Signal Processing 198 (2022) 108599

Contents lists available at ScienceDirect

Signal Processing

journal homepage: www.elsevier.com/locate/sigpro

Finite-window RLS algorithms☆

Lu Shen[a,*], Yuriy Zakharov[a], Maciej Niedźwiecki[b], Artur Gańcza[b]

[a] Department of Electronic Engineering, University of York, UK
[b] The Faculty of Electronics, Telecommunications and Computer Science, Department of Automatic Control, Gdańsk University of Technology, Gdańsk 80-233, Poland

ARTICLE INFO

Article history:
Received 16 February 2022
Revised 20 April 2022
Accepted 21 April 2022
Available online 22 April 2022

Keywords:
Adaptive filter
Finite-window
RLS
Time-varying systems

ABSTRACT

Two recursive least-squares (RLS) adaptive filtering algorithms are most often used in practice, the exponential and sliding (rectangular) window RLS algorithms. This popularity is mainly due to existence of low-complexity versions of these algorithms. However, these two windows are not always the best choice for identification of fast time-varying systems, when the identification performance is most important. In this paper, we show how RLS algorithms with arbitrary finite-length windows can be implemented at a complexity comparable to that of exponential and sliding window RLS algorithms. Then, as an example, we show an improvement in the performance when using the proposed finite-window RLS algorithm with the Hanning window for identification of fast time-varying systems.

1. Introduction

In the adaptive filtering, recursive least-squares (RLS) algorithms are very popular. They possess fast convergence, while the complexity and numerical stability of their implementation can be comparable to that of slower convergent algorithms, such as the least-mean squares algorithms. An RLS algorithm, at every time instant n minimises the cost function

$$J_w(\mathbf{h}) = \sum_{i=-\infty}^{\infty} w(i-n)|e(i)|^2 \;\rightarrow\; \min_{\mathbf{h}} \tag{1}$$

where $e(i) = z(i) - \mathbf{h}^H\mathbf{x}(i)$ is the error signal, $z(i)$ is the desired signal, $\mathbf{x}(i) = [x(i), x(i-1), \ldots, x(i-L+1)]^T$ is the $L \times 1$ regressor vector, and $x(i)$ is the adaptive filter input.

Two RLS algorithms are most often used in practice, the exponential window (ERLS) and sliding window (SRLS) algorithms. The ERLS exploits the (infinite in length) window $w(i)$: $w_{\mathrm{ERLS}}(i) = \lambda^{-i}$ for $i \in (-\infty, 0]$ and $w_{\mathrm{ERLS}}(i) = 0$ otherwise, where $0 < \lambda < 1$ is the forgetting factor, a parameter defining the efficient length of the window. The SRLS uses the finite window $w(i)$: $w_{\mathrm{SRLS}}(i) = 1$ for $i \in [-M+1, 0]$ and $w_{\mathrm{SRLS}}(i) = 0$ otherwise, where M is the length of the (rectangular) sliding window. Using the matrix inversion formula for a low-rank matrix update, these two windows allow the RLS algorithm implementation with a complexity of $\mathcal{O}(L^2)$ arithmetic operations per time instant [1,2]. The complexity can be further reduced to $\mathcal{O}(L)$ arithmetic operations [2–4]. However, as indicated in Niedźwiecki and Ciołek [5], identification of time-varying systems can benefit from using symmetric bell-shaped windows, like the Hamming, Hanning, Parzen, Bartlett, and other windows [6,7]. The main problem of implementing the RLS algorithms with these windows is the high complexity, which in general is $\mathcal{O}(L^3)$ or higher. The main contribution of this paper is to show how RLS algorithms with arbitrary finite window $w(i)$ of length M can be implemented with a complexity dominated by the term $\mathcal{O}((M+L)\log_2(M+L))$. This is achieved by using the fast Fourier transform (FFT) and dichotomous coordinate descent (DCD) iterations [4,8], the later are widely used in adaptive filtering applications such as the active noise control [9], underwater communications [10], power convertion [11], etc. We then demonstrate that other windows, as an example – the Hanning window, can provide a significant improvement in performance when identifying time-varying systems, compared to the exponential and rectangular windows.

When identifying time-varying systems, the excellent performance is achieved by adaptive filters based on the local basis function (LBF) principle [12]. However, such filters are too complicated for practice, with a complexity of at least $\mathcal{O}(P^3L^3)$, where P is the number of basis functions. A similar performance with much lower complexity can be achieved by fast LBF (fLBF) adaptive filters

☆ The work of Y. Zakharov was supported in part by the U.K. EPSRC through Grants EP/V009591/1 and EP/R003297/1. The work of M. Niedźwiecki and A. Gańcza was partially supported by the National Science Center under the agreement UMO-2018/29/B/ST7/00325.

* Corresponding author.
E-mail address: lu.shen@york.ac.uk (L. Shen).

https://doi.org/10.1016/j.sigpro.2022.108599

exploiting two stages, pre-estimation and post-filtering [13]. The pre-estimation stage provides estimates which are almost unbiased but with a large variability; this is obtained by 'inverse filtering' of the estimates yield by the ERLS algorithm [2]. At the post-filtering stage, denoising is carried out to reduce the variability [13]. However, the pre-estimation still introduces some bias, which limits the overall performance of fLBF algorithms when identifying fast time-varying systems. Meanwhile, the fLBF complexity is dominated by the complexity of the ERLS algorithm. In this paper, we show that the finite-window RLS (FRLS) adaptive filter as the pre-estimator can significantly improve the fLBF performance.

This paper is organized as follows. In Section 2, the FRLS algorithm is derived. Section 3 introduces the fLBF algorithm. The identification performance of the FRLS algorithm and the fLBF algorithm with the FRLS pre-estimator are demonstrated by numerical simulation in Section 4. The paper is concluded in Section 5.

Notations: In this paper, we use capital and small bold fonts for matrices and vectors, e.g. $\mathbf{R}$ and $\mathbf{h}$, respectively. We denote the complex conjugate as $(\cdot)^*$, transpose of $\mathbf{h}$ as $\mathbf{h}^T$, and the Hermitian transpose of $\mathbf{h}$ as $\mathbf{h}^H$. The first column of matrix $\mathbf{R}$ is denoted as $\mathbf{R}^{(1)}$. The norm of a vector is denoted as $\|\cdot\|$, the element-wise product of two vectors $\mathbf{u}$ and $\mathbf{v}$ is denoted as $\mathbf{u}\odot\mathbf{v}$, and the Kronecker product of two vectors $\mathbf{u}$ and $\mathbf{v}$ is denoted as $\mathbf{u}\otimes\mathbf{v}$.

2. FRLS adaptive algorithm

The minimization of the cost function in (1), for a window $w(i)$ defined on a support Γ, results in the solution

$$\hat{\mathbf{h}}(i)=\mathbf{R}^{-1}(i)\boldsymbol{\beta}(i), \tag{2}$$

where the $L\times L$ regression matrix $\mathbf{R}(i)$ and the $L\times 1$ cross-correlation vector $\boldsymbol{\beta}(i)$ are given by

$$\mathbf{R}(i)=\textstyle\sum_{k\in\Gamma} w(k)\mathbf{x}(i+k)\mathbf{x}^H(i+k), \tag{3}$$

$$\boldsymbol{\beta}(i)=\textstyle\sum_{k\in\Gamma} w(k)\mathbf{x}(i+k)z^*(i+k). \tag{4}$$

The support Γ for the ERLS algorithm (also known as the FFLS algorithm [14]) is $\Gamma=(-\infty,0]$. The support for the SRLS algorithm (also known as the FDW-RLS algorithm [15]) and FF-FDW-RLS algorithm [16] is $\Gamma=[-M+1,0]$; for the delayed SRLS (SRLSd) algorithm [17], which is a non-causal version of the SRLS algorithm, it is $\Gamma=[-M_0,M_0]$, where $M=2M_0+1$; etc.

Thus, finding the solution in (2), for every time instance i, requires computation of elements of the matrix $\mathbf{R}(i)$ and its inversion, which are most complicated steps of the algorithm. Since $\mathbf{R}(i)$ is a Hermitian matrix, the direct computation in (3) requires about $2ML^2$ real-valued multiply and accumulate (MAC) operations, where it is also taken into account that a complex-valued multiplication requires 4 real-valued multiplications. For the transversal-structured regressor as in (1), $\mathbf{R}(i)$ can be updated as (see proof in Shen et al. [10]): $[\mathbf{R}(i)]_{m+1,n+1}=[\mathbf{R}(i-1)]_{m,n}$, where $\mathbf{R}_{m,n}$ denotes an element of the matrix $\mathbf{R}$ at the mth column and nth row, and $m,n=1,\ldots,L-1$. Therefore, only the first column $\mathbf{R}^{(1)}(i)$ of $\mathbf{R}(i)$ needs an update, which with the direct computation requires $4ML$ operations, this still can be high. However, the complexity of the update can be reduced if using the FFT as follows. The first column of $\mathbf{R}(i)$ is given by

$$\mathbf{R}^{(1)}(i)=\sum_{k\in\Gamma} w(k)x(i+k)\mathbf{x}^{*(i+k)}=\sum_{k\in\Gamma}\tilde{w}(i,k)\mathbf{x}^{*(i+k)},$$

where $\tilde{w}(i,k)=w(k)x(i+k)$, $k\in\Gamma$, are elements of an M-length vector $\tilde{\mathbf{w}}(i)$. The L-length column $\mathbf{R}^{(1)(i)}$ can be considered as convolution of the M-length sequence of elements in $\tilde{\mathbf{w}}(i)$ and the $(M+L)$-length sequence of all elements in the vector $\tilde{\mathbf{x}}^*(i)=[x^*(i-M_0-L),\ldots,x^*(i+M_0)]^T$. The convolution can be efficiently computed in the frequency domain. Specifically, the sequence $\tilde{w}(k)$ is zero-padded to the length $M+L$. Then, the FFTs $\mathbf{s}_w$ and $\mathbf{s}_x$ of $\tilde{\mathbf{w}}(i)$ and $\tilde{\mathbf{x}}^*(i)$ of $(M+L)$-length are computed. Finally, the inverse FFT (IFFT) of their element-wise product $\mathbf{s}_w\odot\mathbf{s}_x$ are computed, whose last L elements represent the column $\mathbf{R}^{(1)(i)}$. In such a case, the complexity is reduced from $4ML$ operations to three FFTs of size $(M+L)$, which is $\mathcal{O}((M+L)\log_2(M+L))$. Another important property of this approach is that the computation of $\mathbf{R}(i)$ is numerically stable, since no recursion is required.

The rest of the algorithm complexity is dominated by computation in (2). For the numerical stability of this computation, it is preferable to avoid the matrix inversion, which is also a computationally demanding operation. Therefore, instead of the matrix inversion and matrix-vector multiplication in (2), we solve the normal equation $\mathbf{R}(i)\mathbf{h}(i)=\boldsymbol{\beta}(i)$ and obtain a (possibly approximate) solution $\hat{\mathbf{h}}(i)$; a direct (precise) solution of the normal equation, e.g., using the Cholesky decomposition, would require $\mathcal{O}(L^3)$ MACs. This can be reduced as follows. Instead of solving the system of equations, we can solve:

$$\mathbf{R}(i)\Delta\mathbf{h}(i)=\mathbf{r}(i), \tag{5}$$

where $\mathbf{r}(i)=\boldsymbol{\beta}(i)-\mathbf{R}(i)\hat{\mathbf{h}}(i-1)$ is a residual vector for instance i when using the solution $\hat{\mathbf{h}}(i-1)$ obtained at the previous instance $i-1$. The solution for instance i is then found as $\hat{\mathbf{h}}(i)=\hat{\mathbf{h}}(i-1)+\Delta\hat{\mathbf{h}}(i)$, where $\Delta\hat{\mathbf{h}}(i)$ is an approximate solution to (5). The benefit of this approach is that a solution to (5) can be found with a few simple iterations, e.g., such as the DCD iterations [4].

With this approach, the most computationally demanding step of computing $\mathbf{r}(i)$ can be simplified if we take into account that, as follows from (3) and (4),

$$\mathbf{r}(i)=\sum_{k\in\Gamma} w(k)e^*(i,k)\mathbf{x}(i+k), \tag{6}$$

where $e(i,k)=z(i+k)-y(i,k)$ is an error signal, and $y(i,k)=\hat{\mathbf{h}}^H(i-1)\mathbf{x}(i+k)$. We denote $\mathbf{y}(i)$ a vector with elements $y(i,k)$, $k\in\Gamma$. These elements represent an M-length sequence, which is a convolution of the L-length sequence of elements in $\hat{\mathbf{h}}(i-1)$ and the $(M+L)$-length sequence of all elements in $\tilde{\mathbf{x}}(i)$. Since the FFT $\mathbf{s}_x$ of $\tilde{\mathbf{x}}^*(i)$ is already available, the computation of $\mathbf{y}(i)$ requires one FFT and one IFFT of length $(M+L)$. The error vector $\mathbf{e}(i)$ with elements $e(i,k)$, $k\in\Gamma$, is then given by $\mathbf{e}(i)=\mathbf{z}(i)-\mathbf{y}(i)$, where $\mathbf{z}(i)$ is a vector with elements $z(i+k)$, $k\in\Gamma$. After computing the vector $\mathbf{e}(i)$, the residual vector $\mathbf{r}(i)$ in (6) can be computed similarly to (5), but, with the available FFT $\mathbf{s}_x$, using only one FFT and one IFFT.

The system of equations in (5) can be solved using the leading DCD algorithm (see details in [4]). At every iteration, the DCD algorithm updates one element of the vector $\Delta\hat{\mathbf{h}}(i)$ corresponding to the element of $\mathbf{r}(i)$ with the maximum magnitude. Four directions of update, $[-1,1,-j,j]$, $j=\sqrt{-1}$, are analysed, and if the cost function can be minimized, the direction providing the minimum is chosen; such an iteration is called successful. If the cost function cannot be reduced after searching over all four directions, the step size is reduced by two. The initial step size H is chosen as a power of two for efficient hardware implementation. The maximum number of times M_b the step size can be reduced is equivalent to the number of bits representing the solution vector. The DCD algorithm requires no multiplication or division and its complexity depends on the number of successful DCD iterations N_u. When identifying a system with time-invariant or slowly time-varying parameters, a small number of updates ($N_u=1,2$) can be used. For idenfication of fast time-varying systems, a higher number of updates can be required; in our examples below, we set $N_u=8$.

The FRLS algorithm is summarized in Table 1. Its complexity is $(18+4N_u)L+16M+M_b$ MACs and 7 FFTs of size $M+L$. The memory requirement of the FRLS algorithm is comparable to that of

Table 1
FRLS algorithm.

Step	Equation
	for $i<0: \mathbf{x}(i)=\mathbf{0}, \hat{\mathbf{h}}(i)=\mathbf{0}$
	$\mathbf{R}(i)=\mathbf{0}, \mathbf{z}(i)=\mathbf{0}$
	for $i=0,1,\ldots,N$
1	Computing the first column of $\mathbf{R}(i)$:
	$\mathbf{s}_w=\mathrm{FFT}\{\tilde{\mathbf{w}}(i)\}$
	$\mathbf{s}_x=\mathrm{FFT}\{\tilde{\mathbf{x}}^*(i)\}$
	$\mathbf{R}^{(1)}(i)=\mathrm{IFFT}\{\mathbf{s}_w\odot\mathbf{s}_x\}$
2	Computing the filter output $y(i,k)$ for $k\in\Gamma$:
	$\mathbf{s}_h=\mathrm{FFT}\{\mathbf{h}(i-1)\}$
	$\mathbf{y}(i)=\mathrm{IFFT}\{\mathbf{s}_x\odot\mathbf{s}_h\}$
3	Computing the error signal:
	$\mathbf{e}(i)=\mathbf{z}(i)-\mathbf{y}(i)$
4	Computing the residual vector:
	$\mathbf{e}_{\mathbf{w}}(i)=\mathbf{e}^*(i)\odot\mathbf{w}$
	$\mathbf{s}_e=\mathrm{FFT}\{\mathbf{e}_{\mathbf{w}}(i)\}$
	$\mathbf{r}(i)=[\mathrm{IFFT}\{\mathbf{s}_e\odot\mathbf{s}_x\}]^*$
6	Solve (5) with DCD iterations
7	$\hat{\mathbf{h}}(i)=\hat{\mathbf{h}}(i-1)+\Delta\hat{\mathbf{h}}(i)$

the SRLS algorithm. The majority of the memory is allocated to the matrix $\mathbf{R}(i)$, which contains L^2 complex-valued elements.

3. fLBF algorithm with FRLS pre-estimator

The fLBF algorithm includes two stages [13]. At the first stage, pre-estimation is carried out to provide unbiased estimates of the system taps. A popular pre-estimate $\tilde{\mathbf{h}}(i)$ is the result of inverse filtering $\tilde{\mathbf{h}}(i)=\frac{1}{1-\lambda}[\hat{\mathbf{h}}_{\mathrm{ERLS}}(i)-\lambda\hat{\mathbf{h}}_{\mathrm{ERLS}}(i-1)]$ of estimates $\hat{\mathbf{h}}_{\mathrm{ERLS}}(i)$ obtained by the ERLS adaptive filter; the inverse filtering is applied to reduce the bias of the pre-estimates [13]. At the second stage of the fLBF algorithm, the pre-estimates are post-filtered to reduce the estimation variance, while keeping the bias small: $\hat{\mathbf{h}}_{\mathrm{fLBF}}(i)=\mathbf{F}(0)\hat{\boldsymbol{\alpha}}_{\mathrm{fLBF}}(i)$ and $\hat{\boldsymbol{\alpha}}_{\mathrm{fLBF}}(i)=\sum_{k=-K_0}^{K_0}\mathbf{F}^H(k)\tilde{\mathbf{h}}(i+k)$, where $\mathbf{F}(k)=\mathbf{I}_L\otimes\mathbf{f}(k)$, $\mathbf{f}(k)=[f_1(k),\ldots,f_P(k)]^T$ is the vector of orthogonal basis functions $f_p(\cdot)$, defined on a time interval $[-K_0,K_0]$, and P is the number of basis functions. In this paper, as an example, we adopt the complex exponential basis set of the form (see Tsatsanis and Giannakis [18], Sayeed and Aazhang [19], Zakharov and Kodanev [20] for a physical justification of such a choice in application to fast-varying communication channels), $\{f_1(j),\ldots,f_P(j)\}=\left\{\frac{1}{\sqrt{K}}e^{\mathrm{i}j\omega_1},\ldots,\frac{1}{\sqrt{K}}e^{\mathrm{i}j\omega_P}\right\}$, where $j\in[-K_0,K_0]$, $\mathrm{i}=\sqrt{-1}$, $\omega_1=0$, $P=2m_0+1$, $K=2K_0+1$, and $\omega_{2l}=-\frac{2\pi l}{K}$, $\omega_{2l+1}=\frac{2\pi l}{K}$, $l=1,\ldots,m_0$. The complexity of the fLBF algorithm is dominated by the ERLS adaptive filtering.

The fLBF algorithm shows high identification performance in time-varying scenarios. However, the performance can be further improved with a better pre-estimator. For this purpose, we propose to use the FRLS algorithm, so that $\tilde{\mathbf{h}}(i)=\hat{\mathbf{h}}_{\mathrm{FRLS}}(i)$.

4. Numerical results

In this section, we compare the identification performance and complexity of the FRLS algorithm with the Hanning window on the support $\Gamma=[-M_0,M_0]$ with that of other algorithms. We consider the following signal model: $z(i)=\mathbf{h}^H(i)\mathbf{x}(i)+n(i)$, where $\mathbf{h}(i)$ is a time-varying impulse response of an unknown system to be identified, $\mathbf{x}(i)$ is the regressor vector with zero-mean uncorrelated complex-valued Gaussian numbers of unit variance and $n(i)$ is a zero-mean complex-valued white Gaussian noise. The $L=50$ system taps are modelled as independent zero-mean unit-variance random processes with a uniform power spectral density within the frequency interval $[-f_{\max},f_{\max}]$. Realizations of the random processes are generated using the FFT-method [21].

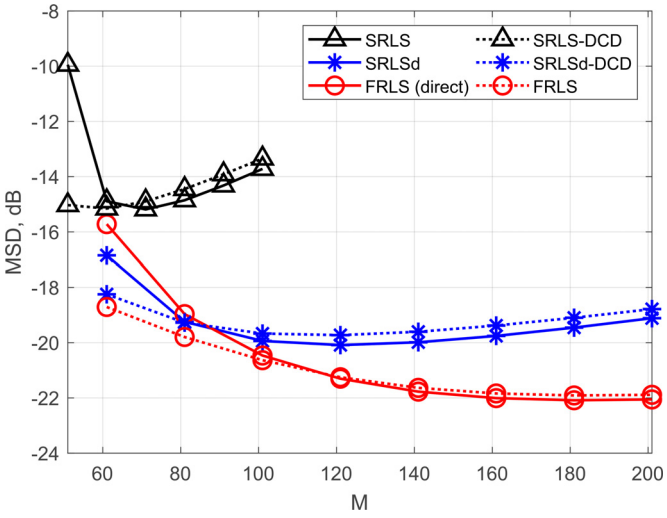

Fig. 1. MSD performance of SRLS and FRLS algorithms when identifying time-varying systems; SNR = 25 dB. Note that FRLS (direct) is the version with the direct solution (2).

The identification performance is evaluated by averaging the mean squared deviation (MSD) over 50 simulation trials. The MSD in every simulation trial is computed as: $\mathrm{MSD}(i)=\|\mathbf{h}(i)-\hat{\mathbf{h}}(i)\|_2^2/E_h$, where $E_h=(1/N)\sum_{i=1}^{N}||\mathbf{h}(i)||^2$ and $N=10^4$ is a number of samples after the algorithm convergence; the MSD is averaged over these N samples.

In the simulations, assuming that the sampling rate is 1000 Hz, we use $f_{\max}=1$ Hz, which is typical for underwater acoustic channels [22]. The signal to noise ratio (SNR) is set to 25 dB.

Fig. 1 compares the MSD performance of the FRLS and SRLS algorithms against the window length M. The classical SRLS algorithm achieves an MSD performance of -15.2 dB when $M=71$. Its non-causal version, the delayed SRLS (SRLSd) algorithm, reduces the MSD to -20.1 dB; note that the SRLSd algorithm is equivalent to the FRLS algorithm with a symmetrical rectangular window. With the Hanning window in the FRLS (direct) algorithm with the direct solution in (2), the MSD is further reduced to -22.1 dB, which is 2.0 dB improvement compared to the SRLSd algorithm and 6.9 dB improvement compared to the classical SRLS algorithm. Note that in this scenario the lowest MSD provided by the ERLS algorithm is -15.8 dB, achieved with the forgetting factor $\lambda=0.94$; this is close to the minimum MSD of the SRLS algorithm.

With $N_u=8$ updates, $H=1$ and $M_b=16$, the DCD-based versions of these algorithms show the MSD performance close to that of their original versions, except at low values of M. At low M, the system of equations becomes ill-conditioned and the DCD solution with its implicit regularization shows a better performance. The minimum MSD of the FRLS algorithm is -21.9 dB when $M=181$.

In Fig. 2, we show the MSD performance of the SRLS and FRLS algorithms under SNR= 5 dB and SNR= 15 dB. The FRLS algorithm outperforms the SRLS algorithm by 5.5 dB and 6.2 dB at SNR= 5 dB and SNR= 15 dB, respectively.

The algorithm complexities at every time instant are summarized in Table 2. They are shown against L in Fig. 3, where the FFT complexity is counted as $4(L+M)\log_2(L+M)$ MACs [23]. The ERLS and SRLS algorithms implemented using DCD iterations [4] have the lowest complexity. Note that the SRLS (SRLS-DCD) and SRLSd (SRLSd-DCD) algorithms have the same complexity. The FRLS complexity is higher than that of the ERLS-DCD and SRLS-DCD algorithms, and this is the payment for the improved performance. However, the FRLS algorithm has a comparable or

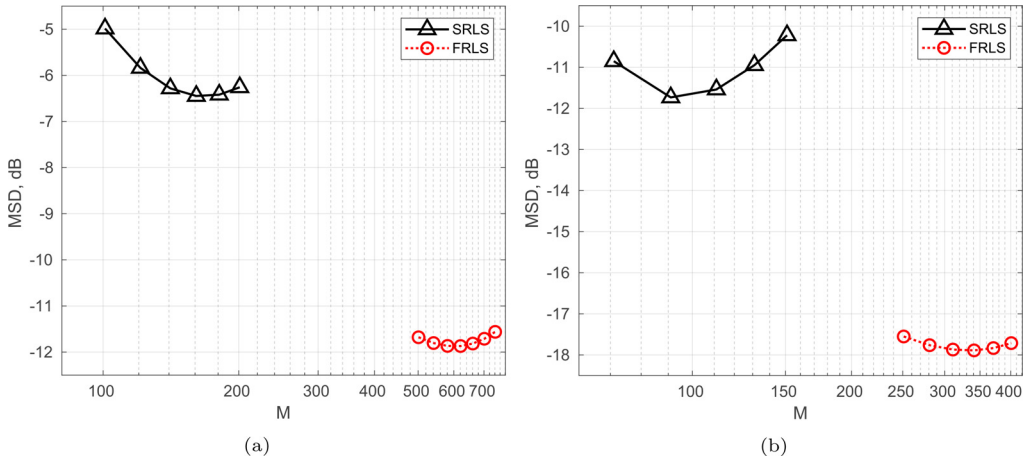

Fig. 2. MSD performance of SRLS and FRLS algorithms when identifying time-varying systems; (a) SNR = 5 dB; (b) SNR = 15 dB.

Table 2
Complexity of the adaptive algorithms.

Adaptive algorithm	MACs	FFTs
ERLS-DCD	$(20+4N_u)L+N_u+M_b$	
SRLS-DCD	$(24+4N_u)L+N_u+M_b$	
FRLS	$(18+4N_u)L+16M+M_b$	7
SRLS	$24L^2+32L$	
ERLS	$12L^2+16L$	
FRLS direct	$2ML(L+1)+4L^3$	

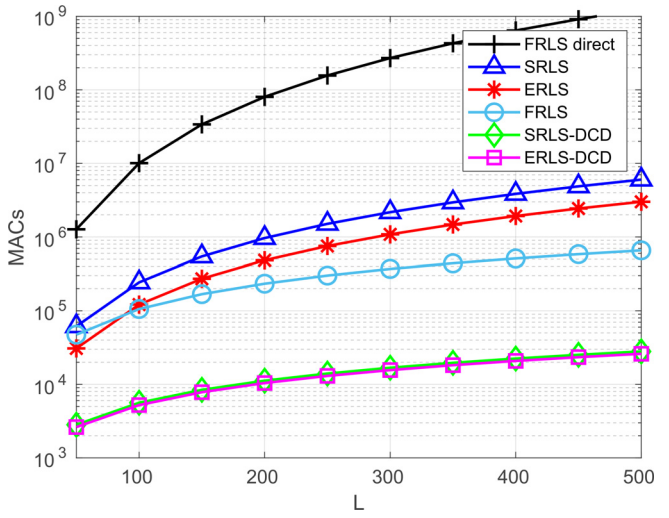

Fig. 3. Complexity of the SRLS and FRLS-based algorithms; the window length is $M = 3L+1$.

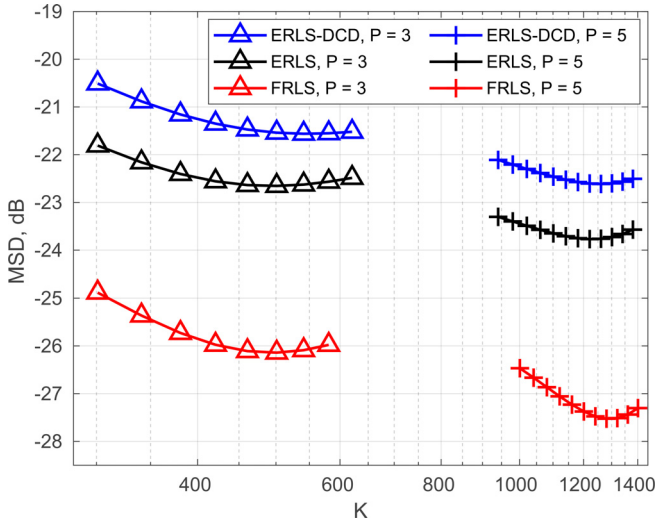

Fig. 4. The performance of the fLBF algorithms with different pre-estimators when identifying fast-varying channels, K is the approximation interval used for the post-filtering.

lower complexity than the classical SRLS algorithm, and significantly lower complexity than the direct implementation of the solution in (2), especially for high L.

We now show that the FRLS algorithm with the Hanning window as a pre-estimator in the fLBF algorithm can significantly improve the fLBF identification performance compared to the use of the ERLS algorithm for this purpose. As recommended in [24], the forgetting factor of the ERLS algorithm is set to $\lambda = \max(0.9, 1-2/L) = 0.96$. In the FRLS algorithm, we set $M = 3L+1 = 151$. We consider the cases with $P = 3$ and $P = 5$ basis functions. Fig. 4 shows the MSD performance of the fLBF algorithm with different pre-estimators against K. The original fLBF algorithm (with the ERLS pre-estimator), provides an MSD of -22.7 dB and -23.8 dB for $P = 3$ and $P = 5$, respectively. The FRLS pre-estimator with $N_u = 8$ DCD iterations shows an improvement of, respectively, 3.5 dB and 3.8 dB against the classical-ERLS pre-estimator. Against the ERLS-DCD pre-estimator, this improvement is higher, 4.6 dB and 4.9 dB, respectively.

5. Conclusion

We have proposed the FRLS algorithm that allows RLS adaptive filtering with any finite-length window in the cost function (1) to be implemented at a complexity dominated by the term $\mathcal{O}((M+L)\log_2(M+L))$, which is comparable or lower than that of the classical ERLS and SRLS algorithms. As demonstrated by the simulation results, the use of a non-uniform window in the FRLS algorithm, such as the Hanning window, significantly improves the identification performance compared to that of the classical ERLS and SRLS algorithms in time-varying scenarios. Same conclusion has been reached when the FRLS algorithm is used as the pre-estimator of the fLBF algorithm. The Matlab code of the FRLS algorithm is provided in [25].

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Lu Shen: Software, Investigation, Validation, Writing – original draft. **Yuriy Zakharov:** Conceptualization, Supervision, Writing – review & editing. **Maciej Niedźwiecki:** Writing – review & editing. **Artur Gańcza:** Writing – review & editing.

References

[1] S. Haykin, Adaptive Filter Theory, Prentice Hall, 2002.
[2] A.H. Sayed, Fundamentals of Adaptive Filtering, John Wiley & Sons, 2003.
[3] J. Cioffi, T. Kailath, Fast, recursive-least-squares transversal filters for adaptive filtering, IEEE Trans. Acoust. J. Speech Signal Process. 32 (2) (1984) 304–337.
[4] Y.V. Zakharov, G.P. White, J. Liu, Low-complexity RLS algorithms using dichotomous coordinate descent iterations, IEEE Trans. Signal Process. 56 (7) (2008) 3150–3161.
[5] M. Niedźwiecki, M. Ciołek, On adaptive covariance and spectrum estimation of locally stationary multivariate processes, Automatica 82 (2017) 1–12.
[6] R.B. Blackman, J.W. Tukey, The measurement of power spectra from the point of view of communications engineering part I, Bell Syst. Tech. J. 37 (1) (1958) 185–282.
[7] F.J. Harris, On the use of windows for harmonic analysis with the discrete fourier transform, Proc. IEEE 66 (1) (1978) 51–83.
[8] J. Liu, Y.V. Zakharov, B. Weaver, Architecture and FPGA design of dichotomous coordinate descent algorithms, IEEE Trans. Circuits Syst. I Regul. Pap. 56 (11) (2009) 2425–2438.
[9] F. Albu, M. Bouchard, Y. Zakharov, Pseudo-affine projection algorithms for multichannel active noise control, IEEE Trans. Audio Speech Lang. Process. 15 (3) (2007) 1044–1052.
[10] L. Shen, Y. Zakharov, L. Shi, B. Henson, BEM Adaptive filtering for SI cancellation in full-duplex underwater acoustic systems, Signal Process. 191 (2022) 108366–108378.
[11] M. Al-Greer, M. Armstrong, M. Ahmeid, D. Giaouris, Advances on system identification techniques for DC–DC switch mode power converter applications, IEEE Trans. Power Electron. 34 (7) (2018) 6973–6990.
[12] M. Niedźwiecki, M. Ciołek, Generalized savitzky–golay filters for identification of nonstationary systems, Automatica 108 (2019) 108477.
[13] M.J. Niedźwiecki, M. Ciołek, A. Gańcza, A new look at the statistical identification of nonstationary systems, Automatica 118 (2020) 109037.
[14] F. Ding, T. Chen, Performance bounds of forgetting factor least-squares algorithms for time-varying systems with finite measurement data, IEEE Trans. Circuits Syst. I Regul. Pap. 52 (3) (2005) 555–566.
[15] J. Ma, R. Ding, Recursive computational formulas of the least squares criterion functions for scalar system identification, Appl. Math. Model. 38 (1) (2014) 1–11.
[16] F. Ding, Y. Xiao, A finite-data-window least squares algorithm with a forgetting factor for dynamical modeling, Appl. Math. Comput. 186 (1) (2007) 184–192.
[17] L. Shen, Y. Zakharov, B. Henson, N. Morozs, P. D. Mitchell, Adaptive filtering for full-duplex UWA systems with time-varying self-interference channel, IEEE Access 8 (2020) 187590–187604.
[18] M.K. Tsatsanis, G.B. Giannakis, Modelling and equalization of rapidly fading channels, Int. J. Adapt. Control Signal Process. 10 (2–3) (1996) 159–176.
[19] A.M. Sayeed, B. Aazhang, Joint multipath-doppler diversity in mobile wireless communications, IEEE Trans. Commun. 47 (1) (1999) 123–132.
[20] Y. Zakharov, V. Kodanev, Multipath-doppler diversity of OFDM signals in an underwater acoustic channel, IEEE Int. Conf. Acoust. Speech Signal Process. 5 (2000) 2941–2944.
[21] D.B. Percival, Simulating Gaussian random processes with specified spectra, Comput. Sci. Stat. 24 (1992) 534–538.
[22] M. Stojanovic, J. Preisig, Underwater acoustic communication channels: propagation models and statistical characterization, IEEE Commun. Mag. 47 (1) (2009) 84–89.
[23] P. Duhamel, H. Hollmann, Split-radix FFT algorithm, Electron. Lett. 20 (1) (1984) 14–16.
[24] M.J. Niedźwiecki, M. Ciołek, A. Gańcza, P. Kaczmarek, Application of regularized Savitzky–Golay filters to identification of time-varying systems, Automatica 133 (2021) 109865.
[25] Finite-Window RLS Algorithm Matlab - code, doi:10.24433/CO.0692087.v1, accessed: 2022-03-01.