

## LEAKAGE ANALYSIS OF THE RSOR ALGORITHM

Metin HATUN \*

Received: 03.10.2020; revised: 29.03.2021; accepted: 09.04.2021

**Abstract:** The RSOR algorithm is a recursive algorithm that has been proposed as an alternative to the RLS algorithm for updating adaptive filter parameters. As with other algorithms, the forgetting factor, filter length and relaxation parameter significantly affects the performance of the RSOR algorithm. In this study, using an adaptive FIR filter in system identification mode, the effect of forgetting factor, filter length and relaxation parameter on the leakage phenomenon of the RSOR algorithm was analyzed. For this purpose, firstly, the effect of measurement noise on the adaptive filter output, namely the leakage phenomenon, was explained analytically, and then the influence of the forgetting factor and other filter parameters on this leakage phenomenon was examined. The results obtained from the simulation studies are compared with similar algorithms.

**Keywords:** Adaptive Filter, System Identification, Leakage Phenomenon, Forgetting Factor

### RSOR Algoritmasının Sızıntı Analizi

**Öz:** RSOR algoritması, uyarlamalı filtre parametrelerini güncellemek için RLS algoritmasına alternatif olarak önerilmiş olan tekrarlamalı bir algoritmadır. Diğer algoritmelerde olduğu gibi, unutma faktörü, filtre uzunluğu ve gevşetme parametresi RSOR algoritmasının performansını önemli ölçüde etkilemektedir. Bu çalışmada, bir uyarlamalı FIR filtre sistem tanıma modunda kullanılarak unutma faktörünün, filtre uzunluğunun ve gevşetme parametresinin RSOR algoritmasındaki sızıntı olayına etkisi incelenmiştir. Bu amaçla, öncelikle ölçme gürültüsünün uyarlamalı filtre çıkışına etkisi, yani sızıntı olayı, analitik olarak açıklanmış, sonra unutma faktörünün ve diğer filtre parametrelerinin bu sızıntı olayına etkisi incelenmiştir. Yapılan benzetim çalışmalarıyla elde edilen sonuçlar, benzer algoritmalar ile karşılaştırılmıştır.

**Anahtar Kelimeler:** Uyarlamalı Filtre, Sistem Tanıma, Sızıntı Olayı, Unutma Faktörü

## 1. INTRODUCTION

The RLS (Recursive Least Squares) algorithm is preferred in many applications due to its high convergence speed despite its high computational load (Haykin, 2002, Diniz, 2013). On the other hand, some alternative algorithms have been proposed: These are the RSOR (Recursive SOR) algorithm based on the single-step SOR (Successive Over-Relaxation) iteration (Hatun and Koçal, 2012), and the RI (Recursive Inverse) algorithm based on the one-step gradient iteration (Ahmad et al., 2011a). In these type algorithms, the forgetting factor, which is a positive number and taken as close to 1, is used for tracking parameter changes. If the forgetting factor is chosen less than 1, the parameter tracking capability of the algorithm increases, but the stability and misadjustment value of the algorithm are affected negatively. If the forgetting factor approaches 1, the stability and misadjustment value of the algorithm changes positively, but in this case, the parameter tracking capability decreases.

\* Electrical-Electronics Engineering Department, Faculty of Engineering, Bursa Uludağ University, 16059, Bursa, Turkey

Adaptive filters are used in system identification mode in many applications. In the system identification mode, the estimated values of the adaptive filter parameters converge to their correct values, and the measurement noise can be recovered from the signal estimation error. This result indicates that the system identification process is working correctly. However, in some cases, the measurement noise corrupts the output signal of the filter. In this case, which was called as “the leakage phenomenon”, the system identification process does not work properly (Paleologu et al., 2008). The mathematical expression of the leakage signal can simply be derived, but the results obtained between different algorithms may not be the same. The leakage term depends on the forgetting factor, filter length, and other filter parameters, if any.

The leakage phenomenon is a fundamental problem which affects the adaptive filtering applications negatively by increasing the misadjustment. For example, the leakage signal causes a residual error and inefficient cancellation of noise signal in adaptive noise cancellation applications, or imperfect rejection of echo signal in echo cancellation applications (Haykin, 2002, Paleologu et al., 2008, Ciochină et al., 2009).

The effects of forgetting factor and filter length on the leakage phenomenon of RLS and RI algorithms have been studied in detail (Ciochină et al., 2009, Ahmad et al., 2011b). In this article, it is aimed to perform a similar leakage analysis for the RSOR algorithm.

In this paper: The leakage signal for the RSOR algorithm was formulated and analyzed depending on the forgetting factor and the other filter parameters in section 2. The simulation results obtained for the RSOR algorithm were compared with the RLS and RI algorithms in section 3, and some conclusions were given in section 4.

## 2. THE LEAKAGE PHENOMENON

### 2.1. System Identification Using The RSOR Algorithm

In system identification process given in Figure 1, an algorithm adjusts the parameters of the adaptive filter. The system output and the adaptive filter output are given as

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

$$\hat{y}(n) = \mathbf{x}^T(n)\hat{\mathbf{w}}(n). \quad (2)$$

The parameter vectors of the system and adaptive filter, and the data input vector are defined as follows, respectively:

$$\mathbf{w} = [w_0 \quad w_1 \quad \cdots \quad w_{M-1}]^T \quad (3)$$

$$\hat{\mathbf{w}}(n) = [\hat{w}_0(n) \quad \hat{w}_1(n) \quad \cdots \quad \hat{w}_{M-1}(n)]^T, \quad (4)$$

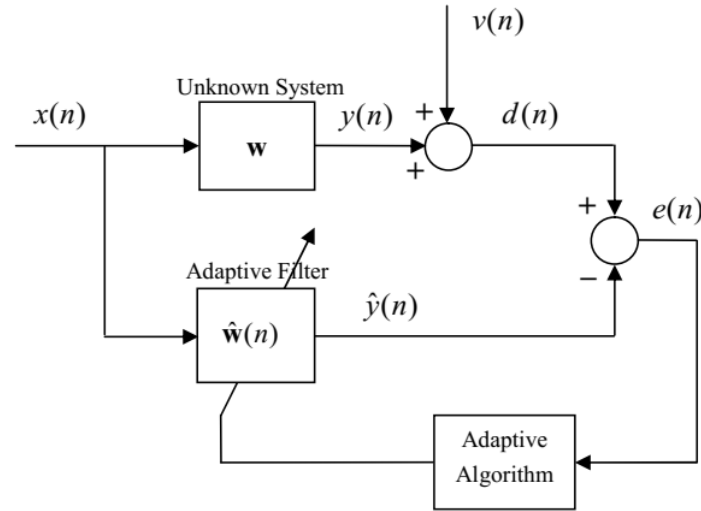
$$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad \cdots \quad x(n-M+1)]^T, \quad (5)$$

where  $M$  is the filter length and  $x(n)$  is the input signal. The desired signal  $d(n)$  is tracked by the output of the filter, and it is written as

$$d(n) = y(n) + v(n) = \mathbf{x}^T(n)\mathbf{w} + v(n), \quad (6)$$

where  $v(n)$  is the measurement noise with a zero-mean and variance  $\sigma_v^2$ , and independent with the input signal  $x(n)$ . The error signal  $e(n)$  is defined as

$$e(n) = d(n) - \hat{y}(n) = [y(n) - \hat{y}(n)] + v(n) = -\mathbf{x}^T(n)[\hat{\mathbf{w}}(n) - \mathbf{w}] + v(n) \quad (7)$$



**Figure 1:**  
System identification diagram

The RSOR algorithm minimizes the least-squares error function

$$V(\mathbf{w}, n) = \sum_{i=1}^n \lambda^{n-i} e^2(i) = \sum_{i=1}^n \lambda^{n-i} [d(i) - \mathbf{x}^T(i)\hat{\mathbf{w}}(i)]^2 \quad (8)$$

and its minimization gives the optimal parameter vector  $\hat{\mathbf{w}}(n) = \mathbf{R}^{-1}(n)\mathbf{p}(n)$ , which is the solution of the normal equation

$$\mathbf{R}(n)\hat{\mathbf{w}}(n) = \mathbf{p}(n) . \quad (9)$$

The estimates for  $\mathbf{R}(n)$  and  $\mathbf{p}(n)$  can be computed as

$$\mathbf{R}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i) \quad , \quad \mathbf{p}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)d(i) \quad (10)$$

but they can be updated using the following recursive equations in practical:

$$\mathbf{R}(n) = \lambda \mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n) \quad (11)$$

$$\mathbf{p}(n) = \lambda \mathbf{p}(n-1) + \mathbf{x}(n)d(n) \quad (12)$$

The RSOR algorithm solves the normal equation (9) using one SOR iteration during a sampling interval as follows (Hatun and Koçal, 2012, 2017).

$$\hat{w}_i(n+1) = \left[ p_i(n) - \sum_{j=1}^{i-1} R_{ij}(n)\hat{w}_j(n+1) - \sum_{j=i+1}^M R_{ij}(n)\hat{w}_j(n) \right] \frac{\omega}{R_{ii}(n)} + (1-\omega)\hat{w}_i(n) \quad (13)$$

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

Equations (11), (12) and (13) are used for implementation of the RSOR algorithm. The parameter  $\omega$  is known as the relaxation parameter and should be taken between  $0 < \omega < 2$  for the stability of the SOR iteration. If  $\omega > 1$  is chosen, the SOR iteration faster converges than the Gauss-Seidel iteration (Golub and Van Loan, 1996).

## 2.2. Quantitative Expression of The Leakage Phenomenon for The RSOR Algorithm

The RSOR algorithm in scalar updating form (13) can be represented in vector updating form as follows (Hatun and Koçal, 2012).

$$\hat{\mathbf{w}}(n+1) = \hat{\mathbf{w}}(n) + \omega[\mathbf{R}_L(n) + \omega\mathbf{R}_D(n)]^{-1}[\mathbf{p}(n) - \mathbf{R}(n)\hat{\mathbf{w}}(n)] \quad (14)$$

which is based on the following splitting as given in the classical SOR method (Golub and Van Loan, 1996).

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

A convergence analysis for the RSOR algorithm can be performed in a more tractable manner by decomposing the symmetric correlation matrix  $\mathbf{R}(n)$  to its lower triangle, diagonal and upper triangle parts as follows

$$\mathbf{R}(n) = \mathbf{R}_L(n) + \mathbf{R}_D(n) + \mathbf{R}_U(n) . \quad (16)$$

Defining the parameter estimation errors as a vector:

$$\tilde{\mathbf{w}}(n) = \hat{\mathbf{w}}(n) - \mathbf{w} \quad (17)$$

and subtracting the correct parameter vector,  $\mathbf{w}$  from the both sides of (14), the following iteration is obtained for the parameter error vector after some arrangements (Hatun and Koçal, 2012).

$$\begin{aligned} \tilde{\mathbf{w}}(n+1) = & \{\mathbf{I} - \omega[\mathbf{R}_L(n) + \omega\mathbf{R}_D(n)]^{-1}\mathbf{R}(n)\} \tilde{\mathbf{w}}(n) \\ & + \omega[\mathbf{R}_L(n) + \omega\mathbf{R}_D(n)]^{-1}[\mathbf{p}(n) - \mathbf{R}(n)\mathbf{w}] \end{aligned} \quad (18)$$

By rewriting the vector quantity in the second term as

$$\mathbf{p}(n) - \mathbf{R}(n)\mathbf{w} = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)d(i) - \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)\mathbf{x}^T(i)\mathbf{w} = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)v(i) \quad (19)$$

and substituting in (18) the following result is obtained for the parameter error vector.

$$\begin{aligned}\tilde{\mathbf{w}}(n+1) &= \{\mathbf{I} - \omega[\mathbf{R}_L(n) + \omega\mathbf{R}_D(n)]^{-1}\mathbf{R}(n)\} \tilde{\mathbf{w}}(n) \\ &+ \omega[\mathbf{R}_L(n) + \omega\mathbf{R}_D(n)]^{-1} \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)v(i)\end{aligned}\quad (20)$$

A similar analysis gives the following iteration for the RI algorithm (Ahmad et al., 2011b, Salman et al., 2017).

$$\tilde{\mathbf{w}}(n+1) = \{\mathbf{I} - \mu(n)\mathbf{R}(n)\} \tilde{\mathbf{w}}(n) + \mu(n) \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)v(i) \quad (21)$$

The following equation is also given for the RLS algorithm (Ciochină et al., 2009).

$$\tilde{\mathbf{w}}(n) = \mathbf{R}^{-1}(n) \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)v(i) \quad (22)$$

Considering (18), the error is represented as

$$e(n) = -\mathbf{x}^T(n)\tilde{\mathbf{w}}(n) + v(n) \quad (23)$$

and thus, the leakage term is introduced as (Ciochină et al., 2009, Ahmad et al., 2011b).

$$\begin{aligned}r(n) &= \hat{y}(n) - y(n) = \hat{y}(n) - [d(n) - v(n)] = \hat{y}(n) - [\hat{y}(n) + e(n) - v(n)] \\ &= \mathbf{x}^T(n)\tilde{\mathbf{w}}(n)\end{aligned}\quad (24)$$

The usage of (20) in (24) gives an estimate of the leakage term for the RSOR algorithm based on the data used and the filter parameters, and this term  $r(n)$ , which is caused by  $v(n)$ , leaks to the output signal of the filter

$$\hat{y}(n) = y(n) + r(n) = y(n) + \mathbf{x}^T(n)\tilde{\mathbf{w}}(n) . \quad (25)$$

On the other hand, using (6) and (10), the vector  $\mathbf{p}(n)$  is represented as

$$\mathbf{p}(n) = \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)[\mathbf{x}^T(i)\mathbf{w} + v(i)] = \mathbf{R}(n)\mathbf{w} + \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)v(i) \quad (26)$$

and thus, the normal equation (9) is rewritten as

$$\mathbf{R}(n)\hat{\mathbf{w}}(n) = \mathbf{R}(n)\mathbf{w} + \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i)v(i) \quad (27)$$

This equation also gives the parameter error vector in (22) for the RLS algorithm. Assuming  $\lambda$  is close to 1 and  $n$  is high enough, the following assumption can be written.

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) v(i) \cong E\{\mathbf{x}(n) v(n)\} = \mathbf{0}_{M \times 1} \quad (28)$$

Using this assumption, when time index goes to infinity or a large value, as  $n \rightarrow \infty$ , the normal equation (27) becomes

$$\mathbf{R}(n) \widehat{\mathbf{w}}(n) \cong \mathbf{R}(n) \mathbf{w} \quad (29)$$

and consequently, the parameter estimations vector  $\widehat{\mathbf{w}}(n)$  is close to correct parameters  $\mathbf{w}$ , namely,  $\widehat{\mathbf{w}}(n) \cong \mathbf{w}$ . Thus, the filter output gives the system output as  $\hat{y}(n) \cong y(n)$ , and the measurement noise can be close to the error signal as  $e(n) \cong v(n)$ . Thus, according to the leakage definition in (24), the measurement noise does not produce a leakage. Under the assumption (28), the parameter error iteration (20) is reduced to

$$\tilde{\mathbf{w}}(n+1) = \{\mathbf{I} - \omega[\mathbf{R}_L(n) + \omega \mathbf{R}_D(n)]^{-1} \mathbf{R}(n)\} \tilde{\mathbf{w}}(n). \quad (30)$$

If  $\lambda < 1$  or  $n$  is not high enough, the second term in (27) is not equal to zero vector, and therefore, the time-averaged normal equation (27) cannot be reduced to (29) and the parameter error iteration cannot be reduced to (30). According to (17), the parameter estimations have fluctuations around the correct parameter values. Thus, according to the leakage definition in (24), the measurement noise  $v(n)$  causes to a leakage term  $r(n)$ . If there is a leakage term  $r(n) \neq 0$ , which leaks to the adaptive filter output as given in (25), the measurement noise cannot be close to the error signal, i.e.,  $e(n) \neq v(n)$ , and thus, an accurate result cannot be obtained by solving the normal equation. The leakage signal is affected from low  $\lambda$  values, and higher values of  $M$  (Ciochină et al., 2009, Ahmad et al., 2011b). If no leakage occurs, then the system identification process works properly, thereby letting  $r(n) = 0$ .

### 2.3. Analysis of The Leakage Signal for The RSOR Algorithm

Taking statistical expectation of the normal equation (27), the following is written

$$E\{\mathbf{R}(n) \widehat{\mathbf{w}}(n)\} = E\{\mathbf{R}(n) \mathbf{w}\} + E\left\{\sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) v(i)\right\} \quad (31)$$

and using the assumption in (28), this equation can be reduced to

$$E\{\mathbf{R}(n) \widehat{\mathbf{w}}(n)\} = E\{\mathbf{R}(n) \mathbf{w}\}. \quad (32)$$

Based on the independence assumption, which considers adaptive parameters are statistically independent from the input signal (Haykin, 2002), the following approximation can be written

$$E\{\mathbf{R}(n)\} E\{\widehat{\mathbf{w}}(n)\} = E\{\mathbf{R}(n)\} \mathbf{w}. \quad (33)$$

By performing a statistical expectation analysis, it can also be shown that the RSOR algorithm produces  $E\{\widehat{\mathbf{w}}(n)\} = \mathbf{w}$  as  $n \rightarrow \infty$ , asymptotically. Taking statistical expectation of (20), the parameter error iteration for the RSOR algorithm can be written as

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

$$+ E \left\{ \omega [\mathbf{R}_L(n) + \omega \mathbf{R}_D(n)]^{-1} \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) v(i) \right\} \quad (34)$$

Assuming the expected value of the correlation matrix estimation converges to its deterministic value at steady-state:

$$\mathbf{R} = E\{\mathbf{R}(n)\} \cong \lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \mathbf{x}^T(i) \quad (35)$$

the following approximations can be written

$$\begin{aligned} E\{\mathbf{I} - \omega [\mathbf{R}_L(n) + \omega \mathbf{R}_D(n)]^{-1} \mathbf{R}(n)\} &\cong \mathbf{I} - \omega [\mathbf{R}_L + \omega \mathbf{R}_D]^{-1} \mathbf{R} \\ E\{\omega [\mathbf{R}_L(n) + \omega \mathbf{R}_D(n)]^{-1}\} &\cong \omega [\mathbf{R}_L + \omega \mathbf{R}_D]^{-1}. \end{aligned} \quad (36)$$

Using these approximations in (34), the following result is obtained.

$$\begin{aligned} \tilde{\mathbf{w}}(n+1) &\cong E\{\mathbf{I} - \omega [\mathbf{R}_L + \omega \mathbf{R}_D]^{-1} \mathbf{R}\} E\{\tilde{\mathbf{w}}(n)\} \\ &\quad + \omega [\mathbf{R}_L + \omega \mathbf{R}_D]^{-1} E \left\{ \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) v(i) \right\} \end{aligned} \quad (37)$$

Under the assumption in (28), and considering the input signal  $x(n)$  is uncorrelated with the measurement noise  $v(n)$ , the iteration (33) is reduced to

$$\tilde{\mathbf{w}}(n+1) \cong E\{\mathbf{I} - \omega [\mathbf{R}_L + \omega \mathbf{R}_D]^{-1} \mathbf{R}\} E\{\tilde{\mathbf{w}}(n)\}. \quad (38)$$

If all eigenvalues of the RSOR algorithm are smaller than 1, this iteration converges to zero vector asymptotically as  $n \rightarrow \infty$  (Hatun and Koçak, 2012, 2017). Thus, the expected value of the parameter estimates for the RSOR algorithm converges to the correct parameter vector (3), namely,  $E\{\hat{\mathbf{w}}(n)\} = \mathbf{w}$  as  $n \rightarrow \infty$ , asymptotically.

A stochastic convergence analysis of the leakage signal can be performed by taking statistical expectation of the leakage term in (24). Considering the independence assumption, i.e., the adaptive parameters are statistically independent from the input signal, the following approximation can be written

$$E\{r(n)\} = E\{\mathbf{x}^T(n) \tilde{\mathbf{w}}(n)\} \cong E\{\mathbf{x}^T(n)\} E\{\tilde{\mathbf{w}}(n)\} \quad (39)$$

Because of the input signal is zero mean, the leakage term converges to zero if the parameter vector converges to zero vector or not zero vector.

A more complicated noise sequence can be also considered as the measurement noise  $v(n)$ , for example a contaminated Gaussian noise with its mean  $v_m$  as

$$v(n) = v_m + v_g(n) + v_{imp}(n) = v_m + v_g(n) + b(n)v_w(n) \quad (40)$$

where  $v_{imp}(n)$  is known as Bernoulli-Gaussian impulsive noise sequence, the  $v_g(n)$  and  $v_w(n)$  are independent Gaussian noise sequences with zero means and variances  $\sigma_g^2$  and  $\sigma_w^2$ , respectively. The  $b(n)$  is a switching sequence which has zeros and ones, and modelled by Bernoulli random process with probability  $Pr[b(n) = 1] = p_r$  and  $Pr[b(n) = 0] = 1 - p_r$ . The variance of  $v(n)$  is  $\sigma_v^2 = \sigma_g^2 + \sigma_{imp}^2 = \sigma_g^2 + p_r \sigma_w^2$ , and the ratio  $r = \sigma_{imp}^2 / \sigma_g^2 = p_r \sigma_w^2 / \sigma_g^2$  determines the impulsive behavior of  $v(n)$  by taking  $\sigma_w^2 > \sigma_g^2$  (Chan and Zou, 2004). The following equation can be written for the following correlation vector in (37)

$$\begin{aligned} E \left\{ \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) v(i) \right\} &= E \left\{ \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) [v_m + v_g(i) + v_{imp}(i)] \right\} \\ &= v_m E \left\{ \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) \right\} + E \left\{ \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) v_g(i) \right\} + E \left\{ \sum_{i=1}^n \lambda^{n-i} \mathbf{x}(i) v_{imp}(i) \right\} = \mathbf{0}_{M \times 1} \end{aligned} \quad (41)$$

The expected values of the second and third sums in (41) become zero vector because of the input signal  $x(n)$  is independent with the measurement noises  $v_g(n)$  and  $v_w(n)$ . The first term is also becomes zero vector if the input signal  $x(n)$  is a zero mean sequence. Consequently, according to (39), the expected value of the leakage term becomes zero, i.e.,  $E\{r(n)\} = 0$ , as  $n \rightarrow \infty$ , asymptotically. Thus, taking expected value of (23), the following result is obtained:

$$E\{e(n)\} = E\{v(n)\} - E\{r(n)\} = v_m + E\{v_g(n)\} + E\{v_{imp}(n)\} - E\{r(n)\} = v_m \quad (42)$$

This result shows that  $E\{r(n)\} = 0$  and  $E\{e(n)\} = E\{v(n)\} = v_m$  in the mean sense, and verified in the following section by computer simulations.

### 3. SIMULATION RESULTS

In this section, adaptive filtering algorithms were used to identify the impulse response of a sample system defined by the correct parameter vector (Diniz, 2013, page 111):

$$\mathbf{w} = [0.1 \quad 0.3 \quad 0.0 \quad -0.2 \quad -0.4 \quad -0.7 \quad -0.4 \quad -0.2]^T,$$

where the filter length  $M = 8$ . The parameter estimations vector  $\hat{\mathbf{w}}(n)$  and the data vector  $\mathbf{x}(n)$  can be formed as follows, respectively:

$$\hat{\mathbf{w}}(n) = [\hat{w}_0(n) \quad \hat{w}_1(n) \quad \hat{w}_2(n) \quad \hat{w}_3(n) \quad \hat{w}_4(n) \quad \hat{w}_5(n) \quad \hat{w}_6(n) \quad \hat{w}_7(n)]^T,$$

$$\mathbf{x}(n) = [x(n) \quad x(n-1) \quad x(n-2) \quad x(n-3) \quad x(n-4) \quad x(n-5) \quad x(n-6) \quad x(n-7)]^T.$$

A low frequency sine wave,  $v(n) = 0.5 \sin(0.01\pi n T_s)$  with a sampling interval  $T_s = 1$  sec., was added to the output signal measurements in the first three simulations to see if the measurement noise can be recovered from the estimation error and to observe the leakage signal. Thus, it can be observed whether the error is close to the noise. In the simulations, the RSOR algorithm was compared with the RLS and RI algorithms. The following initial values were used for initial phase of the algorithms: RLS:  $\mathbf{R}^{-1}(0) = \mathbf{I}$ ; RSOR:  $\mathbf{R}(0) = \mathbf{I}$ ,  $\mathbf{p}(0) = \mathbf{0}$ ,  $\omega = 1.5$ ; RI:  $\mathbf{R}(0) = \mathbf{I}$ ,  $\mathbf{p}(0) = \mathbf{0}$  and  $\mu(n) = \mu_0 / (1 - \lambda^n)$  with different  $\mu_0$  values. The initial parameter estimates were taken as zero for all algorithms. In order to demonstrate and compare the leakage phenomenon for the used algorithms, the error signal  $e(n)$  and the residual errors  $e_r(n) = e(n) - v(n) = -r(n)$  are plotted in time domain.



In the first simulation, the effect of the leakage phenomenon were analyzed and compared for the RLS, RSOR and RI algorithms with different forgetting factors and constant filter length ( $M = 8$ ). The obtained results were given in Figure 2 for the RLS algorithm, in Figure 3 for the RSOR algorithm, and in Figure 4 for the RI algorithm. The leakage values of the used algorithms for different  $\lambda$  values (with  $M = 8$ ) were compared in Table 1. The leakage values in all the tables were computed by averaging the last 2000 values of  $|r(n)| = |\mathbf{x}^T(n)\tilde{\mathbf{w}}(n)|$  obtained in the simulations.

In the second simulation, the effect of the leakage phenomenon were analyzed and compared for the RLS, RSOR and RI algorithms with different filter lengths and constant forgetting factor ( $\lambda = 0.96$ ). The obtained results were given in Figure 5 for the RLS algorithm, in Figure 6 for the RSOR algorithm, and in Figure 7 for the RI algorithm. The leakage values of the used algorithms for different  $M$  values (with  $\lambda = 0.96$ ) were given in Table 2.

In the third simulation, the effect of the leakage phenomenon were analyzed and compared for the RSOR algorithm with different values of relaxation parameter  $\omega$  with constant filter length ( $M = 8$ ) and constant forgetting factor ( $\lambda = 0.9$ ). The obtained results were given in Figure 8. The leakage values of the used algorithms for different  $\omega$  values (with  $\lambda = 0.9$  and  $M = 8$ ) were presented in Table 3.

**Table 1. The leakage values of the used algorithms for several  $\lambda$  values (  $M = 8$  )**

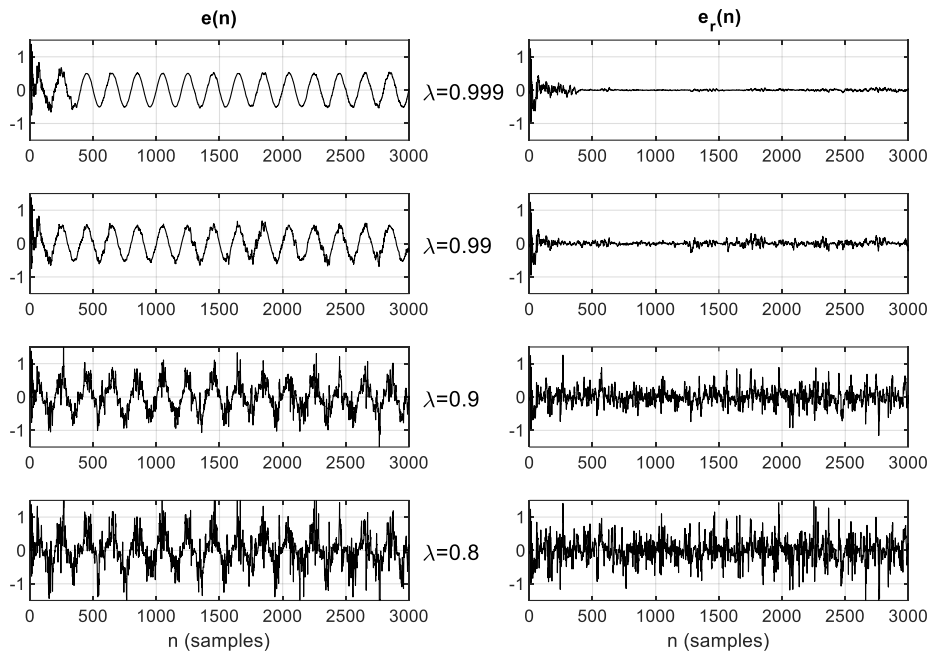
Algorithm:	$\lambda = 0.999$	$\lambda = 0.99$	$\lambda = 0.9$	$\lambda = 0.8$
RLS	0.015041	0.047036	0.164015	0.227187
RSOR	0.015063	0.046385	0.150525	0.197423
RI	0.015018	0.045671	0.110981	0.115101

**Table 2. The leakage values of the used algorithms for several  $M$  values (  $\lambda = 0.96$  )**

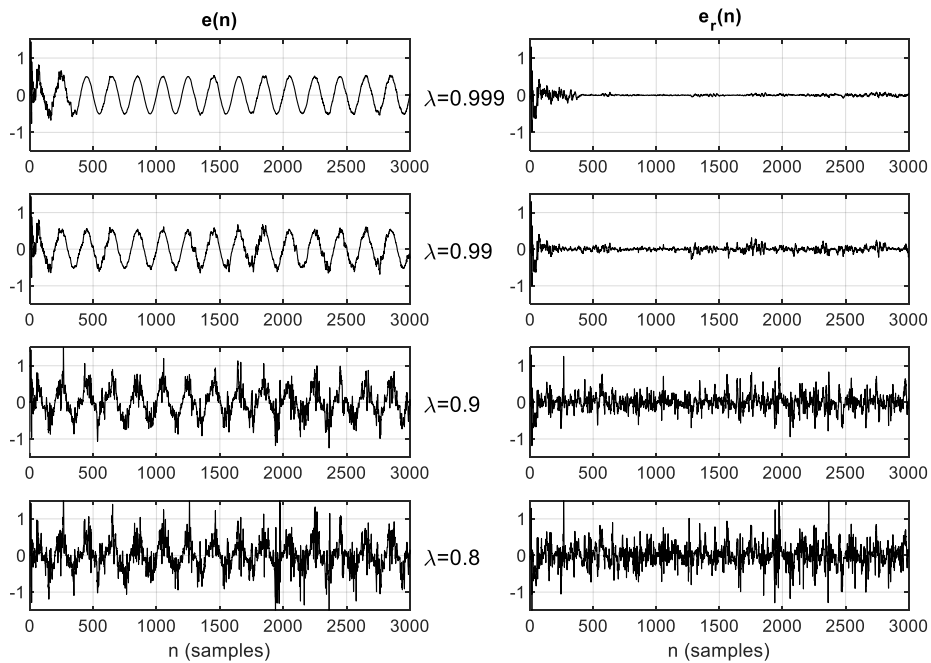
Algorithm:	$M = 8$	$M = 16$	$M = 32$	$M = 64$
RLS	0.101257	0.148180	0.237372	0.347601
RSOR	0.099850	0.143821	0.226781	0.322038
RI	0.088250	0.126342	0.186874	0.243867

**Table 3. The leakage values of the RSOR algorithm for several  $\omega$  values (  $M = 8$  and  $\lambda = 0.96$  )**

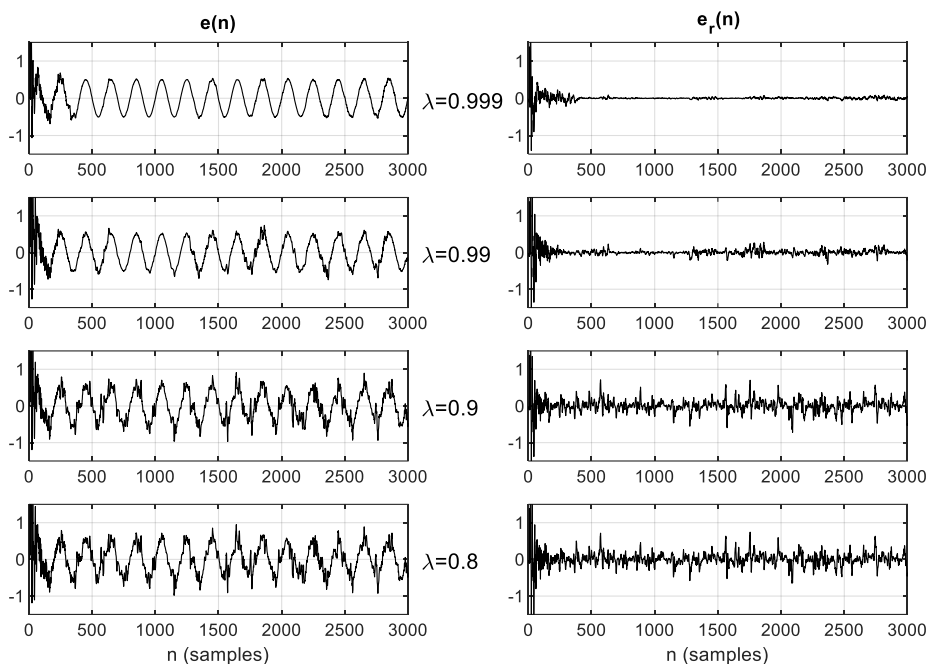
Algorithm:	$\omega = 0.7$	$\omega = 1.0$	$\omega = 1.3$	$\omega = 1.7$
RSOR	0.191218	0.166343	0.154733	0.147789



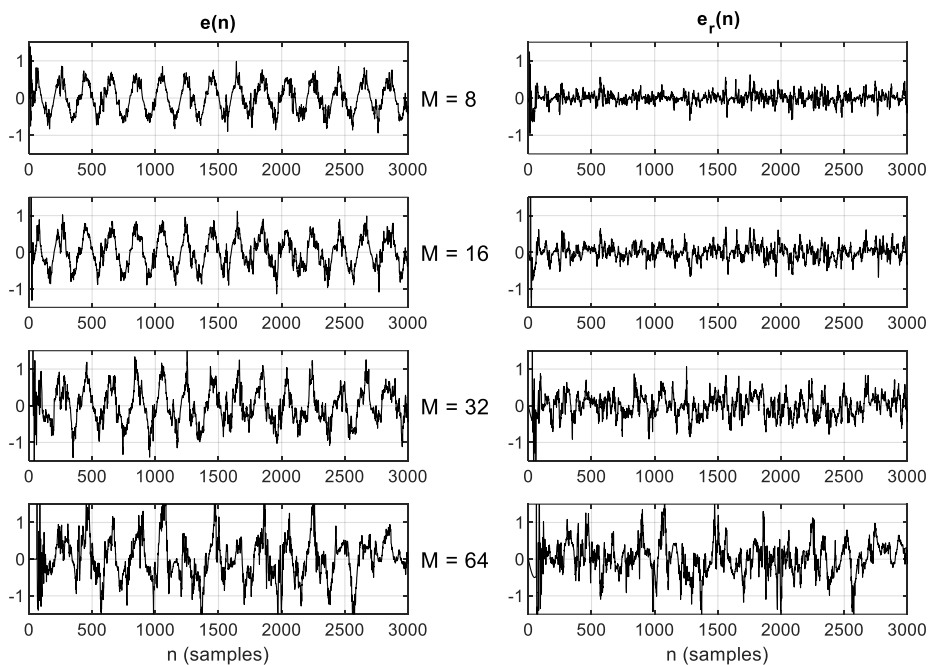
**Figure 2:**  
The leakage effect of the RLS algorithm for several  $\lambda$  values (  $M = 8$  )



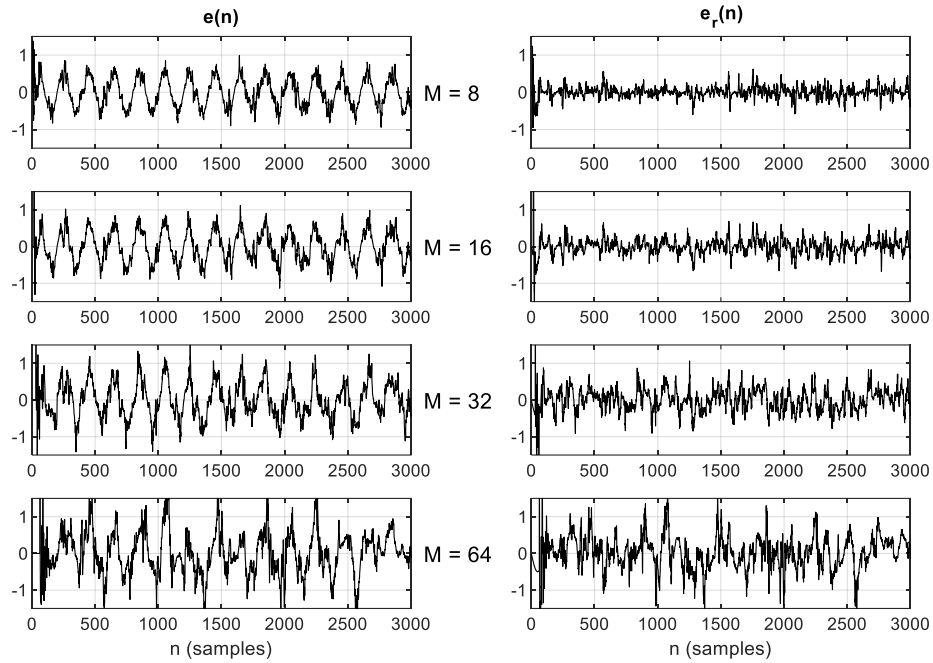
**Figure 3:**  
The leakage effect of the RSOR algorithm for several  $\lambda$  values (  $M = 8$  )



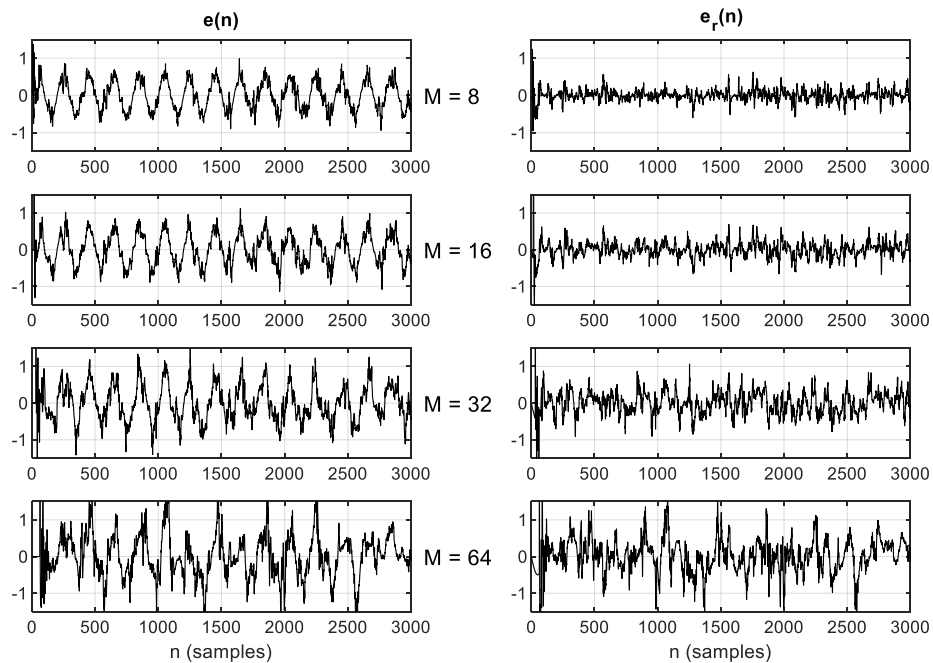
**Figure 4:**  
The leakage effect of the RI algorithm for several  $\lambda$  values ( $M = 8$ )



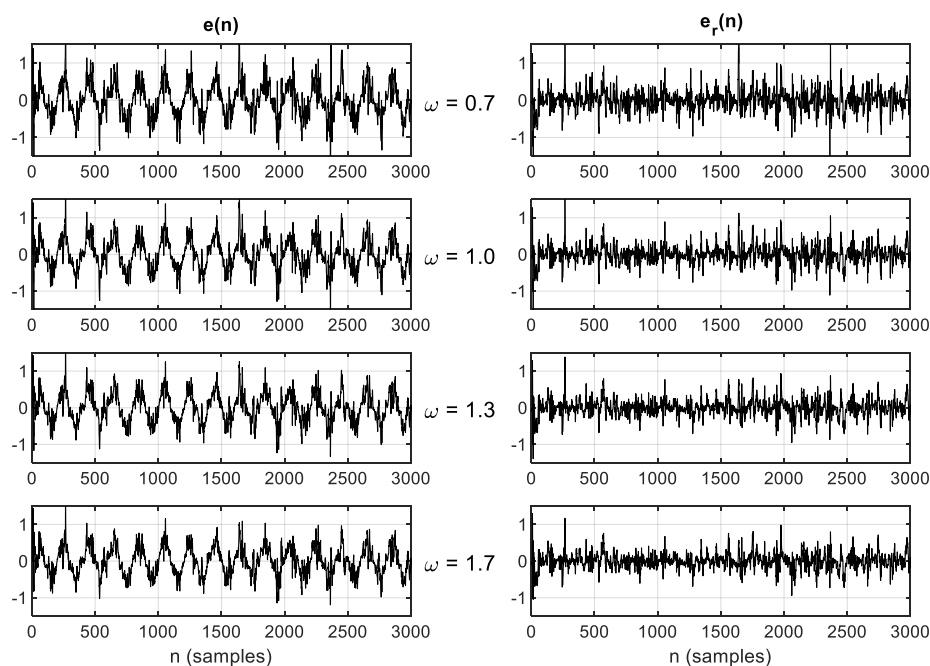
**Figure 5:**  
The leakage effect of the RLS algorithm for several  $M$  values ( $\lambda = 0.96$ )



**Figure 6:**  
The leakage effect of the RSOR algorithm for several  $M$  values ( $\lambda = 0.96$ )



**Figure 7:**  
The leakage effect of the RI algorithm for several  $M$  values ( $\lambda = 0.96$ )

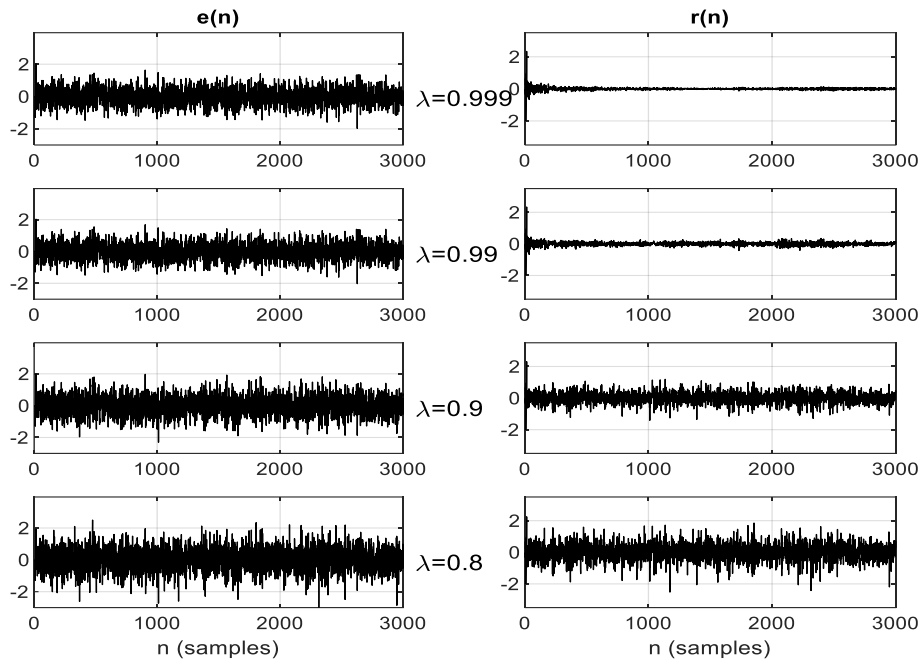


**Figure 8:**  
The leakage effect of the RSOR algorithm for several  $\omega$  values ( $\omega = 0.9$  and  $M = 8$ )

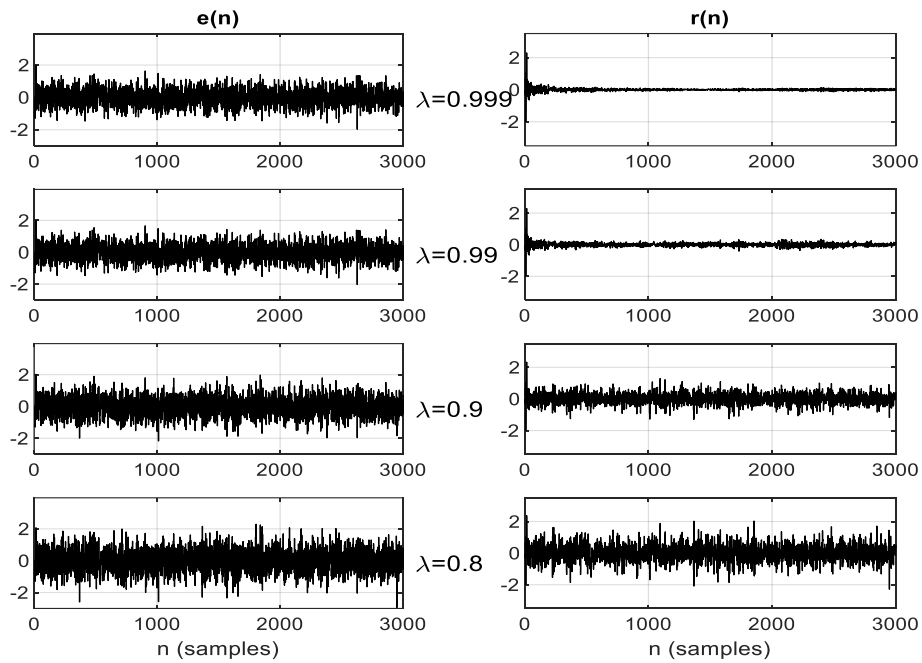
In the fourth simulation, a zero mean Gaussian noise  $v(n) = v_g(n)$  with variance  $\sigma_g^2 = 0.25$  was added to the output signal measurements, and the RSOR algorithm was compared with the RLS and RI algorithms. The following initial values were used: RLS:  $\mathbf{R}^{-1}(0) = \mathbf{I}$ ; RSOR:  $\mathbf{R}(0) = \mathbf{I}$ ,  $\mathbf{p}(0) = \mathbf{0}$ ,  $\omega = 1.2$ ; RI:  $\mathbf{R}(0) = \mathbf{I}$ ,  $\mathbf{p}(0) = \mathbf{0}$  and  $\mu(n) = \mu_0 / (1 - \lambda^n)$  with different  $\mu_0$  values. The initial parameter estimates were taken as zero, and the filter length was taken as  $M = 8$  for all algorithms. The error signal  $e(n)$  and the leakage signal  $r(n) = v(n) - e(n)$  were given in Figure 9 for the RLS algorithm, in Figure 10 for the RSOR algorithm, and in Figure 11 for the RI algorithm. The leakage values of the algorithms for different  $\lambda$  values were compared in Table 4.

**Table 4. The leakage values of the used algorithms for several  $\lambda$  values and  $v(n)$  is a zero mean Gaussian noise sequence with variance  $\sigma_v^2 = 0.25$**

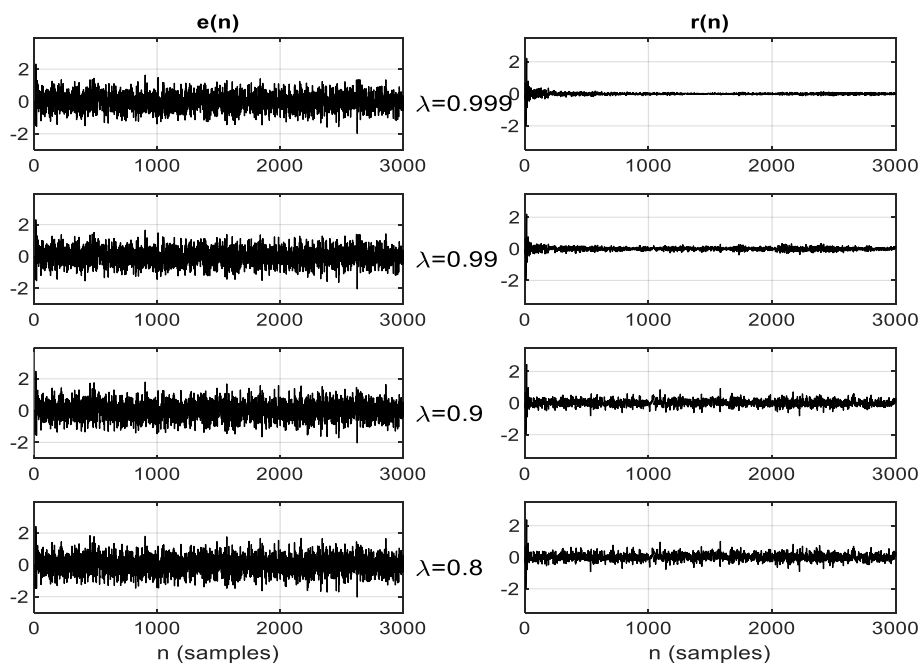
Algorithm:	$\lambda = 0.999$	$\lambda = 0.99$	$\lambda = 0.9$	$\lambda = 0.8$
RLS	0.030298	0.072645	0.255265	0.392667
RSOR	0.030310	0.072518	0.247006	0.371958
RI	0.030228	0.068678	0.146084	0.164779



**Figure 9:**  
*The leakage analysis of the RLS algorithm for several  $\lambda$  values and  $v(n)$  is a zero mean Gaussian noise sequence with variance  $\sigma_v^2 = 0.25$*



**Figure 10:**  
*The leakage analysis of the RSOR algorithm for several  $\lambda$  values and  $v(n)$  is a zero mean Gaussian noise sequence with variance  $\sigma_v^2 = 0.25$*

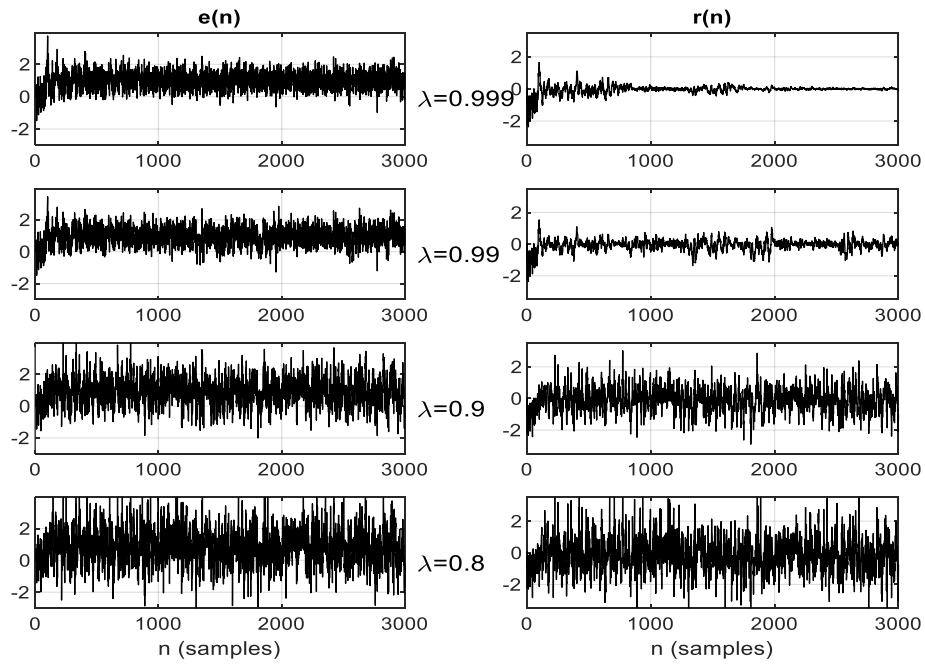


**Figure 11:**  
*The leakage analysis of the RI algorithm for several  $\lambda$  values and  $v(n)$  is a zero mean Gaussian noise sequence with variance  $\sigma_v^2 = 0.25$*

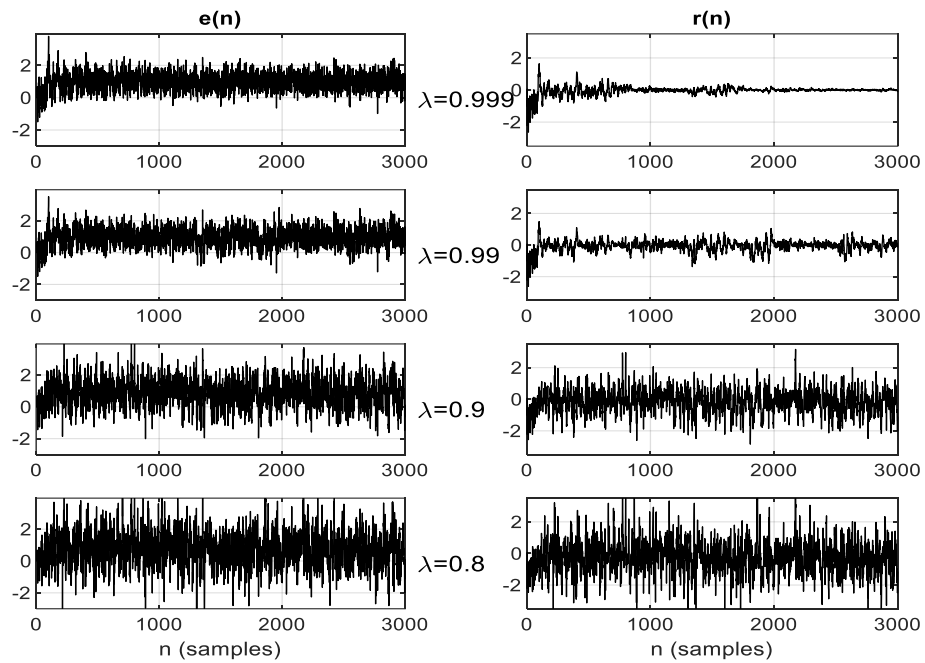
In the fifth simulation, a Gaussian noise  $v(n) = v_m + v_g(n)$  with mean  $v_m = 1$  and variance  $\sigma_g^2 = 0.25$  was added to the output signal measurements. The same initial values were used with the previous simulation. The error signal  $e(n)$  and the leakage signal  $r(n) = v(n) - e(n)$  were given in Figure 12 for the RLS algorithm, in Figure 13 for the RSOR algorithm, and in Figure 14 for the RI algorithm. The obtained leakage values of the algorithms for different  $\lambda$  values were compared in Table 5.

**Table 5. The leakage values of the used algorithms for several  $\lambda$  values and  $v(n)$  is a Gaussian noise sequence with mean  $v_m = 1$  and variance  $\sigma_v^2 = 0.25$**

Algorithm:	$\lambda = 0.999$	$\lambda = 0.99$	$\lambda = 0.9$	$\lambda = 0.8$
RLS	0.065357	0.219315	0.580859	0.820238
RSOR	0.065446	0.219234	0.567595	0.793798
RI	0.065289	0.214251	0.449477	0.507706

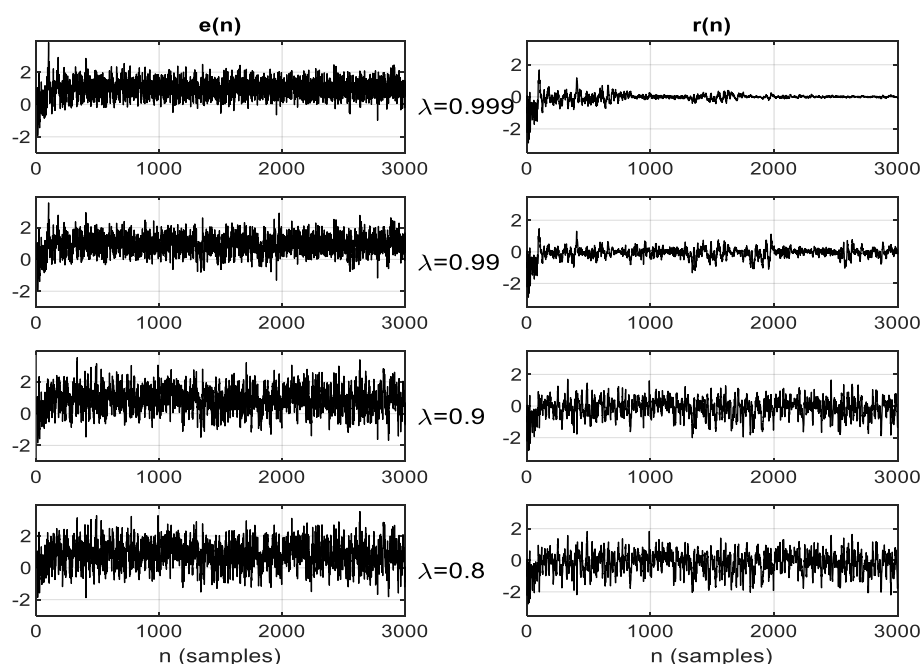


**Figure 12:**  
 The leakage analysis of the RLS algorithm for several  $\lambda$  values and  $v(n)$  is a Gaussian noise sequence with mean  $v_m = 1$  and variance  $\sigma_v^2 = 0.25$



**Figure 13:**  
 The leakage analysis of the RSOR algorithm for several  $\lambda$  values and  $v(n)$  is a Gaussian noise sequence with mean  $v_m = 1$  and variance  $\sigma_v^2 = 0.25$



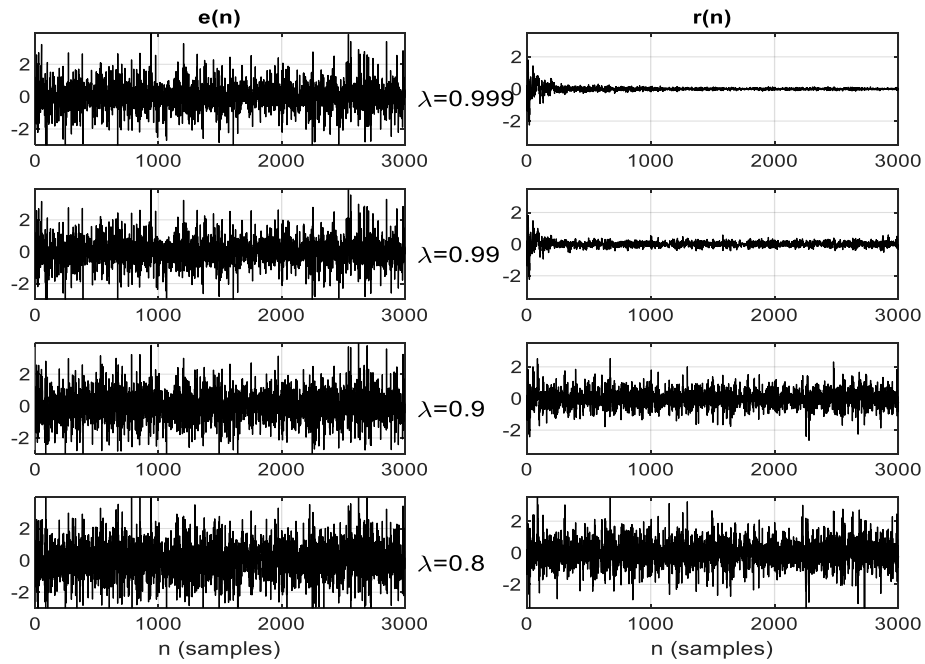


**Figure 14:**  
*The leakage analysis of the RI algorithm for several  $\lambda$  values and  $v(n)$  is a Gaussian noise sequence with mean  $v_m = 1$  and variance  $\sigma_v^2 = 0.25$*

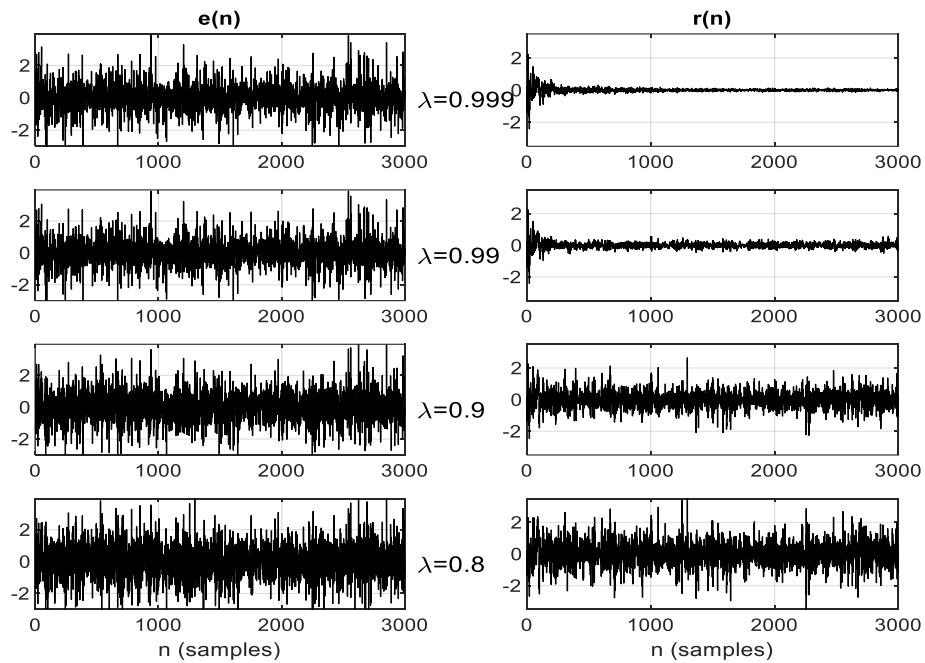
In the sixth simulation, a zero mean impulsive noise formed by  $v(n) = v_g(n) + b(n)v_w(n)$  was added to the output signal measurements. The  $v_g(n)$  and  $v_w(n)$  are independent zero mean Gaussian noise sequences with variances  $\sigma_g^2 = 0.25$  and  $\sigma_w^2 = 1.0$ , and thus, the variance of the measurement noise is  $\sigma_v^2 = \sigma_g^2 + p_r\sigma_w^2 = 0.55$ . A switching sequence  $b(n)$ , which is modelled by Bernoulli random process with probability  $Pr[b(n) = 1] = p_r = 0.3$  and  $Pr[b(n) = 0] = 1 - p_r = 0.7$ , was used. The same initial values were used in the simulation. The obtained results for the error signal  $e(n)$  and the leakage signal  $r(n) = v(n) - e(n)$  were given in Figure 15 for the RLS algorithm, in Figure 16 for the RSOR algorithm, and in Figure 17 for the RI algorithm. The leakage values of the algorithms for different  $\lambda$  values were given in Table 6.

**Table 6. The leakage values of the used algorithms for several  $\lambda$  values and  $v(n)$  is a zero mean impulsive noise sequence with variance  $\sigma_v^2 = 0.55$**

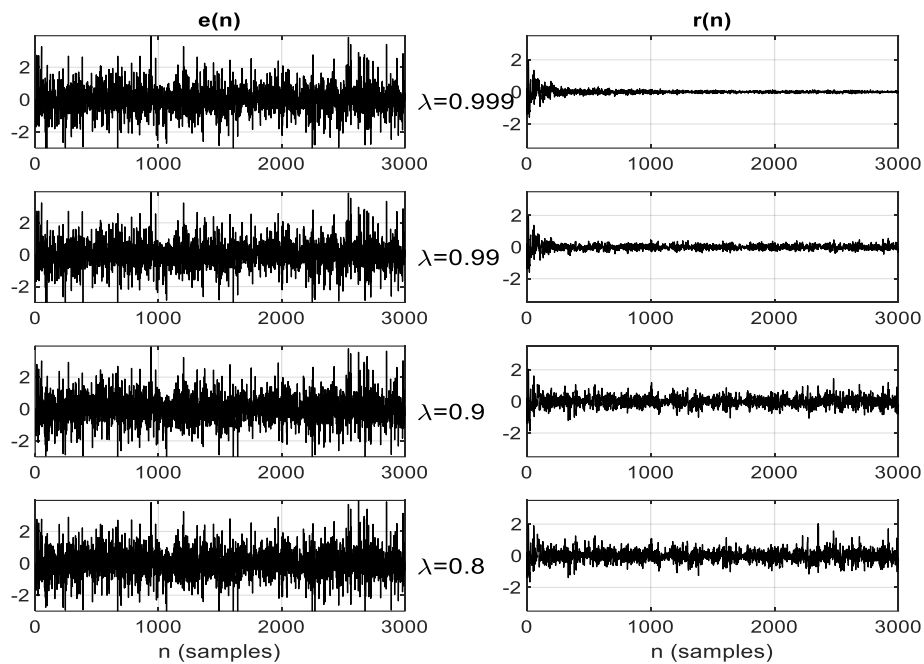
Algorithm:	$\lambda = 0.999$	$\lambda = 0.99$	$\lambda = 0.9$	$\lambda = 0.8$
RLS	0.038638	0.109522	0.387810	0.586417
RSOR	0.038479	0.108594	0.375635	0.542810
RI	0.038167	0.100655	0.225300	0.257336



**Figure 15:**  
*The leakage analysis of the RLS algorithm for several  $\lambda$  values and  $v(n)$  is a zero mean impulsive noise sequence with variance  $\sigma_v^2 = 0.55$*



**Figure 16:**  
*The leakage analysis of the RSOR algorithm for several  $\lambda$  values and  $v(n)$  is a zero mean impulsive noise sequence with variance  $\sigma_v^2 = 0.55$*



**Figure 17:**  
*The leakage analysis of the RI algorithm for several  $\lambda$  values and  $v(n)$  is a zero mean impulsive noise sequence with variance  $\sigma_v^2 = 0.55$*

The following findings were obtained from the simulation results for the RSOR algorithm:

- A sine wave is used as the measurement noise in the first three simulations, and it was observed that the measurement noise can be recovered from the estimation error when the variance of the leakage signal is low.
- The value of leakage increases if the forgetting factor and the filter length are increase.
- The value of leakage decreases if the relaxation parameter is increases.
- The leakage value of the RSOR algorithm is lower than the RLS algorithm and higher than the RI algorithm. The RI algorithm has a decreasing step size parameter and therefore produces a lower leakage value in steady state.
- A zero mean Gaussian noise sequence was used as the measurement noise in the fourth simulation and similar results were obtained.
- In the fifth simulation, a non-zero mean Gaussian noise was used as the measurement noise and similar results were obtained. However, higher leakage values were obtained in all algorithms used compared to the fourth simulation.
- A zero mean impulsive noise sequence was used as the measurement noise in the sixth simulation and similar results were obtained. But, although a measurement noise with a higher variance was used, lower leakage values were obtained in all algorithms compared to the fifth simulation.

#### 4. CONCLUSIONS

In this paper, a quantitative representation for the estimation of the leakage phenomenon in the RSOR algorithm was presented. The leakage analysis was performed in the system identification setup. It was shown that the quantity of the leakage is proportional to the forgetting factor, the filter length, and the relaxation parameter. The theoretical results were verified by computer simulations. The obtained results have shown that the forgetting factor

should be chosen close to 1 to avoid leakage phenomenon in all the algorithms used. In general, the leakage quantity has increased when the filter length was increased. In the RSOR algorithm, increasing of the relaxation parameter reduced the amount of leakage. Comparative simulations have shown that the RSOR algorithm have produced lower leakage value than RLS algorithm, but higher than RI algorithm.

### CONFLICT OF INTEREST

Author approve that to the best of their knowledge, there is not any conflict of interest or common interest with an institution/organization or a person that may affect the review process of the paper.

### AUTHOR CONTRIBUTION

Metin HATUN: determining the concept and design process of the research and research management; data collection and analysis; data analysis and interpretation of results.

### REFERENCES

1. Ahmad, M.S., Kukrer, O., Hocanin, A. (2011a) Recursive inverse adaptive filtering algorithm, *Digital Signal Processing*, 21(4), 491-496. doi: 10.1016/j.dsp.2011.03.001
2. Ahmad, M.S., Kukrer, O., Hocanin, A. (2011b) The effect of the forgetting factor on the RI adaptive algorithm in system identification, *International Symposium on Signals, Circuits and Systems (ISSCS 2011)*, Iasi, Romania, 1-4. doi: 10.1109/ISSCS.2011.5978751
3. Chan, S.-C., Zou, Z.-X. (2004) A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: fast algorithm and convergence performance analysis. *IEEE Transactions on Signal Processing*, 52(4), 975-991. doi: 10.1109/TSP.2004.823496
4. Ciochină, S., Paleologu, C., Benesty, J., Enescu, A.A. (2009) On the influence of the forgetting factor of the RLS adaptive filter in system identification, *International Symposium on Signals, Circuits and Systems (ISSCS 2009)*, Iasi, Romania, 1-4. doi: 10.1109/ISSCS.2009.5206117
5. Diniz, P.S.R. (2013) *Adaptive Filtering: Algorithms and Practical Implementation (4th ed.)*, Springer, New York.
6. Golub, G.H., Van Loan, C.F. (1996) *Matrix Computations (3rd ed.)*, John Hopkins University Press, Baltimore.
7. Hatun, M., Koçal, O.H. (2012) Recursive successive over-relaxation algorithm for adaptive filtering, *The 5th International Conference on Communications, Computers and Applications (MIC-CCA2012)*, İstanbul, Turkey, 90-95.
8. Hatun, M., Koçal, O.H. (2017) Stochastic convergence analysis of recursive successive over-relaxation algorithm in adaptive filtering, *Signal, Image and Video Processing*, 11(1), 137-144. doi: 10.1007/s11760-016-0912-7
9. Haykin, S. (2002) *Adaptive Filter Theory (4th ed.)*, Prentice-Hall, New Jersey.
10. Paleologu, C., Benesty, J. Ciochină, S. (2008) A robust variable forgetting factor recursive least-squares algorithm for system identification, *IEEE Signal Processing Letters*, 15, 597-600. doi: 10.1109/LSP.2008.2001559
11. Salman, M.S., Kukrer, O., Hocanin, A. (2017) Recursive inverse algorithm: mean-square-error analysis, *Digital Signal Processing*, 66, 10-17. doi: 10.1016/j.dsp.2017.04.001