BEM Adaptive Filtering for SI Cancellation in Full-Duplex Underwater Acoustic Systems

Lu Shen[a], Yuriy Zakharov[a], Long Shi[b], Benjamin Henson[a]

[a] Department of Electronic Engineering, University of York, UK
[b] School of Economic Information Engineering, Southwestern University of Finance and Economics, China

Abstract

Self-interference (SI) cancellation (SIC) is the key technology for achieving full-duplex (FD) communications in underwater acoustic systems. In practice, SI channels are often fast-varying, e.g., due to reflections from surface waves. Classical adaptive filters, such as the recursive least squares (RLS) algorithm, predict the channel impulse response when used for channel estimation. If a tracking delay is acceptable, interpolating estimators capable of providing more accurate estimates of time-varying impulse responses can be used. Interpolating estimators with good tracking performance are normally of high complexity. In this paper, we propose low-complexity interpolating adaptive filters which combine the basis expansion model (BEM) approach with the sliding-window RLS (SRLS) algorithm. Specifically, we use the Legendre polynomials as the basis functions and solve the system of equations using dichotomous coordinate descent (DCD) iterations, thus the name the SRLS-L-DCD adaptive filter. A sparse algorithm (HSRLS-L-DCD) based on homotopy iterations is then proposed to exploit the sparsity in the expansion coefficients. The identification performance of the adaptive filters is investigated by a simulation which mimics an FD lake experiment. Both the simulation and lake experiments show that significant improvement in the SIC performance is achieved with the proposed low-complexity algorithms compared to the classical SRLS algorithm.

Keywords: Adaptive filter, full-duplex, low-complexity, time-varying channel estimation, underwater acoustic systems

[⋆] The work of Y. Zakharov and B. Henson was supported in part by the U.K. EPSRC through Grants EP/P017975/1 and EP/R003297/1.

[*] Corresponding author
Email address: ls1215@york.ac.uk (Lu Shen)

1. Introduction

Full-duplex (FD) operation allows simultaneous transmission and reception of signals in the same frequency bandwidth by closely positioned transmit and receive antennas [1, 2]. The FD operation can significantly increase the throughput of underwater acoustic (UWA) systems. The key problem in FD implementation is to cancel the strong self-interference (SI) from the near-end transmitter [3, 4]. Digital SI cancellation (SIC) is considered as a promising practical approach for FD UWA systems due to relatively low (compared to radio systems) frequencies of acoustic signals. Digital SIC recovers the near-end SI signal and subtracts it from the received signal. The signal recovery is based on the SI channel estimates.

In UWA communication systems, for channel estimation, recursive least squares (RLS) adaptive filters are normally used [5–7]. Classical RLS adaptive filters can efficiently estimate the SI channel in time-invariant environments [8]. However, in practice, the SI channel can be fast time-varying [4], especially when the transmit and receive antennas are positioned close to the sea/lake surface [9]. The performance of the classical RLS adaptive filters is limited in fast-varying channels.

Classical adaptive filtering algorithms such as the RLS algorithm, when used in identification scenarios, predict the channel response for the next time instant based on input data received at the current and past time instants. In general, for time-varying channels, predictive estimators are less accurate than interpolating estimators. The later, however, are non-causal since they require input data not only from the past but also from future time instants. If an application can accept a tracking delay, the interpolating adaptive filtering can significantly improve the identification performance.

In communication systems, basis expansion models (BEMs) are widely used for block (non-adaptive) estimation of time-varying channels, e.g. see [10–19]. The most often used BEMs are the Karhunen-Loeve functions [11, 12], discrete prolate spheroidal functions [14, 17], generalized complex exponentials [13–15], B-splines [16, 20], and algebraic polynomials [18, 21] including Legendre polynomials [22, 23]. With a BEM, estimation of a realization of the random process describing the time-variant channel is transformed into estimation of a vector of time-invariant expansion coefficients [17].

In [24], a predictive RLS adaptive filter was proposed based on representation of the time variation with algebraic polynomials. In [25], for the UWA FD application, an interpolating adaptive filter based on the sliding-window RLS (SRLS) algorithm (SRLS-P) was proposed. The SRLS-P algorithm exploits a parabolic approximation of the time-varying SI channel response. It first estimates the channel using the SRLS algorithm and then computes expansion coefficients by solving a system of equations derived based on the initial channel estimates. This algorithm has a high complexity. Even if the system of equations is solved using dichotomous coordinate descent (DCD) iterations, the rest of the computation still requires about $O(ML^2)$ plus $O(L^3)$ MAC (multiply and accumulate) operations per sample, where L is the filter length and M is the sliding window length. In [18], a local basis function (LBF) estimator is proposed which combines the BEM and weighted least squares approaches. The LBF estimator provides an excellent tracking performance at the expense of high computational complexity. A recursive computation is proposed to reduce the LBF estimator complexity for real-valued data for specific choices of basis functions; however, the overall complexity is still high. In [19], a fast version of the LBF estimator (fLBF) is proposed with two steps, pre-estimation and post-filtering. The key idea is to find an initial estimate of the system impulse response by the exponentially weighted RLS (ERLS) algorithm, and then denoise the pre-estimated response by the LBF estimator. It is indicated in [19] that the fLBF algorithm can provide approximately the same performance as that of the LBF algorithm with significantly lower complexity under certain assumptions. This is verified by simulation results for a two-tap system. We show in Section 5 that the performance of the fLBF estimator in FD scenarios, with a large number of taps, is not as good as that of the LBF estimator.

In this paper, we propose the SRLS-L adaptive filter exploiting the BEM approach, which solves the same optimization problem as the LBF estimator, but with a significantly lower complexity. The SRLS-L algorithm can use any basis functions, while the SRLS-P algorithm proposed in [25] is using the second order algebraic polynomial. Although the approach to computation of expansion coefficients in the SRLS-P and SRLS-L algorithms is different, the latter can be considered as a generalization of the SRLS-P algorithm. The use of higher orders of basis functions in the SRLS-L algorithm allows the improvement of the tracking performance in fast-varying channels. Moreover, the complexity is reduced by using recursive computations for the matrices and vectors, using fast Fourier transforms (FFTs) for convolution, and further reduced by solving systems of equations recursively using DCD iterations, which leads to the SRLS-L-DCD algorithm. The DCD algorithm has been previously used to derive adaptive algorithms characterized by low complexity, but also they are numerically stable and well suited to hardware implementation [26–31], e.g., on FPGA design platforms [32].

For fast-varying channels with a large delay spread, the performance of the SRLS-L adaptive filter could be limited due to the increased number of parameters to be estimated when high orders of basis functions are used. UWA channels normally exhibit a sparse multipath structure. Therefore, sparse recovery algorithms are normally used for exploiting the sparsity in the channel impulse response. In this paper, we consider the SI channel in FD UWA systems. The performance of an FD system is highly dependent on the SIC, which in turn depends on the accuracy of the SI channel estimation. The level of sparseness in the SI channel may not be high due to the multiple reflections from underwater objects in the vicinity of the closely positioned transducer and hydrophone. However, different multipath components may have different speed of time variations.

The direct path between the transmitter and receiver and other paths with reflection from static objects can be slowly varying in time, whereas paths with reflection from a moving sea/lake surface can be fast-varying [9]. Therefore, in addition to the normal sparsity of the UWA channel [33], there will be an extra sparsity in the basis expansion coefficients. This sparsity can be exploited to improve the SIC performance. For sparse recovery, convex optimization and greedy methods are normally used, the latter methods are of lower complexity and well-suited for hardware implementation [34–36]. In this paper, we apply a greedy algorithm, the homotopy algorithm [37], which provides high performance in sparse recovery and possesses low-complexity, for exploiting the sparsity in the expansion coefficients and propose the homotopy SRLS-L-DCD (HSRLS-L-DCD) adaptive filter. We will show in Sections 5 and 6 that the HSRLS-L-DCD algorithm outperforms the LBF estimator in the tracking performance.

In this paper, we propose the SRLS-L adaptive filter which provides the same tracking performance as the LBF estimator at significantly lower complexity. To exploit the sparsity in the expansion coefficients, the homotopy SRLS-L-DCD (HSRLS-L-DCD) adaptive filter is proposed, which outperforms the LBF estimator in the tracking performance. The contributions of this paper are as follows:

1. A computationally efficient SRLS-L adaptive filter based on the SRLS algorithm and Legendre polynomials is proposed for identification of time-varying systems.
2. The SRLS-L-DCD adaptive algorithm is proposed to further reduce the complexity by solving the system of equations recursively using the DCD iterations.
3. The HSRLS-L-DCD algorithm is proposed based on the SRLS-L algorithm, homotopy iterations and DCD iterations to exploit the sparsity in the expansion coefficients.
4. The complexity analysis of the SRLS-L, SRLS-L-DCD and HSRLS-L-DCD adaptive filters is presented.
5. The performance of the proposed adaptive filters is evaluated in simulation and lake experiments. Results indicate that the proposed algorithms significantly improve the channel estimation performance compared to the classical SRLS algorithm.

The rest of the paper is organized as follows. In Section 2, the general structure of the digital SI canceller is described. In Section 3, the SRLS-L adaptive filter is derived and techniques proposed to reduce its complexity are described. Section 4 presents the proposed HSRLS-L-DCD adaptive filter. Section 5 investigates the channel estimation performance of the proposed algorithms and known algorithms in an FD scenario. In Section 6, the SIC performance of the proposed algorithms is investigated in FD lake experiments. In Section 7, conclusions are drawn. Complexity of the HSRLS-L-DCD and LBF algorithms is analysed in Appendices A and B, respectively.

Notations: In this paper, we use capital and small bold fonts for matrices and vectors, respectively; e.g, $\mathbf{R}$ and $\mathbf{h}$. The calligraphic bold font $\boldsymbol{\mathcal{R}}$ also represents a matrix. We

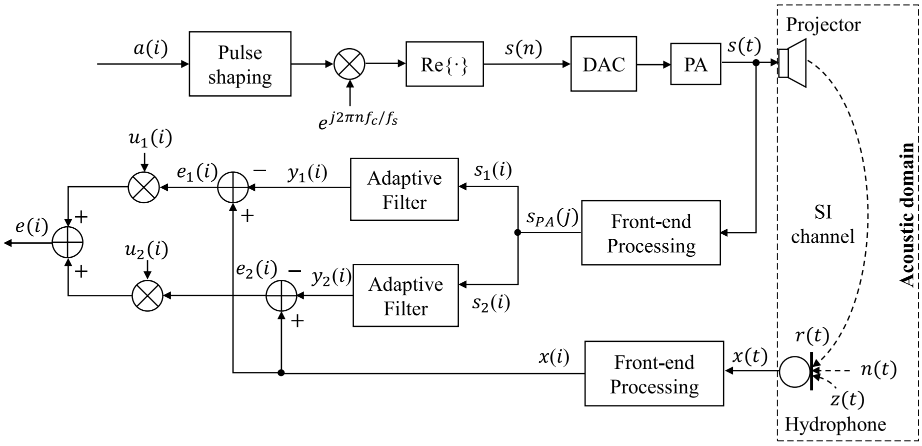


Figure 1: Block diagram of the digital SI canceller. The sample index with sampling rate f_s, symbol rate f_d and $2f_d$ are denoted by n, i and j, respectively.

denote the complex conjugate as $(\cdot)^*$, transpose of $\mathbf{h}$ as $\mathbf{h}^T$, and the Hermitian transpose of $\mathbf{h}$ as $\mathbf{h}^H$. An $L \times L$ identity matrix is denoted as $\mathbf{I}_L$. The pth column of matrix $\mathcal{R}$ is denoted as $[\mathcal{R}]_{:,p}$. The norm of a vector is denoted as $\|\cdot\|$ and $\|\mathbf{h}\|^2 = \mathbf{h}^H\mathbf{h}$. The diagonal of a matrix is denoted as diag$\{\cdot\}$. The expectation operation is denoted as $E\{\cdot\}$ and the Kronecker product is denoted as $\otimes$. The cardinality of I is denoted as $|I|$.

2. General structure of the digital SI canceller

In this section, a general structure of the digital SI canceller is presented.

The block diagram of the digital SI canceller is shown in Fig. 1. Complex-valued data symbols $a(i)$ are transmitted at a symbol rate f_d, pulse-shaped and up-sampled to high sampling rate f_s using a root raised-cosine (RRC) filter, and up-shifted to the carrier frequency f_c. The passband signal $s(n)$ is digital-to-analogue converted (DAC), amplified in the power amplifier (PA) and emitted by a projector. The received signal $x(t)$ at the hydrophone contains the SI signal $r(t)$ together with noise $n(t)$ and far-end signal $z(t)$. At the receiver, the received signal after analog-to-digital conversion (ADC) is down shifted in frequency, low-pass filtered and down sampled to the symbol rate f_d. These samples are applied to the adaptive filter as the desired signal $x(i)$. The same operation is performed on the PA output, which is used as the reference signal to incorporate the PA nonlinearity into the input signal of the adaptive filter to improve the digital cancellation performance [8].

To avoid the signal distortion caused by the up- and down-sampling effect, the PA output is down-sampled to twice the symbol rate, $s_{PA}(j)$, and interleaved into two branches.

The interleaved symbol-rate signals $s_1(i)$ and $s_2(i)$ are then used as the (regressor) input to the adaptive filter on the two branches. The error signals at the two branches are combined with weights, $u_1(i)$ and $u_2(i)$, computed based on the residual signal power in the branches. More details on the multi-branch combining scheme can be found in [38].

This structure is adopted for digital cancellation when investigating the FD UWA system performance with lake experimental data in Section 6.

3. Low-complexity SRLS-L adaptive filter

In subsection 3.1, we introduce the signal model. In subsection 3.2, we describes the BEM based SRLS algorithm in a general form and specify it for Legendre polynomials. Finally, in subsection 3.3, we propose techniques for reducing the algorithm complexity by exploiting the time-shifted structure of the regressor.

3.1. Signal model

The desired signal $x(i)$ is given by

$$x(i) = \mathbf{h}^H(i)\mathbf{s}(i) + n(i), \tag{1}$$

where $\mathbf{s}(i) = [s(i), s(i-1), \ldots, s(i-L+1)]^T$ is an $L \times 1$ regressor vector, $s(i)$ is the input signal to the system (channel) with a time-varying impulse response $\mathbf{h}(i)$ to be identified, and $n(i)$ is a noise signal. To simplify the notation in the derivation, we use $s(i)$ to denote the input signals $s_1(i)$ and $s_2(i)$ on the first and second branch in Fig. 1. For derivation in this section, we will ignore the noise $n(i)$.

We assume that within a time interval $[i - M_o,\ i + M_o]$ centred at the time instant i, the time-varying response can be accurately approximated by $(P+1)$ basis functions $\phi_p(k)$ [18]:

$$\mathbf{h}(i+k) = \sum_{p=0}^{P} \mathbf{c}_p(i)\phi_p(k), \quad k = -M_o, \ldots, M_o, \tag{2}$$

where $M_o = (M-1)/2$ and M is the sliding window length; here, for convenience of presentation, we will assume that M is odd, however the results can be easily extended to the case of even M. The $L \times 1$ vectors of expansion coefficients $\mathbf{c}_p(i)$ should be estimated, where L is the length of the impulse response.

The orthogonal Legendre algebraic polynomials, which we will be using as an example BEM, are defined as [39]:

$$L_p(\tau) = \frac{1}{2^p p!}\frac{d^p}{d\tau^p}\left[(\tau^2-1)^p\right], \ -1 < \tau < 1, \ p \geq 0. \tag{3}$$

Fig. 2 shows the first four Legendre polynomials. The basis functions $\phi_p(k)$ are then given by

$$\phi_p(k) = L_p\left(\frac{2k}{M-1}\right). \tag{4}$$

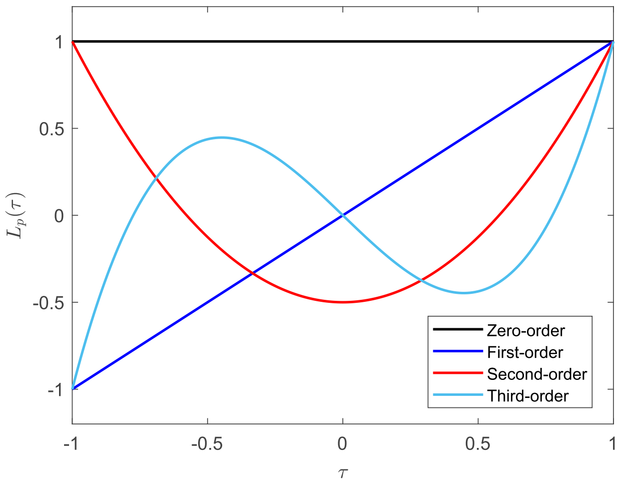


Figure 2: Legendre polynomials.

3.2. SRLS-L algorithm

We now consider the following vectors:

$$\mathbf{b}_p(i) = \mathbf{S}^T(i)\mathbf{W}\mathbf{\Phi}_p\mathbf{x}^*(i),\ p = 0, \ldots, P, \tag{5}$$

where $\mathbf{\Phi}_p = \text{diag}\{\phi_p(M_o), \ldots, \phi_p(-M_o)\}$ is an $M \times M$ diagonal matrix, $\mathbf{S}(i) = [\mathbf{s}(i), \ldots, \mathbf{s}(i-M+1)]^T$ is the $M \times L$ regressor matrix, $\mathbf{x}(i) = [x(i), x(i-1), \ldots, x(i-M+1)]^T$ is an $M \times 1$ desired signal vector, and $\mathbf{W} = \text{diag}\{w(M_0), \ldots, w(-M_0)\}$ is an $M \times M$ diagonal matrix, where the diagonal elements (weights) form a non-negative symmetric bell-shaped window. The weights are applied to put more emphasis on the data close to the middle of the time window [18, 19]. The vector $\mathbf{b}_p(i)$ can be considered as a generalized cross-correlation vector. When we use the zero-order Legendre polynomial and rectangular weighting, i.e, when both $\mathbf{W}$ and $\mathbf{\Phi}_0$ are identity matrices, the vector $\mathbf{b}_0(i)$ is the cross-correlation between the regressor and desired signals, which is exactly the same as that in the classical SRLS algorithm. Equation (5) can be written as:

$$\begin{aligned}
\mathbf{b}_p(i) &= \mathbf{S}^T(i)\mathbf{W}\mathbf{\Phi}_p\mathbf{s}^H(i)\mathbf{h}(i) \\
&= \sum_{k=0}^{2M_o} w(M_o-k)\phi_p(M_o-k)\mathbf{s}(i-k)\mathbf{s}^H(i-k)\mathbf{h}(i-k) \\
&= \sum_{k=0}^{2M_o} \psi_p\left(M_o-k\right)\mathbf{s}(i-k)\mathbf{s}^H(i-k)\mathbf{h}(i-k) \\
&= \sum_{k=-2M_o}^{0} \psi_p\left(M_o+k\right)\mathbf{R}(i+k)\mathbf{h}(i+k), \qquad (6)
\end{aligned}$$

where $\psi_p(k) = w(k)\phi_p(k)$, and the last equality is obtained by replacing k with $-k$, $\mathbf{R}(i) = \mathbf{s}(i)\mathbf{s}^H(i)$ and, further replacing i with $i + M_o$,

$$\mathbf{b}_p\left(i + M_o\right) = \sum_{k=-M_o}^{M_o} \psi_p(k)\mathbf{R}(i+k)\mathbf{h}(i+k). \tag{7}$$

By substituting (2) into (7), we obtain

$$\begin{aligned}\mathbf{b}_p\left(i + M_o\right) &= \sum_{k=-M_o}^{M_o} \psi_p(k)\mathbf{R}(i+k)\sum_{q=0}^{P}\mathbf{c}_q(i)\phi_q(k) \\ &= \sum_{q=0}^{P}\mathcal{R}_{p,q}(i)\mathbf{c}_q(i),\end{aligned} \tag{8}$$

where

$$\begin{aligned}\mathcal{R}_{p,q}(i) &= \sum_{k=-M_o}^{M_o} \psi_p(k)\mathbf{R}(i+k)\phi_q(k) \\ &= \sum_{k=-M_o}^{M_o} w(k)\phi_p(k)\mathbf{R}(i+k)\phi_q(k) \\ &= \sum_{k=-M_o}^{M_o} \varphi_p(k)\mathbf{R}(i+k)\varphi_q(k),\end{aligned} \tag{9}$$

and $\varphi_p(k) = \sqrt{w(k)}\phi_p(k)$. By further denoting

$$\mathcal{R}(i) = \begin{bmatrix} \mathcal{R}_{0,0}(i) & \dots & \mathcal{R}_{0,P}(i) \\ \dots & \dots & \dots \\ \mathcal{R}_{P,0}(i) & \dots & \mathcal{R}_{P,P}(i) \end{bmatrix}, \tag{10}$$

$\mathbf{b}(i) = [\mathbf{b}_0^T(i+M_o), \dots, \mathbf{b}_P^T(i+M_o)]^T$ and $\mathbf{c}(i) = [\mathbf{c}_0^T(i), \dots, \mathbf{c}_P^T(i)]^T$, we obtain a system of equations with respect to the unknown $(P+1)L \times 1$ vector $\mathbf{c}(i)$:

$$\mathcal{R}(i)\mathbf{c}(i) = \mathbf{b}(i). \tag{11}$$

By solving this system, we find an estimate $\hat{\mathbf{c}}(i)$ of the expansion coefficients $\mathbf{c}(i)$ for representation of the time-varying impulse response $\mathbf{h}(i)$ in (2). As follows from (2), we are only interested in the estimate for $k = 0$. Therefore, the channel estimate at time instant i is given by

$$\hat{\mathbf{h}}(i) = \sum_{p=0}^{P}\hat{\mathbf{c}}_p(i)\phi_p(0). \tag{12}$$

The SRLS-L algorithm is summarized in Table 1, where $\varepsilon > 0$ is a regularization parameter used to stabilize the solution of the system in (11). Note that the sliding

Table 1: SRLS-L algorithm

Step	**Equation**
	for $i > 0$, repeat:
1	Generate vectors $\mathbf{b}_p(i + M_o)$ using (5)
2	Compute matrices $\mathcal{R}_{p,q}(i)$ as in (9)
3	Generate the matrix $\mathcal{R}(i)$ as in (10) and vector $\mathbf{b}(i)$
4	Find a solution $\hat{\mathbf{c}}$ to the system $\left[\mathcal{R}(i) + \varepsilon \mathbf{I}_{(P+1)L}\right] \mathbf{c}(i) = \mathbf{b}(i)$
5	Compute the estimate $\hat{\mathbf{h}}(i) = \sum_{p=0}^{P} \hat{\mathbf{c}}_p(i)\phi_p(0)$

window length M in the SRLS-L algorithm should satisfy the condition $M > (P + 1)L$; otherwise, the matrix $\mathcal{R}(i)$ will have a rank less than $(P + 1)L$ and the system (11) will be ill-conditioned.

In [18], the LBF estimator is proposed under the same concept. However, the LBF estimator is only applicable for real-valued data and it is of a high complexity. A recursive computation scheme was proposed in [18] to reduce the complexity of the LBF estimator, but the complexity is still high (see complexity analysis in Appendix B). In addition, to allow the recursive computation, there are constraints on the choice of basis functions. On the other hand, the adaptive filter that we propose here has no constraint on the choice of basis functions, it is also designed for complex-valued systems and its complexity is lower than that of the LBF estimator. We show in subsection 3.3 how the complexity of the SRLS-L adaptive filter can be significantly reduced.

3.3. Complexity of the SRLS-L adaptive filter

In this subsection, the complexity analysis is presented for the complex-valued SRLS-L algorithm and techniques are proposed to reduce the complexity.

The complexity of the SRLS-L algorithm is mainly determined by the computation of the matrices $\mathcal{R}_{p,q}(i)$ in (9) and vectors $\mathbf{b}_p(i + M_o)$ in (5) and solving the system of equations (11). We first show how the complexity of the matrix and vector computation can be reduced and then discuss a reduction in complexity by recursively solving the system of equations.

3.3.1. Matrix $\mathcal{R}_{p,q}(i)$

The direct computation of the matrix $\mathcal{R}_{p,q}(i)$ would require $4ML^2$ MAC operations. We will show that this can be reduced by recursive computations and by using the fast Fourier transforms (FFTs) of size $(L+M)$. Firstly, we show that $\mathcal{R}_{p,q}(i)$ can be efficiently computed using elements in $\mathcal{R}_{p,q}(i - 1)$.

Proposition: The following relationship holds for elements $[\mathcal{R}_{p,q}(i)]_{m+1,n+1}$, $m, n = 1, \ldots, L - 1$, of the matrix $\mathcal{R}_{p,q}(i)$:

$$[\mathcal{R}_{p,q}(i)]_{m+1,n+1} = [\mathcal{R}_{p,q}(i - 1)]_{m,n}. \tag{13}$$

Proof: The matrix $\mathcal{R}_{p,q}(i)$ in (9) can be represented as:

$$\mathcal{R}_{p,q}(i) = \sum_{k=-\infty}^{\infty} v_{p,q}(k)\mathbf{R}(i+k), \tag{14}$$

where coefficients $v_{p,q}(k)$ are defined as:

$$v_{p,q}(k) = \begin{cases} w(k)\phi_p(k)\phi_q(k), & \text{if } -M_o \le k \le M_o \\ 0, & \text{otherwise} \end{cases}. \tag{15}$$

This can also be rewritten as:

$$\mathcal{R}_{p,q}(i) = \sum_{k=-\infty}^{\infty} v_{p,q}(k-i)\mathbf{R}(k). \tag{16}$$

The $(m+1, n+1)$th element of $\mathcal{R}_{p,q}(i)$ is given by

$$[\mathcal{R}_{p,q}(i)]_{m+1,n+1} = \sum_{k=-\infty}^{\infty} v_{p,q}(k-i)[\mathbf{R}(k)]_{m+1,n+1}. \tag{17}$$

Since $[\mathbf{R}(k)]_{m+1,n+1} = s(k-m)s^*(k-n)$, we have

$$[\mathcal{R}_{p,q}(i)]_{m+1,n+1} = \sum_{k=-\infty}^{\infty} v_{p,q}(k-i)s(k-m)s^*(k-n). \tag{18}$$

The (m, n)th element of $\mathcal{R}_{p,q}(i-1)$ is given by

$$[\mathcal{R}_{p,q}(i-1)]_{m,n} = \sum_{k=-\infty}^{\infty} v_{p,q}(k-i+1)[\mathbf{R}(k)]_{m,n}. \tag{19}$$

Since $[\mathbf{R}(k)]_{m,n} = s(k-m+1)s^*(k-n+1)$, we obtain

$$\begin{aligned} &[\mathcal{R}_{p,q}(i-1)]_{m,n} \\ = &\sum_{k=-\infty}^{\infty} v_{p,q}(k-i+1)s(k-m+1)s^*(k-n+1) \\ = &\sum_{k=-\infty}^{\infty} v_{p,q}(k-i)s(k-m)s^*(k-n) \\ = &[\mathcal{R}_{p,q}(i)]_{m+1,n+1}. \end{aligned} \tag{20}$$

This equality holds for $m, n = 1, \ldots, L-1$.

This proposition shows that $(L-1)^2$ elements of the matrix $\mathcal{R}_{p,q}(i)$ at time instant i are the same as elements of the matrix $\mathcal{R}_{p,q}(i-1)$ at time instant $i-1$. Therefore, only $2L-1$ elements of the matrix $\mathcal{R}_{p,q}(i)$ require computation. Since the matrix $\mathcal{R}_{p,q}(i)$ is

Hermitian, only one column of length L should be updated. Thus, the complexity of the computation of $\mathcal{R}_{p,q}(i)$ is reduced from $4ML^2$ to $4ML$ MACs.

We now show how the FFT can be used to further reduce the complexity. We only need to compute the first column of the matrix $\mathcal{R}_{p,q}(i)$. This column can be represented as

$$\begin{aligned}[\mathcal{R}_{p,q}(i)]_{:,1} &= \sum_{k=-M_o}^{M_o} v_{p,q}(k)s(i+k)\mathbf{s}^*(i+k) \\ &= \sum_{k=-M_o}^{M_o} \tilde{v}_{p,q}(k)\mathbf{s}^*(i+k), \qquad (21)\end{aligned}$$

where $\tilde{v}_{p,q}(k) = v_{p,q}(k)s(i+k)$. We therefore can think of the column $[\mathcal{R}_{p,q}(i)]_{:,1}$ as a result of convolution of sequences $[s^*(i-M_o-L+1),\ldots,s^*(i+M_o)]$ and $\tilde{v}_{p,q}(k)$, $k=-M_o,\ldots,M_o$. The time-domain convolution can be replaced with frequency-domain multiplication based on the convolution theorem [40] and the use of FFTs. FFTs of size $(M+L)$ are used to include the length of the sequence M and the maximum delay L. One needs to compute FFTs of these two sequences, multiply them (taking one of them as complex-conjugate), and compute the inverse FFT. Therefore, the complexity of these computations is about three FFT operations of size $(L+M)$. This is instead of the direct computation, which would require about $4ML$ MACs.

Thus, depending on the filter length L and the sliding window length M, the complexity of computing the matrix $\mathcal{R}_{p,q}(i)$ at every time instant i can be either $4ML$ MACs or 3 FFTs of size $(L+M)$, whatever is smaller.

As an example, for the case of $M = 145$ and $L = 80$, direct computation of the matrix $\mathcal{R}_{p,q}(i)$ would require 3.7×10^6 MACs. Instead, direct computation of one column $[\mathcal{R}_{p,q}(i)]_{:,1}$ would require 4.6×10^4 MACs. The complexity is further reduced to around 5.3×10^3 MACs when using the FFTs as described above.

3.3.2. Vector $\mathbf{b}_p(i+M_o)$

The direct computation of the vector requires $4ML$ MAC operations. Note that (5) can be thought of as convolution of the sequences $s(i+k)$ of length $L+M$ and $\mathbf{x}_p^*(i) = \mathbf{W}\mathbf{\Phi}_p\mathbf{x}^*(i)$ of length M. This again can be done using the FFT of length $(L+M)$ with a complexity of 3 FFT operations of size $(L+M)$.

3.3.3. System of equations (11)

A direct solution of the system in (11) would require about $4(P+1)^3L^3$ MACs. However, the expansion coefficient vector at time instant i can be updated based on the estimate $\hat{\mathbf{c}}(i-1)$ found at the previous time instant:

$$\hat{\mathbf{c}}(i) = \hat{\mathbf{c}}(i-1) + \Delta\mathbf{c}(i). \qquad (22)$$

Table 2: Leading DCD algorithm

```
       Input: $P$, $L$, $H$, $M_b$, $N_u$, $\hat{\mathbf{c}}$, $\mathcal{R}$, $\mathbf{r}$        Output: $\hat{\mathbf{c}}$
Step   Initialization: $\delta = H$, $u = 0$
       for $m = 1, \ldots, M_b$
 1       $\delta = \delta/2$, $\boldsymbol{\alpha} = [\delta, -\delta, j\delta, -j\delta]$, Flag = 1
 2       While $u \leq N_u$ and Flag = 1
 3         $[n, s] = \arg\max_{t=1,\ldots,(P+1)L}\{|\Re(r_t)|, |\Im(r_t)|\}$
 4         if $s = 1$, then $r_{\text{tmp}} = \Re(r_n)$,
 5         else $r_{\text{tmp}} = \Im(r_n)$
 6         if $r_{\text{tmp}} > (\alpha/2)\mathcal{R}_{n,n}$
 7           $\hat{c}_n = \hat{c}_n + \text{sign}(r_{\text{tmp}})s\alpha$
 8           $\mathbf{r} = \mathbf{r} - \text{sign}(r_{\text{tmp}})\alpha\mathbf{R}^{(n)}$
 9           $u = u + 1$
10         else, Flag = 0
       end
```

The system of equations at Step 4 of the algorithm in Table 1 is replaced by:

$$\left[\mathcal{R}(i) + \varepsilon\mathbf{I}_{(P+1)L}\right]\Delta\mathbf{c}(i) = \mathbf{r}(i), \tag{23}$$

where $\mathbf{r}(i)$ is a residual vector computed as:

$$\mathbf{r}(i) = \mathbf{b}(i) - \mathcal{R}(i)\hat{\mathbf{c}}(i-1). \tag{24}$$

The computation of the residual vector requires about $4(P+1)^2L^2$ MACs. If direct methods were used for solving the systems of equations in (11) and (23), the complexity would be the same. However, we find an estimate of the increment $\Delta\mathbf{c}(i)$ using a few DCD iterations [27, 32]. The complex-valued DCD algorithm updates elements of the solution vector in four possible directions (-1, 1, j and $-j$) with a set of pre-defined step sizes. The solution vector is only updated in a few elements, which is good enough to achieve a fast convergence and high estimation accuracy of the adaptive algorithm in the steady-state. Here we consider the complex-valued leading DCD algorithm presented in Table 2, where H is an initial step size, M_b is the maximum number of bits used to represent the solution vector, N_u defines the maximum number of ‘successful’ DCD iterations per time instant, $\boldsymbol{\alpha}$ is the direction vector, and δ is the step size used for updating the solution vector. Instead of searching for the update for every element in the solution vector in a cyclic order, the leading DCD algorithm only updates the leading element corresponding to the maximum residual. As the initial step size is chosen as a power of two, the multiplication and division operations are replaced by bit-shift operations, which are much simpler for practical implementation and allow avoiding the numerical truncation, thus leading to numerical stable algorithms. Solving the system of equation in (23) would require at most $4N_u(P+1)L + M_b$ additions.

The SRLS-L algorithm directly solves the system of equations, and therefore its overall complexity is $3(P+1)^2/2 + 3(P+1)$ FFT operations of size $(L+M)$ and extra about $4(P+1)^3L^3$ MACs for solving the system of equations. The overall complexity of the

SRLS-L-DCD adaptive filter, which approximately solves the system of equations, is $3(P+1)^2/2 + 3(P+1)$ FFT operations of size $(L+M)$, and extra about $4(P+1)^2L^2$ MACs and $4N_u(P+1)L+M_b$ additions for DCD iterations. Although solving the system of equations using DCD iterations only requires additions, they are also counted as MAC operations for consistency in computing the algorithm complexity. The complexity of the SRLS-L and SRLS-L-DCD algorithms will be compared with the complexity of other algorithms in Section 5.

4. Homotopy SRLS-L-DCD adaptive filter

For fast-varying channels with a large delay spread, the minimum sliding window length required is significantly increased when high orders of the basis functions are used. In practice, there is sparsity in the expansion coefficients. By exploiting the sparsity, the sliding window length can be reduced, which in turn will improve the tracking performance of the SRLS-L algorithm.

In this section, we modify the SRLS-L algorithm by exploiting the sparsity in the expansion coefficients and propose a new sparse adaptive filter well suited to estimation of fast time-varying channels. Specifically, the sparse recovery problem is solved using the homotopy principle and DCD iterations. The new adaptive filter is named the HSRLS-L-DCD adaptive filter.

In the SRLS-L and SRLS-L-DCD algorithms, the least squares (LS) criterion is used (or weighted LS criterion if non-uniform weightings are used) resulting in a system of equations solved directly or using DCD iterations, respectively. In the HSRLS-L-DCD algorithm, we find a solution by minimizing the following cost function:

$$\begin{aligned} J[\mathbf{c}(i)] = &\frac{1}{2}\mathbf{c}^H(i)\mathcal{R}(i)\mathbf{c}(i) - \Re\{\mathbf{c}^H(i)\mathbf{b}(i)\} \\ &+ \tau\tilde{\mathbf{w}}^T(i)\sum_{i=1}^{(P+1)L}|\mathbf{c}(i)|, \end{aligned} \tag{25}$$

where $c(i)$ is an element of the vector $\mathbf{c}$. The first two terms represent the LS cost, the third term is a penalty function (ℓ_1 norm of the solution) that favours sparse solutions, τ is a positive regularization parameter which controls the balance between the LS fitting and the penalty and $\tilde{\mathbf{w}}$ is a weight vector which is updated in reweighting iterations [42].

The novelty of the proposed homotopy adaptive filter compared to the original homotopy adaptive filter in [37] is in the cost function (25). This cost function allows us to exploit different speeds of variation of different elements in the vector $\mathbf{h}(i)$, and consequently to exploit sparsity in the expansion coefficients $\mathbf{c}(i)$ to improve the estimation accuracy. With such a cost function, instead of directly estimating the unknown time-varying impulse response $\mathbf{h}(i)$ and exploiting the sparsity in elements of the vector $\mathbf{h}(i)$,

Table 3: Hℓ_1-DCD algorithm

Step	Input parameters: M_0, P, L, H, M_b, N_u, τ, γ, μ_d, μ_c, μ_w Output: $\hat{\mathbf{c}}$, $\tilde{\mathbf{w}}$ Initialization: $I = \emptyset$, $\mathcal{R} = \mathcal{R}(i)$, $\mathbf{c} = \mathbf{0}$, $\mathbf{b} = \mathbf{r}(i)$, $\tilde{\mathbf{w}} = \mathbf{1}_L$
	for $i > 0$, repeat:
1	$\tau = \max_k \lvert b_k \rvert$
2	Remove tth element from I ($I \leftarrow I \setminus t$), if $t = \arg\min_{k \in I} \frac{1}{2}\lvert c_k\rvert^2 \mathcal{R}_{k,k} + \Re\{c_k^* b_k\} - \tau \tilde{w}_k \lvert c_k\rvert$ and $\frac{1}{2}\lvert c_k\rvert^2 \mathcal{R}_{k,k} + \Re\{c_k^* b_k\} - \tau \tilde{w}_k \lvert c_k\rvert < 0$
3	If the tth element is removed, then update: $\mathbf{b} = \mathbf{b} + c_t \mathcal{R}^{(t)}$
4	Include tth element into the support ($I \leftarrow I \cup t$), if $t = \arg\max_{k \in I} \dfrac{(\lvert b_k\rvert - \tau \tilde{w}_k)^2}{\mathcal{R}_{k,k}}$ and $\lvert b_t\rvert > \tau \tilde{w}_t$
5	Update the regularization parameter: $\tau = \gamma\tau$
6	Approximately solve the LS-ℓ_1 optimization on the support I using the leading ℓ_1-DCD algorithm [43]
7	Debiasing according to (27)
8	Reweighting according to (28)

the proposed adaptive filter exploits sparsity in the expansion coefficients and provides local estimates of the expansion coefficients, which are then transformed into an estimate of $\mathbf{h}(i)$ using (12).

If the estimate at the previous time instant $\hat{\mathbf{c}}(i-1)$ is used as a warm-start as shown in (22), then (25) can be replaced with:

$$\begin{aligned} \Delta J[\Delta\mathbf{c}(i)] = \frac{1}{2}\Delta\mathbf{c}^H(i)\mathcal{R}(i)\Delta\mathbf{c}(i) - \Re\{\Delta\mathbf{c}^H(i)\mathbf{r}(i)\} \\ + \tau\tilde{\mathbf{w}}^T(i)\sum_{i=1}^{(P+1)L}|\mathbf{c}(i)|. \end{aligned} \tag{26}$$

The LS-ℓ_1 optimization problem can be solved by the homotopy ℓ_1-DCD (Hℓ_1-DCD) algorithm [37], which is based on homotopy with respect to the regularization parameter τ. The key idea is to solve a sequence of optimization problems with different regularization parameters τ. If τ is high, the third term of (25) dominates the cost function and forces the cardinality of the support to zero. The parameter τ is initialized to the highest possible value which guarantees that the algorithm starts with zero support. This is done to allow us to keep a low dimension of the problem and to reduce the complexity [37]. After each homotopy iteration, the regularization parameter is reduced by a positive factor $\gamma < 1$: $\tau \leftarrow \gamma\tau$.

The Hℓ_1-DCD algorithm is summarized in Table 3. In adaptive filtering, homotopy iterations are distributed in time to reduce the complexity [43]. Therefore, only one homotopy iteration is performed at each time instant. In each homotopy iteration, an element is added or removed from the support I based on the criteria given in Step 2 and Step 4 of Table 3 (see details in [37]). Then, the LS-ℓ_1 optimization problem is

Table 4: Leading ℓ_1-DCD algorithm

```
        Input: $H$, $M_b$, $N_u$, $\mu_c$, $\mathbf{c}$, $\mathcal{R}$, $\mathbf{b}$        Output: $\hat{\mathbf{c}}$
Step    Initialization: $\delta = H$, $u = 0$
1       $T_c = \mu_c \max_k |b_k|$
        for $m = 1, \ldots, M_b$
2           $\delta = \delta/2$, $\boldsymbol{\alpha} = [\delta, -\delta, j\delta, -j\delta]$, Flag $= 1$
3           While $u < N_u$ and Flag $= 1$, repeat:
4               $t = \mathrm{argmax}_{k \in I} |b_k|$
5               if $|b_t| < T_c$, break
6               for $k = 1, \ldots, 4$,
                    $\Delta J(k) = -\Re\{\alpha_k^* b_t\} + \tau \tilde{w}_t (|c_t + \alpha_k| - |c_t|)$
7               Find $J_{\min} = \min_k \Delta J(k)$ and $q = \arg\min_k \Delta J(k)$
8               If $J_{\min} < -\frac{1}{2}\delta^2 \mathcal{R}_{t,t}$, do:
9                   $c_t = c_t + \alpha_q$
10                  $\mathbf{b} = \mathbf{b} - \alpha_q \mathcal{R}^{(t)}$
11                  $u = u + 1$
12              else, Flag $= 0$
        end
```

approximately solved on the support by using the leading ℓ_1-DCD algorithm [43]. The leading ℓ_1-DCD algorithm used at Step 6 is summarized in Table 4, where the parameter μ_c defines the stopping threshold T_c. The algorithm stops when the magnitude of the maximum residual element is smaller than T_c. The leading ℓ_1-DCD algorithm operates in the same manner as the leading DCD algorithm, the only difference is in the cost function of the optimization problem.

After the leading ℓ_1-DCD algorithm terminates, the support is re-estimated using the hard thresholding [41]:

$$I = \{k : |c_k| > \mu_d \mathrm{max}_k\{|c_k|\}\}, \tag{27}$$

where μ_d is a predefined parameter between zero and one. Then, the weight vector is recursively updated as:

$$\tilde{\mathbf{w}}(i) = (1 - \mu_w)\tilde{\mathbf{w}}(i-1) + \mu_w \bar{\mathbf{w}}, \tag{28}$$

where $\mu_w \in (0, 1]$ is a parameter which defines the update rate. Elements of the weight vector $\bar{\mathbf{w}}$ are given by:

$$\bar{w}_k = \begin{cases} 0, & k \in I \\ 1, & \text{otherwise.} \end{cases}$$

5. Numerical Results

In this section, we investigate by simulation the identification performance of the SRLS-L algorithm in time-varying channels; we compare it with the LBF estimator [18] and its fast version, the fLBF estimator [19]. In subsection 5.1, we describe the simulation scenario. In subsection 5.2, the identification performance of the algorithms is investigated. In subsection 5.3, we investigate the detection performance of an FD system with different adaptive algorithms used for SI channel estimation. Finally, subsection 5.4 compares the complexity of the algorithms.

5.1. Simulation scenario

The simulation scenario is based on the channel information obtained in an FD lake experiment. Details of the lake experiment are given in Section 6. The SI channel impulse response measured in the lake experiment is shown in Fig. 3. This is obtained using the third-order HSRLS-L-DCD adaptive filter which achieves in this experiment the best SIC performance among the adaptive algorithms that we considered.

The SI to noise ratio is defined as:

$$\mathrm{SNR}_{\mathrm{SI}} = \frac{E\{|x(i)|^2\}}{\sigma_n^2}, \tag{29}$$

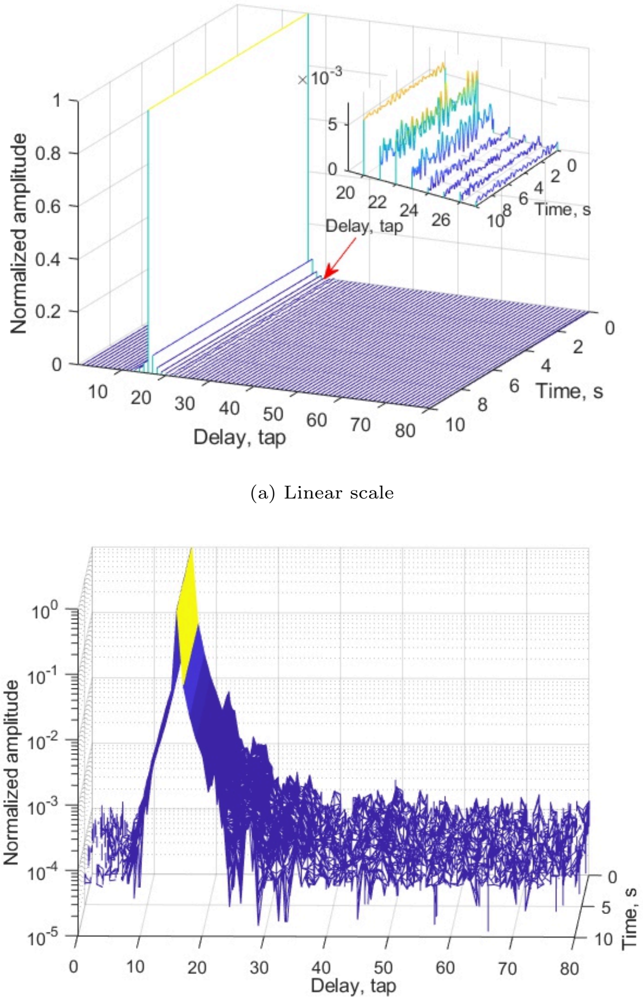
where $x(i)$ is the baseband equivalent received signal and σ_n^2 is the variance of the noise. In this scenario, the SI to noise ratio is 71 dB. The symbol rate is $f_d = 1000$ symbols per second, thus the adaptive filter taps are separated by a 1 ms interval. The filter length is set to $L = 80$.

The channel is modelled based on the power delay profile and cut-off frequencies of each multipath component estimated using the lake experimental data. The lth tap of the time-varying channel response $\mathbf{h}(i)$ is modelled as a stationary random process with a power spectral density $G_h^l(2\pi f)$, which is uniform within the frequency interval $f \in [-f_{\max}^l, f_{\max}^l]$, and independent of random processes describing the other taps. The variance of each multipath component is estimated by averaging in time the channel estimates shown in Fig. 3. The power delay profile is shown in Fig. 4. It can be seen that it is consistent with the channel estimates in Fig. 3 (b) shown in logarithmic scale. Note that the time-varying multipath components have significantly lower variances compared to static direct paths, and still their accurate estimation is very important for the overall performance of SI cancellation.

It can be seen in Fig. 3 that the SI channel contains several strong taps, which are almost time-invariant; these are the direct path and reflections from stationary parts of the experimental equipment and lake bottom. The first fast time-varying tap is due to the reflection from the time-varying lake surface. Further taps are due to more complicated reflections (bottom-surface, surface-equipment, etc.), which include the surface reflection; thus, they are also fast varying. Therefore, it is sensible to use different cut-off frequencies $f_{\max}^{(l)}$ of random processes describing time variation of different taps.

We identify the cut-off frequency of the lth tap based on the periodogram, which is computed as:

$$G_l(k) = \left| \sum_{n=0}^{N-1} \hat{h}_l(n) e^{-j2\pi kn/N} \right|^2, \quad k = 0, \ldots, N-1, \tag{30}$$



(a) Linear scale

(b) Logarithmic scale

Figure 3: Impulse response variation in the FD lake experiment.

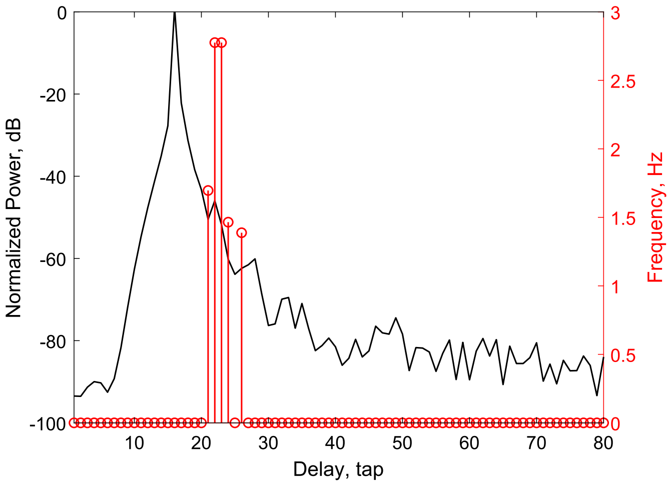


Figure 4: Power delay profile and cut-off frequencies of the multipath components in the FD lake experiment. The power of the multipaths are normalized with respect to the path with the maximum power.

where N is the number of samples of the lth tap over a time period (10 s in our experiment, see Fig. 3), and $\hat{h}_l(n)$ is the estimate of the lth tap in the experiment. For the lth tap, we find the maximum index $k_{\max}$ among k that satisfies,

$$G_l(k) > \eta \max_{k,l} G_l(k), \tag{31}$$

where η is a threshold parameter; we set $\eta = -75$ dB. Then, we have the cut-off frequency of the lth multipath as $f_{\max}^{(l)} = k_{\max}\Delta f$, where $\Delta f = f_d/N$. The cut-off frequencies $f_{\max}^{(l)}$ of all multipath components are shown in Fig. 4. It can be seen that the first few taps in the vicinity of the direct path (at the 16th tap) have $f_{\max}^{l} = 0$, which means that these paths are almost time-invariant. Several taps associated with the first surface reflection are fast-varying with different variation speeds, and the maximum variation speed is close to 3 Hz. This is consistent with time-varying channel estimates observed in Fig. 3.

5.2. MSD performance

In this subsection, we investigate the identification performance of the classical SRLS, SRLS-L and HSRLS-L-DCD adaptive filters. The performance of the LBF [18] and fLBF estimators [19] are also investigated based on the MATLAB codes provided in [44] for comparison. Note that the MATLAB codes in [44] were converted into the complex-valued versions before using them in our simulations.

The mean squared deviation (MSD) used for evaluating the identification performance

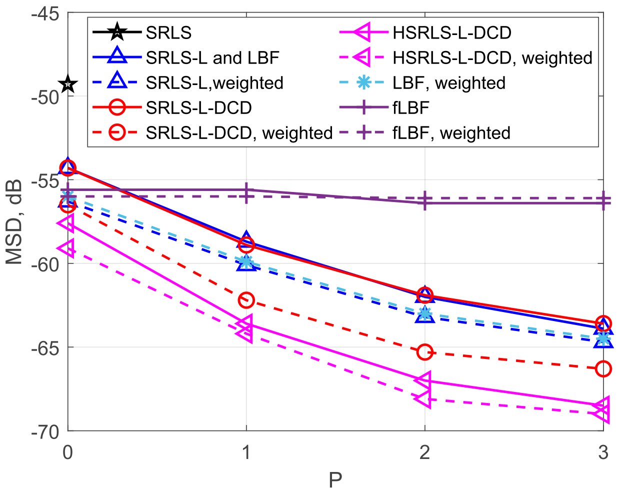


Figure 5: MSD performance of the adaptive algorithms in the simulation similar to the lake experiment.

is defined as:

$$\mathrm{MSD}(i) = \frac{\|\mathbf{h}(i) - \hat{\mathbf{h}}(i)\|_2^2}{\|\mathbf{h}(i)\|_2^2}. \tag{32}$$

The MSD performance in (32) is then averaged over an interval of 10 s after convergence.

The MSD performance of the adaptive filters from $P = 0$ to $P = 3$ are shown in Fig. 5. For each adaptive filter, we consider two cases, e.g., for the 'SRLS-L' adaptive filter, the uniform weighting is used, and for 'SRLS-L, weighted', a non-negative symmetric bell-shaped window (such as the Hanning window) is applied. The parameters of the adaptive filters are chosen to provide their best MSD performance. The optimal sliding window length M for every adaptive filter is shown in Table 5.

We first consider the adaptive algorithms without weighting (solid curves). As can be seen, an MSD performance of around -49 dB is achieved by the classical SRLS algorithm. As expected, the SRLS-L adaptive filters achieve identical performance as that of the LBF estimator. On the other hand, the fLBF estimator shows an inferior performance, which does not change much over P. It is seen that the MSD performance can be significantly improved by using the SRLS-L adaptive filters. The higher order of polynomial P is, the better the MSD performance. The third-order SRLS-L adaptive filter outperforms the classical SRLS adaptive filter by 13.5 dB. Note that the fLBF estimator is no longer considered for SIC in FD lake experiments due to its inferior performance.

It is obvious that the shorter time window M is used, the better the tracking performance can be achieved. However, for the identifiability, M should be longer than the number of estimated parameters, i.e., it should satisfy $M > (P + 1)L$. The number of parameters to be estimated is reduced if the channel is sparse. This in turn allows

Table 5: Optimal M for the adaptive filters

Adaptive filter	M
SRLS	105
SRLS-L, SRLS-L-DCD and LBF, $P = 0$	145
SRLS-L, SRLS-L-DCD and LBF, $P = 1$	185
SRLS-L, SRLS-L-DCD and LBF, $P = 2$	305
SRLS-L, SRLS-L-DCD and LBF, $P = 3$	385
SRLS-L, SRLS-L-DCD and LBF, weighted, $P = 0$	185
SRLS-L, SRLS-L-DCD and LBF, weighted, $P = 1$	205
SRLS-L, SRLS-L-DCD and LBF, weighted, $P = 2$	325
SRLS-L, SRLS-L-DCD and LBF, weighted, $P = 3$	405
fLBF, $P = 0$ and $P = 1$	165
fLBF, $P = 2$ and $P = 3$	385
fLBF, weighted, $P = 0$ and $P = 1$	205
fLBF, weighted, $P = 0$ and $P = 1$	545
HSRLS-L-DCD, $P = 0$	105
HSRLS-L-DCD, $P = 1$	105
HSRLS-L-DCD, $P = 2$	185
HSRLS-L-DCD, $P = 3$	265
HSRLS-L-DCD, weighted, $P = 0$	125
HSRLS-L-DCD, weighted, $P = 1$	125
HSRLS-L-DCD, weighted, $P = 2$	265
HSRLS-L-DCD, weighted, $P = 3$	345

Table 6: Parameters used for the HSRLS-L-DCD algorithm

	γ	μ_d	μ_w
$P = 0$	0.85	1.5×10^{-4}	0.9
$P = 1$	0.7	6×10^{-4}	0.8
$P = 2$	0.75	4×10^{-5}	0.8
$P = 3$	0.7	1×10^{-5}	0.85

reducing the window length M. This can be done using the HSRLS-L-DCD algorithm.

In Fig. 6, we show estimates of the expansion coefficients obtained using the HSRLS-L-DCD algorithm. It can be seen that the expansion coefficients for the first-order, second-order and third-order basis functions exhibit a clear sparse structure. There are several strong taps corresponding to the time-varying paths and the rest of them have magnitudes close to zero. Such a channel allows us to use a shorter estimation window as the number of non-zero elements required to be estimated is smaller.

Detailed parameters used for the HSRLS-L-DCD algorithm are given in Table 6. As shown in Table 5, the optimal M for the HSRLS-L-DCD algorithms (from $P = 1$ to $P = 3$) are significantly reduced compared to the SRLS-L algorithms. Meanwhile, the MSD performance is further improved. With $M = 265$, the third-order HSRLS-L-DCD algorithm outperforms the classical SRLS algorithm by 19.1 dB.

Then, we consider the case when the bell-shaped weightings are applied to the basis functions (dashed line curves). It is clear that the weighting improves the MSD performance for most of the adaptive algorithms, especially for non-sparse algorithms. The symmetric bell-shaped windowing helps to put emphasis on the data around the middle

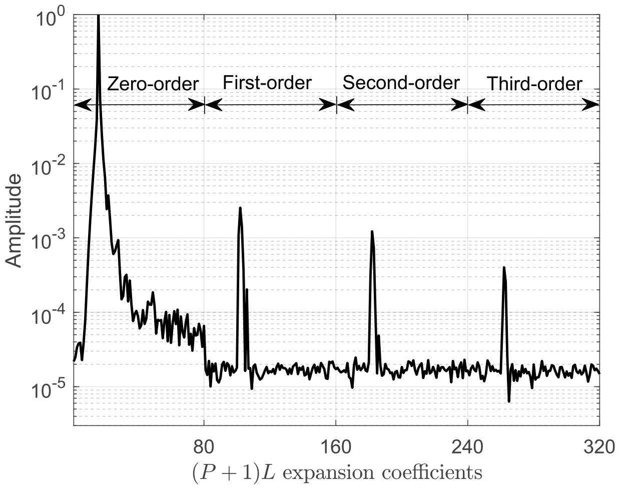


Figure 6: Estimates of expansion coefficients when using the HSRLS-L-DCD algorithm.

Table 7: Number of DCD updates N_u used for the adaptive filters with DCD iterations, $H = 1$ and $M_b = 24$.

Adaptive filter	N_u	Adaptive filter	N_u
SRLS-L-DCD, $P = 0$	12	HSRLS-L-DCD, $P = 0$	4
SRLS-L-DCD, $P = 1$	16	HSRLS-L-DCD, $P = 1$	64
SRLS-L-DCD, $P = 2$	64	HSRLS-L-DCD,$P = 2$	64
SRLS-L-DCD, $P = 3$	80	HSRLS-L-DCD, $P = 3$	64
SRLS-L-DCD, weighted, $P = 0$	4	HSRLS-L-DCD, weighted, $P = 0$	1
SRLS-L-DCD, weighted, $P = 1$	8	HSRLS-L-DCD, weighted, $P = 1$	32
SRLS-L-DCD, weighted, $P = 2$	8	HSRLS-L-DCD, weighted, $P = 2$	32
SRLS-L-DCD, weighted, $P = 3$	8	HSRLS-L-DCD, weighted, $P = 3$	32

of the estimation time window. It can be seen from Table 5 that a slightly longer sliding window length is required for the weighted SRLS-L adaptive filters, which indicates that the overall complexity of the weighted SRLS-L adaptive filters is slightly increased. However, as shown in Table 7, fewer DCD iterations are required for the weighted SRLS-L-DCD algorithm. Similar conclusions can be drawn for the HSRLS-L-DCD adaptive filters.

5.3. BER performance

In this subsection, we investigate the detection performance of the far-end transmission in the FD system. The detection performance is evaluated by computing the bit error rate (BER).

We use the same channel model as described in subsection 5.1 to simulate the time-varying near-end SI channel. The far-end channel is modelled as a single path channel. The BER performance is averaged over 500 simulation trials. In each simulation trial,

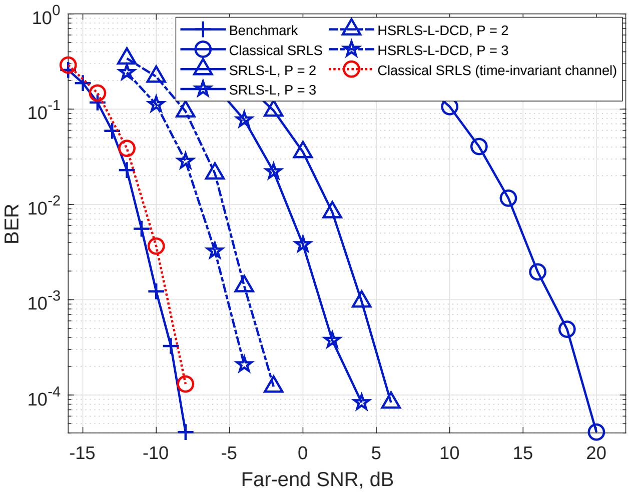


Figure 7: BER performance of the FD communication system with SI cancellation using different adaptive algorithm.

new realizations of the near-end SI channel, noise and transmitted signal are generated. The received signal $x(i)$ is generated by adding the far-end signal $z(i)$ and noise $n(i)$ to the SI channel output $r(i)$. Complex-valued Gaussian independent zero-mean random variables are used as noise samples. The BPSK direct sequence spread spectrum signals are used as the far-end signal $z(i)$. The chip rate is 1 kHz, and the spreading factor is 250. The SI to noise ratio is 71 dB (the same as that in the lake experiment). The far-end signal level is defined by the far-end SNR as: σ_z^2/σ_n^2, where σ_z^2 is the variance of the far-end signal.

Fig. 7 shows the BER performance for the classical SRLS, SRLS-L (P = 2 and P = 3) and HSRLS-L-DCD (P = 2 and P = 3) algorithms. The BER performance for the transmission without near-end SI is given as a benchmark. For the classical SRLS algorithm, we show the BER performance in two scenarios: with static and time-varying SI channels. It can be seen that the classical SRLS is capable of providing a BER performance very close to the benchmark when the SI channel is time-invariant. When the SI channel is time-invariant, the classical SRLS algorithm can reduce the SI to a level lower than the receiver's noise floor, which explains the excellent detection performance. However, the performance of the classical SRLS algorithm degrades significantly in time-varying channels. By comparing the BER curves of the classical SRLS algorithm in time-invariant and time-varying channels, we observe a far-end SNR loss of around 27 dB. This gap is reduced by using the second-order and third-order SRLS-L adaptive filters and it is further reduced when second-order and third-order HSRLS-L-DCD algorithms are

Table 8: Overall complexity of the adaptive filters per sample (analytical expressions)

Adaptive filter	MACs	÷ and $\sqrt{}$
LBF	$18(P+1)^3L^3 + 6(P+1)^2L^2 + 4L^2$	0
SRLS-L	$4(P+1)^3L^3 + [1.5(P+1)^2 + 3(P+1)](L+M)\log_2(L+M)$	0
SRLS-L-DCD	$4(P+1)^2L^2 + [1.5(P+1)^2 + 3(P+1)](L+M)\log_2(L+M)$ $+4N_u(P+1)L + M_b$	0
HSRLS-L-DCD	$4(P+1)^2L^2 + [1.5(P+1)^2 + 3(P+1)](L+M)\log_2(L+M)$ $+(15+N_u)(P+1)L + 2\lvert I\rvert(N_u + M_b)$	$3(P+1)L - \lvert I\rvert$

Table 9: Overall complexity of the adaptive filters per sample (quantitative analysis)

	MACs	÷ and $\sqrt{}$
LBF (weighted), $P = 0$	9.6×10^6	
LBF (weighted), $P = 1$	7.5×10^7	
LBF (weighted), $P = 2$	2.5×10^8	
LBF (weighted), $P = 3$	5.9×10^8	
SRLS-L and SRLS-L, weighted, $P = 0$	2.1×10^6	
SRLS-L and SRLS-L, weighted, $P = 1$	1.6×10^7	
SRLS-L and SRLS-L, weighted, $P = 2$	5.5×10^7	
SRLS-L and SRLS-L, weighted, $P = 3$	1.3×10^8	
SRLS-L-DCD, $P = 0$	3.7×10^4	
SRLS-L-DCD, $P = 1$	1.4×10^5	
SRLS-L-DCD, $P = 2$	3.7×10^5	
SRLS-L-DCD, $P = 3$	6.6×10^5	
SRLS-L-DCD, weighted, $P = 0$	3.7×10^4	
SRLS-L-DCD, weighted, $P = 1$	1.4×10^5	
SRLS-L-DCD, weighted, $P = 2$	3.2×10^5	
SRLS-L-DCD, weighted, $P = 3$	5.8×10^5	
HSRLS-L-DCD, $P = 0$	3.5×10^4	206
HSRLS-L-DCD, $P = 1$	1.4×10^5	422
HSRLS-L-DCD, $P = 2$	3.1×10^5	652
HSRLS-L-DCD, $P = 3$	5.6×10^5	865
HSRLS-L-DCD, weighted, $P = 0$	3.5×10^4	216
HSRLS-L-DCD, weighted, $P = 1$	1.4×10^5	418
HSRLS-L-DCD, weighted, $P = 2$	3.2×10^5	643
HSRLS-L-DCD, weighted, $P = 3$	5.7×10^5	851

used.

5.4. Complexity comparison

In this subsection, we compare the complexity of the SRLS-L, SRLS-L-DCD, and HSRLS-L-DCD adaptive filters at a time instant in terms of number of MACs, divisions and square-roots. We also compare them with the complexity of the LBF estimator which provides the same MSD performance as the SRLS-L adaptive filter.

The complexity analysis of the SRLS-L, SRLS-L-DCD, HSRLS-L-DCD and LBF algorithms is given in subsections 3.2, Appendix A and Appendix B. The overall complexity of all the algorithms at every sample is summarized in Table 8. We compute the complexity of the FFT of size $(L+M)$ as $(L+M)\log_2(L+M)$ MAC operations. Based on that, the overall complexities of the adaptive algorithms at every time instant are given in Table 9 for the window lengths M in Table 5 and the filter length $L = 80$.

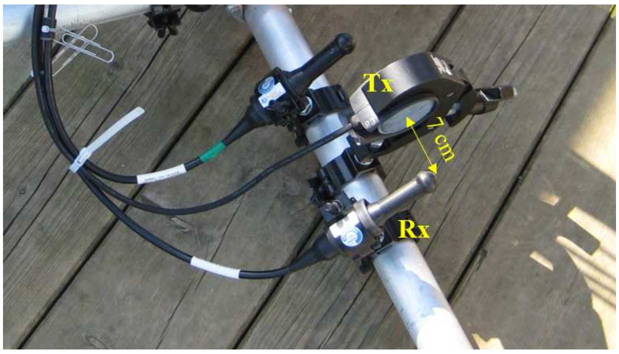


Figure 8: Transducer (Tx) and hydrophone (Rx) used in the lake experiment. Note that the other hydrophone with the green label is not used in this experiment.

Combined with the simulation results shown in Fig. 5, several conclusions can be drawn from Table 9. First, the SRLS-L adaptive filters implemented using the proposed techniques achieve the same performance, but require only about one-fifth of the overall complexity compared to that of the LBF estimators. Further, the complexity of the SRLS-L adaptive filters is significantly reduced by using the DCD iterations, especially for higher-order SRLS-L-DCD algorithms; the overall complexity can be reduced by about hundreds of times. The complexity of the HSRLS-L-DCD algorithm is almost the same as that of the SRLS-L-DCD algorithm, however, the MSD performance is improved by an extra 2 to 3 dB when using the HSRLS-L-DCD algorithm.

Although by applying the Hanning window allows a reduction in the number of DCD iterations, at the same time, the required sliding window length M is increased, thus the overall complexity is about the same in all cases.

6. Experimental results in lake experiments

In this section, the adaptive filters are used for the SI channel estimation with the scheme in Fig. 1 in the FD lake experiments. In the lake experiment, the true channel response is unknown, and therefore the MSD performance cannot be measured. Instead we measure the SIC factor which is the ratio of signal to interference ratios for the far-end signal before and after the SI cancellation. A detailed description of the measurement procedure can be found in [25].

6.1. Experimental setup and transmitted signals

The lake experiment is conducted in the Kelk Lake in East Yorkshire (UK) on the 15th of May, 2019. As shown in Fig. 8, the transducer and hydrophone are fixed on an metallic pipe. The distance between the transducer (Tx) and hydrophone (Rx) is 7 cm.

Table 10: SIC factor (in dB) in the lake experiment and in simulation (Sim), $f_c = 32$ kHz, $f_d = 1000$ symbols per second; $L = 80$.

Adaptive filter	M	SIC (Lake)	SIC (Sim)
SRLS	105	51.2	48.4
SRLS-L, $P = 0$	185	55.5	52
SRLS-L, $P = 1$	225	57.7	56
SRLS-L, $P = 2$	325	58.9	59.5
SRLS-L, $P = 3$	385	59.7	61.9
SRLS-L, weighted, $P = 0$	225	56.7	53.5
SRLS-L, weighted, $P = 1$	245	58.8	57.8
SRLS-L, weighted, $P = 2$	385	60.1	61
SRLS-L, weighted, $P = 3$	445	60.7	62.7
HSRLS-L-DCD, $P = 0$	105	57.3	54.7
HSRLS-L-DCD, $P = 1$	105	60.9	60.5
HSRLS-L-DCD, $P = 2$	165	62.3	63.3
HSRLS-L-DCD, $P = 3$	225	63.4	65.5
HSRLS-L-DCD, weighted, $P = 0$	165	56.9	56
HSRLS-L-DCD, weighted, $P = 1$	165	60.5	60.9
HSRLS-L-DCD, weighted, $P = 2$	285	62.6	64.5
HSRLS-L-DCD, weighted, $P = 3$	285	63.6	65.9

During the experiment, the pipe is placed horizontally at a depth of 4 m. The maximum depth of the experimental site is around 8 m. Based on our observation, during the experiment, the amplitude of the lake surface waves varies from 5 cm to 10 cm.

In the experiment, BPSK signals are transmitted; an RRC filter with a roll-off factor of 0.2 is used for the pulse shaping. The transmitted signal length is 20 s, which includes 5 s of zero-padding at the start of the transmission to measure the background noise level and 15 s of the BPSK signal. The BPSK signals are transmitted at one of the three carrier frequencies $f_c = 12, 32$ or 80 kHz with 1.2 kHz bandwidth. For the transmitted signal with 80 kHz carrier frequency, a wider frequency bandwidth of 4.8 kHz is also considered. Note that the purpose of using transmitted signals at different carrier frequencies is to investigate if consistent cancellation performance can be achieved regardless of the carrier frequency. For the 80 kHz signal, we consider a wider frequency bandwidth to investigate if the estimation performance can be improved by exploiting the higher delay resolution of the channel. For all the experimental data, we use the same FD system design as shown in Fig. 1.

6.2. Experimental vs simulation results

We first consider the FD experiment at the carrier frequency 32 kHz. The symbol rate is $f_d = 1000$ symbols per second. In this experiment, the SI to noise ratio is around 71 dB. The SIC factor is computed over a 10 s interval after the convergence of the adaptive filter. Table 10 shows the SIC factor and the optimal value of M providing the best cancellation for each of the adaptive algorithms. With $P = 3$, the SRLS-L algorithm improves the SIC factor by 8.5 dB compared to the classical SRLS algorithm. On top of that, the third-order HSRLS-L-DCD algorithm improves the SIC factor by an extra 3.7 dB, thus outperforming the classical SRLS algorithm by 12.2 dB. The SIC performance is slightly improved (around 1 dB) when weightings are applied for the SRLS-L algorithms. For the HSRLS-L-DCD algorithms, no noticeable difference is observed.

In Section 5, we considered a simulation scenario which is based on the SI channel estimates obtained using the HSRLS-L-DCD algorithm in this experiment. To ensure a fair comparison, we use the same parameters for the adaptive filters in both the simulation and the lake experiment. We can see that the SIC factors in the simulation match very well to that in the experiment. This result further demonstrates that the channel modelling process we used in subsection 5.1 works well.

Another conclusion from Table 10 and Fig. 7 is that the SIC factor provides the same indication of the system performance as the BER. As can be seen in Table 10, the SIC factor is 65.5 dB when the third-order HSRLS-L-DCD algorithm is used, which means the residual SI level is 5.5 dB above the receiver's noise floor. It can be seen in Fig. 7 that the gap in BER performance between the benchmark curve and that of the third-order HSRLS-L-DCD algorithm is also about 5 dB for a BER level of 10^{-3}. Similar conclusions can be drawn for the rest of the algorithms. This further validates the use of the SIC factor for the performance evaluation instead of analyzing the detection performance of specific communication systems.

6.3. Experiments with different carrier frequencies and bandwidths

We now consider three experimental data sets with different carrier frequencies and bandwidths. This is done to investigate how consistent is the improvement in the SIC performance achieved with the proposed adaptive filters in practical scenarios. The experimental results are shown in Tables 11 and 12.

In the first experiment, the BPSK signal is transmitted at the carrier frequency $f_c =$ 12 kHz at the symbol rate $f_d = 1000$ symbols per second. The SI to noise ratio is around 65 dB. In this experiment, the channel delay spread is longer compared to the $f_c = 32$ kHz channel. Therefore, we use a filter length of $L = 100$. The parameters of every algorithm are chosen to provide the best SIC performance. It can be seen in Table 11 that the SIC factor can be improved by using higher order basis functions. With $P = 3$, the SRLS-L algorithm outperforms the classical SRLS algorithm by 8.7 dB. Additional 3.9 dB of SIC is achieved with the third-order HSRLS-L-DCD algorithm by exploiting the sparsity in the expansion coefficients. Again, the SIC performance is slightly improved by applying weightings when using the SRLS-L adaptive algorithm. However, the improvement in the performance of the HSRLS-L-DCD algorithm, especially for the higher-order cases, is almost negligible (within 0.5 dB). Therefore, we only apply weighting to the SRLS-L adaptive filter for the rest of the data.

Table 11: SIC factor (in dB) in the lake experiment,
$f_c = 12$ kHz, $f_d = 1000$ symbols per second and $L = 100$.

Adaptive filter	M	SIC factor
SRLS	145	43.4
SRLS-L, $P = 0$	165	46.7
SRLS-L, $P = 1$	225	49.6
SRLS-L, $P = 2$	385	51.4
SRLS-L, $P = 3$	525	52.1
SRLS-L, weighted, $P = 0$	285	48.6
SRLS-L, weighted, $P = 1$	285	50.8
SRLS-L, weighted, $P = 2$	465	52.7
SRLS-L, weighted, $P = 3$	545	53.1
HSRLS-L-DCD, $P = 0$	165	49
HSRLS-L-DCD, $P = 1$	165	52.7
HSRLS-L-DCD, $P = 2$	265	55.3
HSRLS-L-DCD, $P = 3$	285	56
HSRLS-L-DCD, weighted, $P = 0$	225	50.1
HSRLS-L-DCD, weighted, $P = 1$	225	52.7
HSRLS-L-DCD, weighted, $P = 2$	325	55.7
HSRLS-L-DCD, weighted, $P = 3$	345	56.3

In Tables 12, we show the SIC performance of the 80 kHz signal with 1 kHz and 4 kHz frequency bandwidth. For the 4 kHz signal, the symbol rate is $f_d = 4000$ symbols per second. The SI to noise ratio in both experiments is around 65 dB. Due to the high attenuation at higher frequencies, the SI channel delay spread is reduced. The adaptive filter length is $L = 60$ for the signal with 1 kHz bandwidth, and $L = 240$ for the 4 kHz bandwidth.

The SIC performance in both cases is very close. The SIC factor is slightly better with the 4 kHz bandwidth. This could be attributed to the clearer multipath structure due to the higher delay resolution. Another observation is that the best estimation time window length for the HSRLS-L-DCD algorithms is significantly smaller compared to the SRLS-L algorithm. With $P = 3$, the estimation time window of the HSRLS-L-DCD algorithm is about twice shorter compared to that of the SRLS-L algorithm. This explains the improvement in the SIC performance.

7. Conclusions

In this paper, low complexity interpolating adaptive filters which combine the SRLS adaptive filter and BEM approach (specifically, the Legendre polynomials) are proposed for identification of time-varying systems (channels). Techniques are proposed to reduce the complexity of the adaptive filters using the FFT and DCD algorithms. This paper focuses on a specific application, SIC in FD UWA systems. To exploit the sparsity in the expansion coefficients, a novel sparse adaptive filter, the HSRLS-L-DCD algorithm is proposed. The MSD performance of the proposed adaptive filters is investigated and compared with that of the LBF estimator in a simulation which mimics an FD lake

Table 12: SIC factor (in dB) in the lake experiment, $f_c = 80$ kHz.

Adaptive filter	$f_d = 1000$ symbols per second $L = 60$		$f_d = 4000$ symbols per second $L = 240$	
	M	SIC factor	M	SIC factor
SRLS	85	47.1	505	47.4
SRLS-L, $P = 0$	135	49.9	465	49.7
SRLS-L, $P = 1$	165	49.6	685	50.6
SRLS-L, $P = 2$	265	50.6	965	50.4
SRLS-L, $P = 3$	345	50.4	1325	50.8
SRLS-L, weighted, $P = 0$	185	50.8	685	50.8
SRLS-L, weighted, $P = 1$	205	51.3	805	51.6
SRLS-L, weighted, $P = 2$	325	51.3	1205	51.4
SRLS-L, weighted, $P = 3$	345	51.4	1385	51.6
HSRLS-L-DCD, $P = 0$	65	52.4	265	52.7
HSRLS-L-DCD, $P = 1$	65	54	265	55.7
HSRLS-L-DCD, $P = 2$	105	55.8	425	57
HSRLS-L-DCD, $P = 3$	115	56.5	445	57.4

experiment. Complexity analysis is presented for all the algorithms used in the simulation. Lake experiments are conducted to evaluate the SIC performance of the proposed adaptive filters within an FD UWA system. Results show that the proposed adaptive filtering algorithms significantly improve the SIC performance compared to classical RLS algorithm. Moreover, the proposed algorithms are of comparatively low complexity, and due to the use of DCD iterations, are well suited to hardware (such as FPGA) implementation, which makes them good candidates for practical implementation.

Appendix A: Complexity analysis of the HSRLS-L-DCD algorithm

The HSRLS-L-DCD algorithm solves the system of equations using the Hℓ_1-DCD algorithm. The complexity of the main steps of the Hℓ_1-DCD algorithm is summarized as follows:

- The residual vector $\mathbf{r}(i)$ in (24) can be computed with $4(P+1)^2L^2$ MAC operations;
- The regularization parameter τ (at Step 1 in Hℓ_1-DCD algorithm) can be computed with $2(P+1)L$ MAC operations;
- Removing an element from the support I and updating the residual vector involve $4(P+1)L+7|I|$ multiplications and $2(P+1)L+4|I|$ additions and $|I|$ square-root operations;
- Adding an element into the support I takes $4((P+1)L-|I|)$ multiplications, $2((P+1)L-|I|)$ additions, $(P+1)L-|I|$ divisions and $(P+1)L-|I|$ square-root operations;
- Solving the optimization problem on the support I using the leading ℓ_1-DCD algorithm:
    - Computing the threshold T_c requires $2(P+1)L$ MACs;

  - Finding the maximum element of the residual vector requires $2|I|$ MAC operations, this could be repeated for $N_u + M_b$ times;
  - Computing the increment of the cost function ($\Delta J(k)$) and finding the direction of update which minimizes the cost function can be done with 6 multiplications and 15 additions, this is repeated for at most $N_u + M_b$ times;
  - Updating the residual vector requires $2(P+1)L$ MAC operations, this can be done at most N_u times;
- Obtaining the final support according to (27) requires $2(P+1)L$ MAC operations and $(P+1)L$ square-root operations;
- Reweighting according to (28) involves $(P+1)L$ MACs.

In total, the Hℓ_1-DCD algorithm minimizes the cost function with $4(P+1)^2L^2 + (15 + 2N_u)(P+1)L + 2|I|(N_u + M_b)$ MAC operations, $(P+1)L - |I|$ divisions and $2(P+1)L$ square-root operations.

Thus, the overall complexity of the HSRLS-L-DCD algorithm is $4(P+1)^2L^2 + (15 + 2N_u)(P+1)L + 2|I|(N_u + M_b)$ MAC operations, $3(P+1)^2/2 + 3(P+1)$ FFT operations of size $(L + M)$, $(P+1)L - |I|$ divisions and $2(P+1)L$ square-root operations.

Appendix B: Complexity analysis of the LBF estimator

Recursive computation methods are proposed in [18] to reduce the complexity of the LBF estimator (see [18] and MATLAB in [44]). The main steps of the LBF estimator are summarized as follows:

- $\mathbf{A}(i) = \boldsymbol{\xi}(i)\boldsymbol{\xi}^H(i)$, where $\boldsymbol{\xi}(i)$ is an $L \times 1$ complex-valued vector. This involves $4L^2$ real-valued MACs.
- $\mathbf{r}_m(i) \leftarrow \beta\boldsymbol{\Gamma}\mathbf{r}_m(i)$, where β is a scalar, $\mathbf{r}_m(i)$ is a $(P+1)L \times 1$ complex-valued vector and $\boldsymbol{\Gamma}$ is a real-valued $(P+1)L \times (P+1)L$ matrix. This would require $2(P+1)^2L^2$ MACs.
- $\mathbf{R}_m(i) \leftarrow \mathbf{R}_m(i) - v_1\mathbf{A}(i) \otimes \mathbf{B}$, where $\mathbf{R}_m(i)$ is a $(P+1)L \times (P+1)L$ complex-valued matrix, $\mathbf{A}(i)$ is an $L \times L$ complex-valued matrix and $\mathbf{B}$ is a $(P+1) \times (P+1)$ real-valued matrix. This requires $2(P+1)^2L^2$ MACs.
- $\mathbf{R}_m(i) \leftarrow \beta(\boldsymbol{\Gamma}\mathbf{R}_m(i)\boldsymbol{\Gamma}^T)$, which requires $2(P+1)^3L^3$ MACs.
- $\mathbf{R}_m(i) \leftarrow \mathbf{R}_m(i) + v_M\mathbf{A}(i) \otimes \mathbf{B}$, which requires $2(P+1)^2L^2$ MACs.
- Solve the system of equations: $\mathbf{R}_m(i)\mathbf{a}(i) = \mathbf{r}_m(i)$. This would require about $4(P+1)^3L^3$ MACs.

The overall complexity of the main steps of the LBF estimator is about $6(P+1)^3L^3 + 6(P+1)^2L^2 + 4L^2$ MACs.

References

[1] J. I. Choi, M. Jain, K. Srinivasan, P. Levis, S. Katti, Achieving single channel, full duplex wireless communication, in: Annual International Conference on Mobile Computing and Networking, 2010, pp. 1–12.

[2] M. Duarte, A. Sabharwal, Full-duplex wireless communications using off-the-shelf radios: Feasibility and first results, in: IEEE Asilomar Conference on Signals, Systems and Computers, 2010, pp. 1558–1562.

[3] A. Sabharwal, P. Schniter, D. Guo, D. W. Bliss, S. Rangarajan, R. Wichman, In-band full-duplex wireless: Challenges and opportunities, IEEE Journal on Selected Areas in Communications 32 (9) (2014) 1637–1652.

[4] G. Qiao, S. Gan, S. Liu, L. Ma, Z. Sun, Digital self-interference cancellation for asynchronous in-band full-duplex underwater acoustic communication, Sensors 18 (6) (2018) 1700–1716.

[5] M. Stojanovic, J. A. Catipovic, J. G. Proakis, Phase-coherent digital communications for underwater acoustic channels, IEEE Journal of Oceanic Engineering 19 (1) (1994) 100–111.

[6] H. Yeo, B. Sharif, O. Hinton, A. Adams, Improved RLS algorithm for time-variant underwater acoustic communications, Electronics Letters 36 (2) (2000) 191–192.

[7] Y. Zhang, Y. V. Zakharov, J. Li, Soft-decision-driven sparse channel estimation and turbo equalization for MIMO underwater acoustic communications, IEEE Access 6 (2018) 4955–4973.

[8] L. Shen, B. Henson, Y. Zakharov, P. Mitchell, Digital self-interference cancellation for underwater acoustic systems, IEEE Transactions on Circuits and Systems II: Express Briefs 67 (1) (2020) 192–196.

[9] M. Siderius, M. B. Porter, Modeling broadband ocean acoustic transmissions with time-varying sea surfaces, The Journal of the Acoustical Society of America 124 (1) (2008) 137–150.

[10] M. Niedzwiecki, Functional series modeling approach to identification of nonstationary stochastic systems, IEEE Transactions on Automatic Control 33 (10) (1988) 955–961.

[11] G. W. Wornell, A Karhunen-Loeve-like expansion for 1/f processes via wavelets, IEEE Transactions on Information Theory 36 (4) (1990) 859–861.

[12] M. Visintin, Karhunen-Loeve expansion of a fast Rayleigh fading process, Electronics Letters 32 (18) (1996) 1712–1713.

[13] M. K. Tsatsanis, G. B. Giannakis, Modelling and equalization of rapidly fading channels, Int. J. Adaptive Contr. Signal Processing 10 (2-3) (1996) 159–176.

[14] Z. Tang, G. Leus, Time-multiplexed training for time-selective channels, IEEE Signal Processing Letters 14 (9) (2007) 585–588.

[15] G. Leus, On the estimation of rapidly time-varying channels, Proc. EUSIPCO'04, Vienna, Austria (2004) 2227–2230.

[16] H. Mai, Y. V. Zakharov, A. G. Burr, Iterative channel estimation based on B-splines for fast flat fading channels, IEEE Transactions on Wireless Communications 6 (4) (2007) 1224–1229.

[17] T. Zemen, C. F. Mecklenbrauker, Time-variant channel estimation using discrete prolate spheroidal sequences, IEEE Transactions on Signal Processing 53 (9) (2005) 3597–3607.

[18] M. Niedźwiecki, M. Ciołek, Generalized Savitzky–Golay filters for identification of nonstationary systems, Automatica 108 (2019) 108477.

[19] M. J. Niedźwiecki, M. Ciołek, A. Gańcza, A new look at the statistical identification of nonstationary systems, Automatica 118 (2020) 109037.

[20] Y. V. Zakharov, T. C. Tozer, Local spline approximation of time-varying channel model, Electronics Letters 37 (23) (2001) 1408–1409.

[21] D. K. Borah, B. D. Hart, Frequency-selective fading channel estimation with a polynomial time-varying channel model, IEEE Transactions on Communications 47 (6) (1999) 862–873.

[22] C. Shin, J. G. Andrews, E. J. Powers, An efficient design of doubly selective channel estimation for OFDM systems, IEEE Transactions on Wireless Communications 6 (10) (2007) 3790–3802.

[23] T. Hrycak, S. Das, G. Matz, Inverse methods for reconstruction of channel taps in OFDM systems, IEEE Transactions on Signal Processing 60 (5) (2012) 2666–2671.

[24] M. Niedzwiecki, T. Klaput, Fast recursive basis function estimators for identification of time-varying processes, IEEE Transactions on Signal Processing 50 (8) (2002) 1925–1934.

[25] L. Shen, B. Henson, Y. Zakharov, P. Mitchell, Adaptive filtering for full-duplex UWA systems with time-varying self-interference channel, IEEE Access 8 (2020) 187590–187604.

[26] Y. V. Zakharov, T. Tozer, Multiplication-free iterative algorithm for LS problem, Electronics Letters 40 (9) (2004) 567–569.

[27] Y. V. Zakharov, G. P. White, J. Liu, Low-complexity RLS algorithms using dichotomous coordinate descent iterations, IEEE Transactions on Signal Processing 56 (7) (2008) 3150–3161.

[28] Y. Ogawa, K. Nishikawa, A kernel adaptive filter based on ERLS-DCD algorithm, in: International Technical Conference on Circuits/Systems, Computers and Communications, no. P4-13, 2011, pp. 1228–1231.

[29] Y. Yu, L. Lu, Z. Zheng, W. Wang, Y. Zakharov, R. C. de Lamare, DCD-based recursive adaptive algorithms robust against impulsive noise, IEEE Transactions on Circuits and Systems II: Express Briefs.

[30] F. Yang, M. Wu, P. Ji, J. Yang, Low-complexity implementation of the improved multiband-structured subband adaptive filter algorithm, IEEE Transactions on Signal Processing 63 (19) (2015) 5133–5148.

[31] G. Kim, H. Lee, J. Chung, J. Lee, A delay relaxed RLS-DCD algorithm for real-time implementation, IEEE Transactions on Circuits and Systems II: Express Briefs 65 (1) (2017) 61–65.

[32] J. Liu, Y. V. Zakharov, B. Weaver, Architecture and FPGA design of dichotomous coordinate descent algorithms, IEEE Transactions on Circuits and Systems I: Regular Papers 56 (11) (2009) 2425–2438.

[33] M. Stojanovic, J. Preisig, Underwater acoustic communication channels: Propagation models and statistical characterization, IEEE Communications Magazine 47 (1) (2009) 84–89.

[34] Y. Huang, L. Wan, S. Zhou, Z. Wang, J. Huang, Comparison of sparse recovery algorithms for channel estimation in underwater acoustic ofdm with data-driven sparsity learning, Physical Communication 13 (2014) 156–167.

[35] Y. Meng, A. P. Brown, R. A. Iltis, T. Sherwood, H. Lee, R. Kastner, Mp core: algorithm and design techniques for efficient channel estimation in wireless applications, in: Proceedings of IEEE Design Automation Conference, IEEE, 2005, pp. 297–302.

[36] P. Maechler, P. Greisen, N. Felber, A. Burg, Matching pursuit: Evaluation and implementatio for lte channel estimation, in: Proceedings of IEEE International Symposium on Circuits and Systems, IEEE, 2010, pp. 589–592.

[37] Y. Zakharov, V. H. Nascimento, Homotopy algorithm using dichotomous coordinate descent iterations for sparse recovery, in: IEEE Asilomar Conference on Signals, Systems and Computers, 2012, pp. 820–824.

[38] L. Shen, B. Henson, Y. Zakharov, P. D. Mitchell, Robust digital self-interference cancellation for full-duplex UWA systems: Lake experiments, in: Underwater Acoustics Conference and Exhibition, 2019, pp. 243–250.

[39] I. S. Gradshteyn, I. M. Ryzhik, Table of integrals, series, and products, Academic press, 2014.

[40] S. Soliman, M. Srinath, Continuous and Discrete Signals and Systems, Prentice Hall, 1998.

[41] Y. V. Zakharov, V. H. Nascimento, R. C. De Lamare, F. G. D. A. Neto, Low-complexity DCD-based sparse recovery algorithms, IEEE Access 5 (2017) 12737–12750.

[42] Y. Wang, W. Yin, Sparse signal reconstruction via iterative support detection, SIAM Journal on Imaging Sciences 3 (3) (2010) 462–491.

[43] Y. V. Zakharov, V. H. Nascimento, Homotopy RLS-DCD adaptive filter, in: International Symposium on Wireless Communication Systems, 2013, pp. 1–5.

[44] Supplement: supplementary material (including MATLAB codes), `https://eti.pg.edu.pl/katedra-systemow-automatyki/Automatica`, [Accessed: 2- Dec- 2020].