



Low Complexity Shalvi-Weinstein Algorithm

Yongjun Sun¹ · Liaoping Zhang¹ · Cuiyuan Jia¹ · Zujun Liu¹

Accepted: 9 June 2021 / Published online: 25 August 2021

© The Author(s), under exclusive licence to Springer Science+Business Media, LLC, part of Springer Nature 2021

Abstract

Blind equalization can effectively reduce the intersymbol interference introduced by the frequency selective channel in the absence of the training sequence. The Shalvi-Weinstein Algorithm (SWA) performs better under most channels, especially for highly distorted ones compared with constant modulus algorithm (CMA) or its modified versions. The disadvantage of the SWA is the high complexity resulting from the computation of the inverse matrix. A low complexity SWA based on dichotomous coordinate descent algorithm is proposed in the paper, whose computation complexity is on the same order of magnitude as the CMA. Besides the low complexity, the proposed algorithm also avoids the possible numerical error resulting from the computation of the matrix inversion. Moreover, a low complexity of decision directed algorithm based on RLS is presented for a dual mode blind equalization. Simulations verify the effectiveness of the algorithm.

Keywords Blind equalization · Low complexity · SWA · RLS · Decision-directed

1 Introduction

With the increasing data rate of wireless communication, the intersymbol interference (ISI) caused by multipath propagation becomes more serious. Adaptive equalizers are widely used to remove the ISI. When there is no training sequence the blind equalization algorithms are generally chosen [1, 2]. The typical blind equalization algorithms include constant modulus algorithm (CMA) [3], Shalvi-Weinstein algorithm (SWA) [4, 5] and the corresponding modified versions [6, 7]. In [4], an improved SWA, which avoids divergence, is given. CMA has slow convergence rate but low computation complexity. On the contrary, SWA converges faster at the expense of higher computation complexity because of the computation of the matrix inversion involved. Generally, the computation of the inverse matrix may also lead to numerical calculation error.

In practical application, the computational complexity is one of the most important indexes to evaluate an algorithm. It is worth studying an algorithm with low complexity and a guaranteed performance. For solving the problem of the norm equation, the dichotomous coordinate descent (DCD) algorithm [8, 9] does not need the computation of the

✉ Yongjun Sun
yjsun@mail.xidian.edu.cn

¹ Xidian University, Xi'an, China

inverse matrix. Compared with LS algorithm for solving the same problem, the complexity of the DCD algorithm is changed from $O(N^2)$ to $O(N)$. By quantization of some parameters, the DCD algorithm has only addition and bit shift operation for hardware implementation [10, 11].

Although the SWA has relatively good equalization performance, it has high computational complexity because it involves matrix inversion. Based on the DCD and the algorithm presented in [4], a low complexity SWA is presented. The total computation complexity of the proposed algorithm is comparable to that of the CMA.

For blind equalization, when output symbols have relatively low error probability, switching to a decision-directed (DD) mode can further improve the equalization performance [12, 13]. Therefore, a low complexity dual-mode algorithm with RLS used at the second stage is also presented.

The paper is organized as follows. In Sect. 2 the system model and the classic algorithms are introduced. The proposed algorithm is presented in Sect. 3. In Sect. 4 simulations are given to verify the effectiveness of the proposed algorithm. Conclusion is given in Sect. 5.

2 Algorithm Description

The received signal $x(n)$ in multipath channel is given by

$$R(\theta) = -\frac{1}{A} \log_2 [E\{(1 + \gamma)^{-A}\}] = -\frac{1}{A} \log_2 \left[\int_0^\infty (1 + \gamma)^{-A} f_\gamma(\gamma) d\gamma \right] \tag{1}$$

where $s(n)$ is transmitted symbols through the channel h , and \otimes represents the convolution operation. $v(n)$ is the white Gaussian noise. At time n the input vector of the equalizer is $\mathbf{x}(n) = [x(n), x(n - 1), \dots, x(n - N + 1)]^H$, where N is the length of the equalizer weights, the superscript H denotes the complex conjugate transpose. The output of the equalizer is

$$y(n) = \mathbf{w}(n)^H \mathbf{x}(n) \tag{2}$$

where $\mathbf{w}(n)$ is the weight vector of the equalizer.

2.1 Introduction of SWA [4]

The cost function of the SWA is

$$J(n) = \sum_{l=0}^n \lambda^{n-l} \left(|y_{n,l}|^2 - \beta \right)^2 \tag{3}$$

where $\beta = E[|s(n)|^4] / E[|s(n)|^2]$ is a statistical constant. $0 \leq \lambda < 1$ is the forgetting factor, $y_{n,l} = \mathbf{w}(n)^H \mathbf{x}(l)$.

According to [4], the procedure of SWA is as follows.

$$\mathbf{w}(n) = \mathbf{w}(n - 1) + \mu e_{SWA}^*(n) \mathbf{R}^{-1}(n) \mathbf{x}(n) \tag{4}$$

$$e_{SWA}(n) = \begin{cases} (|y(n)|^2 - \beta)y(n), & |y(n)|^2 \geq \beta C_{1,1}^s \\ -y(n), & \text{otherwise} \end{cases} \tag{5}$$

$$\mathbf{R}(n) = \sum_{l=1}^n \lambda^{n-l} \mathbf{x}(l)\mathbf{x}^H(l) \tag{6}$$

$$\mu = 1/(\beta - \alpha C_{1,1}^s) \tag{7}$$

where $e_{SWA}(n)$ and μ denote the error and step size of the equalizer respectively, and $*$ stands for complex conjugate. α is a constant, $\mathbf{R}(n)$ is the autocorrelation matrix. $C_{1,1}^s = E\{|s(n)|^2\}$.

2.2 DCD Algorithm [8–10]

For solving the normal equation $\mathbf{Q}\mathbf{x} = \mathbf{p}$, it is equivalent to minimize the following quadratic equation.

$$f(\mathbf{x}) = \frac{1}{2} \mathbf{x}^T \mathbf{Q}\mathbf{x} - \mathbf{x}^T \mathbf{p} \tag{8}$$

where the superscript T denotes the transpose operation.

Standard linear iterative search uses the formula

$$f(\mathbf{x} + \zeta \mathbf{d}) = f(\mathbf{x}) - \frac{1}{2} \frac{(\mathbf{d}^T \mathbf{r})^2}{\mathbf{d}^T \mathbf{Q}\mathbf{d}} \tag{9}$$

where ζ is the searching step size, which is equal to $\mathbf{d}^T \mathbf{r} / \mathbf{d}^T \mathbf{Q}\mathbf{d}$, \mathbf{r} is the residual vector, \mathbf{d} is the search direction vector, which is non-orthogonal to \mathbf{r} , i.e. $\mathbf{d}^T \mathbf{r} \neq 0$.

If the search direction is a Euclidean space coordinate, the formula (9) is the coordinate descent algorithm, and $\mathbf{Q}\mathbf{d} = \mathbf{Q}^{(p)}$, where $\mathbf{Q}^{(p)}$ denotes the p th column of the matrix \mathbf{Q} , thereby removing the multiplication of the matrix and the vector.

$$f(\mathbf{x} + \zeta \mathbf{e}_p) = f(\mathbf{x}) - \frac{1}{2} \frac{\mathbf{r}_p^2}{\mathbf{Q}_{p,p}} \tag{10}$$

where \mathbf{e}_p is the searching direction in Euclidean space and the step size is $\zeta = \mathbf{r}_p / \mathbf{Q}_{p,p}$.

According to the principle of the descent method, for the k th iteration the incremental of the quadratic formula should meet the following condition.

$$\Delta f(\mathbf{x}_k) = f(\mathbf{x}_{k+1}) - f(\mathbf{x}_k) < 0 \tag{11}$$

where \mathbf{x}_k represents the corresponding k th iteration value of \mathbf{x} . Based on (11), the following iteration formula is obtained.

$$\mathbf{x}_{k+1} = \begin{cases} \mathbf{x}_k & \Delta f(\mathbf{x}_k) \geq 0 \\ \mathbf{x}_k + \zeta_k \cdot \mathbf{e}_p & \Delta f(\mathbf{x}_k) < 0 \end{cases} \tag{12}$$

where ζ_k denotes the step size of the k th iteration, and the symbol \cdot denotes the dot product.

The iteration formula of the step size is as follows.

$$\zeta_{k+1} = \begin{cases} \zeta_k & \text{others} \\ c \cdot \zeta_k & n = N \text{ and } \mathbf{x}_k = \mathbf{x}_{k-N+1} \end{cases} \tag{13}$$

where $0 < c < 1$ is a constant.

For DCD algorithm, the iterative direction is still the Euclidean coordinate direction, but the selections of the step size ζ and the constant c are different with the exact line search method. Suppose that each element of the vector \mathbf{x} is limited, i.e. $|\mathbf{x}_n| \leq A$, and A is known. A can be expressed as $A = 2^{N_b+p}$, where p is an arbitrary integer, and N_b is a positive integer. In order to reduce the complexity of hardware implementation, the corresponding parameters in (13) are set to $c = 1/2$ and $\zeta_0 = A/2$ respectively. In such a way, the multiplication is substituted by shift operation for hardware implementation.

3 Low Complexity SWA Based on DCD

The main idea of the low complexity SWA based on DCD (SWA-DCD) algorithm is as follows. The solution of the cost function (3) is equivalent to the solution of the normal equation. Moreover, the weight increment can be derived from the weight incremental canonical equation, which will be transformed to the normal equation form and solved by DCD algorithm. In the following, we will explain this idea in details.

3.1 The SWA-DCD Algorithm

Let the derivative of the cost function (3) to $\mathbf{w}(n)$ equal to zero, we get the following canonical equation.

$$\mathbf{R}(n)\mathbf{w}(n) = \mathbf{q}(n) \tag{14}$$

where $\mathbf{q}(n) = \frac{1}{\beta} \sum_{l=1}^n \lambda^{n-l} |y_{n,l}|^2 y_{n,l}^* \mathbf{x}(l)$.

The $\mathbf{R}(n)$ can be obtained by the iteration form.

$$\mathbf{R}(n) = \lambda \mathbf{R}(n-1) + \mathbf{x}(n)\mathbf{x}^H(n) \tag{15}$$

It is noted that unlike [4] we leave the parameter β on the right side of (14). In the following, we derive the weight incremental canonical equation of the equalizer coefficients.

We denote $\Delta \mathbf{R}(n), \Delta \mathbf{w}(n)$ and $\Delta \mathbf{q}(n)$ as follows.

$$\Delta \mathbf{R}(n) = \mathbf{R}(n) - \mathbf{R}(n-1) \tag{16}$$

$$\Delta \mathbf{w}(n) = \mathbf{w}(n) - \mathbf{w}(n-1) \tag{17}$$

$$\Delta \mathbf{q}(n) = \mathbf{q}(n) - \mathbf{q}(n-1) \tag{18}$$

According to Ref. [4], we have

$$\mathbf{R}(n)\Delta \mathbf{w}(n) = \mathbf{e}_{SWA}^*(n)\mathbf{x}(n)/\beta + \boldsymbol{\rho}(n)/\beta \tag{19}$$

$$\boldsymbol{\rho}(n) \approx \alpha C_{1,1}^s \mathbf{R}(n)\Delta \mathbf{w}(n) \tag{20}$$

Substitute (19) into (20), we get

$$\boldsymbol{\rho}(n) = c_{SWA} e_{SWA}^*(n) \mathbf{x}(n) \tag{21}$$

where $c_{SWA} = \alpha \cdot C_{1,1}^s / (\beta - \alpha \cdot C_{1,1}^s)$.

$\mathbf{q}(n)$ is updated in the following rule [4].

$$\mathbf{q}(n) = \lambda \mathbf{q}(n - 1) + |y(n)|^2 y^*(n) \mathbf{x}(n) / \beta + \boldsymbol{\rho}(n) / \beta \tag{22}$$

Substituting (21) into (22), we get the iteration formula of $\mathbf{q}(n)$.

$$\mathbf{q}(n) = \lambda \mathbf{q}(n - 1) + z^*(n) \mathbf{x}(n) \tag{23}$$

where $z(n) = (|y(n)|^2 y(n) + c_{SWA} \cdot e_{SWA}(n)) / \beta$.

Define the residual vector $\mathbf{r}(n)$

$$\mathbf{r}(n) = \mathbf{q}(n) - \mathbf{R}(n) \mathbf{w}(n) \tag{24}$$

Substitute (17) into (14), we obtain

$$\mathbf{R}(n) [\Delta \mathbf{w}(n) + \mathbf{w}(n - 1)] = \mathbf{q}(n) \tag{25}$$

Combining (16), (18) and (24), we get

$$\begin{aligned} \mathbf{R}(n) \Delta \mathbf{w}(n) &= \mathbf{q}(n) - \mathbf{R}(n) \mathbf{w}(n - 1) \\ &= \mathbf{q}(n) - [\mathbf{R}(n - 1) + \Delta \mathbf{R}(n)] \mathbf{w}(n - 1) \\ &= \mathbf{q}(n) - \mathbf{R}(n - 1) \mathbf{w}(n - 1) - \Delta \mathbf{R}(n) \mathbf{w}(n - 1) \\ &= \Delta \mathbf{q}(n) + [\mathbf{q}(n - 1) - \mathbf{R}(n - 1) \mathbf{w}(n - 1)] - \Delta \mathbf{R}(n) \mathbf{w}(n - 1) \\ &= \Delta \mathbf{q}(n) + \mathbf{r}(n - 1) - \Delta \mathbf{R}(n) \mathbf{w}(n - 1) \end{aligned} \tag{26}$$

Let $\mathbf{q}_0(n)$ represent the right side of (26) and we get the weight incremental canonical equation.

$$\mathbf{R}(n) \Delta \mathbf{w}(n) = \mathbf{q}_0(n) \tag{27}$$

where $\mathbf{q}_0(n) = \lambda \mathbf{r}(n - 1) + \gamma e_{SWA}^*(n) \mathbf{x}(n)$ (The derivation process is shown in the appendix), $\gamma = (1 + c_{SWA}) / \beta$.

The solution of (14) is transformed into solving (27), and (27) has similar form as the normal equation. It is clear that the solution of (27) can be achieved by the DCD algorithm. After we get the $\Delta \mathbf{w}(n)$, the weight vector can be calculated by $\mathbf{w}(n) = \mathbf{w}(n - 1) + \Delta \mathbf{w}(n)$.

We solve (14) iteratively way rather than directly, mainly based on the following two points. First, the calculation of $\mathbf{q}_0(n)$ involves less computation compared with that of $\mathbf{q}(n)$, and the calculation of $\mathbf{q}(n)$ will further enlarge the nonlinear effect involved in the algorithm. Second, with the same accuracy of calculating the weight vector, the iterative way will require a smaller number of iterations [8].

For the original SWA, the instability mainly comes from the nonlinear feedback of the filter output and finite arithmetic problems [4]. For the latter problem, it comes from the update of matrix inversion. For SWA-DCD algorithm, however, there is no need to calculate the inverse matrix, so this problem will not happen. In [4], a region of interest, which is determined by $|y(n)|^2 \leq \alpha \cdot C_{1,1}^s$, is provided to avoid the divergence caused by nonlinear feedback. In this paper, we take the same way to deal with the instability problem.

3.2 A Low Complexity Dual-mode Algorithm

When the output symbols have relatively low errors, using decision directed mode can further improve the performance of equalizer [13]. Here a dual-mode algorithm is presented. At the first stage the SWA-DCD algorithm is used, and low complexity RLS based on DCD (RLS-DCD) is taken at the second state. This dual-mode algorithm is named as SWA-RLS-DCD. The total complexity of the SWA-RLS-DCD algorithm is $O(N)$ order of magnitude.

The main difference between SWA-DCD and RLS-DCD is the calculation $\mathbf{q}_0(n)$ in (27), which is shown as follows.

$$\mathbf{q}_0(n) = \lambda \mathbf{q}_0(n - 1) + e_{RLS}^*(n)\mathbf{x}(n) \tag{28}$$

where $e_{RLS}(n) = \hat{u}(n) - y(n)$ and $\hat{u}(n)$ is the estimated symbol.

For dual-mode the switching point is a key factor, here we use a threshold to switch the different algorithms, which is determined by the following function.

$$g(y) = \cos^2(y\pi/2d) \tag{29}$$

where $2d$ stands for the minimum distance between any two symbols in a standard constellation.

The closer the output signal is to the constellation point of the transmitted signal, the smaller the value $g(y)$ is, and when it coincides with the signal constellation, the value is 0. Therefore the algorithm will switches to the DD mode when the following condition is reached.

$$g(y_{Re}(n)) < Th \text{ and } g(y_{Im}(n)) < Th \tag{30}$$

where Th is the threshold, and y_{Re}, y_{Im} are real and imaginary parts of $y(n)$ respectively.

The computation complexity of different algorithms is given in Table 1, where $P_a \leq (2N + 1)N_u + N_b$. From Table 1, it can be seen that the computation complexity of both SWA-DCD and RLS-DCD is comparable to that of CMA and LMS.

4 Simulation and Performance Analysis

For all simulations the transmitted signal is 64QAM. Two typical channels are used.

$$\begin{aligned} &[-0.005 - 0.004j \quad 0.009 + 0.03j \quad -0.024 - 0.104j \\ \text{Channel 1 [14]: } &0.854 + 0.52j \quad -0.218 + 0.273j \quad 0.049 - 0.074j \\ &-0.016 + 0.02j]; \end{aligned}$$

Table 1 Computation Complexity of different algorithms

| Algorithm | SWA | RLS | SWA-DCD | RLS-DCD |
|------------------------|----------------|----------------|-----------|---------|
| Operator | | | | |
| Real addition | 2 | 1 | $P_a + 1$ | P_a |
| Complex addition | $N^2 + 3N - 1$ | $N^2 + 3N - 1$ | $4N - 1$ | $4N$ |
| Real multiplication | $4N + 4$ | $6N$ | $3N + 2$ | $3N$ |
| Complex multiplication | $N^2 + 3N + 1$ | $N^2 + 3N$ | $3N + 2$ | $3N$ |

$$\text{Channel 2 [15]: } h_n = \begin{cases} \frac{1}{2} \left[1 + \cos \left(\frac{2\pi}{C} (n - 2) \right) \right] & n = 1, 2, 3 \\ 0 & \text{others} \end{cases}$$

Two evaluation indicators, MSE and residual ISI, are used for performance comparison.

Case 1 Comparison of SWA and SWA-DCD.

Here channel 1 is used. The length of the equalizer is 11. The SNR is set to 30 dB. The parameters λ, N_b and N_s are set to be 0.999, 16 and 12 respectively, β is 2.

The result is shown in Fig. 1. We can see that SWA-DCD converges relatively slower than SWA and they reach the same results when the algorithms converge. However, the computation complexity of SWA-DCD is much smaller than that of SWA, which is very obvious when we compare the complex addition and multiplication. The results verify that the proposed algorithm can greatly reduce the implementation complexity while ensuring the equalization performance.

Case 2 Comparison of dual-mode algorithm.

First the channel 1 is chosen. The parameters λ for SWA-RLS-DCD, SWA-DCD and SWA-DCD-LMS (at the second stage the LMS is used) are 0.995, 0.999 and 0.99 respectively. For DD mode the step size of LMS is 6×10^{-4} , N_b and N_s are equal to 16 and 12 respectively, the switching threshold Th is 0.9, The SNR is 30 dB.

The results are shown in Fig. 2. It is clear that dual-mode algorithm has faster convergence speed and lower MSE and residual ISI compared with the single mode. Comparing the two dual-mode algorithms, SWA-RLS-DCD has relatively lower steady state error and residual ISI but with almost the same convergence speed. However, the computation complexity of the two algorithms are almost on the same order, and SWA-RLS-DCD performs better than SWA-DCD-LMS with regard to both MSE and residual ISI.

Then we take channel 2 for simulation. The channel parameter C is set to 3.3, which corresponds a channel with eigenvalue spread more than 21. The length of the channel is 11. The parameter λ for SWA-RLS-DCD, SWA-DCD and SWA-DCD-LMS are 0.994,

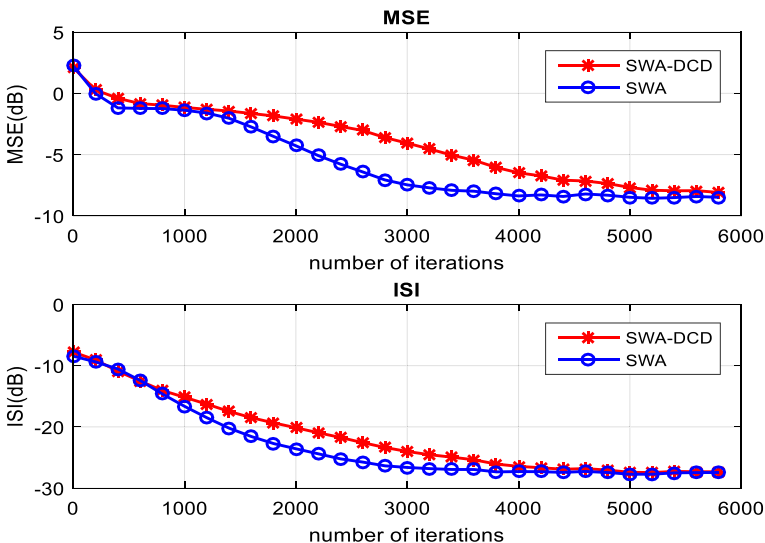


Fig. 1 Performance of SWA and SWA-DCD

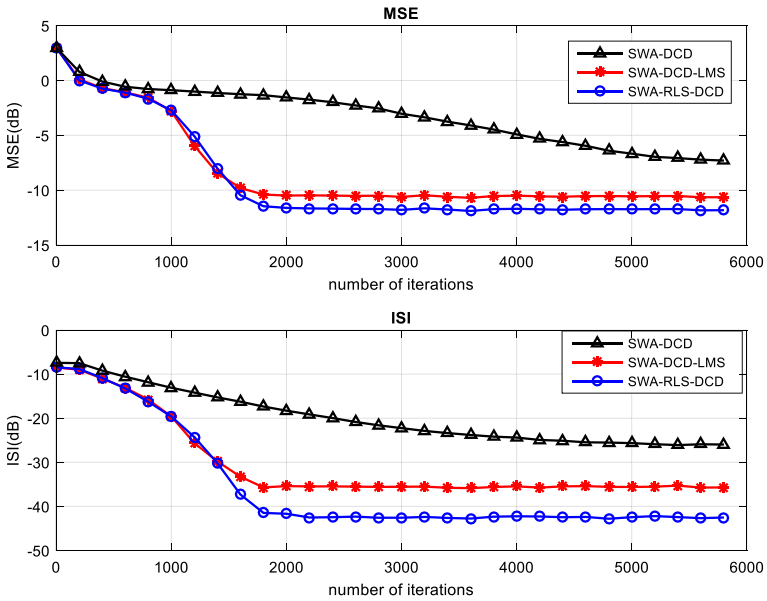


Fig. 2 Performance of dual-mode algorithm with channel 1

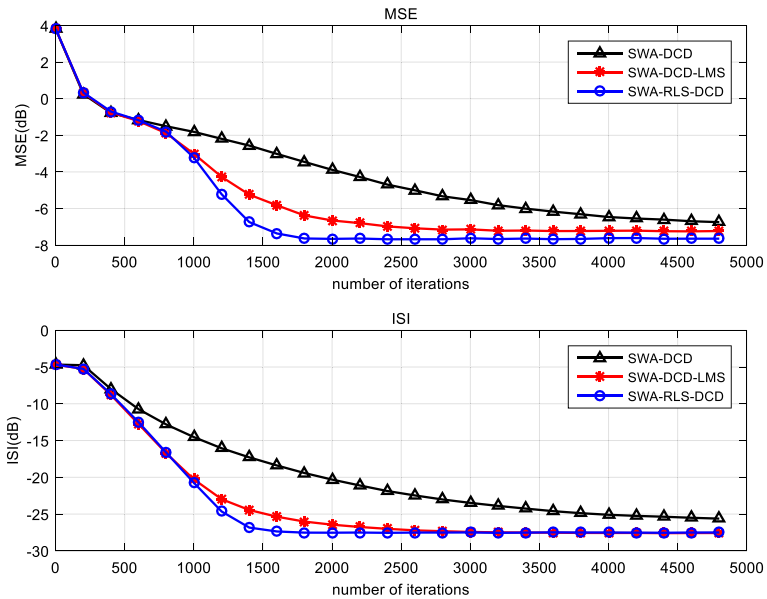


Fig. 3 Performance of dual-mode algorithm with channel 2

0.9995 and 0.996 respectively. The step size of LMS is 3×10^{-4} , N_b and N_s are equal to 8 and 2 respectively. The SNR is 30 dB.

The results are shown in Fig. 3. From the results we see that SWA-RLS-DCD also has relatively faster convergence speed compared with that of SWA-DCD-LMS.

Both the results in Figs. 2 and 3 show that the dual-mode algorithm can achieve better performance compared with the single mode algorithm. Since the computation complexity does not increase much, SWA-RLS-DCD is much more preferred.

The proposed algorithm solves the weight incremental norm equation rather than SWA problem directly, which avoids the computation of the matrix inversion and reduces the computation complexity with a great amount. Moreover, in the case of a small increase in computation complexity, the convergence speed of the dual-mode algorithm is obviously accelerated compared with the single mode algorithm, and the steady-state error and the residual ISI are smaller.

5 Conclusions

In this paper the DCD algorithm is introduced to lower the complexity of the SWA. At the same time a low complexity dual-mode algorithm is also presented. Simulations verify effectiveness of the proposed algorithm, especially for dual mode algorithm the proposed algorithm has not only low computation complexity but also better performance with regard to the MSE and residual ISI.

Appendix

$$\begin{aligned}
 \Delta \mathbf{R}(n) * \mathbf{w}(n-1) &= [\mathbf{R}(n) - \mathbf{R}(n-1)] * \mathbf{w}(n-1) \\
 &= [\lambda * \mathbf{R}(n-1) + \mathbf{x}(n) * \mathbf{x}^H(n) - \mathbf{R}(n-1)] * \mathbf{w}(n-1) \\
 &= (\lambda - 1) * \mathbf{R}(n-1) * \mathbf{w}(n-1) + \mathbf{x}(n) * \mathbf{x}^H(n) * \mathbf{w}(n-1) \\
 &= (\lambda - 1) * [\mathbf{q}(n-1) - \mathbf{r}(n-1)] + y(n)^* * \mathbf{x}(n)
 \end{aligned}$$

Substituting the above equation into the following equation.

$$\begin{aligned}
 \mathbf{q}_0(n) &= \mathbf{r}(n-1) + \Delta \mathbf{q}(n) - \Delta \mathbf{R}(n) * \hat{\mathbf{w}}(n-1) \\
 &= \mathbf{r}(n-1) + \Delta \mathbf{q}(n) - (\lambda - 1) * [\mathbf{q}(n-1) - \mathbf{r}(n-1)] - y(n)^* * \mathbf{x}(n) \\
 &= \lambda * \mathbf{r}(n-1) + [\Delta \mathbf{q}(n) + \mathbf{q}(n-1)] - \lambda * \mathbf{q}(n-1) - y(n)^* * \mathbf{x}(n) \\
 &= \lambda * \mathbf{r}(n-1) + \mathbf{q}(n) - \lambda * \mathbf{q}(n-1) - y(n)^* * \mathbf{x}(n) \\
 &= \lambda * \mathbf{r}(n-1) + |y(n)|^2 * y^*(n) * \mathbf{x}(n) / \beta + \rho(n) / \beta - y(n)^* * \mathbf{x}(n) \\
 &= \lambda * \mathbf{r}(n-1) + [(|y(n)|^2 - \beta) * y^*(n) * \mathbf{x}(n) + \rho(n)] / \beta \\
 &= \lambda * \mathbf{r}(n-1) + [e_{SWA}^*(n) * \mathbf{x}(n) + \rho(n)] / \beta \\
 &= \lambda * \mathbf{r}(n-1) + [e_{SWA}^*(n) * \mathbf{x}(n) + c_{SWA} * e_{SWA}^*(n) * \mathbf{x}(n)] / \beta \\
 &= \lambda * \mathbf{r}(n-1) + (1 + c_{SWA}) * e_{SWA}^*(n) * \mathbf{x}(n) / \beta
 \end{aligned}$$

Declarations

Conflict of interest On behalf of all authors, the corresponding author states that there is no conflict of interest.

References

1. Shalvi, O., & Weinstein, E. (1990). New criteria for blind deconvolution of nonminimum phase systems(channels). *IEEE Transactions on Information Theory*, *42*, 1145–1156.
2. Ahmed, S., Khan, Y., & Wahab, A. (2019). A review on training and blind equalization algorithms for wireless communications. *Wireless Personal Communications*, *108*, 1759–1783.
3. Johnson, R., Jr., Schniter, P., et al. (1998). Blind equalization using the constant modulus criterion: A review. *Proceedings of the IEEE*, *86*(10), 1927–1950.
4. Miranda, M. D., Silva, M., & Nascimento, V. H. (2008). Avoiding divergence in the Shalvi-Weinstein algorithm. *IEEE Transactions on Signal Processing*, *56*(11), 5403–5413.
5. Pavan, F. R. M., Silva, M. T. M., & Maria, D. (2019). Miranda, performance analysis of the multi-user Shalvi-Weinstein algorithm. *Signal Processing*, *163*, 153–165.
6. Abrar, S., & Nandi, A. K. (2010). Blind equalization of square-QAM signals: A multimodulus approach. *IEEE Transactions on Communications*, *58*(6), 1674–1685.
7. Filho, J. M., Miranda, M. D., et al. (2012). A regional multimodulus algorithm for blind equalization of QAM signals: Introduction and steady-state analysis. *Signal Processing*, *92*, 2643–2656.
8. Zakharov, Y. V., & Tozer, T. C. (2004). Multiplication-free iterative algorithm for LS problem. *Electronics Letters*, *40*(9), 567–569.
9. Zakharov, Y. V., White, G. P., & Liu, J. (2008). Low-complexity RLS algorithms using dichotomous coordinate descent iterations. *IEEE Transactions on Signal Processing*, *56*(7), 3150–3161.
10. Liu, J., Zakharov, Y. V., & Weaver, B. (2009). Architecture and FPGA design of dichotomous coordinate descent algorithms. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *56*(11), 2425–2438.
11. Chen, T., & Wang, S. (2020). An efficient nonlinear dichotomous coordinate descent adaptive algorithm based on random Fourier features. *IEEE Signal Processing Letters*, *27*, 1804–1808.
12. Silva, M., & Arenas-Garcia, J. (2013). A soft-switching blind equalization scheme via convex combination of adaptive filters. *IEEE Transactions on Signal Processing*, *61*(5), 1171–1182.
13. Jianqiu, S., Xingguang, L., Kang, C., Wei, C., & Ming, C. (2020). A novel CMA+DD_LMS blind equalization algorithm for underwater acoustic communication. *The Computer Journal*, *63*(6), 974–981.
14. Azim, A. W., Abrar, S., et al. (2015). Steady-state performance of multimodulus blind equalizers. *Signal Processing*, *108*, 509–520.
15. Simon, H. (2013). *Adaptive filter theory* (5th ed.). Pearson Publisher.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Sun Yongjun is an associate professor of Xidian University. He graduated from Northwestern Polytechnical University in 1997, and received his master degree and Ph.D degree from Xidian University in 2003 and 2007 respectively. From 2007 he is a full-time teacher of Xidian University. His research interests include communication signal processing, mobile communication and Satellite communication. He has published numerous papers in the areas of adaptive filtering and digital communications, and co-authored a book named digital communication theory and systems.



Liaoping Zhang received his B.E. degree from China University of Mining and Technology. He is currently a postgraduate of Xidian University. His research interests include communication signal processing and D2D communication.



Cuiyuan Jia was born in 1991. She received her master degree from Xidian University in 2015. She join in Shanghai Engineering Center for Microsatellites from 2016. Her research interests are blind equalization and communication signal processing.



Zujun Liu received BS, MS and Ph.D. degrees from Xi'an Jiaotong University, China, in 1998, 2001 and 2006, respectively. He was a R&D engineer in ZTE Ltd from 2001 to 2002. He joined Xidian University in 2006 as a Lecturer. He is currently an Associate Professor at the State Key Laboratory of Integrated Services Networks, Xidian University, China. His research interests include MIMO, relay communications and radio resource allocation.