

# Stochastic convergence analysis of recursive successive over-relaxation algorithm in adaptive filtering

Metin Hatun<sup>1</sup> · Osman Hilmi Koçal<sup>2</sup>

Received: 5 October 2015 / Revised: 9 May 2016 / Accepted: 17 May 2016 / Published online: 27 June 2016  
© Springer-Verlag London 2016

**Abstract** A stochastic convergence analysis of the parameter vector estimation obtained by the recursive successive over-relaxation (RSOR) algorithm is performed in mean sense and mean-square sense. Also, excess of mean-square error and misadjustment analysis of the RSOR algorithm is presented. These results are verified by ensemble-averaged computer simulations. Furthermore, the performance of the RSOR algorithm is examined using a system identification example and compared with other widely used adaptive algorithms. Computer simulations show that the RSOR algorithm has better convergence rate than the widely used gradient-based algorithms and gives comparable results obtained by the recursive least-squares RLS algorithm.

**Keywords** Adaptive filters · Successive over-relaxation · Gauss–Seidel · System identification · Convergence analysis

## 1 Introduction

Adaptive filter algorithms can be classified into two main groups: gradient based and least squares based. The well-known gradient-based adaptive algorithms are the least-mean-squares (LMS) algorithm [1], the normalized LMS (NLMS) algorithm [2], and the affine projection algorithm (APA) [3]. These algorithms are widely used in practical

implementations because of their relatively low computational complexity. The major limitations of the gradient-based algorithms are their relatively slow convergence rates and sensitivity to the eigenvalue spread of the input correlation matrix. The other type of adaptive algorithm is based on the least squares, and the most widely used one is the recursive least-squares (RLS) algorithm. The RLS algorithm has a faster convergence rate than the gradient-based algorithms, although the computational complexity is greater [4,5].

There is a clear difference between gradient-based algorithms and the RLS algorithm in terms of the convergence rate and computational complexity [4,5]. However, a recursive algorithm based on the solution of the time-averaged normal equation using a single-step Gauss–Seidel iteration between two consecutive data samples has been proposed as an intermediate method [6]. The Gauss–Seidel algorithm has also been used as an optimization algorithm called the Euclidean direction search (EDS) algorithm [7–13]. An accelerated version of the EDS algorithm is also presented in [14]. As an alternative to the Gauss–Seidel iterations, the one-step Jacobi iteration is used in [15]. A recursive implementation of the Gauss–Seidel algorithm (RGS) is also used to adjust self-tuning adaptive controller parameters directly in [16]. The main advantage of the RGS algorithm as an intermediate method is a faster convergence rate than gradient-based algorithms and a lower computational complexity than the RLS algorithm. Similar to the RGS algorithm, the recursive successive over-relaxation (RSOR) algorithm is proposed for adaptive FIR filtering based on the use of a one-step successive over-relaxation (SOR) iteration between two data samples [17].

The aim of this paper is to present a steady-state convergence analysis of the parameter error vector which is performed in the mean sense and mean-square sense. Also, the convergence performance of the RSOR algorithm is

✉ Metin Hatun  
metinh@uludag.edu.tr

Osman Hilmi Koçal  
osman.kocal@yalo.edu.tr

<sup>1</sup> Electrical-Electronic Engineering Department, Faculty of Engineering, Uludağ University, Bursa, Turkey

<sup>2</sup> Computer Engineering Department, Faculty of Engineering, Yalova University, Yalova, Turkey

examined using computer simulations and compared with the well-known gradient-based algorithms and the RLS algorithm.

The paper is organized as follows: in Sect. 2, the use of the RSOR algorithm for adaptive filtering is presented. Stochastic convergence analysis of the RSOR algorithm is presented in Sect. 3. Computer simulations using a system identification example are presented in Sect. 4. Concluding remarks are given in Sect. 5.

## 2 Review of the RSOR algorithm

An adaptive filter is a digital filter whose parameters  $\hat{\mathbf{w}}(n)$  are adjusted by an adaptive algorithm. The output signal  $y(n)$  can be obtained by the following adaptive FIR filter model

$$y(n) = \mathbf{x}^T(n)\hat{\mathbf{w}}(n) \quad (1)$$

where the input vector  $\mathbf{x}(n)$  and the parameter vector  $\hat{\mathbf{w}}(n)$  can be defined as follows:

$$\mathbf{x}(n) = [x(n) \ x(n-1) \ \dots \ x(n-M+1)]^T \quad (2)$$

$$\hat{\mathbf{w}}(n) = [\hat{w}_0(n) \ \hat{w}_1(n) \ \dots \ \hat{w}_{M-1}(n)]^T \quad (3)$$

where  $x(n)$  is the input signal of the filter and  $M$  is filter length. An error signal  $e(n)$  is defined by comparing the desired signal  $d(n)$  and the output signal  $y(n)$  as follows:

$$e(n) = d(n) - \mathbf{x}^T(n)\hat{\mathbf{w}}(n). \quad (4)$$

Least-squares-based algorithms use the following cost function

$$V(n, \mathbf{w}) = \sum_{i=1}^n \lambda^{n-i} e^2(i) \quad (5)$$

where  $\lambda$  is the forgetting factor,  $0 < \lambda \leq 1$ . The estimated parameter vector  $\hat{\mathbf{w}}(n)$ , which minimizes least-squares error function (5) for  $n$ -step data, can be computed by

$$\hat{\mathbf{w}}(n) = \mathbf{R}^{-1}(n)\mathbf{p}(n) \quad (6)$$

where  $\hat{\mathbf{w}}(n)$  can also be obtained by solving the following time-averaged normal equation:

$$\mathbf{R}(n)\hat{\mathbf{w}}(n) = \mathbf{p}(n) \quad (7)$$

where  $\mathbf{R}(n)$  denotes an estimate of the  $M \times M$ -dimensional autocorrelation matrix of the input vector  $\mathbf{x}(n)$ ,  $\mathbf{p}(n)$  denotes an estimate of the  $M \times 1$ -dimensional cross-correlation vector between  $\mathbf{x}(n)$  and the desired signal  $d(n)$ , and  $\hat{\mathbf{w}}(n)$  is an estimate of the  $M \times 1$ -dimensional parameter vector (3).

The estimated values  $\mathbf{R}(n)$  and  $\mathbf{p}(n)$  are computed at time step  $n$  as follows, respectively:

$$\mathbf{R}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i) \quad (8)$$

$$\mathbf{p}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)d(i). \quad (9)$$

In practical applications, the values of  $\mathbf{R}(n)$  and  $\mathbf{p}(n)$  are updated as follows:

$$\mathbf{R}(n) = \lambda\mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n) \quad (10)$$

$$\mathbf{p}(n) = \lambda\mathbf{p}(n-1) + \mathbf{x}(n)d(n). \quad (11)$$

By using a strategy similar to that of the RGS algorithm [16], the recursive implementation of the SOR (RSOR) algorithm to minimize the least-squares error function (5) can be given as

$$\hat{w}_i(n+1) = \left[ p_i(n) - \sum_{j=1}^{i-1} R_{ij}(n)\hat{w}_j(n+1) - \sum_{j=i+1}^M R_{ij}(n)\hat{w}_j(n) \right] \frac{\omega}{R_{ii}(n)} + (1-\omega)\hat{w}_i(n) \quad (12)$$

$$i = 1, 2, \dots, M, \quad (0 < \omega < 2)$$

where  $\omega$  is known as the relaxation parameter,  $\hat{w}_i(n)$  is the  $i$ th element of the estimated parameter vector  $\hat{\mathbf{w}}(n)$ ,  $p_i(n)$  is the  $i$ th element of the estimated cross-correlation vector  $\mathbf{p}(n)$ , and  $R_{ij}(n)$  indicates the  $i$ th row and  $j$ th column of the estimated autocorrelation matrix  $\mathbf{R}(n)$ . Equations (10)–(12) are used to implement the RSOR algorithm. In the RSOR algorithm, the discrete-time index  $n$  is used as the iteration index. Thus, the RSOR algorithm is implemented using a one-cycle SOR iteration between two consecutive data samples. When  $\omega = 1$ , the SOR iterations are equal to the Gauss–Seidel iterations [18]. By taking  $\omega > 1$ , the RSOR algorithm can be used to obtain faster convergence than the RGS algorithm [17].

## 3 Stochastic convergence analysis

In this section, the asymptotic convergence, i.e., the steady-state behavior, of the RSOR parameter estimation vector is analyzed in the mean sense and mean-square sense. The analysis is similar to that in [19, 20].

The parameter convergence of the RSOR algorithm is based on the positive definiteness of the autocorrelation matrix  $\mathbf{R}(n)$ . In the analysis, the following assumption is used:

**Assumption** The excitation signal  $x(n)$  is persistently exciting, i.e., there exist  $\alpha > 0$  and  $\beta > 0$  satisfying

$$0 < \alpha \mathbf{I} \leq \frac{1}{n} \sum_{i=1}^n \mathbf{x}(i)\mathbf{x}^T(i) \leq \beta \mathbf{I} < \infty \tag{13}$$

over a set of  $n$  consecutive data samples for all  $i$ . This assumption means that the minimum and maximum eigenvalues of the sum in (13) are bounded by  $\alpha$  and  $\beta$  [21].

### 3.1 Mean convergence analysis

The estimated autocorrelation matrix  $\mathbf{R}(n)$  can be decomposed as the sum of its lower triangular matrix, diagonal matrix, and upper triangular matrix as

$$\mathbf{R}(n) = \mathbf{R}_L(n) + \mathbf{R}_D(n) + \mathbf{R}_U(n). \tag{14}$$

Given the classical SOR method in [18], the splitting of  $\mathbf{R}(n)$  can be written as

$$\omega \mathbf{R}(n) = [\mathbf{R}_D(n) + \omega \mathbf{R}_L(n)] - [(1 - \omega)\mathbf{R}_D(n) - \omega \mathbf{R}_U(n)]. \tag{15}$$

Based on the splitting in (15) and using (14), the corresponding RSOR algorithm (12) for the solution of (7) is rewritten as recursion

$$[\mathbf{R}_D(n) + \omega \mathbf{R}_L(n)]\hat{\mathbf{w}}(n+1) = \{[\mathbf{R}_D(n) + \omega \mathbf{R}_L(n)] - \omega \mathbf{R}(n)\}\hat{\mathbf{w}}(n) + \omega \mathbf{p}(n). \tag{16}$$

Let us define

$$\mathbf{x}(i)\mathbf{x}^T(i) = \bar{\mathbf{R}} + \tilde{\mathbf{R}}(i) \tag{17}$$

$$\mathbf{x}(i)d(i) = \bar{\mathbf{p}} + \tilde{\mathbf{p}}(i) \tag{18}$$

where  $\tilde{\mathbf{R}}(i)$  denotes the random part of the autocorrelation matrix,  $\tilde{\mathbf{p}}(i)$  denotes the random part of the cross-correlation vector,  $\bar{\mathbf{R}} = E\{\mathbf{x}(i)\mathbf{x}^T(i)\}$ ,  $\bar{\mathbf{p}} = E\{\mathbf{x}(i)d(i)\}$ , and  $E\{\cdot\}$  is the statistical expectation operator. Similar to (14),  $\bar{\mathbf{R}}$  and  $\tilde{\mathbf{R}}(i)$  can also be decomposed as

$$\bar{\mathbf{R}} = \bar{\mathbf{R}}_L + \bar{\mathbf{R}}_D + \bar{\mathbf{R}}_U \tag{19}$$

$$\tilde{\mathbf{R}}(i) = \tilde{\mathbf{R}}_L(i) + \tilde{\mathbf{R}}_D(i) + \tilde{\mathbf{R}}_U(i). \tag{20}$$

By substituting (17) into (8) and (18) into (9), the following equations are, respectively, written as

$$\mathbf{R}(n) = \sum_{i=1}^n \lambda^{n-i} \bar{\mathbf{R}} + \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{R}}(i) \tag{21}$$

$$\mathbf{p}(n) = \sum_{i=1}^n \lambda^{n-i} \bar{\mathbf{p}} + \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{p}}(i). \tag{22}$$

With (14), (19), and (20), the autocorrelation matrix in (21) can be written as

$$\begin{aligned} \mathbf{R}_L(n) + \mathbf{R}_D(n) + \mathbf{R}_U(n) &= \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_L + \bar{\mathbf{R}}_D + \bar{\mathbf{R}}_U) \\ &+ \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_L(i) + \tilde{\mathbf{R}}_D(i) + \tilde{\mathbf{R}}_U(i)] \end{aligned} \tag{23}$$

which can be decomposed into its lower triangular, diagonal, and upper triangular parts as

$$\mathbf{R}_L(n) = \sum_{i=1}^n \lambda^{n-i} \bar{\mathbf{R}}_L + \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{R}}_L(i) \tag{24}$$

$$\mathbf{R}_D(n) = \sum_{i=1}^n \lambda^{n-i} \bar{\mathbf{R}}_D + \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{R}}_D(i) \tag{25}$$

$$\mathbf{R}_U(n) = \sum_{i=1}^n \lambda^{n-i} \bar{\mathbf{R}}_U + \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{R}}_U(i). \tag{26}$$

Using the Eqs. (21), (24), and (25), the following equations can be formed.

$$\begin{aligned} \mathbf{R}_D(n) + \omega \mathbf{R}_L(n) &= \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L) \\ &+ \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i)] \end{aligned} \tag{27}$$

$$\begin{aligned} \mathbf{R}_D(n) + \omega \mathbf{R}_L(n) - \omega \mathbf{R}(n) &= \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) \\ &+ \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i) - \omega \tilde{\mathbf{R}}(i)] \end{aligned} \tag{28}$$

Let us define

$$\hat{\mathbf{w}}(n) = \bar{\mathbf{w}}(n) + \tilde{\mathbf{w}}(n) \tag{29}$$

where  $\bar{\mathbf{w}}(n) = E\{\hat{\mathbf{w}}(n)\}$  and  $\tilde{\mathbf{w}}(n)$  is the stochastic part of  $\hat{\mathbf{w}}(n)$ . Substituting (22), (27), (28), and (29) into the RSOR algorithm (16) gives

$$\left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L) + \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i)] \right\} [\bar{\mathbf{w}}(n+1) + \tilde{\mathbf{w}}(n+1)]$$

$$\begin{aligned}
 &= \left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) \right. \\
 &\quad \left. + \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i) - \omega \tilde{\mathbf{R}}(i)] \right\} [\bar{\mathbf{w}}(n) + \tilde{\mathbf{w}}(n)] \\
 &\quad + \omega \sum_{i=1}^n \lambda^{n-i} \bar{\mathbf{p}} + \omega \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{p}}(i). \tag{30}
 \end{aligned}$$

The deterministic part of (30) is written as

$$\begin{aligned}
 &\left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L) \right\} \bar{\mathbf{w}}(n+1) \\
 &= \left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) \right\} \bar{\mathbf{w}}(n) \\
 &\quad + \omega \sum_{i=1}^n \lambda^{n-i} \bar{\mathbf{p}} \tag{31}
 \end{aligned}$$

which is independent of the sums. Therefore, (31) can be reduced to

$$(\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L) \bar{\mathbf{w}}(n+1) = (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) \bar{\mathbf{w}}(n) + \omega \bar{\mathbf{p}}. \tag{32}$$

Now, let us define

$$\bar{\mathbf{w}}(n) = \mathbf{w}_o + \Delta \mathbf{w}(n) \tag{33}$$

where  $\mathbf{w}_o$  is the optimum solution of the normal equation. Writing the normal equation as

$$\bar{\mathbf{R}} \mathbf{w}_o = \bar{\mathbf{p}} \tag{34}$$

and using (33) and (34) in (32), we can write

$$\begin{aligned}
 &(\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L) [\mathbf{w}_o + \Delta \mathbf{w}(n+1)] \\
 &= (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) [\mathbf{w}_o + \Delta \mathbf{w}(n)] + \omega \bar{\mathbf{R}} \mathbf{w}_o. \tag{35}
 \end{aligned}$$

By multiplying both sides of (35) from the left by  $(\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1}$ , we obtained after some rearrangement

$$\Delta \mathbf{w}(n+1) = [\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}] \Delta \mathbf{w}(n). \tag{36}$$

The result in (36) is a time-invariant equation with a system matrix  $[\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]$ , and its solution is

$$\Delta \mathbf{w}(n+1) = [\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]^{n+1} \Delta \mathbf{w}(0). \tag{37}$$

According to the solution (37), if the maximum eigenvalue of the system matrix satisfies  $|\lambda_{\max}[\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]| <$

1, then  $[\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]^{n+1} \rightarrow \mathbf{0}_{M \times M}$  for  $0 < \omega < 2$ , ensuring that  $\Delta \mathbf{w}(n) \rightarrow \mathbf{0}_{M \times 1}$  as  $n \rightarrow \infty$ . Therefore, the definition (33) shows that the expected value of the filter weight vector converges to its optimum value as

$$\bar{\mathbf{w}}(n) = \mathbf{w}_o. \tag{38}$$

Thus, the RSOR algorithm is an unbiased parameter estimator for the optimal Wiener solution of the normal equation in the mean sense.

### 3.2 Mean-square convergence analysis

The stochastic part of (30) can be written as

$$\begin{aligned}
 &\left\{ \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i)] \right\} \tilde{\mathbf{w}}(n+1) \\
 &\quad + \left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L) \right. \\
 &\quad \left. + \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i)] \right\} \tilde{\mathbf{w}}(n+1) \\
 &= \left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) \right\} \tilde{\mathbf{w}}(n) \\
 &\quad + \left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) \right. \\
 &\quad \left. + \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i) - \omega \tilde{\mathbf{R}}(i)] \right\} \tilde{\mathbf{w}}(n) \\
 &\quad + \omega \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{p}}(i). \tag{39}
 \end{aligned}$$

Using definition (29), (39) can be rearranged as

$$\begin{aligned}
 &\left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L) \right\} \tilde{\mathbf{w}}(n+1) \\
 &= \left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) \right\} \tilde{\mathbf{w}}(n) \\
 &\quad - \left\{ \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i)] \right\} \hat{\mathbf{w}}(n+1) \\
 &\quad + \left\{ \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i) - \omega \tilde{\mathbf{R}}(i)] \right\} \hat{\mathbf{w}}(n) \\
 &\quad + \omega \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{p}}(i). \tag{40}
 \end{aligned}$$

Using (29) and (33), we can write the following equations

$$\hat{\mathbf{w}}(n + 1) = \mathbf{w}_o + \Delta \mathbf{w}(n + 1) + \tilde{\mathbf{w}}(n + 1) \tag{41}$$

$$\hat{\mathbf{w}}(n) = \mathbf{w}_o + \Delta \mathbf{w}(n) + \tilde{\mathbf{w}}(n). \tag{42}$$

If (41) and (42) are used in (40), then the last three terms on the right-hand side of (40) become

$$\begin{aligned} & - \left\{ \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i)] \right\} [\Delta \mathbf{w}(n + 1) + \tilde{\mathbf{w}}(n + 1)] \\ & + \left\{ \sum_{i=1}^n \lambda^{n-i} [\tilde{\mathbf{R}}_D(i) + \omega \tilde{\mathbf{R}}_L(i) - \omega \tilde{\mathbf{R}}(i)] \right\} [\Delta \mathbf{w}(n) + \tilde{\mathbf{w}}(n)] \\ & + \omega \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{p}}(i) - \omega \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{R}}(i) \mathbf{w}_o. \end{aligned} \tag{43}$$

Using (17), (18), and (34), the last term of (43) can be reduced as follows

$$\begin{aligned} & \omega \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{p}}(i) - \omega \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{R}}(i) \mathbf{w}_o \\ & = \omega \sum_{i=1}^n \lambda^{n-i} [\mathbf{x}(i)d(i) - \tilde{\mathbf{p}}] \\ & \quad - \omega \sum_{i=1}^n \lambda^{n-i} [\mathbf{x}(i)\mathbf{x}^T(i) - \tilde{\mathbf{R}}] \mathbf{w}_o \\ & = \omega \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) [d(i) - \mathbf{x}^T(i)\mathbf{w}_o] \\ & \quad - \omega \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{p}} + \omega \sum_{i=1}^n \lambda^{n-i} \tilde{\mathbf{R}} \mathbf{w}_o \\ & = \omega \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) [d(i) - \mathbf{x}^T(i)\mathbf{w}_o]. \end{aligned} \tag{44}$$

If we define the error as

$$e_o(i) = d(i) - \mathbf{x}^T(i)\mathbf{w}_o \tag{45}$$

and consider the orthogonality between  $\mathbf{x}(n)$  and  $e_o(n)$  [4], then the last term of (43) converges to the zero vector as

$$\omega \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)e_o(i) \rightarrow \mathbf{0}_{M \times 1}. \tag{46}$$

In addition, the first two terms in (43) can be considered as the weighted time averages of an ergodic process, and they are equal to their expected values by the ergodicity assumption. Based on the assumption of statistical independence between the  $\mathbf{x}(n)$  and  $\hat{\mathbf{w}}(n)$ , which is used in the analysis of the LMS algorithm [4], the first two terms in (43) can be considered

zero vectors. Consequently, the last three terms on the right-hand side of (40) converge to zero vectors, and therefore, according to (40), the stochastic part of (30) can be reduced to

$$\begin{aligned} & \left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L) \right\} \tilde{\mathbf{w}}(n + 1) \\ & = \left\{ \sum_{i=1}^n \lambda^{n-i} (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) \right\} \tilde{\mathbf{w}}(n). \end{aligned} \tag{47}$$

Removing the deterministic quantities in (47), it can be reduced to

$$(\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L) \tilde{\mathbf{w}}(n + 1) = (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L - \omega \bar{\mathbf{R}}) \tilde{\mathbf{w}}(n). \tag{48}$$

The following result is obtained by multiplying both sides of (48) by  $(\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1}$

$$\tilde{\mathbf{w}}(n + 1) = [\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}] \tilde{\mathbf{w}}(n). \tag{49}$$

To obtain the covariance matrix of the weight error vector, the equation in (49) is multiplied by its transpose:

$$\begin{aligned} & \tilde{\mathbf{w}}(n + 1) \tilde{\mathbf{w}}^T(n + 1) \\ & = [\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}] \tilde{\mathbf{w}}(n) \tilde{\mathbf{w}}^T(n) [\mathbf{I} \\ & \quad - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]^T. \end{aligned} \tag{50}$$

Defining  $\mathbf{K}(n) = E\{\tilde{\mathbf{w}}(n)\tilde{\mathbf{w}}^T(n)\}$  and by taking the statistical expectation of both sides of (50), the following result is obtained:

$$\begin{aligned} \mathbf{K}(n + 1) & = [\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}] \mathbf{K}(n) [\mathbf{I} \\ & \quad - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]^T. \end{aligned} \tag{51}$$

Similar to (36), the solution of (51) can be written as

$$\begin{aligned} \mathbf{K}(n + 1) & = [\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]^{n+1} \mathbf{K}(0) \{ [\mathbf{I} \\ & \quad - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]^T \}^{n+1}. \end{aligned} \tag{52}$$

According to the solution given in (52), if the maximum eigenvalue of the system matrix satisfies  $|\lambda_{\max}[\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]| < 1$ , then  $[\mathbf{I} - \omega (\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \bar{\mathbf{R}}]^{n+1} \rightarrow \mathbf{0}_{M \times M}$  for  $0 < \omega < 2$ , ensuring that  $\mathbf{K}(n) \rightarrow \mathbf{0}_{M \times M}$  as  $n \rightarrow \infty$  in the mean-square sense. In addition, by using both (37) and (52), we can see that the covariance matrix of the weight error vector,  $\mathbf{K}(n)$ , converges to the zero matrix much faster than the weight error vector in (37).

### 3.3 Excess of mean-square error and misadjustment

Considering the filter output error  $e(n) = d(n) - y(n)$  as defined in (4), let us denote the mean-squared error as  $J(n) = E[e^2(n)]$ . The MSE produced by an algorithm can be written as [4]

$$J(n) = J_{\min} + E \left\{ \tilde{\mathbf{w}}^T(n) \mathbf{x}(n) \mathbf{x}^T(n) \tilde{\mathbf{w}}(n) \right\}, \quad (53)$$

where  $J_{\min}$  is the minimum MSE produced by the optimum Wiener filter given as  $J_{\min} = \sigma_d^2 - \bar{\mathbf{p}}^T \mathbf{w}_o$  in which  $\sigma_d^2$  is the variance of the desired signal  $d(n)$ , and the excess of MSE defined by  $J_{\text{exc}}(n) = J(n) - J_{\min}$  [4] can be written for the RSOR algorithm to have the form

$$J_{\text{exc}}(n) = \text{trace}\{\bar{\mathbf{R}}\mathbf{K}(n)\}. \quad (54)$$

It is seen from (52) and (54) that  $J_{\text{exc}}(n)$ , and therefore the misadjustment, which is defined by  $M = J_{\text{exc}}(\infty)/J_{\min}$ , converges to zero as  $n \rightarrow \infty$ . Thus, similar to the RLS algorithm, the RSOR algorithm converges to zero excess of MSE value and zero misadjustment value.

### 3.4 Choice of filter parameters

According to (10)–(12), there are two initial parameters for the RSOR algorithm,  $\omega$  and  $\delta$ , whereas the RGS algorithm has one initial parameter,  $\delta$  [16]. Typical values of the  $\delta$  parameter are  $\delta = 1, 0.1, 0.01, \dots$ , and it is an initialized the autocorrelation as  $\mathbf{R}(0) = \delta \mathbf{I}_{M \times M}$ . Similar to the RGS algorithm,  $\delta$  can be used to control the convergence speed of the parameter estimation vector in the initial steps of the RSOR algorithm. The second parameter  $\omega$  is known as the relaxation parameter, and it can also be used to control the convergence speed of the RSOR algorithm. The initial parameters  $\omega$  and  $\delta$  both affect the convergence speed of the RSOR algorithm, and they must be chosen carefully in the initial steps of the algorithm. According to the analysis of the iteration matrix of the RSOR algorithm in (36), a more suitable condition for convergence can be established easily for uncorrelated input signals. When the autocorrelation matrix is decomposed as  $\bar{\mathbf{R}} = \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T$ , where the columns of  $\mathbf{Q}$  contain the eigenvectors of  $\bar{\mathbf{R}}$  and the diagonal matrix  $\mathbf{\Lambda}$  contains the eigenvectors of  $\bar{\mathbf{R}}$ , the iteration in (36) is written as

$$\Delta \mathbf{w}(n+1) = \left[ \mathbf{I} - \omega(\bar{\mathbf{R}}_D + \omega \bar{\mathbf{R}}_L)^{-1} \mathbf{Q}\mathbf{\Lambda}\mathbf{Q}^T \right] \Delta \mathbf{w}(n). \quad (55)$$

If the input signal  $\mathbf{x}(n)$  is uncorrelated and zero mean, the correlation matrix becomes  $\bar{\mathbf{R}} = \sigma_x^2 \mathbf{I}_{M \times M}$  and thus, (55) reduces to

$$\Delta \mathbf{w}'(n+1) = [\mathbf{I} - \omega(\sigma_x^2)^{-1}] \Delta \mathbf{w}'(n) \quad (56)$$

where  $\Delta \mathbf{w}'(n) = \mathbf{Q}^T \Delta \mathbf{w}(n)$  is the rotated weight error vector [5]. For the iteration (56) to converge, the absolute values of the diagonal elements of the iteration matrix must be less than 1:

$$\left| 1 - \omega(\sigma_x^2)^{-1} \lambda_i \right| < 1 \quad \text{for } i = 1, \dots, M. \quad (57)$$

Thus, the following result, which is the same as that for the accelerated EDS algorithm [14], is obtained from the above inequality for uncorrelated Gaussian input signals:

$$0 < \omega < \frac{2\sigma_x^2}{\lambda_{\max}} \quad (58)$$

where  $\lambda_{\max}$  is the maximum eigenvalue of  $\bar{\mathbf{R}}$ . The result in (58) is valid when an uncorrelated input signal is used, but can also be a guide for highly correlated input signals.

## 4 Simulation results: system identification example

By ensemble-averaged computer simulations, the performance of the RSOR algorithm is examined and compared to that of the NLMS algorithm, the APA, the recursive inverse (RI) algorithm [19], the RGS algorithm [16], and the RLS algorithm. The simulation studies are performed using a system identification problem described in [5]. The following optimum parameter vector is used for the unknown system impulse response

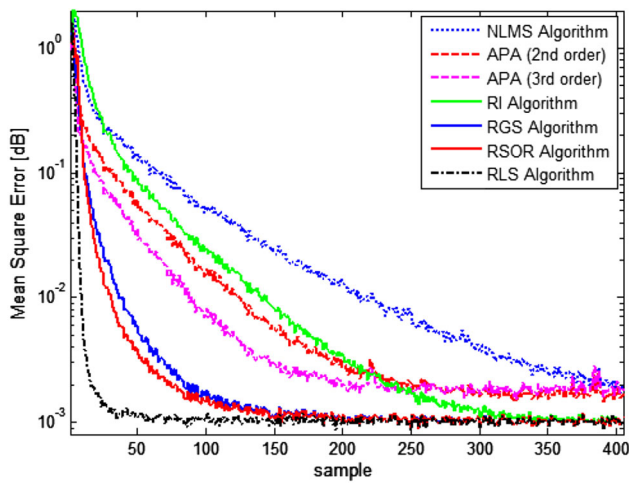
$$\mathbf{w}_o = [1.0 \ 0.9 \ 0.1 \ 0.2]^T. \quad (59)$$

The filter length of algorithm is  $M = 4$  taps, which is equal to the length of the unknown system.

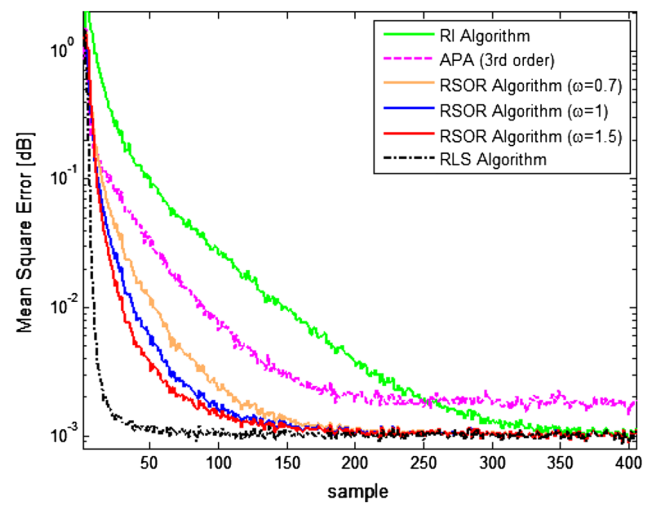
In the first simulation study for the system identification example, the excitation signal of the unknown system was generated using the following AR process:

$$x(n) = -1.20x(n-1) - 0.81x(n-2) + v(n) \quad (60)$$

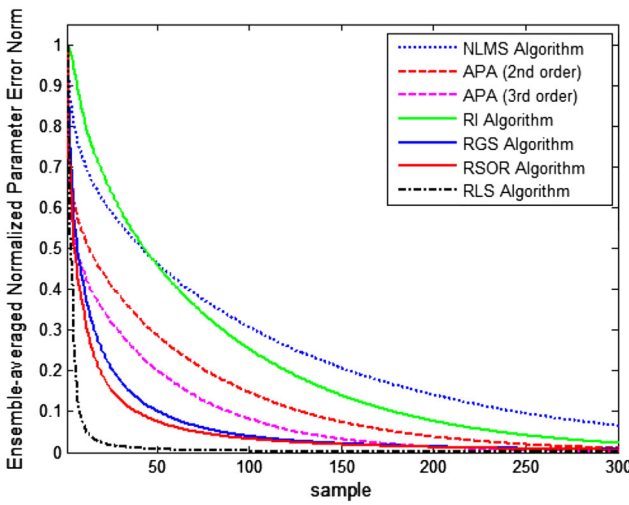
where  $v(n)$  is zero-mean white Gaussian noise sequence with variance  $\sigma_v^2 = 1$ . Thus, the system is excited with a zero-mean correlated input signal with variance  $\sigma_x^2 = 5.863$ . The eigenvalue spread of the autocorrelation matrix was computed as 58.38. The system response is corrupted by an additive white Gaussian signal, which is uncorrelated with the input signal  $x(n)$ . The SNR at the output of the system is 36 dB. The following initial parameters are used in the simulations. For the NLMS algorithm and the APA, the step size parameter  $\mu = 0.3$ . For the RI algorithm,  $\mu_0 = 0.00033$ ,  $\mu(n) = \mu_0/(1 - \lambda^n)$ ,  $\mathbf{R}(0) = \mathbf{0}_{M \times M}$ , and  $\mathbf{p}(0) = \mathbf{0}_{M \times 1}$ . For the RGS algorithm,  $\delta = 1$ ,  $\mathbf{R}(0) = \delta \mathbf{I}_{M \times M}$ , and  $\mathbf{p}(0) = \mathbf{0}_{M \times 1}$ . For the RSOR algorithm,  $\omega = 1.5$ ,  $\delta = 1$ ,



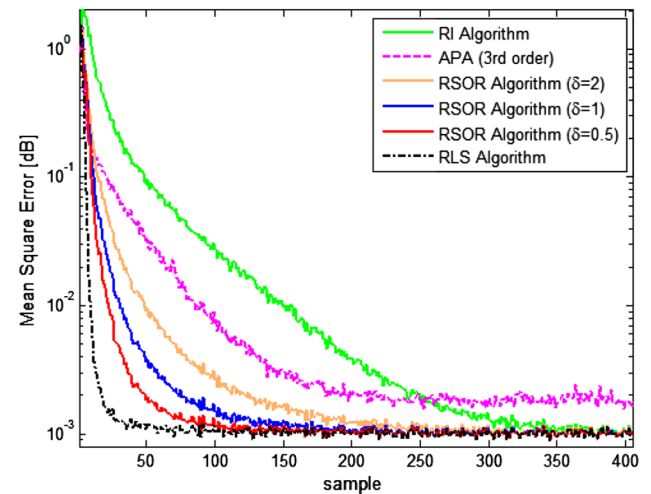
**Fig. 1** Comparison of the MSE curves of the NLMS, APA, RI, RGS, RLS, and RSOR algorithms ( $\omega = 1.5$ )



**Fig. 3** Comparison of the MSE curves for different  $\omega$  values of the RSOR algorithm and the other algorithms ( $\delta = 1$ )



**Fig. 2** Comparison of the normalized parameter error vector norms of the NLMS, APA, RI, RGS, RLS, and RSOR algorithms ( $\omega = 1.5$ )



**Fig. 4** Comparison of the MSE curves for different  $\delta$  values of the RSOR algorithm and the other algorithms ( $\omega = 1.4$ )

$\mathbf{R}(0) = \delta \mathbf{I}_{M \times M}$ , and  $\mathbf{p}(0) = \mathbf{0}_{M \times 1}$ . For the RLS algorithm,  $\mathbf{R}^{-1}(0) = 10 \mathbf{I}_{M \times M}$ . The forgetting factor is  $\lambda = 0.995$  in the RI, RGS, RLS, and RSOR algorithms. The initial value of the parameter estimation vector is the zero vector in all cases. All simulation results were obtained by averaging 800 simulations of all algorithms. Figure 1 shows the MSE curves for the algorithms used. Figure 2 shows the ensemble-averaged normalized values of the parameter error vector norms, defined as

$$\Delta = \|\hat{\mathbf{w}}(n) - \mathbf{w}_o\| / \|\mathbf{w}_o\| \tag{61}$$

where the vector norm is computed as  $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T \mathbf{w}}$ .

The second simulation is performed to show the effect of the  $\omega$  parameter for the constant  $\delta = 1$ . The MSE curves of the algorithms for different  $\omega$  values are shown in Fig. 3. The

third simulation study shows the effect of the  $\delta$  parameter for the constant  $\omega = 1.4$ . The MSE curves of the algorithms for different  $\delta$  values are shown in Fig. 4. The initial parameter values for the second and third simulations are the same as those for the first simulation.

The ensemble-averaged simulation results in Figs. 1 and 2 show that the RSOR algorithm produces results that are very close to the RLS results and better results than the results obtained by the gradient-based algorithms. The RSOR algorithm has a slightly better convergence rate than the RGS algorithm for  $\omega > 1$ . Figure 2 also shows that the normalized parameter error vector of the RSOR algorithm converges to zero vector, i.e., the parameter estimation vector converges to its optimum value. Also, the ensemble-averaged simulation results in Figs. 1, 3, and 4 show that the RSOR algorithm

approaches the same minimum MSE value with the RLS algorithm, i.e., zero excess of MSE and misadjustment value.

## 5 Conclusion

In this paper, as a useful indication of performance, a stochastic convergence analysis of parameter estimation vector obtained by the RSOR algorithm was presented in the mean and mean-square sense. It is shown that the RSOR algorithm gives an unbiased estimate to optimum Wiener solution of normal equation, and the obtained parameter estimations are convergent in the mean and mean-square sense with zero excess of MSE and zero misadjustment. The convergence analysis results were verified by ensemble-averaged computer simulations. The simulation results showed that the RSOR algorithm has better convergence performance than the gradient-based methods, a slightly better convergence rate for  $\omega > 1$  with a slightly higher computational complexity than the RGS algorithm, and comparable results obtained by the RLS algorithm.

## References

- Widrow, B., Stearns, S.D.: Adaptive Signal Processing. Prentice Hall, Englewood Cliffs (1985)
- Nagumo, J.I., Noda, A.: A learning method for system identification. *IEEE Trans. Autom. Control* **12**(3), 282–287 (1967)
- Ozeki, K., Umeda, T.: An adaptive filtering algorithm using an orthogonal projection to an affine subspace and its properties. *Electron. Commun. Jpn.* **67–A**(5), 19–27 (1984)
- Haykin, S.: Adaptive Filter Theory, 5th edn. Prentice Hall, Upper Saddle River (2013)
- Diniz, P.S.R.: Adaptive Filtering: Algorithms and Practical Implementation, 4th edn. Springer, New York (2013)
- Koçal, O.H.: A new approach to least-squares adaptive filtering. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS'98), Monterey, California, USA, vol. 5, 31 May–3 June 1998, pp. 261–264
- Xu, G.F., Bose, T., Schroeder, J.: Channel equalization using an Euclidean direction search based adaptive algorithm. In: Proceedings of Global Telecommunication Conference (GLOBECOM 1998), Sydney, NSW, vol. 6, 8–12 Nov. 1998, pp. 3479–3484
- Mathurasai, T., Bose, T., Etter, D.M.: Decision feedback equalization using an Euclidean direction based adaptive algorithm. In: Proceedings of 33th Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA, USA, vol. 1, 24–27 Oct. 1999, pp. 519–523
- Xu, G.F., Bose, T., Kober, W., Thomas, J., Fast, A.: Adaptive algorithm for image restoration. *IEEE Trans. Circuits Syst. I Fundam. Theory Appl.* **46**(1), 216–220 (1999)
- Bose, T., Xu, G.F.: The Euclidean direction search algorithm in adaptive filtering. *IEICE Trans. Fundam. Electron. Commun. Comput. Sci.* **E85–A**(3), 532–539 (2002)
- Li, C., Liao, X., Yu, J.: A fast complex-valued adaptive filtering algorithm. *Int. J. Gen. Syst.* **31**(4), 395–403 (2002)
- Mabey, G.W., Gunther, J., Bose, T.: A Euclidean direction based algorithm for blind source separation using a natural gradient. In: Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'04), vol. 5, 17–21 May 2004, pp. 561–564
- Zhao, C., Abeysekera, S.S.: Investigation of Euclidean direction search based Laguerre adaptive filter for system identification. In: Proceedings of the International Conference on Communications, Circuits and Systems, vol. 2, 27–30 May 2005, pp. 691–695
- Ahmad, N.A.: Accelerated Euclidean direction search algorithm and related relaxation schemes for solving adaptive filtering problem. In: Proceedings of the 6th International Conference on Information, Communications and Signal Processing (ICICS 2007), Singapore, 10–13 Dec. 2007, pp. 1–5
- Zhang, Z., Bose, T., Gunther, J.: A unified framework for least square and mean square based adaptive filtering algorithms. In: Proceedings of IEEE International Symposium on Circuits and Systems (ISCAS 2005), vol. 5, 23–26 May 2005, pp. 4325–4328
- Hatun, M., Koçal, O.H.: Recursive Gauss–Seidel algorithm for direct self-tuning control. *Int. J. Adapt. Control Signal Process.* **26**(5), 435–450 (2012)
- Hatun, M., Koçal, O.H.: Recursive successive over-relaxation algorithm for adaptive filtering. In: Proceedings of Mosharaka International Conference on Communications, Computers and Applications (MIC-CCA 2012), İstanbul, Turkey, 12–14 Oct. 2012, pp. 90–95
- Golub, G.H., Van Loan, C.F.: Matrix Computations, 3rd edn. John Hopkins University Press, Baltimore (1996)
- Ahmad, M.S., Kukrer, O., Hocanin, A.: Recursive inverse adaptive filtering algorithm. *Digit. Signal Process.* **21**(4), 491–496 (2011)
- Ahmad, M.S., Kukrer, O., Hocanin, A.: A 2-D recursive inverse adaptive algorithm. *Signal Image Video Process.* **7**(2), 221–226 (2013)
- Johnson Jr., C.R.: Lectures on Adaptive Parameter Estimation. Prentice Hall, Englewood Cliffs (1988)