JJEE
Jordan Journal of Electrical Engineering

Volume 8, Number 2, June 2022
Pages 114-132
ISSN (Print): 2409-9600, ISSN (Online): 2409-9619

Design and Optimization of Infinite Impulse Response Full-Band Digital Differentiator Using Evolutionary and Swarm Intelligence Algorithms

Jehad Ababneh[1*], Majid Khodier[2]

[1, 2] Department of Electrical Engineering, Jordan University of Science and Technology, Irbid, Jordan
E-mail: ababnehj@just.edu.jo

Received: January 20, 2022 *Revised: April 05, 2022* *Accepted: April 11, 2022*

Abstract— In this paper, the design and optimization of infinite impulse response full-band digital differentiator (DD) using evolutionary and swarm intelligence algorithms is investigated. Different objective functions based on the absolute error, the squared absolute error and the min-max optimality criterion are investigated. The optimal DD parameters that result in the best minimum value of the investigated objective functions are obtained using differential evolution, particle swarm optimization, genetic algorithm and cuckoo search optimization methods. These algorithms are used due to their simplicity, efficiency and robustness in solving general multidimensional optimization problems. The investigation outcomes show that minimizing the absolute error gives the most flat magnitude response, and minimizing the squared absolute error gives almost the lowest mean error of the designed DD. In addition, a new objective function is proposed to improve the linearity of the phase response of the designed infinite impulse response full-band DD. It is found that the design of the DD using the differential evolution outperforms or at least is comparable to similar designs reported in the literature using other optimization methods.

Keywords— Digital differentiator; Evolutionary algorithms; Swarm intelligence algorithms; Optimization.

1. INTRODUCTION

Digital differentiator (DD) is an important part in signal processing. It has been used in many engineering applications such as radars and sonars [1], image processing [2], biomedical engineering [3, 4], and system identification and fault detection [5].

The frequency response of an ideal DD is given by [1]:

$$H_{DD}(e^{j\omega}) = j\omega, |\omega| < \pi \tag{1}$$

where ω represents the angular frequency in radians per sample. The DD in Eq. (1) can be approximated as a finite impulse response (FIR) or an infinite impulse response (IIR) digital system. It is the targeted application of the DD that mostly controls the use of FIR or IIR differentiator. FIR digital system is inherently stable and can be easily designed to have linear phase but usually has a higher order compared to its IIR counterpart. On the other hand, IIR digital system usually has much less order but does not have a linear phase response. In addition, the stability constraint on IIR digital system makes the design problem more difficult compared to FIR digital system.

Analytical approaches to design DD that approximate the ideal differentiator in Eq. (1) with some desired characteristics have been proposed in literature. In [5], the design of FIR differentiators by means of modulating functions and its application for fault detection is proposed. The design methodology of Chebyshev approximation of non-recursive filter proposed in [6] is used to design full-band and low pass DD filters with linear phase

* Corresponding author
Article's DOI: 10.5455/jjee.204-1642708977

response. Eigenfilter is used in [7] to design higher-order DDs. In [8], maximally-flat DD is presented.

The design of equiripple non-recursive DD using a weighted least-squares technique is presented in [9]. Full-band differentiators based on Taylor series is demonstrated in [10]. The design of stable IIR DD using iterative quadratic programming is presented in [11]. Difference formula and Richardson extrapolation is used to design DD in [12]. In [13], variable frequency range FIR DDs are designed. In [14], low-delay band pass maximally-flat FIR DDs are designed. Design of IIR DD using constrained optimization is proposed in [15]. Design of FIR DD using the L1 optimality criterion is proposed in [16].

On the other hand, only few methods have used evolutionary or swarm intelligence algorithms to design and optimize the DD performance. For example in [17], IIR DD is designed using the bat algorithm (BA). FIR DD design using BA and particle swarm optimization (PSO) based on the L1 optimality criterion is introduced in [18]. The design of IIR DD using harmony search (HS) algorithm is proposed in [19]. Simulated annealing (SA), genetic algorithm (GA), and modified Fletcher and Powell (FP) optimization are used to optimize the design of DD in [20]. Second order DD is designed using GA in [21]. FIR fractional order differentiator design using cuckoo search (CS) algorithm is discussed in [22]. The design of low pass DD has been considered in [23-33]. In [33], the differential evolution (DE) algorithm and the L1 optimality criterion were used to design low pass DD and improve its phase response by imposing the symmetry property of the numerator coefficients of the DD transfer function.

In [17-20], the DD design problem is formulated using different objective functions based on only the absolute magnitude error and the square magnitude error and - to the best of our knowledge - none have considered the min-max criterion to design DD. The magnitude response of the designed DD achieved by these studies is acceptable but can be further improved. In addition, the phase response of the designed DD achieved by these studies does not exhibit linear phase or achieved satisfactory phase response but at the expense of getting bad magnitude response. Furthermore, most available methods in the literature to design DD, have not considered different objective functions to study the effect of the used objective function on the performance of the designed DD. Therefore, we believe that more investigations need to be conducted to study different objective functions and their effect on the frequency response of the designed DD in order to improve its performance.

The purpose of this paper is to present a comprehensive investigation on the use of different objective functions and well-known optimization algorithms and study their effect on the design and performance of IIR DD in order to improve its performance. In this work, we consider the following optimality criteria as objective functions to optimize the design of DD: the absolute error, the squared error and the min-max criteria which are all related to the magnitude response. Empirical studies in [21, 34, 35] have shown that including the phase response with the magnitude response in an objective function does not improve the overall performance of the designed DD but results in that the search algorithm maturely converges to a solution that is a trade-off between the magnitude and the phase response. Well-known optimization algorithms, namely the DE, PSO, GA and the CS are utilized in this study to search for the differentiator parameters that result in the best minimum value of the proposed objective functions. It is worth mentioning that other optimization algorithms such as the BA

[17] , the HS algorithm [19], the SA and the FP [20] can be used to solve the DD design problem in the proposed formulation. However, the algorithms considered in this work are used due to their popularity, implementation simplicity, efficiency and robustness in solving general nonlinear optimization problems with and without constraints. In addition, by using these algorithms, there will be no need to calculate the derivative of the objective functions as usually needed in the traditional optimization methods such as nonlinear programming.

In this work, we present several design examples of IIR DD and compare our results with designs achieved using other techniques available in the literature. In addition, to improve the phase linearity of the designed IIR DD, we propose a new objective function that is a weighted sum of the three investigated criteria. In this proposed new objective function, no constraints are imposed on the coefficients of the DD. Thus, this is a different approach than that used in [33].

The main contributions of this study are summarized in the following points:

a) A comprehensive investigation is carried out on the use of the absolute error, the squared error, and the min-max criteria in addition to the DE, PSO, GA and the CS optimization algorithms and their effect on the design and performance of IIR DD.
b) A new objective function that is a weighted sum of the three investigated criteria is proposed to improve the phase linearity of the designed IIR DD.

The rest of this paper is organized as follows. In section 2, the formulation of the DD filter design problem as an optimization problem is presented. Section 3 briefly describes the DE, GA, PSO and the CS algorithms and their implementation to solve the DD design problem. Design examples and discussion are given in section 4. Finally, the conclusion of this paper is given in section 5.

2. PROBLEM FORMULATION

In this section, the design of IIR DD is formulated as an optimization problem. The frequency response of an Nth order IIR system can be written as [1]:

$$H\left(e^{j\omega}\right) = \frac{b_0 + b_1 e^{-j\omega} + b_2 e^{-j2\omega} + \cdots . + b_N e^{-jN\omega}}{a_0 + a_1 e^{-j\omega} + b_2 e^{-j2\omega} + \cdots . + a_N e^{-jN\omega}} \tag{2}$$

In most approaches, the design of DD is performed by selecting the system coefficients $(b_i, a_i, 0 \leq i \leq N)$ in Eq. (2) to approximate the magnitude of the frequency response in Eq. (1). For this purpose, the design is formulated as an optimization problem with appropriate objective function related to the magnitude response of an ideal DD. There are many different objective functions that can be used to design DD. In this paper, three different objective functions are used based on the absolute magnitude error (AME) defined in Eq. (3). The AME is the absolute difference between the magnitude responses of the ideal DDs ($\left|H_{DD}\left(e^{j\omega}\right)\right|$) and that of the approximated DDs ($\left|H\left(e^{j\omega}\right)\right|$) evaluated at L uniformly distributed samples on the frequency interval $0 \leq \omega \leq \pi$. Here, a value of L = 512 samples is used [19].

$$AME = \left|\left|H_{DD}\left(e^{j\omega}\right)\right| - \left|H\left(e^{j\omega}\right)\right|\right| \tag{3}$$

The IIR DD design in this study is based on the following objective functions as in [19], [17] and [36], respectively:

$$O_1 = \min\left(\sum_{L\ samples} \left||H_{DD}(e^{j\omega})| - |H(e^{j\omega})|\right|^2 + P\right) \tag{4}$$

$$O_2 = \min\left(\sum_{L\ samples} \left||H_{DD}(e^{j\omega})| - |H(e^{j\omega})|\right| + P\right) \tag{5}$$

$$O_3 = \min\left(max\,\langle\left||H_{DD}(e^{j\omega})| - |H(e^{j\omega})|\right|\rangle) + P\right) \tag{6}$$

The goal of the optimization algorithm is to find the system coefficients $(b_i, a_i,\ 0 \leq i \leq N)$ of the approximate frequency response $H(e^{j\omega})$ that result in the best minimum value of the considered objective function. In the search for the objective function minimum, the stability constraint should be also imposed. For an IIR system to be stable, all poles should be inside the unit circle (i.e., $|d_i| < 1$, where d_i and $1 \leq i \leq N$ are the poles of the system). Therefore, to guarantee the stability of the designed IIR DD, a penalty term, P, is used in the objective functions above. In this work, a value of P = 1000 is used whenever any of the poles is outside the unit circle; otherwise P = 0. The value of the penalty term P is chosen to be large enough such that whenever a candidate solution has any of its poles outside the unit circle, the fitness value corresponding to this candidate solution is large. Therefore, this solution is considered as a bad solution. For the three considered objective functions in this study, we have found that a relatively acceptable solution has a fitness value less than 15 so adding a value of 1000 significantly changes the fitness value of the candidate solution and the searching algorithm treats it as a bad solution. On the other hand, choosing P less than 1000 might make the searching algorithm to consider some of the unstable solutions as acceptable solutions especially at early stages of the search. Therefore, for this scenario, a large number of iterations might be needed for the algorithm to converge to a good solution which will in turn increase the computational time, a problem that we tried to avoid.

3. OPTIMIZATION ALGORITHMS

In this section, the DE, PSO, GA and the CS algorithms are briefly described. In addition, their implementation to solve the DD filter design problem is presented. It should be mentioned here that this section is not intended to be a thorough review of the used algorithms; for more details, the reader can refer to the references herein. However, the DD design problem formulation is given in section 2, and then the role of the optimization algorithms is to find the coefficient of the DD system function by solving the formulated DD design problem is presented.

3.1. The DE Algorithm

The DE algorithm was introduced by Price and Storn as a new floating point encoded evolutionary algorithm (EA) for global optimization [37]. The DE algorithm is one of the most popular stochastic optimization algorithms that can be used to optimize a user defined objective function of a given optimization problem. The DE algorithm has been successfully used in many applications such as pattern recognition [38], communications [39] and digital filter design [40]. The DE is an attractive alternative to other algorithms such as the GA and PSO due to three advantages. First, it has the ability to find the true global optima regardless of the initial parameter values. Second, it has fast convergence speed, and finally it uses few parameters to control the progress of the algorithm [36]. In addition, it is simple, fast, easy to use, adequately effective in solving nonlinear constraint optimization including penalty functions and useful for optimizing multi-modal search spaces [41].

Like other EAs, the DE algorithm starts with an initial population of individuals, (NP). Each individual is a candidate solution ($X_i = \{x_{1,i}, x_{2,i}, \ldots\ldots, x_{N,i}\}$) that represents the design parameters of an N-order optimization problem. Then, a new generation of candidate solutions is produced using the current population of candidate solutions through operations named mutation and crossover in the DE algorithm. The fitness values of the new generated candidate solutions are compared with the corresponding fitness values of the current candidate solutions. A new generated solution is taken to the next iteration if its fitness value is better than that of the corresponding current candidate solution. The algorithm keeps iteratively generating solutions from the current available candidate solutions till a termination criterion is met. The termination criterion could be chosen to be a certain number of generations or a certain error value. In the mutation operation, a mutant vector ($V_i = \{v_{1,i}, v_{2,i}, \ldots\ldots, v_{N,i}\}$) is generated for each candidate solution (X_i) in the current generation. There are several strategies to generate the mutant vector. The following are two common strategies [41]:

$$\begin{aligned} &\text{DE/rand/1: } V_i = X_{r_1} + F(X_{r_2} - X_{r_3}) \\ &\text{DE/best/1: } V_i = X_{best} + F(X_{r_2} - X_{r_3}) \end{aligned} \tag{7}$$

The subscripts r_1, r_2 and r_3 are random and mutually different integers generated in the range [1, NP] and are different from the current candidate solution's subscript i. F is a factor in the range [0, 2] to scale the differential vectors. X_{best} is the candidate solution with the best fitness value in the current generation. After the mutation operation, a crossover operation is used to generate a trial vector ($U_i = \{u_{1,i}, u_{2,i}, \ldots\ldots, u_{N,i}\}$) to each pair of the generated mutant vector V_i, and its corresponding target vector X_i. The j^{th} element in the trial vector is obtained according to the following equation [41]:

$$u_{j,i} = \begin{Bmatrix} v_{j,i}, \ if \ (rand_j < CR) or \ (j = j_{rand}) \\ x_{j,i}, \qquad\qquad\qquad\qquad otherwise \end{Bmatrix} \quad j = 1, 2, \ldots., N \tag{8}$$

The parameter CR is a user-specified crossover constant in the range [0, 1] and j_{rand} is a randomly chosen integer in the range [1, NP] to ensure that the trial vector U_i will differ from its corresponding target vector X_i by at least one parameter. $rand_j$ and is a random number drawn from uniform distribution in the range [0, 1]. Therefore, the jth element in the trial vector is constructed using selected elements of the mutant vector V_i, and its corresponding target vector X_i. Finally, the fitness values for both the target vector X_i and its corresponding trial vector U_i are calculated. If the fitness value of the trial vector U_i is better than that of the target vector X_i, the trial vector is moved to the next generation and the target vector is disregarded. Otherwise, the target vector is retained and moved to the next generation. In our implementation of the DE algorithm, DE/rand/1 strategy is used.

3.2. The GA

The GA uses processes analogous to genetic recombination and mutation to produce new individuals that best fits a desired objective function [42, 43]. The algorithm starts with a randomly generated initial population of individuals (candidate solutions) called generation. An individual is a set of the parameters that are to be optimized in the optimization problem. Each parameter is represented as an encoded gene in the algorithm. The collection of genes makes an individual that is called chromosome. Encoding of genes includes binary and real numbers (adopted in this work). The algorithm performs crossover and mutation operations on some selected individuals in the current population to produce new individuals to make a new generation. The new generation is used for the next iteration in the algorithm. In each iteration of the algorithm, selection, crossover and mutation operations are performed to get a new generation. The process of producing new generation is repeated until some termination criterion is satisfied. The selection operation is used to choose some individuals from the current generation to perform crossover and mutation operations on them. The selection is made based on the quality of the individuals measured by their fitness value. Individuals with better fitness value are most likely to be selected. Popular selection schemes include 'roulette wheel' and 'tournament' selection. The former selection method is used in our implementation of the GA. During the crossover operation, the genes of two individuals (parents) are combined to produce a new individual (child). On the other hand, one or more gene values in an individual (chromosome) are stochastically altered during the mutation operation to produce a new individual.

3.3. The PSO

The PSO was introduced by Kennedy and Eberhart in 1995 [44], and since then it gained great popularity and found many applications. All the information needed by the PSO algorithm is contained in the following vectors of each particle in the swarm: X (position), V (velocity), Pbest (personal best) and Gbest (global best). The velocity and position vectors are updated according to [44]:

$$\begin{aligned} V_t &= w\, V_{t-1} + c_1 r1(Pbest_{t-1} - X_{t-1}) + c_2 r2(Gbest_{t-1} - X_{t-1}) \\ X_t &= X_{t-1} + V_t \end{aligned} \tag{9}$$

where the subscript t refers to the time index of the current iteration, r1 and r2 are uniformly distributed random numbers in the interval [0, 1]. The cognitive parameters c_1 and c_2 specify the relative weight of the personal best position versus the global best position. The parameter w is called the "inertial weight," in the range (0, 1), and specifies the weight by which the particle's current velocity depends on its previous velocity and how far the particle is from its personal best and global best positions. More details about the PSO algorithm, typical values of its parameters and sample applications can be found in [45-51].

3.4. The CS Algorithm

Yang and Deb have developed the CS algorithm in 2009 based on the strange breeding behavior of some species of cuckoo birds [52]. The CS algorithm starts by placing the nest population randomly in the search space. Afterwards, the nest locations (potential solutions) are updated with time iterations t using [52]:

$$X_t = X_{t-1} + \alpha \cdot S_t \quad (10)$$

where S_t is step size, $\alpha > 0$ is a scaling factor, and X_t stands for the nest location (potential solution). In [52, 53], it is suggested that the search capability of the CS algorithm is enhanced if the step size S_t is drawn from a L´evy distribution. For more details about the CS, typical values of its parameters and sample applications, the reader may consult the literature [52-58].

3.5. Algorithms Implementation and Parameters

As previously mentioned, during the implementation of the algorithms, all algorithms are run to find the system coefficients $(b_i, a_i, 0 \leq i \leq N)$ of an IIR system that result in the best minimum of the objective functions in Eqs. (4 - 6). The dimension of the optimization problem is the number of the system coefficients $(b_i, a_i, 0 \leq i \leq N)$. Table 1 shows the representation of the optimization parameters (the coefficients of the DD system function) in the different algorithms during the implementation. All algorithms start with a randomly generated initial population of candidate solutions. At the initialization stage, each element of a candidate solution is randomly initialized in the interval [-10, 10]. For each candidate solution, the objective function is evaluated. Then a new population is produced using all/some of the individuals in the current population by applying the mutation and crossover operations for the DE and GA algorithms or the updated Eqs. (9) and (10) for the PSO and CS algorithms, respectively. The production process of new population continues till a termination criterion is reached. Here, the maximum number of iterations is adopted as the termination criterion. The main steps in our implementation of the algorithms are summarized as follows:

Step1: The objective function is defined and a population of candidate solutions (randomly initialized) is produced.

Step2: The objective function is evaluated for all candidate solutions in the current population.

Step3: A new population is produced by applying some certain operations (such as mutation, crossover, etc.).

Step 4: The termination criterion is checked and if it is not met, then go back to step 2.

Table 1. Representation of the optimization parameters in the used algorithms.

Algorithm	DE	GA	PSO	CS
$(b_i, a_i, 0 \leq i \leq N)$	Individual	Chromosome	Position of a particle	Nest

Table 2 gives the control parameters of the utilized algorithms. These parameters are chosen based on our personal and other researchers' experience in this field to achieve satisfactory solution and fast convergence for a wide range of engineering optimization problems [37-58]. Other values of these parameters (in their typical ranges) may be tried, but we found that the values listed in Table 2 are the best for our design problem. Then, the different algorithms were implemented using MATLAB™ [59] on a personal computer with Intel (R) Core (TM) i7 CPU with 2.67 GHz and 4 GB of RAM.

Table 2. Implementation parameters of the used algorithms.

Parameter	Algorithm			
	DE	GA	PSO	CS
Population size	71	71	71	71
Maximum Iterations	500	500	500	500
F, CR	0.47, 0.88	-	-	-
Crossover, Crossover probability	-	2 points, 1	-	-
Selection, Initial mutation probability		roulette wheel, 0.01	-	-
c_1 and c_2	-	-	2	-
$W_{initial}$, W_{final}	-	-	0.9, 0.4	-
Abandon probability, *pa*	-	-	-	0.25

4. DESIGN EXAMPLES AND DISCUSSION

In this section, we present design examples of third and fourth order IIR DDs using the DE, GA, PSO, and CS algorithms. To validate our results, we compared our results with published results in the literature using other techniques and algorithms. The comparison is mainly made based on the value of the objective function considered and the mean error value (in dB). Table 3 shows the comparison of the designed IIR DDs based on the squared AME (O_1) using the DE, GA, PSO, CS and HS algorithms [19]. The AMEs of the designed DDs based on O_1 are shown in Figs. 1 and 2. The results clearly show that the designed DDs using the DE outperformed the designed DDs using other algorithms.

Table 3. IIR DD design based on O_1.

Method	Order	Sum of Squared AME	Mean [dB]
DE	3	**0.001**	**-58.616**
GA		1.553	-26.870
PSO		0.592	-32.684
CS		0.029	-46.499
HS [19]		0.012	-49.855
DE	4	**0.001**	**-73.055**
GA		0.115	-39.998
PSO		0.109	-40.629
CS		0.024	-46.020
HS [19]		0.017	-46.953

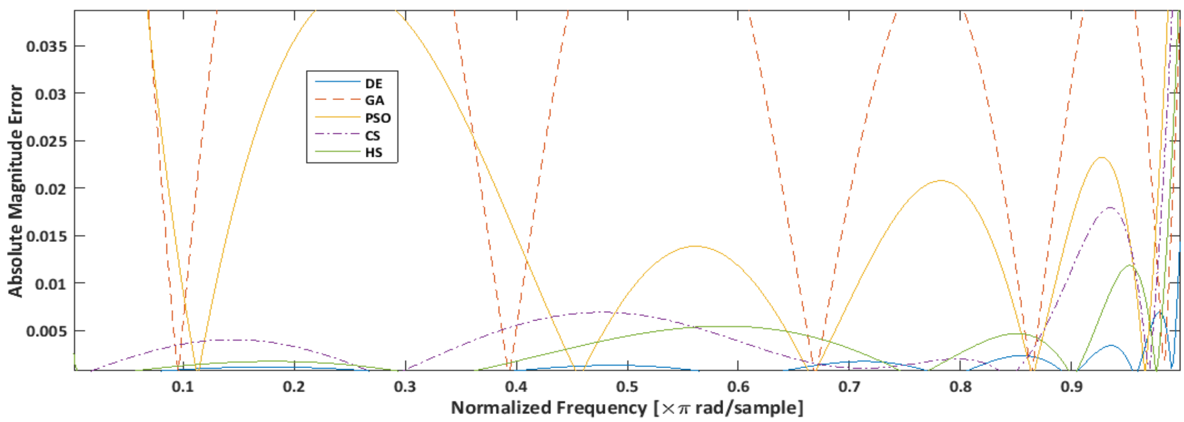


Fig. 1. AME for third order IIR DD design based on O_1 using different algorithms.

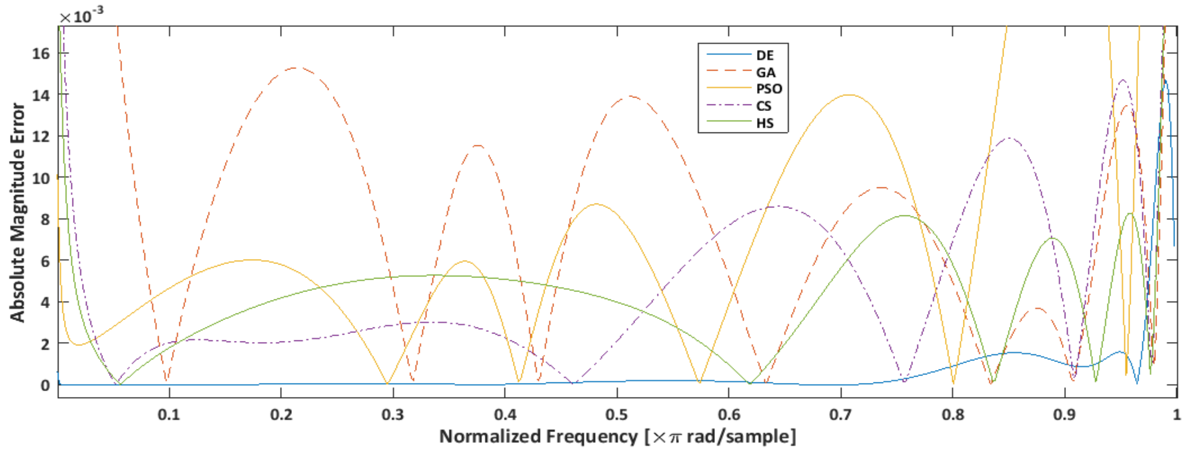


Fig. 2. AME for fourth order IIR DD design based on O_1 using different algorithms.

The optimization results of the designed IIR DDs based on AME (O_2) using the DE, GA, PSO, SA [20], GA [20], FP [20], GA [21] and BA [17] are given in Table 4. The AMEs of the designed DDs based on O_2 are shown in Figs. 3 and 4. As can be seen from Figs. 3 and 4 and Table 4, the DDs designed using the DE based on O_2 has much better performance compared with those designed using other algorithms.

Table 4. IIR DD design based on O_2.

Method	Order	Sum of Squared AME	Mean [dB]
DE	3	0.438	-61.353
GA		13.903	-31.322
PSO		20.819	-27.815
CS		2.790	-45.273
SA [20]		2.336	-46.813
GA [20]		2.962	-44.751
FP [20]		0.856	-55.531
DE	4	0.320	-64.083
GA		11.607	-32.890
PSO		3.276	-43.877
CS		3.708	-42.801
SA [20]		2.151	-47.532
GA [20]		2.067	-47.876
FP [20]		0.757	-56.594

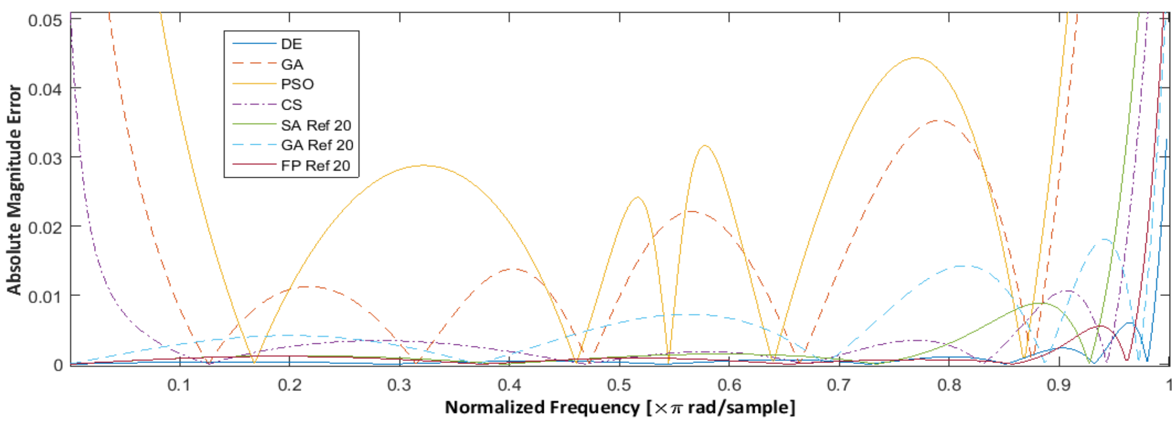


Fig. 3. AME for third order IIR DD design based on O_2 using different algorithms.

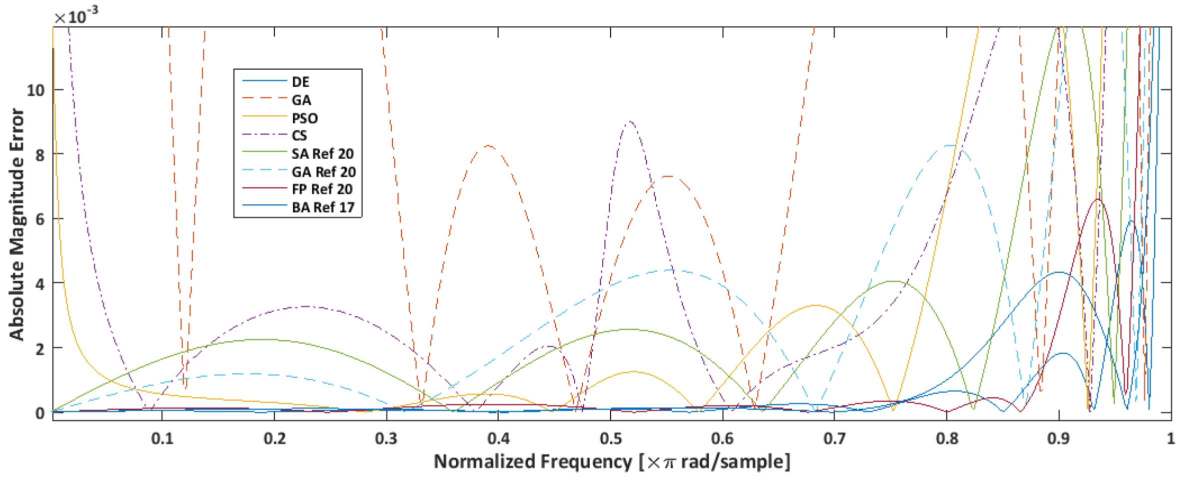


Fig. 4. AME for fourth order IIR DD design based on O_2 using different algorithms.

Table 5 and Figs. 5 and 6 show the performance comparison of the designed DDs based on O_3. Again, the designed DDs using the DE algorithm outperformed those obtained using other algorithms.

Table 5. IIR DD design based on O_3.

Method	Order	Sum of Squared AME	Mean [dB]
DE	3	0.009	-44.391
GA		0.085	-27.274
PSO		0.040	-31.374
CS		0.034	-35.422
DE	4	0.004	-51.800
GA		0.084	-27.479
PSO		0.031	-33.922
CS		0.027	-38.714

As can be seen from Tables 3-5, the DE algorithm was able to find better solution in terms of the minimum value of the objective functions.

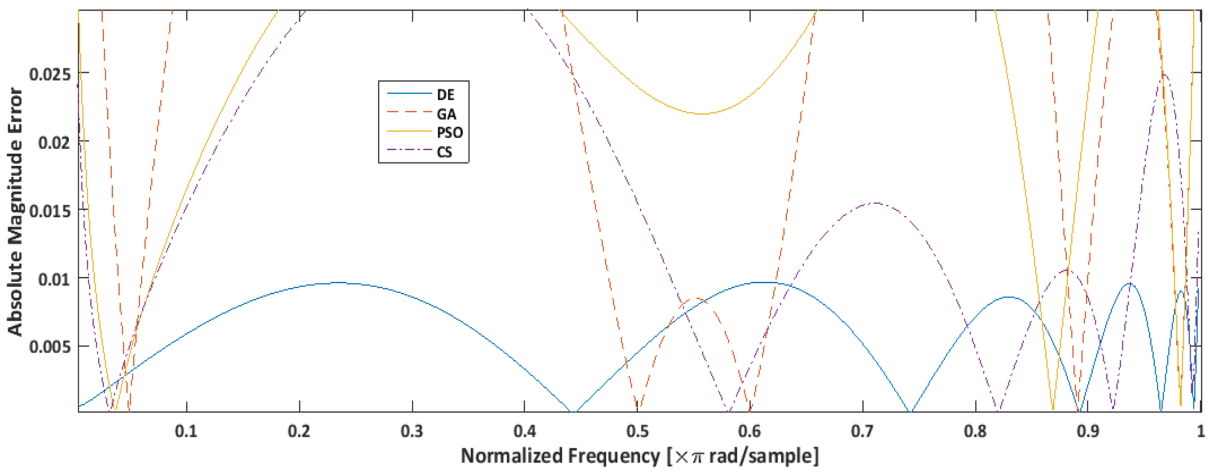


Fig. 5. AME for third order IIR DD design based on O_3 using different algorithms.

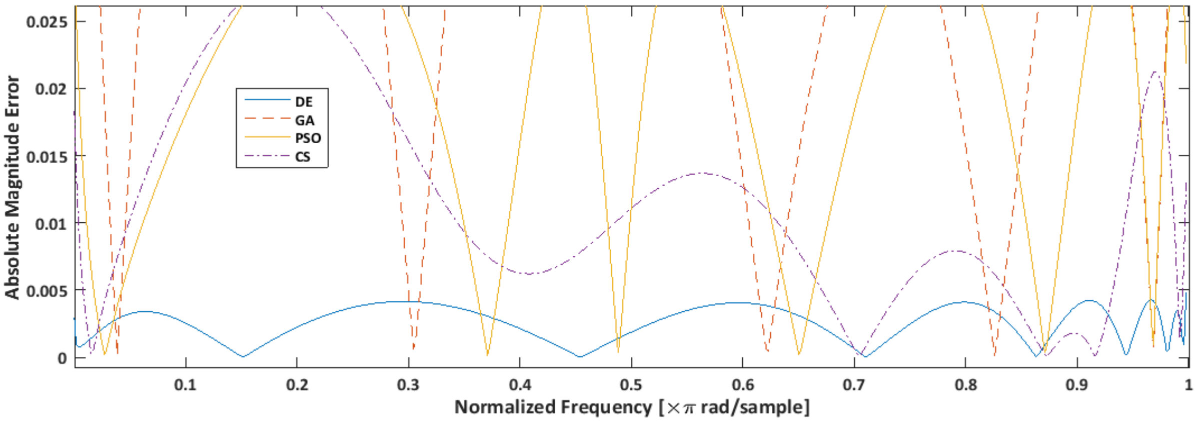


Fig. 6. AME for fourth order IIR DD design based on O_3 using different algorithms.

The comparisons of the magnitude (using AME) and phase response for the designed DDs using the DE algorithm based on the three objective functions are shown in Figs. 7 to 12. It can be seen in Figs. 7 and 10 that the design of the IIR DD based on the AME (O_2) achieved the most flat magnitude response as it has the lowest absolute error for most of the frequency band. On the other hand, as shown in Figs. 8 and 11, the phase response of the designed IIR DD based on the three objective functions (O_1, O_2 and O_3) is close enough to linear for most of the frequency band with deviation from linearity especially at very low and high frequencies. This conclusion is more clearly supported by the group delay in Figs. 9 and 12. The coefficients of the designed DDs using the three objectives are listed in Tables 6-8.

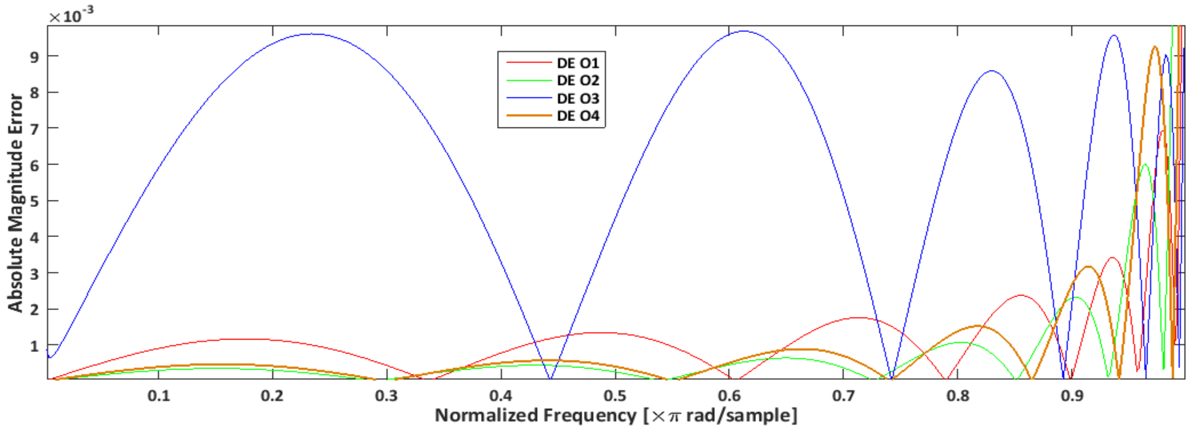


Fig. 7. AME for the best third order IIR DD design using DE based on O_1, O_2, O_3 and O_4.

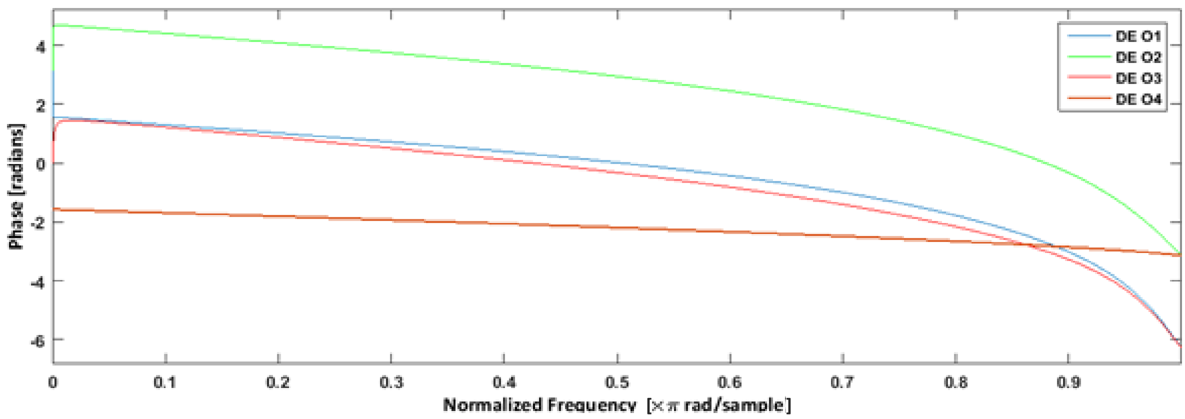


Fig. 8. Phase response for the best third order IIR DD design using DE based on O_1, O_2, O_3 and O_4.

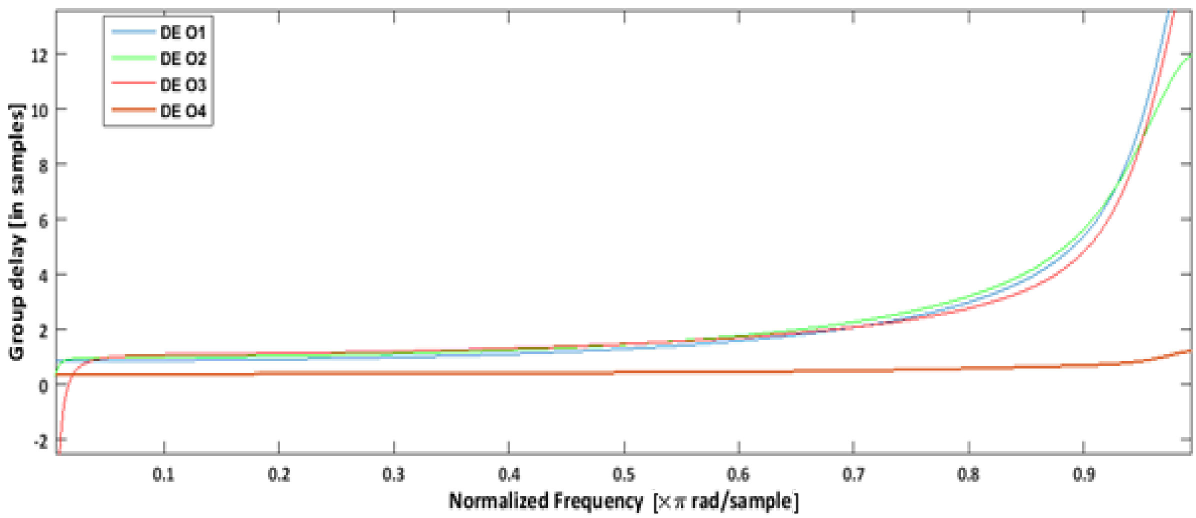


Fig. 9. Group delay for the best third order IIR DD design using DE based on O_1, O_2, O_3 and O_4.

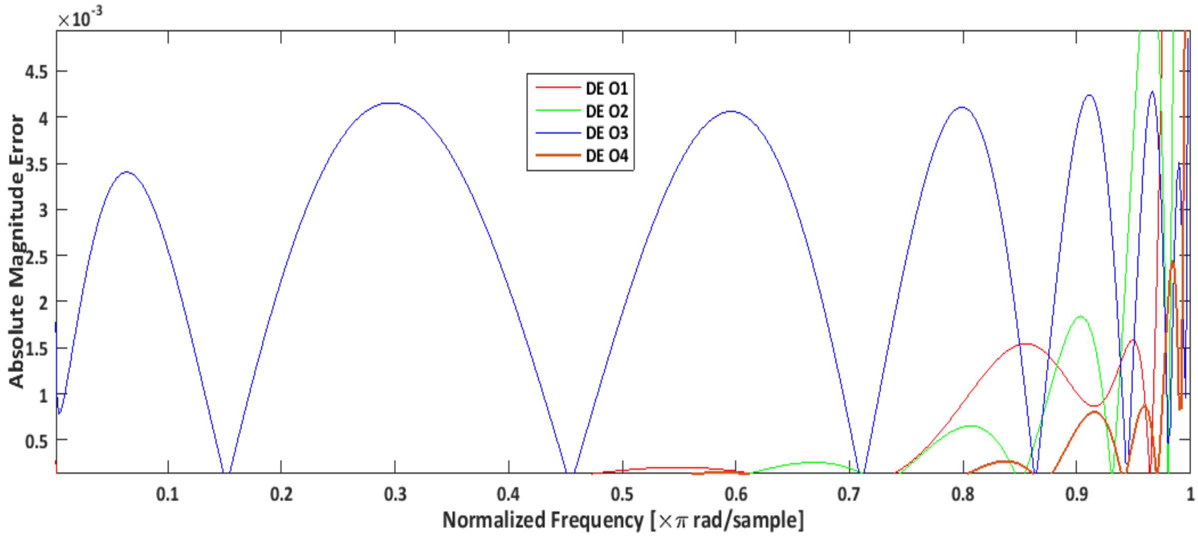


Fig. 10. AME for the best fourth order IIR DD design using DE based on O_1, O_2, O_3 and O_4.

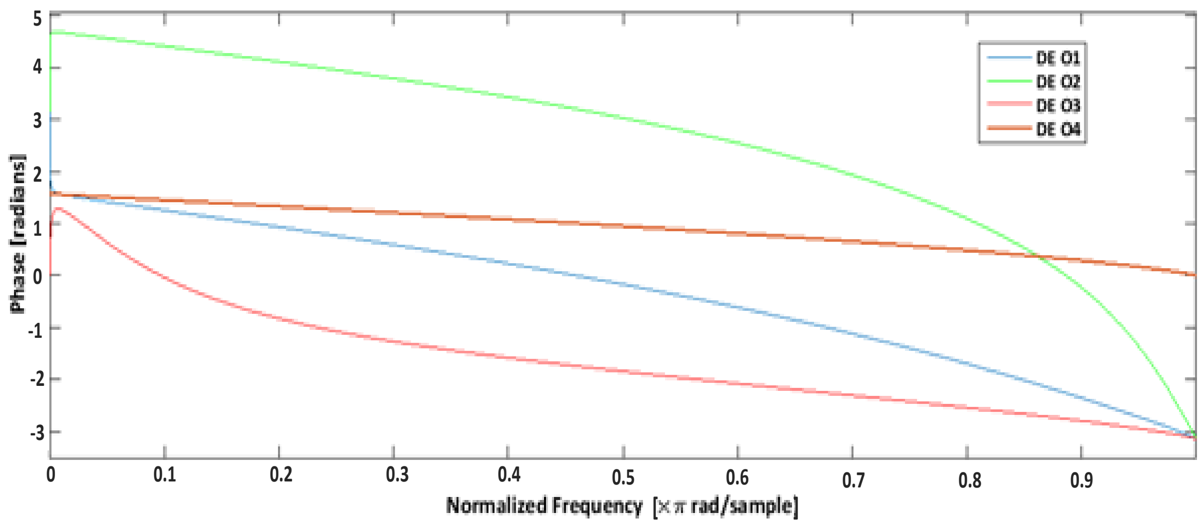


Fig. 11. Phase response comparisons for the best fourth order IIR DD design using DE based on O_1, O_2, O_3 and O_4.

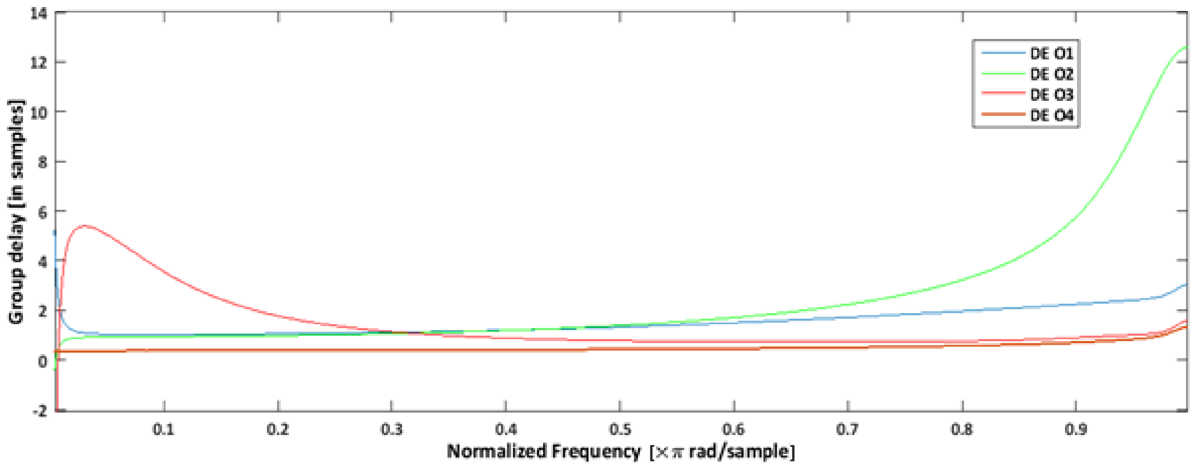


Fig. 12. Group delay for the best fourth order IIR DD design using DE based on O_1,O_2,O_3, and O_4.

Table 6. Coefficients for the IIR DD design based on O_1.

Method	Order	Denominator	Numerator
DE	3	-6.5920 -9.5895 -3.6616 -0.2613	-2.7194 -6.9855 2.0908 7.6142
GA		7.1230 3.1999 0.4390 0.4662	1.9183 -9.4441 3.0 2.2939
PSO		-10.0 -5.4558 -0.0637 -0.1735	-1.0432 4.5391 9.1828 -10.0
CS		-8.6482 -5.8704 0.2980 0.2037	-10.0 4.9764 5.8816 -0.8374
HS [19]		1.0054 1.2538 0.4122 0.0283	1.1644 0.0827 -0.9484 -0.2858
DE	4	1.4090 2.7877 1.7597 0.3735 0.0156	0.3746 1.8408 0.5614 -1.8069 -0.9747
GA		7.0373 2.9544 1.3918 3.0 0.5244	-3.1356 1.4110 -0.5568 -6.7613 7.6310
PSO		-8.1427 -0.8833 -2.6017 -2.5345 -0.1204	9.4885 -10.0 4.1079 -1.2531 -2.1865
CS		7.1088 -0.3195 -3.5724 0.4168 0.1597	-5.4265 2.4254 9.6084 -7.9326 1.4403
HS [19]		1.0383 1.7181 1.0077 0.2480 0.0157	1.1821 0.5283 -0.9171 -0.7236 -0.1644

Table 7. Coefficients for the IIR DD design based on O_2.

Method	Order	Denominator	Numerator
DE	3	-7.4293 -9.7653 -3.2771 -0.1969	2.3158 7.4309 -1.1590 -8.5833
GA		-8.1404 -0.4746 -2.5429 -0.7214	9.8612 -10.0000 4.6163 -3.0000
PSO		-10.0000 -4.8056 -8.3137 -1.7452	9.6163 -4.4445 9.7643 -10.0000
CS		8.6796 6.1637 0.4941 -0.0257	5.6197 4.8268 -9.8274 0.1924
SA [20]		1.0000 0.8662 0.1612 0.0028	1.1555 -0.3582 -0.7140 -0.0833
GA [20]		1.0000 0.7981 0.0884 0	1.1533 -0.4432 -0.7060 -0.0041
FP [20]		1.0000 1.2488 0.4076 0.0256	1.1548 0.0798 -0.9499 -0.2847
DE	4	7.7822 9.7575 2.6120 -0.1804 -0.0360	-2.7590 -7.5992 2.5367 8.8090 -0.9923
GA		-8.8850 -4.9759 1.5771 -1.7181 -0.6224	-7.6728 5.7239 8.5119 -6.2148 2.5804
PSO		10.0000 5.2443 4.4444 2.5841 0.1781	4.7455 6.2915 -10.0000 3.7102 -5.1511
CS		-8.8459 -6.0297 -7.5960 -4.6612 -0.4317	-9.9953 5.3908 -2.7669 4.4455 3.9716
SA [20]		1.0000 1.3788 0.6230 0.1059 0.0059	1.1540 0.2290 -0.8794 -0.4486 -0.0549
GA [20]		1.0000 0.9054 0.1713 0.0066 0	1.1553 -0.3170 -0.7560 -0.0817 -0.0006
FP [20]		1.0000 1.7315 1.0150 0.2208 0.0109	1.1554 0.6388 -0.9050 -0.7518 -0.1374
BA [17]		0.4173 0.7473 0.4200 0.0773 0.0027	0.0431 0.3190 0.4154 -0.2951 -0.4823

Table 8. Coefficients for the IIR DD design based on O_3.

Method	Order	Denominator	Numerator
DE	3	7.8107 9.8474 2.7057 0.0254	1.8356 8.0818 -0.7967 -9.0552
GA		9.6950 6.9309 2.0549 0.6575	10.0000 -6.5521 -4.3850 -0.7140
PSO		10.0000 8.4115 1.8786 0.2389	-1.7667 -10.0 7.1752 5.4159
CS		8.4475 6.2267 -0.5158 -0.2464	1.9830 -10.0 1.2914 7.1816
DE	4	7.3345 6.9194 -3.1142 -3.3986 -0.3668	-5.9456 5.7087 9.6987 -5.1754 -4.2616
GA		-7.9787 -1.5789 0.1850 -0.6104 -0.0724	0.1875 -8.4454 9.8330 0.1748 -0.8967
PSO		-10.0 -9.3626 -6.5103 -6.6038 -1.2208	-7.1854 2.3189 -5.7443 -0.0587 9.6171
CS		8.9065 6.0931 -0.0267 1.0188 0.3171	1.1347 -2.0584 5.7673 5.4710 -10.0

During our investigation of the three different objective functions, we have noticed that the magnitude response (in certain frequency band) of the designed DD based on a given objective function is better than that of the other two objective functions while the phase response is worse and vice versa. Therefore, based on this finding, we propose a new objective function which is a weighted sum of the three objective functions (which represents a compromise between the two contradictory objectives) in an attempt to improve both the magnitude and phase response of the designed DD. The new proposed objective function is:

$$O_4 = \sum_{i=1}^{3} W_i O_i \tag{11}$$

where W_i's are weighting factors that are chosen through different trial runs. We have empirically studied the effect of the weights on the frequency response by considering many different combinations in the range of 1 to 10. Based on the used objective function in Eq. (11), our empirical study showed that using the weights $W_1 = 3$, $W_2 = 1$ and $W_3 = 3$ gives the best effect on the phase linearity improvement.

The AME of the designed IIR DDs using objective function O_4 are given in Figs. 7 and 10 which show that the designs based on O_4 have better, or at least comparable, performance compared with designs based on the objective functions O_1, O_2, and O_3, but the use of O_4 greatly improved the phase response of the designed IIR DDs.

The improvement of phase linearity of the designed IIR DD using objective function O_4 is demonstrated in Figs. 8, 9, 11, and 12 where it is very clear that the phase (Figs. 8 and 11) is almost linear (constant group delay in Figs. 9 and 12) for the entire frequency band except for very small deviation at very high frequencies. The deviation from linearity at very low frequencies is much improved and moderated at high frequencies for all of the IIR DDs designs as compared with IIR DDs designs based on the objective functions O_1, O_2, and O_3, separately. Quantitative measure of the phase linearity using the mean group delay of the designed DDs is given in Table 9. It shows that the DDs designed based on O_4 have the lowest mean group delay.

The designed IIR DDs using the DE algorithm - which has the best performance among all methods based on O_4 - is given in Table 9, where we see that the mean AME of the designed IIR DDs based on O_4 is less than those of the designed IIR DDs based on O_1 and O_3. But the designed IIR DDs based on O_4 has slightly higher maximum AME as compared to the designed IIR DDs based on O_1 and O_3. Table 9 also shows the mean of the group delay for the designed DDs based on the four objective functions. DDs designed based on O_4 have the lowest mean of the group delay. Therefore, the use of O_4 improved the linearity of the phase response for the designed IIR DDs and did not significantly degrade its magnitude response. Table 10 contains the coefficients of the designed IIR DDs based on O_4.

Table 9. IIR DD best design using DE based on the used objective functions O_1, O_2, O_3 and O_4.

Method	Order	Max of AME	Sum of AME	Sum of Squared AME	Max of AME [dB]	Mean of AME [dB]	Mean of group delay
O_1	3	0.015	0.600	0.001	-36.364	-58.616	2.49
O_2		0.032	0.438	0.003	-29.710	-61.353	2.48
O_3		0.009	3.088	0.023	-40.267	-44.391	2.52
O_4		0.020	0.471	0.002	-33.784	-60.707	0.49
O_1	4	0.007	0.113	0.001	-42.562	-73.055	1.52
O_2		0.032	0.320	0.003	-29.898	-64.083	2.48
O_3		0.004	1.315	0.004	-47.641	-51.800	1.44
O_4		0.010	0.095	0.001	-39.816	-74.619	0.49

Table 10. Coefficients for the IIR DD design based on O_4 using DE.

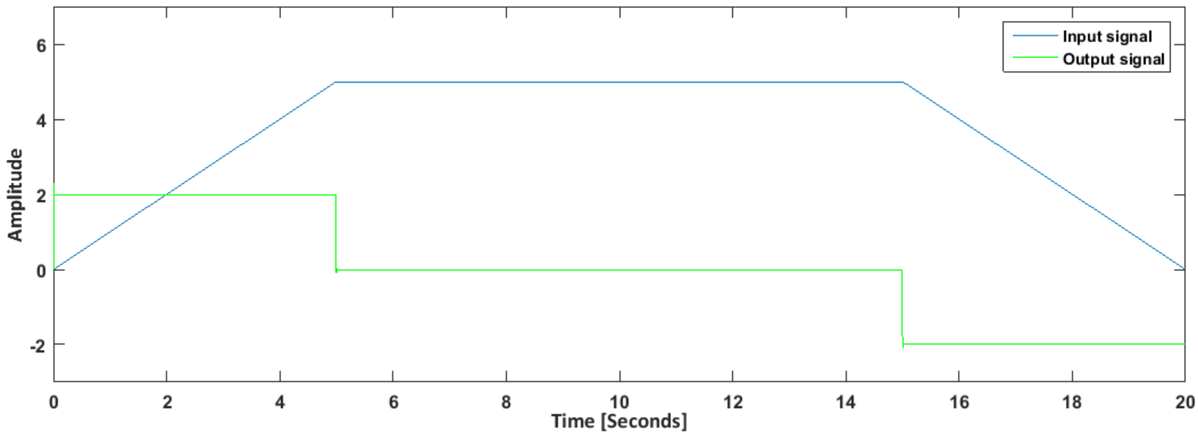
Order	Denominator					Numerator				
3	-6.8852	-9.3649	-3.2685	-0.2055			7.9557	1.4375	-7.0471	-2.3461
4	-4.7651	-9.5208	-6.1220	-1.3460	-0.0606	-5.5069	-4.5101	4.5565	4.6365	0.8240

To show the robustness of the used search algorithms, the designs reported in this work are achieved by running each algorithm hundred independent runs and then report the best frequent design. For example, in Table 11 we show statistics for hundred independent runs of the DE algorithm for selected designs. We see that the DE algorithm reached similar designs in terms of the best value of the used objective function as evident from the small value of the standard deviation for the hundred runs. Of course reaching the same value of the objective function does not necessarily mean the same coefficient values. However, it shows the ability of the DE algorithm to find good solutions with different independent runs. Finally, the ability of the best designed DD using the DE algorithm to determine the derivative of input signal is tested on two synthetic signals.

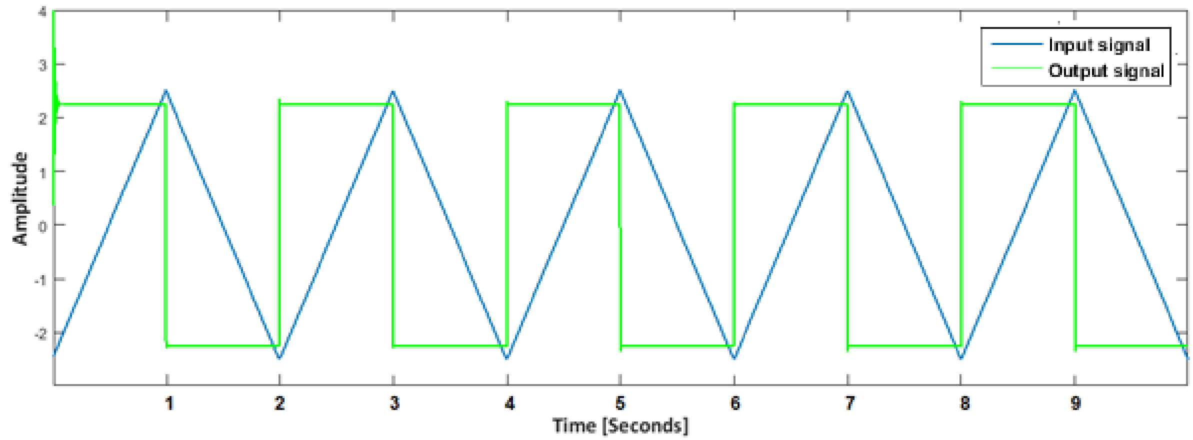
Table 11. Statistics for 100 independent runs of the DE algorithm based on O_2.

Absolute Magnitude Error (AME)					
Order	Mean	Max	Min	Reported	Standard Deviation
3	0.439	0.453	0.437	0.438	0.005
4	0.206	0.348	0.094	0.320	0.101

The response of the designed DD to two sample synthetic signals is shown in Fig. 13 where we clearly see that the designed DD has successfully produced the derivative of the input signals with negligible error.



(a)



(b)

Fig. 13. Response of the fourth order IIR DD to selected synthetic signals designed using the DE based on O4.

5. CONCLUSIONS

In this paper, we presented a thorough investigation of the design of optimum IIR full-band DD based on three different objective functions and using some of the well-known evolutionary and swarm intelligence algorithms. For this purpose, we used the DE, PSO, GA and CS as search algorithms. For the numerical examples studied in this paper, the achieved DD performance in most cases is better than that achieved using other reported in literature methods. It was also found that using the DE algorithm to minimize the absolute error and the squared error gave the most flat magnitude response and almost the lowest mean error of the designed DDs, respectively. Furthermore, the linearity of phase response of IIR DD is greatly improved by using a newly-proposed objective function that is a weighted sum of the three studied objective functions. The robustness of the used algorithms is also verified through a hundred independent runs for some design examples.

REFERENCES

[1] M. Skolnik, *Introduction to Radar Systems*, McGraw-Hill, New York, 1980.

[2] J. Lim, *Two Dimensional Signal and Image Processing*, Prentice-Hall, Englewood Cliffs, NJ, 1990.

[3] S. Susui, I. Amidror, "Digital low-pass differentiation for biological signal processing," *IEEE Transactions on Biomedical Engineering*, vol. 10, pp. 686–693, 1982.

[4] C. Nayak, S. Saha, R. Kar, D. Mandal, "An efficient QRS complex detection using optimally designed digital differentiator," *Circuits, Systems, and Signal Processing*, vol. 38, no. 2, pp. 716–749, 2019.

[5] A. Wolfram, O. Moseler, "Design and application of digital FIR differentiators using modulating functions," *IFAC Proceedings*, pp. 1037–1042, 2000.

[6] T. Parks, J. McClellan, "Chebyshev approximation for nonrecursive digital filters with linear phase," *IEEE Transactions on Circuit Theory*, vol. 19, pp. 189–194, 1972.

[7] S. Pei, J. Shyu, "Eigenfilter design of higher-order digital differentiators," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 37, pp. 505–511, 1989.

[8] B. Carlsson, "Maximum flat digital differentiator," *Electronics Letters*, vol. 27, no. 8, pp. 675–677, 1991.

[9] S. Sunder, V. Ramachadran, "Design of equiripple nonrecursive digital differentiators and Hilbert transformers using a weighted least-squares technique," *IEEE Transactions on Signal Processing*, vol. 42, pp. 2504–2509, 1994.

[10] I. Khan, R. Ohba, "New design of full band differentiators based on Taylor series," *IEE Proceedings-Vision, Image and Signal Processing*, vol. 146, no. 4, pp. 185–189, 1999.

[11] C. Tseng, "Stable IIR digital differentiator design using iterative quadratic programming approach," *Signal Processing*, vol. 80, pp. 857–866, 2000.

[12] C. Tseng, S. Lee, "Design of digital differentiator using difference formula and Richardson extrapolation," *IET Signal Processing*, vol. 2, no. 2, pp. 177-188, 2008.

[13] B. Kumar, S. Dutta, "Design of universal, variable frequency range FIR digital differentiators," *Circuits, Systems, and Signal Processing*, vol. 11, no. 3, pp. 431–439, 1992.

[14] T. Yoshida, N. Aikawa, "Low-delay band-pass maximally flat FIR digital differentiators," *Circuits, Systems, and Signal Processing*, vol. 37, no. 8, pp. 3576–3588, 2017.

[15] R. Nongpiur, D. Shpak, A. Antoniou, "Design of IIR digital differentiators using constrained optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1729–1739, 2014.

[16] A. Aggarwal, T. Rawat, M. Kumar, D. Upadhyay, "Efficient design of digital FIR differentiator using L1-method," Radioengineering, vol. 25, no. 2, pp. 86-92, 2016.

[17] A. Aggarwal, T. Rawat, D. Upadhyay, "Optimal design of L1-norm based IIR digital differentiators and integrators using the bat algorithm," *IET Signal Processing*, vol. 11, no. 1, pp. 26-35, 2016.

[18] A. Aggarwal, M. Kumar, T. Rawat, "Design of digital differentiator using the L1-method and swarm intelligence-based optimization algorithms," *Arabian Journal for Science and Engineering*, vol. 44, no. 3, pp. 1917-1931, 2018.

[19] S. Mahata, S. Saha, R. Kar, D. Mandal, "Optimal design of wideband digital integrators and differentiators using harmony search algorithm," *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, vol. 30, no. 5, pp. 1-20, 2016.

[20] M. Al-Alaoui, M. Baydoun, "Novel wideband digital differentiators and integrators using different optimization techniques," *International Symposium on Signals, Circuits and Systems*, pp. 1–4, 2013.

[21] M. Jain, M. Gupta, N. Jain, "Linear phase second order recursive digital integrators and differentiators," *Radioengineering*, vol. 21, no. 2, pp. 712–717, 2012.

[22] M. Kumar, T. Rawat, "Optimal design of FIR fractional order differentiator using cuckoo search algorithm," *Expert Systems with Applications*, vol. 42, no. 7, pp. 3433-3449, 2015.

[23] B. Kumar, S. Roy, "Design of digital differentiators for low frequencies," *Proceedings of the IEEE*, vol. 76, no. 3, pp. 287–289, 1988.

[24] B. Kumar, S. Roy, "Coefficients of maximally linear FIR digital differentiators for low frequencies," Electronics Letters, vol. 24, no. 9, pp. 563–565, 1988.

[25] B. Kumar, S. Roy, H. Shah, "On the design of FIR digital differentiators which are maximally linear at the frequency pi/p, p in (positive integers)," *IEEE Transactions on Signal Processing*, vol. 40, no. 9, pp. 2334–2338, 1992.

[26] I. Selesnick, "Maximally flat lowpass digital differentiators," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 49, no. 3, pp. 219–223, 2002.

[27] F. Kuo, J. Kaiser, *System Analysis by Digital Filters*, Wiley, New York, 1966.

[28] J. Pan, W. Tompkins, "A real-time QRS detection algorithm," *IEEE Transactions on Biomedical Engineering,* vol. BME-32, no. 3, pp. 230–236, 1985.

[29] M. Al-Alaoui, "Linear phase low-pass IIR digital differentiators," *IEEE Transactions on Signal Processing,* vol. 55, no. 2, pp. 697-706, 2007.

[30] A. Tahmasbi, S. Shokouhi, "New optimised IIR low-pass differentiators," *In 2010 International Conference on Signal Acquisition and Processing*, Bangalore, India, pp. 205-209, 2010.

[31] R. Nongpiur, D. Shpak, A. Antoniou, "Design of IIR digital differentiators using constrained optimization," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1729-1739, 2014.

[32] Z. He, Y. Sun, "Design of low-pass digital differentiators based on B-splines," *Signal Processing: An International Journal*, vol. 8, no. 3, pp. 30-42, 2014.

[33] J. Ababneh, M. Khodier, "Design of approximately linear phase low pass IIR digital differentiator using differential evolution optimization algorithm," *Circuits, Systems, and Signal Processing*, vol. 40, pp. 5054–5076, 2021.

[34] M. Al-Alaoui, "Class of digital integrators and differentiators," *IET Signal Processing*, vol. 5, no. 2, pp. 251–260, 2011.

[35] D. Upadhyay, "Class of recursive wide band digital differentiators and integrators," *Radioengineering*, vol. 21, no. 3, pp. 904–910, 2012.

[36] A. Oppenheim, R. Schafe, *Discrete-Time Sigal Processing*, Prentice Hall, New Jersy, 2009.

[37] R. Storn, K. Price, "Differential evolution-A simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, pp. 341-359, 1997.

[38] J. Ababneh, T. Aldalgamouni, A. Alqudah, "Minimum bit error rate multiuser detection of SDMA-OFDM system using differential evolutionary algorithm," *IEEE 6th International Conference on Wireless and Mobile Computing, Networking and Communications*, pp. 273-279, 2010.

[39] R. Storn, "Differential evolution design of an IIR-filter," *Proceedings of IEEE International Conference on Evolutionary Computation*, pp. 268-273, 1996.

[40] Z. Albataineh, J. Ababneh, F. Salem, "Linear phase FIR low pass filter design using hybrid-differential evolution," *International Journal of Research in Wireless Systems*, vol. 1, no. 2, pp. 43-49, 2012.

[41] D. Karaboğa, S. Ökdem, "A simple and global optimization algorithm for engineering problems: differential evolution algorithm," *Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 12, no. 1, pp. 53-60, 2004.

[42] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer Science and Business Media, New York, 1996.

[43] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, MA , 1989.

[44] J. Kennedy, R. Eberhart, "Particle swarm optimization," *In Proceedings of ICNN'95-International Conference on Neural Networks*, Piscataway, NJ, pp. 1942-1948, 1995.

[45] Y. Shi, R. Eberhart, "Parameter selection in particle swarm optimization," *Proceedings of Evolutionary Programming VII*, pp. 591–600, 1998.

[46] R. Eberhart, Y. Shi, "Particle swarm optimization: developments, applications and resources," *In Proceedings of the 2001 Congress on Evolutionary Computation*, vol. 1, pp. 81-86, 2001.

[47] J. Perez, J. Basterrechea, "Comparison of different heuristic optimization methods for near-field antenna measurements," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 549-555, 2007.

[48] M. Khodier, C. Christodoulou, "Linear array geometry synthesis with minimum sidelobe level and null control using particle swarm optimization," *IEEE Transactions on Antennas and Propagation*, vol. 53 , no. 8, pp. 2674-2679, 2005.

[49] M. Bataineh, J. Ababneh, "Synthesis of a periodic linear phased antenna array using particle swarm optimization," *Electromagnetics*, vol. 26, no. 7, pp. 531-541, 2006.

[50] D. Boeringer, D. Werner, "Particle swarm optimization versus genetic algorithms for phased array synthesis," *IEEE Transactions on Antennas and Propagation*, vol. 52, no. 3, pp. 771–779, 2004.

[51] J. Nanbo, Y. Rahmat-Samii, "Advances in particle swarm optimization for antenna designs: real-number, binary, single-objective and multiobjective implementations," *IEEE Transactions on Antennas and Propagation*, vol. 55, no. 3, pp. 556–567, 2007.

[52] X. Yang, S. Deb, "Cuckoo search via Lévy flights," *Proceedings of World Congress on Nature and Biologically Inspired Computing*, pp. 210–214, 2009.

[53] X. Yang, S. Deb, "Engineering optimization by cuckoo search," *International Journal of Mathematical Modelling and Numerical Optimization*, vol. 1, no. 4, pp. 330–343, 2010.

[54] A. Gandomi, X. Yang, A. Alavi, "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems," *Engineering with Computers*, vol. 29, pp. 17-35, 2013.

[55] X. Yang, S. Deb, "Cuckoo search: recent advances and applications," *Neural Computing and Applications*, vol. 24, no. 1, pp. 169–174, 2014.

[56] B. Mohamad, M. Zain, E. Bazin, "Cuckoo search algorithm for optimization problems—a literature review and its applications,"*Applied Artificial Intelligence*, vol. 28, no. 5, pp. 419-448, 2014.

[57] M. Khodier, "Optimization of antenna arrays using the cuckoo search algorithm," *IET Microwaves, Antennas and Propagation*, vol. 7, no. 6, pp. 458-464, 2013.

[58] M. Khodier, "Comprehensive study of linear antenna array optimization using the cuckoo search algorithm," *IET Microwaves, Antennas and Propagation*, vol. 13, no. 9, pp. 1325-1333, 2019.

[59] *MATLAB R2015a*, The MathWorks Inc., Natick, Massachusetts, United States.