DCD-Based Recursive Adaptive Algorithms Robust Against Impulsive Noise

Yi Yu, Lu Lu, *Member, IEEE*, Zongsheng Zheng, *Student Member, IEEE*, Wenyuan Wang, Yuriy Zakharov, *Senior Member, IEEE*, and Rodrigo C. de Lamare, *Senior Member, IEEE*

***Abstract*—The dichotomous coordinate descent (DCD) algorithm has been successfully used for significant reduction in the complexity of recursive least squares (RLS) algorithms. In this work, we generalize the application of the DCD algorithm to RLS adaptive filtering in impulsive noise scenarios and derive a unified update formula. By employing different robust strategies against impulsive noise, we develop novel computationally efficient DCD-based robust recursive algorithms. Furthermore, to equip the proposed algorithms with the ability to track abrupt changes in unknown systems, a simple variable forgetting factor mechanism is also developed. Simulation results for channel identification scenarios in impulsive noise demonstrate the effectiveness of the proposed algorithms.**

***Index Terms*—Dichotomous coordinate descent, impulsive noise, recursive least squares, variable forgetting factor**

I. Introduction

ADAPTIVE filtering has been a prominent technique in a variety of applications such as system identification, active noise control, and echo cancellation (EC) [1]. The least mean square (LMS) and recursive least squares (RLS) algorithms represent two typical families of adaptive algorithms. The complexity of LMS is $\mathcal{O}(M)$ arithmetic operations per sample (ops), where M is the filter length, but its convergence is slow especially when the input signal is highly correlated. RLS improves the convergence at the cost of a high complexity of $\mathcal{O}(M^2)$ ops. To reduce the complexity, some fast RLS algorithms were proposed as summarized in [1, Chapter 14]. However, these fast algorithms are numerically unstable in finite precision implementation since they are based on the matrix inversion.

Alternatively, the dichotomous coordinate descent (DCD) iterations for solving the normal equations in the RLS algorithms were proposed [2]. They result in not only numerically stable adaptive algorithms but also in performance comparable to that of the original RLS algorithm. An important property of the DCD algorithm is that it only requires addition and shift operations, which are simpler for implementation than multiplication and division, and thus it is well suited to real-time implementation. Moreover, the DCD-RLS algorithm reduces the complexity to $\mathcal{O}(M)$ ops for input signals with the tapped-delay structure. The DCD algorithm was also applied for the complexity reduction in the affine projection algorithm [3], sparse signal recovery [4], and distributed estimation [5].

Regrettably, the LMS and RLS algorithms undergo performance deterioration in impulsive noise [6], owing to the squared-error based minimization criteria. Realizations of impulsive noise process are sparse and random with amplitude far higher than the Gaussian noise, and therefore, best modeled by heavy-tailed distributions, e.g., the α-stable distribution. Such noise scenarios are common in such as echo cancellation, underwater acoustics, audio processing, and communications [7], [8]. For adapting impulsive noise scenarios, existing literature have reported various robust approaches. For instance, the recursive least M-estimate (RLM) algorithm [9] exploits the Hampel's M-estimate function to suppress impulsive interferences. Based on the l_p-norm of errors, the recursive least p-norm (RLpN) algorithm was developed [10]. By gathering all the p-norms from $p = 1$ to 2 of the error, the continuous mixed p-norms (CMPN) algorithm was derived [11]; however, it has slow convergence for correlated inputs due to the gradient descent (GD) principle. Taking advantage of the Geman-McClure (GMC) estimator, a recursive algorithm [12] for Volterra system identification was derived, which shows a better performance than RLpN and RLM algorithms in impulsive noise modeled by the α-stable distribution [7]. When impulsive noise appears, by incorporating the step-size scaler into the update term, a robust subband algorithm was developed [13]. The correntropy measures the similarity between two variables, which is helpful for suppressing large outliers; thus, the maximum correntropy criterion (MCC) has been used for improving the anti-jamming capability of adaptive filters to impulsive noise, yielding the GD-based MCC [14]–[16] and recursive MCC (RMCC) algorithms [17], [18]. However, these robust recursive algorithms have also high complexity of $\mathcal{O}(M^2)$ ops. In particular, the complexity of the fixed-point variant of MCC algorithm in [17] is $\mathcal{O}(M^3)$ due to the direct inverse of an $M \times M$ matrix.

This work focuses on a class of low-complexity robust algorithms against impulsive noise by resorting to the DCD approach. Concretely, a generalized DCD-based robust recur-

This work was supported by National Science Foundation of P.R. China (Nos. 61901400 and 61901285) and Doctoral Research Fund of Southwest University of Science and Technology in China (No. 19zx7122). The work of Y. Zakharov was partly supported by the U.K. EPSRC Grants EP/P017975/1 and EP/R003297/1.

Y. Yu is with School of Information Engineering, Robot Technology Used for Special Environment Key Laboratory of Sichuan Province, Southwest University of Science and Technology, Mianyang, 621010, China (e-mail: yuyi_xyuan@163.com).

L. Lu is with School of Electronics and Information Engineering, Sichuan University, Chengdu, China (e-mail: lulu19900303@126.com).

Z. Zheng and W. Wang are with School of Electrical Engineering, Southwest Jiaotong University, Chengdu, 610031, China (e-mail: wenyuan-wang@my.swjtu.edu.cn, bk20095185@my.swjtu.edu.cn).

Y. Zakharov is with the Department of Electronics, University of York, York YO10 5DD, U.K. (e-mail: yury.zakharov@york.ac.uk).

R. C. de Lamare is with CETUC, PUC-Rio, Rio de Janeiro 22451-900, Brazil. (e-mail: delamare@cetuc.puc-rio.br).

sion is derived. By applying different robust strategies to this recursion, we develop DCD-based robust algorithms, such as the DCD-RMCC, DCD-RLM, and DCD-RLpN algorithms. We also design a variable forgetting factor (VFF) scheme for improving the tracking capability of the algorithms.

II. DCD-Based Robust Algorithms

A. Unified Formulation

Suppose that at time instant n, the desired signal d_n and an $M \times 1$ input signal vector $\boldsymbol{x}_n$ are available and obey the relation $d_n = \boldsymbol{x}_n^{\mathrm{T}} \boldsymbol{w}^o + v_n$, where the $M \times 1$ vector $\boldsymbol{w}^o$ needs to be estimated, and $(\cdot)^{\mathrm{T}}$ denotes the transpose. The additive noise with impulsive behavior, v_n, here is described by the α-stable process[1], also called the α-stable noise. A (symmetric) α-stable random variable is usually characterized by the characteristic function [7]

$$\phi(t) = \exp(-\gamma |t|^{\alpha}). \tag{1}$$

The characteristic exponent $\alpha \in (0, 2]$ describes the impulsiveness of the noise (smaller α leads to more outliers) and $\gamma > 0$ represents the dispersion degree of the noise. Note that when α = 1 and 2, it reduces to the Cauchy and Gaussian distributions, respectively.

To effectively estimate $\boldsymbol{w}^o$ in such noise scenarios, we define a unified robust exponentially weighted least squares problem:

$$\boldsymbol{w}_n = \arg\min_{\boldsymbol{w}} \left\{ \sum_{i=0}^{n} \lambda^{n-i} \varphi\left(d_i - \boldsymbol{x}_i^{\mathrm{T}} \boldsymbol{w}\right) + \delta_n \|\boldsymbol{w}\|_2^2 \right\}, \tag{2}$$

where $0 \ll \lambda < 1$ is the forgetting factor, $\delta_n > 0$ is a regularization parameter, and $\varphi(\cdot)$ is a function that specifies the robustness against impulsive noise.

By setting the derivative of (2) with respect to $\boldsymbol{w}$ to zero, we arrive at the normal equations:

$$\boldsymbol{R}_n \boldsymbol{w}_n = \boldsymbol{z}_n, \tag{3}$$

where

$$\begin{aligned} \boldsymbol{R}_n &= \sum_{i=0}^{n} \lambda^{n-i} f_i \boldsymbol{x}_i \boldsymbol{x}_i^{\mathrm{T}} + \delta_n \boldsymbol{I}_M \\ &= \lambda \boldsymbol{R}_{n-1} + f_n \boldsymbol{x}_n \boldsymbol{x}_n^{\mathrm{T}} + (\delta_n - \lambda \delta_{n-1}) \boldsymbol{I}_M \end{aligned} \tag{4}$$

is the time-averaged autocorrelation matrix of $\boldsymbol{x}_n$,

$$\begin{aligned} \boldsymbol{z}_n &= \sum_{i=0}^{n} \lambda^{n-i} f_i d_i \boldsymbol{x}_i \\ &= \lambda \boldsymbol{z}_{n-1} + f_n d_n \boldsymbol{x}_n \end{aligned} \tag{5}$$

is the time-averaged crosscorrelation vector of d_n and $\boldsymbol{x}_n$, and $\boldsymbol{I}_M$ is an $M \times M$ identity matrix. Also, $f_n = \varphi'(\epsilon_n)/\epsilon_n$, where $\epsilon_n = d_n - \boldsymbol{x}_n^{\mathrm{T}} \boldsymbol{w}_n$ is the *a posteriori* error and $\varphi'(\epsilon_n)$ is the derivative of $\varphi(\epsilon_n)$.

At time instant $n-1$, let $\hat{\boldsymbol{w}}_{n-1}$ denote the approximate solution of (3) for estimating $\boldsymbol{w}^o$, and the corresponding residual vector is $\boldsymbol{r}_{n-1} = \boldsymbol{z}_{n-1} - \boldsymbol{R}_{n-1} \hat{\boldsymbol{w}}_{n-1}$. By defining $\Delta \boldsymbol{w}_n = \boldsymbol{w}_n - \hat{\boldsymbol{w}}_{n-1}$, from (3) we obtain an auxiliary system of equations:

$$\boldsymbol{R}_n \Delta \boldsymbol{w}_n = \boldsymbol{z}_n - \boldsymbol{R}_n \hat{\boldsymbol{w}}_{n-1} \triangleq \boldsymbol{b}_n. \tag{6}$$

Applying the recursive expressions (4) and (5), $\boldsymbol{b}_n$ can be rewritten as

$$\boldsymbol{b}_n = \lambda \boldsymbol{r}_{n-1} + f_n e_n \boldsymbol{x}_n - (\delta_n - \lambda \delta_{n-1}) \hat{\boldsymbol{w}}_{n-1}, \tag{7}$$

where $e_n = d_n - \boldsymbol{x}_n^{\mathrm{T}} \hat{\boldsymbol{w}}_{n-1}$ denotes the *a priori* error.

By using the DCD algorithm to solve the problem in (6), we arrive at an approximate solution of the original normal equations (3):

$$\hat{\boldsymbol{w}}_n = \hat{\boldsymbol{w}}_{n-1} + \Delta \hat{\boldsymbol{w}}_n. \tag{8}$$

Although (7) shows that $\boldsymbol{b}_n$ requires the residual error vector of the original system (3), after some algebra we notice that it is equivalent to the residual error vector for the auxiliary system (6), i.e., $\boldsymbol{r}_n = \boldsymbol{z}_n - \boldsymbol{R}_n \hat{\boldsymbol{w}}_n = \boldsymbol{b}_n - \boldsymbol{R}_n \Delta \hat{\boldsymbol{w}}_n$. At time index n, f_n in (7) is not yet available, but by resorting to the *a priori* error, we may approximate f_n as

$$f_n \approx \varphi'(e_n)/e_n. \tag{9}$$

This completes the derivation of DCD-based robust recursion, summarized in Table I.

TABLE I
DCD-Based Robust Recursive Update

Parameters: $0 \ll \lambda < 1,\ \delta_0 > 0$
Initialization: $\boldsymbol{R}_0 = \delta_0 \boldsymbol{I}_M,\ \hat{\boldsymbol{w}}_0 = \boldsymbol{0},\ \boldsymbol{r}_0 = \boldsymbol{0}$
for $n = 1, \ldots$
$e_n = d_n - \boldsymbol{x}_n^{\mathrm{T}} \hat{\boldsymbol{w}}_{n-1}$
$\boldsymbol{R}_n = \lambda \boldsymbol{R}_{n-1} + f_n \boldsymbol{x}_n \boldsymbol{x}_n^T + (\delta_n - \lambda \delta_{n-1}) \boldsymbol{I}_M$
$\boldsymbol{b}_n = \lambda \boldsymbol{r}_{n-1} + f_n e_n \boldsymbol{x}_n - (\delta_n - \lambda \delta_{n-1}) \hat{\boldsymbol{w}}_{n-1}$
Using DCD iterations to solve $\boldsymbol{R}_n \Delta \boldsymbol{w}_n = \boldsymbol{b}_n$, which yields $\Delta \hat{\boldsymbol{w}}_n$ and $\boldsymbol{r}_n$
$\hat{\boldsymbol{w}}_n = \hat{\boldsymbol{w}}_{n-1} + \Delta \hat{\boldsymbol{w}}_n$
end

Table II presents the leading DCD algorithm for solving the system of equations $\boldsymbol{R}_n \Delta \boldsymbol{w}_n = \boldsymbol{b}_n$ (readers can refer to [2], [3] for details), where $[\boldsymbol{r}_n]_l$ is the l-th entry of $\boldsymbol{r}_n$, and $[\boldsymbol{R}_n]_{l,l}$ and $[\boldsymbol{R}_n]_{:,l}$ are the (l, l)-th entry and the l-th column of $\boldsymbol{R}_n$, respectively. Herein, $[-H, H]$ denotes the amplitude range for elements of the solution vector $\Delta \hat{\boldsymbol{w}}_n$. It is often chosen as a power-of-two number so that all multiplications by μ can be implemented by bit-shifts. M_b is the number of bits for a fixed-point representation of $\hat{\boldsymbol{w}}_n$ within the range $[-H, H]$. N_u stands for a maximum number of elements in $\Delta \hat{\boldsymbol{w}}_n$ that are updated. The solution $\Delta \hat{\boldsymbol{w}}_n$ approaches the optimal one (i.e., $\Delta \hat{\boldsymbol{w}}_n = \boldsymbol{R}_n^{-1} \boldsymbol{b}_n$) as N_u increases. As seen in Table II, the implementation of DCD only requires shift and addition operations, excluding multiplication and division operations.

B. Robust Strategies

Applying a particular robust strategy to define $\varphi(e)$ in (2), we can compute f_n by (9) to arrive at a DCD-based robust algorithm. Table III gives examples of $\varphi(e)$ for the DCD-RMCC, DCD-RLM, and DCD-RLpN algorithms derived from the widely studied MCC, M-estimate, and l_p-norm strategies, respectively. We note the following about the proposed algorithms:

[1] Other models describing the noise with impulses include the contaminated-Gaussian (CG) model [5] and the Gaussian mixture model (GMM) [19].

TABLE II
LEADING DCD ALGORITHM

```
Parameters: H, N_u, M_b,
Initialization: Δŵ_n = 0, r_n = b_n, y = 1, μ = H/2
for j = 1, ..., N_u
  l = arg max_{j=1,...,M} {|[r_n]_j|}
  while |[r_n]_l| ≤ (μ/2)[R_n]_{l,l} and y ≤ M_b
    y = y + 1, μ = μ/2
  end
  if y > M_b
    break
  else
    [Δŵ_n]_l ← [Δŵ_n]_l + μ sign([r_n]_l)
    r_n ← r_n − μ sign([r_n]_l)[R_n]_{:,l}
  end
end
```

TABLE III
SOME ROBUST DCD-BASED ALGORITHMS

Robust Algorithms	$\varphi(e)$ in (2)	$f(e)=\varphi'(e)/e$ in (9)
DCD-RMCC	$\frac{1}{\sqrt{2\pi}\beta}\left[1-\exp\left(-\frac{e^2}{2\beta^2}\right)\right]$	$\exp\left(-\frac{e^2}{2\beta^2}\right)$
DCD-RLM	$\begin{cases} e^2/2, \text{ if } \lvert e\rvert \le \xi \\ \xi^2/2, \lvert e\rvert > \xi, \end{cases}$	$\begin{cases} 1, \text{ if } \lvert e\rvert \le \xi \\ 0, \lvert e\rvert > \xi, \end{cases}$
DCD-RLpN	$\lvert e\rvert^p$	$\lvert e\rvert^p/(\lvert e\rvert^2+\varepsilon)$

1) For the DCD-RMCC algorithm, $\beta > 0$ denotes the kernel width. When $\beta \to \infty$, f_n approaches 1 so that the DCD-RMCC algorithm reduces to the DCD-RLS algorithm. When $\beta \to 0$, f_n becomes 0, and the DCD-RMCC update is frozen. Thus, β balances the robustness and dynamic performance of the algorithm in impulsive noise.

2) The DCD-RLM algorithm uses the modified Huber M-estimate function [20] for $\varphi(e)$[2]. When $|e_n| < \xi$, thus f_n equals 1 so that the DCD-RLM algorithm becomes the DCD-RLS algorithm. Otherwise, f_n becomes 0 to stop the update (ideally, this only happens when the impulsive noise appears). To effectively suppress the impulsive noise, the threshold ξ is adaptively adjusted by $\xi = \tau\hat{\sigma}_{e,n}$,

$$\hat{\sigma}^2_{e,n} = \zeta\hat{\sigma}^2_{e,n-1} + c_\sigma(1-\zeta)\text{med}(\boldsymbol{a}^e_n), \tag{10}$$

where $0 < \zeta < 1$ is a weighting factor (except $\zeta = 0$ at the algorithm start), med$(\cdot)$ is the median operator which helps to remove outliers in the data window $\boldsymbol{a}^e_n = [e^2_n, e^2_{n-1}, ..., e^2_{n-N_w+1}]$, and $c_\sigma = 1.483(1 + 5/(N_w-1))$ is the correction factor [9]. It is worth noting that, the window length N_w should be properly chosen. Larger N_w makes a more robust estimate $\hat{\sigma}^2_{e,n}$ from (10) but requires a higher complexity. A typical value of τ is 2.576. If e_n is assumed to be Gaussian (which is reasonable except when being polluted by impulsive noise), this value means the 99% confidence to prevent e_n from contributing to the update for $|e_n| \ge \xi$ [9].

3) The convergence of the RLpN algorithm in the α-stable noise requires $0 < p < \alpha$. If $p = 2$, the DCD-RLpN algorithm will also become the DCD-RLS algorithm. When $p = 1$, this corresponds to the recursion sign algorithm [22] with good robustness against impulsive noise.

Remark 1: In a nutshell, when impulsive noise happens, its negative influence on the updates of $\boldsymbol{R}_n$ and $\boldsymbol{b}_n$ will be lowered significantly due to by multiplying a tiny scaler f_n into the updates. Then, we can generalize the DCD recursion to find $\Delta\hat{w}_n$ from the system of equations $\boldsymbol{R}_n\Delta\boldsymbol{w}_n = \boldsymbol{b}_n$ with impulse-free. Hence, according to (8), the proposed DCD-based algorithms can work well in impulsive noise.

[2] Other M-estimate functions may also be used, e.g., the Huber [21] and Hampel [9] functions.

TABLE IV
COMPLEXITY OF ALGORITHMS PER INPUT SAMPLE

Algorithms	Additions	Multiplications	Divisions
LMS	$2M$	$2M+1$	0
(R) RLS	$3M^2+M$	$4M^2+4M+1$	1
(R) DCD RLS (general input)	M^2+2M+P_a	$\frac{3}{2}M^2+\frac{7}{2}M+1$	0
(R) DCD RLS (tap-delayed input)	$3M+P_a$	$5M+2$	0

C. Computational Complexity

The direct solution of (3) is $\boldsymbol{w}_n = \boldsymbol{R}_n^{-1}\boldsymbol{z}_n$. The regularization δ_n is to maintain the numerical stability of this solution [1]. However, this leads to the complexity of $\mathcal{O}(M^3)$ due to the matrix inversion $\boldsymbol{R}_n^{-1}$. Generally, δ_n is chosen as $\delta_n = \lambda^{n+1}\delta_0$ (e.g., in this paper), it makes (4) become $\boldsymbol{R}_n = \lambda\boldsymbol{R}_{n-1} + f_n\boldsymbol{x}_n\boldsymbol{x}_n^{\mathrm{T}}$. Then, using the matrix inversion lemma, $\boldsymbol{R}_n^{-1}$ can be calculated in a recursive way so that the complexity of the resulting algorithm is $\mathcal{O}(M^2)$, while it is still high for large M.

Table IV mainly compares the complexity of robust (R) RLS-type with that of proposed (R) DCD variant in terms of ops, where we drop the calculation of f_n dependent on a specific robust strategy. As in [2], the DCD recursion requires $P_a = 2N_uM + M_b$ additions at most for finding $\Delta\hat{\boldsymbol{w}}_n$. Thus, it is clear to see from Table IV, for general input vector form, the DCD version reduces the complexity by at least a factor of 0.5 in contrast with the original algorithm, in terms of multiplications and additions. On the other hand, if the input vector $\boldsymbol{x}_n$ has a tapped-delay structure, i.e., $\boldsymbol{x}_n = [x_n, x_{n-1}, ..., x_{n-M+1}]^{\mathrm{T}}$, where x_n is a data sample at time n, the calculation of $\boldsymbol{R}_n$ will be simplified. Specifically, assuming $f_n \approx f_{n-1}$, we can obtain the lower-right $(M-1)\times(M-1)$ block of $\boldsymbol{R}_n$ by copying the upper-left $(M-1)\times(M-1)$ block of $\boldsymbol{R}_{n-1}$. Then, considering the symmetry of $\boldsymbol{R}_n$, we only need the calculation of its first column:

$$[\boldsymbol{R}_n]_{:,1} \approx \lambda[\boldsymbol{R}_{n-1}]_{:,1} + f_nx_n\boldsymbol{x}_n. \tag{11}$$

Equation (11) is exact when $f_n = 1$ [2]. As claimed in Section II. B, f_n is normally close to 1, and becomes very small to suppress the update only when the impulsive noise happens. As such, using (11) is also suitable for computing $\boldsymbol{R}_n$ in the proposed DCD recursion. In this scenario, the complexity is reduced to the same order of magnitude as that of LMS. This reduction is considerable especially for a long $\boldsymbol{w}^o$ such as in EC applications.

D. Improving Tracking Performance

For the proposed algorithms, there is also a trade-off between steady-state error and tracking capability for abrupt changes of $\boldsymbol{w}^o$, because of using the fixed forgetting factor λ.

To address this problem, one may utilize the adaptive combination (AC) of two independently running DCD-based filters. Like the AC-RLpN algorithm in [10], it combines RLpN filters with the large forgetting factor for low steady-state error and with the small one for good tracking capability. However, it requires at least double complexity of the original algorithm. Alternatively, the VFF has been also an effective mechanism for improving the original RLS algorithm [23]–[25]. Consequently, to equip the proposed DCD-based algorithms, we also propose a simple VFF scheme:

$$\lambda_n = \lambda_{\min} + (1 - \lambda_{\min}) \exp(-\rho e_{n,f}^2), \tag{12}$$

where $\rho > 0$ is a design parameter, $e_{n,f}^2$ is the impulse-free squared error which can be estimated by (10). As $n \to \infty$, $e_{n,f}^2$ converges to a small value, and according to (12), λ_n approaches 1, thus reducing the steady-state error. When $\boldsymbol{w}^o$ has a sudden change, $e_{n,f}^2$ becomes large due the mismatch estimation at that time, and λ_n will approach a small forgetting factor $\lambda_{\min}$, thus speeding up the convergence.

III. Simulation Results

In this section, simulations are conducted for identifying the network echo channel response $\boldsymbol{w}^o$ of length M using an adaptive filter. The echo channels in Fig. 1 are from the ITU-T G.168 standard, with $M = 128$ taps [26]. For the tapped-delay input vector $\boldsymbol{x}_n$, its element x_n is given by the first-order autoregressive model $x_n = \varrho x_{n-1} + \vartheta_n$, where ϑ_n is a zero-mean white Gaussian random process with unit variance. Both $\varrho = 0$ (which is used only in Fig. 2(b)) and $\varrho = 0.9$ correspond to the white and correlated inputs, respectively, with the eigenvalue spreads of 1 and 346. The α-stable noise is set to $\alpha = 1.4$ and $\gamma = 1/20$. We use the normalized mean square deviation, $\text{NMSD}(n) = 10 \log_{10}(||\boldsymbol{w}_n - \boldsymbol{w}^o||_2^2 / ||\boldsymbol{w}^o||_2^2)$, as a performance measure. All simulated curves are the average over 100 independent runs.

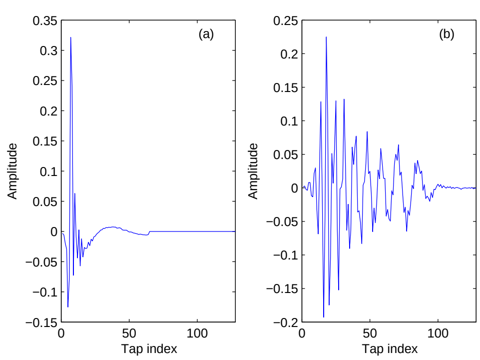

Fig. 1. Network echo channels: (a) sparse channel, (b) disperse channel.

Fig. 2 shows the NMSD performance of the DCD-RLS, GD-based MCC[3], RMCC, and proposed DCD-RMCC algorithms. As expected for impulsive noise scenarios, the performance of the original DCD-RLS algorithm is poor, while the MCC-based algorithms are performing very well. The DCD-RMCC performance approaches that of the original RMCC algorithm as N_u increases. In particular, $N_u = 8 \ll M$ (at most eight entries of $\boldsymbol{w}_n$ are updated per time n) has been enough for the DCD-RMCC performance to approach closely the RMCC performance regardless of whether $\boldsymbol{w}^o$ is sparse or not. However, as seen from Table IV, the DCD-RMCC with $N_u = 8$ reduces significantly the complexity of the RMCC. Although the DCD-RMCC requires 2.5 times multiplications of the GD-based MCC, the former (even if with $N_u = 1$) has much faster convergence than the latter. Likewise, the convergence of the proposed low-cost DCD-RLM and DCD-RLpN versions also approximate well that of the RLM and RLpN algorithms, respectively; these results are omitted for brevity.

[3]The update equation is $\boldsymbol{w}_n = \boldsymbol{w}_{n-1} + \mu f_n e_n \boldsymbol{x}_n$ [16].

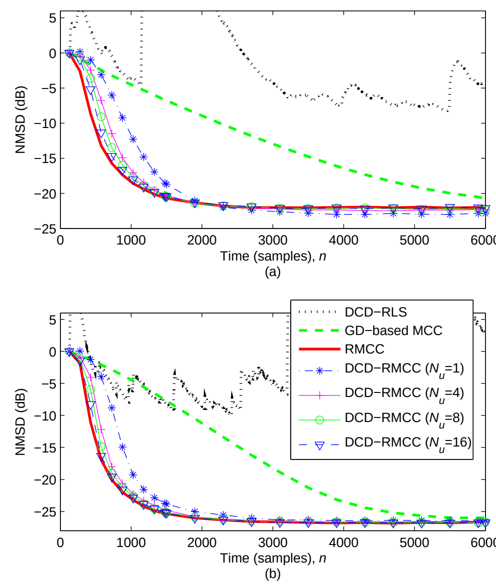

Fig. 2. NMSD curves of the DCD-RLS and MCC-based algorithms: (a) sparse channel and correlated input; (b) disperse channel and white input. Parameters of algorithms are chosen as: $\lambda = 0.998$ (all the algorithms); $\mu = 0.001$, $\beta^2 = 0.6$ (GD-based MCC); $\beta^2 = 0.03$ (RMCC); $H = 1$, $M_b = 16$ (DCD).

Fig. 3 shows the NMSD of the proposed DCD-RMCC, DCD-RLM and DCD-RLpN algorithms, with $N_u = 8$. The proposed algorithms show robustness in α-stable noise and can arrive at similar performance by properly setting their parameters. This reason is they generally behave like the DCD-RLS and use a tiny f_n to suppress the algorithms' adaptation once the impulsive noise appears. In addition, we also show the DCD-CMPN algorithm by applying the CMPN criteria in [11], i.e., $\varphi(e) = \int_1^2 |e|^p dp$ and $f(e) = \left((2|e|-1)\ln(|e|) - |e| + 1\right) / \left(|e| \ln^2(|e|)\right)$. For the l_p-norm based algorithms, p should be slightly less than α in α-stable noise; thus, the DCD-RLpN may outperform the DCD-CMPN, since the latter inherits the behavior of $p > \alpha$.

Fig. 4 demonstrates the tracking capability of the proposed algorithms, in a scenario where the echo channel changes at time $n = 8001$ by shifting its impulse response by 12 samples. As one can see, using the proposed VFF instead of the fixed

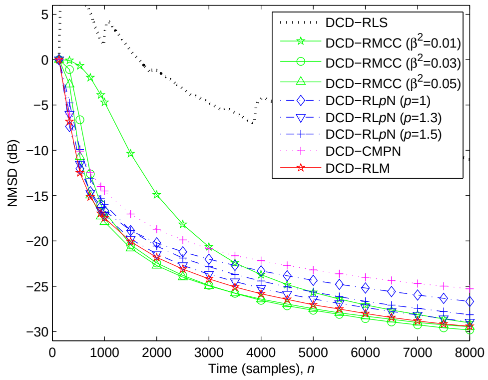

Fig. 3. NMSD curves of DCD-based RLS algorithms for the sparse channel. Parameters setting of algorithms is: $\lambda = 0.9998$ (all the recursive algorithms); $\zeta = 0.99,\ N_w = 9$ (DCD-RLM). [$N_u = 8$].

one, the DCD-based algorithms can reduce the steady-state error and improve the tracking capability.

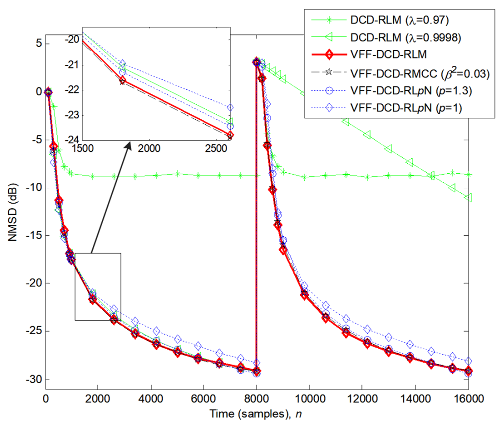

Fig. 4. NMSD curves of DCD-based RLS algorithms for the sparse channel. Parameters of VFF are: $\zeta = 0.99,\ N_w = 9,\ \rho = 3,\ \lambda_{\min} = 0.97$.

IV. Conclusion

We have proposed a general low-complexity recursion for developing RLS-type adaptive filtering algorithms operating in impulsive noise scenarios. This is based on using DCD iterations. As examples of the MCC, M-estimator, and p-norm strategies applied to this recursion, we have developed the DCD-RMCC, DCD-RLM, and DCD-RLpN algorithms, respectively. These algorithms show a performance similar to that of their high-complexity counterparts, RMCC, RLM, and RLpN algorithms, respectively. To improve the tracking capability of the algorithms, a simple time-varying forgetting factor mechanism has also been developed. Simulation results demonstrate the performance of the proposed algorithms.

References

[1] A. H. Sayed, *Fundamentals of adaptive filtering*. John Wiley & Sons, 2003.

[2] Y. V. Zakharov, G. P. White, and J. Liu, “Low-complexity RLS algorithms using dichotomous coordinate descent iterations,” *IEEE Transactions on Signal Processing*, vol. 56, no. 7, pp. 3150–3161, 2008.

[3] Y. V. Zakharov, “Low-complexity implementation of the affine projection algorithm,” *IEEE Signal Processing Letters*, vol. 15, pp. 557–560, 2008.

[4] Y. V. Zakharov, V. H. Nascimento, R. C. De Lamare, and F. G. D. A. Neto, “Low-complexity DCD-based sparse recovery algorithms,” *IEEE Access*, vol. 5, pp. 12 737–12 750, 2017.

[5] Y. Yu, H. Zhao, R. C. de Lamare, Y. V. Zakharov, and L. Lu, “Robust distributed diffusion recursive least squares algorithms with side information for adaptive networks,” *IEEE Transactions on Signal Processing*, vol. 67, no. 6, pp. 1566–1581, 2019.

[6] S. Zhang and J. Zhang, “Enhancing the tracking capability of recursive least p-norm algorithm via adaptive gain factor,” *Digital Signal Processing*, vol. 30, pp. 67 – 73, 2014.

[7] C. L. Nikias and M. Shao, *Signal processing with alpha-stable distributions and applications*. Wiley-Interscience, 1995.

[8] M. Zimmermann and K. Dostert, “Analysis and modeling of impulsive noise in broad-band powerline communications,” *IEEE transactions on Electromagnetic compatibility*, vol. 44, no. 1, pp. 249–258, 2002.

[9] Y. Zou, S. Chan, and T. Ng, “Robust M-estimate adaptive filtering,” *IEE Proceedings-Vision, Image and Signal Processing*, vol. 148, no. 4, pp. 289–294, 2001.

[10] Á. Navia-Vazquez and J. Arenas-Garcia, “Combination of recursive least p-norm algorithms for robust adaptive filtering in alpha-stable noise,” *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1478–1482, 2012.

[11] H. Zayyani, “Continuous mixed p-norm adaptive algorithm for system identification,” *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1108–1110, 2014.

[12] L. Lu, W. Wang, X. Yang, W. Wu, and G. Zhu, “Recursive Geman-mcclure estimator for implementing second-order Volterra filter,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 66, no. 7, pp. 1272–1276, 2019.

[13] J. Hur, I. Song, and P. Park, “A variable step-size normalized subband adaptive filter with a step-size scaler against impulsive measurement noise,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 64, no. 7, pp. 842–846, 2016.

[14] B. Chen, L. Xing, H. Zhao, N. Zheng, and J. C. Príncipe, “Generalized correntropy for robust adaptive filtering,” *IEEE Transactions on Signal Processing*, vol. 64, no. 13, pp. 3376–3387, 2016.

[15] L. Shi, H. Zhao, and Y. Zakharov, “An improved variable kernel width for maximum correntropy criterion algorithm,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2018.

[16] B. Chen, L. Xing, J. Liang, N. Zheng, and J. C. Principe, “Steady-state mean-square error analysis for adaptive filtering under the maximum correntropy criterion,” *IEEE signal processing letters*, vol. 21, no. 7, pp. 880–884, 2014.

[17] B. Chen, J. Wang, H. Zhao, N. Zheng, and J. C. Príncipe, “Convergence of a fixed-point algorithm under maximum correntropy criterion,” *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1723–1727, 2015.

[18] H. Radmanesh and M. Hajiabadi, “Recursive maximum correntropy learning algorithm with adaptive kernel size,” *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 65, no. 7, pp. 958–962, 2018.

[19] S. A. Kassam, *Signal detection in non-Gaussian noise*. Springer Science & Business Media, 2012.

[20] S.-C. Chan and Y.-X. Zou, “A recursive least M-estimate algorithm for robust adaptive filtering in impulsive noise: fast algorithm and convergence performance analysis,” *IEEE Transactions on Signal Processing*, vol. 52, no. 4, pp. 975–991, 2004.

[21] P. Petrus, “Robust Huber adaptive filter,” *IEEE Transactions on Signal Processing*, vol. 47, no. 4, pp. 1129–1133, 1999.

[22] V. Mathews and S. Cho, “Improved convergence analysis of stochastic gradient adaptive filters using the sign algorithm,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 35, no. 4, pp. 450–454, 1987.

[23] D.-J. Park, B.-E. Jun, and J.-H. Kim, “Fast tracking RLS algorithm using novel variable forgetting factor with unity zone,” *Electronics Letters*, vol. 27, no. 23, pp. 2150–2151, 1991.

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

[25] Y. Cai, R. C. de Lamare, M. Zhao, and J. Zhong, “Low-complexity variable forgetting factor mechanism for blind adaptive constrained constant modulus algorithms,” *IEEE Transactions on Signal Processing*, vol. 60, no. 8, pp. 3988–4002, 2012.

[26] *Digital Network Echo Cancellers Recommendation*, Std. ITU-TG.168 (V8), 2015.