Grey Wolf Optimization Algorithm for Embedded Adaptive Filtering Applications

Guillermo Salinas, Eduardo Pichardo, Ángel A. Vázquez, Juan G. Avalos, and Giovanny Sánchez

***Abstract*—Nowadays, metaheuristic algorithms have been emerged as a potential solution in adaptive filtering applications since they offer good convergence properties. Nonetheless, most of them fall into a local minimum since their optimization is based on a single-solution technique. As a consequence, these algorithms present a high misadjustment level and require a large population to find the optimal solution. Recently, the grey wolf optimization (GWO) algorithm has emerged as a potential solution since it requires a smaller population and possesses a stronger global optimization ability with lesser control parameters. From an engineering perspective, its compactness is an attractive feature. Therefore, this opens new horizons in the implementation of this algorithm in resource-constrained devices. In this letter, we present for the first time the use of the GWO algorithm for system identification and acoustic echo canceller (AEC) and its implementation in a field programmable gate array (FPGA) device to validate its effectiveness. Our results show that the use of the GWO algorithm achieves lower steady-state mean square error (MSE) and requires less computational resources when compared with one of the most used metaheuristic algorithm.**

***Index Terms*—Acoustic echo cancelation, adaptive filtering, grey wolf optimization (GWO) algorithm, particle swarm optimization (PSO) algorithm, system identification.**

I. Introduction

NOWADAYS, most of the adaptive algorithms use gradient-descent methods since they offer low computational cost [1]. Nonetheless, some of these algorithms present limitations since they tend to be multimodal, i.e., are more likely to be trapped in the local minimum and never converge to the global optimum [2]. One potential solution can be provided by the use of metaheuristic algorithms since these algorithms extend the search space area. This can be achieved since they offer good exploration and exploitation capabilities. One of the most popular metaheuristic algorithms is the grey wolf optimization (GWO) [3]. Recently, this algorithm has been widely used in several applications. However, its performance has not been fully exploited for adaptive filtering applications. Specifically, the GWO algorithm is very attractive since it shows a good balance between exploration and exploitation compared with the most recent metaheuristic algorithms [4]. In addition, this algorithm possesses the advantage of simplicity and a lower number of parameters to adjust [5]. Recently, several promising works have been developed by considering these features. For example, Verma and Gupta [6] used the GWO algorithm for de-noising electromyogram (EMG) signals. However, it may not be suitable for all usages, specially when high-order adaptive filters are required. In this letter, we propose the use of the block-filtering method to implement the GWO algorithm for acoustic echo cancelation, in which high order adaptive filters are demanded. In this way, its implementation in embedded systems can be feasible.

Manuscript received 16 November 2022; accepted 15 December 2022. Date of publication 19 December 2022; date of current version 29 February 2024. This work was supported by the Instituto Politécnico Nacional, Mexico city. This manuscript was recommended for publication by L. D. Micco. *(Corresponding authors: Juan G. Avalos; Giovanny Sánchez.)*

The authors are with the Instituto Politécnico Nacional, ESIME Culhuacan, Mexico City 04440, Mexico (e-mail: gsalinasr1500@alumno.ipn.mx; edua_95pim@hotmail.es; avazquezp1301@alumno.ipn.mx; javaloso@ipn.mx; gsanchezriv@ipn.mx).

Digital Object Identifier 10.1109/LES.2022.3230364

II. Adaptive Filtering Using GWO Algorithm

Typically, in general adaptive filtering applications, the entire input vector is not always available or is too lengthy. Therefore, this factor significantly reduces the performance of the metaheuristic algorithms. Here, we propose the use of the block-filtering method to segment the input vector. In this way, we can easily estimate the actual error [7]. Another aspect to be considered is the definition of the cost function since this is used for the fitness evaluation of metaheuristic algorithms, which is obtained by performing a difference between the desired signal and the output of the adaptive filter, as shown in Fig. 1. Therefore, the fitness evaluation is in function of the MSE_p, which is expressed as follows:

$$\mathrm{MSE}_{p=1,2,\ldots,P} = 10\,\log_{10}\left(\frac{1}{L}\sum_{n=1}^{L}[d(n)-y_p(n)]^2\right) \tag{1}$$

where P represents the population size, $d(n)$ represents the desired signal, y_p is the output of the pth wolf, and L is the length of the block. In addition, $x(n)$ is known as reference signal and $e(n)$ is the error signal, which is obtained as follows:

$$e(n) = d(n) - y(n). \tag{2}$$

III. Grey Wolf Optimization Algorithm

The GWO algorithm is inspired by the social and leadership behavior of grey wolves [3]. Commonly, their group size is in a range between 5 and 12 wolves, and they have a strict social dominant hierarchy. In their social group, there are four types of wolves. The first category is known as alpha wolf (α), which represents the fittest solution of the population. Beta (β) and delta (δ) wolves are the second and third best solutions in

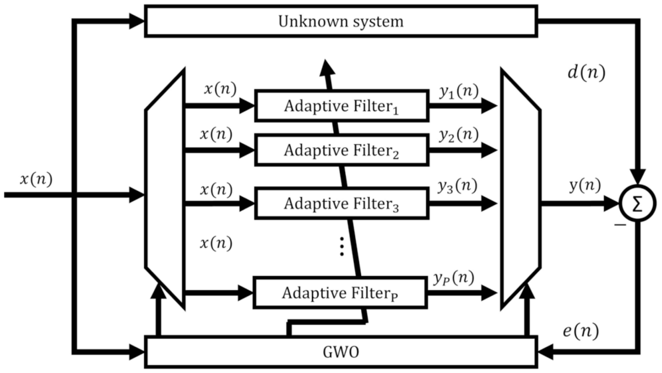

Fig. 1. Proposed structure for adaptive filtering using the GWO as tuning algorithm.

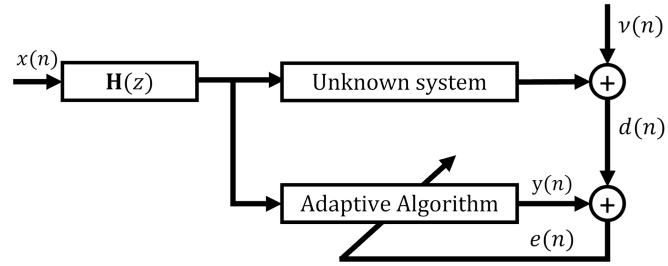

Fig. 2. General structure of a system identification application.

the population, respectively. The fourth category, called omega wolves (ω) are the best solution candidates. In general terms, the GWO algorithm assumes that hunting is performed by alpha, beta, and delta wolves, while omega wolves follow these wolves [8]. On the other hand, the hunting process can be divided into three parts: 1) tracking, chasing, and approaching their prey; 2) pursuing, encircling and harassing the prey until it stops moving; and 3) attacking their prey. To encircle the prey, the following equations are used:

$$\vec{D} = |\vec{C} \times \vec{X}_{\text{prey}}(t) - \vec{X}(t)| \tag{3}$$

$$\vec{X}(t+1) = \vec{X}_{\text{prey}}(t) - \vec{A} \times \vec{D} \tag{4}$$

where $\vec{D}$ depicts the distance between the prey and the current wolf, $\vec{X}_{\text{prey}}$ is the position vector of the prey, $\vec{X}$ is the position of the grey wolves, $\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$ and $\vec{C} = 2 \cdot \vec{r}_2$ define coefficient vectors used to relocate around another solution, $\vec{r}_1$ and $\vec{r}_2$ are random vectors between [0, 1], and $\vec{a}$ is linearly decreased from 2 to 0, over the course of iterations

$$\vec{a}(t) = 2 - \frac{2t}{\text{MaxIter}} \tag{5}$$

where t depicts the current iteration and MaxIter is the total number of iterations.

The mathematical model of the hunting behavior of grey wolves assumes that the three best solutions α, β, and δ wolves have better knowledge about the location of the prey (optimal solution). For this reason, the other wolves in the population move according to the position of these three wolves [3]

$$\begin{aligned} \vec{D}_\alpha &= |C_1 \times \vec{X}_\alpha - \vec{X}(t)| \\ \vec{D}_\beta &= |C_2 \times \vec{X}_\beta - \vec{X}(t)| \\ \vec{D}_\delta &= |C_3 \times \vec{X}_\delta - \vec{X}(t)| \end{aligned} \tag{6}$$

$$\begin{aligned} \vec{X}_1 &= |\vec{X}_\alpha - \vec{A}_1\vec{D}_\alpha| \\ \vec{X}_2 &= |\vec{X}_\beta - \vec{A}_2\vec{D}_\beta| \\ \vec{X}_3 &= |\vec{X}_\delta - \vec{A}_3\vec{D}_\delta| \end{aligned} \tag{7}$$

where $\vec{X}_\alpha$, $\vec{X}_\beta$, and $\vec{X}_\delta$ represent the best three wolves at each iteration. Finally, the new position of the prey is obtained in the function of the mean of the positions of the three best wolves in the population

$$\vec{X}_{\text{prey}}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}. \tag{8}$$

IV. Performance Analysis

The GWO algorithm was simulated in MATLAB to be compared with the particle swarm optimization (PSO) algorithm [9]. To make this comparison, we tested them in two adaptive filtering applications to perform unknown system identification and acoustic echo cancelation.

A. System Identification

Let us consider the following signal $d(n) = \mathbf{w}^t(n) * \mathbf{x}(n) + \nu(n)$, where $\mathbf{x}(n) = [x(n),\ x(n-1)\ ,\ldots,\ x(n-N+1)]$ contains N samples of the reference signal, which is the input to the unknown system and the adaptive algorithm. The input signal is a white Gaussian noise with unit variance, $\mathbf{w}(n) = [w(1),\ w(2)\ ,\ldots,\ w(N-1)]$ represents the FIR filter weights of the unknown system (low-pass filter, $F_c = 400$ Hz). Such a filter generates an input signal with a large eigenvalue spread, which is useful to evaluate the tracking and convergence capabilities of the algorithm [10]. The tap length of the unknown system is $N = 50$. ν is a zero mean additive white Gaussian noise and the signal-to-noise ratio (SNR) is 30 dB, which is used as measurement noise to evaluate adaptive filters [11], [12]. In addition, we produce a highly colored signal by filtering the signal $\mathbf{x}$ by means of the system $\mathbf{H}(z) = 0.35 + z^{-1} + 0.35z^{-2}$, as shown in Fig. 2.

To test the tracking capabilities of both algorithms, an abrupt change is induced midway through the iterations. This was done by multiplying the unknown system by −1. Fig. 3 shows the mean square error (MSE) learning curves of PSO and GWO by selecting different population sizes. It is important to keep in mind that this result was obtained carrying out a single experiment.

As can be observed from Fig. 3, the PSO algorithm reaches the minimum MSE level (30 dB) using $P = 200$. On the other hand, the GWO algorithm achieves a 30-dB MSE level utilizing only $P = 10$. In addition, the GWO algorithm is capable of reducing the MSE to 50 dB by selecting $P = 200$. However, its computational cost increases almost 20 times. Furthermore, both algorithms quickly adapt and reduce the MSE again. Nonetheless, the GWO algorithm converges faster and keeps on reducing its error signal.

B. Acoustic Echo Canceller

The acoustic echo canceller (AEC) is commonly used in speaker phones for teleconferencing and hands-free communication in mobile environments. Fig. 4 shows a general

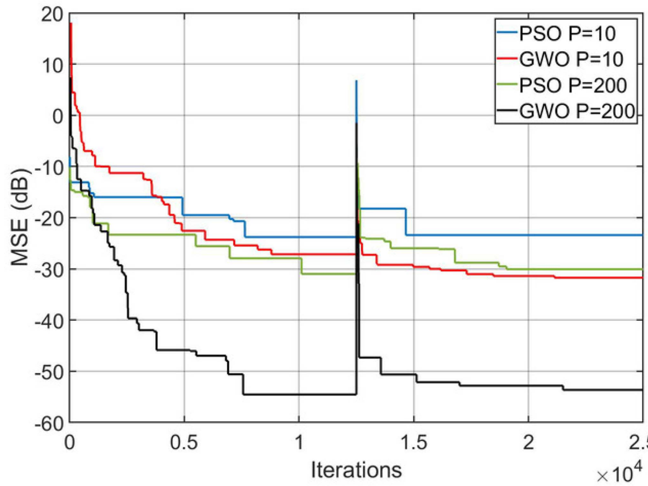

Fig. 3. MSE learning curves using different population sizes with the PSO and GWO algorithms.

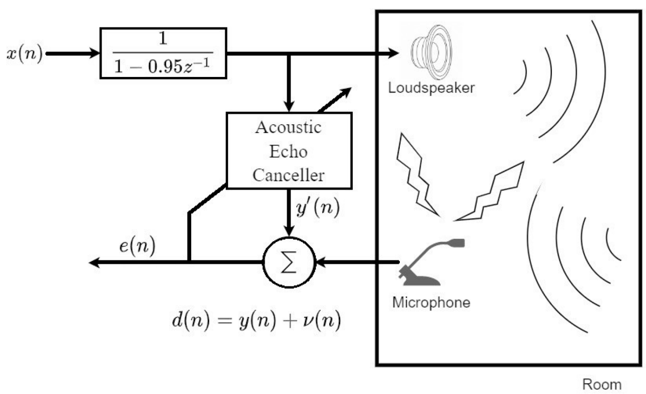

Fig. 4. AEC block diagram.

structure of an AEC system, where $x(n)$ depicts a speech signal from a far end room, $y(n)$ represents the echo of the signal $x(n)$, and $\nu(n)$ is the noise produced in the room where the conference takes place.

For this experiment, the input signal $x(n)$ is an AR(1) process generated by filtering a white Gaussian noise with the system $1/(1-0.95z^{-1})$, the impulse response is obtained from [13]. In this experiment, we use 300 coefficients to represent the adaptive filter, which has the same length as the impulse response of the acoustic path. Finally, to measure the performance of the AEC system, the echo return loss enhancement (ERLE) is obtained, as shown in Fig. 5.

The results demonstrate that the PSO algorithm exhibits a faster convergence speed. In contrast, the GWO achieves 10-dB higher ERLE level requiring almost 82% fewer number of multiplications and 90% fewer additions per iteration.

C. Analysis of Computational Cost

The computational cost is analyzed in terms of number of multiplications and additions, as shown in Table I. For this evaluation, we consider two typical cases: 1) $P = 10$ and $N = 50$ and 2) $P = 200$ and $N = 50$. According to previous experiments, the GWO algorithm exhibits higher

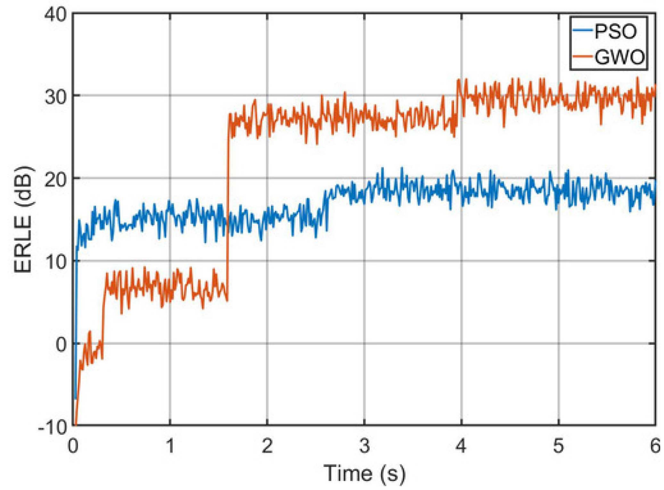

Fig. 5. ERLE curves for $P = 10$ for the GWO algorithm and $P = 200$ for the PSO algorithm.

TABLE I
NUMBER OF MULTIPLICATIONS AND ADDITIONS PER ITERATION REQUIRED BY THE PSO AND GWO ALGORITHMS. TYPICAL CASE: $N = 50$

Operation	Algorithm	Equation	$P = 10$	$P = 200$
Multiplication	PSO	$5NP + 2$	2,502	50,002
	GWO	$15NP + 1$	9,001	150,001
Addition	PSO	$5NP + 2$	2,502	50,002
	GWO	$9NP + 1$	4,501	90,001

performance in terms of convergence properties by using a smaller population when compared with the PSO algorithm.

D. Proposed GWO Processor

Once the GWO algorithm was simulated, we designed a specific 16-bit fixed-point processor called GWO processor to simulate and implement it in a field programmable gate array (FPGA) device. Specifically, we implemented the GWO algorithm in an FPGA DE0 Cyclone V 5CEBA4F23C7N [14], as shown in Fig. 6. To build this architecture, we use an embedded BRAM memory, multipliers, and adders. To optimize the area consumption, we use the time-multiplexing technique to simulate the search agents by using the same processing core [15]. In this way, the area consumption can be optimized at the cost of increasing the processing time. To demonstrate the performance of the GWO processor, we develop a system identifier composed of an adaptive FIR filter of 24 coefficients, see Fig. 7.

To test the prototype, we use a low-pass filter as unknown system $W_{\text{opt}}(z)$, a white Gaussian noise with unit variance as input signal $x(n)$, and an additive Gaussian noise with an SNR of 30 dB to corrupt the output of the unknown system. Fig. 8 shows the performance of the proposed GWO processor in terms of the MSE level. To obtain this performance, the number of taps of the adaptive filter is the same as the unknown system. As can be observed, the GWO exhibits good convergence properties despite using limited precision.

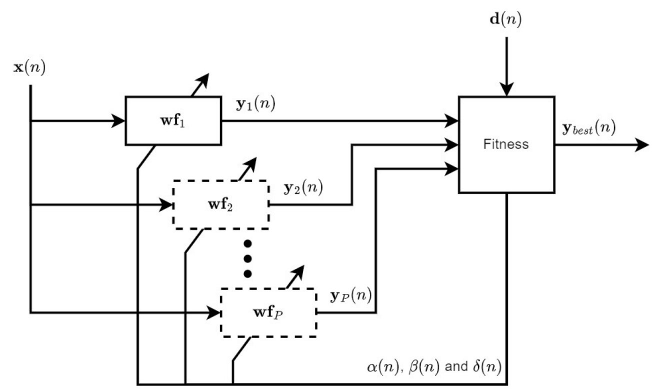

Fig. 6. Proposed GWO processor based on the time-multiplexing technique.

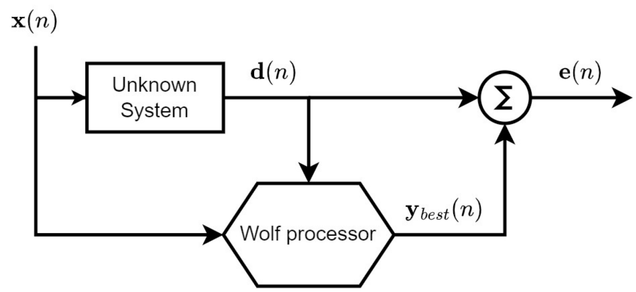

Fig. 7. General scheme of the system identifier based on the proposed GWO processor.

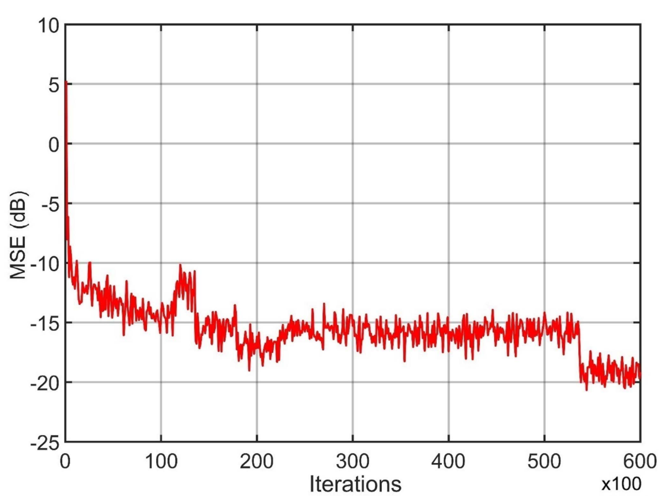

Fig. 8. MSE curve for the GWO algorithm.

In particular, the implementation of the proposed GWO processor demands 24K programmable logic elements and 14-kB embedded memory.

V. Conclusion

In this letter, we propose the use of the metaheuristic algorithm called GWO for adaptive filtering applications. Our results demonstrate that the GWO algorithm can be seen as a potential solution for adaptive system identification and acoustic echo cancelation since it exhibits a high converge speed, a low steady-state MSE, and good tracking capabilities, specially when abrupt changes occur. In addition, in comparison with the PSO algorithm, it offers a better reduction of the error signal. Furthermore, the GWO algorithm requires a small population and it reaches a low misadjustment; such characteristics open new horizons in the development of prototypes in embedded devices, due to its low computational complexity. This can be confirmed since its implementation on FPGA devices can be effective to increase the processing speed by using the parallel capabilities of these embedded devices. As a consequence, these implementations can be used in advanced applications, such as mobile robotics, real-time training artificial neural networks, online parameter estimation, computer vision, among others [16].

References

[1] P. S. R. Diniz, *Adaptive Filtering*, Cham, Switzerland: Springer, 2020.

[2] N. Yu, Z. Li, Y. Wu, and R. Feng, “Quantum-behaved particle swarm optimization-based active noise control system with timing varying path,” *J. Low Freq. Noise Vibrat. Active Control*, vol. 40 no. 1, pp. 524–539, Jan. 2020.

[3] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, Mar. 2014.

[4] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, “Grey wolf optimizer: A review of recent variants and applications,” *Neural Comput. Appl.*, vol. 30, no. 2, pp. 413–435, 2018.

[5] R.-E. Precup, R.-C. David, and E. M. Petriu, “Grey wolf optimizer algorithm-based tuning of fuzzy control systems with reduced parametric sensitivity,” *IEEE Trans. Ind. Electron.*, vol. 64, no. 1, pp. 527–534, Jan. 2017.

[6] A. R. Verma and B. Gupta, “A novel approach adaptive filtering method for electromyogram signal using Gray Wolf optimization algorithm,” *SN Appl. Sci.*, vol. 2, no. 1, pp. 1–11, 2020.

[7] U. K. Acharya and S. Kumar, “Particle swarm optimization exponential constriction factor (PSO-ECF) based channel equalization,” in *Proc. 6th Int. Conf. Comput. Sustain. Global Develop. (INDIACom)*, 2019, pp. 94–97.

[8] F. A. Senel, F. Gökçe, A. S. Yüksel, and T. Yiğit, “A novel hybrid PSO-GWO algorithm for optimization problems,” *Eng. Comput.*, vol. 35, no. 4, pp. 1359–1373, Dec. 2018.

[9] N. K. Rout, D. P. Das, and G. Panda, “Particle swarm optimization based active noise control algorithm without secondary path identification,” *IEEE Trans. Instrum. Meas.*, vol. 61, no. 2, pp. 554–563, Feb. 2012.

[10] S. R. Meena and C. S. Rai, “Effect of eigenvalue spread in noise cancellation of two sensory systems using adaptive algorithms,” *J. Statist. Manag. Syst.*, vol. 23, no. 1, pp. 157–169, Jan. 2020.

[11] W. A. Martins, M. V. Lima, P. S. Diniz, and T. N. Ferreira, “Optimal constraint vectors for set-membership affine projection algorithms,” *Signal Process.*, vol. 134, pp. 285–294, May 2017.

[12] J. H. Choi, H. Cho, J. J. Jeong, and S. W. Kim, “Combination of step sizes for affine projection algorithm with variable mixing parameter,” *Electron. Lett.*, vol. 49, no. 18, pp. 1149–1150, 2013.

[13] C. Paleologu, S. Ciochina, and J. Benesty, “Variable step-size NLMS algorithm for under-modeling acoustic echo cancellation,” *IEEE Signal Process. Lett.*, vol. 15, pp. 5–8, Jan. 2008.

[14] *DE0-CVUserManual*, TerasicInc, Hsinchu City, Taiwan, 2016.

[15] G. Sanchez et al., “A compact neuromorphic architecture with dynamic routing to efficiently simulate the FXECAP-L algorithm for real-time active noise control,” *Appl. Soft Comput.*, vol. 91, Jun. 2020, Art. no. 106233.

[16] A. Ortiz, E. Mendez, D. Balderas, P. Ponce, I. Macias, and A. Molina, “Hardware implementation of metaheuristics through LabVIEW FPGA,” *Appl. Soft Comput.*, vol. 113, Dec. 2021, Art. no. 107908.