

# Design of multiplier-less sharp transition width MDFT filter banks using modified metaheuristic 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

The significant advantage of modified discrete Fourier transform (MDFT) filter banks over the conventional discrete Fourier transform (DFT) filter banks is the structure inherent alias cancellation in the former. When the number of channels is increased, the filters in the filter bank need to be of sharp transition width. This increases the complexity of the filters and hence that of the filter bank. Frequency Response Masking (FRM) approach is known to reduce the complexity of sharp transition width filters. This paper proposes a method to realize MDFT filter banks using FRM with much lesser complexity. To further reduce the complexity, the filter banks are made totally multiplier-less. This is done by converting the coefficients to the canonic signed digit (CSD) representation. Metaheuristic algorithms are used to improve the performance of the CSD represented filter banks. Modified integer coded genetic algorithm, differential evolution, artificial bee colony, harmony search and gravitational search algorithms are proposed to be used for the optimization of the proposed multiplier-less MDFT filter banks. This design method reduces the complexity, power consumption and chip area for the implementation of the uniform filter banks.

## General Terms:

Frequency response masking, Optimization

## Keywords:

Frequency response masking, Modified discrete Fourier transform filter banks, Canonic signed digit, Meta-heuristic algorithms, Genetic algorithm, Differential evolution, Artificial bee colony, Harmony search algorithm, Gravitational search algorithm

## 1. INTRODUCTION

Filter banks play an important role in many digital signal processing applications, such as audio and image coding. They are used for the extraction of the spectral components of a signal. An M-channel filter bank decomposes the input signal into M sub-band signals by applying M analysis filters with different pass-bands. Thus, each of the sub-band signals carries information of the input signal in a particular frequency band.

Among the different filter bank structures, modulated filter banks are the most popular, because of their simple design and ease of

implementation. There are two classes of modulated filter banks, Discrete Fourier Transform (DFT) polyphase filter bank and Cosine Modulated filter bank. DFT filter bank consists of analysis and synthesis filter banks, which are generated from the same prototype filter by exponential modulation. This makes the DFT filter bank very easy to implement, but they do not have any alias cancellation structure. Also, for M-channel uniform filter banks, it is required that for perfect reconstruction, the analysis and synthesis poly-phase matrices are invertible. This disadvantage can be overcome by modifying the DFT filter banks, which results in the Modified DFT (MDFT) filter banks [1, 2, 3]. In MDFT filter banks, a structure inherent alias cancellation can be obtained, which results in near perfect reconstruction (PR) filter banks. MDFT filter banks are modified complex modulated, critically sub-sampled filter banks based on the DFT filter banks.

As the number of channels increases, the transition width of the prototype filter has to be narrow. This results in a prototype filter of very high order. To avoid this problem, the frequency response masking (FRM) technique [4] can be used for the design of the prototype filter. All the analysis and synthesis filters of the MDFT filter bank are derived from the same prototype filter. Design of the MDFT filter banks using the frequency response masking technique is discussed in [5]. FRM reduces the number of multipliers while designing sharp transition width filters with arbitrary bandwidth. The design of FRM filter involves the design of a band edge shaping filter, masking filter and complementary masking filter.

Multipliers in a digital filter are the main power consuming components and they occupy large chip area. If the filter coefficients are represented in the signed power of two (SPT) space, multipliers can be replaced by shift and add operations [6]. Replacing multipliers with shift and add operations results in the reduction of power consumption and chip area. We can further reduce the filter complexity by reducing the number of non-zero SPT terms, which results in the reduction of the number of adders. Canonic signed digit (CSD) representation [7] is a special case of the SPT space, which uses both additions and subtractions. Because of this, only a minimum number of non-zero SPT terms is needed to represent a decimal number.

Representing the filter coefficients in the CSD space can bring down the complexity of the FRM prototype filter and hence that of the MDFT filter bank. The filter coefficients rounded to finite precision values in the SPT space may degrade the FRM filter per-

formance. This can cause deterioration of the MDFT filter bank characteristics. This calls for efficient optimization techniques in the discrete space. Classical gradient based optimization techniques cannot be applied in the discrete space, since here the search space contains integers. Metaheuristic algorithms are good alternatives for this type of optimization problems since these finally can reach a global solution if the parameters are properly selected with respect to a particular design problem.

Genetic algorithm (GA) [8] is an optimization technique, which can deal with discrete search spaces. GA models the evolution process of natural selection, where, in each generation, candidates are modified by genetic operations like cross over, selection and mutation. A ternary coded GA based on CSD look-up-table has been successfully used for the optimization of FRM based reconfigurable channel filters [9]. Operations like crossover and mutation may result in non-canonical bit strings. So, restoration algorithms [10] are needed to approximate the non-canonical filter coefficients to the nearest canonical representation. This increases the computational complexity of ternary coded GA. A modified GA in which the binary representation of the indices of the CSD look-up-table entries are used to obtain the initial seed, is reported [11], where there is no need of restoration algorithms. Integer coded GA in which the integer indices of the look-up-table entries are used to get the solution, is proposed by [12].

Differential evolution [13], artificial bee colony (ABC) [14], harmony search [15, 16] and gravitational search [17] algorithms are some of the best known recently developed metaheuristic algorithms which are found to outperform GA in most of the cases. Integer coded artificial bee colony (ABC) algorithm is used for the optimization of multiplier-less transmultiplexer in [18]. A modified integer coded ABC and integer coded differential evolution (DE) algorithms for the design of CSD encoded FRM filter are presented in [19]. In Paper [20], harmony search algorithm (HSA) and gravitational search algorithm (GSA) are modified for the optimization of the multiplier-less reconfigurable filter banks. In this paper, modified DE, ABC, HSA and GSA algorithms are used for the design of the optimal multiplier-less FRM prototype filter and hence the MDFT filter banks. The performances of these algorithms are compared with that of GA in terms of the number of adders needed to implement the MDFT filter bank, stop-band and pass-band responses of the prototype filter and amplitude distortion of the MDFT filter bank.

This paper is organized as follows. Section 2 gives an overview of MDFT filter banks. In Section 3 the FRM approach is briefed. Section 4 provides the description of the CSD representation. Section 5 gives a brief overview of the optimization techniques, HSA and GSA. Section 6 gives the design of the continuous coefficient FRM filter and MDFT filter bank. This section also discusses the design of optimal multiplier-less MDFT filter bank using meta-heuristic algorithms. The complexity of implementation and performance evaluation are also presented in Section 6. Section 9 gives the conclusion of the paper.

## 2. OVERVIEW OF MDFT FILTER BANKS

In the DFT filter banks, synthesis and analysis filters are derived from the prototype filter by complex modulation. The structure of DFT filter banks is shown in Fig. 1 [1], in which  $H_k(z)$  and  $F_k(z)$  represent the analysis and synthesis filters respectively and  $\uparrow M$  and  $\downarrow M$  represent interpolation and decimation by  $M$  respectively. Here, the synthesis and analysis filters are derived from a linear

phase finite impulse response (FIR) prototype filter  $H(z)$  which is band limited to  $2\pi/M$ , by complex modulation. The analysis filters are represented as

$$H_k(z) = H(zW_M^k) \quad (1)$$

Similarly, the synthesis filters are defined as

$$F_k(z) = MH(zW_M^k) \quad (2)$$

where  $W_M = e^{-j2\pi/M}$ . If the input  $X(z)$  is passed through the filter bank, the reconstructed signal, can be defined as [1]

$$\hat{X}(z) = \frac{1}{M} \sum_{k=0}^{M-1} F_k(z) \sum_{l=0}^{M-1} H(zW_M^{k+l}) X(zW_M^l) \quad (3)$$

Since the prototype filter is band limited to  $2\pi/M$ , all non-adjacent alias components are ignored. Considering only adjacent alias components, the reconstructed signal can be written as [1]

$$\hat{X}(z) = \frac{1}{M} \sum_{k=0}^{M-1} F_k(z) \sum_{l=-1}^1 H(zW_M^{k+l}) X(zW_M^l) \quad (4)$$

From Eq. 4, we can see that the output contains alias components. In the DFT filter banks, there is no inherent mechanism to cancel out the alias signals. So the DFT filter banks do not give perfect reconstruction due to aliasing and linear distortion. This disadvantage can be removed by introducing some modifications to the DFT filter banks which leads to the MDFT filter banks.

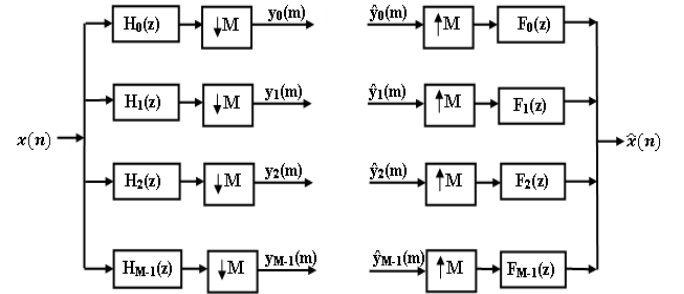


Fig. 1. The structure of DFT filter bank

The MDFT filter bank [1, 2, 3] can be derived from a complex modulated filter bank by decimating the sampling rate with and without a delay of  $M/2$  samples and using either the real or the imaginary part, alternately, in the sub-bands. These modifications eliminate directly adjacent alias spectra which are the main components of aliasing in the DFT filter banks. Non-adjacent alias terms can be kept small by designing the prototype filter with sufficiently high stop-band attenuation. Thus MDFT filter bank gives almost perfect reconstruction of the input signals. The structure of an M-channel MDFT filter bank is given in Fig. 2 [1].

The alias cancellation is independent of the prototype filter design and only depends on the structure of the MDFT filter bank. This feature leads to a filter bank with almost PR. Aliasing happens from  $M - 1$  neighbouring sub-bands. Therefore, the aliasing distortion function can be written as [3]

$$T_{alias}(z) = \sqrt{\sum_{l=1}^{M-1} \left| \frac{1}{M} \sum_{k=0}^{M-1} F_k(z) H_k(zW_M^l) \right|^2} \quad (5)$$

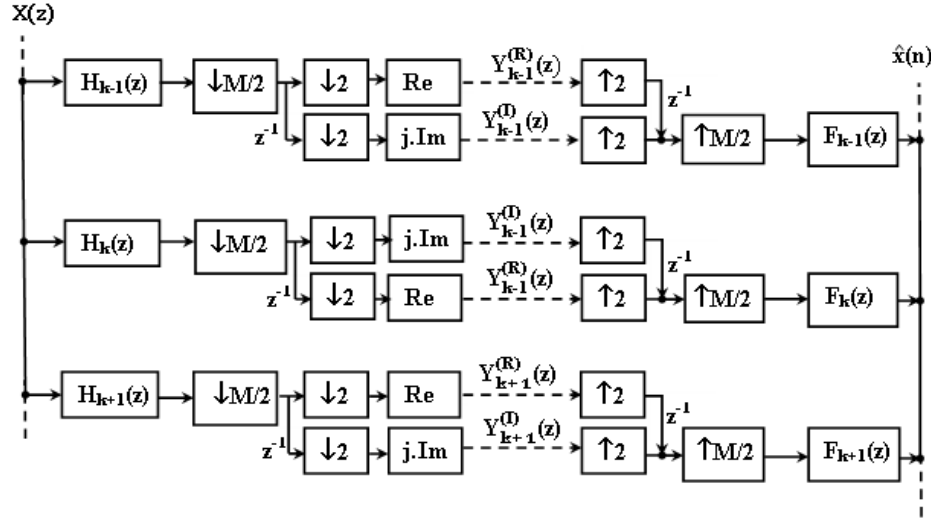


Fig. 2. The structure of MDFT filter bank

$|T_{alias}(z)|$  should tend to zero at all frequencies. Thus, the MDFT filter bank should be suitable to meet the following specifications:

- The transfer functions of adjacent channels must be approximately power complementary between their centre frequencies. This reduces the amplitude distortion in the filter bank.
- All odd alias spectra cancel automatically, i.e., independent of the filter characteristics in the MDFT filter bank. All other alias spectra are suppressed by a sufficiently high stop-band attenuation of the prototype filter. This reduces the aliasing distortion in the MDFT filter bank.
- If the prototype filter is designed as a linear phase filter, all the synthesis and analysis filters of MDFT filter bank will have linear phase. So, the MDFT filter bank is free from phase distortions.

The power complementary property of adjacent channels and the stop-band attenuation depend on the prototype filter design. If the remaining small alias spectra is ignored, the overall transfer function,  $T_{dist}$  of the MDFT filter bank from input to output can be obtained as [3]

$$T_{dist}(z) = \frac{1}{M} \sum_{k=0}^{M-1} F_k(z) H_k(z) \quad (6)$$

### 3. 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

$$H_c(z) = z^{\frac{-(N-1)}{2}} - H_a(z) \quad (7)$$

$H_a(z)$  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  $H(z)$  [4] is given by

$$H(z) = H_a(z^M) H_{Ma}(z) + H_c(z^M) H_{Mc}(z) \quad (8)$$

The structure of the FRM FIR filter is given in Fig. 3 [4]. The design

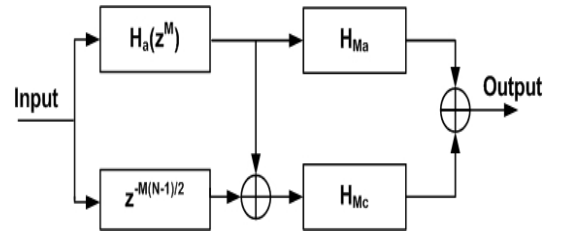


Fig. 3. Basic FRM filter architecture [4]

steps for the sub-filters are given below [4]:

$$m = \lfloor f_p * M \rfloor \quad f_{ap} = f_p M - m \quad f_{as} = f_s M - m \quad (9)$$

$$f_{map} = f_p \quad f_{mas} = \frac{m + 1 - f_{as}}{M} \quad (10)$$

$$f_{mcp} = \frac{m - f_{ap}}{M} \quad f_{mcs} = f_s \quad (11)$$

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.  $\frac{(f_{as}-f_{ap})}{M}$ .

#### 4. CANONIC SIGNED DIGIT REPRESENTATION (CSD)

Any FIR filter can be represented as

$$y(n) = \sum_{k=0}^{N-1} h(k)x(n-k) \quad (12)$$

where  $N$  is the length of the FIR filter,  $h(k)$  are the filter coefficients and  $x(n)$  is the input signal. The FIR filter implementation 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 [6, 7]. A fractional number  $q$  is represented in CSD format as [7].

$$q = \sum_{i=1}^W c_i 2^{R-i} \quad (13)$$

where  $c_i = \{-1, 1, 0\}$  and  $W$  is the word length of the CSD number and the integer  $R$  represents a radix-point in the range  $0 < R < W$ . The features of the CSD format are

- It is a unique SPT representation of a decimal number with minimum number of non-zero bits.
- Since, this encoding uses -1, 0 and 1 digit, it is called ternary coding.
- 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.
- The maximum number of non-zero bits in the CSD representation of an  $n$ -bit number is  $(n+1)/2$ , compared to  $n$  bits in the 2's complement representation.
- The maximum number of adders/subtractors needed to realize the CSD represented  $n$ -bit number will be  $(n+1)/2 - 1$ .

#### 5. OVERVIEW OF HARMONY SEARCH ALGORITHM AND GRAVITATIONAL SEARCH ALGORITHM

This section gives an overview of the harmony search algorithm and gravitational search algorithm.

##### 5.1 Harmony search algorithm (HSA)

Inspired by the music improvisation scheme, Z.Geem introduced the harmony search algorithm [15, 16] 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. On the other hand, in an optimization problem, each decision variable selects a value from the possible range of values which together makes a solution vector. If the solution vector has a good fitness value, that solution will be stored in the memory. The possibility to get a good solution in the next iteration will be increased.

When a musician improvises one pitch, he follows any one of three rules [15]: (1) playing any one pitch from his memory, (2) playing an adjacent pitch of the pitch stored in his memory, and (3) playing totally random pitch from the possible range. Similarly, in the HSA, when each decision variable chooses a value, it follows any one of three rules [15]: (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).

An important feature of HSA is that it does not require differentiable fitness function. So it can handle discrete variables as well as continuous variables. The steps of HSA are described in [20].

##### 5.2 Gravitational search algorithm (GSA)

GSA [17] 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 [17]:

- 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 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 algorithm terminates when a fixed number of iterations is reached. Once the algorithm terminates, the best fitness at final iteration is taken and the positions of the mass of the corresponding agent becomes the global solution of that problem. The various steps of GSA algorithm are given in [20] in detail.

#### 6. DESIGN EXAMPLE

In this work, an FRM filter is designed for the given specifications and by modulating this filter an eight channel MDFT filter bank is designed. The filter bank is made multiplier-less using various metaheuristic algorithms and the performances are compared. 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.

##### 6.1 Design of continuous coefficient MDFT filter bank

The initial phase of this work is the design of the continuous coefficient sharp transition width MDFT filter bank with all the analysis

and synthesis filters having linear phase property. All the analysis and synthesis filters of the MDFT filter bank are derived from the same prototype filter using complex modulation as per Eqs. 1 and 2. Hence, the problem of designing the MDFT filter bank reduces to the problem of designing a single prototype filter. Also, the complexity of implementing the MDFT filter bank is the same as the complexity of implementing the prototype filter. When the number of sub-channels of the MDFT filter bank needs to be very large or when the transition-widths of the sub-channels need to be very narrow, the prototype filter should have sharp transition which results in a high order filter and hence high complexity MDFT filter bank. A substantial reduction in the complexity of the MDFT filter bank can be achieved by designing the prototype filter using the FRM approach as proposed in [5]. All the sub-filters of the FRM filter,  $H_a(z)$ ,  $H_{Ma}(z)$  and  $H_{Mc}(z)$  are realized as per the original work on FRM [4] using the Parks-McClellan algorithm which results in filters with linear phase property.

Let the low pass prototype filter be designed using the following parameters:

$$\begin{aligned} \text{Pass - band(PB) edge frequency} &: 0.124\pi \\ \text{Stop - band(SB) edge frequency} &: 0.127\pi \\ \text{Maximum pass - band ripple} &: 0.004\text{dB} \\ \text{Minimum stop - band attenuation} &: 60\text{dB} \end{aligned}$$

When the prototype filter is designed using the minimax method for this set specifications, the filter order is found to be 2326. The number of multipliers required to implement the filter is found to be 1164. To reduce the complexity of the filter implementation, the prototype filter is realized as an FRM filter. The band-edges of the sub-filters of the FRM filter are obtained as explained in Section 3 and they are designed using Parks-McClellan method [21]. The lengths of the sub-filters  $H_a(z)$ ,  $H_{Ma}(z)$  and  $H_{Mc}(z)$  are obtained as 221, 86 and 88 respectively. The total number of multipliers,  $\Pi$ , required to implement an FRM filter is the sum of the number of multipliers required to implement its sub-filters which can be written as

$$\Pi = \mu_a + \mu_{Ma} + \mu_{Mc} \quad (14)$$

where,  $\mu_a$ ,  $\mu_{Ma}$  and  $\mu_{Mc}$  are the number of multipliers required for the implementation of the filters  $H_a(z)$ ,  $H_{Ma}(z)$  and  $H_{Mc}(z)$  respectively. Using Eq.14, the total number of multipliers in the FRM filter with the above specifications is found to be 198 and hence offers an 83% reduction in the complexity as compared to the traditional minimax method. The zero phase frequency response of the continuous coefficient FRM filter is shown in Fig.4. Now, the eight channel continuous coefficient MDFT filter bank is designed by deriving the analysis filters and synthesis filters using the complex modulation as given in Eqs. 1 and 2. The magnitude response of the continuous coefficient analysis filters of the MDFT filter bank is shown in Fig.5. The amplitude distortion of the continuous coefficient MDFT filter bank is shown in Fig.6. The peak amplitude distortion of the continuous coefficient MDFT filter bank is found to be 0.023dB. The frequency response parameters of the FRM filter and MDFT filter bank along with the complexity of implementation of the FRM filter in terms of the number of multipliers and adders are listed in Table 1.

## 6.2 Design of the multiplier-less MDFT filter bank

The next step is to design the multiplier-less MDFT filter bank for which, it is sufficient to replace the multipliers in the FRM filter implementation with shifters and adders. To this end, the coefficients

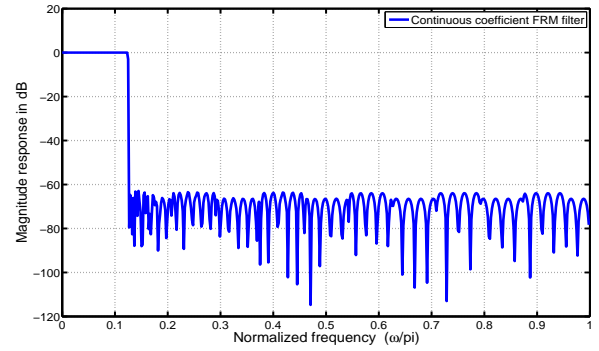


Fig. 4. Magnitude response of the continuous coefficient FRM prototype filter of the MDFT filter bank

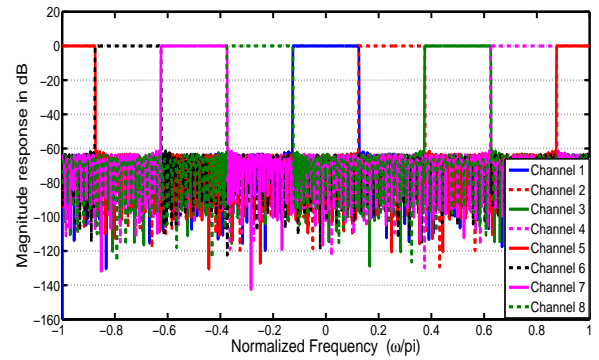


Fig. 5. Magnitude response of the continuous coefficient analysis filters of the MDFT filter bank

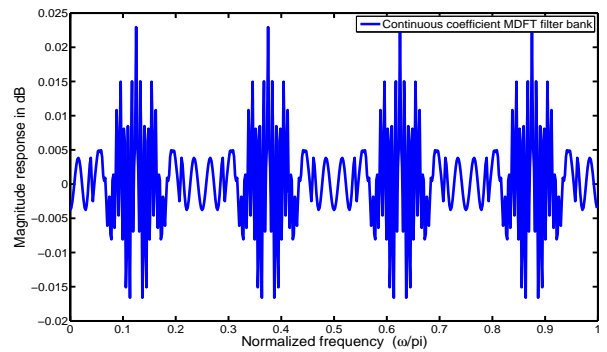


Fig. 6. Amplitude distortion of the continuous coefficient MDFT filter bank

of all the sub-filters of the FRM filter are represented in the CSD space.

**6.2.1 Design of maximum precision FRM filter and MDFT filter bank.** For a given word length, if the filter coefficients are rounded to the nearest CSD representation without any restriction in the number of non-zero bits, it will result in the filter with maximum precision in the CSD space for that word length. The CSD repre-

Table 1. Frequency performance and complexity of the continuous coefficient FRM filter and MDFT filter bank

Parameters	Prototype filter		MDFT filter bank	Complexity of implementation	
	Pass-band ripple(dB)	Stop-band attenuation(dB)	Amplitude distortion(dB)	Number of multipliers	Number of adders
Continuous coefficients [5]	0.00831	62.92	0.02296	198	195

Table 2. Frequency performance and complexity of the maximum precision CSD represented prototype filter and MDFT filter bank when different word-lengths are used

Word-length	Prototype filter		MDFT filter bank	Number of adders due to SPT terms
	Pass-band ripple (dB)	Stop-band attenuation (dB)	Amplitude distortion (dB)	
Continuous coefficients [5]	0.00831	62.92	0.02296	0
12 bits	0.01698	50.24	0.03576	219
14 bits	0.01103	59.81	0.02359	344
16 bits	0.00798	62.9	0.02724	467

sensation of a decimal number using  $n$ -bits of word-length cannot have more than  $(n + 1)/2$  non-zero bits, often it is fewer. For the word-lengths of 12, 14 and 16 bits respectively, the maximum number of non-zero bits that will be present in the CSD representation of a number are 6, 7 and 8 respectively. Table 2 gives the frequency performance parameters of the maximum precision FRM prototype filter and maximum precision MDFT filter bank and the complexity of the maximum precision prototype filter when different word lengths are used for the CSD representation. Here, all the bits are used for the fractional part. The frequency parameters obtained for the maximum precision prototype filter with 16 bits CSD representation are closer to those of the continuous coefficient prototype filter when compared to those of the maximum precision prototype filter with 12 or 14 bits of CSD representation. But the number of adders due to the SPT terms for the maximum precision prototype filter and MDFT filter bank with 16 bits CSD is much higher compared to that of the maximum precision prototype filter and MDFT filter bank with 14 bits CSD. Hence, we have to make some trade-off between the filter response and complexity.

Hence, in this example, a 14 bit CSD representation is used instead of 12 or 16 bits, in order to reduce the complexity without much degradation of the filter performance.

**6.2.2 Design of CSD rounded FRM filter and MDFT filter bank.** Now, to reduce the number of adders in the multiplier-less implementation, the filter coefficients are rounded with restricted number of SPT terms. This causes degradation in the performances of the FRM filter and hence the MDFT filter bank. The magnitude responses of the FRM filter and analysis filters of the MDFT filter bank when the coefficients are CSD rounded using three SPT terms are shown in Figs. 7 and 8 respectively. The amplitude distortion of the CSD rounded MDFT filter bank is shown in Fig.9.

The frequency response parameters of the CSD rounded FRM filter and MDFT filter bank are given in Table 3 and are compared with those of the maximum precision FRM filter and MDFT filter bank respectively. The total number of adders required for the im-

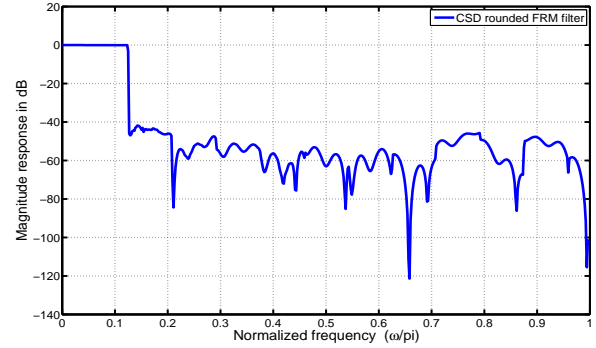


Fig. 7. Magnitude response of the continuous coefficient FRM FIR filter

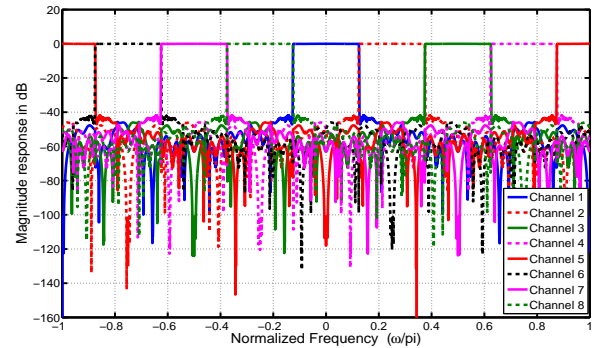


Fig. 8. Magnitude response of the continuous coefficient analysis filters of the MDFT filter bank

plementation of the CSD rounded and maximum precision MDFT filter bank are also compared in Table 3. It is observed from Table 3 that there is a significant reduction in the number of adders when the MDFT filter bank is CSD rounded compared to those of the maximum precision filter bank. But the stop-band attenuation and pass-band ripple of the FRM filter and the amplitude distortion of the MDFT filter bank are degraded when the filter coefficients are CSD rounded compared to that of the maximum precision and continuous coefficient FRM filter and MDFT filter bank respectively.

### 6.3 Proposed design of the optimal totally multiplier-less MDFT filter banks using metaheuristic algorithms

To design the optimal multiplier-less MDFT filter bank, the filter coefficients have to be suitably encoded in the CSD space and an appropriate objective function need to be formulated first. All the

Table 3. Frequency performance and complexity of the CSD rounded FRM filter and MDFT filter bank

SPT terms used	Prototype filter		MDFT filter bank	Complexity of implementation		
	Pass-band ripple (dB)	Stop-band attenuation (dB)		Number of adders	Number of adders due to SPT terms	Total number of adders
Maximum precision (7 SPTs)	0.01103	59.81	0.02359	195	344	539
CSD rounded (3 SPTs)	0.07075	41.81	0.1408	195	287	482

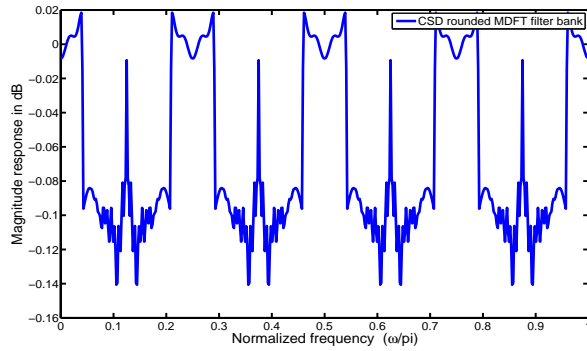


Fig. 9. Amplitude distortion of the continuous coefficient MDFT filter bank

filters in the MDFT filter bank are derived from the same prototype filter. Hence, to optimize the MDFT filter bank, it is enough to modify the coefficients of the prototype filter. Then, the meta-heuristic algorithms, such as DE, ABC, HSA and GSA algorithms are used to improve the performance degradation due to CSD rounding of the filter coefficients. All the algorithms are terminated when a specified number of iterations is reached. The design of the optimal multiplier-less MDFT filter bank using various metaheuristic algorithms is analysed and the performances are compared.

**6.3.1 Encoding of the optimization variables.** The conversion of the filter coefficients into the CSD space is done by encoding them using suitable encoding techniques. One possible way is to use the ternary encoding of the CSD equivalents of the filter coefficients [9], in which CSD representation of the closest counterpart of each continuous coefficient of the filter is taken and they are concatenated to form the initial solution of the optimization problem. Since, in this method, CSD representation is used for generating the initial solution, the dimension of the optimization problem and hence, the time taken for optimization will be large. Also, during the various steps in the optimization, the ternary coded coefficients may become invalid CSD numbers. So suitable restoration algorithms have to be used to convert them back to valid CSD format [10]. This increases the computational load of the algorithm. Hence in this thesis, the filter coefficients are encoded such that the use of restoration algorithms is avoided. To this end, a CSD look-up-table consisting of four fields, namely, index, CSD representation, decimal equivalent and number of SPT terms is created. The encoding method where the CSD filter coefficients are encoded as signed integers by taking the indices corresponding to the magnitude of the CSD filter coefficients from the look-up-table and attaching the signs of the CSD filter coefficients to them is reported in [12]. Hence it is appropriate to encode the filter coefficients using

the signed indices of the corresponding look-up-table locations. If the decimal filter coefficient is negative, then it is encoded as the negative of the index of the location of its positive counterpart [12]. The encoding and decoding of this method is simple and so, this encoding scheme is used in this paper.

Here, converting the filter coefficients into the CSD space is done using a 14 bit CSD look-up-table (LUT) consisting of four fields, namely, index, 14 bit CSD representation, decimal equivalent and number of SPT terms is created. Since, in this example, all the filter coefficients are fractions, all the 14 bits are used for the fractional part. Also, since, the magnitudes of all the filter coefficients are found to be less than 0.5, the size of the LUT is fixed to 9000 so that LUT can cover all the filter coefficient values. The entries of a typical 14 bit CSD look-up-table, in which all 14 bits are used for the fractional part, is shown in Table 4.

Table 4. 14 bit CSD look-up-table entries

Index	CSD Representation	Decimal Equivalent	Number of SPT Terms
6196	10-1000010-1010-1	0.3781	6

Thus, after the continuous filter coefficients are obtained, they are directly rounded to the nearest discrete value, whose number of SPT terms is equal to the allocated number of SPT terms as per the CSD look-up table. The corresponding index after attaching the sign is taken.

In this work, the joint optimization of the sub-filters of the FRM filter  $H_a(z)$ ,  $H_{Ma}(z)$  and  $H_{Mc}(z)$  is done i.e., the coefficients of the sub-filters are concatenated together to form the design vector of the optimization problem. Since, all the sub-filters are designed to have linear phase, only half the number of coefficients of each filter is required to be considered and concatenated to form the initial seed. This reduces the dimension of the optimization problem. Also, since we use the joint optimization of the sub-filters, all of them will be simultaneously adjusted to get a better FRM filter response and hence a better MDFT filter bank response.

Since, here, the search space consists of integers, the classical gradient based optimization techniques cannot be deployed. Therefore, the optimization based on meta-heuristic approaches are adopted here which can be properly modified to suit the problem posed [22]. For this, a suitable objective function need to be formulated which is detailed in the next section.

**6.3.2 Formulation of the objective function for the synthesis of the multiplier-less MDFT filter bank.** The performance degradation of the FRM filter and MDFT filter bank due to CSD rounding can be reduced by employing metaheuristic algorithms. To this end, an



objective function which takes care of the given design parameters is to be formulated.

Let,  $F_p$  and  $F_s$  be the pass band error and stop band error of the FRM low pass filter with pass-band and stop-band cut-off frequencies  $\omega_p$  and  $\omega_s$  respectively and they are defined as

$$F_p = \max_{0 < \omega < \omega_p} ||H(\omega) - 1| \quad (15)$$

$$F_s = \max_{\omega_s < \omega < \pi} |H(\omega)| \quad (16)$$

The pass-band error and the stop-band error are the difference between the zero phase frequency response of the optimized filter,  $H(\omega)$  and the response of the ideal filter in the pass-band and the stop-band respectively.

The amplitude distortion function of the MDFT filter bank in the  $z$ -domain is given in Eq.6. Let, it be represented in the frequency domain as  $T_{dist}(\omega)$ . To have perfect reconstruction property for a filter bank, the magnitude of the amplitude transfer function has to be unity. So the peak error in the amplitude distortion function of the filter bank,  $F_{dist}$  can be written as

$$F_{dist} = \max_{0 < \omega < \pi} |(T_{dist}(\omega) - 1)| \quad (17)$$

The total number of adders required to implement a CSD represented filter is the sum of adders due to SPT terms and structural adders. The number of structural adders is fixed, which is 195 in this case as shown in Table 3. The number of adders due to SPT terms can be reduced by reducing the total number of SPT terms required to represent the filter coefficients in the CSD space instead of a fixed number of SPT terms for each coefficient [23]. For this, a constraint is added to the objective function using the penalty method to reduce the average number of SPT terms in the CSD represented filter [24]. If  $v_H$  denotes the average number of non-zero SPT terms in the filter coefficients and  $V_H$  is the required upper bound of  $v_H$ , the penalty function added to the optimization problem is given as

$$p(v_H, V_H) = \max(0, v_H - V_H) \quad (18)$$

where,  $V_H$  is taken as 2.5 in this case, which is found to limit the total number of adders to an acceptable value which is less than the total number of adders required for implementing maximum precision filter.

Thus, the optimization problem for the multiplier-less MDFT filter bank coefficient synthesis, is modeled as minimization of the objective function  $OF$  which is formulated by including  $F_p$ ,  $F_s$ ,  $F_{dist}$  and the penalty function,  $p(v_H, V_H)$  as given below:

$$\text{Minimize } OF = \alpha_1 F_p + \alpha_2 F_s + \alpha_3 F_{dist} + \alpha_4 p(v_H, V_H) \quad (19)$$

where,  $\alpha_1$ ,  $\alpha_3$  are taken as 1 and  $\alpha_4$  is taken as 0.1. Since in the MDFT filter banks, non-adjacent aliasing components are canceled by designing filters with high stop-band attenuation,  $\alpha_2$  is given a high value of 5.

**6.3.3 Design of a totally multiplier-less MDFT filter bank using DE algorithm.** As explained in Section 6.3.1, the coefficients of the sub-filters of the CSD rounded FRM filter are concatenated together to form the design vector of the optimization problem. The initial solution is randomly perturbed to obtain the initial population for the DE algorithm. The various steps of the modified DE algorithm as explained in [20] are carried out. When the termination criteria is satisfied, the optimized candidate solution is taken and CSD decoded to get the optimal sub-filter coefficients and the

optimal FRM filter. This optimal FRM filter is used as the prototype filter for the design of the optimal MDFT filter bank. The various parameters used for the integer coded DE optimization of the CSD represented FRM filter are shown in Table 5.

Table 5. Parameters of integer coded DE

Population size	Mutation factor	Crossover ratio	Number of generations
50	0.6	0.015	1000

Table 6. Parameters of integer coded GA

Population size	Mutation rate	Popkeep fraction	Number of elite chromosomes	Number of generations
50	0.005	0.4	5	1000

Table 7. Performance of DE optimized FRM filter and MDFT filter bank (Average of 10 simulations)

Algorithm	Pass-band ripple of FRM filter (dB)	Stop-band attenuation of FRM filter (dB)	Amplitude distortion of MDFT filter bank (dB)
Integer coded GA [12]	0.02166	50.08	0.03901
Integer coded DE	0.01728	51.60	0.03464

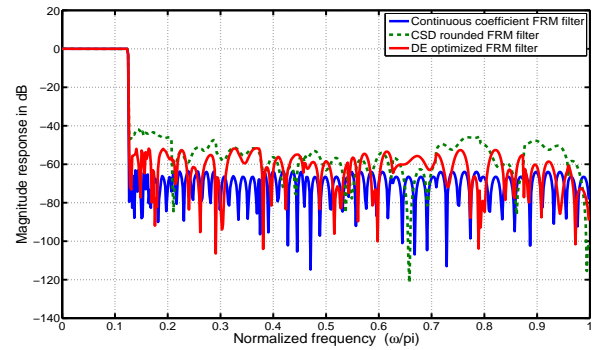


Fig. 10. Magnitude response of the FRM filter before and after using the integer coded DE

The frequency response characteristics of the FRM filter and MDFT filter bank obtained after 1000 iterations is given in Table 7. It is observed that, this algorithm takes 591.7 seconds to converge to these values. For the purpose of comparison, the performances of the DE optimized FRM filter and MDFT filter bank are compared with those of the integer coded genetic algorithm (GA) proposed in [12]. Table 6 gives the parameters employed in the above GA. The comparison of the performances of integer coded DE and GA is given in Table 7. It is found that the MDFT filter bank designed using the modified DE algorithm performs better than that using the integer coded GA [12].

The magnitude response of the integer coded DE optimized FRM filter compared with that of the continuous coefficient and CSD



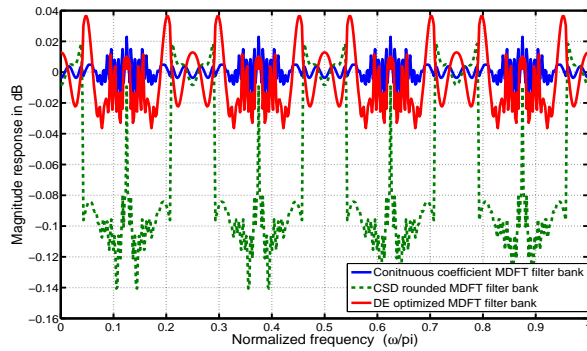


Fig. 11. Amplitude distortion of the DE optimized MDFT filter bank

rounded FRM filter is shown in Fig.10. The amplitude distortion of the DE optimized MDFT filter bank is compared with that of the continuous coefficient and CSD rounded MDFT filter bank in Fig.11.

**6.3.4 Design of a totally multiplier-less MDFT filter bank using ABC algorithm.** The design vector is CSD encoded and then it is randomly perturbed to obtain the initial food sources of the ABC algorithm. The modified ABC algorithm is explained in [20] and the steps are carried out to find the optimal filter coefficients of the sharp transition width FRM filter. This optimal FRM filter is complex modulated for obtaining optimal MDFT filter bank. The various parameters used for the integer coded ABC optimization of the CSD rounded FRM filter are shown in Table 8. ABC algorithm is observed to be very slow. But it gives very good values for the pass-band ripple of the FRM filter and amplitude distortion of the filter bank after 1000 iterations, which are almost equal to those of the maximum precision FRM filter and maximum precision MDFT filter bank respectively.

Table 8. Parameters of integer coded ABC

Number of employed bees	Limit	Number of generations
50	100	1000

Table 9. Performance comparison of the FRM filter and MDFT filter bank (Average of 10 simulations)

Algorithm	Pass-band ripple of FRM filter (dB)	Stop-band attenuation of FRM filter (dB)	Amplitude distortion of MDFT filter bank (dB)
Integer coded GA [12]	0.02166	50.08	0.03901
Integer coded ABC	0.01138	53.55	0.02298

Table 9 gives the different frequency parameters of the FRM filter and MDFT filter bank obtained after 1000 iterations. This algorithm takes 1145.98 seconds to converge to these values. The comparison with integer coded GA [12] is also shown in Table 9. It

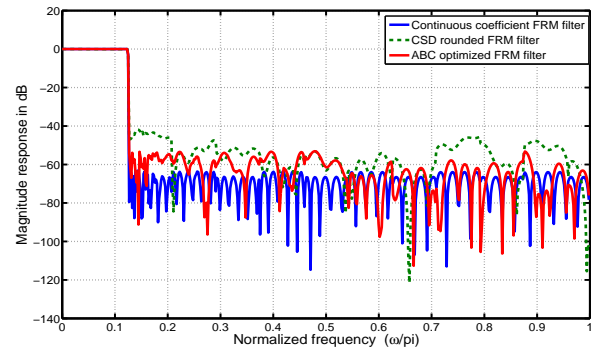


Fig. 12. Magnitude response of the FRM filter before and after using the integer coded ABC

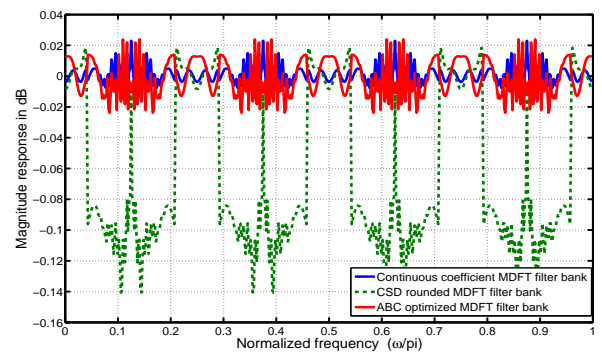


Fig. 13. Amplitude distortion of the ABC optimized MDFT filter bank

is found that the proposed optimization using modified ABC algorithm performs better in terms of the frequency domain specifications of the FRM filter and MDFT filter bank compared to the integer coded GA [12]. The magnitude response of the integer coded ABC optimized FRM filter is shown in Fig.12. The amplitude distortion of the ABC optimized MDFT filter bank is shown in Fig.13.

**6.3.5 Design of a totally multiplier-less MDFT filter bank using HSA algorithm.** The continuous coefficient FRM prototype filter is designed and the initial harmony vector of the HSA algorithm is obtained by concatenating the CSD encoded coefficients of the model filter, masking filter and complementary masking filter. The harmony vector is randomly perturbed to obtain the harmony memory for the HS algorithm. The various steps of the modified HS algorithm as explained in [20] are carried out. When the termination criteria is satisfied, the best harmony vector is taken and CSD decoded to get the optimal FRM filter and hence the MDFT filter bank. The various parameters used for the integer coded HSA optimization of the multiplier-less FRM filter are shown in Table 10.

Table 10. Parameters of integer coded HSA

Harmony memory size (HMS)	Harmony memory considering rate (HMCR)	Pitch adjusting rate (PAR)	Number of generations
50	0.95	0.01	3000

Table 11. Performance of HSA optimized FRM filter and MDFT filter bank (Average of 10 simulations)

Algorithm	Pass-band ripple of FRM filter (dB)	Stop-band attenuation of FRM filter(dB)	Amplitude distortion of MDFT filter bank (dB)
Integer coded GA [12]	0.02166	50.08	0.03901
Integer coded HSA	0.00988	55.29	0.01980

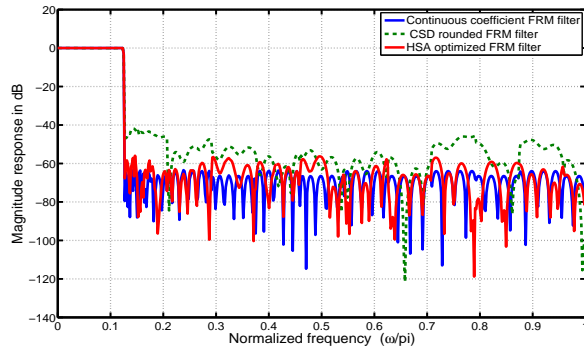


Fig. 14. Magnitude response of FRM filter before and after using the integer coded HSA

Table 11 gives the different frequency parameters of the FRM filter and MDFT filter bank obtained after 3000 iterations. This algorithm takes 738.59 seconds to converge to these values. The comparison of the performances with the integer coded GA proposed in [12] are also given in Table 11. As only one new solution is generated in each iteration of HSA, the number of iterations required is larger than that required for GA. Although, HSA takes larger number of iterations, it requires less time. It is found that the design using the modified HS algorithm outperforms that using the integer coded GA [12]. The pass-band ripple obtained for HSA optimized FRM filter is found to be better than that of the maximum precision filter. Another good result of the HSA optimized MDFT filter bank is that, its amplitude distortion is found to be even better than that of the continuous coefficient filter bank.

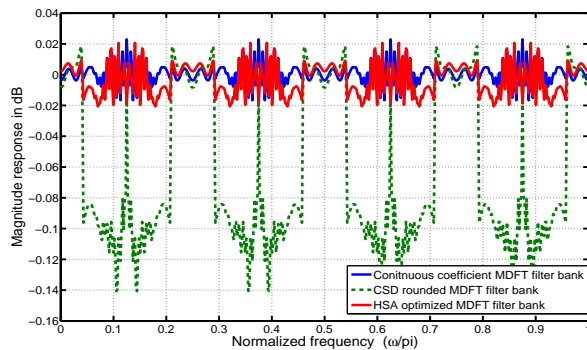


Fig. 15. Amplitude distortion of the HSA optimized MDFT filter bank

The magnitude response of the integer coded HSA optimized FRM filter is shown in Fig.14. The amplitude distortion of the HSA optimized MDFT filter bank is shown in Fig. 15.

**6.3.6 Design of a totally multiplier-less MDFT filter bank using GSA algorithm.** In the gravitational search (GS) algorithm, each candidate solution is considered as a mass. Exploration and exploitation phase are carried out using the rules of gravity and mass interaction. The integer coded GSA suitable for any discrete optimization is discussed in [20] and is used here to find out the optimal FRM prototype filter and hence the MDFT filter bank in the CSD space. After the algorithm terminates the heaviest mass which gives the optimum solution is decoded to get the optimal FRM filter which is used for deriving the optimal MDFT filter bank. The various parameters used for the integer coded GSA optimization of the CSD represented FRM filter are shown in Table 12.

Table 12. Parameters of integer coded GSA

Number of agents	Gravitational constant, $G_0$	$\alpha$	Number of generations
50	100	20	1000

Table 13. Performance of the GSA optimized FRM filter and MDFT filter bank (Average of 10 simulations)

Algorithm	Pass-band ripple of FRM filter (dB)	Stop-band attenuation of FRM filter(dB)	Amplitude distortion of MDFT filter bank (dB)
Integer coded GA [12]	0.02166	50.08	0.03901
Integer coded GSA	0.00817	58.32	0.01634

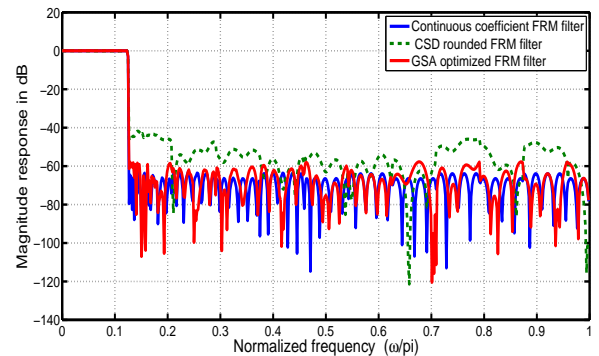


Fig. 16. Magnitude response of the FRM filter before and after using the integer coded GSA

For the purpose of comparison, this algorithm is compared with the integer coded genetic algorithm (GA) [12]. The performance in terms of the pass band ripple and stop band attenuation of the FRM filter and amplitude distortion of the MDFT filter bank are compared in Table 13. It is found that the design approach using the proposed integer coded GSA algorithm, outperforms the one using integer coded GA [12]. The GSA algorithm takes 701.51 seconds to converge to these values. It can be concluded from the table that GSA algorithm gives the highest stop-band attenuation compared

to all the other algorithms discussed here, which is a very desirable feature in the MDFT filter banks to cancel non-adjacent aliasing terms.

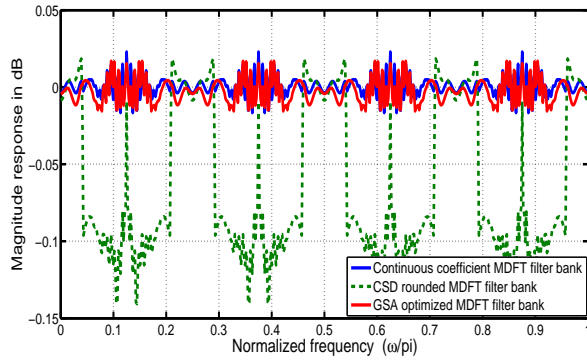


Fig. 17. Amplitude distortion of the GSA optimized MDFT filter bank

The magnitude response of the integer coded GSA optimized FRM filter is shown in Fig.16. In Fig.17, the amplitude distortion of the GSA optimized MDFT filter bank is shown.

## 7. COMPLEXITY OF IMPLEMENTATION

The main objective of this work is to reduce the complexity of implementation of the MDFT filter bank. To achieve this goal, following measures are taken.

- All the filters are implemented as FIR linear phase filters which means they are symmetric or anti symmetric and this property helps to reduce the number of multipliers needed for implementation, by a factor of two. This also reduces the dimension of the optimization variable in the optimization problem.
- Since, all the analysis and synthesis filters are derived from the same prototype filter, the design complexity of an MDFT filter bank is reduced to just the design complexity of the prototype filter.
- The prototype filter is implemented using FRM approach which reduces the complexity and hence the number of multipliers in the implementation of the MDFT filter bank especially when sharp transition width channels are required.
- The more complex multipliers are replaced by less complex shifters and adders by representing the prototype filter coefficients in the SPT space resulting in multiplier-less implementation of the filter bank. Now, the hardware implementation of the filter bank only requires shifters and adders.
- CSD representation is a special case of the SPT space, in which minimum number of adders and shifters are used to represent a decimal number. So the CSD system is chosen for the coefficient representation.
- The total number of adders and shifters in the multiplier-less MDFT filter bank can be reduced by restricting the total number of SPT terms in the CSD representation of the coefficients of the prototype filter.

In Table 14, the complexity of implementation of the MDFT filter bank using various optimization techniques is compared. All the metaheuristic algorithms employed here, lead to totally multiplier-less MDFT filter banks. All the optimization algorithms reduce the

complexity of the MDFT filter bank compared to that of the maximum precision filter bank. Thus, all the algorithms reduce the number of multipliers by 100% when compared to the continuous coefficient implementation. Also, it is found that a saving of 5.75%, 8.16%, 8.16%, and 8.16% in the number of adders when compared to the maximum precision multiplier-less filter bank, can be claimed when modified DE, ABC, HSA and GSA respectively are used. ABC, HSA and GSA result in the MDFT filter bank with the least number of adders and hence the least complexity.

## 8. PERFORMANCE EVALUATION

The performances of the various algorithms are compared in terms of the frequency characteristics of the prototype filter and MDFT filter bank, convergence time, complexity of implementation and average of objective function values.

The magnitude response of the continuous coefficient and CSD rounded FRM prototype filter and the filter optimized using various optimization approaches are shown in Fig.18. The performance in the frequency domain for the design approaches using various optimization algorithms are compared in Table 15 in terms of the pass band and stop band attenuation of the FRM filter and the amplitude distortion of the MDFT filter bank.

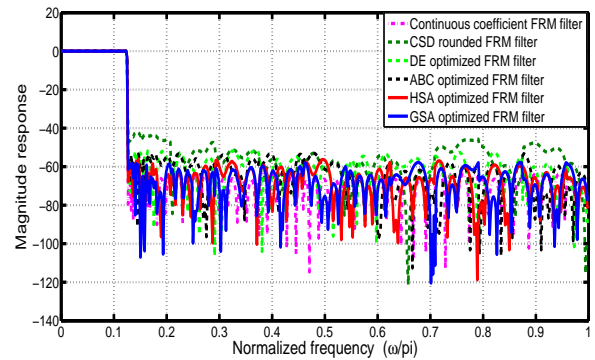


Fig. 18. Magnitude Response of the multiplier-less FRM FIR prototype filter after using the optimization techniques

It can be seen from the table that GSA gives the minimum pass-band ripple for the multiplier-less FRM filter. It also gives the maximum stop-band attenuation for the FRM filter which is very important for the cancellation of the non-adjacent aliasing components in the MDFT filter bank. The amplitude distortion of the GSA optimized MDFT filter bank is the least compared to all the algorithms discussed here. The pass-band ripple of the FRM filter and amplitude distortion of the MDFT filter bank designed using GSA algorithm are even better than those of the continuous coefficient MDFT filter bank.

HSA optimized FRM filter has the second least pass-band ripple which is better than that of the maximum precision filter and is close to the continuous coefficient filter. The stop-band attenuation of the FRM filter obtained using HSA is second highest of all the algorithms. Also, the amplitude distortion of the HSA optimized MDFT filter bank is better than that of the continuous coefficient MDFT filter bank. The optimization approaches using DE, ABC, HSA and GSA offer better performances than those using integer coded GA [12].

Table 14. Complexity comparison of the MDFT filter bank using various optimization algorithms

Optimization Techniques	Number of SPT terms	Number of adders due to SPT terms	Number of adders	Total number of adders	Number of multipliers
Continuous coefficients [5]	0	0	195	195	198
Maximum precision (7 SPTs)	541	344	195	539	0
CSD rounded (3 SPTs)	484	287	195	482	0
Integer coded GA	523	326	195	521	0
Integer coded DE	498	313	195	508	0
Integer coded ABC	497	300	195	495	0
Integer coded HSA	497	300	195	495	0
Integer coded GSA	497	300	195	495	0

Table 15. Performance comparison of the FRM Filter and MDFT Filter Bank (Average of 10 simulations)

Performance parameters	Maximum pass-band ripple of the FRM filter (dB)	Minimum stop-band attenuation of the FRM filter (dB)	Peak amplitude distortion of MDFT filter bank (dB)
Continuous coefficients [5]	$8.31 \times 10^{-3}$	62.92	$2.296 \times 10^{-2}$
Maximum precision (7 SPTs)	$1.103 \times 10^{-2}$	59.81	$2.359 \times 10^{-2}$
CSD rounded (3 SPTs)	$7.075 \times 10^{-2}$	41.81	$1.408 \times 10^{-1}$
Integer coded GA	$2.166 \times 10^{-2}$	50.08	$3.901 \times 10^{-2}$
Integer coded DE	$1.728 \times 10^{-2}$	51.60	$3.464 \times 10^{-2}$
Integer coded ABC	$1.138 \times 10^{-2}$	53.55	$2.298 \times 10^{-2}$
Integer coded HSA	$9.88 \times 10^{-3}$	55.29	$1.98 \times 10^{-2}$
Integer coded GSA	$8.17 \times 10^{-3}$	58.32	$1.63 \times 10^{-2}$

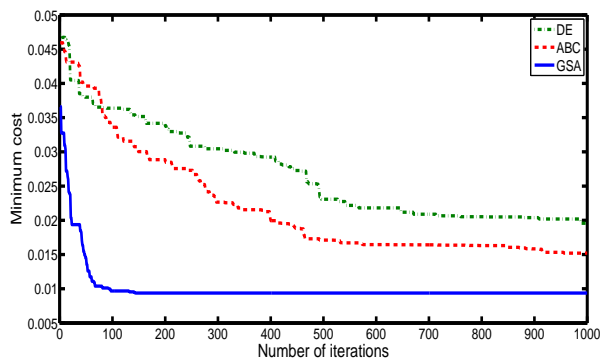


Fig. 19. Comparison of evolution processes of the various optimization techniques

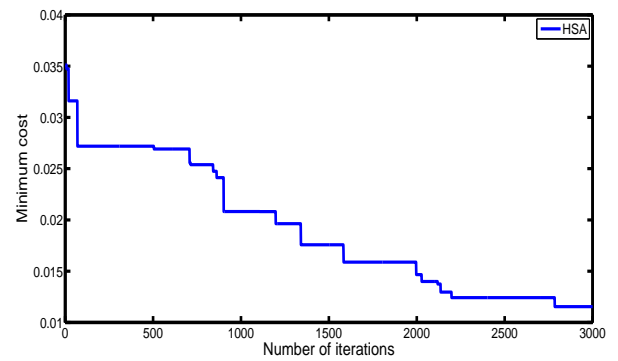


Fig. 20. Comparison of evolution processes of the various optimization techniques

The average objective function value at convergence is taken as the measure to evaluate the performance of the various optimization approaches for the design of the optimal MDFT filter bank. The comparison of the objective function values is given in Table 16. It is clear that GSA gives the minimum cost for the objective function.

One of the objectives of the design of multiplier-less filter bank is to reduce the computation time of the design process. The total run-time for the optimization is used as one of the measures for evaluating the performance of the optimization algorithm. The dimen-

sion of the candidate solution has a significant impact on the run time. The run time of the optimization for the various algorithms are also tabulated in Table 16. The evolution of various algorithms is compared in Figs. 19 and 20. It can be seen that the DE algorithm reaches its minimum objective function value in the shortest time.

A performance matrix for the MDFT filter bank is shown in Table 17. Here, the various performance parameters such as pass-band ripple and stop-band attenuation of the prototype filter, implementation complexity and amplitude distortion of the MDFT filter bank

Table 17. Performance matrix of MDFT filter bank for various integer coded optimization algorithms

Algorithm	Pass-band ripple of FRM filter	Stop-band Attenuation of FRM filter	Amplitude distortion of MDFT filter bank	Complexity	Run-time	Total score
Integer coded GA	5	5	5	3	4	22
Integer coded DE	4	4	4	2	1	15
Integer coded ABC	3	3	3	1	4	14
Integer coded HSA	2	2	2	1	3	10
Integer coded GSA	1	1	1	1	2	6

Table 16. Comparison of the average of the final objective function values obtained and average time for the design of the MDFT filter bank using various optimization techniques (Average of 10 simulations)

Algorithm	Average of final objective function values	Number of iterations	Design time (s)
Integer coded GA	$3.524 \times 10^{-2}$	1000	1140.82
Integer coded DE	$1.942 \times 10^{-2}$	1000	591.70
Integer coded ABC	$1.489 \times 10^{-2}$	1000	1145.98
Integer coded HSA	$1.223 \times 10^{-2}$	3000	738.59
Integer coded GSA	$8.92 \times 10^{-3}$	1000	701.51

and the run time of the algorithm are given scores; a minimum score is given to the best performed optimization technique and a maximum score to the worst performed optimization technique. From the table, it can be seen that GSA having the least total score of 6 which indicates that this algorithm will give the best performed MDFT filter bank. In terms of the complexity, all the algorithms result in multiplier-less MDFT filter bank. For the set of specifications used for the design of the FRM prototype filter,

- GSA gives the best pass-band ripple for the FRM filter and the best amplitude distortion for the MDFT filter bank, which are even better than those of the continuous coefficient FRM filter and MDFT filter bank.
- GSA also gives the best stop-band attenuation for the FRM filter.
- The complexity of the ABC, HSA and GSA optimized MDFT filter bank are the least.
- GSA gives the least complex MDFT filter bank with very good frequency characteristics for the FRM filter and MDFT filter bank. But this algorithm takes more run-time compared to the DE algorithm.
- DE is the fastest algorithm. But, the complexity and frequency characteristics of the DE optimized FRM filter and filter bank are the worse than those designed by ABC, HSA and GSA algorithms.

As a conclusion it can be stated that, the design using GSA results in a multiplier-less MDFT filter bank with the least complexity, and good frequency response characteristics for the FRM filter and MDFT filter bank. For applications which require filter bank with less complexity and less amplitude distortion, HSA or GSA can be chosen. If the alias cancellation and speed are of primary importance, GSA has to be selected. So, depending on the given specifications, the appropriate algorithm can be chosen.

## 9. CONCLUSION

The FRM filter design technique, when applied to the design of the sharp MDFT filter bank, leads to a reduction in the number of filter coefficients as compared to the corresponding conventional design of the MDFT filter banks. Design of FRM based MDFT filter banks in the canonic signed digit space is proposed in this paper which leads to a totally multiplier-less filter bank. Various meta-heuristic algorithms are modified and used to improve the performance of the multiplier-less MDFT filter bank. The performances of the algorithms are compared and it is found that, depending on the specifications, we can choose one of these algorithms to obtain the best performance for the multiplier-less MDFT filter bank. Thus the proposed approach results in optimal MDFT filter banks which are multiplier-less, have linear phase, near perfect reconstruction and sharp transition width with low complexity.

## 10. REFERENCES

- [1] Fliege, N. J. (1994). Multirate digital signal processing: multirate systems, filter banks, wavelets. John Wiley and Sons, Inc.
- [2] Fliege, N. J. (1994, April). Modified DFT polyphase SBC filter banks with almost perfect reconstruction. In Acoustics, Speech, and Signal Processing, 1994. ICASSP-94., 1994 IEEE International Conference on (Vol. 3, pp. III-149). IEEE.
- [3] Karp, T. and Fliege, N. J. (1999). Modified DFT filter banks with perfect reconstruction. Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, 46(11), 1404-1414.
- [4] Lim, Y. (1986). Frequency-response masking approach for the synthesis of sharp linear phase digital filters. Circuits and Systems, IEEE Transactions on, 33(4), 357-364.
- [5] Li, N. and Nowrouzian, B. (2006, May). Application of frequency-response masking technique to the design of a novel modified-DFT filter bank. In Circuits and Systems, 2006. ISCAS 2006. Proceedings. 2006 IEEE International Symposium on (pp. 4-pp). IEEE.
- [6] Lim, Y. C., Yang, R., Li, D., and Song, J. (1999). Signed power-of-two term allocation scheme for the design of digital filters. Circuits and Systems II: Analog and Digital Signal Processing, IEEE Transactions on, 46(5), 577-584.
- [7] Hartley, R. I. (1996). Subexpression sharing in filters using canonic signed digit multipliers. Circuits and Systems II:

- Analog and Digital Signal Processing, IEEE Transactions on, 43(10), 677-688.
- [8] Haupt, R. L., and Haupt, S. E. (2004). Practical genetic algorithms. John Wiley and Sons.
  - [9] Bindiya T. S, V. Satish Kumar and Elizabeth Elias, (2012), Design of Low power and Low Complexity Multiplier-less Reconfigurable Non-uniform Channel filter using Genetic Algorithm, Global Journal of Research in Engineering, Vol. 12, Issue. 6.
  - [10] Fuller, A. T., Nowrouzian, B., and Ashrafzadeh, F. (1998, August). Optimization of FIR digital filters over the canonical signed-digit coefficient space using genetic algorithms. In Circuits and Systems, 1998. Proceedings. 1998 Midwest Symposium on (pp. 456-459). IEEE.
  - [11] Mercier, P., Mohan-Kilambi, S., Nowrouzian, B., 2007, Optimization of FRM FIR digital filters over CSD and DBNS multiplier coefficient spaces employing a novel genetic algorithm, Computers, (7), 20-31.
  - [12] Manoj, V. J., and Elias, E. (2009). Design of multiplier-less nonuniform filter bank transmultiplexer using genetic algorithm. Signal Processing, 89(11), 2274-2285.
  - [13] Storn, R., and Price, K. (1997). Differential evolution a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization, 11(4), 341-359.
  - [14] Karaboga, D., and Basturk, B. (2008). On the performance of artificial bee colony (ABC) algorithm. Applied soft computing, 8(1), 687-697.
  - [15] Lee, K. S., and Geem, Z. W. (2005). A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. Computer methods in applied mechanics and engineering, 194(36), 3902-3933.
  - [16] Z. W. Geem, Music-Inspired Harmony Search Algorithm: Theory and Applications. New York: Springer-Verlag, ser. Studies in Computational Intelligence Series, 2009.
  - [17] Rashedi, E., Nezamabadi-Pour, H., and Saryazdi, S. (2009). GSA: a gravitational search algorithm. Information sciences, 179(13), 2232-2248.
  - [18] Manoj, V. J., and Elias, E. (2012). Artificial bee colony algorithm for the design of multiplier-less nonuniform filter bank transmultiplexer. Information Sciences, 192, 193-203.
  - [19] Manuel, M., and Elias, E. (2013). Design of frequency response masking FIR filter in the Canonic Signed Digit space using modified Artificial Bee Colony algorithm. Engineering Applications of Artificial Intelligence, 26(1), 660-668.
  - [20] Bindiya T. S. and Elias, E. (2012). , Design of Multiplier-less Reconfigurable Non-uniform Channel Filters using Metaheuristic Algorithms, International Journal of Computer Applications, Published by Foundation of Computer Science, New York, USA, 59(11):1-11
  - [21] Saramki, T., and Lim, Y. C. (2003). Use of the Remez Algorithm for Designing FRM Based FIR Filters. Circuits, Systems and Signal Processing, 22(2), 77-97.
  - [22] Yang, X. S. (2009). Harmony search as a metaheuristic algorithm. In Music-inspired harmony search algorithm (pp. 1-14). Springer Berlin Heidelberg.
  - [23] Li, D., Song, J., and Lim, Y. C. (1993, May). A polynomial-time algorithm for designing digital filters with power-of-two coefficients. In Circuits and Systems, 1993., ISCAS'93, 1993 IEEE International Symposium on (pp. 84-87). IEEE.
  - [24] Boyd, S. P., and Vandenberghe, L. (2004). Convex optimization. Cambridge university press.