown in Table I. The term $F[|\hat{w}_l(k)|, k]$, with $l \in \{1, 2, \dots, L\}$, governs how each coefficient is updated. In the case when $F[|\hat{w}_l(k)|, k]$ is less than γ_{min}, the quantity γ_{min} is used to set the minimum gain a coefficient can receive. The constant δ_p, where $\delta_p \geq 0$, is important in the beginning of learning when all of the coefficients are zero and together with ρ, where $\rho \geq 0$, prevents the very small coefficients from stalling. $\mathbf{G}(k) = \mathbf{Diag}\ \{g_1(k), \dots, g_L(k)\}$ is the time-varying stepsize control diagonal matrix. The constant δ is typically a small positive number used to avoid overflowing. Some common examples of the term $F[|\hat{w}_l(k)|, k]$ are $F[|\hat{w}_l(k)|, k] = 1$, $F[|\hat{w}_l(k)|, k] = |\hat{w}_l(k)|$, and $F[|\hat{w}_l(k)|, k] = \ln(1 + \mu|\hat{w}_l(k)|)$, which result in the complex NLMS (CNLMS), complex PNLMS (CPNLMS), and complex MPNLMS (CMPNLMS) algorithms, respectively. The parameter μ is used to adjust the amount of compression created by the logarithm function.

Similarly the CPtNLMS algorithm using separate real-valued gains for coefficients real and imaginary parts is shown in Table II. In this case the user specifies two functions $F_R[|\hat{w}_{R,l}(k)|, k]$ and $F_I[|\hat{w}_{I,l}(k)|, k]$ which govern how the real and imaginary parts of a coefficient are updated. Next a minimum gain for the real and imaginary parts are calculated, $\gamma_{min,R}(k)$ and $\gamma_{min,I}(k)$, respectively. Finally, the stepsize control matrix for the real, $\mathbf{M}_R(k)$, and imaginary, $\mathbf{M}_I(k)$, parts are obtained. An example of the terms $F_R[|\hat{w}_{R,l}(k)|, k]$ and $F_I[|\hat{w}_{I,l}(k)|, k]$ is $F_R[|\hat{w}_{R,l}(k)|, k] = |\hat{w}_{R,l}(k)|$ and $F_I[|\hat{w}_{I,l}(k)|, k] = |\hat{w}_{I,l}(k)|$. This yields the complex PNLMS algorithm using separate gains for coefficient real and imaginary parts. In a similar vein the complex MPNLMS algorithm can be created by setting $F_R[|\hat{w}_{R,l}(k)|, k] = \ln(1+\mu|\hat{w}_{R,l}(k)|)$ and $F_I[|\hat{w}_{I,l}(k)|, k] = \ln(1 + \mu|\hat{w}_{I,l}(k)|)$.

Table II		
Stepsize Control Matrix for CPtNLMS Algorithm Using Separate Real-valued Gains for Coefficient Real and Imaginary Parts		
$F_R[\|\hat{w}_{R,l}(k)\|, k]$	=	Specified by the user
$F_I[\|\hat{w}_{I,l}(k)\|, k]$	=	Specified by the user
$\gamma_{min,R}(k)$	=	$\rho \max\{\delta_p, F_R[\|\hat{w}_{R,1}(k)\|, k], \dots, F_R[\|\hat{w}_{R,L}(k)\|, k]\}$
$\gamma_{min,I}(k)$	=	$\rho \max\{\delta_p, F_I[\|\hat{w}_{I,1}(k)\|, k], \dots, F_I[\|\hat{w}_{I,L}(k)\|, k]\}$
$\gamma_{R,l}(k)$	=	$\max\{\gamma_{min,R}(k), F_R[\|\hat{w}_{R,l}(k)\|, k]\}$
$\gamma_{I,l}(k)$	=	$\max\{\gamma_{min,I}(k), F_I[\|\hat{w}_{I,l}(k)\|, k]\}$
$m_{R,l}(k)$	=	$\frac{\gamma_{R,l}(k)}{\frac{1}{L}\sum_{i=1}^{L}\gamma_{R,i}(k)}$
$m_{I,l}(k)$	=	$\frac{\gamma_{I,l}(k)}{\frac{1}{L}\sum_{i=1}^{L}\gamma_{I,i}(k)}$
$\mathbf{M}_R(k)$	=	$\mathbf{Diag}\{m_{R,1}(k), \dots, m_{R,L}(k)\}$
$\mathbf{M}_I(k)$	=	$\mathbf{Diag}\{m_{I,1}(k), \dots, m_{I,L}(k)\}$

5. SIMULATION RESULTS

In this section we compare the mean square error (MSE) versus iteration for several CPtNLMS algorithms. In Figures 1 and 2 we plot the real and imaginary parts of the impulse response. The impulse response used has length $L = 512$. The stepsize control parameter β was set to a value of 0.5 for all of the CPtNLMS algorithms. The input signal used in the first set of simulations was white, circular, complex, Gaussian, stationary with power $\sigma_x^2 = 1$. The noise used in all simulations was white, circular, complex, Gaussian, stationary with power $\sigma_v^2 = 10^{-4}$. The parameter values $\rho = 0.01$, $\delta = 0.0001$, $\delta_p = 0.01$, and $\mu = 2960$ were also used.

The MSE learning curve performance for the CNLMS, CPNLMS with one real-valued gain per coefficient, simplified CPNLMS with separate gains for real and imaginary parts, CPNLMS with separate gains for the real and imaginary parts, CMPNLMS with one real-valued gain per coefficient, simplified CMPNLMS with separte gains for the real and imaginary parts, and CMPNLMS with separate gains for the real and imaginary parts algorithms are depicted in Figure 3. Here we see that the CPtNLMS algorithms outperform the CNLMS algorithm in the transient regime. The simplified algorithms perform virtually the same as the exact separate gain algorithms. Additionally, the separate gain for real and imaginary parts of coefficients version result in better convergence than the single real-valued gain versions.

In Figure 4 the MSE performance is displayed for a colored input signal. The input signal consists of colored noise generated by a single pole system as follows:

$$x(k) = \gamma x(k-1) + \alpha(k) \quad (19)$$

where $x(0) = \alpha(0)$, $\alpha(k)$ is a white, circular, complex, Gaussian, stationary process with power $\sigma_\alpha^2 = 1$, and complex pole γ. The value $\gamma = 0.3560 - 0.8266j$ was used in this simulation. The magnitude of γ is equal to 0.9, which implies $\sigma_x^2 = \sigma_\alpha^2/(1 - |\gamma|^2) = 5.263$. Similar trends hold for the colored and white input signal

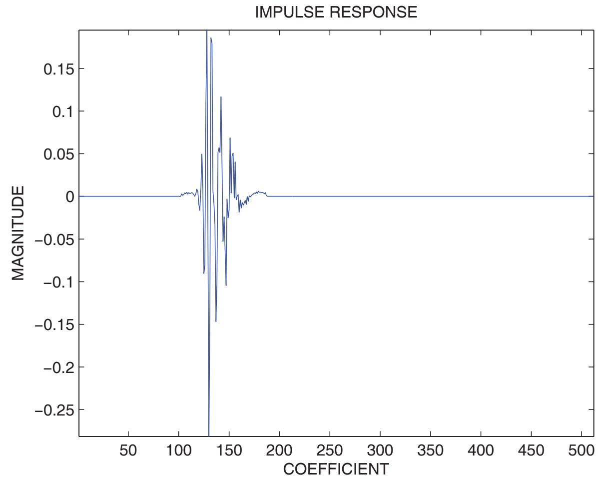

Fig. 1. Real Part of Impulse Response

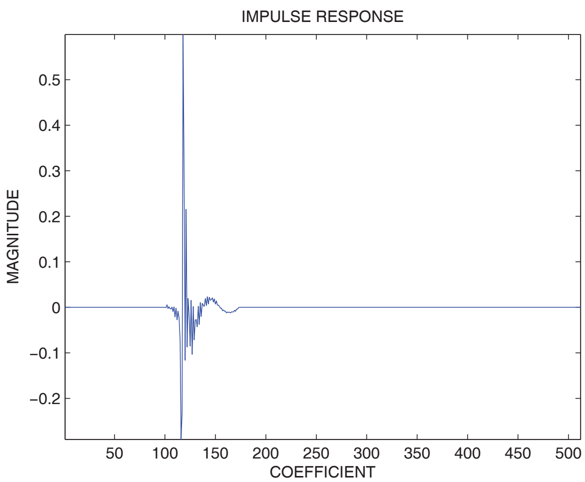

Fig. 2. Imaginary Part of Impulse Response

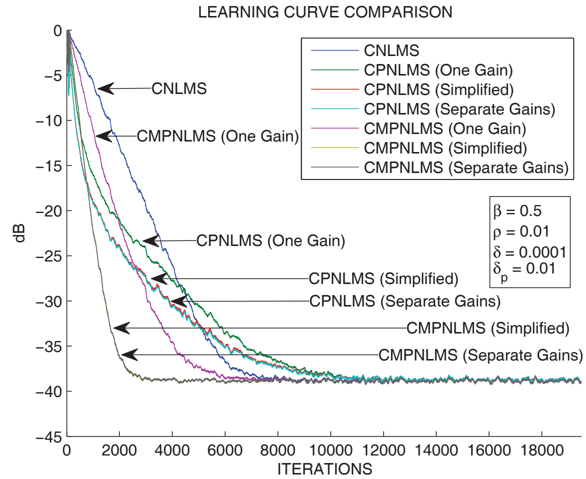

Fig. 3. CPtNLMS Learning Curve Comparison

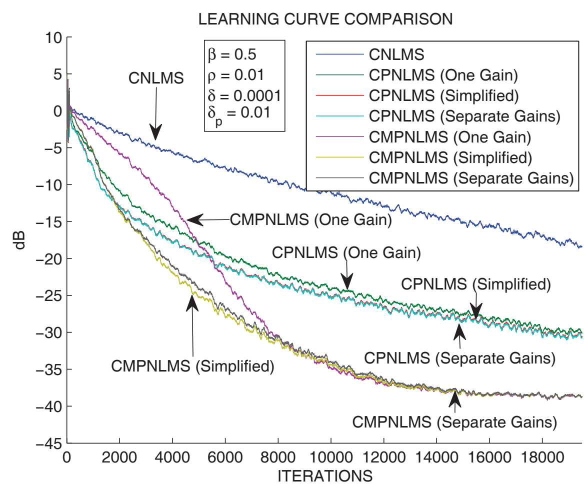

Fig. 4. CPtNLMS Learning Curve Comparison with Colored Input

cases. Of course the convergence rate is lower because of the colored input.

6. CONCLUSION

In this work we have introduced a family of complex PtNLMS algorithms which were derived by minimizing the second norm of the weighted difference between the estimated impulse response at the current time step and the next time step under the constraint that the adaptive filter output at the next time step equals the measured output at the current time. The choice of the stepsize control matrix is arbitrary during the derivation which allows the user to design algorithms to suit their needs. Additionally, it was shown through simulation that adapting the real and imaginary components of the estimated impulse response independently of each other results in increased MSE rate of convergence. Moreover, it can be noticed from the simulations that the performances of the complex algorithm with separate gains for real and imaginary parts and its simplified version are nearly indistinguishable.

7. REFERENCES

[1] B. Widrow, J. McCool, and M. Ball "The Complex LMS Algorithm," *Proceedings of the IEEE*, pp. 719-720, April 1975.

[2] D. Duttweiler, "Proportionate normalized least-mean-squares adaptation in echo cancellers," *IEEE Trans. Speech Audio Processing*, vol. 8, pp. 508-518, Sept. 2000.

[3] H. Deng and M. Doroslovački, "Improving convergence of the PNLMS algorithm for sparse impulse response identification," *IEEE Signal Processing Letters*, vol. 12, no. 3, pp. 181-184, Mar. 2005.

[4] S. Haykin, *Adaptive Filter Theory*, fourth edition, Prentice Hall, 2002.