Abstract—Digital filters play a key role in the field of digital signal processing. This paper presents a linear phase digital low pass finite impulse response (FIR) filter design using particle swarm optimization (PSO) and attractive and repulsive particle swarm optimization (ARPSO) and illustrates the superiority of the ARPSO method over PSO. ARPSO uses a diversity measure to control the swarm and trying to overcome the problem of premature convergence. The considered fitness is able to control the ripples in both bands separately. A comparison of simulation results demonstrates the performance of PSO and ARPSO in designing digital low pass FIR filters.

Index Terms—Digital filters design, Low pass FIR filters, PSO, ARPSO, Diversity.

I. INTRODUCTION

Digital signal processing (DSP) systems are the backbone of modern age technology. In the DSP there are two types of systems. The first system is used to perform signal filtering in time domain and known as digital filter. The second system gives signal representation in frequency domain and known as spectrum analyzer [1]. Digital filter is the indispensable part of DSP. Finite impulse response (FIR) and infinite impulse response (IIR) are the two major classes of the digital filters. FIR filters or non-recursive filters are those for which the output of the filter depends only on the present input. FIR filters are widely used due to its advantages like it is inherently stable since the poles lie within the unit circle and can be designed as linear phase filters, making them a better choice in phase sensitive application [2].

There are many different techniques available for the design of digital filters, such as window methods and frequency sampling methods and Parks-McClellan equiripple algorithm [3, 4]. In these methods, designer always has to compromise on one or more of the design specifications.

The intelligent optimization techniques have been successfully implemented in the design of digital filters and provide better parameter control as well as better approximate the ideal filter. Genetic algorithm [5], Simulated annealing [6], Tabu search [7], Differential evolution [8] and artificial bee colony algorithm [9] are some intelligent optimization techniques, which have proved their capability of designing digital filters.

The particle swarm optimization (PSO) is population based intelligent optimization technique that has proven to be effective in the design of digital FIR filters. A number of modifications have been successfully implemented in digital FIR filter design [10-15]. In this paper we propose a new variant of PSO, a diversity-guided particle swarm optimizer known as attractive and repulsive PSO (ARPSO), for the design of digital low pass FIR (LPFIR) filter.

DOI: 01.IJCOM.5.1.2
© Association of Computer Electronics and Electrical Engineers, 2014
II. ISSUES OF FIR FILTER DESIGN

The transfer function of FIR digital filter is given by

\[ H(z) = \sum_{n=0}^{N} h(n)z^{-n}, n = 0, 1, \ldots, N \]  

(1)

And the frequency response of FIR digital filter is given by

\[ H(e^{j\omega}) = \sum_{n=0}^{N} h(n)e^{-j\omega n} \]  

(2)

In this paper we design an even order, odd length and even symmetry low pass FIR filter. For this the symmetry condition is given as

\[ h(n) = h(N + 2 - n) \]  

(3)

Where N is the order of the filter and from (3) it is clear that the number of coefficients is to be optimized \((N/2+1)\). The main objective of FIR filter design is to find the filter coefficients in optimized way that results in optimum filter. The filter is optimum, when maximum weighted error is minimized. In the PM algorithm [16], an approximate error function is defined by

\[ E(\omega) = \bar{G}(\omega)|H \bar{d}(e^{j\omega}) - H_i(e^{j\omega})| \]  

(4)

Where \(H \bar{d}(e^{j\omega})\) and \(H_i(e^{j\omega})\) are the frequency responses of the designed approximate filter and ideal filter respectively. The weighting function, \(\bar{G}(\omega)\), is used to provide the approximation error differently in different frequency bands. For an ideal LP filter, \(H_i(e^{j\omega})\) is defined as

\[ H_i(e^{j\omega}) = 1 \text{ for } 0 \leq \omega \leq \omega_c \]  

(5)

0 otherwise

where \(\omega_c\) is the cutoff frequency. In this paper we use the fitness function [17] is given as

\[ fitness = \max_{\omega \in \omega_{ap}} \left( |E(\omega)| - \delta_p \right) + \max_{\omega \in \omega_{azq}} \left( |E(\omega)| - \delta_q \right) \]  

(6)

This fitness function provides better control over the ripples in pass band and stop band separately. Where \(\omega_p\) and \(\omega_q\) are the pass band and stop band normalized cutoff frequencies.

III. INTELLIGENT OPTIMIZATION TECHNIQUES

A. Particle Swarm Optimization (PSO)

PSO is inspired by the observation of social behavior of bird flocking and fish schooling. This powerful global optimization technique was first found by Kennedy (a social psychologist) and Eberhart (an electrical engineer) in 1995 [18]. PSO is simple, fast, requires less storage and can be coded in few lines. In PSO every particle remembers its best solution \((X_{i}^{best})\) as well as the group’s best solution \((X^{best})\). It means that PSO have good memory. The PSO is worked on the concept of “constructive cooperation” between particles, so that it is easily able to solve multidimensional optimization problems. In PSO global optimum is achieved by an iterative procedure. The PSO technique is based on the five basic principles of the swarm intelligence [19]. These are, Proximity, i.e., the swarm must be able to perform simple space and time computations. Quality, i.e., the swarm should be able to respond to quality factors in the environment. Diverse response, i.e., the swarm should not commit its activities along excessively narrow channels. Stability, i.e., the swarm should not change its behavior every time the environment changes. Adaptability, i.e., the swarm must be able to change its behavior, when the computation cost is affordable.

PSO starts with a population of random particles (potential solution) in a D-dimension space. A position ‘X’ and velocity ‘V’ are associated with each particle. The position and velocity of the \(i^{th}\) particle are given as

\[ X_i = (X_{i1}, X_{i2}, \ldots, X_{iN}) \]  

(7)

\[ V_i = (V_{i1}, V_{i2}, \ldots, V_{iN}) \]  

(8)

The velocity and position are updated according to the formula given as

\[ V_{i}^{n+1} = \omega V_{i}^{n} + C_1 \cdot \text{rand}_1 \cdot (X_{i}^{best} - X_i) + C_2 \cdot \text{rand}_2 \cdot (X^{best} - X_i) \]  

(9)

\[ X_i^{n+1} = X_i + V_i^{n+1} \]  

(10)

Where \(X_{i}^{best}\) and \(X^{best}\) is the individual best and global best positions respectively, \(X_i\) is the current position of the \(i^{th}\) particle, \(n+1\) and \(n\) denote the current and the previous iterations, \(\text{rand}_1\) and \(\text{rand}_2\) are random numbers in the range \([0, 1]\). These random numbers are update every time they occur. \(C_1\) and \(C_2\) are the two positive constants, called cognitive and social acceleration factors respectively and \(\omega^{n}\) is the inertia weight in the \(n^{th}\) iteration. A linearly damped inertia weight is preferred for better convergence [20].
The PSO algorithm for filter designing is as follows:
1) Define the filter specifications, fitness function, and population size and set the boundaries, i.e. maximum and minimum value of coefficient.
2) Initialize a population array of particles with random positions and velocities in the problem space.
3) For each particle, Compare particle’s fitness evaluation with its \( X_i^{\text{best}} \) and \( X^{\text{best}} \). If fitness\( (x) \) better than fitness \( (X_i^{\text{best}}) \) then \( X_i^{\text{best}} = X_i \) (current value) and if fitness\( (x) \) better than fitness\( (X^{\text{best}}) \) then \( X^{\text{best}} = X_i \).
4) Update the velocity according to (9) and Move each particle to new position according to (10).
5) Recalculate the fitness value and compare.
6) Loop from 2-5 until stopping criterion is satisfied.
7) Output is the coefficient of the desired filter \((\text{N}/2+1)\).

B. Attractive and Repulsive PSO (ARPSO)

In order to improve the performance of PSO, a novel algorithm called attractive and repulsive PSO (ARPSO) is proposed in [21] trying to overcome the problem of premature convergence. In this paper we propose this algorithm in the design of linear phase digital low pass FIR filter. ARPSO uses a diversity measure to control the swarm. The result is an algorithm that alternates between phases of attraction and repulsion. In the attraction phase the swarm is contracting, and consequently the diversity decreases. When the diversity drops below a lower bound, \( d_{\text{low}} \), it indicates the repulsion phase, in which the swarm expands. Finally, when a diversity of \( d_{\text{high}} \) is reached, it switches back to the attraction phase. In order to set direction for the swarm use the following function

\[
\text{if} \ (\text{dir} > 0 \ \& \& \ \text{diversity} < d_{\text{low}}) \ \text{dir} = -1 \quad (11)
\]

\[
\text{if} \ (\text{dir} < 0 \ \& \& \ \text{diversity} > d_{\text{high}}) \ \text{dir} = 1
\]

set Direction determines which phase the algorithm is currently in, simply by setting a sign-variable, \( \text{dir} \), either to 1 or \(-1\) depending on the diversity. The diversity of the swarm is set according to the diversity measure

\[
\text{diversity}(S) = \frac{1}{|s|.|L|} \sum_{i=1}^{|s|} \sum_{j=1}^{N} (p_{ij} - \bar{p}_j)^2 \quad (12)
\]

\( |s| \) is the swarm size, \(|L|\) is the length of longest the diagonal in the search space, \( N \) is the dimension of the problem, \( p_{ij} \) is the \( j \)th value of the \( i \)th particle and \( p_i \) is the \( j \)th value of the average point \( \bar{p} \).

Finally, the velocity-update formula, (9), is changed by multiplying the sign-variable \( \text{dir} \) to the two last terms in it. This decides directly whether the particles attract or repel each other and given as

\[
V_i^{n+1} = \omega V_i^n + \text{dir} \left( C_1 \ast \text{rand}_1 \ast (X_i^{\text{best}} - X_i) + C_2 \ast \text{rand}_2 \ast (X^{\text{best}} - X_i) \right) \quad (13)
\]

The ARPSO algorithm for the filter designing is as follows:
1) Define the filter specifications, fitness function, and population size and set the boundaries, i.e. maximum and minimum value of coefficient.
2) Initialize a population array of particles with random positions and velocities in the problem space.
3) Evaluate the fitness function value according to (6)
4) Set Direction according to (11) by calculating the diversity using (12)
5) Update velocity according to (13) and update the position by adding this velocity (10)
6) Reevaluate the fitness function value and compare with its \( X_i^{\text{best}} \) and \( X^{\text{best}} \)
7) Loop from 2-6 until stopping criterion is satisfied.
8) Output is the coefficient of the desired filter \((\text{N}/2+1)\).

IV. DESIGN EXAMPLES AND DISCUSSION

In this section, we demonstrate the use of the PSO and ARPSO to design a 20 order low pass FIR filter. All the simulation results shown in this paper have been made on MATLAB 7.10 with 1GB RAM and using dual-core processor. The specifications of the filter to be designed using these algorithms are: \( \delta_p = 0.1 \), \( \delta_s = 0.01 \), \( \omega_p = 0.45 \) and \( \omega_s = 0.55 \). Table 1 shows the parameters and its corresponding value consider during this work.

Fig. 1 shows the frequency response in dB of LP FIR designed by ARPSO. The PM and PSO results for the same specifications are also shown on the same figure. The PM algorithm gives \( \delta_p = 0.066 \) and \( \delta_s = 0.066 \) (not the desired values) which indicate that PM algorithm does not allow complete control on the pass band.
Table 1: PSO and ARPSO Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>PSO</th>
<th>ARPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Swarm Size</td>
<td>75</td>
<td>55</td>
</tr>
<tr>
<td>No. of Iteration</td>
<td>500</td>
<td>400</td>
</tr>
<tr>
<td>$C_1$</td>
<td>2.05</td>
<td>2</td>
</tr>
<tr>
<td>$C_2$</td>
<td>2.05</td>
<td>2</td>
</tr>
<tr>
<td>$w_{max}$</td>
<td>0.95</td>
<td>0.95</td>
</tr>
<tr>
<td>$w_{min}$</td>
<td>0.40</td>
<td>0.40</td>
</tr>
<tr>
<td>$V_{max}$</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>$d_{low}$</td>
<td>-</td>
<td>0.25</td>
</tr>
<tr>
<td>$d_{high}$</td>
<td>-</td>
<td>1.5</td>
</tr>
</tbody>
</table>

Fig. 1: Frequency Response in dB of LP FIR

ripples and stop band ripples and PSO gives $\delta_p=0.1041$ and $\delta_s=0.0189$ (close to the desired values) which indicate that the PSO algorithm allow complete control on these values. Whereas ARPSO gives $\delta_p=0.1064$ and $\delta_s=0.0179$ (very close to the desired values) which indicate that the ARPSO algorithm also allow complete control on these values. Table 2 shows the optimized coefficients of LP FIR in PM, PSO and ARPSO algorithms. Hence in the case of ARPSO result quality is improved as indicated in Table 3. PSO gives 30.90dB stop band attenuation. Whereas ARPSO gives 32.21dB stop band attenuation which indicate that the ARPSO gives better stop band attenuation as compare to the PSO. Fig. 2 shows the normalized frequency response. Fig. 3 and fig. 4 shows the normalized pass band ripples and normalized stop band ripples respectively.

A. Convergence of PSO and ARPSO

Fig 5-6 shows the convergence behavior of PSO and ARPSO respectively. These plots provide the error fitness value of the algorithms with number of iterations. PSO converges to the minimum error fitness value of 0.9392 in 55.39sec. ARPSO converges fast as compare to PSO and reach to the minimum error fitness value of 0.0829 in 38.55sec. Hence the convergence of ARPSO is better than PSO in finding the desired filter coefficients.

V. CONCLUSION

In this paper, the application of PSO and ARPSO in the design of linear phase low pass FIR filter has been carried out. It is found that PSO and ARPSO are ease to implement in both the context of coding and parameter selection. Comparison of results of PM, PSO and ARPSO has been made. It is analyzed that the
Fig. 2: Normalized Frequency Response of LP FIR

Fig. 3: Normalized Pass band ripples of LP FIR

Fig. 4: Normalized Stop band ripples of LP FIR
Fig. 5: Convergence of PSO

Fig. 6: Convergence of ARPSO

The performance of ARPSO is better than PSO in the context of stop band attenuation. ARPSO is a diversity guided optimizer and overcome the problem of premature convergence. The convergence behavior of the algorithms is also compared and it is examined that ARPSO quickly converges with a very low error fitness value.

<table>
<thead>
<tr>
<th>h(n)</th>
<th>PM</th>
<th>PSO</th>
<th>ARPSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>h(1) = h(21)</td>
<td>0.000016462026203</td>
<td>0.022587593788748</td>
<td>0.027369466516418</td>
</tr>
<tr>
<td>h(2) = h(20)</td>
<td>0.048051046361716</td>
<td>0.03423459454764</td>
<td>0.031584806423413</td>
</tr>
<tr>
<td>h(3) = h(19)</td>
<td>-0.000023455414888</td>
<td>-0.017086577157864</td>
<td>-0.009609596370614</td>
</tr>
<tr>
<td>h(4) = h(18)</td>
<td>-0.036911143268907</td>
<td>-0.046131029539532</td>
<td>-0.042136759394785</td>
</tr>
<tr>
<td>h(5) = h(17)</td>
<td>-0.000014804257488</td>
<td>0.000229126085020</td>
<td>0.008838062445680</td>
</tr>
<tr>
<td>h(6) = h(16)</td>
<td>0.057262893095235</td>
<td>0.058644950198106</td>
<td>0.048999974886322</td>
</tr>
<tr>
<td>h(7) = h(15)</td>
<td>0.00000677226645</td>
<td>-0.006546001812412</td>
<td>-0.020752750732276</td>
</tr>
<tr>
<td>h(8) = h(14)</td>
<td>-0.102172983403192</td>
<td>-0.097122365891821</td>
<td>-0.117280201862331</td>
</tr>
<tr>
<td>h(9) = h(13)</td>
<td>0.000011850968750</td>
<td>0.013760466527049</td>
<td>-0.002127035195637</td>
</tr>
<tr>
<td>h(10) = h(12)</td>
<td>0.316962289494363</td>
<td>0.322156879042563</td>
<td>0.314190832325194</td>
</tr>
<tr>
<td>h(11)</td>
<td>0.500018538901555</td>
<td>0.500748972249363</td>
<td>0.485623316789776</td>
</tr>
</tbody>
</table>
REFERENCES