A Practical Overview of Recursive Least-Squares Algorithms for Echo Cancellation

Camelia Elisei-Iliescu, Constantin Paleologu, Cristian Stanciu, Cristian Anghel, Silviu Ciochină

University Politehnica of Bucharest, Romania
Email: {pale,cristian,canghel,silviu}@comm.pub.ro

Jacob Benesty

INRS-EMT
University of Quebec, Montreal, Canada
Email: benesty@emt.inrs.ca

Abstract—**Due to its fast convergence rate, the recursive least-squares (RLS) algorithm is very popular in many applications of adaptive filtering. However, the computational complexity of this algorithm represents a major limitation in some applications that involve long filters, like echo cancellation. Moreover, the specific features of this application require good tracking capabilities and double-talk robustness for the adaptive algorithm, which further imply an optimization process on its parameters. In the case of most RLS-based algorithms, the performance can be controlled in terms of two main parameters, i.e., the forgetting factor and the regularization term. In this paper, we outline the influence of these parameters on the overall performance of the RLS algorithm and present several solutions to control their behavior, taking into account the specific requirements of echo cancellation application. The resulting variable forgetting factor RLS (VFF-RLS) and variable-regularized RLS (VR-RLS) algorithms could represent appealing solutions for real-world scenarios, as indicated by simulations performed in the context of both network and acoustic echo cancellation.**

Keywords–Adaptive filters; Echo cancellation; Recursive least-squares (RLS) algorithm; Variable forgetting factor RLS (VFF-RLS); Variable regularized RLS (VR-RLS).

I. Introduction

The recursive least-squares (RLS) algorithm [1], [2], [3] is one of the most popular adaptive filters. As compared to the normalized least-mean-square (NLMS) algorithm [2], [3], the RLS offers a superior convergence rate especially for highly correlated input signals. Of course, there is a price to pay for this advantage, which is an increase in the computational complexity. For this reason, it is not very often involved in echo cancellation [4], [5], where long filters are required.

In both network and acoustic echo cancellation contexts [4], [5], the basic principle is to build a model of the echo path impulse response that needs to be identified with an adaptive filter, which provides at its output a replica of the echo (that is further subtracted from the reference signal). The main difference between these two applications is the way in which the echo arises. In the network (or electrical) echo problem, there is an unbalanced coupling between the 2-wire and 4-wire circuits which results in echo, while the acoustic echo is due to the acoustic coupling between the microphone and the loudspeaker (e.g., as in speakerphones). However, in both cases, the adaptive filter has to model an unknown system, i.e., the echo path. The system model for echo cancellation is summarized in Section II.

Even if the formulation of the echo cancellation problem is straightforward, its specific features represent a challenge for any adaptive algorithm. There are several issues associated with this application, and they are as follows. First, the echo paths can have excessive lengths in time, e.g., up to hundreds of milliseconds. Consequently, long length adaptive filters are required (hundreds or even thousands of coefficients), influencing the convergence rate of the algorithm. Besides, the echo paths are time-variant systems, requiring good tracking capabilities for the echo canceller. Second, the echo signal is combined with the near-end signal; ideally, the adaptive filter should separate this mixture and provide an estimate of the echo at its output as well as an estimate of the near-end from the error signal. This is not an easy task since the near-end signal can contain both the background noise and the near-end speech; the background noise can be non-stationary and strong while the near-end speech acts like a large level disturbance. Last but not least, the input of the adaptive filter (i.e., the far-end signal) is mainly speech, which is a non-stationary and highly correlated signal that can influence the overall performance of adaptive algorithms.

Different types of adaptive filters have been involved in the context of echo cancellation. The RLS-based algorithms would represent a very appealing choice (especially in terms of the convergence rate), if the computational complexity issue could be overcome. In this paper, we provide a practical overview on several RLS-based algorithms that could be used for echo cancellation, focusing on their key parameters.

It is well known that the performance of the RLS algorithm is mainly controlled by two important parameters, i.e., the forgetting factor and the regularization term. Similar to the attributes of the step-size from the NLMS-based algorithms, the performance of RLS-type algorithms in terms of convergence rate, tracking, misadjustment, and stability depends on the forgetting factor [2], [3]. The classical RLS algorithm uses a constant forgetting factor (between 0 and 1) and needs to compromise between the previous performance criteria. When the forgetting factor is very close to one, the algorithm achieves low misadjustment and good stability, but its tracking capabilities are reduced [6]. A small value of the forgetting factor improves the tracking but increases the misadjustment, and could affect the stability of the algorithm [7]. Motivated by these aspects, a number of variable forgetting factor RLS (VFF-RLS) algorithms have been developed, e.g., [8]–[11] (and references therein).

It should be mentioned that in the context of system identification (like in echo cancellation), where the output of the unknown system is corrupted by another signal (which is usually an additive noise), the goal of the adaptive filter is not to make the error signal goes to zero, because this will

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

introduce noise in the adaptive filter. The objective instead is to recover the "corrupting signal" from the error signal of the adaptive filter after this one converges to the true solution. This was the approach behind the VFF-RLS algorithm proposed in [10], which is analyzed in Section III.

As compared to the forgetting factor, the regularization parameter has been less addressed in the literature. Apparently, it is required in matrix inversion when this matrix is ill conditioned, especially in the initialization stage of the algorithm. However, its role is of great importance in practice, since regularization is a must in all ill-posed problems (like in adaptive filtering), especially in the presence of additive noise [12]–[15]. Consequently, in Section IV, we focus on the regularized RLS algorithm [3]. Following the development from [13], a method to select an optimal regularization parameter is presented, so that the algorithm could behave well in all noisy conditions. Since the value of this parameter is related to the echo-to-noise ratio (ENR), a simple and practical way to estimate the ENR in practice is also presented, which leads to a variable regularized RLS (VR-RLS) algorithm. Also, a low-complexity version of the proposed VR-RLS algorithm is developed, based on the dichotomous coordinate descent (DCD) method [16], [17].

The simulation results (presented in Section V) are performed in the context of both network and acoustic echo cancellation. The results support the theoretical findings and indicate the good performance of these algorithms. Finally, the conclusions are provided in Section VI.

II. System Model for Echo Cancellation

In the context of echo cancellation (Figure 1), the microphone or desired signal at the discrete-time index n is

$$d(n) = \mathbf{x}^T(n)\mathbf{h} + v(n) = y(n) + v(n), \quad (1)$$

where

$$\mathbf{x}(n) = \left[\begin{array}{cccc} x(n) & x(n-1) & \cdots & x(n-L+1) \end{array} \right]^T \quad (2)$$

is a vector containing the L most recent time samples of the zero-mean input (loudspeaker) signal $x(n)$, superscript ^{T} denotes transpose of a vector or a matrix,

$$\mathbf{h} = \left[\begin{array}{cccc} h_0 & h_1 & \cdots & h_{L-1} \end{array} \right]^T \quad (3)$$

is the impulse response (of length L) of the system (from the loudspeaker to the microphone) that we need to identify, and $v(n)$ the zero-mean near-end signal. In case of single-talk (i.e., the near-end speech is absent), $v(n)$ can usually be considered a zero-mean stationary white Gaussian noise signal with the variance $\sigma_v^2 = E\left[v^2(n)\right]$, where $E[\cdot]$ denotes mathematical expectation. The signal $y(n)$ is called the echo in the context of echo cancellation that we want to cancel with an adaptive filter [4], [5].

Then, our objective is to estimate or identify $\mathbf{h}$ with an adaptive filter:

$$\widehat{\mathbf{h}}(n) = \left[\begin{array}{cccc} \widehat{h}_0(n) & \widehat{h}_1(n) & \cdots & \widehat{h}_{L-1}(n) \end{array} \right]^T, \quad (4)$$

in such a way that for a reasonable value of n, we have for the (normalized) misalignment:

$$\frac{\left\| \mathbf{h} - \widehat{\mathbf{h}}(n) \right\|_2^2}{\|\mathbf{h}\|_2^2} \leq \iota, \quad (5)$$

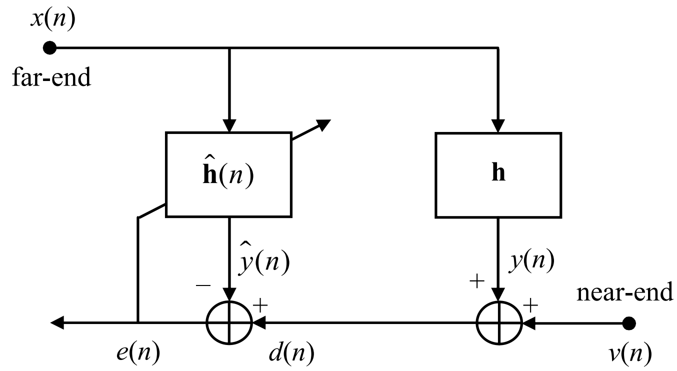

Figure 1. General configuration for echo cancellation.

where ι is a predetermined small positive number and $\|\cdot\|_2$ is the ℓ_2 norm. In this context, the a priori error signal is defined as

$$e(n) = d(n) - \mathbf{x}^T(n)\widehat{\mathbf{h}}(n-1) = d(n) - \widehat{y}(n), \quad (6)$$

where the vector $\widehat{\mathbf{h}}(n-1)$ contains the adaptive filter coefficients at time $n-1$ and $\widehat{y}(n)$ is the output of the adaptive filter.

III. Variable Forgetting Factor RLS Algorithm

The classical RLS algorithm can be immediately deduced from the normal equations, which are

$$\widehat{\mathbf{R}}_{\mathbf{x}}(n)\widehat{\mathbf{h}}(n) = \widehat{\mathbf{r}}_{dx}(n), \quad (7)$$

where

$$\begin{aligned} \widehat{\mathbf{R}}_{\mathbf{x}}(n) &= \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)\mathbf{x}^T(i) \\ &= \lambda\widehat{\mathbf{R}}_{\mathbf{x}}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n), \quad (8) \end{aligned}$$

$$\begin{aligned} \widehat{\mathbf{r}}_{\mathbf{x}d}(n) &= \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)d(i) \\ &= \lambda\widehat{\mathbf{r}}_{\mathbf{x}d}(n-1) + \mathbf{x}(n)d(n), \quad (9) \end{aligned}$$

and the parameter λ is the forgetting factor. According to (1), the normal equations become

$$\begin{aligned} &\sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)\mathbf{x}^T(i)\widehat{\mathbf{h}}(n) \\ &= \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)y(i) + \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)v(i). \quad (10) \end{aligned}$$

For a value of λ very close to 1 and for a large value of n, it may be assumed that

$$\frac{1}{n}\sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)v(i) \approx E\left[\mathbf{x}(n)v(n)\right] = 0. \quad (11)$$

Consequently, taking (10) into account,

$$\begin{aligned} \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)\mathbf{x}^T(i)\widehat{\mathbf{h}}(n) &\approx \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)y(i) \\ &= \sum_{i=1}^{n} \lambda^{n-i}\mathbf{x}(i)\mathbf{x}^T(i)\mathbf{h}, \quad (12) \end{aligned}$$

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

thus $\widehat{\mathbf{h}}(n) \approx \mathbf{h}$ and $e(n) \approx v(n)$. Now, for a small value of the forgetting factor, so that $\lambda^k \ll 1$ for $k \geq n_0$, it can be assumed that

$$\sum_{i=1}^{n} \lambda^{n-i}(\bullet) \approx \sum_{i=n-n_0+1}^{n} \lambda^{n-i}(\bullet).$$

According to the orthogonality theorem [2], [3], the normal equations become

$$\sum_{i=n-n_0+1}^{n} \lambda^{n-i}\mathbf{x}(i)e(i) = \mathbf{0}_{L\times 1},$$

where $\mathbf{0}_{L\times 1}$ denotes a vector with all its L elements equal to zero. This is a homogeneous set of L equations with n_0 unknown parameters, $e(i)$. When $n_0 < L$, this set of equations has the unique solution $e(i) = 0$, for $i = n - n_0 + 1, \ldots, n$, leading to $\widehat{y}(n) = y(n) + v(n)$. Consequently, there is a "leakage" of $v(n)$ into the output of the adaptive filter. In this situation, the signal $v(n)$ is cancelled; even if the error signal is $e(n) = 0$, this does not lead to a correct solution from the system identification point of view. A small value of λ or a high value of L intensifies this phenomenon.

Summarizing, for a low value of λ the output of the adaptive system is $\widehat{y}(n) \approx y(n) + v(n)$, while $\lambda \approx 1$ leads to $\widehat{y}(n) \approx y(n)$. Apparently, for a system identification application, a value of λ very close to 1 is desired; but in this case, even if the initial convergence rate of the algorithm is satisfactory, the tracking capabilities suffer a lot. In order to provide fast tracking, a lower value of λ is desired. On the other hand, taking into account the previous aspects, a low value of λ is not good in the steady-state. Consequently, a VFF-RLS algorithm (which could provide both fast tracking and low misadjustment) can be a more appropriate solution, in order to deal with these aspects.

Let us start the development by writing the relations that define the classical RLS algorithm:

$$\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{\lambda + \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)}, \tag{13}$$

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mathbf{k}(n)e(n), \tag{14}$$

$$\mathbf{P}(n) = \frac{1}{\lambda}\left[\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\right], \tag{15}$$

where $\mathbf{k}(n)$ is the Kalman gain vector, $\mathbf{P}(n)$ is the inverse of the input correlation matrix, and $e(n)$ is the a priori error signal defined in (6). The a posteriori error signal can be defined using the adaptive filter coefficients at time n, i.e.,

$$\varepsilon(n) = d(n) - \mathbf{x}^T(n)\widehat{\mathbf{h}}(n) \tag{16}$$

Using (6) and (14) in (16), it results

$$\varepsilon(n) = e(n)\left[1 - \mathbf{x}^T(n)\mathbf{k}(n)\right]. \tag{17}$$

According to the problem statement, it is desirable to recover the system noise from the error signal. Consequently, it can be imposed the condition:

$$E\left[\varepsilon^2(n)\right] = \sigma_v^2. \tag{18}$$

Using (18) in (17) and taking (13) into account, it finally results

$$E\left\{\left[1 - \frac{\theta(n)}{\lambda(n) + \theta(n)}\right]^2\right\} = \frac{\sigma_v^2}{\sigma_e^2(n)}, \tag{19}$$

where $\theta(n) = \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)$. In (19), we assumed that the input and error signals are uncorrelated, which is true when the adaptive filter has started to converge to the true solution. We also assumed that the forgetting factor is deterministic and time dependent. By solving the quadratic equation (19), it results a variable forgetting factor

$$\lambda(n) = \frac{\sigma_\theta(n)\sigma_v}{\sigma_e(n) - \sigma_v}, \tag{20}$$

where $E\left[\theta^2(n)\right] = \sigma_\theta^2(n)$. In practice, the variance of the error signal is estimated based on

$$\widehat{\sigma}_e^2(n) = \alpha\widehat{\sigma}_e^2(n-1) + (1-\alpha)e^2(n), \tag{21}$$

where $\alpha = 1 - 1/(KL)$, with $K \geq 1$. Also, the variance of $\theta(n)$ is evaluated in a similar manner, i.e.,

$$\widehat{\sigma}_\theta^2(n) = \alpha\widehat{\sigma}_\theta^2(n-1) + (1-\alpha)\theta^2(n). \tag{22}$$

The estimate of the noise power, $\widehat{\sigma}_v^2$ [which should be used in (20) from practical reasons], can be evaluated in different ways, e.g., [10], [19], [20].

Theoretically, $\sigma_e(n) \geq \sigma_v$ in (20). Compared to the least-mean-square algorithms [where there is the gradient noise, so that $\sigma_e(n) > \sigma_v$], the RLS algorithm with $\lambda(n) \approx 1$ leads to $\sigma_e(n) \approx \sigma_v$. In practice (since power estimates are used), several situations have to be prevented in (20). Apparently, when $\widehat{\sigma}_e(n) \leq \widehat{\sigma}_v$, it could be set $\lambda(n) = \lambda_{\max}$, where $\lambda_{\max}$ is very close or equal to 1. But this could be a limitation, because in the steady-state of the algorithm $\widehat{\sigma}_e(n)$ varies around $\widehat{\sigma}_v$. A more reasonable solution is to impose that $\lambda(n) = \lambda_{\max}$ when

$$\widehat{\sigma}_e(n) \leq \rho\widehat{\sigma}_v, \tag{23}$$

with $1 < \rho \leq 2$. Otherwise, the forgetting factor of the proposed VFF-RLS algorithm is evaluated as

$$\lambda(n) = \min\left[\frac{\widehat{\sigma}_\theta(n)\widehat{\sigma}_v}{\zeta + |\widehat{\sigma}_e(n) - \widehat{\sigma}_v|}, \lambda_{\max}\right], \tag{24}$$

where the small positive constant ζ prevents a division by zero. Before the algorithm converges or when there is an abrupt change of the system, $\widehat{\sigma}_e(n)$ is large as compared to $\widehat{\sigma}_v$; thus, the parameter $\lambda(n)$ from (24) takes low values, providing fast convergence and good tracking. When the algorithm converges to the steady-state solution, $\widehat{\sigma}_e(n) \approx \widehat{\sigma}_v$ [so that the condition (23) is fulfilled] and $\lambda(n)$ is equal to $\lambda_{\max}$, providing low misadjustment. The resulted VFF-RLS algorithm is summarized in Table I. It can be noticed that the mechanism that controls the forgetting factor is very simple and not expensive in terms of multiplications and additions.

IV. Variable Regularized RLS Algorithm

In this section, a different version of the RLS algorithm is presented, which allows us to outline the importance of the regularization parameter. Let us consider the regularized least-squares criterion:

$$J(n) = \sum_{i=0}^{n} \lambda^{n-i}\left[d(i) - \widehat{\mathbf{h}}^T(n)\mathbf{x}(i)\right]^2 + \delta\left\|\widehat{\mathbf{h}}(n)\right\|_2, \tag{25}$$

where λ is the same exponential forgetting factor and δ is the regularization parameter. From (25), the update of the regularized RLS algorithm [3] results in

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \left[\widehat{\mathbf{R}}_{\mathbf{x}}(n) + \delta\mathbf{I}_L\right]^{-1}\mathbf{x}(n)e(n), \tag{26}$$

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. VFF-RLS algorithm.

Initialization:
$\mathbf{P}(0) = \gamma \mathbf{I}_L \ (\gamma > 0)$
$\widehat{\mathbf{h}}(0) = \mathbf{0}_{L\times 1}$
$\widehat{\sigma}_e^2(0) = \widehat{\sigma}_\theta^2(0) = 0$

Parameters:
$\alpha = 1 - \frac{1}{KL}$ (with $K > 1$) weighting factor
$\lambda_{\max}$, upper bound of the forgetting factor (very close or equal to 1)
$\zeta > 0$, very small number to avoid division by zero
$\widehat{\sigma}_v^2$, system noise power (estimated)
For time index $n = 1, 2, \ldots$:
$e(n) = d(n) - \mathbf{x}^T(n)\widehat{\mathbf{h}}(n-1)$
$\theta(n) = \mathbf{x}^T(n)\mathbf{P}(n-1)\mathbf{x}(n)$
$\widehat{\sigma}_e^2(n) = \alpha\widehat{\sigma}_e^2(n-1) + (1-\alpha)e^2(n)$
$\widehat{\sigma}_\theta^2(n) = \alpha\widehat{\sigma}_\theta^2(n-1) + (1-\alpha)\theta^2(n)$
$\lambda(n) = \begin{cases} \lambda_{\max}, & \text{if } \widehat{\sigma}_e(n) \le \rho\widehat{\sigma}_v \text{ (where } 1 < \rho \le 2) \\ \min\left[\frac{\widehat{\sigma}_\theta(n)\widehat{\sigma}_v}{\zeta + |\widehat{\sigma}_e(n) - \widehat{\sigma}_v|}, \lambda_{\max}\right], & \text{otherwise} \end{cases}$
$\mathbf{k}(n) = \frac{\mathbf{P}(n-1)\mathbf{x}(n)}{\lambda(n) + \theta(n)}$
$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \mathbf{k}(n)e(n)$
$\mathbf{P}(n) = \frac{1}{\lambda(n)}\left[\mathbf{P}(n-1) - \mathbf{k}(n)\mathbf{x}^T(n)\mathbf{P}(n-1)\right]$

where the matrix $\widehat{\mathbf{R}}_{\mathbf{x}}(n)$ from (8) is an estimate of the correlation matrix of $\mathbf{x}(n)$ at time n, $\mathbf{I}_L$ is the identity matrix of size $L \times L$, and $e(n)$ is the a priori error signal defined in (6). We will assume that the matrix $\widehat{\mathbf{R}}_{\mathbf{x}}(n)$ has full rank, although it can be very ill conditioned. As a result, if there is no noise, regularization is not really required; however, the more the noise, the larger should be the value of δ.

Summarizing, the regularized RLS algorithm is defined by the relations (6), (8), and (26). In the following, we present one reasonable way to find the regularization parameter δ. It can be noticed that the update equation of the regularized RLS can be rewritten as [13]

$$\widehat{\mathbf{h}}(n) = \mathbf{Q}(n)\widehat{\mathbf{h}}(n-1) + \widetilde{\mathbf{h}}(n), \tag{27}$$

where

$$\mathbf{Q}(n) = \mathbf{I}_L - \left[\widehat{\mathbf{R}}_{\mathbf{x}}(n) + \delta\mathbf{I}_L\right]^{-1}\mathbf{x}(n)\mathbf{x}^T(n) \tag{28}$$

and

$$\widetilde{\mathbf{h}}(n) = \left[\widehat{\mathbf{R}}_{\mathbf{x}}(n) + \delta\mathbf{I}_L\right]^{-1}\mathbf{x}(n)d(n) \tag{29}$$

is the correctiveness component of the algorithm, which depends on the new observation $d(n)$. In this context, we can notice that $\mathbf{Q}(n)$ does not depend on the noise signal and $\mathbf{Q}(n)\widehat{\mathbf{h}}(n-1)$ in (27) can be seen as a good initialization of the adaptive filter. In fact, (29) is the solution of the noisy linear system of L equations:

$$\left[\widehat{\mathbf{R}}_{\mathbf{x}}(n) + \delta\mathbf{I}_L\right]\widetilde{\mathbf{h}}(n) = \mathbf{x}(n)d(n). \tag{30}$$

Let us define

$$\widetilde{e}(n) = d(n) - \widetilde{\mathbf{h}}^T(n)\mathbf{x}(n), \tag{31}$$

the error signal between the desired signal and the estimated signal obtained from the filter optimized in (29). Consequently, we could find δ in such a way that the expected value of $\widetilde{e}^2(n)$ is equal to the variance of the noise, i.e.,

$$E\left[\widetilde{e}^2(n)\right] = \sigma_v^2. \tag{32}$$

This is reasonable if we want to attenuate the effects of the noise in the estimator $\widetilde{\mathbf{h}}(n)$.

For the sake of simplicity, let us assume that $x(n)$ is stationary and white. Apparently, this assumption is quite restrictive, even if it was widely used in many developments in the context of adaptive filtering [2], [3]. However, the resulting VR-RLS algorithm will still use the full matrix $\widehat{\mathbf{R}}_{\mathbf{x}}(n)$ and, consequently, it will inherit the good performance feature of the RLS family in case of correlated inputs. In this case and for n large enough (also considering that the forgetting factor λ is on the order of $1 - 1/L$), we have

$$\begin{aligned}\left[\widehat{\mathbf{R}}_{\mathbf{x}}(n) + \delta\mathbf{I}_L\right] &\approx \left[\frac{\sigma_x^2}{1-\lambda} + \delta\right]\mathbf{I}_L \\ &\approx \left[L\sigma_x^2 + \delta\right]\mathbf{I}_L\end{aligned} \tag{33}$$

and $\mathbf{x}^T(n)\mathbf{x}(n) \approx L\sigma_x^2$, where $\sigma_x^2 = E\left[x^2(n)\right]$ is the variance of the input signal. Next, from (1), we can define the echo-to-noise ratio (ENR) as

$$\text{ENR} = \frac{\sigma_y^2}{\sigma_v^2}, \tag{34}$$

where $\sigma_y^2 = E\left[y^2(n)\right]$ is the variance of $y(n)$. Developing (32) and based on the previous approximations, we obtain the quadratic equation:

$$\delta^2 - 2\frac{L\sigma_x^2}{\text{ENR}}\delta - \frac{\left(L\sigma_x^2\right)^2}{\text{ENR}} = 0, \tag{35}$$

with the obvious solution:

$$\begin{aligned}\delta &= \frac{L\left(1 + \sqrt{1 + \text{ENR}}\right)}{\text{ENR}}\sigma_x^2 \\ &= \beta\sigma_x^2,\end{aligned} \tag{36}$$

where

$$\beta = \frac{L\left(1 + \sqrt{1 + \text{ENR}}\right)}{\text{ENR}} \tag{37}$$

is the normalized regularization parameter of the RLS algorithm.

As we can notice from (36), the regularization parameter δ depends on three elements, i.e., the length of the adaptive filter, the variance of the input signal, and the ENR. In most applications, the first two elements (L and σ_x^2) are known, while the ENR can be estimated. Using a proper evaluation of the ENR, the algorithm should own good robustness features against the additive noise.

Let us assume that the adaptive filter has converged to a certain degree, so that we can use the approximation

$$y(n) \approx \widehat{y}(n). \tag{38}$$

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Hence,

$$\sigma_y^2 \approx \sigma_{\widehat{y}}^2, \tag{39}$$

where $\sigma_{\widehat{y}}^2 = E\left[\widehat{y}^2(n)\right]$. Since the output of the unknown system and the noise can be considered uncorrelated, (1) can be expressed in terms of power estimates as

$$\sigma_d^2 = \sigma_y^2 + \sigma_v^2, \tag{40}$$

where $\sigma_d^2 = E\left[d^2(n)\right]$. Using (39) in (40), we obtain

$$\sigma_v^2 \approx \sigma_d^2 - \sigma_{\widehat{y}}^2. \tag{41}$$

The power estimates can be evaluated in a recursive manner [similar to (21) and (22)] as

$$\widehat{\sigma}_d^2(n) = \alpha\widehat{\sigma}_d^2(n-1) + (1-\alpha)d^2(n), \tag{42}$$

$$\widehat{\sigma}_{\widehat{y}}^2(n) = \alpha\widehat{\sigma}_{\widehat{y}}^2(n-1) + (1-\alpha)\widehat{y}^2(n). \tag{43}$$

Therefore, based on (39) and (41), an estimation of the ENR is obtained as

$$\widehat{\text{ENR}}(n) = \frac{\widehat{\sigma}_{\widehat{y}}^2(n)}{\left|\widehat{\sigma}_d^2(n) - \widehat{\sigma}_{\widehat{y}}^2(n)\right|}, \tag{44}$$

so that the variable regularization parameter results in

$$\begin{aligned}\delta(n) &= \frac{L\left[1+\sqrt{1+\widehat{\text{ENR}}(n)}\right]}{\widehat{\text{ENR}}(n)}\sigma_x^2 \\ &= \beta(n)\sigma_x^2, \end{aligned} \tag{45}$$

where

$$\beta(n) = \frac{L\left[1+\sqrt{1+\widehat{\text{ENR}}(n)}\right]}{\widehat{\text{ENR}}(n)} \tag{46}$$

is the variable normalized regularization parameter. Consequently, based on (45), we obtain a variable-regularized RLS (VR-RLS) algorithm, with the update:

$$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \left[\widehat{\mathbf{R}}_{\mathbf{x}}(n) + \delta(n)\mathbf{I}_L\right]^{-1}\mathbf{x}(n)e(n), \tag{47}$$

where $\widehat{\mathbf{R}}_{\mathbf{x}}(n)$ is recursively evaluated according to (8) and $\delta(n)$ is computed based on (42)–(45).

At this point, some practical issues should be outlined. The absolute values in (44) prevent any minor deviations (due to the use of power estimates) from the true values, which can make the denominator negative. The VR-RLS is a non-parametric algorithm, since all the parameters in (44) are available. Also, good robustness against the additive noise variations is expected. The main drawback is due to the approximation in (39). This assumption will be biased in the initial convergence phase or when there is a change of the unknown system. Concerning the initial convergence, we can use a constant regularization parameter δ in the first steps of the algorithm (e.g., in the first L iterations).

However, the VR-RLS algorithm faces two main challenges in terms of computational complexity. The first one is the update of the matrix $\widehat{\mathbf{R}}_{\mathbf{x}}(n)$ from (8), while the second issue is related to the evaluation of the last term from the right-hand side of (47), which contains both the matrix inversion and the product with the input vector.

The complexity of (8) can be greatly reduced taking into account that the vector $\mathbf{x}(n)$ has the time shift property [see (2)] and the matrix $\widehat{\mathbf{R}}_{\mathbf{x}}(n)$ is symmetric. Thus, only the first column of this matrix has to be computed, i.e.,

$$\widehat{\mathbf{R}}_{\mathbf{x}}^{(1)}(n) = \lambda\widehat{\mathbf{R}}_{\mathbf{x}}^{(1)}(n-1) + \mathbf{x}(n)x(n), \tag{48}$$

since the lower-right $(L-1)\times(L-1)$ block of $\widehat{\mathbf{R}}_{\mathbf{x}}(n)$ can be obtained by copying the $(L-1)\times(L-1)$ upper-left block of the matrix $\widehat{\mathbf{R}}_{\mathbf{x}}(n-1)$.

The evaluation of the last term from the right-hand side of (47) is more challenging. In fact, the basic problem can be interpreted in terms of solving the normal equations [3]:

$$\mathbf{R}(n)\widehat{\mathbf{h}}(n) = \widehat{\mathbf{r}}_{\mathbf{x}d}(n), \tag{49}$$

where

$$\mathbf{R}(n) = \widehat{\mathbf{R}}_{\mathbf{x}}(n) + \delta(n)\mathbf{I}_L \tag{50}$$

and $\widehat{\mathbf{r}}_{\mathbf{x}d}(n)$ is defined in (9). As an alternative to the classical approaches [2], [3], the normal equations (49) can be recursively solved using the dichotomous coordinate descent (DCD) method [16]. The basic idea is to express the problem in terms of auxiliary normal equations with respect to increments of the filter weights [17]. In our case, we need to solve

$$\mathbf{R}(n)\triangle\widehat{\mathbf{h}}(n) = \mathbf{p}(n), \tag{51}$$

where $\triangle\widehat{\mathbf{h}}(n)$ is the increment of the filter weights and

$$\mathbf{p}(n) = \lambda\mathbf{r}(n-1) + \mathbf{x}(n)e(n), \tag{52}$$

with $\mathbf{r}(n)$ representing the so-called residual vector associated to the solution [17]. Consequently, following the previous development and the steps presented in [17], the low-complexity version of the proposed VR-RLS algorithm, namely VR-RLS-DCD, is summarized in Table II, where step 6 involves the DCD iterations.

The DCD algorithm [16] is based on coordinate descent iterations with a power of two variable step-size, q. It does not need multiplications or divisions (these operations are simply replaced by bit-shifts), but only additions, so that it is well suited for hardware implementation. In our case, the auxiliary normal equations from step 6 are solved by using the DCD with a leading element [17]. An insightful analysis of this algorithm can be found in [17]. Also, detailed implementation aspects are discussed in [18].

Here, we briefly outline some of the important parameters of the DCD algorithm (using the notation from [17]). First, the parameters H and M_{b} represent the maximum amplitude expected for the values of $\triangle\widehat{\mathbf{h}}(n)$, respectively the number of bits used for their representation. If the value of H is chosen accordingly, the values of the step-size q correspond to the powers of 2 and are associated with the bits comprising the binary representation of each computed value in the solution vector. In this case, any multiplication with q can be replaced by a bit-shift. Second, the parameter N_{u} represents the maximum number of allowed (or "successful") iterations performed for $\triangle\widehat{\mathbf{h}}(n)$ [17]; in practice $N_{\text{u}} \ll L$. The arithmetic complexity of the DCD algorithm is proportional to LN_{u} but using only additions. Consequently, the complexity associated to the matrix inversion is greatly reduced as compared to the classical method [which requires $O(L^3)$ operations] and

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE II. VR-RLS-DCD algorithm.

Initialization:	$\widehat{\mathbf{h}}(0) = \mathbf{0},\ \mathbf{r}(0) = \mathbf{0},\ \widehat{\mathbf{R}}_{\mathbf{x}}(0) = \mathbf{0}_L$
For $n = 1, 2, \ldots$	
Step 1:	$\widehat{\mathbf{R}}_{\mathbf{x}}(n) = \lambda\widehat{\mathbf{R}}_{\mathbf{x}}(n-1) + \mathbf{x}(n)\mathbf{x}^T(n)$ [using (48)]
Step 2:	Compute $\delta(n)$ based on (42)–(45)
Step 3:	$\mathbf{R}(n) = \widehat{\mathbf{R}}_{\mathbf{x}}(n) + \delta(n)\mathbf{I}_L$
Step 4:	$e(n) = d(n) - \widehat{\mathbf{h}}^T(n-1)\mathbf{x}(n)$
Step 5:	$\mathbf{p}(n) = \lambda\mathbf{r}(n-1) + \mathbf{x}(n)e(n)$
Step 6:	$\mathbf{R}(n)\triangle\widehat{\mathbf{h}}(n) = \mathbf{p}(n) \ \Rightarrow\ \triangle\widehat{\mathbf{h}}(n),\ \mathbf{r}(n)$ (to be solved with DCD iterations [17])
Step 7:	$\widehat{\mathbf{h}}(n) = \widehat{\mathbf{h}}(n-1) + \triangle\widehat{\mathbf{h}}(n)$

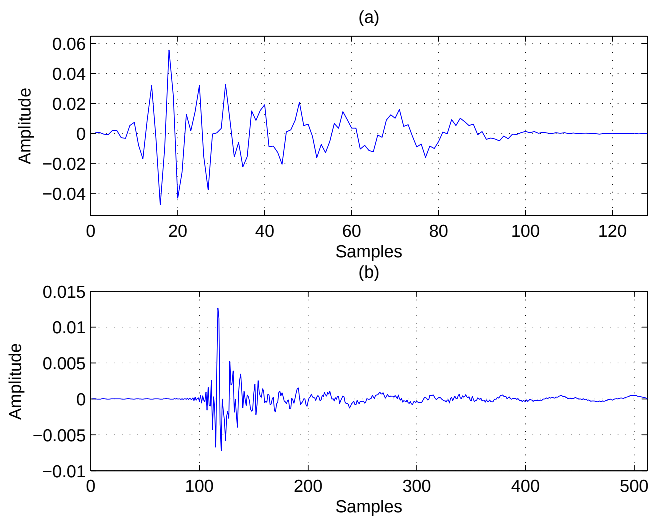

Figure 2. Impulse responses used in simulations.

even to the regular RLS algorithm [2], [3] [which is based on the matrix inversion lemma and needs $O(L^2)$ operations]. Therefore, the DCD-based algorithms are very appealing for real-world applications.

Nevertheless, the RLS-DCD algorithm proposed in [17] uses a constant regularization for $\widehat{\mathbf{R}}_{\mathbf{x}}(0)$, but the influence of this parameter is negligible due to the forgetting factor in the update of the matrix $\widehat{\mathbf{R}}_{\mathbf{x}}(n)$. On the other hand, using a proper estimation of the regularization parameter within the algorithm (i.e., steps 2 and 3 in Table II), the robustness against additive noise can be improved. Thus, the proposed VR-RLS-DCD algorithm owns this robustness feature, but also the low-complexity advantage inherited from the DCD method.

V. Simulation Results

First, let us consider a network echo cancellation scenario, in the framework of G168 Recommendation [21]. The echo path is depicted in Figure 2(a); it is the fourth impulse response (of length $L = 128$) from the above recommendation. The sampling rate is 8 kHz. All adaptive filters used in the experiments have the same length as the echo path. The far-end signal (i.e., the input signal) is a speech signal. The output

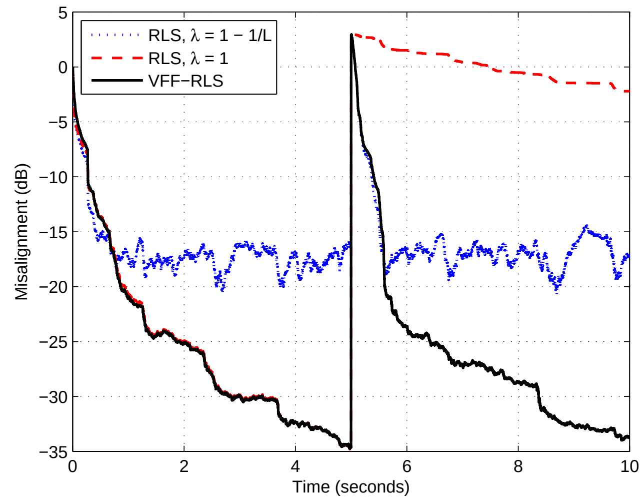

Figure 3. Misalignment of the RLS algorithm (using different constant values of the forgetting factor) and the VFF-RLS algorithm. The input signal is speech, $L = 128$, and ENR = 20 dB. Echo path changes at time 5 seconds.

of the echo path is corrupted by an independent white Gaussian noise with 20 dB ENR. An echo path change scenario is some experiments (in order to evaluate the tracking capabilities of the algorithms), by shifting the impulse response to the right by 8 samples in the middle of simulation. The performance measure is the normalized misalignment (in dB) evaluated as

$$\mathrm{Mis}(n) = 20\log_{10}\frac{\left\|\mathbf{h}(n) - \widehat{\mathbf{h}}(n)\right\|_2}{\left\|\mathbf{h}(n)\right\|_2}. \quad (53)$$

In the first experiment, the performance of the VFF-RLS algorithm (presented in Section III) is evaluated, as compared to the classical RLS algorithm defined in (13)–(15), which uses different constant values of the forgetting factor. A single-talk case is considered and the echo path changes in the middle of simulation. It can be noticed in Figure 3 that the VFF-RLS algorithm achieves the same initial misalignment as the RLS with its maximum forgetting factor, but it tracks as fast as the RLS with the smaller forgetting factor. As expected, the classical RLS algorithm using constant forgetting factors has to compromise between these performance criteria, i.e., the larger the value of λ, the better the misalignment level but worse the tracking capability.

Next, the performance of the VR-RLS algorithm (from Section IV) is investigated, as compared to the regularized RLS algorithm defined by the update (26), using different constant values of the regularization parameter. Based on (37), we can determine the values of the optimal normalized regularization parameter of the RLS algorithm for different cases; for example, let us consider two values of the ENR, i.e., 20 dB (the true one) and 0 dB. Using appropriate notation, we obtain $\beta_{20} = 14.14$ and $\beta_0 = 309.01$, respectively. Next, we compare the regularized RLS algorithm using these constant regularization parameters with the VR-RLS algorithm. The constant forgetting factor is set to $\lambda = 1 - 1/(3L)$ for all the algorithms.

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

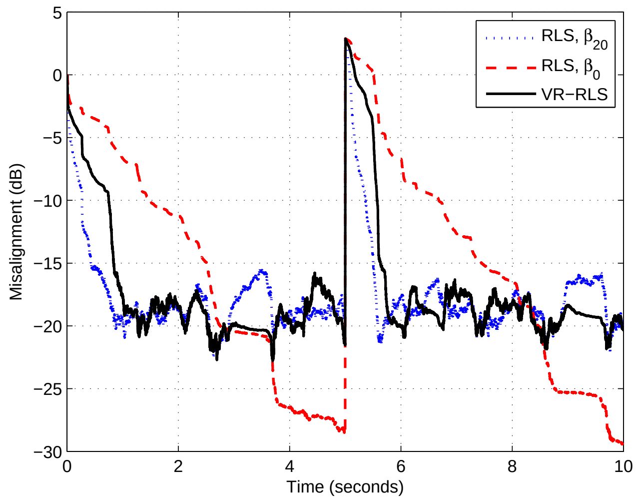

Figure 4. Misalignment of the regularized RLS algorithm (using different constant values of the regularization parameter) and the VR-RLS algorithm. The input signal is speech, $L = 128$, and ENR = 20 dB. Echo path changes at time 5 seconds.

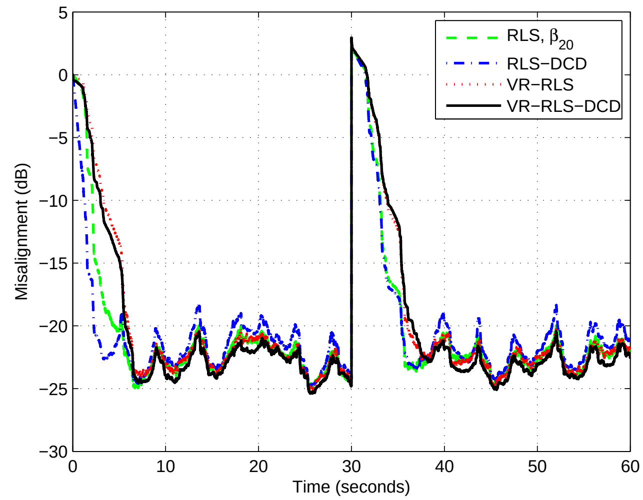

Figure 6. Misalignment of the regularized RLS (using β_{20}), RLS-DCD, VR-RLS, and VR-RLS-DCD algorithms. The input signal is speech, $L = 512$, and ENR = 20 dB. Echo path changes at time 30 seconds.

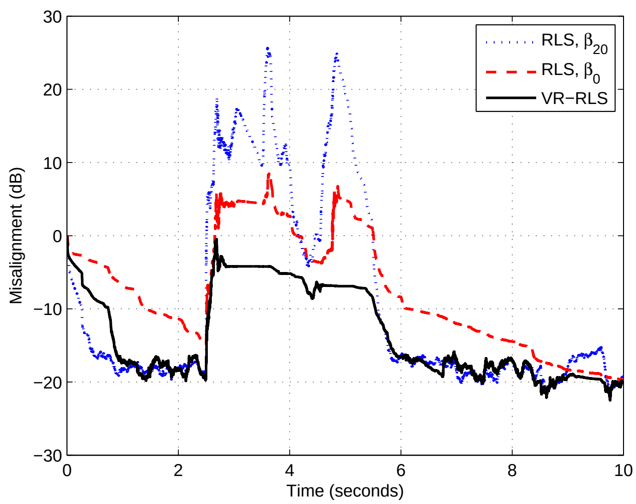

Figure 5. Misalignment of the regularized RLS algorithm (using different constant values of the regularization parameter) and the VR-RLS algorithm. The input signal is speech, $L = 128$, and ENR = 20 dB. Near-end speech appears between time 2.5 and 5 seconds (double-talk scenario).

In Figure 4, a single-talk scenario is considered and an echo path change is introduced in the middle of the simulation. It can be noticed that the VR-RLS algorithm behaves similarly to the RLS algorithm using the constant parameter β_{20}, which is associated to the value of the true ENR. Also, it can be noticed that a larger value of the normalized regularization parameter (β_0) improves the misalignment but affects the convergence rate and tracking.

In Figure 5, a double-talk scenario [4], [5] is considered. The near-end speech appears between time 2.5 and 5 seconds, so that the signal $v(n)$ is now non-stationary, since it contains both noise and speech. It is clear that the VR-RLS algorithm is more robust in this case as compared to the regularized RLS using constant values of β. It should be outlined that we do not use any double-talk detector (DTD) [4], [5] with the VR-RLS algorithm. Therefore, the VR-RLS algorithm owns good robustness features against double-talk, which is an important gain in practice.

The second set of simulations is performed in the context of acoustic echo cancellation [4], [5]. The unknown system, i.e., the echo path, is a measured acoustic impulse response depicted in Figure 2(b). It has 512 coefficients and the same length is used for the adaptive filter ($L = 512$). The output of the echo path is corrupted by a white Gaussian noise with different ENRs, i.e., 20 dB, 10 dB, and 0 dB. Based on (37), we can determine the values of the optimal normalized regularization parameter in these cases. Using appropriate notation, we obtain $\beta_{20} = 56.57$, $\beta_{10} = 221.01$, and $\beta_0 = 1236.07$, respectively. In simulations, we compare the regularized RLS algorithm using these constant regularization parameters with the proposed VR-RLS and VR-RLS-DCD algorithms. Also, the RLS-DCD algorithm [17] is included for comparison, using $N_{\rm u} = 8$, $M_{\rm b} = 16$, and $H = 1$ (the same parameters are used in the VR-RLS-DCD algorithm). The forgetting factor is set to $\lambda = 1 - 1/(16L)$ for all the algorithms.

In the first set of experiments, the value of the ENR is set to 20 dB. In Figure 6, an echo path change scenario is simulated in the middle of the experiment, by shifting the impulse response to the right by 25 samples. First, it can be noticed that the VR-RLS and VR-RLS-DCD algorithms behave very similarly and are close to the regularized RLS algorithm using the constant (optimal) parameter β_{20}, which is associated to the value of the ENR. As expected, there is an inherent delay in the initial convergence rate and tracking reaction of the variable-regularized algorithms (as compared to the RLS-DCD algorithm), due to the approximation in (39). In Figure 7, a double-talk scenario is considered; the near-end speech appears between time 27 and 30 seconds. It is clear that the VR-RLS and VR-RLS-DCD algorithms are more robust in

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

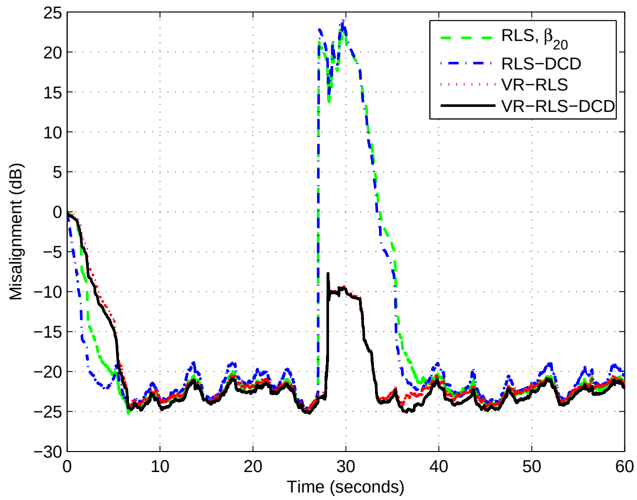

Figure 7. Misalignment of the regularized RLS (using β_{20}), RLS-DCD, VR-RLS, and VR-RLS-DCD algorithms. The input signal is speech, $L = 512$, and ENR = 20 dB. Near-end speech appears between time 27 and 30 seconds (double-talk scenario).

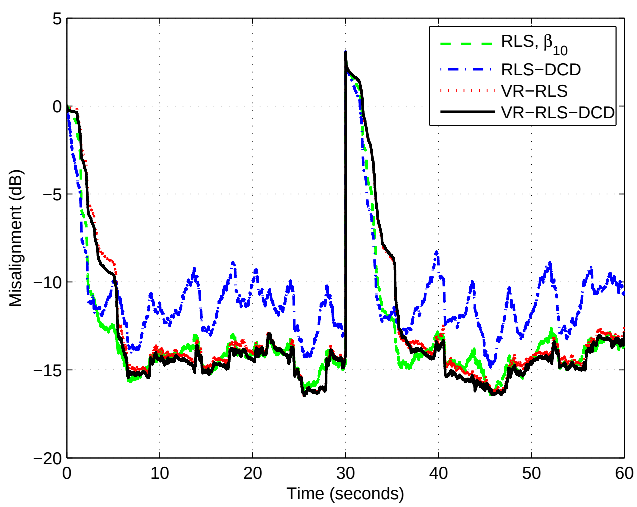

Figure 8. Misalignment of the regularized RLS (using β_{10}), RLS-DCD, VR-RLS, and VR-RLS-DCD algorithms. The input signal is speech, $L = 512$, and ENR = 10 dB. Echo path changes at time 30 seconds.

this case, since the estimated ENR from (44) also includes the contribution of the near-end signal.

In the second set of experiments, we select a lower ENR value, i.e., 10 dB. In this case, the importance of regularization becomes more apparent. As we can see from Figure 8, the VR-RLS and VR-RLS-DCD algorithms behave similarly to the regularized RLS using the constant (optimal) parameter β_{10}, and outperform the RLS-DCD algorithm (in terms of misalignment). Also, as we can notice in Figure 9, the variable-regularized algorithms are much more robust to double-talk, as compared to their counterparts.

Finally, in the last set of experiments, we consider ENR = 0 dB. As expected, according to the results in Figure 10, the

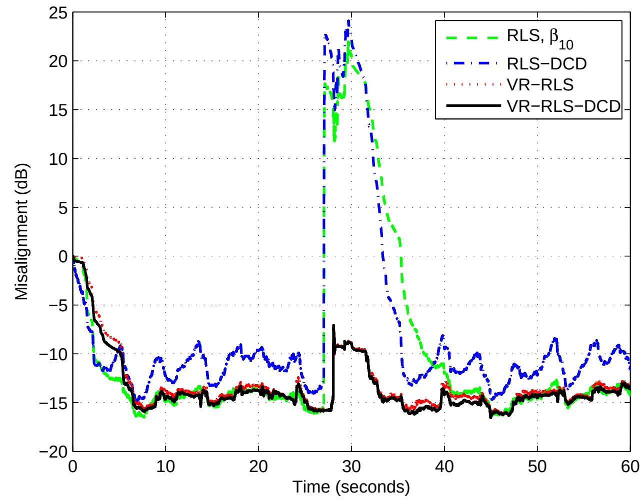

Figure 9. Misalignment of the regularized RLS (using β_{10}), RLS-DCD, VR-RLS, and VR-RLS-DCD algorithms. The input signal is speech, $L = 512$, and ENR = 10 dB. Near-end speech appears between time 27 and 30 seconds (double-talk scenario).

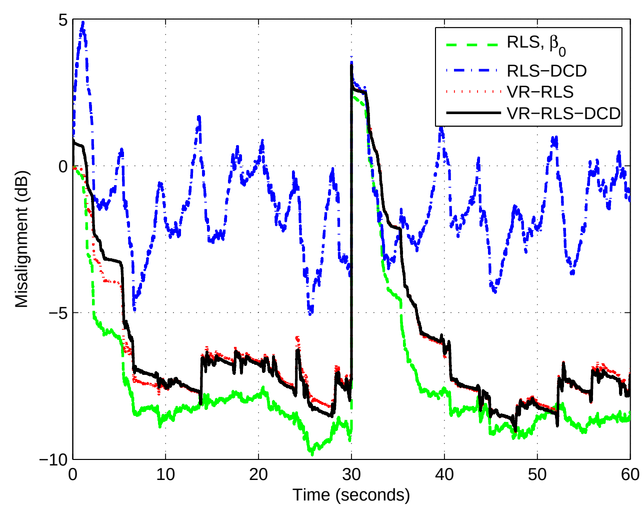

Figure 10. Misalignment of the regularized RLS (using β_0), RLS-DCD, VR-RLS, and VR-RLS-DCD algorithms. The input signal is speech, $L = 512$, and ENR = 0 dB. Echo path changes at time 30 seconds.

VR-RLS and VR-RLS-DCD algorithms behave now similarly to the regularized RLS using the constant (optimal) parameter β_0, and are much better as compared to the RLS-DCD algorithm. Besides, according to Figure 11, the variable-regularized algorithms outperform by far their counterparts in terms of double-talk robustness.

VI. Conclusions

The RLS algorithms are very appealing due to their fast convergence rate. In this paper, we have focused on the main parameters that control the performance of these algorithms, i.e., the forgetting factor and the regularization term. In order to achieve a better compromise between the performance

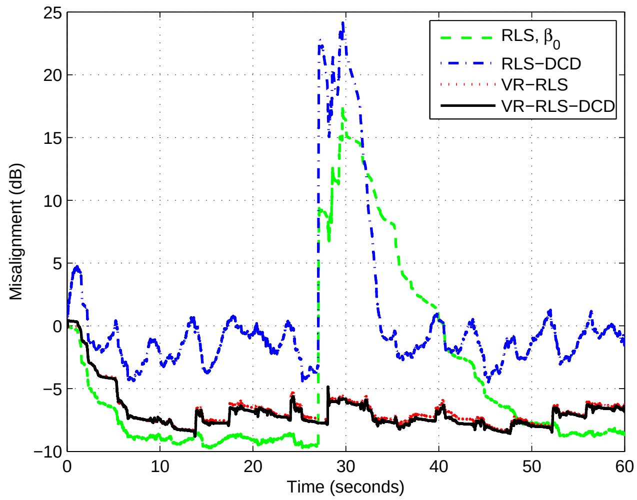

Figure 11. Misalignment of the regularized RLS (using β_0), RLS-DCD, VR-RLS, and VR-RLS-DCD algorithms. The input signal is speech, $L = 512$, and ENR = 0 dB. Near-end speech appears between time 27 and 30 seconds (double-talk scenario).

criteria (i.e., convergence and tracking versus misadjustment and robustness), these parameters could be controlled. In this context, the solutions presented in Sections III and IV led to the VFF-RLS and VR-RLS algorithms, respectively. Also, in Section IV, a low-complexity version of the VR-RLS algorithm was derived, based on the DCD method, namely the VR-RLS-DCD.

The first set of experiments was performed in the context of network echo cancellation. According to the simulation results, the VFF-RLS and VR-RLS algorithms perform very well as compared to their classical counterparts (which use constant values of the key parameters).

The second set of simulations was performed in an acoustic echo cancellation scenario. The results indicate that the VR-RLS and VR-RLS-DCD algorithms own good robustness features against the near-end signal. In other words, the robustness of the algorithm against ENR variations (e.g., like double-talk) can be controlled in terms of the regularization parameter. Moreover, due to its low-complexity feature, the VR-RLS-DCD algorithm could be a reliable candidates for real-world echo cancellation applications.

ACKNOWLEDGMENT

This work was supported by European Space Agency under Grant no. 4000121222/17/NL/CBi and University Politehnica of Bucharest under Grant no. 18/05.10.2017.

REFERENCES

[1] C. Elisei-Iliescu and C. Paleologu, “Recursive least-squares algorithms for echo cancellation – An overview and open issues,” in *Proc. ICN*, 2017, pp. 87–91.

[2] S. Haykin, *Adaptive Filter Theory*. Fourth Edition, Upper Saddle River, NJ: Prentice-Hall, 2002.

[3] A. H. Sayed, *Adaptive Filters*. New York, NY: Wiley, 2008.

[4] J. Benesty, T. Gaensler, D. R. Morgan, M. M. Sondhi, and S. L. Gay, *Advances in Network and Acoustic Echo Cancellation*. Berlin, Germany: Springer-Verlag, 2001.

[5] C. Paleologu, J. Benesty, and S. Ciochină, *Sparse Adaptive Filters for Echo Cancellation*. Morgan & Claypool Publishers, 2010.

[6] S. Ciochină, C. Paleologu, J. Benesty, and A. A. Enescu, “On the influence of the forgetting factor of the RLS adaptive filter in system identification,” in *Proc. IEEE ISSCS*, 2009, pp. 205–208.

[7] S. Ciochină, C. Paleologu, and A. A. Enescu, “On the behaviour of RLS adaptive algorithm in fixed-point implementation,” in *Proc. IEEE ISSCS*, 2003, pp. 57–60.

[8] S. Song, J. S. Lim, S. J. Baek, and K. M. Sung, “Gauss Newton variable forgetting factor recursive least squares for time varying parameter tracking,” *Electronics Lett.*, vol. 36, pp. 988–990, May 2000.

[9] S.-H. Leung and C. F. So, “Gradient-based variable forgetting factor RLS algorithm in time-varying environments,” *IEEE Trans. Signal Processing*, vol. 53, pp. 3141–3150, Aug. 2005.

[10] C. Paleologu, J. Benesty, and S. Ciochină, “A robust variable forgetting factor recursive least-squares algorithm for system identification,” *IEEE Signal Processing Lett.*, vol. 15, pp. 597–600, 2008.

[11] Y. J. Chu and S. C. Chan, “A new local polynomial modeling-based variable forgetting factor RLS algorithm and its acoustic applications,” *IEEE/ACM Trans. Audio, Speech, Language Processing*, vol. 23, pp. 2059–2069, Nov. 2015.

[12] P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems: Numerical Aspects of Linear Inversion*. Philadelphia, PA: SIAM, 1998.

[13] J. Benesty, C. Paleologu, and S. Ciochină, “Regularization of the RLS algorithm,” *IEICE Trans. Fundamentals*, vol. E94-A, pp. 1628–1629, Aug. 2011.

[14] Y. V. Zakharov and V. H. Nascimento, “Sparse sliding-window RLS adaptive filter with dynamic regularization,” in *Proc. EUSIPCO*, 2016, pp. 145–149.

[15] C. Elisei-Iliescu, C. Stanciu, C. Paleologu, J. Benesty, C. Anghel, and S. Ciochină, “Robust variable regularized RLS algorithms,” in *Proc. IEEE HSCMA*, 2017, pp. 171–175.

[16] Y. V. Zakharov and T. C. Tozer, “Multiplication-free iterative algorithm for LS problem,” *IEE Electronics Lett.*, vol. 40, pp. 567–569, Apr. 2004.

[17] Y. V. Zakharov, G. P. White, and J. Liu, “Low-complexity RLS algorithms using dichotomous coordinate descent iterations,” *IEEE Trans. Signal Processing*, vol. 56, pp. 3150–3161, July 2008.

[18] J. Liu, Y. V. Zakharov, and B. Weaver, “Architecture and FPGA design of dichotomous coordinate descent algorithms,” *IEEE Trans. Circuits and Systems I: Regular Papers*, vol. 56, pp. 2425–2438, Nov. 2009.

[19] C. Paleologu, S. Ciochină, and J. Benesty, “Variable step-size NLMS algorithm for under-modeling acoustic echo cancellation,” *IEEE Signal Processing Lett.*, vol. 15, pp. 5–8, 2008.

[20] M. A. Iqbal and S. L. Grant, “Novel variable step size NLMS algorithms for echo cancellation,” in *Proc. IEEE ICASSP*, 2008, pp. 241–244.

[21] *Digital Network Echo Cancellers*, ITU-T Rec. G.168, 2002.

2017, © Copyright by authors, Published under agreement with IARIA - www.iaria.org