Signal Processing 108 (2015) 1–12

Contents lists available at ScienceDirect

Signal Processing

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

Decision Feedback Equalization using Particle Swarm Optimization

Naveed Iqbal [a], Azzedine Zerguine [a,*], Naofal Al-Dhahir [b]

[a] Department of Electrical Engineering, King Fahd University of Petroleum & Minerals, Dhahran 31261, Saudi Arabia
[b] Department of Electrical Engineering, University of Texas at Dallas, Richardson, TX 75083, USA

ARTICLE INFO

Article history:
Received 3 March 2014
Received in revised form
3 July 2014
Accepted 28 July 2014
Available online 13 August 2014

Keywords:
Decision Feedback Equalization
Particle Swarm Optimization
Least Mean Square algorithm

ABSTRACT

It is well-known that the Decision Feedback Equalizer (DFE) outperforms the Linear Equalizer (LE) for highly dispersive channels. For time-varying channels, adaptive equalizers are commonly designed based on the Least Mean Square (LMS) algorithm which, unfortunately, has the limitation of slow convergence specially in channels having large eigenvalue spread. The eigenvalue problem becomes even more pronounced in Multiple-Input Multiple-Output (MIMO) channels. Particle Swarm Optimization (PSO) enjoys fast convergence and, therefore, its application to the DFE merits investigation. In this paper, we show that a PSO-DFE with a variable constriction factor is superior to the LMS/RLS-based DFE (LMS/RLS-DFE) and PSO-based LE (PSO-LE), especially on channels with large eigenvalue spread. We also propose a hybrid PSO–LMS-DFE algorithm, and modify it to deal with complex-valued data. The PSO–LMS-DFE not only outperforms the PSO-DFE in terms of performance but its complexity is also low. To further reduce its complexity, a fast PSO–LMS-DFE algorithm is introduced.

1. Introduction

The decision feedback equalizer (DFE) [1] is an effective Inter-symbol Interference (ISI) mitigation technique and can significantly outperform the linear equalizer (LE) on highly dispersive channels. Adaptive equalization is attractive for time-varying channels, and for this purpose, adaptive algorithms, e.g., the Least Mean Square (LMS) and the Recursive Least Squares (RLS) [2], are widely used. Recently, heuristic techniques applied to equalization/estimation problems, in particular the Particle Swarm Optimization (PSO) technique, showed significant improvement over conventional algorithms [3–6]. It was shown in [6] that the application of PSO to an adaptive linear equalizer provides fast convergence compared to its LMS-based counterpart. To further explore its equalization capabilities, this work investigates the effectiveness of the PSO when applied to a DFE structure.

The PSO is a robust algorithm with fast convergence. It is simple, very easy to code, and has low memory requirements. It can be used to solve multi-modal, non-differential and nonlinear problems [7]. It uses position and velocity update equations to search for the global minimum. Each particle uses its own information and its neighbors information to adjust its position within the search space. In addition, the PSO works based on cooperation among the particles as opposed to the other Evolutionary Algorithms (EA). The PSO has demonstrated its distinguished performance in many engineering applications. To mention a few of its recent applications, it has been used in image processing [8], channel prediction [9], and nonlinear active noise control systems [10].

The PSO is used to optimize real- and continuous-valued functions in an l-dimensional space. The particles

* Corresponding author.
E-mail addresses: naveediqbal@kfupm.edu.sa (N. Iqbal), azzedine@kfupm.edu.sa (A. Zerguine), aldhahir@utdallas.edu (N. Al-Dhahir).

http://dx.doi.org/10.1016/j.sigpro.2014.07.030
0165-1684/

constitute the swarm, also known as population, and move in a predefined search space. The position of each particle within the search space represents a possible solution to the problem. Here, in the case of adaptive equalization, the position represents the weights of the equalizer.

Despite its advantages, the PSO is vulnerable to local minima, i.e., the particles become stagnant around the global minima and may not be able to reach the global minimum [11]. To deal with this issue, we have introduced a hybrid of PSO and LMS algorithms, which not only solves the problem of particle stagnancy but also reduces the number of computations required in the PSO. Another disadvantage of PSO is that it assumes real-valued data. In [6], the authors use a BPSK signal with a real-valued channel impulse response. However, in reality, we have to deal with complex numbers for pass-band transmission then the tap weights of the equalizer become complex. To solve this issue, we have modified the PSO algorithm to handle the complex case without increasing its complexity. To further reduce the complexity, we have introduced the fast PSO–LMS-DFE. Since the major complexity factor in the PSO is the convolution operation required to find the equalizer output, the PSO–LMS-DFE performs this operation in the frequency-domain using the FFT to save on computations.

Here, this work is extended to Multiple-Input Multiple-Output (MIMO) scenario. Due to its high computational complexity, the most challenging task in designing the MIMO receiver is its corresponding MIMO channel equalizer. A MIMO equalizer has to deal with the inter-symbol and the inter-stream interference. Several works proposed different methods for adaptive MIMO DFE. Among them, the Vertical Bell Labs layered space–time (V-BLAST) architecture [12] is one of the promising methods for MIMO equalization. Computationally efficient V-BLAST techniques have also been proposed in [13–15] assuming a known channel. Its application to time varying channels is difficult due to frequent estimation of the MIMO channel. An efficient adaptive MIMO equalizer based on V-BLAST and generalized DFE [16] has been presented in [17], where are symbol detection order as well as the equalizer taps is updated recursively; however, this structure suffers from numeric instability. To address this problem, a technique based on square-root factorization of the equalizer input correlation matrix was proposed in [18]. However, unlike the application of MIMO DFE to time-invariant channels, the application of MIMO DFE to time-varying channels still requires excessive computations for the estimation of the parameters. Another challenging problem in these techniques is that substantial training is required when the equalizer length becomes large (as in [19]) and therefore a large number of symbols are needed before the algorithm converges. Algorithms based on reduced rank equalization [20] are less complex and require less training symbols as compared to full rank equalization, while requiring matrix inversion at each iteration. To overcome this problem, in [21,22] the covariance matrix is estimated iteratively and hence the matrix inversion operation is avoided and therefore have a moderate complexity. However, all of the above mentioned techniques use the RLS algorithm which is often complex to implement and prone to instability in a real time environment. Therefore, PSO-based algorithms can be a substitute to the RLS-based algorithms with moderate complexity and guaranteed stability as they do not have to calculate the inverse of the autocorrelation matrix of the input signal. Moreover, some algorithms may require more than 150 symbols for the training phase which is not suitable for frame-based applications, e.g., IEEE 802.11*p*, where the frame contains less than 150 Orthogonal Frequency Division Multiplexing (OFDM) symbols.

This work reports a detailed analysis for the adjustment of the PSO parameters to ensure the best performance. The superiority of the PSO algorithms is tested on channels with different eigenvalue spreads specifically in MIMO channels where the performance of the LMS/RLS-DFE can be very bad due to the severe eigenvalue spread problem. Our results demonstrate the performance gain of the proposed algorithms over conventional algorithms.

The paper is organized as follows. In Section 2, we revise the basic PSO algorithm and its variants. Problem formulations for SISO and MIMO systems are given in Section 3. Section 4 introduces the PSO–LMS algorithm and in Section 5, PSO–LMS is modified to deal with complex-valued data. In Section 6, simulation results are shown to verify the benefits of proposed algorithms. The complexity of PSO algorithms is discussed in Section 7. Finally, Section 8 draws the conclusions.

2. The PSO algorithm

2.1. BASIC PSO

Initially, random solutions are assigned to n particles in a d-dimensional search space. The basic PSO algorithm [7] consists of the following elements:

Particle position ($\boldsymbol{p}_{i,k}$): The particle position is represented by a real-valued l-dimensional vector which is the potential solution to the problem at hand. The particle position is the weight vector of the equalizer in our case. The position of the ith particle at instant k is denoted by $\boldsymbol{p}_{i,k} = [p_i(0), p_i(1), p_i(2), \ldots, p_i(l)]$, where $p_i(l)$ represents the ith particle position in the lth dimension.

Particle velocity ($\boldsymbol{v}_{i,k}$): The velocity is also represented by a real-valued l-dimensional vector. The velocity of the ith particle at instant k is given as $\boldsymbol{v}_{i,k} = [v_i(0), v_i(1), v_i(2), \ldots, v_i(l)]$, where $v_i(l)$ represents the ith particle velocity in the lth dimension.

Inertia weight (i_w): This parameter controls the change of velocity between successive iterations. It affects the local and global search capabilities of the particles.

Particle or local best ($\boldsymbol{pbest}_{i,k}$): Each particle remembers its best position $\boldsymbol{pbest}_{i,k}$. The best position is the one which results in the minimum (or the maximum depending on the problem at hand) value of the cost function.

Global best ($\boldsymbol{gbest}_k$): The best value of all the $\boldsymbol{pbest}_{i,k}, i = 1, 2, \ldots, n$ is calculated by comparing the cost function values associated with them. This is the global best $\boldsymbol{gbest}_k$ of the swarm.

Stopping criteria: The algorithm is terminated when the global minimum (or maximum) is attained or after reaching a predefined number of iterations.