

# A Novel Design Method for Digital FIR/IIR Filters Based on the Shuffle Frog-Leaping Algorithm

Daniel Jiménez-Galindo  
Universidad de Valladolid  
Valladolid, Spain  
danieljiga@gmail.com

Pablo Casaseca-de-la-Higuera  
Dept. Signal Theory and Communications  
Universidad de Valladolid  
Valladolid, Spain  
casaseca@lpi.tel.uva.es

Luis M. San-José-Revuelta  
Dept. Signal Theory and Communications  
Universidad de Valladolid  
Valladolid, Spain  
lsanjose@tel.uva.es

**Abstract**—The design of both FIR and IIR digital filters is a multi-variable optimization problem, where traditional algorithms fail to obtain optimal solutions. A modified Shuffled Frog Leaping Algorithm (SFLA) is here proposed for the design of FIR and IIR discrete-time filters as close as possible to the desired filter frequency response. This algorithm can be considered a type of memetic algorithm. In this paper, simulations prove the obtained filters outperform those designed using the traditional bilinear Z transform (BZT) method with elliptic approximation. Besides, results are close to, and even slightly better, than those reported in recent bio-inspired approaches using algorithms such as particle swarm optimization (PSO), differential evolution (DE) and regularized global optimization (RGA).

**Index Terms**—FIR design, IIR design, shuffled frog-leaping algorithm, memetic algorithm, bio-inspired algorithm

## I. INTRODUCTION

Digital filters are essential components in every signal processing system as they enhance valuable information, either by separating it from other unwanted signals or by attenuating the noisy components found in the raw signal.

Two types of digital filters are studied in this paper: Finite Impulse Response (FIR) and Infinite Impulse Response (IIR) digital filters. FIR filters produce an output that is only based on current and past inputs, ensuring stability. On the other hand, IIR filters produce an output based not only on its input, but also on the previous values of the output. As a consequence, stability is not guaranteed and is a desirable characteristic when designing them.

IIR digital filters imply about twice as much coefficients as the FIR filters, resulting in a better frequency response for the same filter order, nevertheless increasing the computational load of the proposed memetic algorithm. In addition, the search space must be limited to ensure the stability of the filter, introducing a number of checks that slow down convergence speed.

Many conventional methods allow the design of digital filters, but they only offer a limited range of parameters to be tuned-up in order to control the impulse response of the filter [1]. In addition, several optimization algorithms have already been used for digital filter design, but they suffer from a great limitation as they frequently get stuck in local minima, failing to achieve an optimal solution [2].

One of the first bio-inspired algorithms proposed for filter design used Genetic Algorithms (GAs) [3]. These are based on the theory of evolution formulated by Charles Darwin and are quite simple to implement. Genetic algorithms established the basis for more recent memetic algorithms (MAs), which introduce the concept of *meme* [2], alluding the capability of each individual to communicate with the others.

The here proposed memetic algorithm is based on the traditional *Shuffled Frog-Leap Algorithm* (SFLA) initially described in [4]. The algorithm relies on the analogy of a pond populated by frogs leaping to different positions that represent possible solutions to the studied problem. Various modifications have been applied so that the algorithm tackles the filter design challenge in a more efficient way.

Different techniques have been introduced to improve the basic algorithm such as the one mentioned in [4], which aims at ensuring a good population diversity, avoiding premature convergence and taking advantage of diversified search. These desired characteristics imply a more complex algorithm (compared to the standard one) that focuses on the quality of the final solution while sacrificing speed and computational resources.

The rest of the paper is organized as follows: Section II contains the literature review focused on bio-inspired algorithms. Section III describes the mathematical framework for IIR and FIR filter design. Section IV outlines the main characteristics of the proposed method. Section V presents the results obtained by the studied SFLA together with a comparison with those obtained by both classical and bio-inspired strategies. Finally, conclusions are gathered in Section VI.

## II. LITERATURE REVIEW

Bio-inspired optimization algorithms have been studied since the 50's, but they did not gain enough importance till the 90's due to the lack of computers with enough processing capability [5]. As soon as they were efficiently implemented, results were promising due to the fact that they handle a higher number of variables and achieve better results than traditional methods [2].

Memetic algorithms are the descendants of genetic algorithms [5] as they introduce several concepts and modifica-

tions in order to obtain nearly optimal results. Standard GAs suffered from several limitations, such as the tendency of getting stuck into local minima and slow convergence speed. In order to tackle these challenges, MAs exploit the concept of *meme*, where the best individuals of a certain population share beneficial information about the fittest solutions, allowing even the worst individuals to follow good paths in the search space [6].

However, the tendency of all individuals to reach the same solution lead to local minima and premature convergence problems. In order to avoid premature convergence and enhance global search, memetic algorithms like Particle Swarm Optimization (PSO) [4], [7] introduced the concept of *chaotic movement*. These MAs tackle the premature convergence problem by adding a chaotic component to individual evolution, so the way in which the worst individuals behave is determined by both the best ones and a certain random variable.

Finally, additional information about the search space is used by MAs in order to achieve a faster convergence and greater quality of solutions [8]. This is achieved by imposing some restrictions or modifications to the initial solutions, consequently they are not generated in a complete random process but by a pseudorandom procedure, which allows to achieve better solutions.

### III. THEORETICAL FOUNDATION

#### A. Digital FIR filter design

FIR digital filters are defined by their transfer function  $H(z)$ , which only features the coefficients of the numerator  $b_n$  due to the lack of poles, consequently all denominator coefficients are set to one:

$$H(z) = \sum_{n=0}^{N-1} b_n z^{-n} \quad (1)$$

Coefficients  $b_n$  in Eq. (1) are the coefficients of the numerator and determine the module and phase of each zero in the  $Z$ -plane and  $N$  is the number of coefficients, representing the *order* of the digital filter.

The zeros of the transfer function determine the position and magnitude of the minima observed in the module of the frequency response function. The phase of the zeros determines the normalized frequency in which minima are located and the module determines the magnitude.

#### B. Digital IIR filter design

Contrary to FIR filters, IIR filters are defined by a transfer function  $H(z)$  given by Eq. (2), which includes both numerator and denominator coefficients. The first ones determine the zeros while the second ones specify the location of poles.

$$H(z) = \frac{\sum_{n=0}^{N-1} b_n z^{-n}}{1 + \sum_{n=0}^{N-1} a_n z^{-n}} \quad (2)$$

In this transfer function, the coefficients of the denominator,  $a_n$ , determine the phase and module of the poles in the  $Z$ -plane which represent the maxima in the module of the transfer function.

When designing IIR digital filters, stability must be ensured by limiting the module of the poles by one. As seen in Eq. (2), for the same filter order, IIR digital filters need twice as much coefficients as FIR digital filter, which implies an increase in the computational load of the algorithm (notice that, for simplicity, we have chosen the same order in numerator and denominator; in the general case, orders could be different). However, for the same filter order, IIR filters obtain better frequency response than FIR filters.

## IV. DESIGN METHODOLOGY

### A. Fitness function

Shuffled Frog-Leaping Algorithms are based on the behaviour of a pond full of frogs looking for food, each one positioned over a rock whose position determines a solution to the problem. In order to determine the amount of food that a frog finds in a position, the fitness function given by

$$f(\omega) = \frac{1}{1 + J} \quad (3)$$

where  $J$  is the cost function, determined by the mean square error (MSE) between the desired filter and the current solution. The MSE is calculated using

$$MSE = \frac{1}{n} \sum_{\omega=0}^{n-1} [h_i(\omega) - h(\omega)]^2 \quad (4)$$

where  $n$  is the number of samples of the obtained filter,  $h_i(\omega)$  represents the amplitude of the desired filter and  $h(\omega)$  is the amplitude of the obtained filter.

Higher fitness values in Eq. (3) correspond to lower MSE values between the desired filter and the obtained one, so the quality of a solution is proportional to its fitness value. Frogs finding more food are the ones who represent better solutions to the problem, in this case, filters whose response is closer to the ideal one. The best frogs act as leaders for the rest of the population, conditioning the direction in which the other frogs leap.

### B. Modeling the solutions

Each solution from a particular iteration is made up by an array which contains all the filter coefficients, i.e.

$$w = [a_0, a_1, \dots, a_n, b_0, b_1, \dots, b_m] \quad (5)$$

This equation particularly determines the position of a frog when designing IIR filters as two types of coefficients are used. When FIR filters are being designed, only the  $b_m$  coefficients are used as the  $a_n$  coefficients have zero value, with the exception of  $a_0$ , whose value is fixed and set to one:

$$w = [b_0, b_1, \dots, b_m] \quad (6)$$

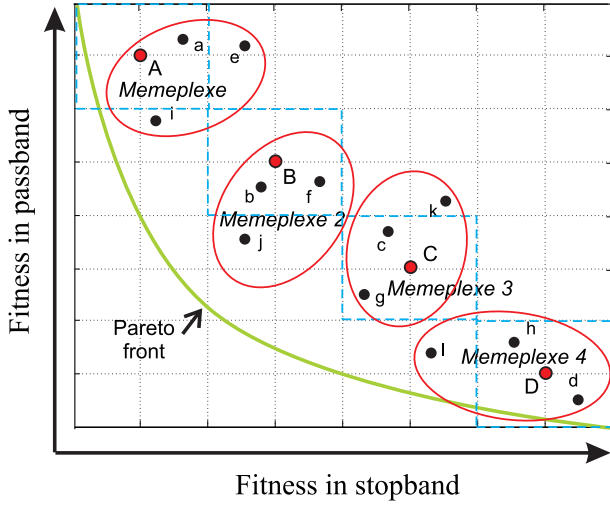


Fig. 1. Memeplex generation and location of each frog in the fitness plane.

### C. Memeplex generation

In order to enhance communication and information interchange between frogs, groups of memetic individuals –or *memeplexes*– are created.

Each memeplex should contain the same amount of frogs, so the total population  $N$  must be a multiple of the number of memeplexes  $M$ . These memeplexes are built regarding the characteristics of each frog, so frogs with similar characteristics are grouped together [9]. All frogs are represented in the plane, as shown in figure 1, where each axis represents the cost function of the resulting filter in both passband and stopband. The centroid of each cluster is positioned as shown in figure 1. This way, solutions with similar fitness values are grouped together in the same memeplex. This technique allows a better evolution of each memeplex as mentioned in [9].

### D. Adaptation to multiobjective problems

In this paper a multiobjective problem is tackled, as the intention is to minimize the MSE value of the obtained filter with respect to the ideal one, in both passband and stopband, so two objectives arise.

As two functions are used to evaluate the fitness for each frog, determination of the best and worst frogs is not an obvious task. In this algorithm, the module of the positioning vector of each frog in the fitness plane determines the best and the worst one within each memeplex. The following equation is used to calculate the module, so the one with the highest module value would be considered as the best frog (the analogous criteria is applied to the worst frog):

$$\text{module} = \sqrt{(\text{origin}_x - \text{fitness}_x)^2 + (\text{origin}_y - \text{fitness}_y)^2}$$

where subindex  $x$  refers to the stopband fitness axis, and the  $y$  subindex refers to the passband fitness axis.

### E. Leaping process

Each iteration,  $N$  of the worst frogs would leap towards a new position searching for a better solution to the problem [10]. First, each frog leaps according to Eq. (7), then, the

new position is evaluated against the previous position. If an improvement is achieved, the new position is saved:

$$x'_w = x_w + C \cdot \text{rand}(0, 1)[x_b - x_w] \quad (7)$$

where  $x_b$  is the position of the best frog of the leaping frog memeplex, and  $x_w$  is the position of the leaping frog. If the new position does not match requirements, a new leap is performed based on the global best frog, according to

$$x'_w = x_w + C \cdot \text{rand}(0, 1)[x_g - x_w] \quad (8)$$

where  $x_g$  is the position of the global best frog. In both Eqs. (7) and (8), a uniform random variable is introduced, denoted as  $C \cdot \text{rand}(0, 1)$ , so each leap is scaled by a random number from a uniform distribution in the interval  $[0, 1]$  multiplied by the factor  $C$ . This factor decreases linearly from its maximum in the first iteration towards its minimum value in last iteration. This technique allows horizontal (explorative) search at the beginning of the process and a more exploitative search during last iterations [11].

The above allows the algorithm to look for solutions in a wider area of the search space at the beginning, and to fine-tune solutions in the last iterations enhancing local search, so small changes are applied to each frog during the latest iterations [11]. Thirdly, if any improvement is made after applying both leaps, a new random frog is generated, which substitutes the previous worst frog as mentioned in [9].

### F. Diversity control

The lack of diversity of solutions found during the optimization process can lead to local minima due to premature convergence. To ensure that diversity is maintained along iterations a new entropy-based method is here proposed.

By calculating the Shannon entropy of the vector containing the distances from one frog to the others following Eq. (9), diversity can be quantified, so that solutions that get a lower entropy value are usually better distributed (with higher diversity) in the solutions space. In contrast, those sets of potential solutions with a higher entropy value have a more evenly distributed vector of distances, thus lowering the diversity of the population [12]. Entropy is calculated as

$$H(x) = - \sum_{i=0}^{n-1} p(x_i) \log_2 p(x_i) \quad (9)$$

where  $n$  is the length of the distances vector and  $x_i$  are the values of the calculated distances.

To maintain a higher diversity among the first and the second best frogs, the one with a lower entropy value is chosen as the fittest.

## V. RESULTS

This section shows the results of the numerical simulations carried out with Matlab® and the performance of the proposed filter design method. First, the obtained IIR filter is compared with the one obtained using the BZT method with elliptic approximation. Then, properties of our designed FIR filters

are compared to filters obtained using the windowing method. Finally, performance of the proposed SFLA is compared to the results reported in [13].

Regarding FIR and IIR design, different filter orders have been used in order to test the evolution of the customized SFLA when parameters are modified. In addition, before any comparison is made, parameter tuning has been carried out in order to obtain the highest fitness value. Optimal parameter values are next shown in Table I.

TABLE I  
BEST EMPIRICAL PARAMETER VALUES.

Parameter	Value
Max. iterations	500
Number of leaps	8
Number of memplexes	5
Total population	40

### A. IIR filter design

When testing the proposed algorithm, passband and stop-band ripples are measured, then an elliptic filter is generated imposing the same values of the mentioned metrics. Finally, a fitness value is calculated for both filters, comparing them to an ideal low-pass filter (LPF), setting  $\omega = 0.25$  as normalized cut-off frequency. Numerical results are shown in Table II, where all ripple values are given in dB.

TABLE II  
IIR DESIGN RESULTS FOR PROPOSED SFLA MEMETIC ALGORITHM AND BZT WITH ELLIPTIC APPROXIMATION.

SFLA order	Elliptic order	Passband ripple	Stopband ripple	SFLA fitness	Elliptic fitness
5	2	0.708	14.095	0.9783	0.9750
10	3	0.536	21.072	0.9918	0.9904
15	4	0.73	22.2	0.9958	0.9931
20	5	0.872	29.02	0.9971	0.9947
25	4	0.805	23.267	0.9979	0.9929

As shown in Table II, fitness values corresponding to the filters generated by the proposed MA are greater than those using the BZT with the elliptic approximation. However, when using the BZT method, a lower filter order is required and a narrower transition band is obtained. The reason for the elliptic method to obtain a lower fitness value is a constant stopband and passband ripple, as the filters obtained by the proposed SFLA present a decreasing ripple in both bands as shown in Figure 2, which is a desired feature in certain applications.

Figure 2 shows the frequency response of the proposed SFLA filter in blue and the elliptic filter in red, proving that the mean ripple in both bands obtained with the proposed method is better. Further testing was carried out using high-pass, band-pass and stop-band filters, obtaining similar results as the ones achieved in the low-pass case.

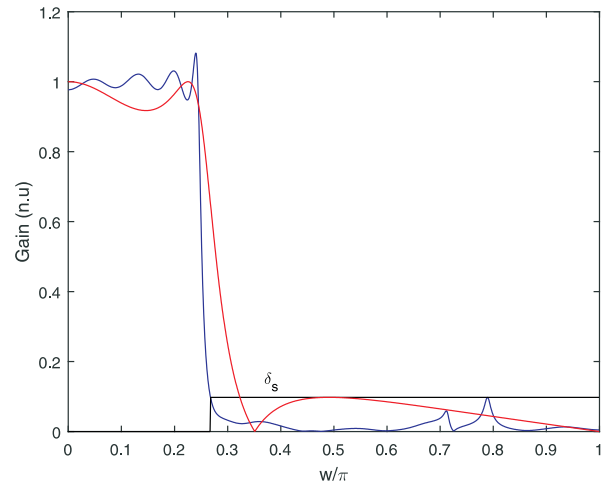


Fig. 2. LPF obtained by the proposed SFLA memetic algorithm (blue) and BZT with elliptic approximation (red), for order 25.

### B. FIR filter design

For more detailed results regarding performance, FIR LPFs were also generated. To calculate the cut-off frequency  $\omega_p$ , the passband ripple was established at 1 dB, while ripple in the stopband is set by the filter with the highest value. Table III shows the fitness values obtained for different filter orders with both the proposed SFLA and the windowing method (with a Blackman-Tuckey window).

TABLE III  
FIR DESIGN RESULTS FOR SFLA MA AND WINDOWING METHOD (BLACKMAN-TUCKEY).

Filter order	Stopband attenuation (dB)	SFLA cut-off frequency	Windowed cut-off frequency	SFLA fitness	Windowed fitness
5	15.488	0.157	0.181	0.9702	0.9568
10	20.258	0.202	0.109	0.9829	0.9568
15	17.278	0.22	0.152	0.9876	0.9754
20	16	0.232	0.203	0.9903	0.9834

Results in Table III show that fitness values obtained with SFLA are far superior than the ones for the windowing method. However, as the filter order increases, fitness values for both methods tend to match. For further testing, the transition bandwidth was also measured for both methods, showing up the superiority of the proposed method as much lower values were achieved.

TABLE IV  
TRANSITION BANDWIDTH FOR PROPOSED SFLA AND WINDOWING METHODS.

Order	Proposed SFLA	Windowing
5	0.224	0.329
10	0.143	0.263
15	0.09	0.17
20	0.062	0.099

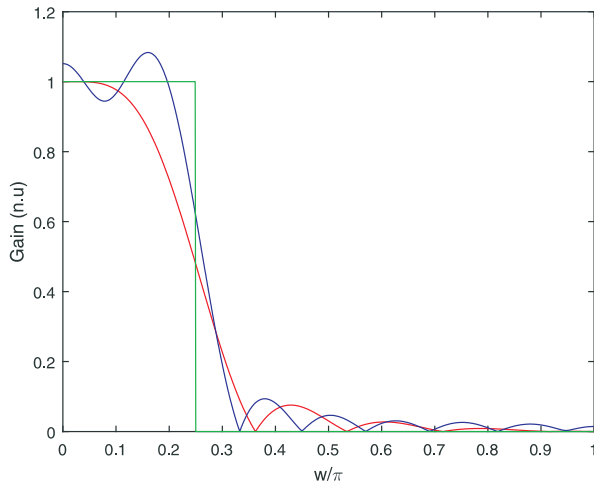


Fig. 3. FIR filter comparison between: ideal one (green), proposed SFLA (blue) and windowing (red).

### C. Comparison with other optimization algorithms

In order to compare the obtained results with the ones presented by Upadhyaya et al. in [13], a 5th-order LPF filter was approximated by a 4th-order filter using the proposed memetic algorithm. Eq. (10) represents the frequency response of the objective filter.

$$H(z) = \frac{0.1084 + 0.5419z^{-1} + 1.0837z^{-2} + 1.0837z^{-3} + 0.5419z^{-4} + 0.1084z^{-5}}{1 + 0.9853z^{-1} + 0.9738z^{-2} + 0.3864z^{-3} + 0.1112z^{-4} + 0.0113z^{-5}} \quad (10)$$

For both algorithms, five independent runs were carried out, using the MSE value as indicator. The MSE values for the proposed SFLA and the three bio-inspired methods compared –differential evolution (DE), particle swarm optimization (PSO), and regularized global optimization (RGA)– are shown in Table V.

TABLE V  
MINIMUM SQUARE ERROR (MSE) VALUES FOR SEVERAL COMPARED ALGORITHMS.

Execution	DE	PSO	RGA	Proposed SFLA
1	6.88E-04	0.0277	0.0356	7.249E-4
2	14.0E-04	0.0103	0.0507	5.807E-4
3	22.0E-04	0.0068	0.0991	7.4292E-4
4	9.72E-04	0.0177	0.0307	10.001E-4
5	16.0E-04	0.0035	0.0556	6.3074E-4

As shown in Table V, the proposed SFLA MA outperforms all the three algorithms compared, in terms of mean squared error (MSE).

## VI. CONCLUSIONS

In this paper we have presented a metaheuristic optimization method based on the standard SFLA with several modifications in order to improve filter design performance of the resulting memetic algorithm. The main aim of the proposed method is the efficient design of digital filters (both FIR and IIR) using two objective functions.

First, a memplex generation method, based on the objective function plane, establishes the center of each memetic subgroup in order to improve the convergence speed. Additionally, a mutation mechanism is introduced on the best solutions that come into play when stagnation into local optima is detected. This way, convergence is maintained and quality of final solution improves. The proposed algorithm has been adapted to multiobjective problems in order to control which characteristics of the solutions are important in the optimization process.

Experimental results are promising as an improvement has been made with regards to solution quality and adaptability to different problems. However, some drawbacks raised in relation to computational load as the proposed SFLA takes more time than traditional methods in order to reach the final solution. Consequently, a trade-off between solution quality and running time must be established by properly adjusting parameters. This situation was observed for IIR design when BZT was used for comparisons. On the other hand, results with our SFLA were better in FIR design whichever method was used for comparison (traditional or bio-inspired).

## REFERENCES

- [1] W.-K. Chen, *Passive, Active, and Digital Filters*, ser. The Circuits and Filters Handbook, 3rd Edition. CRC Press, June 2009.
- [2] N. Krasnogor, A. Aragón, and J. Pacheco, *Memetic Algorithms*. Springer Science, Springer, 2006, vol. 16, ch. 12, pp. 225–248.
- [3] T. Back, U. Hammel, and H. Schwefel, “Evolutionary computation: Comments on the history and current state,” *Transactions on Evolutionary Computing*, vol. 1, no. 1, pp. 3–17, 1997.
- [4] A. Aggarwal, T. Rawat, and D. Upadhyay, “Design of optimal digital fir filters using evolutionary and swarm optimization techniques,” *AEU Int. J. of Electronics and Communications*, vol. 38, no. 2, pp. 129–154, 2004.
- [5] P. Moscato and C. Cotta, “Una introducción a los algoritmos meméticos,” *Revista Iberoamericana de Inteligencia Artificial*, vol. 7, no. 19, pp. 131–148, 2003.
- [6] M. Dorigo and G. DiCaro, “Ant colony optimization: a new meta-heuristic,” in *Proc. of the 1999 Congress on Evolutionary Computation*, 1999.
- [7] P. Upadhyaya, R. Kar, D. Mandal, and S. Ghoshal, “Craziness-based particle swarm optimization algorithm for iir system identification problem,” *Journal of Electronics and Communications*, vol. 687, no. 5, pp. 369–378, 2014.
- [8] D. Sidhu, J. Dhillon, and D. Kaur, “Design of digital iir filter with conflicting objectives using hybrid gravitational search algorithm,” *Computation Intelligence Magazine*, vol. 2015, no. 1, pp. 1–16, 2015.
- [9] J. Luo, T. Yang, Q. Liu, X. Li, M. Chen, and K. Gao, “A new hybrid memetic multi-objective optimization algorithm for multi-objective optimization,” *Information Sciences*, vol. 448–449, pp. 164–186, 2018.
- [10] M. Eusuff, K. Lansey, and F. Pasha, “Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization,” *Engineering optimization*, vol. 38, no. 2, pp. 129–154, 2004.
- [11] M. Schonlau, M. Welch, and D. Jones, “Global versus local search in constrained optimization of computer models,” *New Developments and Applications in Experimental Design*, vol. 34, pp. 11–25, 1998.
- [12] M. Laumanns, L. Thiele, K. Deb, and E. Zitzler, “Combining convergence and diversity in evolutionary multiobjective optimization,” *Evolutionary Computation*, vol. 10, no. 3, pp. 263–282, 2002.
- [13] P. Upadhyaya, R. Kar, D. Mandal, and S. Ghoshal, “A new design method based on firefly algorithm for iir system identification problem,” *Journal of King Saud University*, vol. 28, no. 2, pp. 174–198, 2016.