Robust Quasi-Newton Adaptive Filtering Algorithms

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

***Abstract*—Two robust quasi-Newton (QN) adaptive filtering algorithms that perform well in impulsive-noise environments are proposed. The new algorithms use an improved estimate of the inverse of the autocorrelation matrix and an improved weight-vector update equation, which lead to improved speed of convergence and steady-state misalignment relative to those achieved in the known QN algorithms. A stability analysis shows that the proposed algorithms are asymptotically stable. The proposed algorithms perform data-selective adaptation, which significantly reduces the amount of computation required. Simulation results presented demonstrate the attractive features of the proposed algorithms.**

***Index Terms*—Adaptive filters, impulsive noise in adaptive filters, quasi-Newton algorithms, robust adaptation algorithms.**

I. Introduction

KNOWN approaches for improving the performance of adaptive filters in impulsive-noise environments involve the use of nonlinear clipping [1], [2], robust statistics [3]–[5], or order statistics. The common step in the adaptive filters reported in [1]–[5] is to detect the presence of impulsive noise by comparing the magnitude of the error signal with the value of a threshold parameter, which is a scalar multiple of the variance of the error signal, and then either stop or reduce the learning rate of the adaptive filter. In [1], the variance of the error signal is estimated by averaging the square of its instantaneous values, but this approach is not robust with respect to impulsive noise. In [2]–[5], improved robustness is achieved by estimating the variance of the error signal using the median absolute deviation [6].

The adaptation algorithms in [1] and [3] use the Huber mixed-norm M-estimate objective function [6], and the algorithms in [4] and [5] use the Hampel three-part redescending M-estimate objective function [6]. The Huber function uses the L_1 norm to measure the amplitude of the error signal when the absolute error is larger than the threshold. The Hampel function, on the other hand, assigns a constant value to the error signal when the absolute error becomes larger than the threshold. Algorithms based on the Huber and Hampel functions offer similar performance. The nonlinear recursive least-squares (NRLS) algorithm in [2] uses nonlinear clipping to control the learning rate and offers better performance in impulsive-noise environments than the conventional RLS algorithm. The recursive least-mean (RLM) algorithm reported in [4] offers faster convergence and better robustness than the NRLS algorithm in impulsive-noise environments. The quasi-Newton algorithm (QN) in [7] is not robust against impulsive noise. Simulation results in [5] show that the recursive QN (RQN) algorithm offers faster convergence and improved robustness in impulsive-noise environments relative to the QN algorithm in [7]. Algorithms of the Newton family such as those in [2], [4], [5], and [7] converge much faster than algorithms of the steepest-descent family [8]. The RLS, RLM, and RQN algorithms exponentially forget past input signal vectors in the estimate of the autocorrelation matrix, and therefore, the positive definiteness of the inverse of the autocorrelation matrix can be lost [9], [10]. As a result, this type of adaptive filter can become unstable in finite-precision implementations. This form of explosive divergence in RLS adaptive filters is well documented in the literature, and ways and conditions to stabilize these filters are addressed in [9] and [10]. The QN algorithm reported in [7] is shown to be robust in terms of explosive divergence, as compared with RLS adaptive filters.

In this brief, we propose two new robust QN algorithms that perform data-selective adaptation in updating the inverse of the autocorrelation matrix and the weight vector. The new algorithms are essentially enhancements of the algorithms we reported in [16] for applications that entail impulsive noise. A stability analysis shows that the proposed algorithms are asymptotically stable. Furthermore, simulation results show that the proposed algorithms offer improved performance relative to two known QN (KQN) algorithms with respect to robustness, steady-state misalignment, computational efficiency, and tracking capability.

This brief is organized as follows. In Section II, the proposed robust QN algorithms are described. In Section III, stability issues of the algorithms are discussed, and in Section IV, some practical issues concerning the implementation of the proposed algorithms are examined. Simulation results are presented in Section V, and conclusions are drawn in Section VI.

II. Proposed Robust QN Algorithms

Two slightly different robust QN algorithms are possible, one using a fixed threshold and the other using a variable threshold, as will now be demonstrated.

A. RQN Algorithm with a Fixed Threshold

The objective of the proposed adaptation algorithm is to generate a series of weight vectors that would eventually solve the optimization problem

$$\underset{\boldsymbol{w}}{\text{minimize}} \quad E\left[(d_k - \boldsymbol{w}^T \boldsymbol{x}_k)^2\right] \tag{1}$$

Manuscript received October 1, 2010; revised December 31, 2010 and March 16, 2011; accepted May 14, 2011. Date of publication July 22, 2011; date of current version August 17, 2011. This work was supported by the Natural Sciences and Engineering Research Council of Canada. This paper was recommended by Associate Editor Z. Lin.

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).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2011.2158722

recursively, where $E[\cdot]$ is the expected value of $[\cdot]$, $\boldsymbol{x}_k$ is a vector of dimension M representing the input signal, d_k is the desired signal, and $\boldsymbol{w}$ is the weight vector, which is also of dimension M. An approximate solution of the problem in (1) can be obtained by using the weight-vector update equation

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + 2\mu_k \boldsymbol{S}_{k-1}\boldsymbol{x}_k e_k \tag{2}$$

where μ_k is the step size, $\boldsymbol{S}_{k-1}$ is a positive definite matrix of dimension $M \times M$, and

$$e_k = d_k - y_k \tag{3}$$

is the *a priori* error for the output signal $y_k = \boldsymbol{w}_{k-1}^T \boldsymbol{x}_k$. If $\boldsymbol{S}_{k-1}$ in (2) is chosen as the $M \times M$ identity matrix, then the update equation in (2) would minimize the objective function

$$J_{\boldsymbol{w}_{k-1}} = \left(d_k - \boldsymbol{w}_{k-1}^T \boldsymbol{x}_k\right)^2 \tag{4}$$

with respect to the steepest-descent direction, and a series of updates would eventually yield an approximate solution of the problem in (1). Other choices of $\boldsymbol{S}_{k-1}$ would entail different search directions but would serve the same purpose as long as $\boldsymbol{S}_{k-1}$ is positive definite. In order to use an approximation of the Newton direction $\boldsymbol{S}_{k-1}$ in (2), we obtain $\boldsymbol{S}_{k-1}$ by using the gradient of $J_{\boldsymbol{w}_{k-1}}$ in (4) in the rank-one update formula of the classical QN optimization algorithm [11], which is given by

$$\boldsymbol{S}_k = \boldsymbol{S}_{k-1} - \frac{(\boldsymbol{\delta}_k - \boldsymbol{S}_{k-1}\boldsymbol{\rho}_k)(\boldsymbol{\delta}_k - \boldsymbol{S}_{k-1}\boldsymbol{\rho}_k)^T}{(\boldsymbol{\delta}_k - \boldsymbol{S}_{k-1}\boldsymbol{\rho}_k)^T \boldsymbol{\rho}_k} \tag{5}$$

where

$$\boldsymbol{\delta}_k = \boldsymbol{w}_k - \boldsymbol{w}_{k-1} \tag{6}$$

$$\boldsymbol{\rho}_k = \frac{\partial e_{k+1}^2}{\partial \boldsymbol{w}_k} - \frac{\partial e_k^2}{\partial \boldsymbol{w}_{k-1}}. \tag{7}$$

This formula satisfies the Fletcher QN condition $\boldsymbol{S}_k \boldsymbol{\rho}_k = \boldsymbol{\delta}_k$ [12]. From (3), we note that e_{k+1} would require future data $\boldsymbol{x}_{k+1}$ and d_{k+1}. To circumvent this problem, we use the *a posteriori* error

$$\epsilon_k = d_k - \boldsymbol{x}_k^T \boldsymbol{w}_k \tag{8}$$

in place of e_{k+1} in (7). As a first step in the proposed algorithm, we obtain a value of step size μ_k in (2) by solving the optimization problem

$$\underset{\mu_k}{\text{minimize}} \begin{cases} \left|d_k - \boldsymbol{x}_k^T \boldsymbol{w}_k\right| - \gamma, & \text{if } |e_k| > \gamma \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where γ is a prespecified error bound. The solution of this problem can be obtained as

$$\mu_k = \alpha_k \frac{1}{2\tau_k} \tag{10}$$

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

$$\alpha_k = \begin{cases} 1 - \frac{\gamma}{|e_k|}, & \text{if } |e_k| > \gamma \\ 0, & \text{otherwise.} \end{cases} \tag{11}$$

In effect, step size μ_k is chosen to force equality $|\epsilon_k| = \gamma$ whenever $|e_k| > \gamma$. Since μ_k in (10) forces ϵ_k to assume the value of the prespecified error bound, i.e.,

$$\epsilon_k = \gamma \cdot \text{sign}(e_k) \tag{12}$$

we obtain $\boldsymbol{\nabla}\epsilon_k^2 = \boldsymbol{0}$, and hence from (7), we have

$$\boldsymbol{\rho}_k = 2e_k \boldsymbol{x}_k. \tag{13}$$

Vector $\boldsymbol{\delta}_k$, which is linearly dependent on $\boldsymbol{S}_{k-1}\boldsymbol{x}_k$, can be obtained by using (2) and (6). Since the equality in (12) is satisfied for each update, we can use the *a posteriori* error to obtain

$$\boldsymbol{\delta}_k = 2\gamma \cdot \text{sign}(e_k) \boldsymbol{S}_{k-1}\boldsymbol{x}_k \tag{14}$$

instead of the *a priori* error used in the KQN algorithm reported in [8]. Now, substituting $\boldsymbol{\rho}_k$ and $\boldsymbol{\delta}_k$ given by (13) and (14) in (5), we obtain an update equation for matrix $\boldsymbol{S}_k$ for the proposed robust QN algorithm as

$$\boldsymbol{S}_k = \boldsymbol{S}_{k-1} - \alpha_k \frac{\boldsymbol{S}_{k-1}\boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{S}_{k-1}}{\boldsymbol{x}_k^T \boldsymbol{S}_{k-1} \boldsymbol{x}_k}. \tag{15}$$

Substituting μ_k given by (10) in (2), we obtain the corresponding weight-vector update equation as

$$\boldsymbol{w}_k = \boldsymbol{w}_{k-1} + \alpha_k \frac{\boldsymbol{S}_{k-1}\boldsymbol{x}_k}{\boldsymbol{x}_k^T \boldsymbol{S}_{k-1}\boldsymbol{x}_k} e_k. \tag{16}$$

In order to achieve robust performance against impulsive noise, we choose error bound γ as

$$\gamma = \begin{cases} |e_k| - \upsilon\theta_k, & \text{if } |e_k| > \theta_k \\ \gamma_c, & \text{otherwise} \end{cases} \tag{17}$$

where γ_c is a prespecified error bound chosen as $\sqrt{5}\sigma_v$, where σ_v^2 is the variance of the measurement noise, $0 \leq \upsilon \ll 1$ is a scalar, and θ_k is a threshold parameter, which is estimated as

$$\theta_k = 1.98\sigma_k \tag{18}$$

where

$$\sigma_k^2 = \lambda\sigma_{k-1}^2 + (1 - \lambda)\min(\boldsymbol{g}_k) \tag{19}$$

with $0 \ll \lambda < 1$, $\boldsymbol{g}_k = [\, e_k^2 + \epsilon \quad \cdots \quad e_{k-P+1}^2 + \epsilon \,]$ is a vector of dimension P, where ϵ is a small scalar. Whenever $|e_k| < \gamma$, no update is applied. Consequently, the amount of computation and the required storage are significantly reduced since $\boldsymbol{S}_k$ is not evaluated in every iteration. A similar weight-vector update strategy has been used in set-membership adaptation algorithms [13]–[15], but the mathematical framework of those algorithms is very different from that of the proposed robust QN algorithm. The estimator in (19) is robust to outliers. A large σ_0^2 would cause $|e_k|$ to be less than θ_k during the transient state, and therefore, the algorithm would work with error bound γ_c, which would increase the initial rate of convergence. For a sudden system disturbance, θ_k would also be very large, in which case we obtain $\gamma_c < |e_k| < \theta_k$, and thus, the algorithm would again use error bound γ_c, and therefore, the tracking capability of the algorithm would be retained. For an impulsive noise-corrupted error signal, θ_k would not increase, in which case the error bound would be $\theta_k = |e_k| - \upsilon\theta_k$, and this would suppress the impulsive noise.

B. RQN Algorithm With a Variable Threshold

The RQN algorithm with a variable threshold is essentially the same as before except that the error bound γ_c in (17) is estimated as

$$\xi_k = \beta\xi_{k-1} + (1-\beta)\min\left(\xi_{k-1}, \frac{\left|d_k^2 - y_k^2\right|}{d_k^2}\right) \tag{20}$$

$$\hat{\sigma}_k^2 = \beta\hat{\sigma}_{k-1}^2 + (1-\beta)\min\left(\hat{\sigma}_{k-1}^2, \sigma_k^2\right) \tag{21}$$

$$\gamma_{c,k} = \sqrt{\xi_k\gamma_{c,0}} + 1.12\left[1 + \mathrm{sign}(1-\xi_k)\right]\hat{\sigma}_k \tag{22}$$

where $\gamma_{c,0}$ is a rough estimate of σ_v, and $\xi_0 \gg 1$. During steady state, we obtain $\xi_k \approx 0$ and $\hat{\sigma}_k^2 \approx \sigma_v^2$, and hence, $\gamma_{c,k}$ would be $2.24\sigma_v$, and this would yield reduced steady-state misalignment. In the transient state, $\xi_k \approx \xi_0$, and therefore, the algorithm would work with $\gamma_{c,k} = \sqrt{\xi_k\gamma_{c,0}}$, and this would yield faster convergence. The parameters in (20) and (21) are robust with respect to outliers as each is based on the minimum of its two most recent values.

The variance is estimated as

$$\sigma_k^2 = \lambda\sigma_{k-1}^2 + (1-\lambda)\mathrm{median}(\boldsymbol{g}_k). \tag{23}$$

For Gaussian signals, the median of the squares of the signal samples in the variance estimator usually gives a more accurate estimate of the variance of the signal than the instantaneous values of the squares of the signal samples.

The use of the median operation in adaptive filters was introduced by Zou et al. in [4] and was later used by other researchers, e.g., in [18].

A variable threshold γ_c is useful in applications where the noise variance σ_v^2 is unknown.

C. Discussion

In the proposed algorithms, we obtain $0 \le \alpha_k < 1$ for both values of γ in (17), and hence, the estimated $\boldsymbol{S}_k$ in (15) would remain positive definite indefinitely if $\boldsymbol{S}_k$ is initialized with a positive definite matrix [16]. The RLS-type robust algorithms in [17] and [18] are implemented using a fast transversal filter implementation [8] to reduce their computational complexity from order M^2, which is denoted by $O(M^2)$, to $O(M)$, and therefore, both algorithms would inherit the problems associated with the RLS algorithms of order M^2. Moreover, the Huber function used in [17] does not have a closed-form solution, and hence, the solution obtained can be suboptimal, and in addition, its tracking capability can be compromised [18]. The solution obtained in [18] can also be suboptimal, as shown in [18]. The adaptation algorithm in [18] is robust in the sense that it returns to the true solution without losing its initial convergence speed after being subjected to impulsive noise. Similarly, the proposed robust QN algorithms are robust with respect to impulsive noise in the sense that they return to the true solution faster than the initial convergence. The KQN and known RQN (KRQN) algorithms do not employ a variable step size. However, the KRQN algorithm requires an additional amount of computation of $O[M\ \log_2(M)]$ per iteration, as compared with the KQN algorithm and the proposed QN algorithm with a fixed threshold.

III. Stability Analysis

Here, we address stability issues associated with the proposed RQN algorithms. For purposes of analysis, we assume that the desired response for the adaptive filter is generated as

$$d_k = \boldsymbol{x}_k^T\boldsymbol{w}_{\mathrm{opt}} \tag{24}$$

where $\boldsymbol{w}_{\mathrm{opt}}$ is the weight vector of an FIR filter. We establish the convergence behavior of the proposed RQN algorithm by examining the behavior of the weight-error vector, which is defined as

$$\triangle\boldsymbol{w}_k = \boldsymbol{w}_{\mathrm{opt}} - \boldsymbol{w}_k. \tag{25}$$

Using (24) and (25), the error signal in (3) can be expressed as

$$e_k = \boldsymbol{x}_k^T\triangle\boldsymbol{w}_{k-1}. \tag{26}$$

Subtracting $\boldsymbol{w}_{\mathrm{opt}}$ from both sides of (16) and using the above relations, we obtain

$$\begin{aligned}\triangle\boldsymbol{w}_k &= \left(\mathbf{I} - \alpha_k\frac{\boldsymbol{S}_{k-1}\boldsymbol{x}_k\boldsymbol{x}_k^T}{\boldsymbol{x}_k^T\boldsymbol{S}_{k-1}\boldsymbol{x}_k}\right)\triangle\boldsymbol{w}_{k-1}\\ &= \boldsymbol{P}_k\triangle\boldsymbol{w}_{k-1}.\end{aligned} \tag{27}$$

The global asymptotic convergence of the weight-error vector can be assured if and only if $E[\boldsymbol{P}_k]$ is time invariant and its eigenvalues are strictly inside the unit circle. However, certain strong independence assumptions have to be made to obtain a time-invariant description of a system such as that represented by (27). As an alternative, conditions for convergence with probability 1 can be achieved by using a system based on $\triangle\boldsymbol{w}_k$ rather than $E[\triangle\boldsymbol{w}_k]$. A similar approach has been used to demonstrate the stability of other algorithms, such as, for example, the KQN algorithm in [7] and the constrained affine projection algorithm in [13]. Before considering the stability of the proposed RQN algorithm, we apply the matrix inversion lemma [8] to (15) to obtain

$$\boldsymbol{R}_k = \boldsymbol{S}_k^{-1} = \boldsymbol{R}_{k-1} + \frac{\alpha_k}{(1-\alpha_k)\tau_k}\boldsymbol{x}_k\boldsymbol{x}_k^T \tag{28}$$

where $\tau_k = \boldsymbol{x}_k^T\boldsymbol{S}_{k-1}\boldsymbol{x}_k$. Since $0 < \alpha_k < 1$ in each update, matrices $\boldsymbol{S}_k$ and $\boldsymbol{R}_k$ remain positive definite and bounded indefinitely [16], and therefore, the following theorem can be established.

Theorem 1: If the input signal is persistently exciting, then the system in (27) is asymptotically stable, and consequently, the proposed RQN algorithms are also asymptotically stable.

Proof: Since matrices $\boldsymbol{S}_k$ and $\boldsymbol{R}_k$ are bounded and invertible according to Lemma 1 in [7], an equivalent system of equations (in the Lyapunov sense) for $\triangle\boldsymbol{w}_k$ in (27) can be obtained as

$$\overline{\triangle\boldsymbol{w}}_k = \boldsymbol{R}_k^{1/2}\triangle\boldsymbol{w}_k \tag{29}$$

where $\boldsymbol{R}_k = \boldsymbol{R}_k^{T/2}\boldsymbol{R}_k^{1/2}$. If the system represented by (29) is stable or unstable, then the system represented by (27) is also stable or unstable, as appropriate. Taking the Euclidean norm of both sides of (29), we obtain

$$\|\overline{\triangle\boldsymbol{w}}_k\|^2 = \triangle\boldsymbol{w}_k^T\boldsymbol{R}_k\triangle\boldsymbol{w}_k. \tag{30}$$

Substituting $\triangle \boldsymbol{w}_k$ given by (27) in (30), we get

$$\|\overline{\triangle \boldsymbol{w}}_k\|^2 = \triangle \boldsymbol{w}_{k-1}^T \boldsymbol{P}_k^T \boldsymbol{R}_k \boldsymbol{P}_k \triangle \boldsymbol{w}_{k-1}. \tag{31}$$

On the other hand, substituting $\boldsymbol{R}_k$ given by (28) in (31), we have

$$\begin{aligned}\|\overline{\triangle \boldsymbol{w}}_k\|^2 =& \triangle \boldsymbol{w}_{k-1}^T \boldsymbol{P}_k^T \boldsymbol{R}_{k-1} \boldsymbol{P}_k \triangle \boldsymbol{w}_{k-1} \\ &+ \frac{\alpha_k}{(1-\alpha_k)\tau_k} \cdot \triangle \boldsymbol{w}_{k-1}^T \boldsymbol{P}_k^T \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{P}_k \triangle \boldsymbol{w}_{k-1}. \end{aligned} \tag{32}$$

It is easy to verify that

$$\boldsymbol{P}_k^T \boldsymbol{R}_{k-1} \boldsymbol{P}_k = \boldsymbol{R}_{k-1} - (2-\alpha_k)\frac{\alpha_k}{\tau_k} \boldsymbol{x}_k \boldsymbol{x}_k^T \tag{33}$$

$$\boldsymbol{P}_k^T \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{P}_k = (1-\alpha_k)^2 \boldsymbol{x}_k \boldsymbol{x}_k^T. \tag{34}$$

Now, if we use (33) and (34) in (32) and then use (30), we have

$$\|\overline{\triangle \boldsymbol{w}}_k\|^2 = \|\overline{\triangle \boldsymbol{w}}_{k-1}\|^2 - \frac{\alpha_k}{\tau_k} \triangle \boldsymbol{w}_{k-1}^T \boldsymbol{x}_k \boldsymbol{x}_k^T \triangle \boldsymbol{w}_{k-1}. \tag{35}$$

Since $\alpha_k \subset (0,1)$ and $\tau_k > 0$ for all k, we obtain

$$\|\overline{\triangle \boldsymbol{w}}_k\|^2 \leq \|\overline{\triangle \boldsymbol{w}}_{k-1}\|^2. \tag{36}$$

The left-hand side in (36) would be equal to the right-hand side in an interval $[k_1, k_2]$ if and only if $\boldsymbol{x}_k$ remains orthogonal to $\triangle \boldsymbol{w}_{k-1}$ for all $k \in [k_1, k_2]$. However, in such a situation, we would obtain $\triangle \boldsymbol{w}_k = \triangle \boldsymbol{w}_{k-1}$ for all $k \in [k_1, k_2]$ in (27). Therefore, $\|\overline{\triangle \boldsymbol{w}}_k\|^2 = \|\overline{\triangle \boldsymbol{w}}_{k-1}\|^2$ would hold true if and only if $\overline{\triangle \boldsymbol{w}}_{k_1} = \overline{\triangle \boldsymbol{w}}_{k_1+1} = \cdots = \overline{\triangle \boldsymbol{w}}_{k_2} = \overline{\triangle \boldsymbol{w}}$. However, if the input signal is persistently exciting, then there is an infinite number of sets $S_i = \{\boldsymbol{x}_{k_{1,i}}, \ldots \boldsymbol{x}_{k_{2,i}}\}$ with $M \leq k_{2,i} - k_{1,i} < M'$ such that each set S_i completely spans $\Re^M$ for some finite value of $M' > 0$ [7], [14]. Thus, it would be impossible for $\triangle \boldsymbol{w}_{k-1}$ to be orthogonal to $\boldsymbol{x}_k$ for all $\boldsymbol{x}_k \in S_i$, and as a result, $\|\overline{\triangle \boldsymbol{w}}_k\|^2 < \|\overline{\triangle \boldsymbol{w}}_{k-1}\|^2$, which proves that the system in (27) and, in turn, the proposed RQN algorithms are stable. As the number of sets is infinite, it follows that $\|\overline{\triangle \boldsymbol{w}}_k\|^2 \to 0$ for $k \to \infty$, and therefore, the system in (27) and the proposed RQN algorithms are both stable and asymptotically stable. ■

IV. Practical Considerations

In low-cost fixed-point hardware implementations, the accumulation of roundoff errors can cause $\boldsymbol{S}_k$ to lose its positive definiteness. This problem can be eliminated by periodically reinitializing $\boldsymbol{S}_k$ using the identity matrix. This is also done in the QN algorithm in [7] whenever $\boldsymbol{x}_k^T \boldsymbol{S}_{k-1} \boldsymbol{x}_k < 10^{-3}$ for the case of a fixed-point implementation and also in the classical optimization context. However, very frequent reinitializations could slow the convergence of the algorithm. A compromise number of reinitializations can be achieved by reinitializing $\boldsymbol{S}_k$ whenever $|e_k| > 2.576\sigma_k$, in which case the probability of reinitialization in a given iteration assumes the value of 0.01. Reinitialization of other parameters is not required. In floating-point arithmetic, reinitialization is unnecessary.

The variance estimator of the error signal in (19) should be initialized with a large value to ensure that the probability that $\gamma = \gamma_c$ in the transient state is increased. Although a rough choice of σ_0^2 would work, we have used $\sigma_0^2 = c_1 M/\sigma_v^2$, where $0 < c_1 < 1$ is a positive constant. This choice yields good results. The forgetting factors in (19) and (21) are chosen as $\lambda = 1 - 1/(c_2 M)$ and $\beta = 1 - 1/(c_3 M)$, respectively, where c_2 and c_3 are positive scalars [18]. The length P of vector $\boldsymbol{g}_k$ should be greater than the duration of the impulsive noise. A reduced υ in (17) would yield reduced steady-state misalignment and improved robustness with respect to impulsive noise. On the other hand, an increased γ_c and θ_k would reduce the number of updates and yield reduced steady-state misalignment. However, the convergence speed would be compromised in such a situation.

V. Simulation Results

In order to evaluate the performance of the proposed RQN algorithms with fixed and variable γ_c, which are designated as the PRQN-I and PRQN-II algorithms, respectively, a system identification application was considered, as detailed below. For the sake of comparison, simulations were carried out with the KQN and KRQN algorithms in [5] and [7], respectively.

The adaptive filter was used to identify a 36th-order low-pass FIR filter with a normalized cutoff frequency of 0.3. The input signal was generated by filtering a white Gaussian noise signal with zero mean and unity variance through a recursive filter with a single pole at 0.95. The measurement noise added to the desired signal was a white Gaussian noise signal with zero mean and variances of 10^{-3} and 10^{-6} (signal-to-noise ratios (SNRs) of 30 and 60 dB, respectively). The impulse response of the FIR filter was multiplied by -1 at iteration 2000 to check the tracking capability of the algorithm. Impulsive noise with a maximum duration of $3T_s$, where T_s is the sampling duration, was added to the desired signal at iterations 1000, 1300, and 3300 using a Bernoulli trial with probability 0.001 [19]. All algorithms were initialized with $\boldsymbol{S}_0 = \boldsymbol{I}$, and $\boldsymbol{w}_0 = \boldsymbol{0}$. The parameters for the PRQN-I and PRQN-II algorithms were set to $\upsilon = 0.30$, $P = 5$, $c_1 = 1/(2M)$, $c_2 = 0.5$ and $\upsilon = 0.30$, $P = 15$, $\xi_0 = 10$, $\hat{\sigma}_0^2 = 10$, $c_3 = 2$, respectively.

We have explored two options for the evaluation of $\boldsymbol{S}_k$ in the implementation of the KRQN algorithm, first using

$$\boldsymbol{S}_k = \lambda \boldsymbol{S}_{k-1} + \left\{ \left[\frac{1 - \lambda q(e_k)\tau_k}{q(e_k)\tau_k^2} \right] \boldsymbol{S}_{k-1} \boldsymbol{x}_k \right\} \boldsymbol{x}_k^T \boldsymbol{S}_{k-1} \tag{37}$$

and then using

$$\boldsymbol{S}_k = \lambda \boldsymbol{S}_{k-1} + \left[\frac{1 - \lambda q(e_k)\tau_k}{q(e_k)\tau_k^2} \right] \left\{ \boldsymbol{S}_{k-1} \boldsymbol{x}_k \boldsymbol{x}_k^T \boldsymbol{S}_{k-1} \right\}. \tag{38}$$

Although (37) and (38) are equivalent, the implementation of (37) was subject to numerical ill-conditioning, which caused instability. On the other hand, the implementation of (38) was found to be more robust, although it requires increased computational effort. We have used (38) in our implementation of the KRON algorithm. The parameters for the KRQN algorithm were set to the values suggested in [5].

The learning curves obtained from 1000 independent trials by using the KQN, KRQN, PRQN-I, and PRQN-II algorithms are illustrated in Fig. 1(a) and (b). As shown, the PRQN-I and PRQN-II algorithms offer robust performance with respect to impulsive noise and yield significantly reduced steady-state misalignment relative to those in the other algorithms. On the other hand, the KQN algorithm is seriously compromised

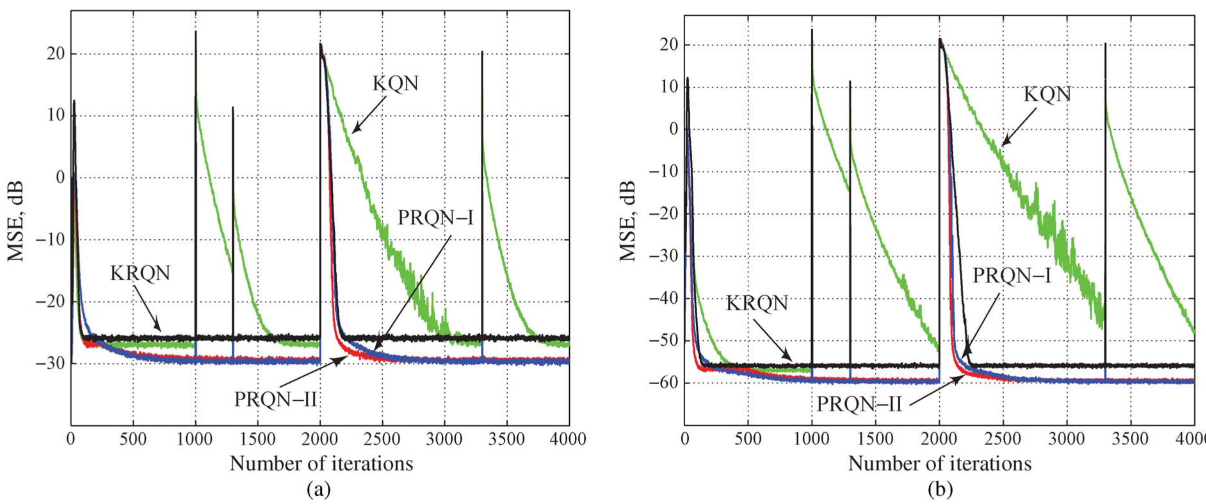

Fig. 1. Learning curves for system-identification application. (a) SNR $= 30$ dB. (b) SNR $= 60$ dB.

by impulsive noise in terms of both robustness and tracking capability. The KRQN algorithm also offers robust performance with respect to impulsive noise, but its tracking performance is not as good as those of the PRQN-I and PRQN-II algorithms. The total numbers of updates required by the PRQN-I and PRQN-II algorithms were 2230 and 1356 for an SNR of 30 dB, and 2070 and 1207 for an SNR of 60 dB, respectively, as compared with 4000 in the other algorithms. Note that two systems were identified in this experiment, one before iteration 2000 and the other after iteration 2000. Otherwise, the number of updates would be reduced to half.

VI. Conclusion

Two new robust QN adaptation algorithms have been developed on the basis of the mathematical framework of the classical QN optimization algorithm, which lead to an improved estimate of the inverse of the Hessian. Like the data-selective QN algorithm we described in [16], the proposed RQN algorithms incorporate data-selective adaptation, which significantly reduces the number of adaptations required. A stability analysis shows that the proposed RQN algorithms are asymptotically stable.

Simulation results obtained in the case of a system identification application demonstrate that the proposed RQN algorithms converge faster than the KQN algorithms in [7] for medium to high SNRs. In addition, they offer improved robustness against impulsive noise, as well as improved computational efficiency and tracking relative the KRQN algorithm reported in [5].

References

[1] S. Koike, "Adaptive threshold nonlinear algorithm for adaptive filters with robustness against impulsive noise," *IEEE Trans. Signal Process.*, vol. 45, no. 9, pp. 2391–2395, Sep. 1997.

[2] J. F. Weng and S. H. Leung, "Adaptive nonlinear RLS algorithm for robust filtering in impulse noise," in *Proc. IEEE Inter. Symp. Circuits Syst.*, Jun. 1997, pp. 2337–2340.

[3] P. Petrus, "Robust Huber adaptive filter," *IEEE Trans. Signal Process.*, vol. 47, no. 4, pp. 1129–1133, Apr. 1999.

[4] Y. Zou, S. C. Chan, and T. S. Ng, "A recursive least M-estimate (RLM) adaptive filter for robust filtering in impulse noise," *IEEE Signal Process. Lett.*, vol. 7, no. 11, pp. 324–326, Nov. 2000.

[5] Y. Zou and S. C. Chan, "A robust quasi-Newton adaptive filtering algorithm for impulse noise suppression," in *Proc. IEEE Inter. Symp. Circuits Syst.*, May 2001, pp. 677–680.

[6] P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection*. NewYork: Wiley, 1987.

[7] M. L. R. de Campos and A. Antoniou, "A new quasi-Newton adaptive filtering algorithm," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 44, no. 11, pp. 924–934, Nov. 1997.

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

[9] 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.

[10] G. E. Bottomley and S. T. Alexander, "A novel approach for stabilizing recursive least squares filters," *IEEE Trans. Signal Process.*, vol. 39, no. 8, pp. 1770–1779, Aug. 1991.

[11] A. Antoniou and W. S. Lu, *Practical Optimization*. New York: Springer-Verlag, 2007.

[12] R. Fletcher, *Practical Methods of Optimization*. Hoboken, NJ: Wiley, 1980.

[13] S. Werner, J. A. Apolinario, M. L. R. de Campos, and P. S. R. Diniz, "Low complexity constrained affine-projection algorithms," *IEEE Trans. Signal Process.*, vol. 53, no. 12, pp. 4545–4555, Dec. 2005.

[14] P. S. R. Diniz and S. Werner, "Set-membership binormalized data-reusing LMS algorithms," *IEEE Trans. Signal Process.*, vol. 51, no. 1, pp. 124–134, Jan. 2003.

[15] S. Werner and P. S. R. Diniz, "Set-membership affine projection algorithm," *IEEE Signal Process. Lett.*, vol. 8, no. 8, pp. 231–235, Aug. 2001.

[16] M. Z. A. Bhotto and A. Antoniou, "Improved quasi-Newton adaptive filtering algorithm," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 2109–2118, Aug. 2010.

[17] S. C. Chan and Y. Zou, "A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: Fast algorithm and convergence analysis," *IEEE Trans. Signal Process.*, vol. 52, no. 4, pp. 975–991, Apr. 2004.

[18] L. R. Vega, H. Rey, J. Benesty, and S. Tressens, "A fast robust recursive least-squares algorithm," *IEEE Trans. Signal Process.*, vol. 57, no. 3, pp. 1209–1216, Mar. 2009.

[19] P. Z. Peebles, *Probability, Random Variables and Random Signal Principles*. NewYork: McGraw-Hill, 2000.