Accepted Manuscript

Implementation of Fractional Order Filters Discretized by Modified Fractional Order Darwinian Particle Swarm Optimization

Abdullah Ates, Baris Baykant Alagoz, Gurkan Kavuran, Celaleddin Yeroglu

PII: S0263-2241(17)30294-4
DOI: http://dx.doi.org/10.1016/j.measurement.2017.05.017
Reference: MEASUR 4743

To appear in: *Measurement*

Received Date: 11 January 2017
Revised Date: 27 March 2017
Accepted Date: 7 May 2017

Please cite this article as: A. Ates, B. Baykant Alagoz, G. Kavuran, C. Yeroglu, Implementation of Fractional Order Filters Discretized by Modified Fractional Order Darwinian Particle Swarm Optimization, *Measurement* (2017), doi: http://dx.doi.org/10.1016/j.measurement.2017.05.017

Implementation of Fractional Order Filters Discretized by Modified Fractional Order Darwinian Particle Swarm Optimization

[a*]Abdullah Ates, [a]Baris Baykant Alagoz, [b]Gurkan Kavuran ,[a]Celaleddin Yeroglu

[a]*Inonu University, Engineering Faculty, Computer Engineering Department, Malatya, TURKEY*

[b]*Firat University, Technology Faculty, Mechatronics Engineering Department, Elazig, TURKEY*

Corresponding Author: Abdullah Ates

e-mail: abdullah.ates@inonu.edu.tr

Abstract

Digital systems are placed at the core of information technology and they are used extensively in electronics. The digital filter realization has become a central topic of signal processing studies. This paper presents a discrete IIR filter design method for approximate realization of fractional order continuous filters in digital systems. For this purpose, Fractional Order Darwinian Particle Swarm Optimization (FODPSO) method is modified to provide better fitting of a discrete IIR filter function to a fractional order continuous filter and we implemented a hybrid version of FODPSO method, where the initial particle generation is carried out by arithmetical candidate point selection technique of the Base Optimization Algorithm (BaOA). This modification expands the search range of the FODPSO and thus the optimized discrete IIR filter can provide better approximation to amplitude response of fractional order continuous filter functions. In the paper, several illustrative examples are presented to demonstrate the performance of proposed methods.

Keywords: Fractional order filter; discrete realization; optimization; particle swarm

1. Introduction

Filters are essentially used to obtain desired frequency selectivity, by shaping pass band, stop band and transition band of systems. Filter realization is a fundamental topic of analog and digital system design. As a result of wide spreading of digital technology, the digital filters and digital systems are getting more substantial role in daily life. In practice, digital filter realization are commonly carried out by using discrete Linear Time Invariant (LTI) systems and the design of these filters are taken into account as the effort of obtaining filter coefficients that provide a desired amplitude response.

In recent years, fractional order systems have gained importance for applied science and engineering problems because of their performance improvement over integer order counterparts (Das et al., 2011; Das, 2008). Recent developments in practice of fractional order LTI systems enable to implement fractional order filter functions. In contrast to integer order continuous filter design, the fractional order continuous filter design process includes determination of the not only filter coefficients but also fractional orders, and this gives more degree of freedom in shaping frequency response. In fact, the fractional order filter function is a more general form of filter function family, which also includes the integer-order filter functions. The integer-order filters are simply a tight subset of fractional-order filters (Radwan et al., 2008). A major advantage of fractional order system in design of filter frequency responses is that the slope of transition bands can be adjusted fractionally by using fractional order parameters (Freeborn et al., 2015). Despite its advantages in the shaping the frequency response, the digital realization of fractional system is more complicated than the realization of integer order filters because of the long memory effect of fractional order derivative (Ren et al., 2016). For a fully approximation to the response of fractional order filters, the integer order transfer functions may need infinite number of filter coefficients. Advances in hardware practice enable to implement fractional order elements by using various forms of hardware realizations and approximation approaches (Krishna, 2011). These approaches aim to approximate the original fractional order systems to a certain degree of accuracy in operating ranges of systems (Santamaria et al., 2008).

For the practical purposes, a band limited implementation of the fractional order (FO) elements can be useful. Principally, FO elements, which are theoretically infinite dimensional linear filters, can be approximated in a specified frequency band of practical interest (Podlubny et al., 2002) by using indirect and direct discretization methods (Al-Alaoui, 2009). The indirect discretization method can be achieved in two steps. Firstly, the frequency domain fitting is carried out in a desired frequency range and then the fitted continuous time transfer function is discretized. Direct discretization based methods (Chen et al., 2009) include the application of Power Series Expansion (PSE), Continuous Fractional Expansion (CFE) (Chen et al., 2004), MacLaurin Series Expansion (Valerio and Sa da Costa, 2005) etc. with a suitable generating function. The mapping relation or formula for conversion from the continuous time to discrete time operator is called as the generating function. Among the family of expansion methods, CFE based digital realization has been extensively studied with various types of generating functions such as Tustin (Vinagre et al., 2003), Simpson (Tseng, 2007), Al-Alaoui (Chen and Moore, 2002), mixed Tustin Simpson (Chen and Vinagre, 2003), mixed Euler-Tustin-Simpson (Zhu-Zhong and Ji-Liu, 2008), impulse response based (Maione, 2008) and other higher order generating functions (Gupta et al., 2011; Visweswaran et al., 2011). Recently, digital realization of non-integer order filters with discrete time-domain Laguerre impulse response approximation was presented (Baranowsk et al., 2016).

Optimization methods are also utilized to improve the approximation of fractional order filter in a desired frequency range. In recent works, heuristic optimization methods were employed to improve amplitude and phase responses of discrete IIR realization of fractional order differentiator and integrators: Das et al. (2011) have used genetic algorithm to improve rational approximation of fractional order differ-integrators obtained by CFE method. Particle Swarm Optimization (PSO) method was also employed for the improvement of rational approximations to fractional-order systems (Zhe and Liao, 2012). A novel method to design fractional-order differentiator operators through optimization using Nelder–Mead simplex algorithm is presented (Rana et al., 2016).

This paper focuses on discrete implementation of fractional order continuous filters, which are based on IIR filter approximation. For this purpose, the FODPSO method is employed to minimize the discrepancy between amplitude responses of a fractional order original filter and the corresponding IIR target filter. We addressed two important assets in the filter design process: (i) Better fitting of amplitude responses of IIR discrete filter to fractional order filter, and (ii) ensuring the stability of resulting IIR discrete filter.

The search skill of FODPSO optimization is further enhanced by using candidate solution generation technique of BaOA in the process of initial swarm generation for FODPSO algorithm and this combined optimization method is referred to as Hybrid Base-Fractional Order Darwinian Particle Swarm Optimization (HB-FODPSO) method. The BaOA benefits from basic arithmetic operators to generate candidate solutions, from a nearby point to a far distant point of the search space. This property of BaOA is used to modify initial swarm generation process of FODPSO algorithm. With this modification, the HB-FODPSO can well spread particles into the search region and thus improves search skill of FODPSO. This property is particularly needed for filter coefficient optimization problems so that the ranges of numerator and denominator polynomials coefficients can be wide apart. Therefore, approximation to satisfactory filter solutions may require the spreading of initial particles to a wide range of coefficients. On the other hand, the stability of IIR discrete filter is an important concern for practical filter realization studies. In order to ensure the stability of the resulting IIR filter, a stability checking mechanism is employed in the modified FODPSO algorithm and initial particles set are formed by stable filter solutions. Thus, the proposed HB-FODPSO ensures the stability of optimized IIR discrete filters approximating to the original continuous time fractional order filters. We present illustrative examples to show the performance of proposed methods.

2. Fundamentals of PSO Algorithms

2.1 Classical PSO Algorithm

The classical PSO was developed in 1995 by Kennedy and Eberhart (1995). In the classical PSO, the candidate solutions are referred to as particles. Each particle in the swarm is modeled by its position $x_n(t)$ and velocity $v_n(t)$ values in a multidimensional search space. The position and velocity values of particles depends on the local best $\chi_{1n}(t)$ also known as the cognitive component, and global best $\chi_{2n}(t)$ typically known as the social component, as written bellow,

$$v_n[t+1] = wv_n[t] + \sum_{i=1}^{2} \rho_i r_i (\chi_{in}[t] - x_n[t]) \tag{1}$$

$$x_n[t+1] = x_n[t] + v_n[t+1] \tag{2}$$

Parameters w, ρ_i are the weight coefficients for inertia of particles. Equation (1) determines the new velocity $v_n[t+1]$ at the iteration time (t). Parameters ρ_i ($i = \{1,2\}$) are the constant integer values, which adjust weights of "cognitive" and "social" components, respectively.

2.2 Darwinian PSO Algorithm

DPSO was formulated by Tillett et al. (2005). Main difference of DPSO from the classical PSO is that the swarm of DPSO is composed of particle groups. In the DPSO, each group individually performs as an ordinary PSO. The collections of groups are governed by the natural selection mechanism which is also known as Darwinian principle of survival-of-the-fittest, enhances the escaping ability of the PSO from local optima. The idea is to run many simultaneous PSO algorithms on the same search problem and apply a natural selection mechanism to improve search skill of algorithm (Tsai, 2002; Darwin, 1872). When a search tends to a local optimum, the search in that area is simply discarded and a new search in another area is initiated. In this approach, at each step, groups giving better results are rewarded by extending particle life or spawning a new descendant, and groups giving inappropriate results are punished by reducing group life or deleting particles. In order to analyze the general state of the each group, the solution of all particles is evaluated, and the global best of a given group, as well as each individual local best position of each particle within such group, are updated. If a new global solution is found, a new particle group is generated. Particles are removed when the group fails to find a better solution in a defined number of steps (Couceiro and Ghamisi, 2015).

2.3. FODPSO Algorithm

Fractional order system model presents the advantage of better representation of real world phenomenon. This advantage comes from the long-term memory effect of derivative terms, because the fractional order derivatives include the terms associated with the past besides the term related with current time. However, integer order derivative operator does not have a long-term memory effect because only current time term is considered by integer order derivatives.

Considering the inertial influence of Eq. (1) as $w=1$, for a specific swarm, one may obtain:

$$v_n[t+1]=v_n[t]+\sum_{i=1}^{2}\rho_i r_i\left(\chi_{in}[t]-x_n[t]\right) \tag{3}$$

This expression can be rewritten:

$$v_n[t+1]-v_n[t]=\sum_{i=1}^{2}\rho_i r_i\left(\chi_{in}[t]-x_n[t]\right) \tag{4}$$

Hence, $v_n[t+1]-v_n[t]$ corresponds to the discrete version of the fractional order difference for the order $\alpha=1$, that is, the first-order forward difference $D^{\alpha=1}v_n=\frac{1}{T}(v_n[t+T]-v_n[t])$. Assuming unit time increment $T=1$ and Eq. (5), which indeed calculates Grünwald-Letnikov fractional order derivative, the Eq. (6) can be written for fractional order inertia and the Eq. (7) is obtained for the fractional order velocity of FODPSO (Couceiro and Ghamisi, 2015) as follows.

$$D^{\alpha}[x[t]]=\frac{1}{T^{\alpha}}\sum_{k=0}^{r}\frac{(-1)^k\,\Gamma[\alpha+1]x[t-kT]}{\Gamma[k+1]\Gamma[\alpha-k+1]} \tag{5}$$

$$D^{\alpha}[v_n[t+1]]=\sum_{i=1}^{2}\rho_i r_i\left(\chi_{in}[t]-x_n[t]\right) \tag{6}$$

$$v_n^{(\alpha)}[t+1]=-\sum_{k=0}^{r}\frac{(-1)^k\,\Gamma[\alpha+1]v_n[t+1-kT]}{\Gamma[k+1]\Gamma[\alpha-k+1]}+\sum_{i=1}^{2}\rho_i r_i\left(\chi_{in}[t]-x_n[t]\right) \tag{7}$$

By using fractional order velocity of particles given by Eq. (7), the order of velocity derivative can be configured to a real number, thus FODPSO can exhibit rather smoother variation and longer memory in particle motion as a results of long memory effect of fractional derivatives. As a consequence, FODPSO is the same as having multiple PSOs, wherein particles try to find the best solution for their own "survival", with the benefit of inherently having a memory of past decisions. This new architecture deals with the first drawback pointed out for the classical PSO, which is the premature convergence of swarms. Similar to DPSO (Tillett et al., 2005), the FODPSO discards swarms, which are prematurely converge toward non-optimal solutions. At the same time, it promotes the creation of new swarms formed by particles that may "genetically" share some of the knowledge already retrieved by other particles. Additionally, the each FODPSO particle can behave smarter than PSO and DPSO particles due to the support of long memory effect of fractional order velocity. This allows FODPSO algorithm with a smaller population have similar performance compared to the DPSO algorithm. Indeed, for $\alpha = 1$, FODPSO algorithm performs the traditional DPSO that is based on integer order velocity. The pseudo code of FODPSO is shown in Figure 1.

Start

Set Initial parameters $v_n[0], x_n[0]\ \chi_{1n}[0]\ \chi_{2n}[0])$

for i=1:1:Max. Number of the iteration

Generated swarm matrix

    for n=1:1:Number of swarm matrix row

    Calculate fitness function of each row

    end

    Obtain min. fitness function's parameter configuration

    if min. fitness function(i)< min. fitness function(i-1)

        Update $\chi_{1n}^{i}[t], \chi_{2n}^{i}[t]$

        Update $v_n^i[t+1], x_n^i[t+1]$

        else

        Kill all swarm matrix member

        Go to "generated swarm matrix"

    end

end

end

Figure 1. Pseudo code for FODPSO algorithm

3. HB-FODPSO for Discretization of Fractional Order Filter

The proposed HB-FODPSO method benefits from the favored features of BaOA and FODPSO. BaOA uses basic arithmetic operators to guide and redirect the solutions towards the optimum point (Salem, 2012). For the sake of simplicity, the arithmetic operators of BaOA are limited with the addition (+), subtraction (-), multiplication (*), and division (/). In the current study, we modify initial particle generation process of FODPSO algorithms by candidate solution selection mechanism of BaOA that is based on the basic arithmetic operator set {+, -, *, /}. Here, by using random numbers between zero and one, the multiplication operation can provide nearby particle generation; the division operator can result in very distant initial particle configurations. Thus, the search range of the HB-FODPSO can be increased to reduce the possibility of getting stuck in unwanted local solutions.

In this paper, we consider the discrete implementation of fractional order continuous filter functions with the second order transfer function. These filter functions were expressed in the form of, (Ogata, 2010)

$$F_c(s) = \frac{\omega_0^{\ 2}}{s^{2\alpha} + 2\xi\omega_0 s^{\alpha} + \omega_0^{\ 2}} \tag{8}$$

In the current study, these continuous fractional order filter-functions are implemented by 4th order IIR discrete filter form written as,

$$F_d(a,b,z) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2} + a_3 z^{-3} + a_4 z^{-4}}{b_0 + b_1 z^{-1} + b_2 z^{-2} + b_3 z^{-3} + b_4 z^{-4}} \tag{9}$$

In order to obtain desired frequency selectivity, the amplitude response shaping is more important than the phase response shaping in filter design. Therefore, cost function for minimization are defined as,

$$J(a,b) = \sum_{\omega_i \in (\omega_{\min}, \omega_{\max})} \left(20\log_{10}\left|F_c(s)\right| - 20\log_{10}\left|F_d(a,b,z)\right|\right)^2 \tag{10}$$

where, $a = [a_1\ a_2\ a_3\ \ldots\ a_k]$ and $b = [b_1\ b_2\ b_3\ \ldots\ b_k]$ vectors are IIR filter coefficients to be optimized in the sampled frequency range $a_i \in (a_{\min}, a_{\max})$.

For the solution of this optimization problem, we employed HB-FODPSO algorithm in this study. In order to find out filter coefficients providing satisfactory approximation to the amplitude response of fractional order filters, the following two modifications are applied:

(i) Improvement of search ranges of initial particle generation process of FODPSO algorithms by using the candidate solution selection mechanism of BaOA, which can be expressed arithmetically as,

$$a = \left[(a_i - \Psi) \;\; (a_i + \Psi) \;\; (a_i * \Psi) \;\; (a_i / \Psi) \right] \tag{11}$$

$$b = \left[(b_i - \Psi) \;\; (b_i + \Psi) \;\; (b_i * \Psi) \;\; (b_i / \Psi) \right] \tag{12}$$

where the parameter Ψ represents a random number in the range of (0,1]. The division operator provides long-range searches, while multiplication can provide short-range search. The subtraction and addition operators are useful for mid-range searches.

(ii) The particles causing unstable filters are removed from the initial swarm set. This enforces that the optimization takes place in a stable filter solution subspace of the search space.

In general, filters are designed to obtain a desired frequency selectivity that is specified by magnitude response of systems. Since the proposed optimization algorithm ensures stability of resulting discrete filters, the cost function, given by Eq. (10), does not need for a phase response consideration in the optimization process. This approach is advantageous for filter design because the optimization algorithm can focus on only matching of the magnitude responses in stable filter solution subspaces of the search space, and it can further improve the magnitude response approximation of resulting discrete filters to fractional order original filter functions. Figure 2 presents a flow chart demonstrating basic steps of proposed HB-FODPSO algorithm. Tasks indicated by asterisk show the modifications that are added to improve filter design.

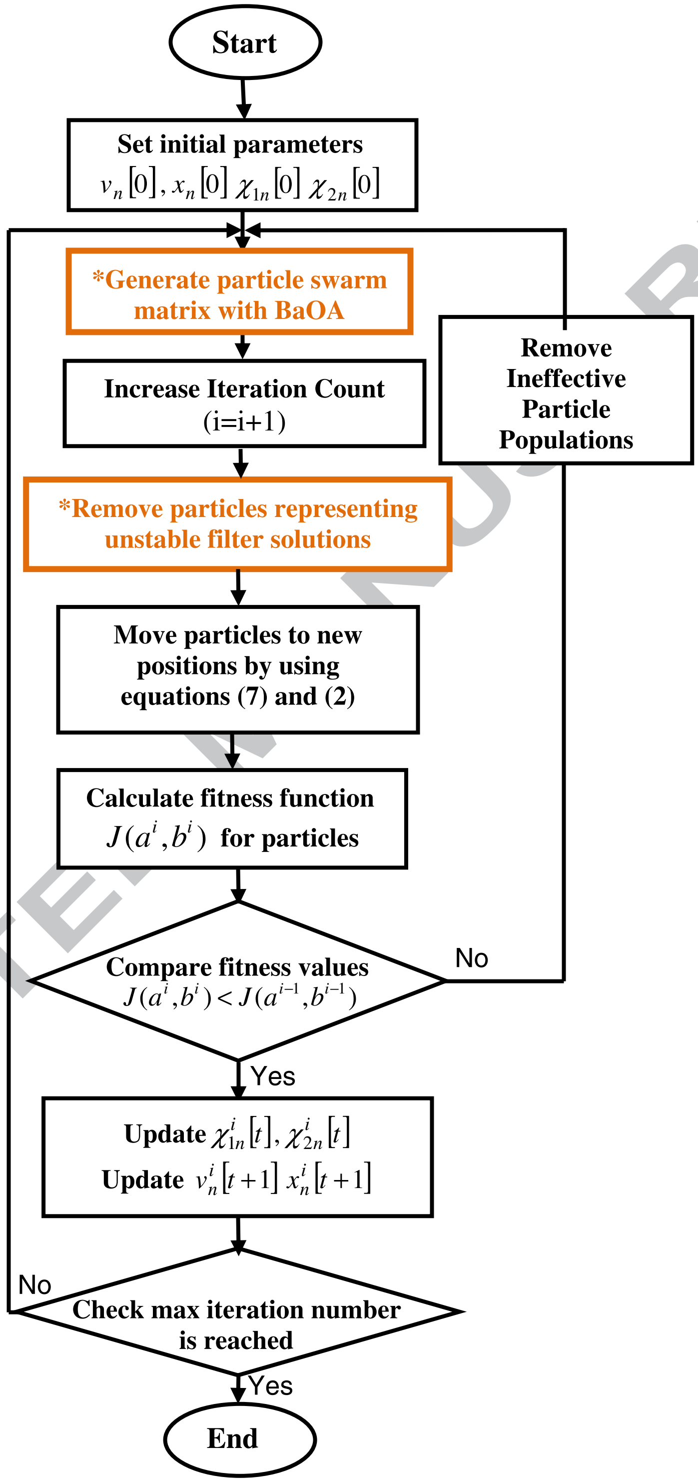


Figure 2. An algorithm for HB-FODPSO algorithm

4. Simulation Study

Due to their stochastic nature, it is difficult to fairly compare results of heuristic optimization techniques. In order to relevantly assess the result of proposed algorithm, we applied two test strategies:

(i) *The fixed random number tests*: In these tests, the same random number set is used under different configurations. Thus, one compensates the effect of different random number generation on the performance of the algorithms and this allows fairly comparing the response of algorithm for various initial configurations.

(ii) *The free random number tests*: In these tests, new random numbers are used to observe performance of algorithms for the real working conditions. It should also be noticed that the heuristic optimization algorithms utilizing random numbers should exhibit a consistent performance under free random number generation. Two filter design examples are given to illustrate performance of the proposed algorithm:

Example 1: Lets design an IIR discrete filter that approximates to the fractional order filter function $F_1(s) = \frac{1}{s^{1.6}+1}$ in the frequency range of $(0, 3140)$ rad/sec. with a unit frequency increment of 10 rad/sec. Sampling frequency for digital IIR filter was taken $T_s = 0.001$ sec.

Firstly, we checked the stability of continuous filter $F_1(s)$. For this propose, stability check procedure based on the root placement analysis in the first Riemann sheet was used (Radwan et al., 2009; Alagoz et al., 2015; Senol et al., 2014). Figure 3(a) demonstrates that the root in the first Riemann sheet satisfies the stability condition of characteristic root argument, which is defined as $\pi/20 < Arg(v) < \pi/10$. Here, $Arg(v)$ represents arguments of roots of characteristic equation of $F_1(s)$ under $s = v^{10}$ conformal mapping (Radwan et al., 2009; Alagoz et al., 2015; Senol et al., 2014). In Figure 3(b), we also confirm the stability of $F_1(s)$ by considering the step response of $F_1(s)$. Since the continuous fractional order filter is a stable filter, the discrete IIR filter implementation approximating to $F_1(s)$ should also be stable. At this point, the proposed HB-FODPSO algorithm ensures filter stability by constraining particle generation into the search regions producing stable discrete IIR filters.

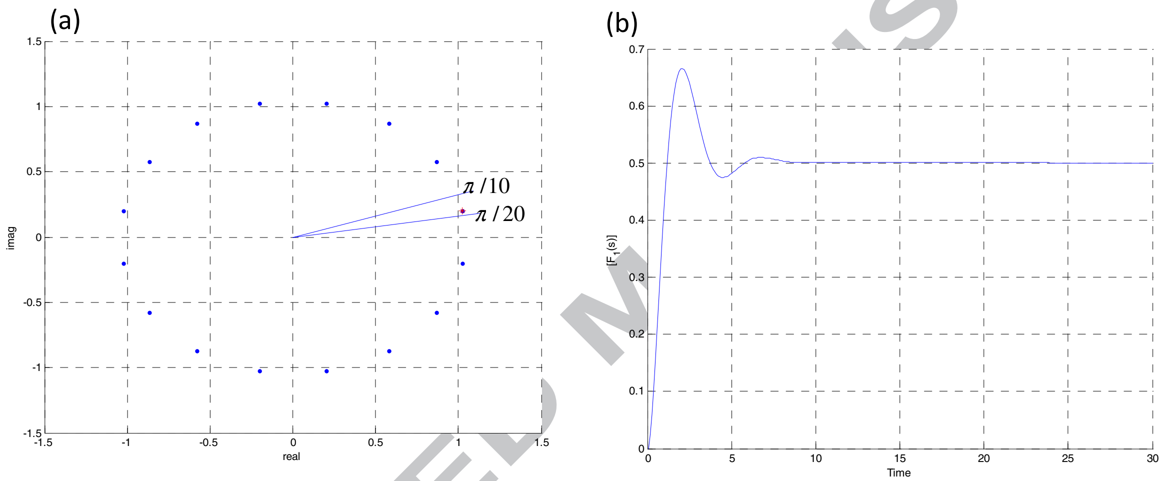


Figure 3. (a) Root placement analysis; (b) Step response for $F_1(s)$

For $a = 1$, the evolution of cost function of HB-FODPSO is shown in Figure 4. The characteristic of approximating towards zero indicates that the algorithm converges to an optimal solution point. At the end of iterations, the optimized discrete IIR filter function was obtained as,

$$F_{1a}(z) = \frac{0.5159 + 1.3659z^{-1} + 1.3801z^{-2} + 1.4044z^{-3} + 0.9121z^{-4}}{7.2848\,10^{4} - 5.7512\,10^{3}z^{-1} - 916.5092z^{-2} + z^{-3} - 6.6173\,10^{4}z^{-4}} \tag{13}$$

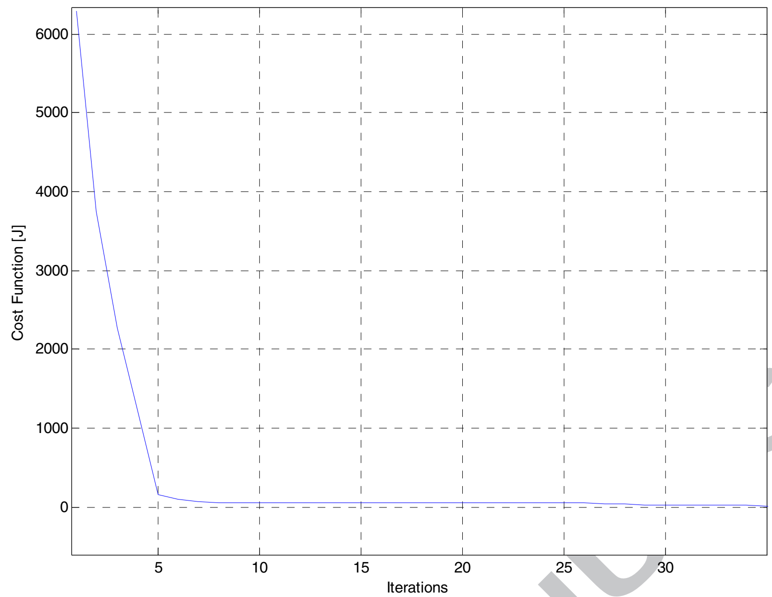


Figure 4: Change of cost function during HB-FODPSO for $F_{1a}(z)$

Figure 5 shows the amplitude responses of discrete IIR filters obtained for $\alpha \in [0.1, 3.0]$ with 0.1 increments. The results were obtained for the fixed random number set. Thus, one can compare the results of different test configurations of HB-FODPSO by omitting effects of random number generation process on optimization performance. The dashed line indicates the frequency response of original continuous fractional order function $F_1(s)$ that the proposed algorithm runs to approximate by discrete IIR filter functions. In order to see impacts of fractional order velocity of particles in optimization performance of HB-FODPSO, we compared the cost values of optimized designs for various fractional orders $\alpha \in [0.1, 3.0]$ in Figure 6. As one can see from the figure, in the region $\alpha > 0.9$, the values of the cost function do not change significantly. Table 1 lists values of the cost function for $\alpha \in [0.1, 3.0]$. Here, the best performance is obtained for $\alpha = 0.9$ and the resulting discrete IIR filter was obtained as,

$$F_{1b}(z)=\frac{10.4003+1.2023z^{-1}+1.2886z^{-2}+1.2688z^{-3}+0.8358z^{-4}}{6.6517\,10^{4}+z^{-1}+z^{-2}-6.4114\,10^{3}\,z^{-3}-5.9733\,10^{4}\,z^{-4}} \tag{14}$$

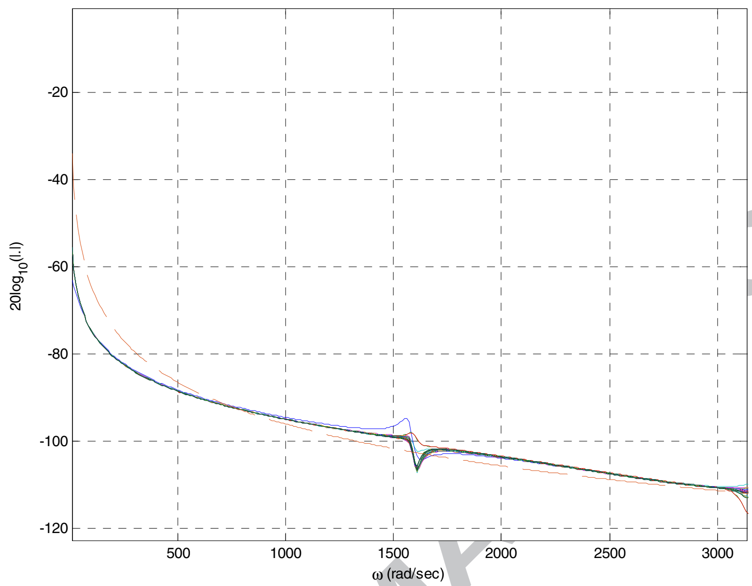


Figure 5: Amplitude response of filters for various $\alpha \in [0.1,3]$ with 0.1 increment by using fixed random number set.

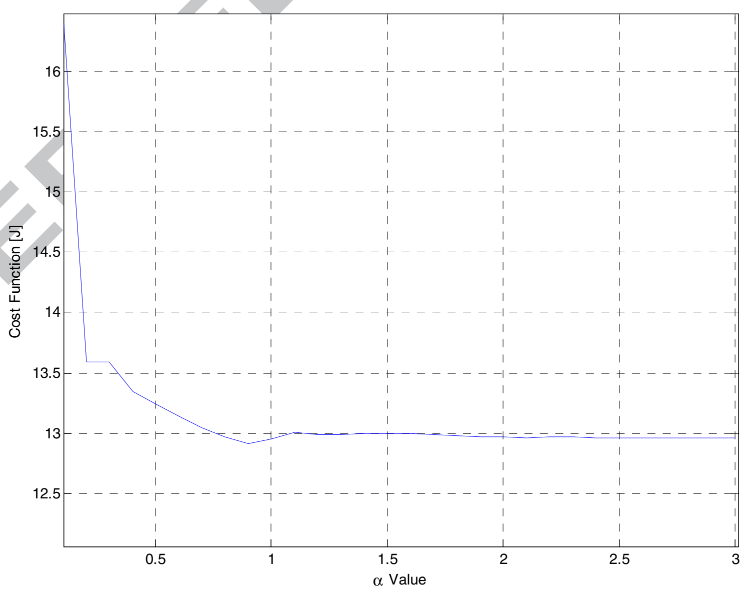


Figure 6: Cost values of HB-FODPSO for various $\alpha \in [0.1,3]$ with 0.1 increment by using fixed random number set.

Table 1: List of cost values for various $a \in [0.1,3]$ with 0.1 increment by using fixed rand number set.

a Value	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
J	16.4570	13.5916	13.5853	13.3433	13.2387	13.1343	13.0428	12.9687	12.9167	12.9514
a Value	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2**
J	13.0038	12.9881	12.9882	12.9943	12.9941	12.9934	12.9896	12.9774	13.9730	12.9735
a Value	**2.1**	**2.2**	**2.3**	**2.4**	**2.5**	**2.6**	**2.7**	**2.8**	**2.9**	**3**
J	12.9642	12.9646	12.9664	12.9582	12.9597	12.9595	12.9593	12.9592	12.9590	12.9598

In order to test by free random number set, Figure 7 shows the amplitude responses of discrete IIR filters obtained for $a \in [0.1, 3.0]$ by 0.1 increments. Here, free random numbers are generated randomly for each test, and results in Figure 7 demonstrate that the performance of the proposed optimization algorithm is robust. Here, the dashed line indicates the frequency response of original continuous fractional order function $F_1(s)$. To observe effects of fractional order on the optimization performance of HB-FODPSO, cost values of the HB-FODPSO obtained for various fractional orders $a \in [0.1,3.0]$ are compared in Figure 8. Table 2 lists these values of cost function and it indicates the best performance at $a = 2.8$. The resulting discrete IIR filter was found for $a = 2.8$ as,

$$F_{1c}(z) = \frac{1.1988 + 2.2971z^{-1} + 2.9429z^{-2} + 1.7334z^{-3} + 0.5961z^{-4}}{1.0044\,10^5 - 2.4048\,10^3 z^{-1} - 229.0755z^{-2} - 9.7804z^{-3} + z^{-4}} \tag{15}$$

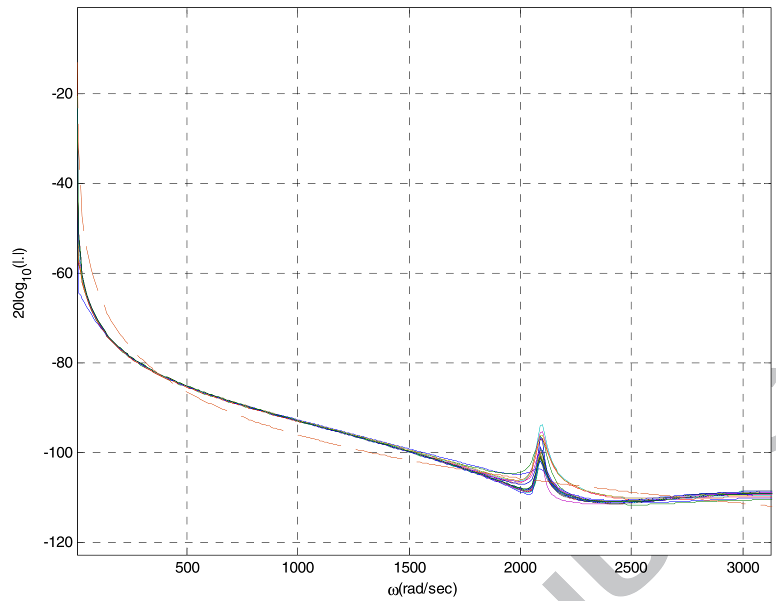


Figure 7: Amplitude response of filters for various $\alpha \in [0.1,3]$ with 0.1 increment by using free random number set.

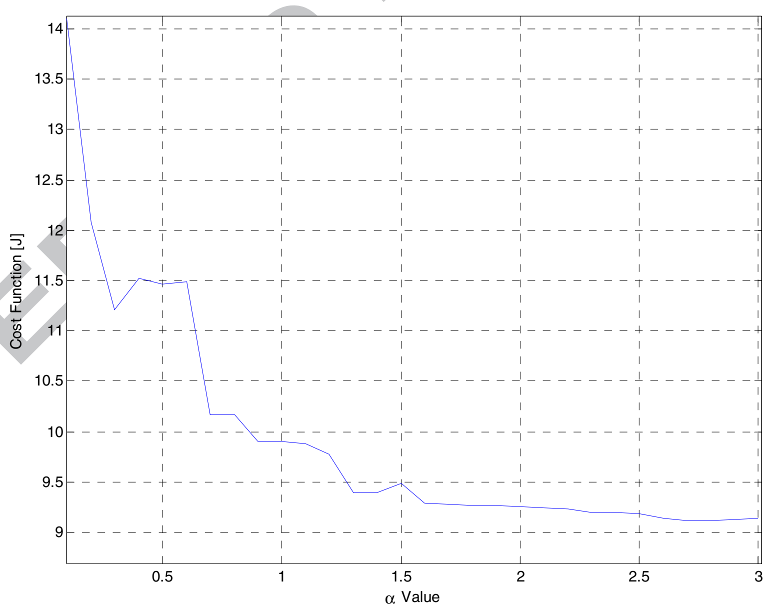


Figure 8. Cost values of HB-FODPSO for various $\alpha \in [0.1,3]$ with 0.1 increment by using free random number set.

Table 2. List of cost values for various $\alpha \in [0.1, 3]$ with 0.1 increment by using free random number set.

α Value	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**	**1**
J	14.1072	12.0751	11.2046	11.5169	11.4602	11.4822	10.1656	10.1642	9.9048	9.9057
α Value	**1.1**	**1.2**	**1.3**	**1.4**	**1.5**	**1.6**	**1.7**	**1.8**	**1.9**	**2**
J	9.8770	9.7740	9.3956	9.3885	9.4805	9.2841	9.2766	9.2649	9.2687	9.2581
α Value	**2.1**	**2.2**	**2.3**	**2.4**	**2.5**	**2.6**	**2.7**	**2.8**	**2.9**	**3**
J	9.2431	9.2280	9.2014	9.1966	9.1904	9.1405	9.1209	9.1124	9.1295	9.1391

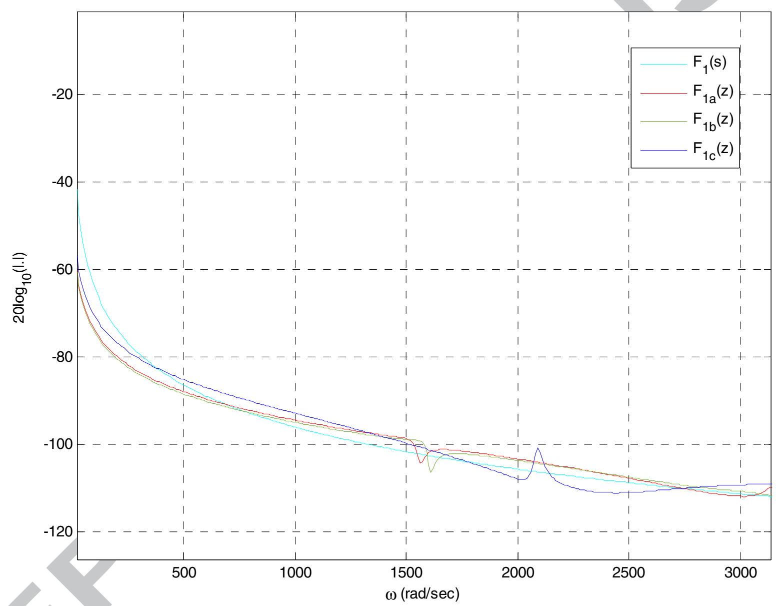


Figure 9: Comparisons of amplitude responses of functions $F_1(s)$, $F_{1a}(z)$, $F_{1b}(z)$ and $F_{1c}(z)$.

Figure 9 shows comparison of amplitude responses for original continuous filter $F_1(s)$ and the IIR filter approximations $F_{1a}(z)$, $F_{1b}(z)$ and $F_{1c}(z)$ in the same figure. We observed that the results are consistent for all tests and concluded that the proposed algorithm can be effective for practical filter design applications.

In order to better evaluate performance of the method, we also compared results of HB-FODPSO method with the results of “nidiz” Matlab toolbox and stochastic perturbation methods. The nidiz toolbox is a collection of analytical filter design functions, which was developed fractional order system design. The stochastic perturbation optimization is a fundamental heuristic optimization approach, which was developed for random walk searching. It has been reported that a cooperation of nidiz toolbox and stochastic perturbation method can improve discrete IIR filter design for fractional order filters (Ates et al, 2016). Results of nidiz toolbox are taken as initial solution candidates and these candidate solutions can be further improved by stochastic perturbation method. The nidiz toolbox yielded the following filter function for this example.

$$F_{1d}(z)=\frac{1}{0.1913\,10^{6}-0.6121\,10^{6}\,z^{-1}+0.9793\,10^{6}\,z^{-2}-1.2486\,10^{6}\,z^{-3}+1.4885\,10^{6}\,z^{-4}} \tag{16}$$

Stochastic perturbation method with nidiz initialization found the following filter function for this example.

$$F_{1e}(z)=\frac{17.9914+15.0044\,z^{-1}+18.7161\,z^{-2}+11.1775\,z^{-3}+6.7076\,z^{-4}}{0.1913\,10^{6}-0.6121\,10^{6}\,z^{-1}+0.9793\,10^{6}\,z^{-2}-1.2486\,10^{6}\,z^{-3}+1.4885\,10^{6}\,z^{-4}} \tag{17}$$

Figure 10 compares result of HB-FODPSO method ($F_{1a}(z)$), results of nidiz toolbox ($F_{1d}(z)$) and results of stochastic perturbation method ($F_{1e}(z)$). In the figure, HB-FODPSO method provides better approximation to amplitude response of original continuous filter function $F_1(s)$ than the nidiz toolbox and the stochastic perturbation method. Another advantage of HB-FODPSO method is that it does not need good initial solution candidates. It starts by a random set of initial candidates. whereas, stochastic perturbation method needs results of nidiz toolbox to provide a good approximation to original filter.

Figure 11 shows results of 30 tests of HB-FODPSO for the same setting. This test is carried out to show consistency of the proposed algorithm. Average value of cost function is 16.2340 with a standard deviation of 6.7150. All resulting filters were stable and acceptable to implement.

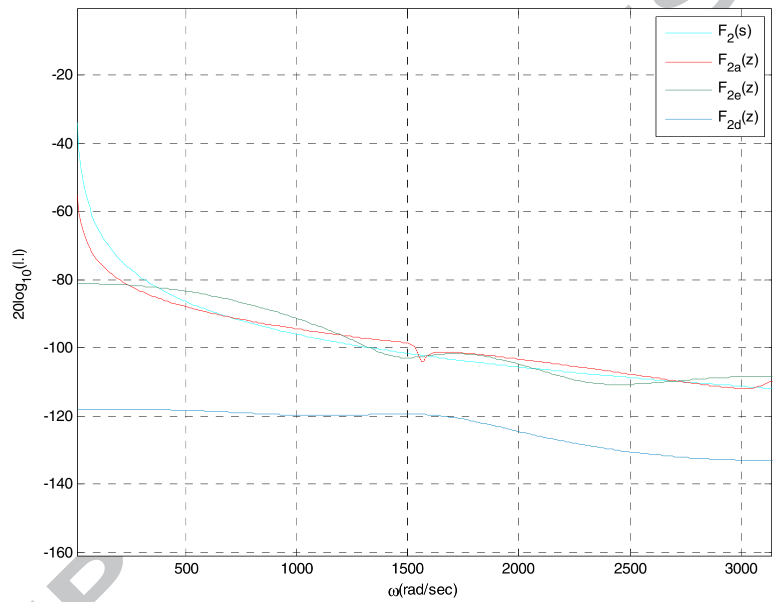


Figure 10: Comparisons of amplitude responses of functions $F_1(s)$, $F_{1a}(z)$, $F_{1d}(z)$ and $F_{1e}(z)$.

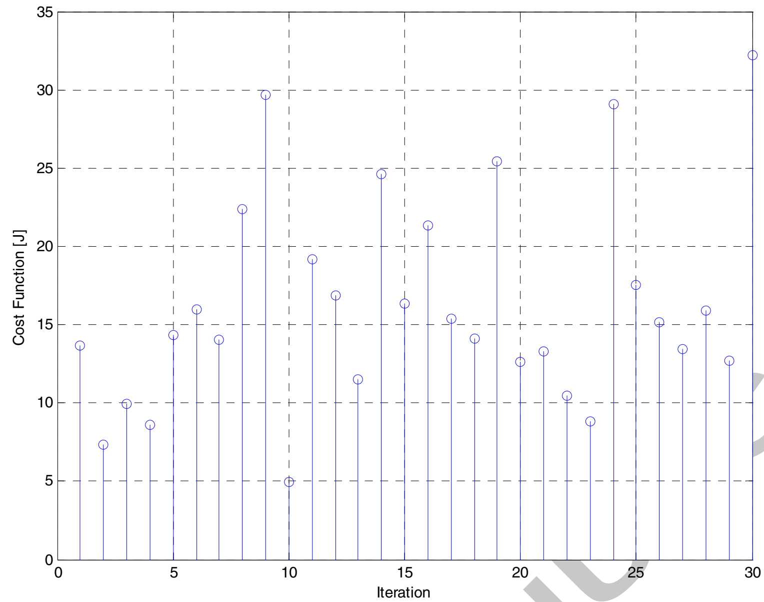


Figure 11: Statistical analysis of HB-FODPSO optimization process for Example 1

Example 2: Let us design an IIR discrete filter that approximates to the fractional order continuous filter function $F_2(s) = \dfrac{1}{s^{0.8} - 1.4s^{0.4} - 1}$ in the frequency range of $(0, 3140)$ rad/sec with a unit frequency increment of 10 rad/sec. The sampling frequency for digital IIR filter was taken $T_s = 0.001$ sec.

In order to check stability of continuous filter $F_2(s)$, the stability check procedure based on the root placement analysis in the first Riemann sheet (Radwan et al., 2009; Alagoz et al., 2015; Senol et al., 2014) was applied. Figure 12(a) shows that the root in the first Riemann sheet satisfies the stability condition of characteristic root arguments. Figure 12(b) also validates the stability of the filter function $F_2(s)$ by step response.

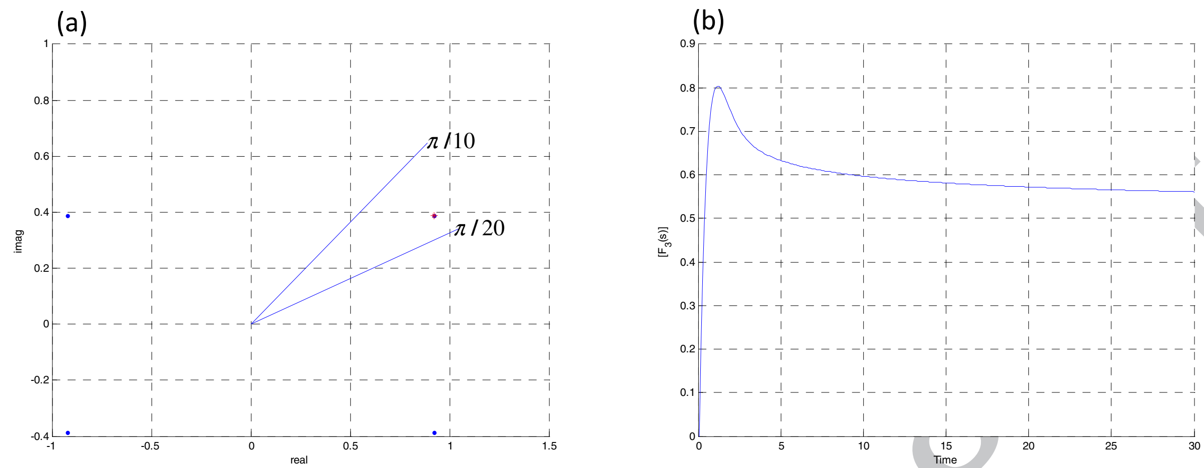


Figure 12. (a) Root placement analysis for $F_2(s)$, (b) Step response of closed loop transfer function for $F_2(s)$.

The time evolution of the cost function of HB-FODPSO for the case $\alpha = 1$ is shown in Figure 13. The figure shows that the algorithm converges to an optimal solution. At the end of iterations, the optimized discrete IIR filter function was obtained as,

$$F_{2a}(z) = \frac{00.9711 + z^{-1} + 0.9201z^{-2} + 0.6111z^{-3} - 0.0306z^{-4}}{1256.1769 + 9.4322z^{-1} - 12.9107z^{-2} - 81.7992z^{-3} - 164.861z^{-4}} \tag{18}$$

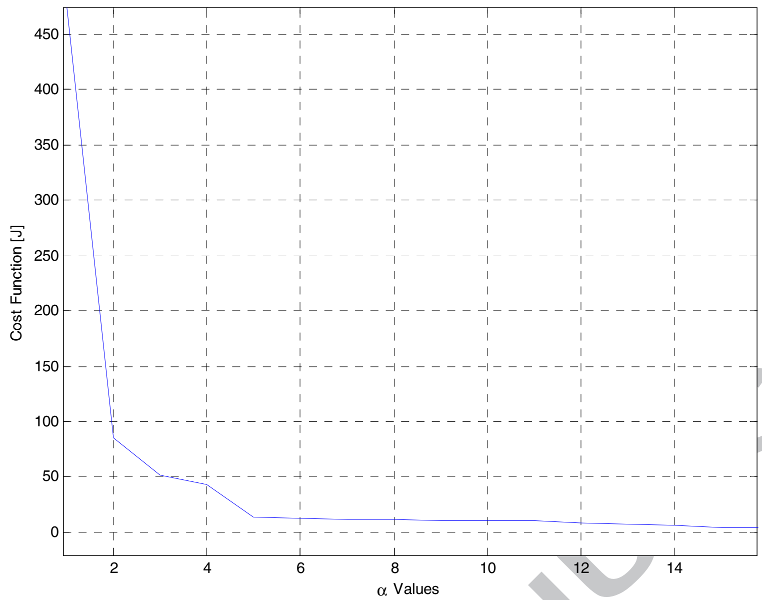


Figure 13: Change of cost function during HB-FODPSO for $F_{2a}(z)$

Figure 14 shows the amplitude responses of discrete IIR filters obtained for $\alpha \in [0.1, 3.0]$ with 0.1 increments for the fixed random number set. The dashed line indicates the frequency response of continuous fractional order function $F_2(s)$. In order to better see impacts of fractional order particle velocity on the optimization performance of HB-FODPSO, we compared the values of the cost function for various fractional order $\alpha \in [0.1,3.0]$ in Figure 15. The figure indicates that, for $\alpha > 1$, the values of the cost function do not exhibit significant change. The best performance for the fixed random number set was obtained at the order $\alpha = 1.2$ and the corresponding discrete IIR filter was found as,

$$F_{2b}(z) = \frac{1.089 + z^{-1} + 0.9028z^{-2} + 0.6574z^{-3} - 0.0336z^{-4}}{286.2704 - 19.4177z^{-1} - 4.738z^{-2} - 66.7652z^{-3} + -175.051z^{-4}} \tag{19}$$

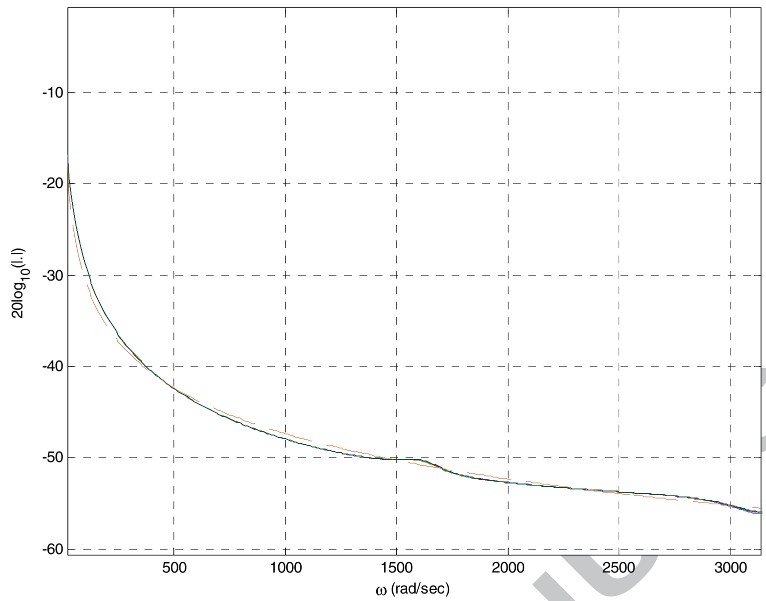


Figure 14: Amplitude response of filters for various $\alpha \in [0.1,3]$ with 0.1 increment by using fixed random number set.

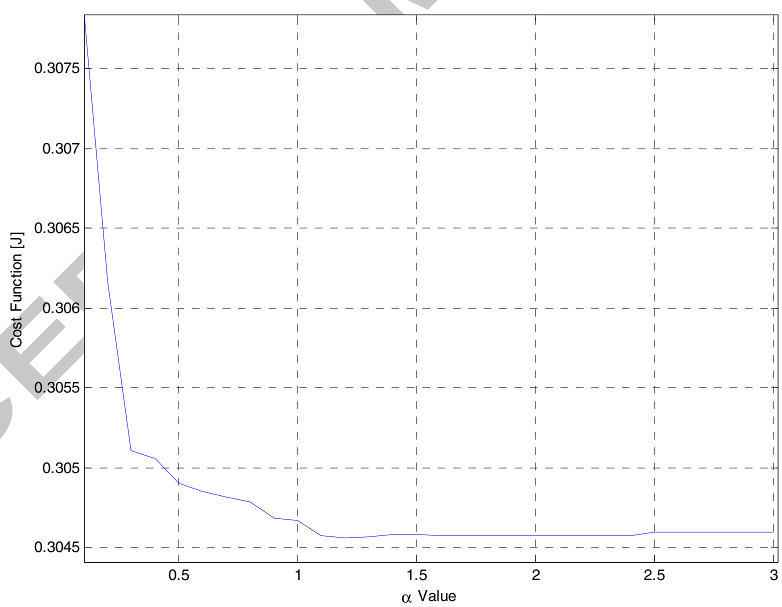


Figure 15: Cost values of HB-FODPSO for various $\alpha \in [0.1,3]$ with 0.1 increment by using fixed random number set.

Figure 16 shows the amplitude responses of discrete IIR filters obtained for $\alpha \in [0.1, 3.0]$ with 0.1 increments and free random number set. The dashed line indicates the amplitude response of continuous fractional order function $F_2(s)$. The effects of fractional order on the optimization performance of the HB-FODPSO are illustrated in Figure 17. When $\alpha > 1.0$, the value of the cost function decreases and the best performance is found at $\alpha = 3.0$. The resulting discrete IIR filter was obtained as,

$$F_{2c}(z) = \frac{-0.0204 + 0.6678z^{-1} + 0.7482z^{-2} + z^{-3} + 1.0673z^{-4}}{279.449 - 13.5838z^{-1} - 37.4491z^{-2} - 30.5846z^{-3} - 181.4794z^{-4}} \quad (20)$$

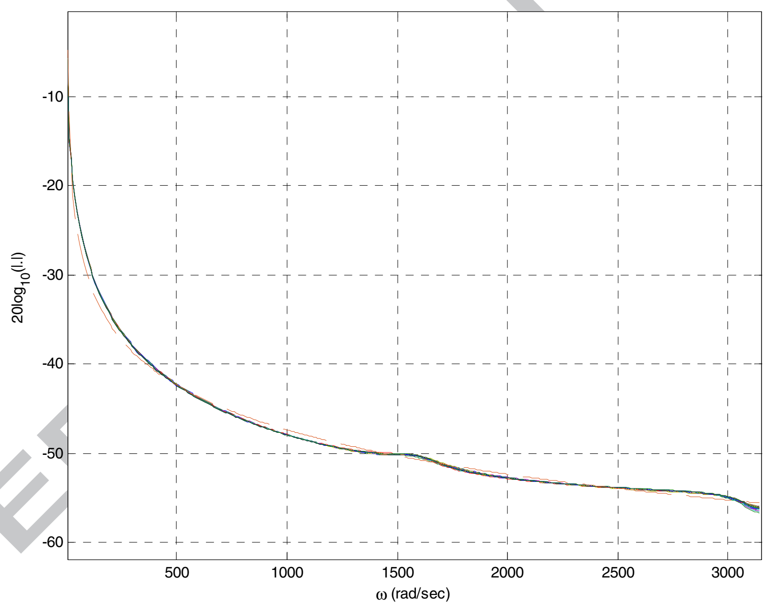


Figure 16: Amplitude responses of filters for various $\alpha \in [0.1, 3]$ with 0.1 increment by using free random number set.

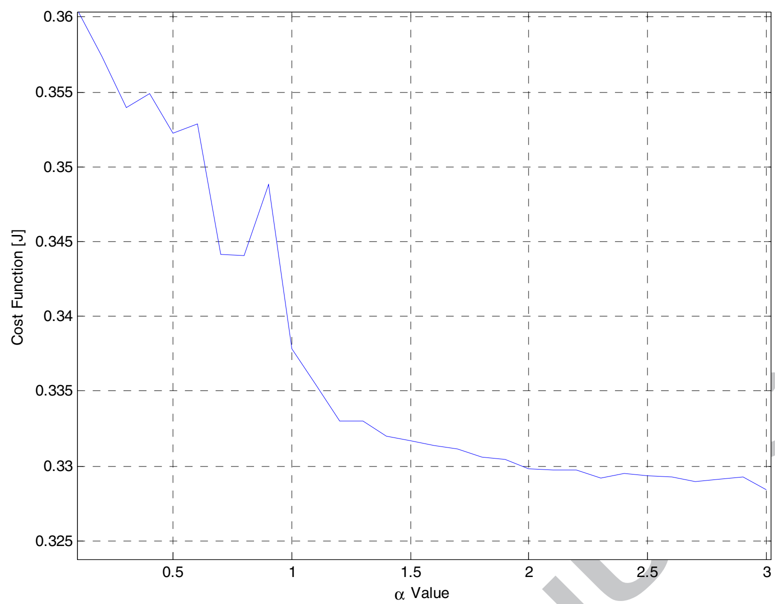


Figure 17: Cost values of HB-FODPSO for various $\alpha \in [0.1,3]$ with 0.1 increment by using free random number set

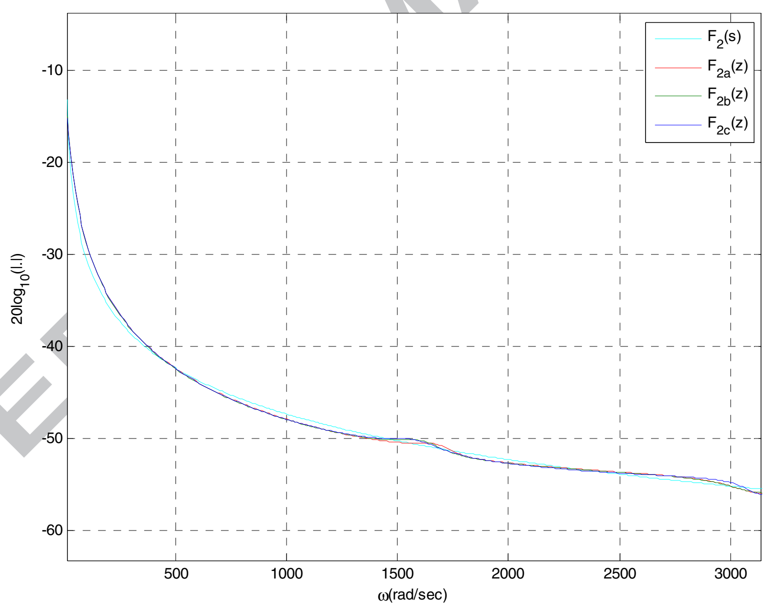


Figure 18: Comparisons of amplitude responses of functions $F_2(s)$, $F_{2a}(z)$, $F_{2b}(z)$ and $F_{2c}(z)$

Figure 18 shows comparison of amplitude responses for $F_2(s)$, $F_{2a}(z)$, $F_{2b}(z)$ and $F_{2c}(z)$. We observed that results are very consistent and discrete IIR filter approximations are useful for digital system applications.

It is also useful to compare the result of HB-FODPSO with the results of other analytical and heuristic optimization methods.

The nidiz toolbox yielded the following filter function for this example.

$$F_{2d}(z) = \frac{1}{409.0669 - 676.3294z^{-1} + 550.4325z^{-2} - 521.5055z^{-3} + 485.8786z^{-4}} \tag{21}$$

Stochastic perturbation method with nidiz initialization found the following filter function by using equation (21) as an initial design.

$$\mathrm{F}_{2e}(\mathrm{z}) = \frac{2.0790 - 0.2055\mathrm{z}^{-1} + 1.7995\mathrm{z}^{-2} - 0.8037\mathrm{z}^{-3} + 0.5467\mathrm{z}^{-4}}{410.1525 - 678.0050\mathrm{z}^{-1} + 551.5572\mathrm{z}^{-2} - 522.4457\mathrm{z}^{-3} + 486.4424\mathrm{z}^{-4}} \tag{22}$$

Results are compared in Figure 19. In the figure, HB-FODPSO method ($\mathrm{F}_{2a}(\mathrm{z})$) yields much better results than the nidiz toolbox, and the stochastic perturbation method. One of the reasons is that, as complexity of fractional order filter such as number of term and fractional order may severely affect the approximation performance of nidiz toolbox. Increasing complexity of transfer function of continuous filter can degrade approximation performance of nidiz toolbox because it is not flexible to adapt itself according to problem complexity. In this example, the stochastic perturbation method cannot yield satisfactory results because initial candidate solution from nidiz toolbox was not good enough to improve its solution. Figure 20 shows results of 30 HB-FODPSO tests for the same setting. Statistical performance evaluation shows that average value of cost function is 1.1142 with a standard deviation of 1.0350. The HB-FODPSO method also exhibits better results for this example.

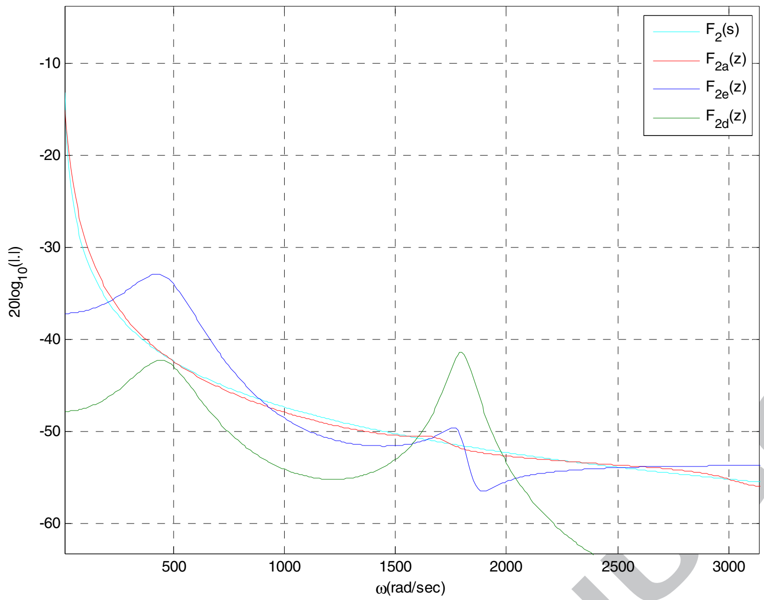


Figure 19: Comparisons of amplitude responses of functions $F_2(s)$, $F_{2a}(z)$, $\mathrm{F}_{2\mathrm{d}}(\mathrm{z})$ and $\mathrm{F}_{2\mathrm{e}}(\mathrm{z})$

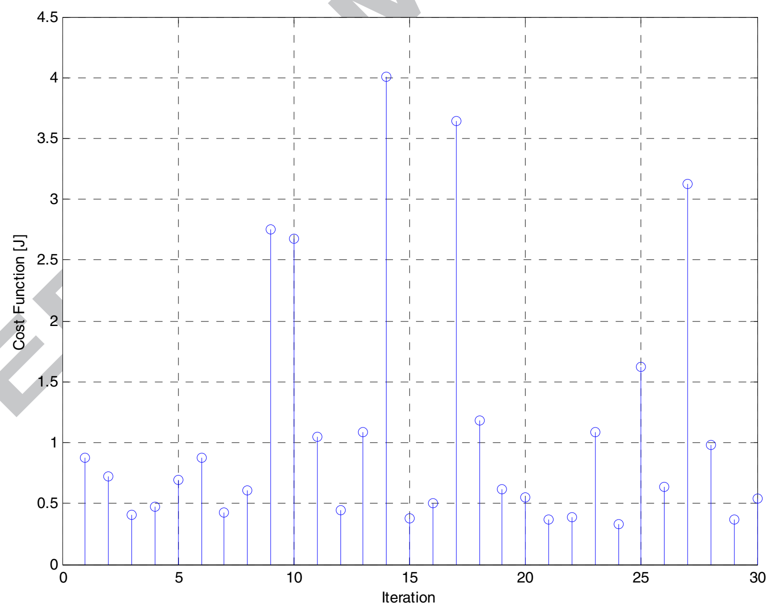


Figure 20: Statistic analysis of HB_FODPSO optimization process for Example 2

4.1 Comparisons with analytical discretization methods:

This section presents a comparison study. Oustaloup's method (Oustaloup, 1991; Oustaloup et. al, 2000), CFE method (Monje et al., 2010) and nidiz toolbox are a collection of well-known analytical approximation methods, which are commonly used for comparison purposes. We compare results of these methods obtained for discrete filter approximation to fractional order Chebyshev lowpass filters, which are given in the form of Eq. 23 (Freeborn et. al, 2015).

$$F_f(s) = \frac{a_0}{a_1 s^{1+\alpha} + a_2 s^{\alpha} + 1} \tag{23}$$

Stable IIR filters approximating to the continuous fractional order Chebyshev lowpass filter function are designed in the frequency range of $(0, 3140)$ rad/sec with a unit frequency increment of 10 rad/sec. Sampling frequency for digital IIR filter was taken $T_s = 0.001$ sec. The filter parameters are $\alpha = 0.2$, $a_0 = 1, a_1 = 3$ and $a_2 = 5$, which leads to following fractional order filter function:

$$F_3(s) = \frac{1}{3s^{1,2} + 5s^{0.2} + 1} \tag{24}$$

FO-HBDPSO method found the following filter function for the transfer function in Eq. 24,

$$F_{3a}(z) = \frac{0.2136 + 0.8818\, z^{-1} + 0.8947\, z^{-2} + 0.9268\, z^{-3} + 0.672\, z^{-4}}{1.017 * 10^{4} - 120.8\, z^{-1} + z^{-2} - 637.9\, z^{-3} - 9402\, z^{-4}} \tag{25}$$

The Oustaloup's method yielded the following filter function for this fractional order Chebyshev lowpass filter.

$$F_{3b}(z) = \frac{\begin{array}{c} 3.333 * 10^{-7} + 7.251 * 10^{-5} z^{-1} - 0.0002189\, z^{-2} + 0.0002287\, z^{-3} \\ - 9.224 * 10^{-5}\, z^{-4} + 9.604 * 10^{-4} z^{-5} - 1.088 * 10^{-10}\, z^{-6} \end{array}}{1 - 4.376\, z^{-1} + 7.578\, z^{-2} - 6.478\, z^{-3} + 2.726\, z^{-4} - 0.45\, z^{-5} + 5.881 * 10^{-8}\, z^{-6}} \tag{26}$$

The following filter function is generated by using CFE method.

$$F_{3c}(z) = \frac{\begin{array}{l}-0.000656 + 0.005389\, z^{-1} - 0.01936\, z^{-2} + 0.03973\, z^{-3} - 0.05095\, z^{-4} \\ + 0.04179\, z^{-5} - 0.02142\, z^{-6} + 0.006272\, z^{-7} - 0.0008031\, z^{-8}\end{array}}{\begin{array}{l}1 - 8.105\, z^{-1} + 28.74\, z^{-2} - 58.2\, z^{-3} + 73.66\, z^{-4} - 59.66\, z^{-5} \\ + 30.19\, z^{-6} - 8.729\, z^{-7} + 1.104\, z^{-8}\end{array}} \qquad (27)$$

The nidiz toolbox yielded the following filter function for this fractional order Chebyshev lowpass filter.

$$F_{3d}(z) = \frac{1}{2.746*10^4 - 6.586*10^4\, z^{-1} + 7.902*10^4\, z^{-2} - 8.517*10^4\, z^{-3} + 9.061*10^4\, z^{-4}} \qquad (28)$$

Figure 21 shows comparisons of amplitude responses for original fractional order Chebyshev lowpass filter $F_3(s)$ and the IIR filter approximations results of HB-FODPSO $F_{3a}(z)$, the result of Oustolop function $F_{3b}(z)$, the result of CFE method $F_{3c}(z)$ and the result of nidiz toolbox $F_{3d}(z)$ in the same figure. We observed that although the proposed method has implemented with 4th order filter function, which is less than others, results are comparable with other methods using even higher order filter function.

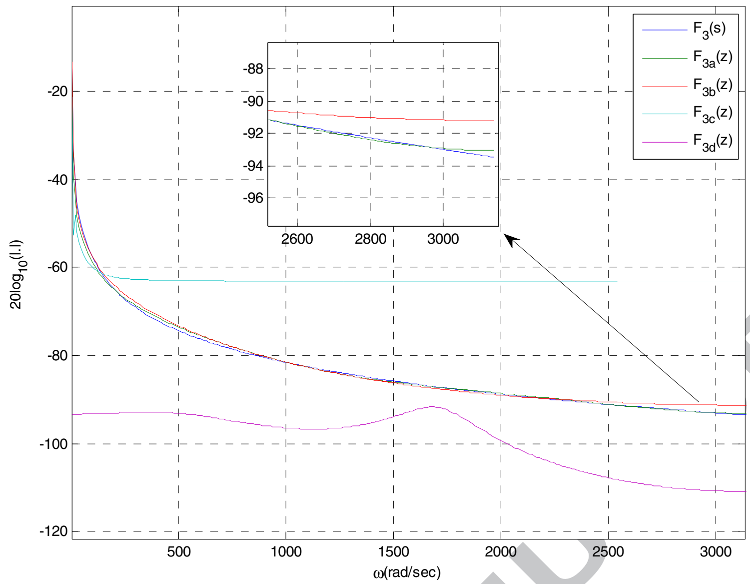


Figure 21: Comparisons of amplitude responses of functions $F_3(s)$, $F_{3a}(z)$, $F_{3b}(z)$, $F_{3c}(z)$, $F_{3d}(z)$

Table 3 shows comparisons of H_2 norm of HB-FODPSO with other well known analytical methods.

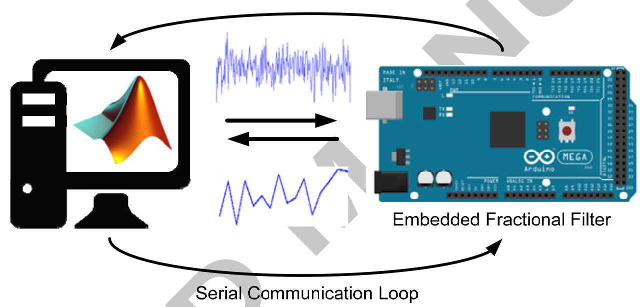
Table 3. Comparisons H_2 norm of HB FODPSO with other well known analytical methods

	HB-FODPSO Method	Oustaloup's Method	CFE Method	Nidiz Toolbox
H_2 Norm	$1.66*10^2$	$2.152*10^2$	$1.55*10^5$	$9.83*10^4$

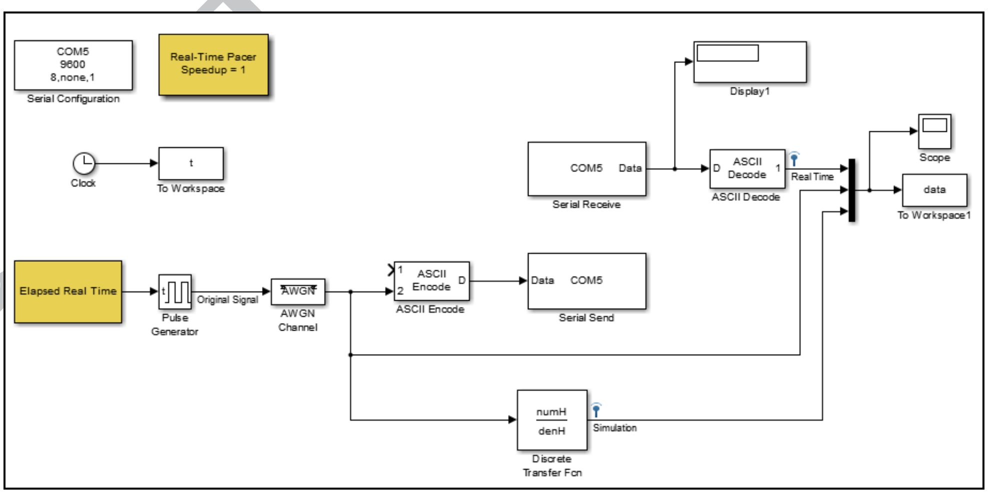
5. Experimental Study

This section presents an experimental study illustrating the discrete realization of fractional order filter functions by low-cost control cards. The optimized IIR filter function approximating to the amplitude response of an original fractional order filter by HB-FODPSO was embedded to Arduino control cards. Arduino cards provide a low-cost and practical microcontroller solution. Nowadays, it can be used by academic and industrial communities for educational and experimental purposes. In our experimental study, we used a basic Arduino microcontroller card including 8-bit AVR Atmel Microcontroller with 16/32/64 KB in system programmable flash.

This card has up to 16 MIPS throughput at 16MHz. Figure 22 shows pictures of our test platform including Arduino card and Matlab Simulink environment. In the test process, the optimized IIR filter design obtained by proposed HB-FODPSO algorithm was expressed in difference equation form in discrete time, and we generate embedded code according to direct-form I implementation of the IIR filter function. Then, this code was uploaded to Arduino card. Afterward, Matlab simulink environment was used to send test input signals from the PC to digital input ports of the card. Then, the output of the embedded digital filter implementation was captured from the digital output ports of the card (Kavuran et al., 2016).



(a)



(b)

Figure 22. a) Schematic diagram of experimental system; b) Simulink model of the system.

The difference equation used for embedded code generation is written for the sampled time ($n = 1,2,3..,\ t = nT_s$) as follows,

$$y[n] = \frac{1}{d_0}\left(\sum_{i=0}^{p_n} c_i u(n-i) - \sum_{i=1}^{q_d} d_i y(n-i)\right) \tag{29}$$

In this section, we implement the fractional order continuous filter function $F_r(s) = \frac{20}{s^{0.8}+20}$ and test it for white noise filtering application. The optimized IIR filter function by using proposed HB-FODPSO algorithm was obtained as follows:

$$F_r(z) = \frac{1.0129 + z^{-1} + 0.9012z^{-2} + 0.7677z^{-3} - 0.0056z^{-4}}{15.0156+1.4109z^{-1} + 0.7735z^{-2} - 0.607z^{-3} - 10.1106z^{-4}} \tag{30}$$

This IIR filter function was implemented on Arduino microcontroller card as the embedded code for real time applications. Besides, it was also implemented in simulink environment as the simulation model. Figure 23 shows the filter output from Arduino microcontroller card and the filter output from the Matlab simulink simulation for the same input signal. The input signal of the digital filter is a 30dB white Gaussian noise. The figure demonstrates that both filter implementations can suppress the white noise~~,~~ effectively. We observed that results of simulation (Simulink) and results of real time embedded filter application (Arduino microcontroller card) are also consistent. This indicates that the discrete implementation of fractional order filters on a low-cost Arduino microcontroller card with restricted word length can present a similar filtering performance with the discrete implementation of fractional order filters in the Matlab Simulink environment on a PC.

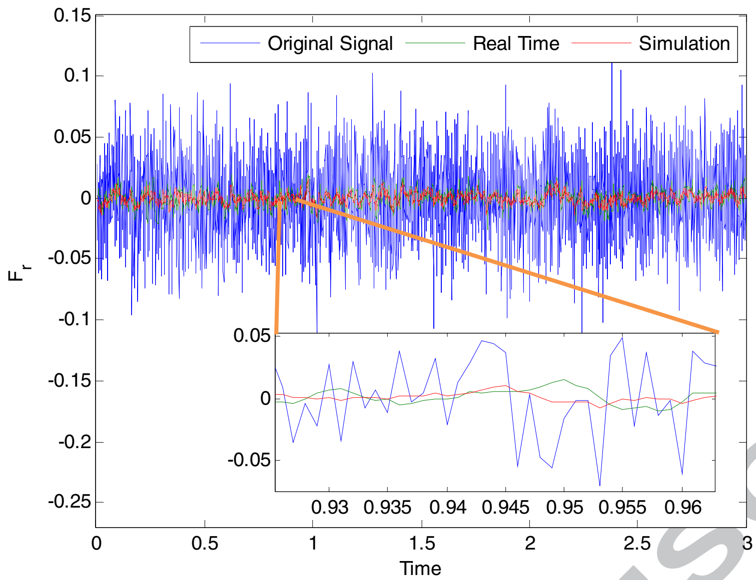


Figure 23. The filter outputs from Arduino microcontroller card and from the simulation for the additive white Gaussian noise.

6. Conclusion

In summary, this paper demonstrates a discrete IIR filter design algorithm based on a combination of FODPSO and BaOA algorithms for the digital realization of fractional order continuous filters. To provide better fitting of the amplitude response of discrete IIR filter design to the amplitude response of fractional order continuous filters, the conventional FODPSO method was modified to increase the search range of particles by using the candidate solution selection mechanism of BaOA. For designed filter stability, proposed algorithm confines search region into stable filter design region of IIR filters. These modifications provide two major advantages: firstly it expands the search range of the FODPSO to have a better access to spreading solution points, where very low and very high filter coefficients may be needed for approximation to fractional order filter function, secondly, initial particles are generated from stable discrete filter region of the search space of filter coefficients to obtain stable optimized IIR digital filter solutions. As a consequence, these modifications contribute to discrete IIR filters design for engineering applications of fractional order filters.

Illustrative design examples were presented to evaluate effectiveness of the proposed method. In these examples, fractional order dependence of the HB-FODPSO algorithm was also investigated relevantly by using a fixed random number set in our tests. In these analyses, the best order providing the lowest cost value was found and the best filter design were compared with other analytical and heuristic filter design methods. Moreover, an experimental study was also presented to validate realization of the optimized IIR filter design in real digital systems.

Acknowledgement

This study is supported by The Scientific and Technological Research Council of Turkey (TÜBİTAK) with 215E261 project number.

7. References

Al-Alaoui, M.A. 2009. "Discretization methods of fractional parallel PID controllers". 16th International Conf. on Electronics, Circuits, and Systems, IEEE, pp. 327-330. DOI: 10.1109/ICECS.2009.5411002.

Alagoz, B.B., Yeroglu, C., Senol, B. and Ates, A. 2015. "Probabilistic robust stabilization of fractional order systems with interval uncertainty". ISA transactions 57: 101-110. DOI: 10.1016/j.isatra.2015.01.003.

Ates, A., Kavuran, G., Alagoz, B. B. and Yeroglu, C. 2016. "Improvement of IIR filter discretization for fractional order filter by discrete stochastic optimization". 39th International Conference on Telecommunications and Signal Processing (TSP), IEEE, pp. 583-586. DOI: 10.1109/TSP.2016.7760948.

Baranowski, J., Bauer, W., Zagórowska, M. and Piątek, P. 2016. "On Digital Realizations of Non-integer Order Filters." 35 (6): 2083-2107. DOI: 10.1007/s00034-016-0269-8

Chen, Y.Q. and Moore, K. L. 2002. "Discretization schemes for fractional-order differentiators and integrators. IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications". 49 (3): 363-367. DOI: 10.1109/81.989172.

Chen, Y.Q., Petras, I. and Xue, D. 2009. "Fractional order control-a tutorial, American Control Conference". ACC '09, pp. 1397-1411. DOI: 10.1109/ACC.2009.5160719.

Chen, Y.Q., Vinagre, B.M. and Podlubny, I. 2004. "Continued fraction expansion approaches to discretizing fractional order derivatives-an expository review". Nonlinear Dynamics, 38 (1-4): 155-170. DOI: 10.1007/s11071-004-3752-x.

Chen, Y.Q. and Vinagre, B. M. 2003. "A new IIR-type digital fractional order differentiator". Signal Processing 83(11): 2359-2365. DOI: 10.1016/S0165-1684(03)00188-9.

Couceiro, M. and Ghamisi, P. 2015. Fractional Order Darwinian Particle Swarm Optimization: Applications and Evaluation of an Evolutionary Algorithm. Springer. DOI: 10.1007/978-3-319-193680_1.

Darwin, C. 1872. "On the origin of species by means of natural selection, or the preservation of favored races in the struggle for life". London: Public Domain Books.

Das, S. 2008. "Functional Fractional Calculus for System Identification and controls". Springer. DOI: 10.1515/auto-2014-1083.

Das, S., Majumder, B., Pakhira, A., Pan, I., Das, S. and Gupta, A., 2011: Optimizing continued fraction expansion based IIR realization of fractional order differ-integrators with genetic algorithm, 2011 International Conference on Process Automation, Control and Computing (PACC), IEEE, pp. 1-6. DOI: 10.1109/PACC.2011.5979043.

Freeborn, T., Maundy, B. and Elwakil, A.S. 2015. “Approximated fractional order chebyshev lowpass filters”. Hindawi Publishing Corporation Mathematical Problems in Engineering, 2015 (2015): 1-7. DOI: 10.1155/2015/832468.

Gupta, M., Varshney, P., Visweswaran, G.S. 2011. “Digital fractional-order differentiator and integrator models based on first order and higher order operators”. International Journal of Circuit Theory and Applications 39 (5): 461-174. DOI: 10.1002/cta.650.

Kavuran G., Alagoz, B.B., Ateş A. and Yeroğlu C. 2016. “Implementation of Fractional Order Filters by Embedded Programming Techniques”. International Conference on Fractional Differentiation and its Applications ICFDA’16.

Kennedy, J. and Eberhart, R. 1995. “A new optimizer using particle swarm theory”. Proceedings of the IEEE 6th International Symposium on Micro Machine and Human Science, pp. 39–43. DOI: 10.1109/MHS.1995.494215.

Krishna, B.T. 2011. “Studies on fractional order differentiators and integrators: a survey.” Signal Processing, 91 (3): 386-426. DOI: 10.1016/j.sigpro.2010.06.022.

Maione, G. 2008. “Continued fractions approximation of the impulse response of fractional-order dynamic systems”. IET Control Theory and Applications 2(7): 564-572. DOI: 10.1049/iet-cta:20070205.

Monje, C.A., Chen, Y. Q., Vinagre, B.M., Xue, D., Feliu, V. 2010. “Fractional-order systems and controls”. Fundamentals and applications. Advances in Industrial Control, Springer, London.

Ogata, K. 2010. Modern Control Engineering. Prentice Hall, Boston.

Oustaloup, A. 1991. “La commande CRONE: commande robuste d’ordre non entire”. Hermes.

Oustaloup, A., Levron, F., Mathieu, B., Nanot, F.M. 2000. "Frequency-band complex noninteger differentiator: characterization and synthesis". IEEE Trans. Circuits Syst. I Fundam. Theory Appl. 47 (1): 25–39. DOI: 10.1109/81.817385.

Podlubny, I., Petras, I., Vinagre, B.M., O'Leary, P. and Dorcak, L. 2002. "Analogue realization of fractional-order controllers". Nonlinear Dynamics 29 (1-4): 281-296. DOI: 10.1023/A:1016556604320.

Radwan, A.G., Soliman, A.M. and Elwakil, A.S. 2008. "First-order filters generalized to the fractional domain". Journal of Circuits, Systems, and Computers 17 (1): 55-66. DOI: 10.1142/S0218126608004162

Radwan, A.G., Soliman, A.M., Elwakil, A.S. and Sedeek, A. 2009. "On the stability of linear systems with fractional-order elements". Chaos, Solitons & Fractals 40 (5): 2317-2328. DOI: 10.1016/j.chaos.2007.10.033.

Rana, K. P. S., Kumar, V., Garg, Y. and Nair, S. S. 2016. "Efficient Design of Discrete Fractional-Order Differentiators Using Nelder–Mead Simplex Algorithm". Circuits, Systems, and Signal Processing 35 (6): 2155-2188. DOI: 10.1007/s00034-015-0149-7.

Ren, J., Sun, Z.Z., Dai, W., Cole, K.S. and Cole, R.H. 2016. "New approximations for solving the caputo-type fractional partial differential equations". Applied Mathematical Modeling 40 (4): 2625–2636. DOI: 10.1016/j.apm.2015.10.011.

Santamaria, G.E., Valverde, J.V., Perez-Aloe, R. and Vinagre, B.M. 2008. Microelectronic implementations of fractional order integro differential operators". ASME Journal of Computational and Nonlinear Dynamics 3 (2): 1-6. DOI:10.1115/1.2833907

Salem, S.A. 2012. “BOA: A novel optimization algorithm”. In IEEE 2012 International Conference on Engineering and Technology, 1-5. DOI: 10.1109/ICEngTechnol.2012.6396156.

Senol, B., Ates, A., Alagoz, B.B. and Yeroglu, C. 2014. “A numerical investigation for robust stability of fractional-order uncertain systems”. ISA transactions 53 (2): 189-198. DOI: 10.1016/j.isatra.2013.09.004.

Senol, B., “Fractional Graphical Analysis Tool (FGATool)”, www.fgatool.com, (Last Avaliable: 01.02.2017).

Tillett, J., Rao, T.M., Sahin, F., Rao, R., Brockport, S., 2005: Darwinian particle swarm optimization. In B. Prasad Proceedings of the 2nd Indian International Conference on Artificial Intelligence, pp. 1474–1487.

Tsai, W. 2002. “Social structure of “coopetition” within a multiunit organization: coordination, competition, and intraorganizational knowledge sharing”. Organization Science, 13 (2): 179– 190. DOI: 10.1287/orsc.13.2.179.536.

Tseng, C.C., 2007: “Design of FIR and IIR fractional order Simpson digital integrators”. Signal Processing 87 (5): 1045-1057. DOI:10.1016/j.sigpro.2006.09.006.

Valerio, D. and Sa da Costa, J. 2005. “Time-domain implementation of fractional order controllers”. IEEE Proceedings-Control Theory and Applications 152 (5): 539-552. DOI: 10.1049/ip-cta:20045063.

Vinagre, B.M., Chen, Y.Q. and Petras, I. 2003. “Two direct Tustin discretization methods for fractional-order differentiator/integrator”. Journal of the Franklin Institute 340 (5): 349-362. DOI: 10.1016/j.jfranklin.2003.08.001.

Visweswaran, G.S., Varshney, P., Gupta, M. 2011. " New approach to realize fractional power in z-domain at low frequency". IEEE Transactions on Circuits and Systems-II: Express Briefs 58 (3): 179-183. DOI: 10.1109/TCSII.2011.2110350.

Zhe, G., and Liao, X. 2012. "Rational approximation for fractional-order system by particle swarm optimization". Nonlinear Dynamics 67 (2): 1387-1395. DOI: 10.1007/s11071-011-0075-6.

Zhu-Zhong, Y. and Ji-Liu, Z. 2008, "An improved design for the IIR type digital fractional order differential filter, International Seminar on Future BioMedical Information Engineering". pp. 473-476. DOI: 10.1109/FBIE.2008.39.