IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Journal Européen des Systèmes Automatisés
Vol. 53, No. 4, August, 2020, pp. 559-566
Journal homepage: http://iieta.org/journals/jesa

Design of Two-Dimensional Recursive Digital Filter Using Multi Particle Swarm Optimization Algorithm

Lakhdar Kaddouri*, Amel B.H. Adamou-Mitiche, Lahcene Mitiche

Laboratoire de Recherche Modélisation Simulation et Optimisations des Systèmes Complexes Réels, Université ZIANE Achour de Djelfa, Djelfa 17000, Algeria

Corresponding Author Email: kaddourilakhdar@gmail.com

https://doi.org/10.18280/jesa.530415

Received: 11 June 2020
Accepted: 3 August 2020

Keywords:
2D filter, recursive filters, optimization, multi-PSO, stability

ABSTRACT

Particle Swarm Optimization (PSO) is an evolutionary algorithm widely used in optimization problems. It is characterized by a fast convergence, which can lead the algorithm to stagnate in local optima. In the present paper, a new Multi-PSO algorithm for the design of two-dimensional infinite impulse response (IIR) filters is built. It is based on the standard PSO and uses a new initialization strategy. This strategy is relayed to two types of swarms: a principal and auxiliaries. To improve the performance of the algorithm, the search space is divided into several areas, which allows a best covering and leading to a better exploration in each zone separately. This solved the problem of fast convergence in standard PSO. The results obtained demonstrate the effectiveness of the Multi-PSO algorithm in the filter coefficients optimization.

1. INTRODUCTION

The applications of two-dimensional (2-D) digital signal processing have been developed rapidly in recent decades. This led to a growing interest in the design of 2-D filters due to a variety of applications in different domains such as digital image, medical data, artificial vision, radar and sensor data processing [1].

The main functions of the filter are to remove unwanted parts of a signal, such as random noise, or to extract useful parts. For example, an image degraded by wideband noise, can be improved without blurring its edges. The magnetic record can be made more readable by suppressing certain signal of large amplitude and low frequency [2].

The digital filters are typically classified into two groups: infinite impulse response (IIR) or recursive filter and finite impulse response (FIR) or non-recursive filter [3]. The FIR filters are generally easier to implement because they are non-recursive and always stable (by definition). On the other hand, it is much more difficult to obtain linear phase responses and to control overall frequency responses with IIR filters. However, very sharp and narrow transition band frequency responses can be easily realized with IIR filters. this feature makes them suitable for a broad range of applications. Generally, the IIR filters are capable of having more accurate responses than those yielded by FIR filters [4]. But the IIR filters suffer from their instability problem. To remedy this problem, some researchers have adopted certain solutions as revealed in literature [5-8].

To solve the instability problem in any IIR design, important optimization-based methods have been emerged especially, those based on evolutionary techniques (ETs), [4, 9-11], where the main idea of optimization-based methods is the design of the two-dimensional digital IIR filters considered as a constrained minimization problem.

Among these methods, the particle swarm optimization (PSO) appears as a powerful tool in optimization problems. It uses only two mathematical equations with primitive operators and is conceptually very simple. Inspired from the animal social behavior evolving in swarms like fishes and flights [12], PSO shows an ease of implementation in comparison with the genetic algorithm (GA), and it has fewer parameters to adjust. It has already been applied successfully in many application areas, including function optimization [13, 14]. However, several studies [15] showed some problems in controlling the balance between exploration (global investigation of the search place) and exploitation (the fine search a round a local optimum). This is why, considerable efforts have addressed this problem [4, 10, 11].

In this paper, a novel Multi-PSO algorithm is utilized for the design of two-dimensional IIR filters. it is based on Standard PSO and characterized by the diversity of particles without decreasing the convergence speed and it uses a new initialization strategy. This strategy is relayed to two types of swarms: the main swarm denoted by S1, and s auxiliary ones denoted by S2i, where $1 \leq i \leq s$. To enhance the performance of the algorithm, the search space is partitioned into several zones. Then, each initialization of S2i is performed in one of these zones. This partitioning allows a best covering of the search space, thus leading to a better exploration in each zone separately. The final table with all the best auxiliary swarm positions will compose the initial population of the main swarm. Once the main swarm is initialized, the auxiliary swarms are no longer used. The main swarm takes over and continues the search in the same way as the standard PSO algorithm until a stop criterion is reached.

2. FORMULATION OF THE DESIGN PROBLEM

Consider a two-dimensional IIR filter with its transfer function, $H(z_1, z_2)$ being in the form [6]:

$$H(z_1,z_2)=A\prod_{k=1}^{K}\frac{\sum_{j=0}^{J_2{}^{(k)}}\sum_{i=0}^{J_1{}^{(k)}}b_{ij}^{(k)}z_1^{-i}z_2^{-j}}{\sum_{j=0}^{I_2{}^{(k)}}\sum_{i=0}^{I_1{}^{(k)}}d_{ij}^{(k)}z_1^{-i}z_2^{-j}} \qquad (1)$$

where,

- $H(z_1, z_2)$: consists of a cascade of K filter sections,
- $b_{ij}^{(k)}$ and $d_{ij}^{(k)}$, $k=\overline{1,K}$: are the unknown coefficients of each section,
- A is a constant gain that may be left arbitrary during the design procedure,
- $J_1{}^{(k)}, J_2{}^{(k)}, I_1{}^{(k)} and\ I_2{}^{(k)}$ are integers, they must be chosen between 0 and 2 in order insure the stability.

The design task of 2-D filters is to find the transfer function coefficients $H(z_1, z_2)$ as in Eq. (1), such that the magnitude function $M(\omega_1,\omega_2)=\left|H(e^{-j\omega_1},e^{-j\omega_2})\right|$ approximates the desired amplitude response $M_d(w_1,w_2)$ in some optimum sense [4].

Thus, the design of a 2-D recursive filter is equivalent to the following constrained minimization problem [1]:

$$Minimize\ error=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}(M(\omega_1,\omega_2)-M_d(\omega_1,\omega_2))^2 \qquad (2)$$

where,

- N_1 and N_2 are sampling numbers of the signal on two axes
- $\omega_i=n_i\frac{\pi}{N_i}\ (i=1{,}2)$.

In this study, the model filter is defined as:

$$M_d(\omega_1,\omega_2)=\begin{cases}1 & if \quad \sqrt{\omega_1^2+\omega_2^2}\le 0.08\pi\\ 0.5 & if\ \ 0.08\pi\le\sqrt{\omega_1^2+\omega_2^2}\le 0.12\pi\\ 0 & otherwise.\end{cases} \qquad (3)$$

with $N_1=N_2=50$ [4, 5].

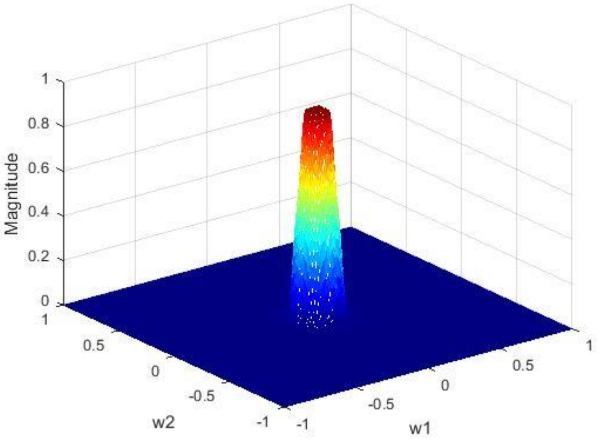


Figure 1. Desired amplitude response M_d of the 2-D filter

To solve the stability problem the work in [5] is exploited and a polynomial of the following form is used

$$D(s_1,s_2)=\sum_{j=0}^{I_2}\sum_{i=0}^{I_1}d_{ij}s_1^i\,s_2^j \qquad (4)$$

where,

(s_1,s_2) are two variables of the Laplace transform and (I_2, I_1) are two integers equal to 1 or 2. The coefficients d_{ij} are expressed by nonzero real variables $[q_l]$. The relationship between the terms d_{ij} and the parameters $[q_l]$ is expressed by the analog network (Contains n_1 capacitors $c_1=s_1$ and n_2 capacitors $c_2=s_2$) as show in Figure 2 [16].

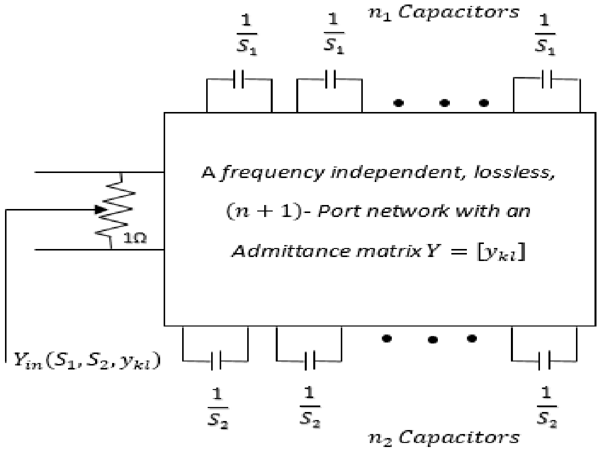


Figure 2. Lossless and frequency-independent network

If the network is lossless and independent of frequency, its admittance Y is real and can be expressed as [5]:

$$Y=\left[\begin{array}{c|cccc}0 & y_{12} & y_{13} & \cdots & y_{1n}\\ \hline -y_{12} & 0 & y_{23} & \cdots & y_{2n}\\ -y_{13} & -y_{23} & 0 & \cdots & y_{3n}\\ \vdots & \vdots & \vdots & & \cdots\\ -y_{1n} & -y_{2n} & -y_{3n} & \cdots & 0\end{array}\right]=\left[\begin{array}{c|c}Y_{11} & Y_{12}\\ \hline -Y_{12}^T & Y_{22}\end{array}\right] \qquad (5)$$

The network admittance, which terminate by a resistor on one end, and n_1+n_2 capacitors on the other, can be expressed as follows [5]:

$$\hat{Y}(s_1,s_2,y_{kl})=Y+\ diag\left[1\ \ \overbrace{s_1\ \cdots\ s_1}^{n_1}\ \ \overbrace{s_2\ \cdots\ s_2}^{n_2}\right] \qquad (6)$$

According to the network's theory, the input admittance can be deduced as:

$$Y_{in}(s_1,s_2,y_{kl})=1+\frac{Y_{12}adj\left[\hat{Y}_{22}(s_1,s_2,y_{kl})\right]Y_{12}^T}{det\left[\hat{Y}_{22}(s_1,s_2,y_{kl})\right]} \qquad (7)$$

The determinant of the admittance matrix $[Y_{in}(s_1,s_2,y_{kl})]$ is the polynomial $D(s_1,s_2,y_{kl})$ Because of the lossless network terminated by a resistor, the $D(s_1,s_2,y_{kl})$ is a strictly Hurwitz polynomial of two variables s_1 and s_2 , for any set of real parameters values: $Y_{in}(s_1,s_2,y_{kl})$; $1 < k < l \leq n$ .It means that the obtained polynomial $D(s_1,s_2,y_{kl})$ is the determinant of the admittance matrix of a lossless and frequency-independent network terminated by capacitors s_1, s_2 and an input resistor which satisfies the condition:

$$D(s_1, s_2, y_{kl}) \neq 0 \text{ For } Re(s_1) \geq 0 \text{ and } Re(s_2) \geq 0.$$

The obtained above result is illustrated in the network of three terminals, which ends with a capacitor s_1, capacitor s_2 and resistor R = 1 Ohm, as shown in Figure 3. The admittance matrix is [16]:

$$\hat{Y}(s_1, s_2, y_{kl}) = \begin{bmatrix} 1 & y_{12} & y_{13} \\ -y_{12} & s_1 & y_{23} \\ -y_{13} & -y_{23} & s_2 \end{bmatrix} \tag{8}$$

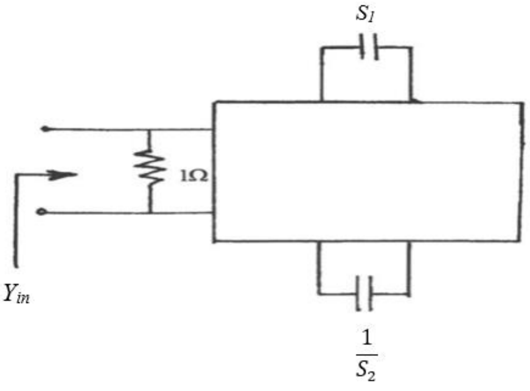


Figure 3. Network of three terminals

The determinant of $\hat{Y}(s_1, s_2, y_{kl})$ is [12]:

$$\det(\hat{Y}(s_1, s_2, y_{kl})) = y_{23}^2 + y_{13}^2 s_1 + y_{12}^2 s_2 + s_1 s_2 \tag{9}$$

and also:

$$D(s_1, s_2) = \sum_{j=0}^{I_2} \sum_{i=0}^{I_1} d_{ij} s_1^i s_2^j = d_{00} + d_{10}s_1 + d_{01}s_2 + d_{11}s_1 s_2 \tag{10}$$

By comparing (9) and (10), the following result is obtained:

$$d_{00} = y_{23}^2 \text{ , } d_{10} = y_{13}^2 \text{ , } d_{01} = y_{12}^2 \text{ et } d_{11} = 1.$$

Put $y_{23}=q_1$; $y_{13}=q_2$; and $y_{12}=q_3$.
The coefficients d_{ij} are expressed by the parameters q_i.
Case I: $I_1=I_2=1$, thus,
$d_{00} = q_1^2,$
$d_{10} = q_2^2;$
$d_{01} = q_3^2,$
$d_{11} = 1.$

Let's consider a lossless network terminated by two capacitors s_1, one capacitor s_2, and one input resistor ($n_1=2$, $n_2=1$) [16].

$$\hat{Y}(s_1, s_2, y_{kl}) = \begin{bmatrix} 1 & y_{12} & y_{13} & y_{14} \\ -y_{12} & s_1 & y_{23} & y_{24} \\ -y_{13} & y_{23} & s_1 & y_{34} \\ -y_{14} & -y_{24} & -y_{34} & s_2 \end{bmatrix} \tag{11}$$

By taking:

$y_{12}=q_1$, $y_{13}=q_2$, $y_{14}=q_3$, $y_{23}=q_4$, $y_{24}=q_5$ and $y_{34}=q_6$, the coefficients d_{ij} can be deduced as:

Case II ($I_1=2$ and $I_2=1$), thus,

$$\begin{aligned} d_{00} &= (q_1q_6 - q_2q_5 + q_3q_4)^2 \\ d_{10} &= q_5^2 + q_6^2 \\ d_{20} &= q_3^2 \\ d_{01} &= q_4^2 \\ d_{11} &= q_1^2 + q_2^2 \\ d_{21} &= 1 \end{aligned}$$

Similarly, when $I_1=2$ and $I_2=2$, we have [16]:

$$\begin{aligned} d_{00} &= (q_5q_{10} + q_6q_9 + q_7q_8)^2, \\ d_{10} &= (q_1q_{10} - q_3q_7 + q_4q_6)^2 + (q_2q_{10} - q_3q_9 + q_4q_8)^2, \\ d_{20} &= q_{10}^2, \\ d_{01} &= (q_1q_8 - q_2q_6 + q_3q_5)^2 + (q_1q_9 - q_2q_7 + q_4q_5)^2, \\ d_{11} &= q_6^2 + q_7^2 + q_8^2 + q_9^2, \\ d_{21} &= q_3^2 + q_4^2, \\ d_{02} &= q_5^2, \\ d_{12} &= q_1^2 + q_2^2, \\ d_{22} &= 1, \end{aligned}$$

It is worth noting, that when $D(s_1, s_2)$ satisfies the stability conditions, the $D(z_1, z_2)$, deduced by the bilinear transformation, satisfies the stability conditions [16]:

$$D(s_1, s_2)\Big|_{s_1=\frac{(1-z_1^{-1})}{(1+z_1^{-1})}, s_2=\frac{(1-z_2^{-1})}{(1+z_2^{-1})}} = D(z_1, z_2) \neq 0 \quad for \;\; |z_1| \geq 1 \;\; et \;\; |z_2| \geq 1. \tag{12}$$

This means that [16], the polynomial $D(z_1, z_2)$ has not zeros outside the region:

$$\{|z_1| \cup |z_2|; |z_1| < 1 \;\; and \; |z_2| < \; 1\}. \tag{13}$$

For this investigation, two types of transfer function $H(z_1,z_2)$ will be studied:

2.1 General transfer function of the form

$$H(z_1, z_2) = h_0\Big(\frac{a_{00} + a_{01}Z_2 + a_{02}Z_2^2 + a_{10}Z_1 + a_{20}Z_1^2}{(1 + b_1Z_1 + C_1Z_2 + d_1Z_1Z_2)(1 + b_2Z_1 + C_2Z_2 + d_2Z_1Z_2)} + \frac{a_{11}Z_1Z_2 + a_{12}Z_1Z_2^2 + a_{21}Z_1^2Z_2 + a_{22}Z_1^2Z_2^2}{(1 + b_1Z_1 + C_1Z_2 + d_1Z_1Z_2)(1 + b_2Z_1 + C_2Z_2 + d_2Z_1Z_2)}\Big) \tag{14}$$

the magnitude $M(\omega_1, \omega_2)$ is given as:

$$M(\omega_1, \omega_2) = |H(z_1, z_2)| \tag{15}$$

with: $z_1 = e^{-j\omega_1}$ et $z_2 = e^{-j\omega_2}$
then

$$M(\omega_1, \omega_2) = H_0 \left| \frac{A_R - jA_I}{(B_{1R} - jB_{1I})(B_{2R} - jB_{2I})} \right| \tag{16}$$

with:

$$\begin{aligned} A_R = a_{00} &+ a_{01}\cos(\omega_2) + a_{02}\cos(2\omega_2) + a_{10}\cos(\omega_1) \\ &+ a_{20}\cos(2\omega_1) + a_{11}\cos(\omega_1 + \omega_2) \\ &+ a_{12}\cos(\omega_1 + 2\omega_2) + a_{21}\cos(2\omega_1 \\ &+ \omega_2) + a_{22}\cos(2\omega_1 + 2\omega_2) \end{aligned}$$

$$\begin{aligned} A_I = a_{01}\sin(\omega_2) + a_{02}\cos(2\omega_2) + a_{10}\sin(\omega_1) \\ + a_{20}\sin(2\omega_1) + a_{11}\sin(\omega_1 + \omega_2) \\ + a_{12}\sin(\omega_1 + 2\omega_2) + a_{21}\sin(2\omega_1 \\ + \omega_2) + a_{22}\sin(2\omega_1 + 2\omega_2) \end{aligned}$$

$$B1_R = 1 + b_1\cos(\omega) + c_1 cos(\omega_2) + d_1 cos(\omega_1 + \omega_2)$$

$$B1_I = b_1\sin(\omega_1) + c_1 sin(\omega_2) + d_1 sin(\omega_1 + \omega_2)$$

$$B2_R = 1 + b_2\cos(\omega_1) + c_2 cos(\omega_2) + d_2 cos(\omega_1 + \omega_2)$$

$$B2_I = b_2\sin(\omega_1) + c_2 sin(\omega_2) + d_2 sin(\omega_1 + \omega_2)$$

The optimized vector x given as:

$$x(\, a_{01}\, a_{02}\, a_{10}\, a_{11}\, a_{12}\, a_{20}\;\, a_{21}\, a_{22}\, b_1\, c_1\, d_1\, b_2\, c_2 d_2\, H_0).$$

2.2 Transfer function with numerator as mirror image polynomial

A two-dimensional filter given by its transfer function:

$$H(z_1, z_2) = \frac{N(z_1, z_2)}{D(z_1, z_2)} \tag{17}$$

The polynomial N(z₁, z₂) is expressed by the form [16]:

$$N(z_1, z_2) = \begin{bmatrix} 1 & z_1^{-1} & z_1^{-2} \end{bmatrix} \begin{bmatrix} b_{00}^{(k)} & b_{01}^{(k)} & b_{02}^{(k)} \\ b_{10}^{(k)} & b_{11}^{(k)} & b_{10}^{(k)} \\ b_{02}^{(k)} & b_{01}^{(k)} & b_{00}^{(k)} \end{bmatrix} \begin{bmatrix} 1 \\ z_2^{-1} \\ z_2^{-2} \end{bmatrix} \tag{18}$$

It satisfies the conditions:

$$b_{i,j} = b_{-i,-j} \text{ and } b_{-i,j} = b_{i,-j}.$$

In this case of J_1=J_2=2:

$$H(z_1, z_2) = H_0\left(\frac{z_1^{-1} z_2^{-1}\left(b_{11} + b_{01}(z_1 + z_2^{-1}) + b_{10}(z_2 + z_2^{-1})\right)}{D_1 \cdot D_2} + \frac{z_1^{-1} z_2^{-1}\left(b_{02}(z_1 z_2^{-1} + z_2 z_1^{-1}) + b_{00}(z_1 z_2 + z_1^{-1} z_2^{-1})\right)}{D_1 \cdot D_2}\right) \tag{19}$$

where,

$$D_1 = 1 + b_1 z_1^{-1} + C_1 z_2^{-1} + d_1 z_1^{-1} z_2^{-1}$$
$$D_2 = 1 + b_2 z_1^{-1} + C_2 z_2^{-1} + d_2 z_1^{-1} z_2^{-1}$$

The magnitude $M(\omega_1, \omega_2)$ is given as:

$$M(\omega_1, \omega_2) = H_0 \left| \frac{A_r}{(B_{1r} + jB_{1i})(B_{2r} + jB_{2i})} \right| \tag{20}$$

with:
$A_r = b_{11} + 2b_{01}\cos(\omega_1) + 2b_{10}\cos(\omega_2) + 2b_{02}\cos(\omega_1 - \omega_2) + 2b_{00} cos(\omega_1 + \omega_2)$,
$B_{1r} = 1 + b_1\cos(\omega_1) + c_1\cos(\omega_2) + d_1\cos(\omega_1 + \omega_2)$,
$B_{1i} = b_1\sin(\omega_1) + c_1\sin(\omega_2) + d_1\sin(\omega_1 + \omega_2)$,
$B_{2r} = 1 + b_2\cos(\omega_1) + c_2\cos(\omega_2) + d_2\cos(\omega_1 + \omega_2)$,
$B_{2i} = b_2\sin(\omega_1) + c_2\sin(\omega_2) + d_2\sin(\omega_1 + \omega_2)$.

The optimized vector x is then [16]

$$x(b_{00}\;\; b_{01}\;\; b_{02}\;\, b_{10}\;\; b_{11}\;\; b_1\;\; c_1\;\; d_1\;\; b_2\;\; c_2\;\; d_2\;\; H_0).$$

3. PSO ALGORITHM

Particle Swarm Optimization (PSO) is an evolutionary algorithm that uses a population of candidate particles, to develop an optimal solution to the problem.

Proposed in 1995 by Eberhart and Kennedy [12, 17]. It was inspired from the social behavior of the animals evolving in swarms such as fish and birds. In fact, these animals have relatively complex displacement dynamics, whereas each individual has a limited "intelligence" and only a local knowledge of its position in the swarm. The local information and the memory of each individual are used to decide its displacement.

At the beginning of the algorithm, the swarm particles are randomly initialized in the searching space of the problem. Then, at each iteration, every particle moves with combining of the two linear components cited below. At the iteration t + 1, the velocity and position vectors are calculated from Eqns. (21) and (22), respectively:

$$v_{i,j}^{t+1} = w * v_{i,j}^t + C_1 * r_1\left(pbest_{i,j}^t - x_{i,j}^t\right) + C_2 * r_2\left(gbest_{i,j}^t - x_{i,j}^t\right) \tag{21}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \qquad j \in \{1,2,3, \dots, D\} \tag{22}$$

where,

w: a constant, called the inertia coefficient.

C_1 and C_2: two constants, called acceleration coefficients.

r_1 and r_2: two random numbers, chosen uniformly from the interval [0,1].

In a searching space of dimension D, the particle i of the swarm is modelled by its position vector $\overrightarrow{x_i}\,(x_{i1}, x_{i2}, x_{i3}, x_{i4}, \dots \dots \dots, x_{iD})$ and its velocity vector $\overrightarrow{v_i}\,(v_{i1}, v_{i2}, v_{i3}, v_{i4}, \dots \dots \dots, v_{iD})$. The quality of its position is determined by the value of the objective function at this point. The particle remembers the best local position, by which it has already passed, is denoted by $\vec{P}best(pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$.

The best global position [18], reached by the swarm's particles, is denoted by: $\vec{G}best(gbest_{i1}, gbest_{i2}, \dots, gbest_{iD})$, referring to the general version of PSO, where all of the particles in the swarm are considered to be neighbors to the particle i.

Some improvements have been made to the basic algorithm, especially in terms of divergence control. In particular, the inclusion of the Vmax parameter that makes it able to limit the divergence of the particles. In addition, other studies have been conducted on the dynamics of particles, attempted to analyze the algorithm when the convergence conditions of the swarm are insured [19, 20].

The combination between the parameters w, c_1 and c_2 adjusts the balance between the diversification phases and intensification of the searching process [14]. Clerc and Kennedy [21] have shown that good convergence can be obtained by making these parameters dependent. The use of a constriction coefficient χ (or constriction factor) makes it possible to better control the divergence of the swarm and to get rid of the definition of Vmax. This variant of PSO, which has been widely used in the literature, is known as the canonical PSO. Using the constriction coefficient, Eq. (21) becomes:

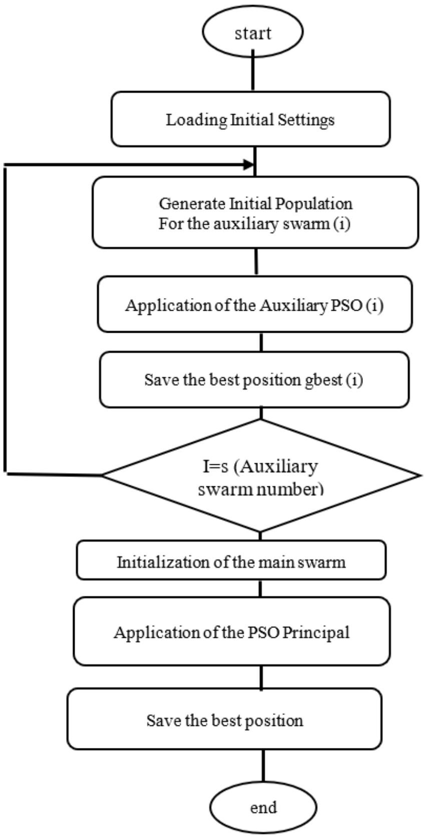
$$v_{i,j}(t+1) = \chi * (w * v_{i,j}(t) + \phi_1 * r_1 * \left(pbest_{i,j} - x_{i,j}(t)\right) + \phi_1 * r_1 * \left(gbest_{i,j} - x_{i,j}(t)\right)) \quad (23)$$

with: $\chi = \frac{2}{\emptyset - 2 + \sqrt{\emptyset^2 - 4\emptyset}}$ and $\emptyset = \emptyset_1 + \emptyset_2$, $\emptyset > 4$

3.1 The Multi-PSO algorithm

Multi-PSO is a new variant of the original Particle Swarm Optimization algorithm, in which two ideas were added; The first idea is to use two types of swarms, a principal and auxiliaries. The second idea is the partition of the search space into several zones. in another meaning, the algorithm is based on Standard PSO with a new initialization strategy. This strategy is based on the use of two kinds of swarms: the main swarm denoted by S1, and s auxiliary ones denoted by S2i, where 1 ≤ i ≤ s [18]. In the Multi-PSO algorithm, every auxiliary swarm is intended to discover an area in the search space by performing a number of iterations, its role ends when the best position of the zone is reached. More precisely, the populations of the auxiliary swarms are initialized randomly in the different zones, according to a partition already made in the searching space. Then, the particles of each swarm perform n predefined generations in their area and, at the end, the best-found position is saved in a vector array. The best obtained positions by all auxiliary swarms are therefore grouped in a vector of size equal to the number of swarms initialized and also to the number of zones. The all best positions will then compose the initial population of the main swarm. Once the main swarm is initialized, the auxiliary swarms are no longer used. The main swarm continues the search by the same way as the standard PSO algorithm until a stop criterion is reached.

3.2 Flowchart of the used program


start
Loading Initial Settings
Generate Initial Population
For the auxiliary swarm (i)
Application of the Auxiliary PSO (i)
Save the best position gbest (i)
I=s (Auxiliary swarm number)
Initialization of the main swarm
Application of the PSO Principal
Save the best position
end

3.3 Proposed algorithm

1: Input model filter (M_d) as given in (3).
2: Start the program and load the initial global settings.
3: Generation of the initial population of the auxiliary swarm(i).
4: Execution of auxiliary PSO with K iterative periods (update of velocity $\overrightarrow{v_i}$ and position $\overrightarrow{x_i}$ vectors in Eqns. (21) and (22) respectively and take the best global position).
5: Save the best position of auxiliary swarm(i) in gbest (i).
6: If the auxiliary swarm(i) is not the last one then increases i and go to 3.
7: Initialization of the main swarm by all best positions of auxiliary swarms.
8: Execution of the main PSO (update of velocity $\overrightarrow{v_i}$ and position $\overrightarrow{x_i}$ vectors in Eqns. (21) and (22) respectively and take the best global position).
9: Save the global best position.
10: Output the coefficients and show frequency magnitude response of the optimized filter.

4. RESULTS OF SIMULATION AND DISCUSSION

For the simulation of the program. The Multi-PSO algorithm parameters are presented in the Table 1:

Table 1. Parameters of the Multi-PSO algorithm

Parameters	Value
Number of auxiliary swarms	10
Population size of auxiliary swarm	150
Iterative periods of auxiliary PSO	100
Iterative periods of main PSO	100
Inertia coefficient w	0.4 to 0.9
Acceleration coefficients	C_1=C_2=2.05

4.1 The general transfer function

$$H(z_2, z_2) = h_0 \frac{1 + a_{01}z_2 + a_{02}z_2^2 + a_{10}z_1 + a_{20}z_1^2 + a_{11}z_1z_2 + a_{12}z_1Z_2^2 + a_{21}z_1^2Z_2 + a_{22}z_1^2z_2^2}{(1 + b_1z_1 + c_1z_2 + d_1z_1z_2)(1 + b_2z_1 + c_2z_2 + d_2z_1z_2)} \quad (24)$$

The optimized vector x:

$x(\ a_{01}\ a_{02}\ a_{10}\ a_{11}\ a_{12}\ a_{20}\ \ a_{21}\ a_{22}\ b_1\ c_1\ d_1\ b_2\ c_2\ d_2\ h_0)$
x(0.4083-0.3521 0.2272 0.3112-0.7478-1.114 0.7196
1.7258 -0.9279-0.9274 0.8765-0.4518-0.4249-0.0175 0.001)

Minimum error obtained after optimization is:
Error1 = 1.2216

4.2 Transfer function with a mirror image polynomial numerator

$$H(z_1, z_2) = H_0\left(\frac{z_1^{-1}z_2^{-1}\left(b_{11} + b_{01}(z_1 + z_2^{-1}) + b_{10}(z_2 + z_2^{-1})\right)}{D_1 \cdot D_2} + \frac{z_1^{-1}z_2^{-1}\left(b_{02}(z_1z_2^{-1} + z_2z_1^{-1}) + b_{00}(z_1z_2 + z_1^{-1}z_2^{-1})\right)}{D_1 \cdot D_2}\right) \quad (25)$$

where,

$$D_1 = 1 + b_1z_1^{-1} + C_1z_2^{-1} + d_1z_1^{-1}z_2^{-1}$$

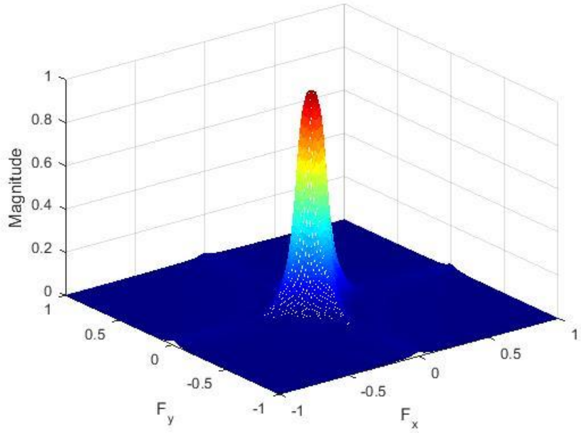
$$D_2 = 1 + b_2 z_1^{-1} + C_2 z_2^{-1} + d_2 z_1^{-1} z_2^{-1}$$

The amplitude is:

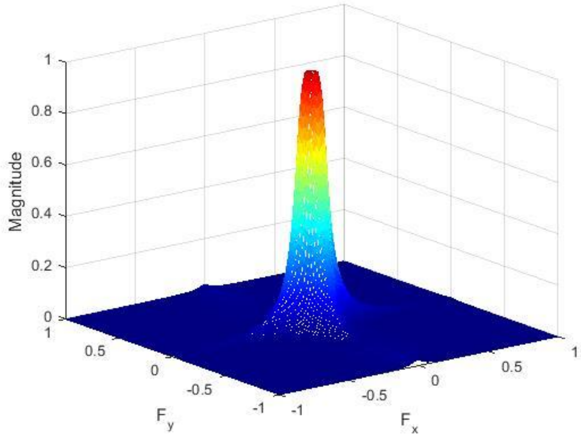
$$M(\omega_1, \omega_2) = H_0 \left| \frac{A_r}{(B1_r + jB1_i)(B2_r + jB2_i)} \right| \qquad (26)$$

Minimum error obtained after optimization:
Error2 = 1.1428
the vector:

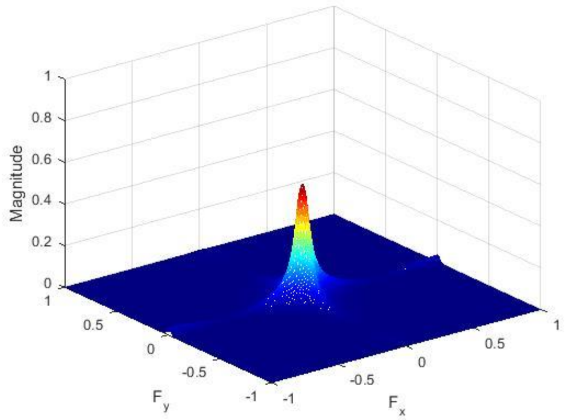
$x(b_{00}\ b_{01}\ b_{10}\ b_{11}\ b_1\ c_1\ d_1\ b_2\ c_2\ d_2\ H_0)$
x(0.2444 -0.9507 -0.3804 -1.4545 -0.1707 -0.0008 -0.1742 -0.6908 -0.3747 0.1565 -0.5511 0.0065)



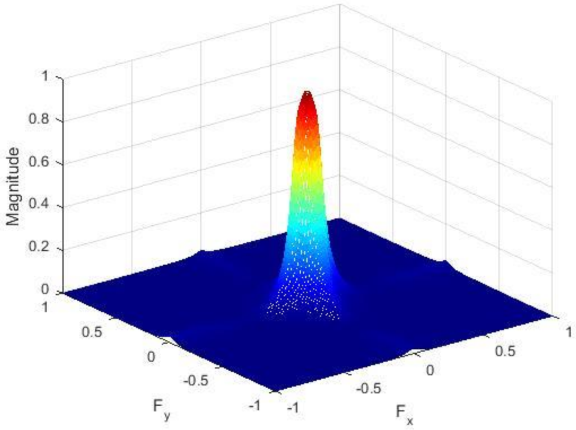
(a) HMAPSO



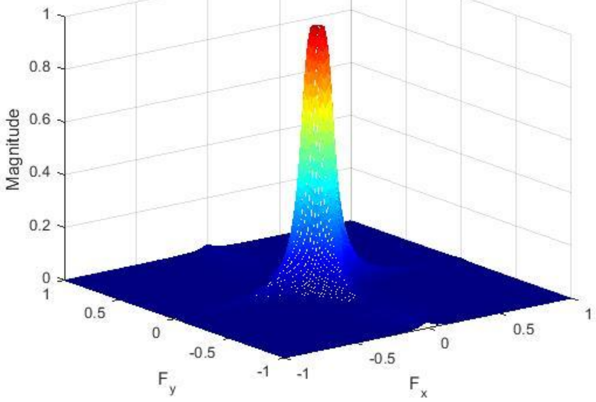
(b) Proposed algorithm (equation N° 24)



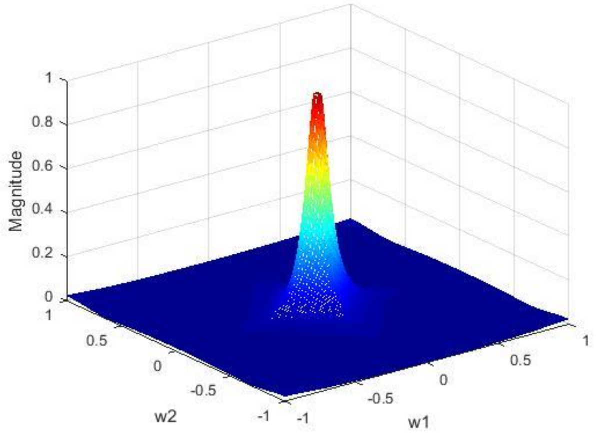
(c) MEPSO



(d) SA-PSO



(e) Proposed algorithm (equation N° 25)



(f) BBO–PSO

Figure 4. Comparison of amplitude responses M(w1, w2) by using different algorithm (MEPSO, HMAPSO, SA-PSO, BBO–PSO and proposed algorithm)
Note: in Figure 4 $F_x = \omega_1$ and $F_y = \omega_2$

4.3 Discussion

In order to prove the efficiency of our proposed algorithm, it is compared to some important works: MEPSO [4], HMAPSO [11], BBO–PSO [22] and SA-PSO [10] dealing with the PSO algorithm and proving their good performance compared to other algorithms.

Table 2. Final results (filter coefficients and minimum error) with different algorithms

Coefficients	**BBO–PSO**	**SA-PSO**	**MEPSO**	**HMAPSO**	**Proposed Algorithm**
a_{01}	0.3840	0.3069	0.3061	*0.5815*	0.4083
a_{02}	− 0.8816	-0.9806	0.9949	*0.2207*	-0.3521
a_{10}	− 0.4478	0.1681	0.3935	0.4387	0.2272
a_{11}	− 0.7609	-0.0431	-0.0338	0.4045	0.3112
a_{12}	1.1371	-0.1820	0.6481	-1.4084	-0.7478
a_{20}	− 0.1803	-0.7270	1.2345	-0.5720	-1.1144
a_{21}	0.2938	-0.3249	0.5030	*-0.8418*	0.7196
a_{22}	0.2473	1.6358	0.4481	2.277	1.7258
b$_1$	-0.1249	-1.4201	-1.0239	-0.9078	-0.9279
b$_2$	-1.0553	-0.9178	0.0342	-0.9058	-0.9274
c$_1$	-0.1890	-0.6530	-0.9605	0.8373	0.8765
c$_2$	-1.0594	-0.9127	-0.0371	-0.9075	-0.4518
d$_1$	-0.8986	1.0081	0.9523	-0.9101	-0.4249
d$_2$	1.1366	0.8545	-0.9056	0.8406	-0.0175
h$_0$	0.0013	0.0022	0.00034	0.00024	0.001
Minimum Error	2.8518	2.629	9.0005	1.58657	1,221

- Checking the Stability condition [4, 10, 11, 22]:

$$|b_k + c_k| - 1 < d_k < 1 - |b_k - c_k|$$

In order to verified stability of the filters resulted by MEPSO, HMAPSO, BBO–PSO and SA-PSO, we use the values of b$_k$, c$_k$, and d$_k$ (k=1, 2) in Table 2

Table 3. Stability checking table

ALGORITHM	**filter resulted**
MEPSO	unstable
SA-PSO	unstable
BBO–PSO	stable
HMAPSO	stable
Proposed Algorithm (Multi-PSO)	stable

From Table 2 it can be easily verified that our proposed algorithm (Multi-PSO) is classified in better in terms of good optimization quantified by the minimum error value (error =1.2216), in second rank, comes the HMAPSO with an error value (error =1.586576) and the lastly BBO–PSO with MEPSO.

We can point out that for the SA-PSO and MEPSO algorithms, the instability problem is solved by good optimization that guarantees the stability, whereas their resulted filters are unstable when checking the stability condition (Table 3**)**. On other hand our proposed algorithm always ensures filter stability.

According to the obtaining results it is clear that the frequency Magnitude responses in the two methods (equation (24) and (25)) of our proposed algorithm (see Figure 4) are very close to the model filter (see Figure 1).

In addition, the Minimum error obtained by our model after optimization (error1 and error2) are very small compared to the results given by the HMAPSO, SA-PSO, BBO–PSO and MEPSO algorithms. More than that, our model is always stable.

5. CONCLUSION

In this study, a new Multi-PSO algorithm for the two-dimensional recursive digital IIR filter design is investigated. It is characterized by the diversity of the particles and the use of two ideas. The first consists of using two types of swarms, a principal and auxiliaries. The second idea is the partition of the search space into several zones. Thus, leading to a better exploration of the problem. The results obtained with Multi-PSO are compared with other algorithms previously reported. They indicate that Multi-PSO based methods exhibit better performance in all experiments and provide best optimum solution during search mechanism. In conclusion, these new ideas presented above opened the gate to new methods that could be applied to other optimization algorithms, such as evolutionary algorithms. but the proposed Multi-PSO takes a long time in simulation which is disadvantage.

REFERENCES

[1] Mladenov, V.M., Mastorakis, N.E. (2001). Design of two-dimensional recursive filters by using neural networks. IEEE Transactions on Neural Networks, 12(3): 585-590. https://doi.org/10.1109/72.925560

[2] Lu, W.S., Antoniou, A. (1992). Two-Dimensional Digital Filters. Marcel Dekker. INC, CRC Press.

[3] Lim, J.S. (1990). Two-Dimensional Signal and Image Processing. PH, 195-196.

[4] Das, S., Konar, A. (2007). A swarm intelligence approach to the synthesis of two-dimensional IIR filters. Engineering Applications of Artificial Intelligence, 20(8): 1086-1096. https://doi.org/10.1016/j.engappai.2007.02.004

[5] Ramamoorthy, P.A., Bruton, L.T. (1979). Design of stable two-dimensional analogue and digital filters with applications in image processing. International Journal of Circuit Theory and Applications, 7(2). https://doi.org/10.1002/cta.4490070210

[6] Aly, S., Fahmy, M. (1978). Design of two-dimensional recursive digital filters with specified magnitude and group delay characteristics. IEEE Transactions on Circuits and Systems, 25(11): 908-915. https://doi.org/10.1109/TCS.1978.1084407

[7] O'Connor, B.T., Huang, T.S. (1978). Stability of general two-dimensional recursive digital filters. IEEE Transactions on Acoustics, Speech, and Signal Processing, 26(6): 550-560. https://doi.org/10.1109/TASSP.1978.1163162

[8] Xie, L.J. (2011). A criterion for Hurwitz polynomials and its applications. International Journal of Modern Education and Computer Science (IJMECS), 125(4): 38-

40. https://doi.org/10.5815/ijmecs.2011.01.06

[9] Mastorakis, N.E., Gonos, I.F., Swamy, M.N.S. (2003). Design of two-dimensional recursive filters using genetic algorithms. IEEE Transactions on Circuits and Systems, 50(5): 634-639. https://doi.org/10.1109/TCSI.2003.811019

[10] Dhabal, S., Venkateswaran, P. (2014). Two-dimensional IIR filter design using simulated annealing based particle swarm optimization. Journal of Optimization, 2014: 1-10. https://doi.org/10.1155/2014/239721

[11] Kumar, R., Kumar, A. (2010). Design of two-dimensional infinite impulse response recursive filters using hybrid multiagent particle swarm optimization. Applied Artificial Intelligence, 24(4): 295-312. https://doi.org/10.1080/08839511003715204

[12] Kennedy, J., Eberhart, R.C. (1995). Particle Swarm Optimization. IEEE International Conference on Neural Networks, Perth, WA, Australia, Australia, pp. 1942-1948. https://doi.org/10.1109/ICNN.1995.488968

[13] Shi, Y., Eberhart, R.C. (1999). Empirical study of particle swarm optimization, in: Proceedings of Congress on Evolutionary Computation. Washington, DC, USA, pp. 1945-1949. https://doi.org/10.1109/CEC.1999.785511

[14] Shi, Y., Eberhart, R.C. (1998). A modified particle swarm optimizer. IEEE International Conference on Evolutionary Computation, Anchorage, AK, USA, pp. 69-73. https://doi.org/10.1109/ICEC.1998.699146

[15] Angeline, P.J. (1998). Evolutionary optimization versus particle swarm optimization and philosophy and performance difference. 7th Annual Conference on Evolutionary Programming, San Diego, USA, pp. 601-610. https://doi.org/10.1007/BFb0040811

[16] Shenoi, B.A. (1999). Magnitude and Delay Approximation of 1-D and 2-D Digital Filters. Springer-Verlag Berlin Heidelberg, 190-195. 1st edition.

[17] Eberhart, R., Kennedy, J. (1995). A new optimizer using particle swarm theory. IEEE the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 39-43. https://doi.org/10.1109/MHS.1995.494215

[18] El Dor, A., Clerc, M., Siarry, P. (2012). A multi-swarm PSO using charged particles in a partitioned search space for continuous optimization. Computational Optimization and Applications, 53: 271-295. https://doi.org/10.1007/s10589-011-9449-4

[19] Kennedy, J. (1998). The behavior of particles. LNCS, Springer, In Proceedings of the 7th Conference on Evolutionary Computation, pp. 581-589. https://doi.org/10.1007/BFb0040809

[20] Ozcan, E., Mohan, C. (1999). Particle Swarm Optimization: surfing the waves. IEEE Congress on Evolutionary Computation, pp. 1939-1944. https://doi.org/10.1109/CEC.1999.785510

[21] Clerc, M., Kennedy, J. (2002). The particle swarm: explosion, stability, and convergence in multi-dimensional complex space. IEEE Transactions on Evolutionary Computation, 20(1): 1671-1676. https://doi.org/10.1109/4235.985692

[22] Lv, C., Yan, S., Cheng, G., Xu, L., Tian, X.Y. (2016). Design of two-dimensional IIR digital filters by using a novel hybrid optimization algorithm. Multidim. Syst. Sign. Process, 28: 1267-1281. https://doi.org/10.1007/s11045-016-0397-0

NOMENCLATURE

2-D	two-dimensional
ETs	evolutionary techniques
FIR	finite impulse response
GA	genetic algorithm
HMAPSO	Hybrid Multiagent Particle Swarm Optimization
IIR	infinite impulse response
MEPSO	modified particle swarm optimizer
Min. Err	Minimum error
PSO	particle swarm optimization