Improved Quasi-Newton Adaptive-Filtering Algorithm

Md Zulfiquar Ali Bhotto, *Student Member, IEEE*, and Andreas Antoniou, *Life Fellow, IEEE*

***Abstract*—An improved quasi-Newton (QN) algorithm that performs data-selective adaptation is proposed whereby the weight vector and the inverse of the input-signal autocorrelation matrix are updated only when the *a priori* error exceeds a prespecified error bound. The proposed algorithm also incorporates an improved estimator of the inverse of the autocorrelation matrix. With these modifications, the proposed QN algorithm takes significantly fewer updates to converge and yields a reduced steady-state misalignment relative to a known QN algorithm proposed recently. These features of the proposed QN algorithm are demonstrated through extensive simulations. Simulations also show that the proposed QN algorithm, like the known QN algorithm, is quite robust with respect to roundoff errors introduced in fixed-point implementations.**

***Index Terms*—Adaptation algorithms, adaptive filters, convergence speed in adaptation algorithms, quasi-Newton algorithms, steady-state misalignment.**

I. Introduction

THE least-mean-squares (LMS) algorithm minimizes the Weiner-Hopf function iteratively by using the instantaneous values of the autocorrelation function of the input signal and the crosscorrelation function between the input and desired signals [1]. Due to its simplicity, the LMS algorithm is frequently used in current practice. However, when the input signal is highly colored or bandlimited, the LMS algorithm as well as other algorithms of the steepest-descent family converge slowly and the capability of such algorithms in tracking nonstationarities deteriorates. In such situations, more sophisticated algorithms that belong to the Newton family are preferred. However, the computational complexity of these algorithms is usually prohibitively large especially in real-time applications where low-cost digital hardware must be employed. Numerical instability is also a major issue in these algorithms. The conventional recursive least-squares (CRLS) algorithm converges much faster than algorithms of the steepest-descent family [1]. However, it can become unstable and if a large forgetting factor is chosen it can actually lose its tracking capability. The known quasi-Newton (KQN) algorithm reported in [2], [3] offers better numerical robustness whereas the LMS-Newton (LMSN) algorithms reported in [4] offer better convergence performance than the CRLS algorithm.

Two methods are available for the development of Newton-type adaptation algorithms: methods based on the direct solution of the normal equations of a least-squares problem (see Section II) and methods based on the orthogonal decomposition of the input-signal matrix. The CRLS, KQN, and the LMSN algorithms are based on the normal equations and the QR decomposition algorithm (QRD) is based on the orthogonal decomposition of the input-signal matrix [5]. The computational complexity of these algorithms is of order M^2 designated as $O\left(M^2\right)$. The fast QRD (FQRD) algorithms in [6]–[12] are actually efficient implementations of the QRD algorithm whose computational complexity is of $O(M)$. Fast RLS (FRLS) algorithms with computational complexities of $O(M)$ are also available in the literature for FIR adaptive filtering and autoregressive (AR) prediction [5], [13]–[16]. FRLS algorithms exploit the Toeplitz structure of the input-signal autocorrelation matrix. The fast QN (FQN) algorithm reported in [17], which has a computational complexity of $O(M)$, also exploits the Toeplitz structure of the autocorrelation matrix to reduce the computational complexity. The FRLS and FQRD algorithms suffer from numerical instability problems that are inherited from the numerical instability problems of the CRLS and QRD algorithms, respectively, and also the simplifications used to obtain these algorithms [8]. However, the FQRD algorithm reported in [8] offers numerically stable operation in low-precision implementations and in the absence of persistent excitation. The numerical instability problems associated with the CRLS algorithm are discussed in [18] where an upper bound on the relative precision to assure the BIBO stability of the CRLS algorithm in stationary and nonstationary environments is derived. Formulas for choosing the forgetting factor to avoid explosive divergence for a given precision in the CRLS algorithm are also given in [18]. However, these formulas were derived on the assumption that the input signal is persistently exciting. Furthermore, the input-signal statistics must be known *a priori* in order to use these formulas. Consequently, a prudent strategy for the derivation of fast Newton-type algorithms would be to start with a parent algorithm that is inherently stable. The numerical robustness of the quasi-Newton (QN) algorithm reported in [2], [3] is achieved by using a biased estimate of the autocorrelation matrix, which can reduce the tracking capability of the algorithm relative to that of the CRLS algorithm (see [19, p. 678]).

In this paper, we propose an improved version of the QN algorithm reported in [2], [3] that incorporates data-selective adaptation. The proposed QN (PQN) algorithm takes fewer weight updates to converge and yields a reduced steady-state misalign-

Manuscript received April 14, 2009; revised September 26, 2009; accepted November 11, 2009. Date of publication February 05, 2010; date of current version August 11, 2010. This work was supported by the Natural Sciences and Engineering Research Council of Canada. This paper was recommended by Associate Editor H. Johansson.

The authors are with the Department of Electrical and Computer Engineering, University of Victoria, Victoria, BC V8W 3P6, Canada (e-mail: zbhotto@ece.uvic.ca, aantoniou@ieee.org).

Digital Object Identifier 10.1109/TCSI.2009.2038567

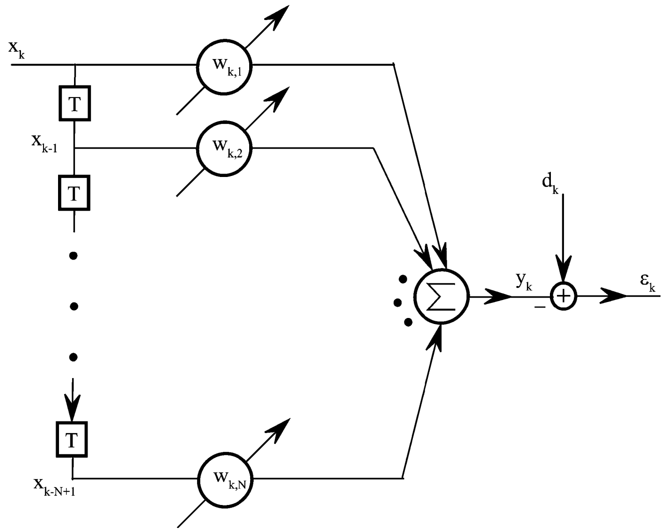

Fig. 1. Adaptive filter.

ment relative to the KQN algorithm in [2], [3]. These features of the new algorithm are demonstrated through MATLAB simulations in stationary and nonstationary environments. Simulations also show that the PQN algorithm, like the KQN algorithm, is quite robust with respect to roundoff errors in fixed-precision implementations.

The paper is organized as follows. In Section II, the proposed QN algorithm is described. Performance analysis is carried out in Section III. Simulation results for stationary and nonstationary environments are given in Section IV. Finally, conclusions are drawn in Section V.

II. Proposed Quasi-Newton Algorithm

The simplest and most commonly used configuration for adaptive filtering is the tapped delay-line structure illustrated in Fig. 1, which is essentially an FIR digital filter. Vectors $\boldsymbol{w}_k = [w_{k,1}\ w_{k,2}\ \cdots\ w_{k,N}]^T$ and $\boldsymbol{x}_k = [x_k\ x_{k-1}\ \cdots\ x_{k-N+1}]^T$ represent the weight and input-signal sequences at iteration k, respectively. The *a priori* error signal at iteration k is given by

$$e_k = d_k - \boldsymbol{w}_{k-1}^T \boldsymbol{x}_k \tag{1}$$

where d_k is the desired signal. Adaptation algorithms update the weight vector, $\boldsymbol{w}_k$, in such a way as to obtain the solution of the optimization problem

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad E\left[(d_k - \boldsymbol{w}^T \boldsymbol{x}_k)^2\right] \tag{2}$$

where E is the expectation operator. This is usually referred to as the Wiener solution. Adaptation algorithms use either the steepest-descent or the Newton direction. LMS-type algorithms use the steepest-descent direction and, therefore, their convergence rate depends on the nature of the objective function and spread of the eigenvalues of the Hessian matrix of the objective function. Most algorithms of the Newton family use the update formula

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + 2\mu_k e_k \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k \tag{3}$$

where μ_k is the step size and $\hat{\boldsymbol{R}}_{k-1}^{-1}$ is an estimate of the inverse of the autocorrelation matrix at iteration k. The *a posteriori* error at iteration k is defined as

$$\epsilon_k = d_k - \boldsymbol{w}_k^T \boldsymbol{x}_k. \tag{4}$$

In the KQN algorithm, μ_k is determined by minimizing $\xi_k = \epsilon_k^2$ with respect to μ_k in every iteration. In the PQN algorithm, data-selective weight adaptation is performed whereby μ_k is updated only when the magnitude of the *a priori* error e_k is greater than a prespecified error bound γ. In other words, whenever $|e_k| > \gamma$, μ_k is chosen to force the equality $|\epsilon_k| = \gamma$. The associated optimization problem can be stated as

$$\mu_k = \begin{cases} \underset{\mu_k}{\text{minimize}}\ |d_k - \boldsymbol{x}_k^T \boldsymbol{w}_k| - \gamma & \text{if } |e_k| > \gamma \\ 0 & \text{otherwise} \end{cases}. \tag{5}$$

Straightforward analysis leads to

$$\mu_k = \alpha_k \frac{1}{2\tau_k} \tag{6}$$

where $\tau_k = \boldsymbol{x}_k^T \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k$ and

$$\alpha_k = \begin{cases} 1 - \frac{\gamma}{|e_k|} & \text{if } |e_k| > \gamma \\ 0 & \text{otherwise} \end{cases}.$$

As in the KQN algorithm in [3], the PQN algorithm also uses the rank-one quasi-Newton updating formula given in Eq. (7.20) in [20] to deduce the update of the inverse of the autocorrelation matrix as

$$\hat{\boldsymbol{R}}_k^{-1} = \hat{\boldsymbol{R}}_{k-1}^{-1} + \frac{\left(\boldsymbol{\delta}_k - \hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{\rho}_k\right)\left(\boldsymbol{\delta}_k - \hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{\rho}_k\right)^T}{\left(\boldsymbol{\delta}_k - \hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{\rho}_k\right)^T \boldsymbol{\rho}_k}. \tag{7}$$

The estimator in (7) satisfies the classical QN hereditary condition and, therefore, the associated algorithm belongs to the QN family according to Fletcher's classification in [21]. For the adaptation of $\hat{\boldsymbol{R}}_k^{-1}$, $\boldsymbol{\delta}_k$ is chosen to be in proportion to ϵ_k along the Newton direction $\hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{x}_k$. Since the relation

$$\epsilon_k = \gamma \cdot \text{sign}(e_k) \tag{8}$$

holds true during each update, we use

$$\boldsymbol{\delta}_k = 2\gamma \cdot \text{sign}(e_k) \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k \tag{9}$$

and

$$\begin{aligned} \boldsymbol{\rho}_k &= \frac{\partial e_{k+1}^2}{\partial \boldsymbol{w}_k} - \frac{\partial e_k^2}{\partial \boldsymbol{w}_{k-1}} \\ &= \boldsymbol{g}_k - \boldsymbol{g}_{k-1}. \end{aligned} \tag{10}$$

Unfortunately, in the context of adaptive filtering, gradient $\boldsymbol{g}_k$ is not available at iteration k since it requires future data $\boldsymbol{x}_{k+1}$ and d_{k+1}. However, parameter $\boldsymbol{\rho}_k$ in (10) can be approximated as

$$\begin{aligned} \boldsymbol{\rho}_k &= \frac{\partial \epsilon_k^2}{\partial \boldsymbol{w}_k} - \frac{\partial e_k^2}{\partial \boldsymbol{w}_{k-1}} \\ &= \tilde{\boldsymbol{g}}_k - \boldsymbol{g}_{k-1} \end{aligned} \tag{11}$$

TABLE I
PROPOSED QN ALGORITHM.

```
Initialize $\boldsymbol{w}_0 = \boldsymbol{0}$ and $\hat{\boldsymbol{R}}_0^{-1} = \boldsymbol{I}$.
Choose $\gamma$.
Input $d_k$, $\boldsymbol{x}_k$ and compute
  $e_k = d_k - \boldsymbol{w}_{k-1}^T \boldsymbol{x}_k$
  if $|e_k| > \gamma$
    $\alpha_k = 1 - \frac{\gamma}{|e_k|}$
    $\boldsymbol{t}_k = \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k$
    $\tau_k = \boldsymbol{x}_k^T \boldsymbol{t}_k$
    $\boldsymbol{g}_k = \alpha_k \boldsymbol{t}_k$
    if $\tau_k < \varsigma$
      $\hat{\boldsymbol{R}}_k^{-1} = \boldsymbol{I}$
      $\tau_k = 0.5$
    else
      $\hat{\boldsymbol{R}}_k^{-1} = \hat{\boldsymbol{R}}_{k-1}^{-1} - \frac{\boldsymbol{g}_k \boldsymbol{t}_k^T}{\tau_k}$
    end
  $\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + \frac{1}{\tau_k} e_k \boldsymbol{g}_k$
  end
```

where e_k and ϵ_k are defined in (1) and (4), respectively. Substituting (1) and (8) in (11), we obtain

$$\boldsymbol{\rho}_k = 2 e_k \boldsymbol{x}_k. \tag{12}$$

Now substituting (9) and (12) in (7), we obtain an update formula for the inverse of the autocorrelation matrix as

$$\begin{aligned} \hat{\boldsymbol{R}}_k^{-1} &= \hat{\boldsymbol{R}}_{k-1}^{-1} - \left(1 - \frac{\gamma}{|e_k|}\right) \frac{\hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k \boldsymbol{x}_k^T \hat{\boldsymbol{R}}_{k-1}^{-1}}{\boldsymbol{x}_k^T \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k} \\ &= \hat{\boldsymbol{R}}_{k-1}^{-1} - \frac{\alpha_k}{\tau_k} \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k \boldsymbol{x}_k^T \hat{\boldsymbol{R}}_{k-1}^{-1}. \end{aligned} \tag{13}$$

The weight-vector update formula can be obtained by using (6) in (3) as

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + \frac{\alpha_k}{\tau_k} e_k \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k. \tag{14}$$

Updates are applied to $\boldsymbol{w}_{k-1}$ and $\hat{\boldsymbol{R}}_{k-1}^{-1}$ only if the *a priori* error exceeds the prespecified error bound γ as is done in the basic QN optimization algorithm (see [20, p. 184]). Otherwise, no updates are applied. The PQN algorithm can be implemented as detailed in Table I.

The crosscorrelation vector of the PQN algorithm can be defined as

$$\hat{\boldsymbol{p}}_k = \hat{\boldsymbol{p}}_{k-1} + \frac{\alpha_k}{\tau_{p,k}} d_k \boldsymbol{x}_k \tag{15}$$

where $\tau_{p,k} = (1 - \alpha_k)\tau_k$. If we apply the matrix inversion lemma given in [1], [22] to (13), we obtain the input-signal autocorrelation matrix as

$$\hat{\boldsymbol{R}}_k = \hat{\boldsymbol{R}}_{k-1} + \frac{\alpha_k}{\tau_{p,k}} \boldsymbol{x}_k \boldsymbol{x}_k^T. \tag{16}$$

Using (1) in (14), straightforward manipulation yields

$$\begin{aligned} \boldsymbol{w}_k =& \boldsymbol{w}_{k-1} + \frac{\alpha_k}{\tau_{p,k}} d_k \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k - \frac{\alpha_k}{\tau_k} \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{w}_{k-1} \\ & - \frac{\alpha_k^2}{\tau_{p,k}} d_k \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k. \end{aligned} \tag{17}$$

Using the normal equation

$$\hat{\boldsymbol{R}}_{k-1} \boldsymbol{w}_{k-1} = \boldsymbol{p}_{k-1}$$

in (17) and simplifying the expression obtained, we have

$$\boldsymbol{w}_k = \left(\hat{\boldsymbol{R}}_{k-1}^{-1} - \frac{\alpha_k}{\tau_k} \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k \boldsymbol{x}_k^T \hat{\boldsymbol{R}}_{k-1}^{-1}\right) \left(\hat{\boldsymbol{p}}_{k-1} + \frac{\alpha_k}{\tau_{p,k}} d_k \boldsymbol{x}_k\right). \tag{18}$$

Now using (13) and (15) in (18), we obtain

$$\hat{\boldsymbol{R}}_k \boldsymbol{w}_k = \hat{\boldsymbol{p}}_k \tag{19}$$

which comprises the normal equations of the objective function

$$J_{\boldsymbol{w}_k} = \sum_{i=1}^{k} \frac{\alpha_i}{\tau_{p,i}} \left(d_i - \boldsymbol{w}_k^T \boldsymbol{x}_i\right)^2 \tag{20}$$

where $\tau_{p,i} = (1 - \alpha_i) \boldsymbol{x}_i^T \hat{\boldsymbol{R}}_{i-1}^{-1} \boldsymbol{x}_i$. Hence the weight-vector update equation of the PQN algorithm in (14) minimizes the objective function in (20).

As can be seen, the objective function of the PQN algorithm is similar to that of the weighted LS or LMSN algorithm (see, (8), [4]). The updating formulas for the KQN algorithm are

$$\hat{\boldsymbol{R}}_k^{-1} = \hat{\boldsymbol{R}}_{k-1}^{-1} + \frac{(\mu_k - 1)}{\tau_k} \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k \boldsymbol{x}_k^T \hat{\boldsymbol{R}}_{k-1}^{-1} \tag{21}$$

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + 2\mu_k e_k \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k \tag{22}$$

where

$$\mu_k = \frac{1}{2\tau_k}. \tag{23}$$

The objective function of the KQN algorithm, on the other hand, assumes the form

$$J_{\boldsymbol{w}_k} = \sum_{i=1}^{k} (1 - \mu_i) \left(\frac{d_i - \mu_i \boldsymbol{x}_i^T \boldsymbol{w}_{i-1}}{1 - \mu_i} - \boldsymbol{x}_i^T \boldsymbol{w}_k\right)^2 + \frac{1}{2} \boldsymbol{w}_k^T \hat{\boldsymbol{R}}_0 \boldsymbol{w}_k \tag{24}$$

where $\mu_i = 1/(2\boldsymbol{x}_i^T \hat{\boldsymbol{R}}_{i-1} \boldsymbol{x}_i)$. It turns out that in the KQN algorithm the value of $\mu_k - 1$ in the estimator in (21) approaches zero, as can be verified by examining Fig. 4 in [3]. As a result, the adaptation of $\hat{\boldsymbol{R}}_k^{-1}$ will stop after a certain number of iterations regardless of the value of $||\boldsymbol{\delta}_k||$ whereas the basic QN optimization algorithm as reported in [20] suggests that the adaptation of $\hat{\boldsymbol{R}}_k^{-1}$ should continue until the value of $||\boldsymbol{\delta}_k||$ becomes sufficiently small. Consequently, the steady-state value of the misalignment and the speed of convergence will be affected.

An unbiased estimate of the inverse of the input-signal autocorrelation matrix cannot be obtained by using the rank-one update formula given in (7). However, the undesired consequences of using a biased estimate in the adaptation of the weight vector

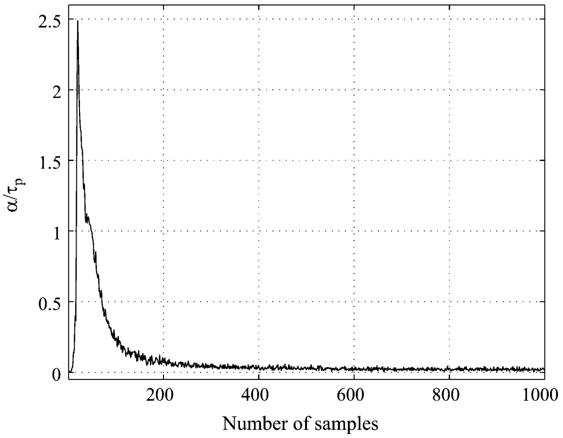

Fig. 2. Evolution of α/τ_p.

can be avoided by using a convergence factor μ in (3) [22]. The autocorrelation matrix in (16) can be expressed as

$$\hat{\boldsymbol{R}}_k = \hat{\boldsymbol{R}}_0 + \sum_{i=1}^{k} \frac{\alpha_i}{\tau_{p,i}} \boldsymbol{x}_i \boldsymbol{x}_i^T. \tag{25}$$

As can be seen, the update formula for $\hat{\boldsymbol{R}}_k$ is a weighted sum of the outer product $\boldsymbol{x}_i \boldsymbol{x}_i^T$ and the weights depend on the input-signal statistics and the *a priori* error signal. Taking the expectation of both sides in (25) and invoking the assumption that τ and $\boldsymbol{x}_i \boldsymbol{x}_i^T$ are statistically independent, we obtain

$$E[\hat{\boldsymbol{R}}_k] = \hat{\boldsymbol{R}}_0 + \boldsymbol{R}_{x,x} \sum_{i=1}^{k} E\left(\frac{\alpha_i}{\tau_{p,i}}\right). \tag{26}$$

An expression for the expectation of $\alpha_i/\tau_{p,i}$ is difficult to deduce but an approximate value can be obtained in terms of its time average based on simulation. In such an experiment the error bound can be chosen as $\gamma = \sqrt{5\sigma_v^2}$ where σ_v^2 is the variance of the measurement noise. The evolution of α/τ_p for a white Gaussian input signal with zero mean and unit variance assuming a variance of the measurement noise, σ_v^2, of 10^{-4} is illustrated in Fig. 2. This was determined by averaging the ensemble of α/τ_p in 100 runs in a 36th-order system identification application.

Note that since $\alpha_i = 0$ for $|e_i| < \gamma$ and $\tau_{p,i} > 0$ as $\hat{\boldsymbol{R}}_i^{-1}$ is positive definite (see Theorem 1 below), we have $\alpha_i/\tau_{p,i} = 0$ for $|e_i| < \gamma$. Since $\boldsymbol{g}_k = \alpha_k \boldsymbol{t}_k$, the value of $\boldsymbol{t}_k$ does not need to be specified if $|e_k| < \gamma$.

As can be seen in Fig. 2, the time average is a positive quantity and, therefore, on the average a significant improvement can be brought about in the estimate of $\hat{\boldsymbol{R}}_k$ with respect to $\hat{\boldsymbol{R}}_0$. The effect of using a biased estimate on the weight vector will, therefore, be less pronounced in the proposed algorithm as the quantity α_k in the step size μ_k in (6) approaches zero at steady state. Since $\alpha_i \subset [0, 1)$ and $\tau_{p,i} > 0$ for all i, the weights used in (20) and (25) are nonnegative. Therefore, if $\hat{\boldsymbol{R}}_0$ is positive definite it is straightforward to show that the estimate in (16), i.e., the Hessian of (20), would remain positive definite indefinitely. Consequently, the objective function in (20) would remain convex indefinitely.

The temporal behavior of α/τ_p in (20) can also be observed in Fig. 2. Since $\tau_{p,i} > 0$ and it is also bounded due to the fact that $\hat{\boldsymbol{R}}_k^{-1}$ is bounded (see Theorem 2 below), α/τ_p can approach zero only when $\alpha_i \approx 0$, i.e., when the solution of (20) is achieved. During transience, $\alpha_i \approx 1$ and during steady state, $\alpha_i \approx 0$; therefore, α/τ_p is large during transience and becomes small at steady state.

The stability of Newton-type algorithms depends on the positive definiteness of $\hat{\boldsymbol{R}}_k^{-1}$ [3], [4]. Furthermore, $\hat{\boldsymbol{R}}_k^{-1}$ must be bounded for a bounded input signal. Otherwise, the BIBO stability of the algorithm cannot be assured. The formula in (14) could also lead to a biased solution for an unbounded $\hat{\boldsymbol{R}}_k^{-1}$. Both of these requirements are satisfied in the PQN algorithm according to the following theorems.

Theorem 1: If $\hat{\boldsymbol{R}}_{k-1}^{-1}$ is a symmetric positive definite matrix, then $\hat{\boldsymbol{R}}_k^{-1}$ is also a symmetric positive definite matrix for all $k > 0$.

Proof: Since $\hat{\boldsymbol{R}}_{k-1}^{-1}$ is a real symmetric matrix, we can express $\hat{\boldsymbol{R}}_{k-1}^{-1}$ as

$$\begin{aligned} \hat{\boldsymbol{R}}_{k-1}^{-1} &= \boldsymbol{U}\Lambda\boldsymbol{U}^T \\ &= \left(\boldsymbol{U}\Lambda^{1/2}\boldsymbol{U}^T\right)\left(\boldsymbol{U}\Lambda^{1/2}\boldsymbol{U}^T\right) \\ &= \hat{\boldsymbol{R}}_{k-1}^{-1/2}\hat{\boldsymbol{R}}_{k-1}^{-1/2} \end{aligned} \tag{27}$$

where $\boldsymbol{U}$ is a unitary matrix such that $\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{U}\boldsymbol{U}^T = \boldsymbol{I}$. If we let $\boldsymbol{u} = \hat{\boldsymbol{R}}_{k-1}^{-1/2}\boldsymbol{x}_k$ and $\boldsymbol{v} = \hat{\boldsymbol{R}}_{k-1}^{-1/2}\boldsymbol{y}$, then for any nonzero vector $\boldsymbol{y} \in \Re^N$, the relation in (13) can be used to obtain

$$\begin{aligned} \boldsymbol{y}^T\hat{\boldsymbol{R}}_k^{-1}\boldsymbol{y} &= \boldsymbol{y}^T\hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{y} - \alpha_k \cdot \frac{\boldsymbol{y}^T\hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{x}_k\boldsymbol{x}_k^T\hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{y}}{\boldsymbol{x}_k^T\hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{x}_k} \\ &= \boldsymbol{y}^T\hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{y} - \alpha_k \cdot \frac{\left(\boldsymbol{y}^T\hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{x}_k\right)^2}{\boldsymbol{x}_k^T\hat{\boldsymbol{R}}_{k-1}^{-1}\boldsymbol{x}_k}. \end{aligned}$$

Substituting (27) and the definitions of $\boldsymbol{u}$ and $\boldsymbol{v}$ in the above equation, we obtain

$$\begin{aligned} \boldsymbol{y}^T\hat{\boldsymbol{R}}_k^{-1}\boldsymbol{y} &= \boldsymbol{v}^T\boldsymbol{v} - \alpha_k \cdot \frac{\left(\boldsymbol{v}^T\boldsymbol{u}\right)^2}{\boldsymbol{u}^T\boldsymbol{u}} \\ &= \|\boldsymbol{v}\|^2 - \alpha_k \cdot \frac{\|\boldsymbol{v}\|^2\|\boldsymbol{u}\|^2\cos^2\theta}{\|\boldsymbol{u}\|^2} \\ &= \|\boldsymbol{v}\|^2 - \alpha_k \cdot \|\boldsymbol{v}\|^2(0.5 + 0.5\cos 2\theta). \end{aligned}$$

Since $0 < \alpha_k < 1$, the lowest possible value of the right-hand side in the above equation will occur when $\theta = 0$ and, therefore,

$$\boldsymbol{y}^T\hat{\boldsymbol{R}}_k^{-1}\boldsymbol{y} > \|\boldsymbol{v}\|^2(1 - \alpha_k) > 0 \tag{28}$$

for any $\boldsymbol{y} \in \Re^{N\times 1}$. Hence the estimate in (13) is positive definite for all $k > 0$. The symmetry of $\hat{\boldsymbol{R}}_k^{-1}$ can be easily demonstrated by noting the symmetry of $\hat{\boldsymbol{R}}_{k-1}^{-1}$.

Theorem 2: The estimate of $\hat{\boldsymbol{R}}_k^{-1}$ in (13) is always bounded in the sense that the quadratic factor $\boldsymbol{x}_k^T\hat{\boldsymbol{R}}_k^{-1}\boldsymbol{x}_k$ is bounded provided that the input signal is bounded.

Proof: If we premultiply both sides in (13) by $\boldsymbol{x}_k^T$ and postmultiply them by $\boldsymbol{x}_k$, we obtain

$$\boldsymbol{x}_k^T \hat{\boldsymbol{R}}_k^{-1} \boldsymbol{x}_k = \boldsymbol{x}_k^T \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k (1 - \alpha_k). \tag{29}$$

Since $0 < (1 - \alpha_k) < 1$ holds true for each adaptation, we have

$$\boldsymbol{x}_k^T \hat{\boldsymbol{R}}_k^{-1} \boldsymbol{x}_k < \boldsymbol{x}_k^T \hat{\boldsymbol{R}}_{k-1}^{-1} \boldsymbol{x}_k. \tag{30}$$

Therefore, we conclude that if the input signal is bounded, then $\hat{\boldsymbol{R}}_k^{-1}$ is also bounded.

III. Analysis of the Proposed QN Algorithm

In this section, we examine the performance of the PQN algorithm in terms of the mean square-error (MSE) after initial convergence in stationary and nonstationary environments. The update formula for the weight vector in the PQN algorithm is similar to that of the LMS-Newton algorithm given in (29) in [4, p. 620]. The difference resides in the estimation of the inverse of the autocorrelation matrix and the *reduction factor*, q. The KQN algorithm uses $q = 1$ instead of a prespecified fixed reduction factor q and the PQN algorithm uses a variable reduction factor $q = \alpha_k = [0, 1)$. However, the steady-state MSE of the PQN algorithm depends on the steady-state value of $\alpha_k = p$, not on its transient value. As reported in [4], the steady-state mean-square error given in Eqs. (40) and (46) of [4] for stationary and nonstationary environments, respectively, is independent of the way the inverse of the autocorrelation matrix is estimated and hence it will not be different for other Newton-type algorithms as long as (1) they use a weight-vector update equation of the type given in [4], (2) they use an approximate Newton direction, and (3) the assumptions made in [4] hold true. This conclusion is confirmed in [p. 931, [3]] where the expression for the steady-state MSE in the KQN algorithm is shown to be identical with that of the LMSN algorithms. As can be verified, using $q = 1$ in Eq. (46) of [4], (22) of [3] can be obtained. Since the PQN algorithm follows an approximate Newton direction, employs a similar step size, and uses the same update equations for the weight vector, the formulas for the excess mean-square error are the same as those in Eqs. (40) and (46) in [4]. For stationary environments, the excess mean-square error for the PQN algorithm is given by

$$J_{ex,PQN} = \frac{p}{2-p} J_{\min} \tag{31}$$

where $J_{\min}$ is the minimum mean-square error and p is the value of α_k as $k \to \infty$. For the case of the KQN algorithm, this becomes $J_{ex,QN} = J_{\min}$. As $p \ll 1$ at steady state, we have $J_{ex,PQN} < J_{ex,KQN}$. In addition, we have $J_{ex,PQN} < J_{ex,LMS-Newton}$ for any prespecified q.

If the weights of the unknown plant change according to the update formula

$$\boldsymbol{w}_{o,k} = \boldsymbol{w}_{o,k-1} + \boldsymbol{\nu}_k \tag{32}$$

the excess mean-square error is given by

$$J_{ex,PQN} = \frac{1}{2-p} \left(p J_{\min} + \frac{\sigma_x^2 \sigma_\nu^2 M^2}{p} \right) \tag{33}$$

where σ_x^2 is the variance of the input signal and σ_ν^2 is the variance of the elements of $\boldsymbol{\nu}$. As can be seen, the second term in the parenthesis is inversely proportional to p and, therefore, α_k should not be allowed to become too small.

The optimal value of the reduction factor, q_o, that minimizes the excess mean-square error is given in Eq. (47) of [4]. As can be easily verified, it is difficult to determine the optimal reduction factor as parameter a in Eq. (47) of [4] is unknown *a priori*. Since the derivation of q_o involves certain assumptions, there is no guarantee that the minimum excess mean-square error will be obtained with q_o [4]. To circumvent these problems, a small positive constant ϵ can be added to α_k to be used in the weight update formula for nonstationary environments so that $\alpha_{k\to\infty} \approx \epsilon$.

A numerical value for p is difficult to obtain if the input signal is colored. However, for a white Gaussian input signal an approximate range for p can be obtained as

$$2Q\left(\frac{\gamma}{\sigma_v}\right) \le p \le 2Q\left(\frac{\gamma}{\sqrt{2\sigma_v^2 + \gamma^2}}\right) \tag{34}$$

where σ_v^2 is the variance of the noise signal added to the desired signal and $Q(\cdot)$ is the complementary Gaussian cumulative distribution function [23]. The maximum reduction in the number of weight updates can be obtained by using the Chebyshev inequality

$$Pr(|e_k| < \gamma)_{\max} = 1 - Pr(|e_k| > \gamma) = 1 - \frac{\sigma_v^2}{\gamma^2} \tag{35}$$

where σ_v^2 is the minimum value of σ_e^2 and γ^2 is chosen as an integer multiple of σ_v^2, i.e., $\gamma^2 = k\sigma_v^2$ with $k = 1, 2, \ldots, L$.

The number of updates, convergence speed, and steady-state MSE depend on the value of parameter γ. From (35), a larger value of γ would reduce the number of updates. From (34), on the other hand, we note that a larger γ would reduce p, and according to (31) a reduced steady-state misalignment would be achieved in stationary environments. However, in such a case the convergence speed of the algorithm would be compromised. A smaller value of γ, on the other hand, would increase the number of updates and p and, therefore, an increased steady-state misalignment would result in the case of stationary environments. The convergence speed of the algorithm in this case would be improved. Similar conclusions about the influence of γ on the number of updates, convergence speed, and steady-state MSE were also drawn in [23]. However, such conclusions cannot be deduced for nonstationary environments as the relation in (33) is nonlinear. In nonstationary environments, a reduced error bound would improve the tracking capability of the algorithm because the algorithm would continue to carry out updates after convergence. Experimentation has shown that good performance can be achieved in stationary and nonstationary environments by choosing integer k in γ in the range of 1 to 5 in the first case and 1 to 3 in the second case.

As far as stability is concerned, the proposed algorithm is inherently stable as the *a posteriori* error is forced to be equal to the prespecified error bound, γ, whenever an update is made. A rough approximation of the variance of the measurement noise would be enough to choose the error bound. In certain engineering applications, the measurement noise has an upper bound

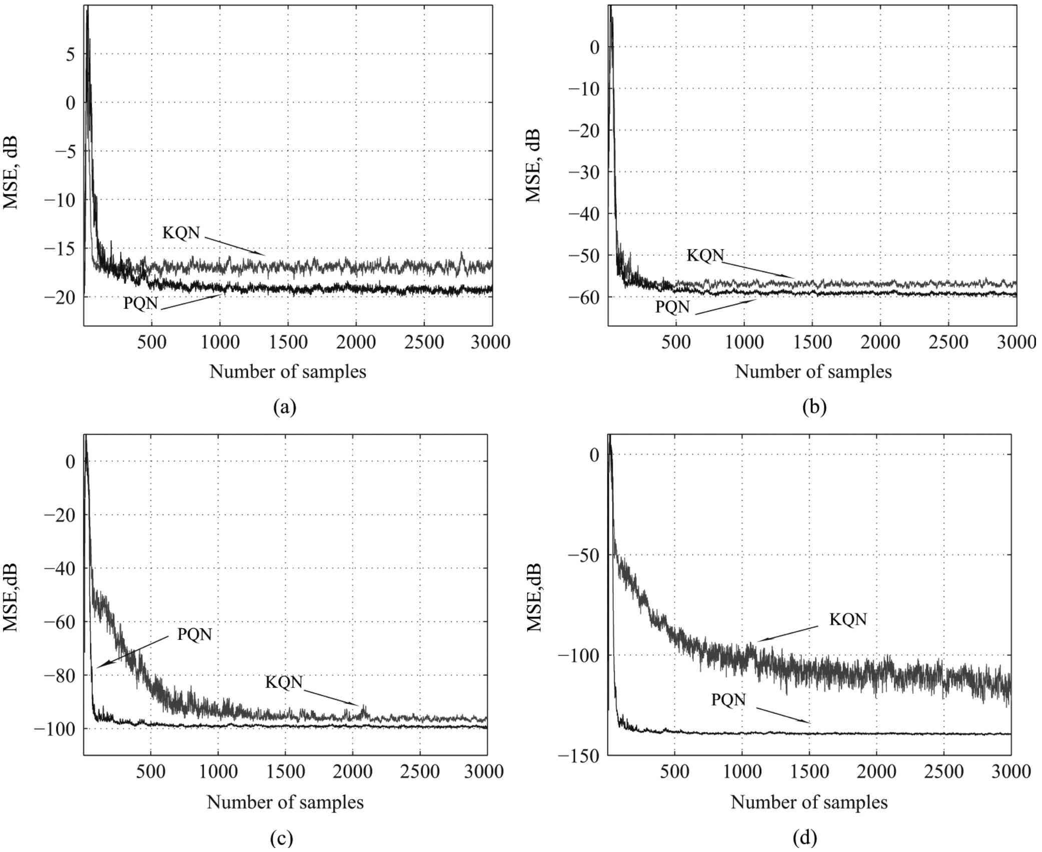

Fig. 3. Learning curves, stationary environment (a) $\mathrm{SNR} = 20\ \mathrm{dB}$ (b) $\mathrm{SNR} = 60\ \mathrm{dB}$ (c) $\mathrm{SNR} = 100\ \mathrm{dB}$ (d) $\mathrm{SNR} = 140\ \mathrm{dB}$.

TABLE II
COMPARISON OF PROPOSED WITH KNOWN QN ALGORITHM.

SNR	20 dB		60 dB		100 dB		140 dB	
Algorithm	PQN	KQN	PQN	KQN	PQN	KQN	PQN	KQN
Iterations to converge	120	120	130	130	120	700	100	3000
Weight updates	207	3000	229	3000	225	3000	228	3000
Reduction, %	93	-	92	-	92	-	92	-
Steady-state misalignment, dB	-19.3	-16.9	-59.3	-56.9	-99.3	-96.6	-139.4	-114.8

[24] and in such applications the PQN algorithm can be readily applied.

IV. Simulation Results

In order to evaluate the performance of the proposed QN algorithm several experiments were carried out as detailed below.

In the first experiment, an adaptive filter was used to identify a 36th-order lowpass FIR filter with a cutoff frequency of $0.3\omega_s$, where ω_s is the sampling frequency, using normalized coefficients to assure that the total power is unity. The input signal was a sinusoid of amplitude 1 and frequency of $\omega_s/16$ and was contaminated by a Gaussian noise signal of zero mean and variance 0.1. The contaminating Gaussian noise signal was colored using a 10th-order lowpass FIR filter with a cutoff frequency of $0.5\omega_s$. A sinusoidal signal was chosen because it causes the input signal to be severely ill-conditioned. With such a system identification problem, the convergence behavior of Newton-type algorithms can be better understood as their convergence speed would be low enough to facilitate comparison. The measurement noise added to the desired signal was white Gaussian with zero mean and variance of 10^{-2}, 10^{-6}, 10^{-10}, and 10^{-14} to achieve signal-to-noise ratios (SNRs) of 20, 60, 100, and 140 dB, respectively. The prespecified error bound was chosen as $\gamma = \sqrt{5\sigma_v^2}$ where σ_v^2 is the variance of the measurement noise. The initial weight vector was assumed to be the zero vector and the estimate of the inverse of the autocorrelation matrix was assumed to be the identity matrix in all experiments in the PQN as well as the KQN algorithms. The tolerance factor ς used in the fixed-point implementations was $\varsigma = 10^{-3}$ [2]. The learning curves obtained for different SNRs from 1000 independent trials by using the PQN and the KQN algorithms in a stationary environment are illustrated Figs. 3(a)–3(d). The number of iterations required for convergence, the steady-state misalignment, the number of weight updates required by the PQN and KQN algorithms in the above experiment in 3000 iterations, and the reductions in the number of updates achieved are given in Table II.

The second experiment was identical to the first experiment except that a nonstationarity was introduced in the filter taps according to the first-order Markov model

$$\boldsymbol{h}_k = \boldsymbol{h}_{k-1} + \boldsymbol{\nu}_k \tag{36}$$

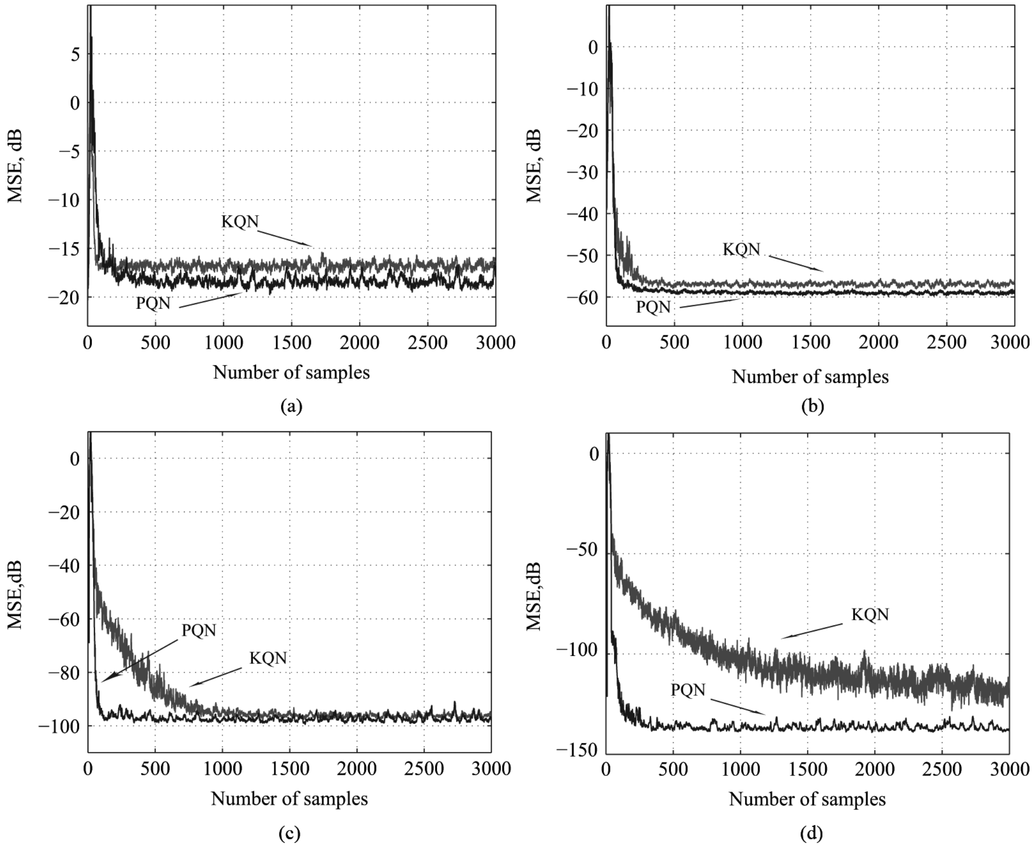

Fig. 4. Learning curves, nonstationary environment (a) $\mathrm{SNR} = 20\ \mathrm{dB}$ (b) $\mathrm{SNR} = 60\ \mathrm{dB}$. (c) $\mathrm{SNR} = 100\ \mathrm{dB}$ (d) $\mathrm{SNR} = 140\ \mathrm{dB}$.

TABLE III
COMPARISON OF PROPOSED WITH KNOWN QN ALGORITHM.

SNR	20 dB		60 dB		100 dB		140 dB	
Algorithm	PQN	KQN	PQN	KQN	PQN	KQN	PQN	KQN
Iterations to converge	120	120	130	200	130	860	200	3000
Weight updates	505	3000	450	3000	685	3000	874	3000
Reduction, %	83	-	85	-	77	-	71	-
Steady-state misalignment, dB	-18.4	-16.7	-59.0	-56.8	-97.3	-96.0	-137.5	-118.3

where the entries of $\boldsymbol{\nu}$ were the samples of a white Gaussian noise sequence with zero mean and variance equal to 10^{-8}, 10^{-12}, 10^{-14}, and 10^{-18}. The prespecified error bound in nonstationary environments was chosen as $\gamma = \sqrt{3\sigma_v^2}$. With a smaller error bound, the number of adaptations is increased and, therefore, the tracking of the changes in $\boldsymbol{h}_k$ as given in (36) improves after reaching steady state. The learning curves for different SNRs obtained from 1000 independent trials by using the PQN and KQN algorithms in a nonstationary environment are illustrated Figs. 4(a)–4(d). The number of iterations to converge, the steady-state misalignment, and the number of weight updates required by the PQN and KQN algorithms in 3000 iterations and the reductions achieved are given in Table III. As can be seen in Figs. 3 and 4 and Tables II and III, the PQN algorithm yields reduced misalignment while requiring fewer iterations to converge than the KQN algorithm for stationary and nonstationary environments. As in the KQN algorithm, the learning curves at steady-state in the PQN algorithm are not noisy. The improvement in the steady-state misalignment becomes more prominent for high SNRs in the PQN algorithm because in the KQN algorithm adaptation of the inverse of the autocorrelation matrix stops prior to reaching steady state. Tables II and III also show that the required numbers of weight updates in the PQN algorithm are only a fraction of those required by the KQN algorithm.

In the third and fourth experiments, we verified the formulas in (31) and (33) for the excess MSE for different system orders and different values of the error bound. The input signal was white Gaussian noise with zero mean and variance 0.1. The limiting values of p for the error bound $\gamma = \sqrt{k\sigma_v^2}$ for $k = 1, \ldots, 10$, were obtained using (34). The results presented in Table IV are the outcome of an ensemble of 100 runs where $p_{\min}$ and $p_{\max}$ are the limiting values of p and σ_v^2 and M denote the variance of the measurement noise and system order, respectively. As can be seen in Table IV, the excess MSE obtained form simulation lies within the range of the excess MSE obtained by using $p_{\min}$ and $p_{\max}$ in (31) for stationary environments as expected. In addition, the excess MSE reduces as the error bound is increased as discussed in the Section III.

The fourth experiment was the same as the third except that a nonstationarity was introduced as in the second experiment. The results obtained are given in Table V where σ_ν^2 is the variance of

TABLE IV
EXCESS MSE IN dB IN PROPOSED QN ALGORITHM.

$M = 36,\ \sigma_v^2 = 10^{-4}$

Error bound	Eq. (31)	Simulation	Eq. (31)
k	p_{min}		p_{max}
1	−39.2	−38.4	−38.5
2	−39.6	−39.0	−38.8
3	−39.8	−39.2	−38.9
4	−39.9	−39.5	−38.9
5	−39.9	−39.6	−39.0
6	−39.9	−39.7	−39.0
7	−39.9	−39.6	−39.0
8	−39.9	−39.5	−39.1
9	−39.9	−39.4	−39.1
10	−39.9	−39.2	−39.1

$M = 36,\ \sigma_v^2 = 10^{-8}$

Error bound	Eq. (31)	Simulation	Eq. (31)
k	p_{min}		p_{max}
1	−79.2	−78.2	−78.5
2	−79.6	−78.7	−78.8
3	−79.8	−79.0	−78.9
4	−79.9	−79.2	−78.9
5	−79.9	−79.3	−79.0
6	−79.9	−79.5	−79.0
7	−79.9	−79.5	−79.0
8	−79.9	−79.4	−79.1
9	−79.9	−79.2	−79.1
10	−79.9	−79.0	−79.1

$M = 56,\ \sigma_v^2 = 10^{-4}$

Error bound	Eq. (31)	Simulation	Eq. (31)
k	p_{min}		p_{max}
1	−39.2	−38.4	−38.5
2	−39.6	−38.9	−38.8
3	−39.8	−39.1	−38.9
4	−39.9	−39.2	−38.9
5	−39.9	−39.2	−39.0
6	−39.9	−39.2	−39.0
7	−39.9	−39.1	−39.0
8	−39.9	−39.1	−39.1
9	−39.9	−38.9	−39.1
10	−39.9	−38.7	−39.1

$M = 56,\ \sigma_v^2 = 10^{-8}$

Error bound	Eq. (31)	Simulation	Eq. (31)
k	p_{min}		p_{max}
1	−79.2	−78.2	−78.5
2	−79.6	−78.8	−78.8
3	−79.8	−79.1	−78.9
4	−79.9	−79.3	−78.9
5	−79.9	−79.4	−79.0
6	−79.9	−79.5	−79.0
7	−79.9	−79.4	−79.0
8	−79.9	−79.2	−79.1
9	−79.9	−79.0	−79.1
10	−79.9	−78.8	−79.1

TABLE V
EXCESS MSE IN dB IN PROPOSED QN ALGORITHM.

$M = 36,\ \sigma_v^2 = 10^{-4},\ \sigma_\nu^2 = 10^{-8}$

Error bound	Eq. (33)	Simulation	Eq. (33)
k	p_{min}		p_{max}
1	−37.3	−37.8	−37.6
2	−37.0	−38.1	−37.5
3	−36.8	−38.1	−37.5
4	−36.8	−38.1	−37.4
5	−36.7	−38.0	−37.4
6	−36.7	−37.8	−37.4
7	−36.7	−37.6	−37.4
8	−36.7	−37.4	−37.3
9	−36.7	−37.2	−37.3
10	−36.7	−37.0	−37.3

$M = 36,\ \sigma_v^2 = 10^{-8},\ \sigma_\nu^2 = 10^{-14}$

Error bound	Eq. (33)	Simulation	Eq. (33)
k	p_{min}		p_{max}
1	−77.6	−78.2	−78.0
2	−77.3	−78.7	−77.9
3	−77.1	−78.9	−77.8
4	−77.0	−79.1	−77.8
5	−77.0	−79.2	−77.7
6	−77.0	−79.3	−77.7
7	−77.0	−79.3	−77.7
8	−76.9	−79.3	−77.7
9	−76.9	−79.1	−77.7
10	−76.9	−79.0	−77.7

$M = 56,\ \sigma_v^2 = 10^{-4},\ \sigma_\nu^2 = 10^{-8}$

Error bound	Eq. (33)	Simulation	Eq. (33)
k	p_{min}		p_{max}
1	−36.8	−37.1	−37.2
2	−36.6	−37.1	−37.1
3	−36.5	−37.1	−37.0
4	−36.4	−37.0	−37.0
5	−36.4	−36.9	−37.0
6	−36.3	−36.7	−36.9
7	−36.3	−36.7	−36.9
8	−36.3	−36.6	−36.9
9	−36.3	−36.2	−36.9
10	−36.3	−36.0	−36.9

$M = 56,\ \sigma_v^2 = 10^{-8},\ \sigma_\nu^2 = 10^{-14}$

Error bound	Eq. (33)	Simulation	Eq. (33)
k	p_{min}		p_{max}
1	−77.6	−78.3	−78.0
2	−77.3	−78.8	−77.9
3	−77.1	−79.0	−77.8
4	−77.0	−79.2	−77.8
5	−77.0	−79.2	−77.7
6	−77.0	−79.3	−77.7
7	−77.0	−79.2	−77.7
8	−76.9	−79.1	−77.7
9	−76.9	−78.9	−77.7
10	−76.9	−78.7	−77.7

the Gaussian noise added to the weight vector. In nonstationary environments, the excess MSE obtained by simulation may not be within the range of the excess MSE obtained by using (33) since this formula remains nonlinear at steady state. However, the simulation results given in Table V are very close to the theoretical results.

The last and final experiment was carried out to demonstrate the robustness of the proposed algorithm in the case of a fixed-point implementation. The system to be identified was the same as that used in the first experiment. Here the input signal was a zero-mean white Gaussian noise with a variance of unity and was colored using an 10th-order lowpass FIR filter with a cutoff frequency of $0.5\omega_s$. The error bound was chosen as $\gamma = \sqrt{5\sigma_v^2}$ where σ_v^2 is the variance of the measurement noise. Fixed-point arithmetic was assumed using a word length of 20 bits with no scaling or rescuing procedures and overflow was handled using saturation arithmetic. The error signal, error bound, and the desired signal were quantized and the learning curves were not smoothed. The learning curves obtained by using the PQN and KQN algorithms in 100 trials with $\sigma_v^2 = 10^{-4}$ and 10^{-8}, and SNRs of 40 and 80 dB are illustrated in Fig. 5. These results are consistent with the results obtained with floating-point

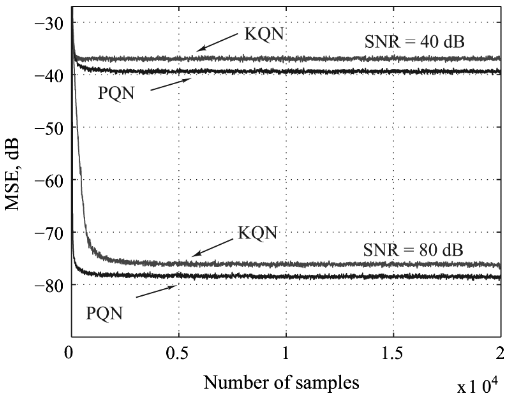

Fig. 5. MSE for a fixed-point implementation.

arithmetic. Furthermore, Fig. 5 shows that when implemented in fixed-point arithmetic the PQN algorithm is as robust as the KQN algorithm.

V. Conclusions

An improved QN algorithm for adaptive filtering that incorporates data selective adaptation for the weight vector and the inverse of the autocorrelation matrix has been proposed. The proposed algorithm was developed on the basis of the framework of classical QN optimization algorithms and in this way an improved estimator of the autocorrelation matrix was deduced. Analysis has shown that the PQN algorithm should perform better than the KQN algorithm in terms of convergence speed and steady-state misalignment. This expectation has been substantiated by simulations which have shown that the proposed algorithm requires fewer iterations to converge, fewer weight updates, and yields reduced steady-state misalignment when compared with the KQN algorithm in stationary as well as nonstationary environments. In addition, analytical results obtained by using closed-form formulas for the steady-state misalignment agree well with those obtained by simulations. The PQN algorithm was also found to be as robust as the KQN algorithm in the case of fixed-point implementations.

VI. Acknowledgment

The authors are grateful to the Natural Sciences and Engineering Research Council of Canada for supporting this work.

References

[1] A. H. Sayed, *Fundamentals of Adaptive Filtering*. New Jersey: Wiley, 2003.

[2] M. L. R. de Campos and A. Antoniou, “A robust quasi-Newton adaptive filtering algorithm,” in *IEEE Int. Symp. Circuits and Syst.*, Nov. 1994, pp. 229–232.

[3] M. L. R. de Campos and A. Antoniou, “A new quasi-Newton adaptive filtering algorithm,” *IEEE Trans. Circuits and Syst.*, vol. 44, no. 11, pp. 924–934, Nov. 1997.

[4] P. S. R. Diniz, M. L. R. de Campos, and A. Antoniou, “Analysis of LMS-Newton adaptive adaptive filtering algorithms with variable convergence factor,” *IEEE Trans. Signal Process.*, vol. 43, pp. 617–627, Mar. 1995.

[5] N. Kalouptsidis and S. Theodoridis, *Efficient System Identificaiton and Signal Processing Algorithms*. Englewood cliffs, NJ: Prentice-Hall, 1993.

[6] J. M. Cioffi, “The fast adaptive rotor’s RLS algorithm,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 38, no. 4, pp. 631–651, Apr. 1990.

[7] I. K. Proudler, J. G. McWhirter, and T. J. Shepherd, “Fast QRD-based algorithms for least squares linear prediction,” in *Proc. IMA Conf. Math. Signal Process.*, Warwick, England, Dec. 1998.

[8] P. A. Regalia and M. G. Bellanger, “On the duality between fast QR methods and lattice methods in least squares adaptive filtering,” *IEEE Trans. Signal Process.*, vol. 39, no. 4, pp. 879–892, Apr. 1991.

[9] D. Slock, “Reconsiling fast RLS lattice and QR algorithms,” in *IEEE Intern. Conf. Acoust., Speech, Signal Process.*, Apr. 1990, pp. 1591–1594.

[10] B. Yang and J. F. Bohme, “Rotation-based RLS algorithms: Unified derivations, numerical properties and parallel implementation,” *IEEE Trans. Signal Process.*, vol. 40, no. 5, pp. 1151–1167, May 1992.

[11] S. T. Alexander and A. Ghirnikar, “A method for recursive least squares filtering based upon an inverse QR decomposition,” *IEEE Trans. Signal Process.*, vol. 41, no. 1, pp. 20–30, Jan. 1993.

[12] A. A. Rontogiannis and S. Theodoridis, “New fast QR least squares adaptive algorithms,” in *IEEE Intern. Conf. on Acoust., Speech, and Signal Process.*, May 1995, pp. 1412–1415.

[13] L. Jung, M. Morf, and D. Falconer, “Fast calculations of gain matrices for recursive estimation schemes,” *Intern. Journal Contr.*, vol. 27, pp. 1–19, Jan. 1978.

[14] G. Carayannis, D. Manolakis, and N. Kalouptsidis, “A fast sequential algorithm for least-squares filtering and prediction,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 31, no. 6, pp. 1394–1402, Dec. 1983.

[15] J. Coiffi and T. Kailath, “Fast recursive LS tansversal filters for adaptive processing,” *IEEE Trans. Acoust., Speech, Signal Process.*, vol. 32, no. 4, pp. 304–337, Apr. 1984.

[16] G. Carayannis, D. Manolakis, and N. Kalouptsidis, “A unified view of parametric processing algorithms for prewindowed signals,” *Signal Process.*, vol. 10, pp. 335–368, 1986.

[17] D. F. Marshall and W. K. Jenkins, “A fast quasi-Newton adaptive filtering algorithm,” *IEEE Trans. Signal Process.*, vol. 7, no. 40, pp. 1652–1662, Jul. 1992.

[18] A. P. Liavas and P. A. Regalia, “On the numerical stability and accuracy of the conventional recursive least squares algorithm,” *IEEE Trans. Signal Process.*, vol. 47, no. 1, pp. 88–96, Jan. 1999.

[19] Y. Zou and S. C. Chan, “A robust quasi-newtion adaptive filtering algorithm for impulse noise suppression,” in *IEEE Intern. Symp. Circuits Syst.*, May 2001, pp. 677–680.

[20] A. Antoniou and W. S. Lu, *Pracitcal Optimization*. New York, USA: Springer, 2007.

[21] R. Fletcher, *Practical Methods of Optimization*. New York: Wiley, 1987.

[22] P. S. R. Diniz, *Adaptive Filtering: Algorithms and Practical Implementation*, 3rd ed. NewYork: Springer, 2008.

[23] P. S. R. Diniz and S. Werner, “Set-membership binormalized datareusing LMS algorithms,” *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 124–134, 2003.

[24] B. Peterson and K. Narendra, “Bounded error adaptive control,” *IEEE Trans. Automatic Contol*, vol. 27, no. 6, pp. 1161–1168, Dec. 1982.

Md Zulfiquar Ali Bhotto (SM’09) obtained the B.Sc. in electrical and electronic engineering from the Department of Electrical and Electronic Engineering of Rajshahi University of Engineering and Technology, Bangladesh in 2002. He worked as a lecturer and assistant professor in the Rajshahi University of Engineering and Technology from 2003 to 2007. Currently, he is pursuing a Ph.D. degree in the Department of Electrical and Computer Engineering at the University of Victoria, Victoria, Canada. He was awarded a University of Victoria fellowship for 2007–2008.

Andreas Antoniou (M'69-SM'79-F'82-LF'04) received the B.Sc.(Eng.) and Ph.D. degrees in electrical engineering from the University of London in 1963 and 1966, respectively, and is a Fellow of the IET and IEEE. He taught at Concordia University from 1970 to 1983, was the founding Chair of the Department of Electrical and Computer Engineering, University of Victoria, B.C., Canada, from 1983 to 1990, and is now Professor Emeritus. His teaching and research interests are in the area of digital signal processing. He is the author of Digital Signal Processing: Signals, Systems, and Filters, McGraw-Hill, 2005 and the co-author with Wu-Sheng Lu of Practical Optimization: Algorithms and Engineering Applications, Springer, 2007.

Dr. Antoniou served as Associate/Chief Editor for IEEE Transactions on Circuits and Systems (CAS) from 1983 to 1987, as a Distinguished Lecturer of the IEEE Signal Processing and the Circuits and Systems Societies during 2003–2004 and 2006–2007, respectively, and as General Chair of the 2004 International Symposium on Circuits and Systems.

He was awarded the CAS Golden Jubilee Medal by the IEEE Circuits and Systems Society, the B.C. Science Council Chairman's Award for Career Achievement for 2000, the Doctor Honoris Causa degree by the National Technical University, Athens, Greece, in 2002, the IEEE Circuits and Systems Society Technical Achievement Award for 2005, the 2008 IEEE Canada Outstanding Engineering Educator Silver Medal, and the IEEE Circuits and Systems Society Education Award for 2009.