

# Design of Multiplier-less Reconfigurable Non-uniform Channel Filters using Meta-heuristic Algorithms

Bindiya T.S.

Department of Electronics and Communication Engineering  
National Institute of Technology Calicut  
Kerala 673601, India

Elizabeth Elias

Department of Electronics and Communication Engineering  
National Institute of Technology Calicut  
Kerala 673601, India

## ABSTRACT

Low complexity and reconfigurability are reported to be the key features in a software defined radio (SDR). To obtain these features, a reconfigurable architecture based on frequency response masking (FRM) technique can be used for the implementation of the channel filters in the SDR. The frequency response masking approach is proved to be a good candidate for the realization of a sharp digital finite impulse response (FIR) filter with low complexity. To reduce the complexity and power consumption for hardware realization, a design method which makes the channel filters totally multiplier-less is proposed in this paper. Continuous filter coefficients are first converted to finite precision coefficients using signed power of two (SPT) space to obtain a multiplier-less filter. The representation of the FRM filter coefficients in the SPT space can degrade the filter performance. This calls for the use of a suitable optimization technique. The classical gradient based optimization techniques cannot be deployed here, because the search space consists of integers. In this context, meta-heuristic algorithm is a good choice as it can be tailor made to suit the problem under consideration. They are especially useful in finding near optimal solutions in multimodal, multidimensional space. Several meta-heuristic algorithms are modified in this paper to be used for the discrete optimization.

## General Terms:

Frequency Response Masking, Optimization

## Keywords:

Frequency Response Masking, Canonic Signed Digit, Meta-heuristic Optimization Algorithms, Genetic Algorithm, Differential Evolution, Artificial Bee Colony, Harmony Search Algorithm, Gravitational Search Algorithm

## 1. INTRODUCTION

For applications like software defined radio (SDR) [1], a technology which provides multi-band, multi-standard and multi-service is required. Low complexity and reconfigurability are essential features of the channel filters in SDR. The channelizer in the SDR receiver extracts non-uniform sub-bands from the input signal [2]. Since the channelizer operates at the highest sampling rate, channel filters need to be implemented with less complexity and should consume less power. Frequency Response Masking (FRM) is a method to implement filters with less complexity and sharp transition width [3]. The complexity reduction of the filters implemented using FRM method is due to the large number of zero valued multiplier coefficients. A reconfigurable architecture based on FRM is reported in [4]-[7] in which non uniform channels are extracted from the same prototype filter using different interpolation factors and masking filters for each channel. A modification of this approach using multiple stage FRM for the prototype filter and multiple level of masking filters when

their orders are high is reported [8], which reduces the complexity of the channel filters further. The sub-filters of the FRM filter are designed using Remez Exchange algorithm [9].

In this paper, a further reduction in the complexity is proposed, whereby the filter representation becomes totally multiplier-less. This is achieved by converting the infinite precision filter coefficients into finite precision coefficients using signed power-of-two (SPT) space. The SPT system allows the multiplications to be replaced by shift and add operations [10]. Since the multipliers are the major power and area consuming components of a filter, this results in low power hardware realizable digital filters. Canonic Signed Digit (CSD) representation is a special case of the SPT space, which uses both additions and subtractions [11]. Because of this, only a minimum number of non-zero SPT terms is needed to represent a decimal number.

When the continuous filter coefficients are rounded using finite number of bits in the CSD space, the performance of the filter degrades. So, efficient optimization techniques in the discrete space are required to improve the performance of the CSD represented filters. Since, here, the search space contains integers, meta-heuristic algorithms have to be used instead of classical gradient based optimization techniques. If the parameters are properly selected, these algorithms will result in a global solution.

A ternary coded genetic algorithm based on the look up table for the optimization of FRM channel filters is presented in [8]. When ternary coded filter coefficients are used, the various operations of the optimization algorithm may result in non canonical bit strings. So, suitable restoration algorithms are needed to convert them into canonical bit strings [12]. Integer coded GA, in which the integer indices of the look up table entries are used to get the solution, is proposed in [13]. Integer coded Artificial Bee Colony (ABC) algorithm [14] is used for the optimization of multiplier-less transmultiplexer by Manoj and Elizabeth [15]. A modified integer coded ABC and integer coded Differential Evolution (DE) algorithm for the design of CSD encoded FRM filter are presented by Manju and Elizabeth [16]. In this paper, multiplier-less reconfigurable channel filters are implemented using various meta-heuristic algorithms like, Differential Evolution (DE), Harmony Search Algorithm (HSA) [17], Gravitational Search Algorithm (GSA) [18] and Artificial Bee Colony Algorithm (ABC). Such an approach is not reported in the literature so far. All the optimization algorithms in this work use integer coded filter coefficients. A comparative study of the performance of the multiplier-less channel filters in terms of the pass-band ripple and stop-band attenuation for various optimization techniques, is presented in this paper.

The paper is organized as follows. Section 2 gives an overview of the FRM approach. Section 3 discusses Reconfigurable Filters Based on FRM. Section 4 provides a brief description of the CSD representation. Section 5 gives a brief overview of vari-

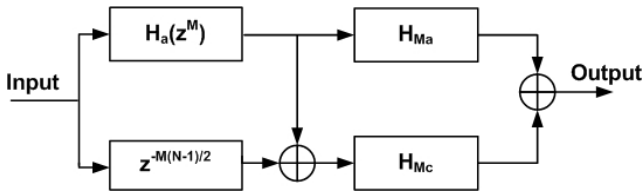


Fig. 1. Basic FRM Filter Architecture

ous meta-heuristic optimization techniques ABC, DE, HSA and GSA. The problem statement is discussed in Section 6. Section 7 gives the design of the continuous coefficient and CSD represented channel filters. In Section 8, the design of multiplier-less channel filters using various meta-heuristic algorithms is illustrated. The results and discussions are presented in Section 9. Section 10 concludes the paper.

## 2. REVIEW OF FRM APPROACH

Let  $H(z)$  be the transfer function of the desired FIR low pass filter with pass-band and stop-band edge frequencies  $f_p$  and  $f_s$  respectively. FRM FIR filter is composed of a band edge shaping filter  $H_a(z)$ , masking filter  $H_{Ma}(z)$  and complementary masking filter  $H_{Mc}(z)$ . The complementary filter  $H_c(z)$  of  $H_a(z)$  can be expressed as given below

$$H_c(z) = z^{-\frac{(N-1)}{2}} - H_a(z) \quad (1)$$

$H_a(z)$  and  $H_c(z)$  are interpolated with a factor  $M$  and are cascaded with the masking filters  $H_{Ma}(z)$  and  $H_{Mc}(z)$  respectively. Thus, the transfer function of the overall FIR FRM filter  $H(z)$  [3] is given by

$$H(z) = H_a(z^M)H_{Ma}(z) + H_c(z^M)H_{Mc}(z) \quad (2)$$

The structure of the FRM FIR filter is given in Fig.1 [3]. The design steps for the sub-filters are given below [3]:

$$m = \lfloor f_p * M \rfloor \quad f_{ap} = f_p M - m \quad f_{as} = f_s M - m \quad (3)$$

$$\begin{aligned} f_{map} &= f_p & f_{mas} &= \frac{m+1-f_{as}}{M} \\ f_{mcp} &= \frac{m-f_{ap}}{M} & f_{mcs} &= f_s \end{aligned} \quad (4)$$

where  $\lfloor x \rfloor$  denotes the largest integer less than  $x$ ,  $M$  is the interpolating factor,  $f_p$  and  $f_s$  respectively are the pass-band and stop-band frequencies of the final filter  $H(z)$ .  $f_{ap}$  and  $f_{as}$  are the pass-band and stop-band frequencies respectively of the prototype filter  $H_a(z)$ .  $f_{map}$  and  $f_{mcp}$  are the pass-band frequencies and  $f_{mas}$  and  $f_{mcs}$  are the stop-band frequencies of the masking filters  $H_{Ma}(z)$  and  $H_{Mc}(z)$  respectively. The transition width of the overall filter  $H(z)$  is  $\frac{1}{M}$  times the transition width of  $H_a(z)$  i.e.  $\frac{(f_{as}-f_{ap})}{M}$ . The frequency responses of each filter are given in Fig.2 [3].

## 3. RECONFIGURABLE FILTERS BASED ON FRM

The reconfigurable filter structure given in [4]-[7] is based on the FRM technique, where, different channel filters with non-uniform bandwidths are derived from the same prototype filter. Different interpolation factors and masking filters are used to derive the different channel filters from the same prototype filter. For example, two channel filters with pass-band frequencies  $f_{p1}$  and  $f_{p2}$  and stop-band frequencies  $f_{s1}$  and  $f_{s2}$  respectively need to be designed. Let the pass-band and stop-band frequencies of the prototype filter be  $f_{ap}$  and  $f_{as}$  respectively. For this, two

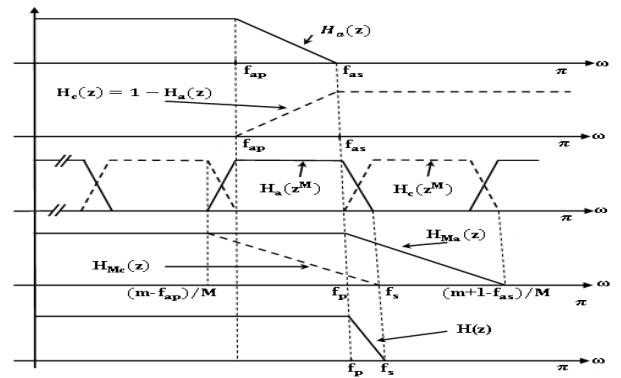


Fig. 2. Frequency response illustration of FRM approach

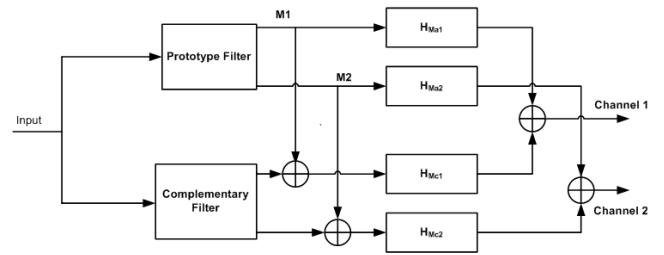


Fig. 3. The structure of the two channel reconfigurable filters

interpolation factors  $M_1$  and  $M_2$  are to be found using the following equations [4]-[7]:

$$f_{ap} = f_{p1} M_1 - \lfloor f_{p1} M_1 \rfloor = f_{p2} M_2 - \lfloor f_{p2} M_2 \rfloor \quad (5)$$

$$f_{as} = f_{s1} M_1 - \lfloor f_{s1} M_1 \rfloor = f_{s2} M_2 - \lfloor f_{s2} M_2 \rfloor \quad (6)$$

The masking filters for each channel can be designed using Eq. 4. The structure of the two channel reconfigurable filters is given in Fig.3 [4].  $H_{Ma1}$  and  $H_{Mc1}$  are the masking filters for channel 1 and  $H_{Ma2}$  and  $H_{Mc2}$  are the masking filters for channel 2.  $M_1$  and  $M_2$  are the interpolation factors for channel 1 and channel 2 respectively.

## 4. CANONIC SIGNED DIGIT REPRESENTATION (CSD)

Any FIR filter can be represented as

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (7)$$

where  $N$  is the length of the FIR filter,  $h(k)$  are the filter coefficients and  $x(n)$  is the input signal. The implementation of the continuous coefficient FIR filter consists of multipliers, which are the main power consuming components. If the filter coefficients are represented in the SPT space, multipliers can be replaced by shifters and adders. The number of non-zero bits in the filter coefficient representation decides the number of partial product additions and hence the adders. So the number of adders can be reduced by reducing the number of non-zero bits in the filter coefficients. CSD representation is a unique representation of the filter coefficients with minimum number of non-zero bits [10]-[11]. A fractional number  $q$  is represented in CSD format as [11].

$$q = \sum_{i=1}^W c_i 2^{R-i} \quad (8)$$

where  $c_i = \{-1, 1, 0\}$  and  $W$  is the word length of the CSD number. No adjacent bits in the CSD representation can be non-zero

i.e.  $c_i * c_{i-1} = 0$ , where  $c_i$  is the  $i^{th}$  bit in the CSD representation.

## 5. OVERVIEW OF OPTIMIZATION ALGORITHMS

Since, in this work, the performance of the CSD represented channel filters is optimized, the search space consists of integers. So classical gradient based optimization algorithms cannot be used. The meta-heuristic algorithms are found to be suitable for optimization problems in which the search space contains integers. This section gives an overview about meta-heuristic algorithms like DE, ABC, GSA and HSA algorithms.

### 5.1 Differential Evolution (DE)

The DE algorithm is a simple evolutionary algorithm and is introduced by R. Storn and K. V. Price [19]. Here, a new offspring is generated by adjusting the existing vectors using a scaled differential operator. DE is a simple and fast converging algorithm and has only few control parameters. The main steps of the DE algorithm include Mutation, Recombination and Selection and are explained in Section 8.1.

### 5.2 Harmony Search Algorithm (HSA)

Inspired by the music improvisation scheme, Z. Geem introduced the Harmony Search Algorithm [17], [20] for the optimization of mathematical problems. In the music improvisation scheme, each musician plays a note within the possible range which together makes one harmony vector. If all the pitches make a good harmony decided by an aesthetic standard, that experience is stored in the musician's memory. The possibility to make a good harmony is increased next time. When a musician improvises one pitch, he follows any one of three rules [17]: (1) playing any one pitch from his memory, (2) playing an adjacent pitch of the pitch stored in his memory, and (3) playing a totally random pitch from the possible range. Similarly, in the HSA, when each decision variable chooses a value, it follows any one of the three rules [17]: (1) Memory considerations in which the value is selected from the harmony memory, (2) pitch adjustments in which a value is chosen which is adjacent to the value from the harmony memory, and (3) randomization in which a random value from the possible range is selected. These three rules in HS algorithm are based on two parameters, i.e., harmony memory considering rate (HMCR) and pitch adjusting rate (PAR). The various steps of HSA algorithm are discussed in Section 8.2.

### 5.3 Gravitational Search Algorithm(GSA)

GSA [18] is a population based algorithm based on the law of gravity and mass interactions. GSA can be considered as an artificial world of masses. In GSA, each mass (agent) has four specifications: position, inertial mass, active gravitational mass, and passive gravitational mass. The position of the masses constitutes the solution space and the performance of each solution is measured in terms of their masses using fitness function. The interactions of the masses are based on the Newtonian laws of gravity and motion. Masses attract each other by the force of gravity and objects are moved towards the object with heavier mass, which represents the optimum solution. The heavy masses move more slowly compared to lighter ones. Masses obey the following laws [18]:

**Law of gravity:** Each particle attracts every other particle and the gravitational force between two particles is directly proportional to the product of their masses and inversely proportional to the distance,  $R$ , between them.

**Law of motion:** the current velocity of any mass is equal to the sum of the fraction of its previous velocity and the acceleration.

Acceleration of any mass is equal to the force acted on the system divided by the mass of inertia.

The inertial mass of an agent represents its resistance to make its movement slow. The velocity of an agent is controlled by the gravitational mass and the inertial mass, which are computed by the fitness function. The positions of the agents are updated with every iteration. The various steps of GSA algorithm are described in Section 8.3 in detail.

### 5.4 Artificial Bee Colony (ABC) Algorithm

Based on the foraging behaviour of honey bee swarm, Karaboga proposed ABC algorithm in 2005 for solving multi-dimensional and multi-modal optimization problems [14]. The colony of bees consists of three types of bees: employed bees, onlooker bees and scout bees.

Initially, each employed bee selects a food source randomly and its nectar quality is determined. It searches the neighbouring area for better food source. If a food source with higher nectar amount is found, the employed bee memorizes the position of the new food source and forgets the position of the old food source. Otherwise, the position of old food source will be retained. Once the search by all the employed bees are over, they pass the information of their food source to the onlooker bees. Now, the onlooker bees determine the nectar information taken from all the employed bees and a food source is selected with a probability depending on the nectar amount. If the nectar amount is more, the corresponding food source will have higher probability to get more number of onlooker bees. As in the case of employed bees, onlookers also search for neighbouring area for better food source. If a food source with higher nectar amount is found, onlooker bee memorizes the position of the new food source and forgets the position of the old food source. The employed bee whose food source is abandoned due to poor nectar quality becomes a scout bee. Scout bees search for new food sources randomly. Occasionally, the scouts can find food source with good nectar quality. In ABC algorithm, the position of food sources represent the possible solutions to the problem and the nectar quality of a food source represents the fitness value of the corresponding solution. The detailed steps are explained in Section 8.4.

## 6. STATEMENT OF THE PROBLEM

In this work, low complexity non-uniform channel filters are designed, which are suitable for hardware implementation. When the continuous coefficient of the channel filters are rounded with number of SPT terms, the performance of the channel filters may degrade. So by selecting suitable optimization technique and objective function, we can improve the performance of the channel filters, which are totally multiplier-less. Since the different channel filters are derived from the same prototype filter, the optimization of the prototype filter is done first and then that of the masking filters of each channel as presented in [8]. The optimized prototype filter and optimized masking filters of each channel are used to design the optimized channel filters. The formulation of the objective function is discussed below.

### 6.1 Encoding of CSD filter coefficients

Once the continuous coefficients of the sub-filters of the channel filters are obtained, they are converted into the CSD space. Converting the filter coefficients into the CSD space is done using a look up table approach [21]. In this work, a 12 bit CSD representation is used. A CSD look up table consisting of four fields are created, namely, index, CSD representation, decimal equivalent and number of SPT terms as shown in Table 1.

**Table 1. CSD look up table entries**

Index	CSD Representation	Decimal Equivalent	No: of SPT Terms
1183	0100101000-10	0.5771	4

The optimization of the prototype filter will be done first. Since the prototype filter is designed to have linear phase, only half the number of coefficients of the filter need to be taken to form the initial seed. If the prototype filter is made up of sub-filters using the FRM method as discussed in Section 2, joint optimization of these sub-filters has to be done i.e., the coefficients of the sub-filters are concatenated together to form the initial seed of the optimization problem. Since all the sub-filters are designed to have linear phase, only half the number of coefficients of each filter need to be taken and concatenated to form the initial seed. This reduces the dimension of the optimization problem. Also, since the joint optimization of the sub-filters is used, all of them will be simultaneously adjusted to get better FRM filter response.

Now, the optimization of the masking filters need to be done. The masking filters are designed to have linear phase. So, only half the number of coefficients of each filter need to be concatenated to form the initial seed in the optimization of masking filters.

Ternary encoding based optimization is discussed in [8]. When the ternary encoding is used, dimension of the optimization problem is large. Also, restoration algorithms [12] are needed to bring the non-canonical bit strings resulting from various operations into the CSD format. So, in this work, they are encoded as the signed indices of the look up table entries of the nearest CSD equivalent [13], [15], [16]. Thus the initial solution is obtained by concatenating these signed indices.

When the fitness is evaluated in each iteration, each index need to be decoded to get the CSD equivalent. For this, the magnitudes of indices are used to address the look-up table, to access the magnitudes of CSD filter coefficients and the number of SPT terms in them. The signs of the indices are attached to the magnitudes of the CSD filter coefficients, to get the actual values.

## 6.2 Formulation of the objective function

When the filter coefficients are represented in the CSD format, all the multipliers are replaced with shifters and adders. The number of adders required to implement the filter is proportional to the number of SPT terms in the CSD representation. When the filter coefficients are rounded to the nearest CSD representation with restricted number of non-zero SPT terms, the frequency response of the filter degrades. Suitable objective function need to be used to optimize the performance of the multiplier-less filter with minimum number of non-zero SPT terms.

In this paper, the pass-band ripple and stop-band attenuation of the FRM filter is to be optimized with restricted average number of non-zero bits. So the objective function should include these parameters. The pass-band and stop-band error in the frequency domain is defined respectively as:

$$F_1(x) = (|H(\omega, x)| - 1), \quad 0 < \omega < \omega_p \quad (9)$$

$$F_2(x) = |H(\omega, x)|, \quad \omega_s < \omega < \pi \quad (10)$$

where,  $F_1(x)$  is the pass-band error and  $F_2(x)$  is the stop-band error. The pass-band error and the stop-band error are the difference between the zero phase frequency response of the optimized filter,  $H(\omega, x)$  and the response of the ideal filter which is 1 in the pass-band and 0 in the stop-band.

In this work, instead of representing each filter coefficient with restricted number of non-zero SPT terms, the average number of non-zero SPT terms in a filter is limited to a maximum value. This allocation has more flexibility, since, here, the CSD filter

coefficients can contain different number of non-zero SPT terms. To include the constraint for limiting the average number of non-zero SPT terms in the CSD represented coefficients, a function is added to the objective function using the penalty method [22]. If  $n(x)$  denotes the average number of non-zero SPT terms in the optimized coefficients and  $n_b$  is the required upper bound of  $n(x)$ , the penalty function added to the optimization problem is given as

$$g(x) = \max(0, n(x) - n_b) \quad (11)$$

where  $x$  is the vector constituted by the design variables.

In this paper, the average number of non-zero SPT terms is fixed as 2. Thus after the optimization, we get channel filters with better performance and reduced number of non-zero SPT terms. Here, the objective function is intended to reduce the pass-band and stop-band errors of the channel filters and the average number of non-zero SPT terms and is given below:

$$\text{Minimize } F(x) = \alpha_1 F_1(x) + \alpha_2 F_2(x) + \alpha_3 g(x) \quad (12)$$

where,  $\alpha_1, \alpha_2$  and  $\alpha_3$  are positive weighting coefficients and can be taken considering the relative importance of each term in the optimization problem. In this work, for the optimization of the prototype filter,  $\alpha_1$  and  $\alpha_2$  are taken as 1 and  $\alpha_3$  is taken as 0.1. For the optimization of the masking filters of channel 1,  $\alpha_1$  is taken as 0.5,  $\alpha_2$  is taken as 1 and  $\alpha_3$  is taken as 0.25 and for the optimization of the masking filters of the channel 2,  $\alpha_1$  is taken as 0.5 and  $\alpha_2$  and  $\alpha_3$  are taken as 1.

## 7. DESIGN EXAMPLE

In this work, reconfigurable two channel filters using FRM method is designed as shown in Fig.4. They are made totally multiplier-less using various meta-heuristic algorithms and the performances are analysed. All the simulations are done using MATLAB 7.10.0.499 on an Intel(R) Core(TM) i5-2400 processor operating at 3.10 GHz.

### 7.1 Design of Continuous Coefficient Channel Filters

The design of a two channel software radio channelizer which supports Code Division Multiple Access (CDMA) and Wideband-CDMA (WCDMA) is illustrated. The specifications for each channel are given below. The first step is to design the continuous coefficient channel filters.

#### First Channel (WCDMA)

Pass-band (PB) frequency: 5000 kHz  
Maximum pass-band ripple: 0.01 dB  
Stop-band (SB) frequency: 5250 kHz  
Minimum stop-band attenuation: 60 dB  
Sampling Rate: 21 MHz

#### Second Channel (CDMA)

Pass-band frequency: 1250 kHz  
Maximum pass-band ripple: 0.01 dB  
Stop-band frequency: 1300 kHz  
Minimum stop-band attenuation: 60 dB  
Sampling Rate: 21 MHz

When these values are substituted in Eq. 5 and 6, the interpolation factors for channel 1 and channel 2 are obtained as  $M_1 = 5$  and  $M_2 = 20$  respectively. Also the prototype filter pass-band and stop-band cut off frequencies are found to be  $f_{ap} = 0.2$  and  $f_{as} = 0.25$ . The length of a filter can be estimated using Bellanger's equation given below [23]:

$$N = \frac{-2 \log(10\delta_1\delta_2)}{3\Delta f} - 1 \quad (13)$$

where  $\delta_1$  and  $\delta_2$  are the peak pass-band and stop-band ripple magnitudes respectively, and  $\Delta f$  is the normalized transition-bandwidth. When these values are substituted in the above equa-

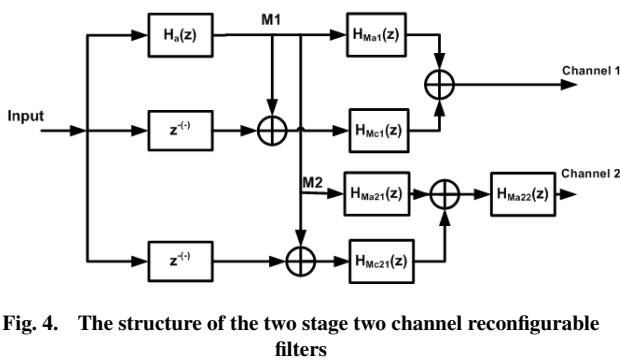


Fig. 4. The structure of the two stage two channel reconfigurable filters

tion, the length of the prototype filter is obtained as 65. The order of the prototype filter implemented using Parks McClellan method is also found to be 65. The number of multipliers needed to implement the prototype filter is 33.

The WCDMA channel filter with small interpolation factor of  $M_1 = 5$ , is implemented using single stage masking. The lengths of the masking and masking complementary filters  $H_{Ma1}$  and  $H_{Mc1}$  are obtained as 30 and 36 respectively. The interpolation factor of CDMA channel filter is found to be,  $M_2 = 20$ . When the interpolation factor is high, the order of the masking filters becomes high. In order to reduce the order of masking filters, multi-stage masking can be used, in which the interpolation factor is factorized and masking filters are designed for each of the factors as discussed in [5]-[7]. Here, the interpolation factor of the second channel,  $M_2$ , is factorized into two,  $M_{21} = 5$  and  $M_{22} = 4$ .  $H_{Ma21}$  and  $H_{Mc21}$  are the first stage masking filters corresponding to  $M_{21}$  designed using Eq. 4 and they are interpolated with value  $M_{22}$ .  $H_{Ma22}$  is the second stage masking filters designed for interpolation factor  $M_{22}$ . So, for the second channel, the length of the first stage masking and masking complementary filters  $H_{Ma21}$  and  $H_{Mc21}$  are obtained as 29 and 37 respectively. The length of the second stage masking filter  $H_{Ma22}$  of channel 2 is 23. The filters are implemented using Remez Exchange algorithm. The structure of the reconfigurable filters with two-stage masking is given in Fig.4 [7]. The pass-band ripple and stop-band attenuation of the continuous coefficient channel filters are listed in Table 8 and Table 9 respectively. The magnitude response of the continuous coefficient channel 1 and channel 2 filters are shown in Fig.5 and Fig.6 respectively.

## 7.2 Design of CSD rounded channel filters

Now, to replace the multipliers in the filter implementation with shifters and adders, the coefficients of all the sub-filters are directly rounded to the nearest CSD representation with restricted number of non-zero bits. The performances of channel filters degrade. The magnitude response of the CSD rounded channel filters, channel 1 and channel 2 respectively are shown in Fig.5 and Fig.6. Hence, suitable optimization algorithms in the CSD space have to be used to improve the performance of the channel filters.

## 8. DESIGN OF MULTIPLIER-LESS CHANNEL FILTERS USING META-HEURISTIC ALGORITHMS

Since, both the channels are derived from the same prototype filter, the CSD represented prototype filter is optimized first. Then the CSD represented masking filters of each channel are optimized separately. All the sub-filters of both channels are designed to have linear phase. Hence only half of the filter coefficients need to be taken for the optimization.

The optimization of the prototype filter is done first. Half the coefficients of the prototype filter are taken to form the initial

solution vector for the prototype filter optimization. The average number of SPT terms in a filter coefficient are fixed to be 2.

Now, the masking filters of each channel are optimized. Since the masking filters are also designed to have linear phase, only half of the coefficients of  $H_{Ma1}$  and  $H_{Mc1}$  need to be concatenated to get the initial vector for the optimization of channel 1 masking filters. Similarly, only half of the coefficients of  $H_{Ma21}$ ,  $H_{Mc21}$  and  $H_{Ma22}$  need to be concatenated to get the initial vector for the optimization of channel 2 masking filters. The optimized masking filters along with the optimized prototype filter result in totally multiplier-less two channel filters.

In all the optimization algorithms discussed below, the average number of non-zero bits in a filter coefficient are fixed to be 2.

### 8.1 Design of Multiplier-less channel filters using DE Algorithm

A modified integer coded DE algorithm is proposed for the optimization of FRM filter in [16]. This modified integer coded DE algorithm is used in the optimization of the channel filters. The various steps of DE algorithm are explained below [19].

**8.1.1 Initialization.** A population of size N is generated by randomly perturbing the initial vector. To have a wider search space, the initial number of vectors in the population is taken as the integer multiple of N chosen. A typical vector in the  $G^{th}$  generation is represented as

$$x_{i,G} = [x_{1i,G} \ x_{2i,G} \ x_{3i,G} \ \dots \ x_{Di,G}]$$

where, D is the dimension of the solution vector.

**8.1.2 Prioritized enlisting of solution vectors using the fitness function.** The fitness of the candidate solutions is evaluated using the objective function given in Section 6.2 and only N number of the best solutions are passed to the next stage [16].

**8.1.3 Mutation.** In this process, a new parameter vector called mutant vector is generated for each target vector  $x_{i,G}$ . This is done by adding the scaled difference between two population vectors to a third vector. Three vectors,  $x_{r_1,G}$ ,  $x_{r_2,G}$  and  $x_{r_3,G}$  are randomly chosen and for each target vector,  $x_{i,G}$ , the mutant vector is generated as given below.

$$v_{i,G+1} = x_{r_1,G} + [(x_{r_2,G} - x_{r_3,G})F], \quad i \in 1, 2, \dots, N \quad (14)$$

where  $r_1, r_2, r_3$  are the different random indices lying between  $\{1, 2, \dots, N\}$  and are different from the running index  $i$ . F is the scaling factor which controls the amplification of difference to be added and varies in the range  $[0, 2]$ .  $[\cdot]$  operation ensures that the values obtained from the mutation operation are integers. Also, to ensure that the values in the new vector is within the boundaries of look up table, following step is also included [16].

$$v_{ji,G+1} = \begin{cases} v_{lb} & \text{if } v_{ji,G+1} < v_{lb} \\ v_{ub} & \text{if } v_{ji,G+1} > v_{ub} \end{cases} \quad (15)$$

where,  $v_{lb}$  and  $v_{ub}$  respectively are the lower bound and upper bound of look up table and  $j \in 1, 2, \dots, D$

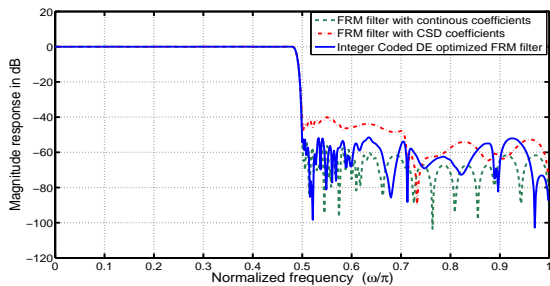
**8.1.4 Crossover.** Now, the parameters of the vector obtained from the mutation step is mixed with those of the target vector to obtain the trial vector,  $u_{i,G+1} = [u_{1i,G+1} \ u_{2i,G+1} \ u_{3i,G+1} \ \dots \ u_{Di,G+1}]$ . The mixing of vectors is controlled using a parameter called crossover ratio, CR, which is in the range  $[0, 1]$  and is done as below

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if } rand_i \leq CR \text{ or } j = jrand_i \\ x_{ji,G} & \text{otherwise} \end{cases} \quad (16)$$

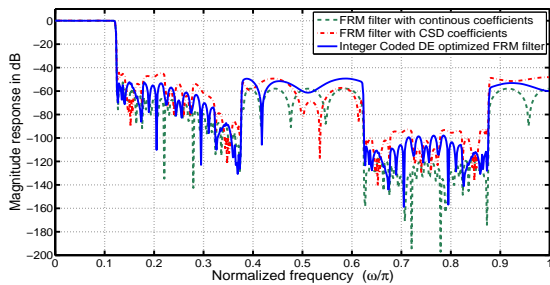
where,  $jrand_i$  is a random integer in the range  $[1, D]$  and  $rand_i$  is a random value in the range  $[0, 1]$ .

**Table 2. Parameters of Integer Coded DE Algorithm**

	Population Size, N	Scaling Factor, F	Crossover Ratio, CR	Number of Generations
Prototype Filter	50	0.8	0.05	300
Channel 1 Masking Filters	50	0.8	0.05	300
Channel 2 Masking Filters	50	0.6	0.05	300



**Fig. 5. Magnitude response of the Channel 1 filter designed using the integer coded DE**



**Fig. 6. Magnitude response of the Channel 2 filter designed using the integer coded DE**

**8.1.5 Selection.** The fitness of the target vector and the trial vector is evaluated using the objective function given in Section 6.2 and is compared. If the trial vector has a better fitness, the target vector will be replaced by the trial vector in the next generation.

**8.1.6 Termination.** The steps from 8.1.3 to 8.1.5 are repeated until the specified number of iterations are reached. Once the algorithm is terminated, the best solution is taken and is decoded to get the optimum filters.

The different parameters used for DE algorithm are shown in Table 2. The magnitude responses of the integer coded DE optimized channel 1 and channel 2 filters are shown in Fig.5 and Fig.6 respectively.

## 8.2 Design of multiplier-less channel filters using HSA algorithm

The various steps of HSA algorithm are listed below [17].

**8.2.1 Initialization of Harmony Memory.** Different solutions are generated by perturbing the initial harmony vector. To have a wider search space, the initial number of harmony memory locations is taken as the integer multiple of harmony memory size (HMS) chosen. A typical harmony vector of size D in the  $k^{th}$  location of the harmony matrix can be represented as:

$$x_k = [x_{k,1} \ x_{k,2} \ x_{k,3} \ \dots \ x_{k,D}]$$

**Table 3. Parameters of Integer Coded HSA**

	Harmony Memory Size (HMS)	Harmony Considering Rate (HMCR)	Pitch Adjusting Rate (PAR)	Number of Generations
Prototype Filter	50	0.9	0.01	1000
Channel 1 Masking Filters	50	0.9	0.01	1000
Channel 2 Masking Filters	50	0.9	0.02	1000

where D is the number of optimization variables.

**8.2.2 Prioritized enlisting of harmony memory locations using the fitness function.** The fitness function is evaluated for each harmony vector, and the 'HMS' number of best solutions will be passed to the next stage.

**8.2.3 Harmony Improvisation.** Harmony improvisation is based on the following steps [17]:

**Memory Consideration:** Values of the decision variable  $x_i^{new}$  in the new vector are chosen from any of the values stored in the harmony memory in the range  $x_{1,i}, x_{2,i}, \dots, x_{HMS,i}$  with probability equal to the HMCR.

**Pitch Adjustment:** Every new decision variable obtained in the memory consideration step is examined to decide whether it should be pitch adjusted. This is decided by the parameter PAR. If pitch adjustment has to be done, it is done as shown below:

$$x_i^{new} = x_i + [rand(1, -1)FW(i)] \quad (17)$$

where FW(i) is an arbitrary distance bandwidth for the  $i^{th}$  design variable.  $[.]$  operation ensures that the new values obtained from the pitch adjustment step are integers.

**Random Selection:** Generate random decision variables for the new vector with a probability equal to (1- HMCR).

**8.2.4 Update the harmony memory.** Now the fitness function is evaluated and if the fitness of the newly generated harmony vector is better than the worst harmony in the harmony memory (HM), the worst harmony is replaced by the new harmony.

**8.2.5 Check the stopping criterion.** The HSA is terminated when the specified number of iterations are reached. Otherwise, the steps given in sections 8.2.3 and 8.2.4 are repeated. After algorithm terminates, fitness is evaluated for all harmony vectors which are retained and the best solution is taken and it is encoded to get the optimized filter coefficients.

The various parameters used for the integer coded HSA optimization of multiplier-less FRM prototype filter and masking filters of channel 1 and channel 2 are shown in Table 3. The magnitude responses of the integer coded HSA optimized channel 1 and channel 2 filters are shown in Fig.7 and Fig.8 respectively.

## 8.3 Design of multiplier-less channel filters using GSA algorithm

The various steps of GSA algorithm are listed below [18].

**8.3.1 Initialization of agents.** Let there be a system of N masses (agents) and the position of the  $i^{th}$  agent is defined as

$$x_i = (x_i^1, \dots, x_i^d, \dots, x_i^N) \quad \text{for } i = 1, 2, 3, \dots, N$$

where  $x_i^d$  represents the position of the  $i^{th}$  agent in the  $d^{th}$  dimension. Different agents are generated by randomly perturbing the initial solution vector. To have a wider search space, the initial number of agents is taken as the integer multiple of N.

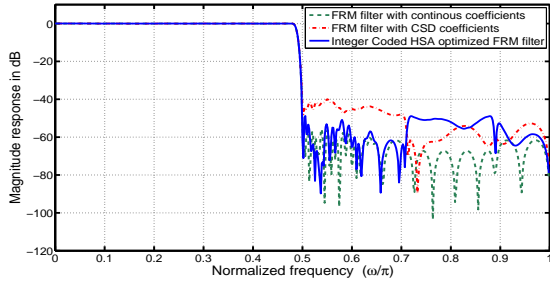


Fig. 7. Magnitude response of the Channel 1 filter designed using the integer coded HSA

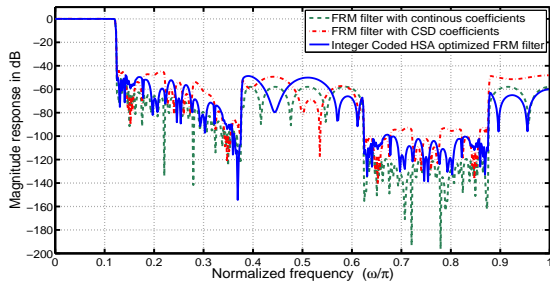


Fig. 8. Magnitude response of the Channel 2 filter designed using the integer coded HSA

8.3.2 *Sort the agents based on fitness value.* The fitness value of agents is calculated using the objective function and sort the agents based on the fitness value. Only N number of agents with the best fitness will be passed to the next step.

8.3.3 *Evaluate fitness and update gravitational constant, G.* In each iteration, the best fitness, best(t) and worst fitness, worst(t) of the agents are updated using the equation given below.

$$worst(t) = \max_{j \in \{1, \dots, N\}} fit_j(t) \quad (18)$$

$$best(t) = \min_{j \in \{1, \dots, N\}} fit_j(t) \quad (19)$$

where  $fit_i(t)$  represents the fitness value of the agent  $i$  at time  $t$ . The gravitational constant depends on the age of the universe and is reduced with time  $t$ . Gravitational constant  $G$  at each iteration is computed by the following equation:

$$G(t) = G_0 e^{-\alpha t/T} \quad (20)$$

$G_0$  is set to 100,  $\alpha$  is taken as 20 and  $T$  is the total number of iterations.

8.3.4 *Calculate mass of each agent.* Gravitational and inertia masses of an agent are calculated using the fitness function and they are updated by the following equations:

$$M_{ai} = M_{pi} = M_{ii} = M_i, \quad i = 1, 2, 3, \dots, N \quad (21)$$

$$m_i(t) = \frac{fit_i(t) - worst(t)}{best(t) - worst(t)} \quad M_i(t) = \frac{m_i(t)}{\sum_{j=1}^N m_j(t)} \quad (22)$$

where,  $M_{ai}$ ,  $M_{pi}$  and  $M_{ii}$  are the active gravitational mass, passive gravitational mass and inertia mass respectively of the  $i^{th}$  agent.

8.3.5 *Calculate acceleration of each agent.*  $a_i^d(t)$ , the acceleration of the agent  $i$  at time  $t$ , and in dimension  $d$ , is given by:

$$a_i^d(t) = \frac{F_i^d(t)}{M_{ii}(t)} \quad (23)$$

Table 4. Parameters of Integer Coded GSA

	Number of Agents	Gravitational Constant, $G_0$	$\alpha$	Number of Generations
Prototype Filter	50	100	20	300
Channel 1 Masking Filters	50	100	20	300
Channel 2 Masking Filters	50	100	20	300

Total force that acts on agent  $i$  in a dimension  $d$  is equal to the randomly weighted sum of the  $d^{th}$  components of the forces applied from other agents, which is given by

$$F_i^d(t) = \sum_{j=1, j \neq i}^N rand_j F_{ij}^d(t) \quad (24)$$

where  $rand_j$  is a random number in the interval  $[0,1]$ . To avoid the algorithm to trap in a local optimum, in the beginning, algorithm must explore and after some iterations, exploration must be decreased and exploitation must be increased. Thus, the performance of GSA can be improved by modifying  $F_i^d(t)$  such that only the Kbest agents will attract the others. Kbest is a function of time, with the initial value  $K_0$  at the beginning and decreases with time. Thus, at the beginning, all agents apply the force, and after a number of iterations, Kbest is decreased linearly and at the end there will be only one agent applying force to the others. Therefore, Eq. (24) for  $F_i^d(t)$  can be modified as:

$$F_i^d(t) = \sum_{j \in Kbest, j \neq i}^N rand_j F_{ij}^d(t) \quad (25)$$

where Kbest is the set of the first K agents with the best fitness value i.e, the biggest mass. At a particular time  $t$ , the force acting on mass  $i$  from mass  $j$  can be defined as

$$F_{ij}^d(t) = G(t) \frac{M_{pi}(t)M_{aj}(t)}{R_{ij}(t) + \epsilon} (x_j^d(t) - x_i^d(t)) \quad (26)$$

where,  $R_{ij}(t)$  is the distance between two agents  $i$  and  $j$  and  $\epsilon$  is a small constant.

8.3.6 *Update the velocity and position of each agent.* The next position and velocity of an agent can be calculated as follows:

$$v_i^d(t+1) = rand_i v_i^d(t) + a_i^d(t) \quad (27)$$

$$x_i^d(t+1) = \lfloor x_i^d(t) + v_i^d(t+1) \rfloor \quad (28)$$

where  $rand_i$  is a uniform random variable in the interval  $[0, 1]$ .  $\lfloor \cdot \rfloor$  denotes rounding to the lower value. This operation ensures that the new candidate solution is an integer.

8.3.7 *Termination.* The steps given in the sections 8.3.3 to 8.3.6 are repeated until the maximum number of iterations is reached. The agent with the best fitness is taken and is decoded to obtain the optimum filters in the CSD space.

The various parameters used for integer coded GSA optimization of multiplier-less prototype filter and channel 1 and channel 2 masking filters is shown in Table 4. The magnitude responses of the integer coded GSA optimized channel 1 and channel 2 filters are shown in Fig.9 and Fig.10 respectively.

#### 8.4 Design of multiplier-less channel filters using ABC algorithm

An integer coded algorithm for the optimization of transmultiplexer is discussed in [15]. A modified integer coded ABC algorithm is proposed for the optimization of FRM filter in [16]. This modified integer coded ABC algorithm is extended to be used in

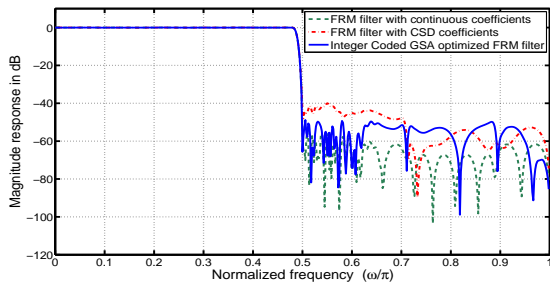


Fig. 9. Magnitude response of the Channel 1 filter designed using the integer coded GSA

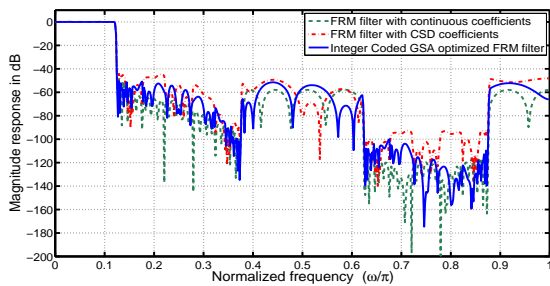


Fig. 10. Magnitude response of the Channel 2 filter designed using the integer coded GSA

the optimization of the channel filters. The various steps of ABC algorithm are given below [14].

**8.4.1 Initialize the food sources.** Filter coefficients are concatenated to get the initial food source. Other different food sources are generated by randomly perturbing the initial food source. To have wider search space, the initial number of food sources is taken as the integer multiple of the number of employed bees.

**8.4.2 Sort the food sources based on fitness value.** The nectar amount of each food source which represents the fitness of the solution vectors are calculated. The solution vectors are sorted based on their fitness value and only N vectors will be passed to the next step [16].

**8.4.3 Employee Bee Phase.** Each employed bee is associated with a food source i.e. a solution vector. Employed bees search for new food sources in the neighbourhood of the current food sources i.e., they try to find out new solution vectors which are adjacent to the existing vectors. A new food source adjacent to the  $i^{th}$  food source is found by changing the randomly chosen  $j^{th}$  value in  $i^{th}$  food source as in the equation given below:

$$v_{ij} = x_{ij} + \phi_{ij}(x_{ij} - x_{kj}) \quad (29)$$

where  $\phi_{ij}$  is a random value in the range [-1, 1], j is a random value in the range (1, 2, 3, ..., D) and k is a random value in the range (1, 2, 3, ..., N). D is the dimension of the solution vectors and N is the number of employed bees. To ensure that the newly generated value is an integer the above equation is modified as below

$$v_{ij} = x_{ij} + \lfloor \phi_{ij}(x_{ij} - x_{kj}) \rfloor \quad (30)$$

Also, to ensure that the values in the new vector is within the boundaries of the look up table, the following step is also included [16].

$$v_{ij,G+1} = \begin{cases} v_{lb} & \text{if } v_{ij,G+1} < v_{lb} \\ v_{ub} & \text{if } v_{ij,G+1} > v_{ub} \end{cases} \quad (31)$$

Table 5. Parameters of Integer Coded ABC

	Number of Food Sources	Limit	Number of Generations
Prototype Filter	50	200	300
Channel 1 Masking Filters	50	200	300
Channel 2 Masking Filters	50	250	300

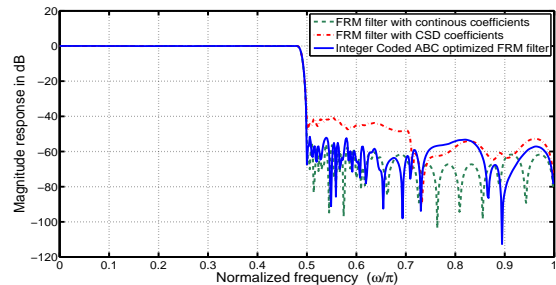


Fig. 11. Magnitude response of the Channel 1 filter designed using the integer coded ABC

Now, a greedy selection mechanism is used to select between the old and new solution vectors. The fitness value of the new vector representing the nectar amount of the new food source is calculated and it is compared with that of the existing food source. If the fitness value of the new vector is equal to or greater than the existing one, it is replaced with the new vector, otherwise, it is retained.

**8.4.4 Onlooker Bee Phase.** The employed bees share the information of their food source with the onlooker bees. The onlooker bees calculate the nectar information collected from all employed bees and depending on the nectar amount they select the food source. If the nectar amount of a food source is high, the probability of selecting that food source by an onlooker bee is also more. This indicates that if the fitness of a solution vector is high, the probability of selecting that vector by an onlooker bee is also more. An onlooker bee selects a food source depending on the probability distribution function associated with that food source. So the solution vector with high fitness value will get more onlooker bees. Now the onlooker bees search for better food sources in the neighbourhood of the current food source and it will be selected or rejected depending on the fitness value. The search and the selection/rejection of the new food source is done by the same method as in the employee bee phase.

**8.4.5 Scout Bee Phase.** If the fitness value of a solution vector representing the food source is not improved after a fixed number of iterations, called limit cycles, that food source will be abandoned and the associated employee bee becomes the scout bee. The scout bee randomly finds a new food source and it occupies the position of abandoned food source.

**8.4.6 Termination.** The steps from 8.4.2 to 8.4.5 are repeated until the number of predetermined maximum number of iterations is reached. Once the algorithm terminates, the solution vector with the best fitness is taken and is decoded to obtain the optimum filters filter in the CSD space.

The various parameters used for the integer coded ABC optimization of multiplier-less prototype filter and channel 1 and channel 2 masking filters are shown in Table 5. The magnitude responses of the integer coded ABC optimized channel 1 and channel 2 filters are shown in Fig.11 and Fig.12 respectively.



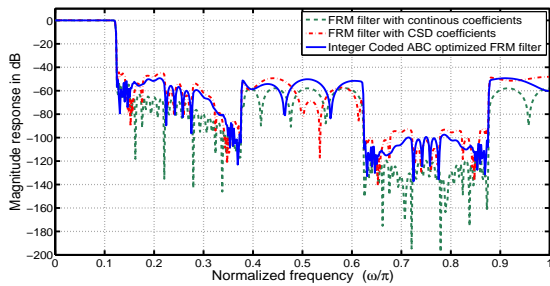


Fig. 12. Magnitude response of the Channel 2 filter designed using the integer coded ABC

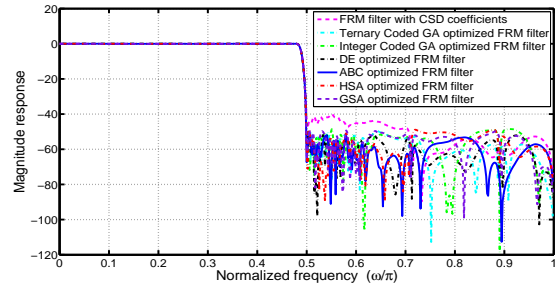


Fig. 13. Magnitude response of the Channel 1 filter designed using various algorithms

Table 6. Parameters of Ternary Coded GA

	Pop. Size	Mut. Rate	Popkeep Fraction	No: of Elite Chromosomes	No: of Generations
Prototype Filter	50	0.1	0.3	20	500
Channel 1 Masking Filters	50	0.1	0.3	20	500
Channel 2 Masking Filters	50	0.1	0.3	20	500

Table 7. Parameters of Integer Coded GA

	Pop. Size	Mut. Rate	Popkeep Fraction	No: of Elite Chromosomes	No: of Generations
Prototype Filter	50	0.02	0.3	5	300
Channel 1 Masking Filters	50	0.02	0.3	5	300
Channel 2 Masking Filters	50	0.02	0.4	20	300

## 9. RESULT ANALYSIS

Design of reconfigurable two channel filters using ternary coded GA algorithm is discussed in [8]. The use of the integer coded GA algorithm for the optimization of transmultiplexer is illustrated in [13]. Optimization of FRM filter using modified integer coded ABC and modified integer coded DE algorithm are discussed in [16]. In this paper, CSD encoded reconfigurable channel filters based on FRM are designed using various meta-heuristic algorithms like modified Integer Coded DE, modified Integer Coded ABC, modified Integer Coded HSA and modified Integer Coded GSA algorithms. The parameters are compared with those of Ternary Coded GA and Integer Coded GA.

The various parameters used for the optimization of CSD rounded FRM filter using ternary coded GA and integer coded GA are shown in Table 6 and 7 respectively. The frequency response parameters of the various optimization techniques are compared in Table 8 and Table 9. The performances are compared in terms of the average pass-band ripple and stop-band attenuation for 10 simulations.

In Table 8, the average pass-band ripple obtained for various algorithms for 10 simulations are compared with those of the continuous and CSD represented channel filters. It is seen that, the ABC algorithm gives the lowest pass-band ripple for channel 1 and GSA gives the lowest passband ripple for channel 2 compared to all other optimization algorithms.

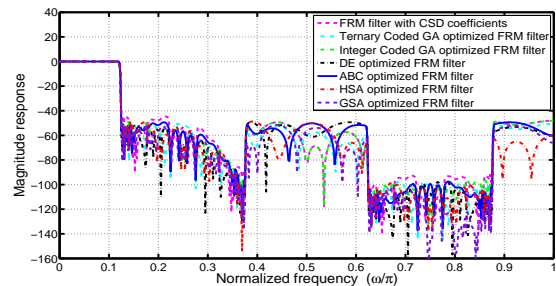


Fig. 14. Magnitude response of the Channel 2 filter designed using various algorithms

Table 8. Pass-band (PB) ripple of the channel filters (Average of 10 simulations)

Optimization Techniques	Average PB Ripple (dB)	
	Channel 1	Channel 2
Continuous Coefficients	0.02156	0.08502
CSD Rounded	0.05625	0.1228
Ternary Coded GA [8]	0.04702	0.04946
Integer Coded GA	0.05571	0.07403
Integer Coded DE	0.03381	0.0628
Integer Coded ABC	0.02775	0.04504
Integer Coded HSA	0.04424	0.04528
Integer Coded GSA	0.04658	0.043

The average stop-band attenuation obtained for 10 simulations when different optimization algorithms are used is compared in Table 9. ABC optimized channel 1 and channel 2 filters have the maximum average stop-band attenuation. The magnitude responses of the various optimization techniques for channel 1 and channel 2 are shown in Fig.13 and Fig.14 respectively.

In Table 10, the average of the minimum cost obtained for 10 simulations for channel 1 and channel 2 is presented. The ABC algorithm gives minimum value for the average minimum cost for the objective function for both channels.

Table 11 compares the run time of the various optimization techniques for an average of 10 simulations. The average run time of the GSA algorithm is found to be the minimum and that of the HSA algorithm is the maximum.

In Table 12, the complexity of implementation of the channel filters using various optimization techniques is compared. All the optimization algorithms reduce the complexity of the channel filters compared to that of the continuous coefficient channel filters and than that of the maximum precision channel filters, in which the filter coefficients are directly rounded with maximum number of non-zero SPT terms which is obtained as 6.

**Table 9. Stop-band (SB) Attenuation of channel filters (Average of 10 simulations)**

Optimization Techniques	Average SB Attenuation (dB)	
	Channel 1	Channel 2
Continuous Coefficients	54.81	52.68
CSD Rounded	39.97	44.21
Ternary Coded GA [8]	48.388	47.904
Integer Coded GA	47.723	45.607
Integer Coded DE	49.629	46.927
Integer Coded ABC	50.22	48.983
Integer Coded HSA	47.625	47.427
Integer Coded GSA	47.658	48.104

**Table 10. Minimum cost of various optimization algorithms (Average of 10 simulations)**

Optimization Techniques	Minimum Cost	
	Channel 1	Channel 2
Ternary Coded GA [8]	0.0065364	0.0069335
Integer Coded GA	0.0074541	0.0096806
Integer Coded DE	0.0052932	0.0084675
Integer Coded ABC	0.0046857	0.0062785
Integer Coded HSA	0.0067334	0.0073753
Integer Coded GSA	0.0068436	0.006479

**Table 11. Total Run time for various optimization algorithms (Average of 10 simulations)**

Optimization Techniques	Run Time (Seconds)
Ternary Coded GA [8]	231.2128
Integer Coded GA	201.7026
Integer Coded DE	187.436
Integer Coded ABC	213.3106
Integer Coded HSA	264.905
Integer Coded GSA	104.531

**Table 12. Complexity Comparison of the channel filters using various optimization algorithms**

Optimization Techniques	No: of SPT Terms	Adders due to SPT Terms	No: of Adders	Total No: of Adders	No: of Multipliers
Continuous Coefficients [7]	-	-	106	106	112
Maximum Precision	266	157	106	263	0
CSD Rounded	245	141	106	247	0
Integer Coded GA	246	137	106	243	0
Integer Coded DE	245	142	106	248	0
Integer Coded ABC	247	138	106	244	0
Integer Coded HSA	241	132	106	238	0
Integer Coded GSA	244	135	106	241	0

It is seen that the ABC algorithm gives low complexity totally multiplier-less channel filters with very good frequency specifications. But the running time of the ABC algorithm is more. From the tables given it can be seen that GSA algorithm converges faster and also gives acceptable values for the frequency specifications. So, when the running time is of primary concern, GSA algorithm can be chosen which also uses less number of adders compared to ABC algorithm. But, if the application demands better frequency specifications, ABC algorithm can be selected.

## 10. CONCLUSION

The reconfigurable channel filters based on FRM gives non-uniform bandwidth channel filters with much reduced complexity. Design of channel filters in the canonic signed digit space leads to totally multiplier-less channel filters. Modified meta-heuristic algorithms are proposed to improve the performance of the channel filters. For the given specifications, it is observed that ABC results in the best frequency response characteristics. In terms of computational complexity, all the proposed algorithms result in multiplier-less non-uniform channel filters. But the number of adders is the least when ABC is deployed. In terms of run time, Gravitational Search Algorithm is found to be more suitable to get faster optimization with acceptable values of frequency response characteristics and complexity.

## 11. REFERENCES

- [1] Mitola J, Software radio architecture, John Wiley, 2000.
- [2] Hentschel T., Henker M., Fettweis G, The digital front-end of software radio terminals, IEEE Person. Commun. Magazine, 1999, pp.40-46.
- [3] Lim Y. C., Frequency-response masking approach for the synthesis of sharp linear phase digital filters, IEEE Trans. Circuits. Syst., Apr. 1986, vol. 33, pp. 357-364.
- [4] Mahesh R., A. P. Vinod, Edmund M-K. Lai and Amos Omondi, Filter Bank Channelizers for Multi-Standard Software Defined Radio Receivers, Journal of Signal Processing Systems, February 2011, Volume 62, Issue 2, Page 157-171
- [5] Mahesh R. and A. P. Vinod, Reconfigurable Low Area Complexity Filter Bank Architecture Based on Frequency Response Masking for Non-uniform Channelization in Software Radio Receivers, IEEE Transactions on Aerospace and Electronic Systems, April 2011, vol. 47, no. 2, 1241-1255.
- [6] Smitha K.G. A. P. Vinod, A Low Complexity Reconfigurable Multi-stage Channel Filter Architecture for Resource-Constrained Software Radio Handsets, J Sign Process Systems, 2011, 217-231.
- [7] Smitha K.G., R. Mahesh and A. P. Vinod, A Reconfigurable Multi-stage Frequency Response Masking Filter Bank Architecture for Software Defined Radio Receivers, IEEE International Symposium on Circuits and Systems, 2008, 85-88.
- [8] Bindya T. S, V. Satish Kumar and Elizabeth Elias, Design of Low power and Low Complexity Multiplier-less Reconfigurable Non-uniform Channel filter using Genetic Algorithm, Global Journal of Research in Engineering, May 2012, Vol. 12, Issue. 6.
- [9] Saramaki T. and Lim Y.C., Use of Remez algorithm for designing FRM based FIR filters, Circuits Syst. Signal Process, 2003, 22 (2), 77-97.
- [10] Yong Ching Lim D. L., Rui Yang and J. Song, Signed power-of-two term allocation scheme for the design of digital filters, IEEE Transactions on Circuits and Systems -

- Analog and Digital Signal Processing, 1999, vol. II, pp. 577-584.
- [11] Hartley R.I., Subexpression sharing in filters using canonic signed digit multipliers, *IEEE Trans. Circuits Syst. II*, 1996, vol. 43, no. 10, pp. 677-688.
- [12] Fuller A, B. Nowrouzian, and F. Ashrafzadeh, Optimization of FIR digital filters over the canonical Signed-digit coefficient space using genetic algorithms, in *Proceedings of the 1998 Midwest Symposium on Circuits and Systems*, August 1999, pp. 456-469.
- [13] Manoj, V.J., Elizabeth, E., Design of multiplier-less nonuniform filter bank transmultiplexer using genetic algorithm, *Signal Processing*, 2009, 89(11), 2274-2285.
- [14] Karaboga, D., Basturk, B., On the performance of artificial bee colony (ABC) algorithm, *Appl. Soft Comput.*, 2008, 8(1), 687-697.
- [15] Manoj, V.J., Elias, E., Artificial bee colony algorithm for the design of multiplier-less nonuniform filter bank transmultiplexer, *Inf. Sci.*, doi: 10.1016/j.ins.2011.02.023.
- [16] Manju Manuel and Elizabeth Elias, Design of frequency response masking FIR filter in the Canonic Signed Digit space using modified Artificial Bee Colony algorithm, *Engineering Applications of Artificial Intelligence*, doi:10.1016/j.engappai.2012.02.010.
- [17] Kang Seok Lee, Zong Woo Geem, A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice, *Comput. Methods Appl. Mech. Engrg.*, 2005, 194, 3902-3933
- [18] Esmat Rashedi, Hossein Nezamabadi-pour, Saeid Saryazdi, GSA: A Gravitational Search Algorithm, *Information Sciences*, 2009, Vol. 179, No.13, 2232-2248.
- [19] Rainer Storn and Kenneth Price, Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, *Journal of Global Optimization*, December 1997, Volume 11, Issue 4, 341-359
- [20] Geem Z. W., 2009. *Music-Inspired Harmony Search Algorithm: Theory and Applications*. New York: Springer-Verlag, ser. *Studies in Computational Intelligence Series*.
- [21] Yu Y.J., Y.C. Lim, A novel genetic algorithm for the design of a signed power-of-two coefficient quadrature mirror filter lattice filter bank, *Circuit Syst. Signal Process*, 2002, 21 (3), 263-276.
- [22] Boyd, S., Vandenberghe, L., *Convex Optimization*, Cambridge University Press, Cambridge, 2004.
- [23] Bellanger M., On computational complexity in digital filters, in *Proc. Of the European Conference on Circuit Theory and Design*, Hague, Netherlands, pp. 58-63, August 1981.