Design of Digital Finite Impulse Response Filter with DE3 Optimization Technique

Alamdeep Singh  
Student  
RIET Phagwra

Prof. Manit Kapoor  
Assistant Professor  
RIET Phagwra

Dr. Naveen Dhillon  
Professor  
RIET Phagwra

Abstract

The research on optimal design of Finite impulse response (FIR) filter based on various optimization techniques, including evolutionary algorithm (EAs) have gained much attention in recent years. Genetic algorithm is a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover. The New Differential evolution algorithm based on reserve genes (Eclectic Differential Evolution) have been used here for the design of finite impulse response filters. This algorithm is applied in order to obtain the actual filter response as close as possible to the ideal response. In this method, the new vectors can be produced by the combination of genes of selected chromosomes. These new vectors as the individuals are evolved with other individuals in the population and also increase the diversity of population.

Keywords: Digital Filters, FIR filters, Optimization techniques, Differential Evolutions Algorithm

I. INTRODUCTION

A digital filter is a system that uses discrete time signal as input and produce a discrete time output signal for the purpose of achieving a filtering objective [1]. The main advantage of digital filters lies in its easy alternation of Characteristics changing the discrete values stored in the registers. The digital filters are classified into two categories as infinite impulse response (IIR) filters and finite impulse response (FIR) filters. Among the two filters, because of their inherent stable response due to absence of poles in the transfer function and easily achievable linear phase property, FIR filters are the most widely used in almost all modern digital applications such as image processing, speech processing, channel equalization, signal enhancement, digital audio. Linear phase response, required in applications such as data communications and crossover filters, can be achieved using symmetric coefficient FIR filters [1,2]. A digital filter is a device that performs mathematical operations on a sampled, discrete time signal to reduce or enhance certain aspects of that signal. A digital filter system usually consists of an analog-to-digital converter to sample the input signal, followed by a microprocessor and some peripheral components such as memory to store data and filter coefficients etc. Finally, digital-to-analog converter is used to complete the output stage.

1) It operates on the digital samples of the signals.
2) These kinds of filters are defined using linear difference equations.
3) While implementing the digital filters in hardware or software, we need adders, sub tractors, delays, etc. which is classified under digital logic components.
4) In this filter, the filter coefficients are designed to meet the desired or expected frequency response.
5) Mathematically the transfer function $H (z)$ is required to be a rational function of $z$, where the coefficients of $z$ are real to meet the stability and causality requirement.

A. There are two types of Digital Filters:

1) Infinite Impulse Response Digital Filter
2) Finite Impulse Response Digital Filter

1) Infinite Impulse Response Digital Filter:

IIR filters are digital filters with infinite impulse response. They have the feedback (a recursive part of a filter) and are known as recursive digital filters. Unlike FIR filter, the output of an IIR filter depends on both the previous $M$ inputs and the previous $N$ outputs. This feedback mechanism is inherent in any IIR structure. It is responsible for the infinite duration of the impulse response [10].
The design equation of IIR filters in Z-domain:

\[ H(z) = \frac{a_0 + a_1 z^{-1} + \cdots + a_N z^{-N}}{b_0 + b_1 z^{-1} + \cdots + b_N z^{-N}} \]

Digital IIR filter is one of the most frequently used computation tools in digital signal processing systems. In many applications, such as high-speed and low-power communication transceivers, it is routinely employed as a custom designed digital block. An IIR filter has a number of useful properties which sometimes make it preferable to an infinite impulse response (IIR) filter.

2) **Finite Impulse Response Digital Filter:**

In signal processing, a finite impulse response (FIR) filter is a filter whose impulse response (or response to any finite length input) is of finite duration, because it settles to zero in finite time. The impulse response of an Nth-order discrete-time FIR filter lasts for N + 1 samples, and then settles to zero. FIR filters can be discrete-time or continuous-time, and digital or analog. The output y(n) of a linear time invariant system is determined by convolving its input signal x with its impulse response b [10].

For a discrete-time FIR filter, the output is a weighted sum of the current and a finite number of previous values of the input. The operation is described by the following equation, which defines the output sequence y[n] in terms of its input sequence x[n]:

\[ y[n] = b_0 x[n] + b_1 x[n-1] + \cdots + b_N x[n-N] \]

where:
- x[n] is the input signal,
- y[n] is the output signal,
- \( b_i \) are the filter coefficients, also known as tap weights, that make up the impulse response,
- N is the filter order; an Nth-order filter has (N+1) terms on the right-hand side. The x[n-i] in these terms are commonly referred to as taps, based on the structure of a tapped delay line that in many implementations or block diagrams provides the delayed inputs to the multiplication operations.

3) **Properties of FIR Filter:**

Require no feedback. This means that any rounding errors are not compounded by summed iterations. The same relative error occurs in each calculation. This also makes implementation simpler.

Are inherently stable. This is due to the fact that, because there is no required feedback, all the poles are located at the origin and thus are located within the unit circle.

They can easily be designed to be linear phase by making the coefficient sequence symmetric. This property is sometimes desired for phase-sensitive applications, for example data communication, crossover filters, and mastering.

The FIR filter is non-recursive structure; finite precision arithmetic error is very small. While IIR filter is recursive structure, and parasitic oscillation may occur in the operation of IIR filter.

**B. FIR Coefficient Calculation methods**

The objective of most FIR coefficient calculation methods is to obtain values of h(n) such that the resulting filter meets the design specifications, such as amplitude frequency response and throughput requirements. The most commonly method to obtain h(n) are the window, optimal and frequency sampling method. All three can lead to linear phase FIR filters.

1) **Filter Structures**

In order to realize a digital filter, the given differential equation is broken into small equations. Then for each small equation, a structure using elementary blocks is drawn. Finally all these blocks are interconnected. The filter structures characterizing the
difference equations are represented using basic elements such as multipliers, adders and delay elements. This way of presenting the difference equations in the form of block diagram and signal flow diagram makes it easy to write an algorithm, which can be implemented in the digital computer. A digital filter structure is said to be canonic if the number of delays in the block diagram representation is equal to the order of the transfer function otherwise, it is a non-canonic structure. The different types of structures used to realize the digital filter are as follows:

1) Direct form structure
2) Cascade form structure
3) Parallel form structure

2) Choice between different Structures

The choice between structures depends on a number of factors and trade-offs which include ease of implementation that is the implied hardware or software complexity, how difficult it is to obtain the impulse response or transfer coefficients and their relative sensitivity to coefficient quantization. The direct structure is very easy to program and is efficiently implemented by most DSP chips as these have instructions tailored to transversal FIR filtering. It is the most common structure used to realize non-recursive filters and its main attraction is its simplicity, requiring only a minimum of components and uncomplicated memory accesses for data. The cascade is less sensitive to coefficients errors and quantization noise, but the coefficients require more effort to obtain and the programming is not suited to DSP chips architectures.

II. METHODOLOGY: DIFFERENTIAL EVOLUTION

Differential Evolution (DE), proposed by Storn and Price (1997), is a simple yet powerful algorithm for real parameter optimization, which uses the greedy criterion to make decision. Under the greedy criterion, a new parameters vector is accepted if and only if it reduces the value of the objective. It has been shown to perform better than the Genetic Algorithm (GA) or the Particle Swarm Optimization (PSO) over several numerical benchmarks. Differential evolution is a global optimization technique that is exceptionally simple, significantly fast and robust. The fittest of an offspring competes one-to-one with that of corresponding parent, which is different from the other evolutionary algorithms. This one-to-one competition gives rise to a faster convergence rate. Differential evolution is the real coded genetic algorithm combined with an adaptive random search using a normal random generator. It uses floating-point numbers that are more appropriate than integers for representing points in a continuous space. The success rate is the best measure for the performance of the technique, defined as the ratio of the total number of times the optimal solution is found to the total number of test runs; DE has a success rate of 100%. Differential Evolution (DE) algorithm is a heuristic approach for minimizing nonlinear and non-differentiable continuous space functions (Storm, 1997). However, the implementation of DE requires some estimate of the global minimum to be provided, which is difficult for any arbitrary function. Also, the termination with maximum number of iterations does not assure the global minima. Moreover, as the region of global minimum approaches, the convergence of the algorithm is very slow.

A. Differential Evolution Method:

One of the important aspects of any algorithm is the termination criterion. For theoretical aspects of evolutionary algorithms stopping criteria are usually not important. However, for practical applications the choice of stopping criteria can significantly influence the duration of an optimization run as well as the global minimum. DE terminates either with user defined number of generations or some error estimate. These criteria are perfectly suitable for comparing the performance of different algorithms. Differential evolution is a population based method using operators: mutation, crossover and selection. The algorithm randomly chooses a population vector of fixed size. During at the each iteration of algorithm a new population of same size is generated. It uses mutation operation as a search mechanism. This operation generates new parameter vector. During at the each iteration of algorithm new population of same size is generated. It uses mutation operation as a search mechanism. This operation generates new parameter vector by adding a weighted difference vector between two population members to a third member. In order to increase the diversity of the parameter vectors, the crossover operation produces a trial vector which is a combination of a mutant vector and a parent vector. Then the selection directs the search toward the prospective regions in the search space. In addition, the best parameter vector is evaluated for every generation in order to keep track of the progress that is made during the minimization process. The above iterative process of mutation, crossover and selection on the population will continue until a user specified stopping criterion, normally, the maximum number of generations or the maximum number of function evaluations allowed is met. The process is assumed to have converged if the difference between the best function values in the new and old population, and the distance between the new best point and the old best point are less than the specified respective tolerances. The other type of stopping criterion could be if the global minimum of the problem is known a-priori. Then DE will be terminated if the difference between the best function value in the new population and the known global minimum is less than the user defined tolerance level. Differential evolution is a population Based stochastic algorithm that exploits a population of potential solutions, individuals, to probe the search space. New individuals are generated by the combination of randomly chosen individuals from the population. Mutation constant controls the amplification of the difference between two individuals so as to avoid the stagnation of the search process.

To increase further the diversity of the mutant individuals, the resulting individuals are combined with other predetermined individuals. The target individuals, this operation is called crossover or recombination. Finally, the trial individual is accepted for the next generation if and only if it yields a reduction in the value of the error function. This is the selection operation. Proposed
methods in convert a constrained problem into an unconstrained problem. Proposed methods in convert a constrained problem into an unconstrained problem by assigning different weights to each objective. This allows a simpler minimization process but require the knowledge of the relative importance of each objective and the explicit relationship between the objectives. A study of unconstrained problems is important for the following reasons. The constrained do not have significant influence in certain design problems. Some of the powerful and robust methods of solving constrained minimization problems require the use of unconstrained minimization techniques. The study of unconstrained minimization techniques provide the basic understanding necessary for the study of constrained minimization methods. The unconstrained minimization methods can be used to solve certain complex engg. Analysis problems. For example, the displacement response (linear or nonlinear) of any structure under any specified load condition.

The three different mutation schemes suggested by Storn and Price are summarized as:

1) **Scheme DE1:**
Our first variant of DE works as follows: for each vector \( x_{i,G} \), \( i = 0, 1, 2, \text{NP}-1 \), a trial vector \( v \) is generated according to

\[
V = x_{r1,G} + F ( x_{r2,G} - x_{r3,G} )
\]  

(1)

With \( r1, r2 \) and \( r3 \) integer and mutually different and \( F > 0 \).

The integer’s \( r1, r2 \) and \( r3 \) are chosen randomly from the interval \([0, \text{NP}-1]\) and are different from the running index \( i \). \( F \) is a real and constant factor which controls the amplification of the differential variation \( (x_{r2,G} - x_{r3,G}) \).

2) **Scheme DE2:**

Basically, scheme DE2 works the same way as DE1 but generates the vector \( v \) according to equation

\[
V = x_{i,G} + \mu ( x_{\text{best},G} - x_{i,G} ) + F ( x_{r2,G} - x_{r3,G} )
\]  

(2)

Which introducing an additional control variable \( \mu \). The idea behind \( l \) is to provide a means to enhance the greediness of the scheme by incorporating the current best vector \( x_{\text{best}} \), \( G \). This feature can be useful for non-critical objective functions. The construction of \( u \) from \( v \) and \( x_i \), \( G \) as well as the decision process are identical to DE1.

3) **Scheme DE3:**

\[
(3)
\]

a) The evolution of DE3 is described by the equation:

\[(3)\text{ in which } r, r1, r2 \in [0, \text{NP}-1] \text{ are randomly chosen and they must be different from each other. Fitness of } x_{i,G}(t) \text{ is not less than the fitness of } x_{\text{best},G}(t) \text{. } F \text{ is usually set between random numbers 0 and 2.}\]

b) **Crossover and Selection**

The parent vector is mixed with the mutated vector to produce trial vector as:

\[
\text{in which } \text{rand} , \epsilon [0, 1] \text{ is also a random number, } CR \in [0, 1] , r_n \in D. \text{ The performance of the produced trial vector and its parent are compared and the better one is selected. Then the better one of the trial solution and its parent wins the competition; this provides the significant advantage of converging performance over the GA [10].}\]

**B. Design of FIR filter using GA and DE algorithm**

The transfer function of an FIR filter is given by the equation

\[
H(z)=\sum_{n=0}^{N} a_n z^{-n}
\]

where \( a_n \) represents the filter coefficients to be determined in the design process and \( N \) represents the polynomial order of the function. There are various parameters in filter design. They are the stopband and passband normalized frequencies (\( \omega_s, \omega_p \)), the passband and stopband ripple \( \delta_s \) and \( \delta_p \), the stopband attenuation and the transition width. In any filter design problem, some of the parameters are fixed while others are determined. The DE and Genetic algorithms are used in order to obtain the actual filter response as close as possible to the ideal response. This discusses the most widely used FIR filter whose length is odd and the order is even. The length of filter is \( N + 1 \) means the number of \( a_n \) is also \( N + 1 \). The individuals is represented by \( a_n \). In each iteration of algorithm, these individuals generate new offspring, which is the new set of coefficients. Fitness of each individual is calculated using the new coefficients. This fitness is used to improve the search in each iteration and result obtained after a certain number of iterations or after the error is below a certain limit is taken to be the final result. Because its coefficients are matched, the problem dimension reduces by a factor of 2. The \( N/2 + 1 \) coefficients are then flipped and concatenated to calculate the required \( N + 1 \) coefficients.
III. RESULTS AND DISCUSSIONS

Performance of a system can be checked by comparing its output characteristics with the ideal output characteristics of a standard system. We can get system performances by simulating the program through some specific software or implementing the system in hardware. Simulation platform has been used throughout this work for getting the system responses. The Differential algorithm and DE3 algorithm have been developed in MATLAB for designing Band-pass finite impulse response (FIR) filter. Simulation results have been obtained by executing the program in number of times. The phase of the designed filter is linear with frequency as symmetric FIR filter has been synthesized in this work. Therefore, magnitude response is the main characteristics of FIR filter. From magnitude response of a filter we can find out its pass band ripple, stop band attenuation, transition bandwidth etc. For an ideal low pass filter, pass band ripple is 0dB, transition from pass band to stop band is very sharp and attenuation in stop band region is maximum

<table>
<thead>
<tr>
<th>Parameters of DE3</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Population size, NP</td>
<td>15</td>
</tr>
<tr>
<td>Maximum Iterations</td>
<td>200</td>
</tr>
<tr>
<td>Scaling Factor</td>
<td>0.8</td>
</tr>
<tr>
<td>Crossover probability</td>
<td>0.5</td>
</tr>
</tbody>
</table>

evaluate the individuals. It takes the mean squared error between the frequency response of the ideal and actual filter. An ideal filter has a magnitude of 1 on the passband and a magnitude of 0 on the stopband. So the error for the fitness function is the squared difference between the magnitudes of this filter and filter designed using the evolutionary algorithms.
There are basically two main types of FIR Filter that are used for realization i.e. Structural adders (SA) and the Multiplier Adder (MA). The SA is used to add the input signal X(n), then multiplied by filter coefficient product value, i.e. along with stored value in delay element. Hence, the adder cost of SA becomes equal to order of the filter. The MA is basically used to obtain the product value of filter coefficient multiplied by input X(n), by shift, and add methodology. The number of adders used in a multiplier is used for coefficient multiplication and is dependent on the number of non zero bits that is present in the filter coefficients. Hence, the adder cost of MA mainly depends on number of SPT terms present in filter coefficients. The DE algorithm should be designed in such a way that the number of SPE terms are minimum in the computational expression. The more the number of adders the more complex the system becomes so effort must be made to simplify the design of the FIR filter. Therefore, the basic aim to design the DE algorithm is to have favorable FIR response with reduced complexity.

Table 2:

<table>
<thead>
<tr>
<th>Sr No.</th>
<th>No. of iteration</th>
<th>Best Fit values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>10</td>
<td>0.0383</td>
</tr>
<tr>
<td>2.</td>
<td>20</td>
<td>0.3803</td>
</tr>
<tr>
<td>3.</td>
<td>30</td>
<td>0.5595</td>
</tr>
<tr>
<td>4.</td>
<td>40</td>
<td>0.8543</td>
</tr>
<tr>
<td>5.</td>
<td>50</td>
<td>0.9726</td>
</tr>
<tr>
<td>6.</td>
<td>60</td>
<td>1.0001</td>
</tr>
<tr>
<td>7.</td>
<td>70</td>
<td>0.9954</td>
</tr>
</tbody>
</table>

Therefore, the various designed filters and their properties are shown in Table1 where ‘N’ is the order of the FIR filter, WL represents the word length of the filter coefficients used, Asb is the stop band attenuation of the filter, SPT is the total number of SPT terms that are present in a filter coefficient set. The MA and SA are the number of multiplier adder and structural adder, respectively, that are used to implement filter coefficient set. The MA’s evaluated after applying CSE elimination. The SA is estimated using non-zero coefficient values. Also, the TA is the total adder cost that is obtained by adding MA and SA coefficients.

The comparison between the DE1, DE2, DE3 filters is shown below:

![Properties of DE1, DE2, DE3 Filter](image)

Fig. 4: The comparison between DE1, DE2, DE3 filter.

Therefore, the order, N of DE1 filter is 30, that of DE2 is 29 and of DE3 is 26. The SPT gain% in DE3 is reduced to 21.21%, in DE2 its 30.3% and that in DE1 is 24.24%. The MA of DE1 and DE3 is 8 and of DE2

Table 3

Comparison of various optimization Techniques:

<table>
<thead>
<tr>
<th>FILTER</th>
<th>N</th>
<th>WL</th>
<th>Asb</th>
<th>SPT</th>
<th>SPT gain(%)</th>
<th>MA</th>
<th>SA</th>
<th>TA</th>
<th>TA gain(%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remez</td>
<td>30</td>
<td>10</td>
<td>-50.23</td>
<td>66</td>
<td>-</td>
<td>10</td>
<td>30</td>
<td>40</td>
<td>-</td>
</tr>
<tr>
<td>DE1</td>
<td>30</td>
<td>8</td>
<td>-51.89</td>
<td>50</td>
<td>24.242</td>
<td>8</td>
<td>23</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>DE2</td>
<td>29</td>
<td>8</td>
<td>-51.63</td>
<td>46</td>
<td>30.303</td>
<td>7</td>
<td>23</td>
<td>30</td>
<td>25</td>
</tr>
<tr>
<td>DE3</td>
<td>26</td>
<td>10</td>
<td>-50.89</td>
<td>53</td>
<td>21.21</td>
<td>8</td>
<td>19</td>
<td>27</td>
<td>32.5</td>
</tr>
</tbody>
</table>

Table 4

Comparison of DE1, DE2, DE3 AND Remez

<table>
<thead>
<tr>
<th>FILTER</th>
<th>Area</th>
<th>Area gain (%)</th>
<th>Delay</th>
<th>Delay gain (%)</th>
<th>Power</th>
<th>Power gain (%)</th>
<th>PDP</th>
<th>PDP gain (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remez</td>
<td>98,214</td>
<td>-</td>
<td>7.3</td>
<td>-</td>
<td>5.5</td>
<td>-</td>
<td>41.7</td>
<td>-</td>
</tr>
<tr>
<td>Shi</td>
<td>77,498</td>
<td>21.09</td>
<td>8.1</td>
<td>-10.2</td>
<td>4.4</td>
<td>21.2</td>
<td>36.2</td>
<td>12.8</td>
</tr>
<tr>
<td>DE1</td>
<td>77,947</td>
<td>20.52</td>
<td>7.3</td>
<td>1.3</td>
<td>4.2</td>
<td>26.5</td>
<td>30.1</td>
<td>27.5</td>
</tr>
</tbody>
</table>
is7. The TA gain is maximum in DE3 filter. The adder cost can be optimised by proper evaluation of the order of designed FIR filter So, by the proper reduction in the value of order of the filter the better cost of structural adder(SA) can be achieved.

The above table depicts the synthesis results of the designed filters. The basic difference between the Remez and DE3 filter is shown below:-

<table>
<thead>
<tr>
<th>Filter</th>
<th>Area</th>
<th>Delay</th>
<th>Power</th>
<th>PDP</th>
</tr>
</thead>
<tbody>
<tr>
<td>DE2</td>
<td>80.490</td>
<td>18.12</td>
<td>5.6</td>
<td>4.1</td>
</tr>
<tr>
<td>DE3</td>
<td>76.373</td>
<td>22.22</td>
<td>6.9</td>
<td>4.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Therefore, the delay in DE3 filter is 6.9ns and in Remez is 7.3ns. Also, the area gain is maximum in DE3 as 22%.
Therefore, the synthesis is carried out in terms of area, delay, power and PDP. DE1 has the maximum power gain of 26%. The DE3 has maximum PDP of 31.4 compared to DE1 filter. Although, the delay of DE2 and DE3 are comparable. The PDP of DE2 is minimum as compared to Remez i.e. 41.7. So, proper measures are required to be taken for the better optimization of FIR filter coefficient design.

IV. CONCLUSION

In this paper, the evolutionary algorithm that is Differential Evolution algorithm is used so as to have a filter design with low complexity, reduced hardware and reduced delay for FIR filter design. The proposed algorithm compares the various factor involved in FIR filter design in terms of area, delay, power and PDP. For the better evaluation initialization, mutation, crossover, evaluation and selection of optimal solution is carried out. The design of filters is done for different word length by using the Differential evolution. The same design is then carried by transposed direct form structure. On comparing, our results with Remez algorithm it is found that our proposed filter has the PDP gain of 28.2% that improvises the previous results. Therefore, the proposed filter parameters better optimizes the filter design.

REFERENCES