

# A New Hybrid Descent Method with Application to the Optimal Design of Finite Precision FIR Filters

K.F.C. Yiu <sup>1</sup>, W.Y. Yan <sup>2</sup>, K.L. Teo <sup>3</sup> and S.Y. Low <sup>4</sup>

**Abstract:** In this paper, the problem of optimal design of discrete coefficient FIR filters is considered. A novel hybrid descent method, consisting of a simulated annealing algorithm and a gradient-based method, is proposed. The simulated annealing algorithm operates on the space of orthogonal matrices and is used to locate descent points for previously converged local minima. The gradient-based method is derived from converting the discrete problem to a continuous problem via the Stiefel manifold, where convergence can be guaranteed. To demonstrate the effectiveness of the proposed hybrid descent method, several numerical examples show that better discrete filter designs can be sought via this hybrid descent method.

Keywords: Hybrid descent method, simulating annealing, FIR filter design, finite precision.

<sup>1</sup>Department of Applied Mathematics, The Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong, PR China

<sup>2</sup>Department of Electrical and Computer Engineering, Curtin University of Technology, Perth, Australia

<sup>3</sup>Department of Mathematics and Statistics, Curtin University of Technology, Perth, Australia

<sup>4</sup>WATRI, Western Australian Telecommunications Research Institute, The University of Western Australia, Perth, Australia

## I. INTRODUCTION

Most existing FIR filter design methods are based on infinite precision arithmetic and thus lead to filters which cannot be readily implemented with microprocessors particularly in fixed point implementation. Such infinite precision filters have to be transformed into finite precision filters using techniques such as coefficient quantization and rounding before they can be implemented with hardware. There are several major problems with these finite precision filters. One problem is the likely sacrifice of filter performance because of the deviation from the infinite precision filters. Although this problem can be alleviated by the increase of the level of precision, there is a limitation to the maximum wordlength which can be used in practice. Moreover, simply by increasing the number of bits tends to increase the cost and complexity of the hardware implementation.

There are several methods proposed to use integer programming techniques to design FIR digital filters with discrete coefficients, such as [1]. However, simply using the integer space or the finite word length space where each coefficient value is represented by a finite number of bits suffers from the disadvantages that it saves only a few bits in coefficient word length when compared to the solution obtained by simple coefficient rounding. A better space has been introduced in [2], which is the space of powers-of-two. Under this space, the filters can be implemented with simple adders and shifters only, eliminating the need to use any multipliers which are generally costly to use. This approach has become very popular and many different algorithms have been proposed. These include the linear programming technique [2], [3], [4], the stochastic optimization techniques such as simulated annealing [5], [6], [7], tabu search [8], simulated evolution [9], and other heuristic methods such as the proportional relation-preserve method [10].

Because of the discrete nature of the problem, every feasible solution can be considered as a local minimum in continuous sense. Most of the methods proposed so far is to search through this vast space of local minima and try to find the best out of it. Clearly, this is a formidable task because of limited computer power. A better approach is to consider a continuous transformation of the discrete optimization problem such that solving the equivalent continuous optimization problem will give a discrete solution to the original discrete problem. In this way, a good continuous transformation will be able to by-pass a lot of poor discrete solutions automatically and many optimization techniques derived for continuous problems can be applied. The remaining problem is to search for the global minimum of the transformed continuous problem.

In this paper, this problem is addressed. The Stiefel manifold of real orthogonal matrices is applied to convert the discrete problem into an equivalent continuous problem [11]. A new hybrid descent algorithm is then proposed to search for the global solution of the continuous problem. In general, a good global optimization technique should have the capability of avoiding local minima, and the speed of convergence to approach stationary points. The hybrid descent algorithm is composed of two components: an analytic gradient-based approach which ensures rapid convergence to local minima, and a modified simulated annealing algorithm iterating on the Stiefel manifold to local descent points by-passing previously converged local minima. Since the simulated annealing algorithm is mainly used to locate a descent point from a previously converged local solution, if a solution obtained is not global, the set of possible descent points is infinite with a strictly positive measure; therefore the probability of locating a descent point is strictly positive. Coupling with a gradient-based algorithm to speed up the task of local search, the hybrid method is much more efficient computationally in finding the global minimum. Since the simulated annealing is mainly used for seeking descent points, the decrease

in the objective function after executing each simulated annealing search might be very small. But it is sufficient to by-pass previously converged local solutions and resume local neighbourhood search. Finally, to demonstrate the effectiveness of the proposed algorithm, numerical examples are presented.

## II. PROBLEM FORMULATION

Consider a given FIR filter of length  $N$  with frequency response given by

$$H_0(j\omega) = \sum_{n=0}^{N-1} h_0(n)e^{-j\omega n}. \quad (1)$$

It is assumed that this filter is obtained from some conventional design methods and the corresponding coefficients have very high precision. There are many variety of applications in the literature for FIR filters. From our experience, it can be used as a prototype filter for subband processing where the filter decomposes the signal into different bands of frequency [12], [13]. It can also be used in noise reduction such as echo cancellation [14] or signal enhancement [15], [16], [17]. Obviously, the literature is vast and we are still omitting many other important applications in many different areas.

For practical hardware implementation, we are interested in designing a finite wordlength FIR filter which approximates optimally the original filter  $H_0(j\omega)$  in a specific way. This discrete FIR filter is assumed to take the form

$$H(j\omega) = \sum_{n=0}^{N-1} h(n)e^{-j\omega n}, \quad (2)$$

where each coefficient  $h(n)$  is constrained to being a sum of at most  $m$  signed powers-of-two (SPT) elements from the set

$$P^2 = \{-2^0, -2^{-1}, \dots, -2^{-B+1}, 0, 2^{-B+1}, \dots, 2^{-1}\}. \quad (3)$$

Here,  $B$  is a positive integer determining the maximum number of bits and  $m$  is the maximum number of SPT terms allowed for each of the filter coefficients. They are usually specified to meet practical implementation constraints.

Let  $\Psi$  denote the set of all finite wordlength filters within the set of SPT elements. To pose the problem formally, we want to find a filter  $H(j\omega) \in \Psi$  which minimizes the weighted quadratic error criterion

$$J \equiv \int_{-\pi}^{\pi} |W(\omega)(H(j\omega) - H_0(j\omega))|^2 d\omega \quad (4)$$

where  $W(\omega)$  is a specified frequency weighting function. Note that for a filterbank application, a set of uniform frequency weighting function, which spans the required frequency band, can be conveniently defined. In evaluating the cost function (4), the error transfer function  $H(z) - H_0(z)$  can be shown to have the state-space representation  $(A_e, B_e, C_e, d_e)$  where

$$A_e = \begin{pmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix}, \quad B_e = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \quad (5)$$

$$C_e = (\mathbf{h} - \mathbf{h}_0)^T E_2, \quad d_e = (\mathbf{h} - \mathbf{h}_0)^T E_1 \quad (6)$$

where  $\mathbf{h} = [h(0) \cdots h(N-1)]^T$ ,  $\mathbf{h}_0 = [h_0(0) \cdots h_0(N-1)]^T$ , and the  $N \times N$  identity matrix  $I_N$  is partitioned as  $I_N = [E_1 \ E_2]$  with  $E_1$  being  $N \times 1$  and  $E_2$  being  $N \times (N-1)$ . Assume we specify the frequency weight function in terms of a state-space realization as

$$W(z) = d_w + C_w(zI - A_w)^{-1}B_w, \quad (7)$$

therefore, the weighted error transfer function  $W(z)[H(z) - H_0(z)]$  has the state-space representation  $(A_{we}, B_{we}, C_{we}, d_{we})$  where

$$A_{we} = \begin{pmatrix} A_e & B_e C_w \\ 0 & A_w \end{pmatrix}, \quad B_{we} = \begin{pmatrix} B_e d_w \\ B_w \end{pmatrix} \quad (8)$$

$$C_{we} = (C_e \ d_e C_w), \quad d_{we} = d_e d_w. \quad (9)$$

As a result, the cost criterion (4) can be expressed as

$$J = C_{we} P C_{we}^T + d_{we}^2 \quad (10)$$

where  $P$  denotes the controllability Gramian of the above realization, i.e., the unique solution to the Lyapunov equation

$$A_{we} P A_{we}^T - P + B_{we} B_{we}^T = 0. \quad (11)$$

Since

$$C_{we} = (\mathbf{h} - \mathbf{h}_0)^T \begin{pmatrix} 0 & C_w \\ I_{N-1} & 0 \end{pmatrix} \quad \text{and} \quad D_{we} = d_w (\mathbf{h} - \mathbf{h}_0)^T E_1,$$

we can deduce that (10) is equivalent to

$$J = (\mathbf{h} - \mathbf{h}_0)^T S (\mathbf{h} - \mathbf{h}_0) \quad (12)$$

with

$$S = (I_N \ \bar{C}_w) \begin{pmatrix} d_w^2 & 0 \\ 0 & P \end{pmatrix} (I_N \ \bar{C}_w)^T \quad (13)$$

where  $\bar{C}_w$  is an  $N \times N_w$  matrix with the first row being  $C_w$  and all the other rows being 0. As  $S$  is a constant matrix, the formula (12) provides a simple way to evaluate the cost function.

### III. CONTINUOUS TRANSFORMATION

In the sum of  $m$  powers-of-two formulation, each coefficient  $h(n)$  is constrained to being a sum of at most  $m$  signed powers-of-two (SPT) elements from the set  $P^m$  as defined by (3). Define  $\mathbf{v} \equiv [-2^0 - 2^{-1} \cdots - 2^{-B+1} \ 0 \ \cdots \ 0 \ 2^{-B+1} \cdots 2^{-1}]^T$  and  $\mathbf{g}$  is an arbitrarily chosen  $(2B+m-1)$ -dimensional row vector with  $m$  elements of one and zeroes otherwise. Each filter coefficient  $h(n)$  can be rewritten via a permutation matrix  $P$  as

$$h(n) = \mathbf{g} P \mathbf{v}, \quad (14)$$

where  $P$  is an unknown matrix in this combinatorial representation. Note that a permutation matrix is a square matrix of 0s and 1s in which each row and each column contains exactly one 1. Therefore, a permutation matrix is within the set of orthogonal matrix

$$St_{2B+m-1} \equiv \{U \in \mathbb{R}^{(2B+m-1) \times (2B+m-1)} | U^T U = I\} \quad (15)$$

which is the well-known Stiefel manifold of real orthogonal  $(2B + m - 1) \times (2B + m - 1)$  matrices [18]. In order to search for the correct permutation matrices from the Stiefel manifold, a few constraints are needed. The first constraint is to restrict the sign of the elements to be positive, which can be achieved by writing

$$P = U \odot U \quad U \in St_{2B+m-1} \quad (16)$$

where  $\odot$  denotes entrywise matrix multiplication. The second constraint is to enforce  $P$  being a permutation matrix. One important property (see lemma 3.1 in [11]) of a permutation matrix is the following:

$$\|\Lambda\|_2 = \|\text{diag}(U^T \Lambda U)\|_2 \quad (17)$$

where  $\Lambda$  is a diagonal matrix with distinct elements and  $\text{diag}(\cdot)$  denotes the diagonal part of a matrix. With these two constraints, a permutation matrix can be identified as a subset from the Stiefel manifold. The first constraint can be directly substituted into

$$\mathbf{h} = \begin{pmatrix} \mathbf{g}_1(U_1 \odot U_1) \\ \mathbf{g}_2(U_2 \odot U_2) \\ \vdots \\ \mathbf{g}_N(U_N \odot U_N) \end{pmatrix} \mathbf{v}, \quad (18)$$

where  $\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_N$  are  $N$  arbitrarily chosen  $(2B + m - 1)$ -dimensional row vectors with  $m$  elements of one and zeroes otherwise. The second constraint has to be added as a penalty term. Let

$$J_c(U_1, \dots, U_N) \equiv \sum_{i=1}^N (\|\Lambda\|_2^2 - \|\text{diag}(U_i^T \Lambda U_i)\|_2^2) \quad (19)$$

the final optimization over the continuous Stiefel manifold together with the second constraint can be formulated as

$$\min_{U_1, \dots, U_N \in St_{2B+m-1}} J_k(U_1, \dots, U_N) = J(U_1, \dots, U_N) + kJ_c(U_1, \dots, U_N), \quad (20)$$

where  $k$  is a penalty constant.

#### IV. GRADIENT-BASED ALGORITHM

In deriving the gradient of the function  $J_k$ , each of the unknown matrices is perturbed sequentially to form the generalized Taylor expansion as

$$J_k(U_1, \dots, U_i + \delta U_i, \dots, U_N) - J_k(U_1, \dots, U_i, \dots, U_N) = \langle \partial J_k / \partial U_i, \delta U_i \rangle + o(\delta U_i), \quad (21)$$

where  $\langle \cdot \rangle$  is a suitable inner product and  $o(\cdot)$  denotes higher order terms. For the Stiefel manifold, a suitable inner product is defined in terms of the trace of a matrix

$$\langle \xi, \eta \rangle \equiv 2\text{trace}(\xi^T \eta) \quad (22)$$

where  $\xi, \eta$  are in the tangent plane of  $St_{2B+m-1}$  [18]. By doing so,  $\nabla J_k$  can be derived for all variations of  $U$  and is given by [11]

$$\nabla J_k = (R_1 - U_1 R_1^T U_1, \dots, R_N - U_N R_N^T U_N) \quad (23)$$

where

$$R_i = 2(\mathbf{h} - \mathbf{h}_0)^T \mathbf{s}_i (U_i \odot (\mathbf{v}\mathbf{g}_i)^T) - 2k\Lambda U_i \text{diag}(U_i^T \Lambda U_i), \quad (24)$$

with  $\mathbf{s}_i$  being the  $i$ th column of the constant matrix  $S$  defined in (12). Therefore, any minimum of  $J_k(U_1, \dots, U_N)$  must obey the optimality conditions

$$R_i - U_i R_i^T U_i = 0, \quad i = 1, \dots, N. \quad (25)$$

When these conditions are not fulfilled, a gradient flow algorithm can be derived to update  $U_1, \dots, U_N$  via the ordinary differential equations

$$\dot{U}_i(t) = U_i(t) R_i^T(t) U_i(t) - R_i(t), \quad i = 1, \dots, N. \quad (26)$$

Since

$$\frac{dJ_k}{dt} = 2\text{trace}((\partial J_k / \partial U_i)^T \dot{U}^T(t)) = -2\|\dot{U}(t)\|_2^2 \leq 0, \quad (27)$$

a descent direction for  $J_k$  is clearly defined. In solving (26), the ordinary differential equations can be integrated as

$$U_i^{(n+1)} = e^{\alpha \Lambda_i^{(n)}} U_i^{(n)}, \quad i = 1, \dots, N \quad (28)$$

where  $\alpha$  is a step-size of integration and

$$\Lambda_i = U_i R_i^T - R_i U_i^T, \quad i = 1, \dots, N. \quad (29)$$

Since each  $\Lambda_i$  is skew-symmetric, the exponential of the matrix is therefore orthogonal for any scalar  $\alpha$ . Thus, the integration step (28) preserves the orthogonality of  $U$ . The final gradient-based algorithm can be summarized as:

- 1 Set  $k = 0$ , choose the integration stepsize  $\alpha$  and  $N$  initial orthogonal matrices  $(U_1^{(0)}, \dots, U_N^{(0)})$ .
- 2 Calculate (28) iteratively until convergence. Denote the final orthogonal matrices by  $(U_1(k), \dots, U_N(k))$ .
- 3 Compute  $J_k$ .
- 4 Go back to Step 2 with

$$U_i^{(0)} = U_i(k), \quad i = 1, \dots, N \quad (30)$$

and increase  $k$  if (17) is not satisfied.

The convergence of this algorithm to a stationary point has been proven in [11] for a fixed penalty constant  $k$ . As  $k$  increases, it is known that the sequences have the following property [19]:

- (i)  $J_k(U_1, \dots, U_N)$  is non-decreasing,
- (ii)  $J_c(U_1, \dots, U_N)$  is non-increasing,
- (iii)  $J(U_1, \dots, U_N)$  is non-decreasing.

As  $J$  increases with  $k$ , a lot of the converged solutions are useless because they are poorer than the simple direct rounding of the infinite precision solution. Therefore, in order to get the global minimum of the problem, the increase of  $J$  with  $k$  must be controlled properly.

## V. SIMULATED ANNEALING ALGORITHM

It is obvious that (20) is a highly nonlinear minimization problem despite the fact that the solution space is now continuous. This problem usually have numerous local minima. A gradient descent type of minimization algorithm often converges to local minima. One way of obtaining a global minimum solution to the nonlinear minimization problem is to use a global optimization algorithm, e.g., a simulated annealing algorithm. In general, the algorithm is composed of three key steps, namely the generation of the next point in the solution, using random perturbations, a choice of a probability distribution to govern the acceptance of uphill steps, and an annealing schedule.

Although the choice of the probability distribution and the annealing schedule are relatively easy, the generation of random changes is the most problematic. If the random changes cover a vast area in the solution domain, uphill moves are often generated. However, if random changes are restricted to a small area, local minima are often reached instead unless the starting point is sufficiently close to the global minimum. Therefore, this is an important step requiring much experimentation and is usually problem dependent.

In this paper, following [20] and [21], the Boltzmann probability distribution is used. The annealing schedule is determined by the parameters  $\gamma$  (the cooling speed),  $N_c$  (the number of cooling steps),  $N_p$  (the number of random perturbations for each temperature), and  $T$  (the initial temperature). Their values are chosen via numerical experiments.

There are many ways to introduce random perturbations to the solution. A simple method is to re-generate a new solution. However, this method suffers from the fact that there is little control on the magnitude of the random perturbation and frequent uphill moves might be incurred. Another way is to modify some of the unknowns only in the current solution in order to produce gradual changes. We find from experiments that it is better to generate a number of new potential solutions and select among them randomly. It is required that some of the perturbations resemble local searches, while some might deviate from the current solution significantly. Consequently, the set of perturbations applied here include:

- (i) Re-generate the orthogonal matrix.
- (ii) Re-generate two rows of the orthogonal matrix and orthogonalize the final matrix.
- (iii) Multiply a randomly selected submatrix by another random matrix and orthogonalize the final matrix.
- (iv) Apply a Givens rotation to the orthogonal matrix.
- (v) Apply a Householders reflection to the orthogonal matrix.
- (vi) The orthogonal matrix remains unchanged.

In the above perturbations, (i) and (v) create larger deviations from the current solution while (ii), (iii) and (iv) resemble local searches. (vi) is added as we need to include the possibility of the current solution is a accepted solution for some of the orthogonal matrices. Note that for the Givens rotation, the rotational angle is generated randomly. As for the Householders reflection  $P = I - \frac{2\mathbf{w}\mathbf{w}^T}{\mathbf{w}^T\mathbf{w}}$ , the vector  $\mathbf{w}$  is generated randomly. It is expected that local perturbations will help to reduce  $J$  and the global moves will decrease  $J_c$  effectively. Since there is no reason why any potential solution generated would have any preference over any other potential solutions, we choose to run each cooling step  $N_p$  times and choose randomly among the six potential solutions.

The simulated algorithm can be implemented as follows:

### **Initiation**

Fix a penalty constant  $k$ . Select  $\gamma$ ,  $N_c$ ,  $N_p$ , initial  $T$ , and  $N$  initial orthogonal matrices.

### Cooling

- (a) Let  $j$  be the cooling step. Set  $j = 1$ .
- (b) If  $j \in (1, \dots, N_c)$ 
  - (i) For each of the orthogonal matrices, calculate  $i = \text{random}\{1, 2, 3, 4, 5, 6\}$  and generate the corresponding perturbation according to the outcome of  $i$ .
  - (ii) Calculate  $D = \tilde{J}_k - J_k$ . If  $D < -\delta_k$  or  $\text{random}[0, 1] < T \exp(-D/T)$ , then update the set of orthogonal matrices.
  - (iii) Set  $j := j + 1$  and return to (i) until  $N_p$  perturbations are executed.
- (c) Set  $T := \gamma T$  and  $j := j + 1$ . Return to Step (b) until  $N_c$  cooling steps are executed.

## VI. HYBRID DESCENT ALGORITHM

Since  $J_k$  is not convex, we cannot expect to locate its global minimum using a gradient-based approach. Theoretically, the global minimum of  $J_k$  could be sought by using the simulated annealing algorithm alone. However, its convergence rate is usually very slow. On the other hand, a gradient-based algorithm is much more efficient in converging to a stationary point. Thus, by combining the simulated annealing algorithm with a gradient-based local minimization algorithm suitably, an efficient hybrid descent method can be derived which inherits the advantages of both methods. Let  $J_{DR}$  be the cost function of the solution derived from rounding the infinite precision solution directly. Since both  $J_k$  and  $J$  are increasing sequences of  $k$ , once  $J$  crosses the  $J_{DR}$ , the converged solution will be useless. It is therefore essential to keep

$$J \leq J_{DR} \quad \forall k. \quad (31)$$

Because of the global convergence nature of the simulated annealing algorithm, for any particular penalty constant  $k$ , a descent point can be identified easily so that the value of  $J_k$  is decreased and prevents  $J$  from growing too fast. The final hybrid descent algorithm can be formally stated in the following:

### A Hybrid Descent Algorithm

- 1 Set  $k = 0$ . Select  $N$  initial orthogonal matrices  $(U_1(0), \dots, U_N(0))$ .
- 2 Solve for the local minimum of  $J_k$  via the gradient-based algorithm to get  $(U_1^*(k), \dots, U_N^*(k))$ .
- 3 If  $J \geq \beta J_{DR}$ , starting from  $(U_1^*(k), \dots, U_N^*(k))$ , execute the simulating annealing algorithm to get a new set of orthogonal matrices  $(\hat{U}_1(k), \dots, \hat{U}_N(k))$ . If  $J_k(\hat{U}_1(k), \dots, \hat{U}_N(k)) - J(U_1^*(k), \dots, U_N^*(k)) \leq 0$ , return to Step 2.
- 4 Increase  $k$ . If  $J_c > \epsilon$ , return to Step 2.

## VII. NUMERICAL RESULTS

In the following, the hybrid descent algorithm is applied to design high-pass, low-pass and band-pass filters. Matlab 7 is used for the design of the infinite precision filters. Initial finite precision filters are obtained by seeking the best approximation for each filter coefficient in the set  $P^2$  defined in (3). These filters will be the baseline for further improvement. Note that there is no uniformly better solution than the direct-discretized filter as this filter is optimal with respect to a uniform weight  $W(\omega)$  [2]. Therefore, a suitable weight is imposed to perfect the performance of the filter.

In the first example, a simple FIR filter with 5 taps was used as a simple test of the algorithm. The number of bits  $B = 6$  was applied and each coefficient was a sum of two powers-of-two terms. In

the normalized frequency scale, the stopband was  $[0, 0.4]$  while the passband was  $[0.6, 1]$ .  $W(\omega)$  was fixed such that a larger weight is imposed towards the stopband. By discretizing the infinite precision filter directly, a cost value  $J = 0.2785$  was incurred. To test the hybrid descent algorithm, it was executed many times with a different set of randomly generated initial orthogonal matrices. All of them converged to the global solution with a cost value  $J = 0.085$ , similar to the result obtained in [11]. In addition, we do not need to try out many initial guesses to obtain the global minimum here. A typical convergence history is depicted in Fig. 1. Note that a very small  $\beta$  is used in the algorithm which corresponds to activate the simulated annealing algorithm each time after a local search with the gradient-based algorithm.

In the second example, a FIR filter with 13 taps was used to design a high-pass filter. The number of bits  $B = 6$  was applied and each coefficient was a sum of two powers-of-two terms. In the normalized frequency scale, the stopband was  $[0, 0.7]$  while the passband was  $[0.8, 1]$ . The direct-discretized finite precision filter is shown in Fig. 2. Clearly, the transition region has been poorly shifted and the suppression in the very low frequency is relatively poor. To correct these defects, a suitable weight  $W(\omega)$  skewed towards the stopband was used, giving an initial cost  $J = 1.3372$ . The hybrid descent algorithm was executed and the final optimized filter has a cost  $J = 1.127$ . The designed filter is shown in Fig. 2 and the convergence history is depicted in Fig. 3. Note that in this case,  $\beta = 0.2$  is used in the algorithm so that the simulated annealing algorithm is activated occasionally. Similarly, for the low-pass case, the hybrid descent algorithm was executed to reduce the cost from  $J = 1.336$  initially to  $J = 1.065$  in the final design. The result is shown in Fig. 4. For the band-pass case, the initial cost is  $J = 0.899$  which is subsequently reduced to  $J = 0.471$  by the algorithm. The result is depicted in Fig. 5.

## VIII. CONCLUSIONS

In this paper, a new hybrid descent method has been proposed for solving the discrete FIR filter design problem. The method combines the capability of simulated annealing method to by-pass local solutions, and a novel gradient flow method on the set of orthogonal matrices to gain rapid convergence for local neighbourhood improvement. Numerical results have shown that the global minimum can be sought using this hybrid descent method with a very nice monotonic convergence history. As a future extension of the present work, it is certainly of interest to look at the design of high order filters, and to investigate procedures for the weight selection in the present error criterion. Furthermore, different error criteria can also be investigated and different finite precision filters can be studied and compared.

## Acknowledgements

This research was supported by the Research Grants Council of HKSAR (PolyU 7191/06E) and the Research Committee of the Hong Kong Polytechnic University. We would like to thank the anonymous referees for their constructive comments.

## REFERENCES

- [1] D.M. Kodek, "Design of optimal finite wordlength FIR digital filters using integer programming techniques," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 28, no. 3, pp. 304–308, 1980.
- [2] Y.C. Lim and S.R. Parker, "FIR filter design over a discrete powers-of-two coefficient space," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 31, no. 3, pp. 583–591, 1983.
- [3] J.T. Kim, W.J. Oh, and Y.H. Lee, "Design of nonuniformly spaced linear-phase FIR filters using mixed integer linear programming," *IEEE Transactions on Signal Processing*, vol. 44, no. 1, pp. 123–126, 1996.

- [4] N.I. Cho and S.U. Lee, "Optimal design of finite precision FIR filters using linear programming with reduced constraints," *IEEE Transactions on Signal Processing*, vol. 46, pp. 195–199, 1998.
- [5] N. Benvenuto and M. Marchesi, "Digital filters design by simulated annealing," *IEEE Transactions on Circuits and Systems*, vol. 36, no. 3, pp. 459–460, 1989.
- [6] N. Benvenuto, L.E. Franks, and F.S. Hill, "On the design of FIR filter with powers-of-two coefficients," *IEEE Transactions on Communications*, vol. 32, no. 12, pp. 1299–1307, 1994.
- [7] H.H. Dam, S. Nordebo, K.L. Teo, and A. Cantoni, "FIR filter design over discrete coefficients and least square error," *IEE Proc. – Vision, Image and Signal Processing*, vol. 147, no. 5, pp. 543–548, 2000.
- [8] S. Traferro and A. Uncini, "Powers-of-two adaptive filters using Tabu search," *IEEE Transactions on Circuits and Systems II: Analog and Digital Signal Processing*, vol. 47, no. 6, pp. 566–569, 2000.
- [9] S.S. Rao and A. Ramasubrahmanyam, "Design of discrete coefficient FIR filters by simulated evolution," *IEEE Signal Processing Letters*, vol. 3, no. 5, pp. 137–140, 1996.
- [10] Q.F. Zhao and Y. Tadokoro, "A simple design of FIR filters with powers-of-two coefficients," *IEEE Transactions on Circuits and Systems*, vol. 35, no. 5, pp. 566–570, 1988.
- [11] W.-Y. Yan and K. L. Teo, "Optimal finite-precision approximation of FIR filters," *Signal Processing*, vol. 82, pp. 1695–1705, 2002.
- [12] K.F.C. Yiu, N. Grbic, S. Nordholm, and K.L. Teo, "Multi-criteria design of oversampled uniform dft filter banks," *IEEE Signal Processing Letters*, vol. 11, pp. 541–544, 2004.
- [13] K.F.C. Yiu, N. Grbic, S. Nordholm, and K.L. Teo, "A hybrid method for the design of oversampled uniform dft filter banks," *Signal Processing*, vol. 86, pp. 1355–1364, 2006.
- [14] J.Q. Huo, K.F.C. Yiu, S. Nordholm, and K.L. Teo, "A robust transform domain echo canceller employing a parallel filter structure," *Signal Processing*, vol. 86, pp. 3752–3760, 2006.
- [15] S. Nordholm, I. Claesson, and M. Dahl, "Adaptive microphone array employing calibration signals: an analytical evaluation," *IEEE Transactions on Speech and Audio Processing*, vol. 7, no. 3, pp. 241–252, 1999.
- [16] K.F.C. Yiu, N. Grbic, K.L. Teo, and S. Nordholm, "A new design method for broadband microphone arrays for speech input in automobiles," *IEEE Signal Processing Letters*, vol. 9, pp. 222–224, 2002.
- [17] K.F.C. Yiu, X.Q. Yang, S. Nordholm, and K.L. Teo, "Near-field broadband beamformer design via multidimensional semi-infinite linear programming techniques," *IEEE Transactions on Speech and Audio Processing*, vol. 11, pp. 725–732, 2003.
- [18] U. Helmke and J.B. Moore, *Optimization and Dynamical Systems*, Springer-Verlag, 1994.
- [19] R. Fletcher, *Practical Methods of Optimization*, Wiley, 1987.
- [20] S. Kirkpatrick, C. D. Gelatt Jr., and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
- [21] V. Cerny, "Thermodynamical approach to the traveling salesman problem: an efficient simulation algorithm," *Journal of Optimization Theory and applications*, vol. 45, pp. 41–51, 1985.

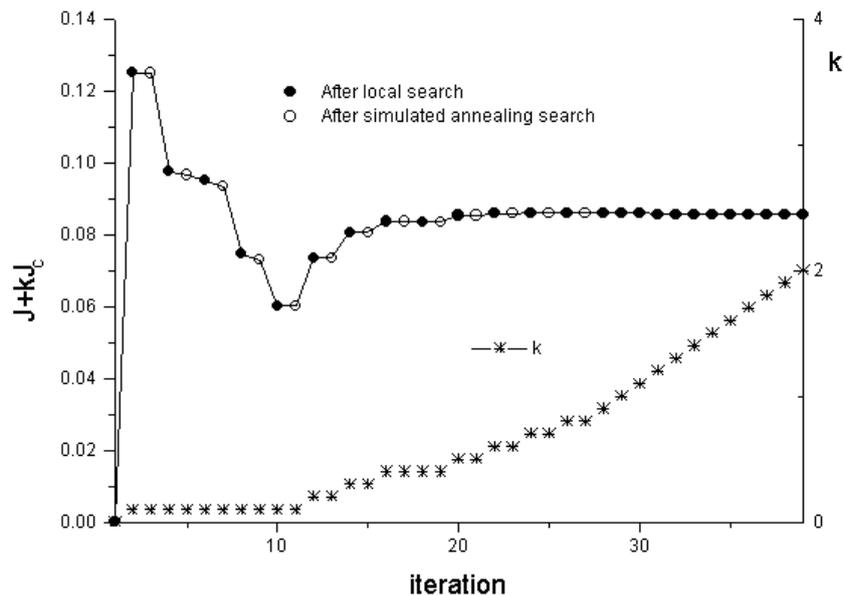


Fig. 1. The convergence of the hybrid descent algorithm for test 1 with a small  $\beta$

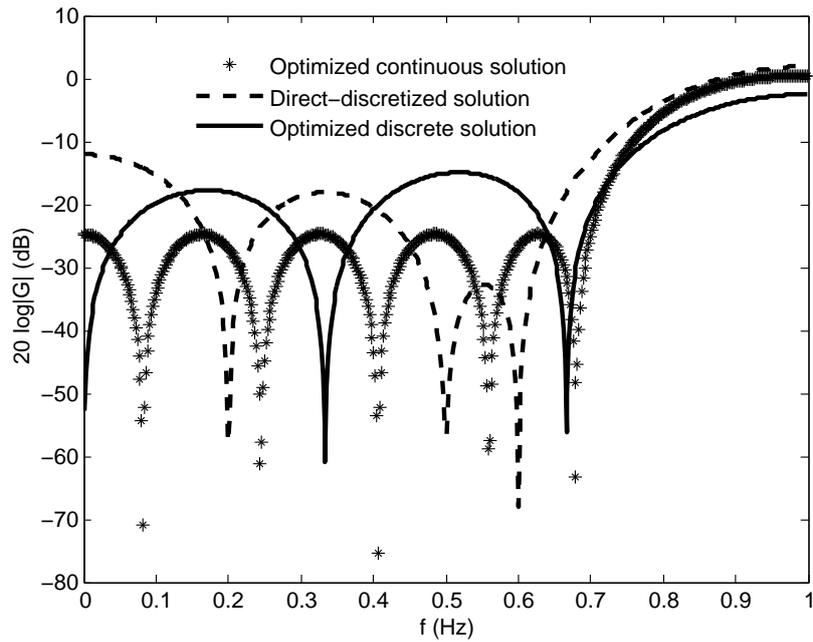


Fig. 2. The comparison of the initial and designed high-pass filters

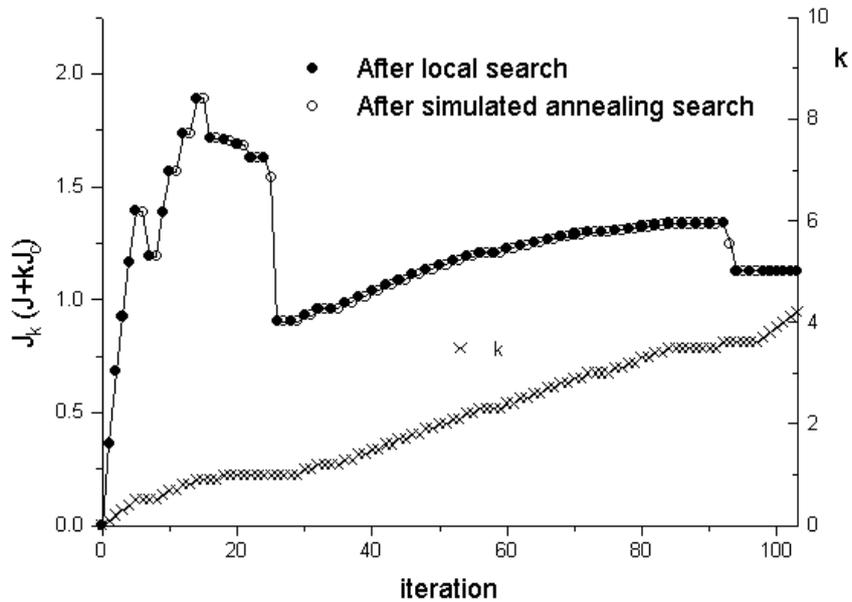


Fig. 3. The convergence of the hybrid descent algorithm for test 2 with a larger  $\beta$

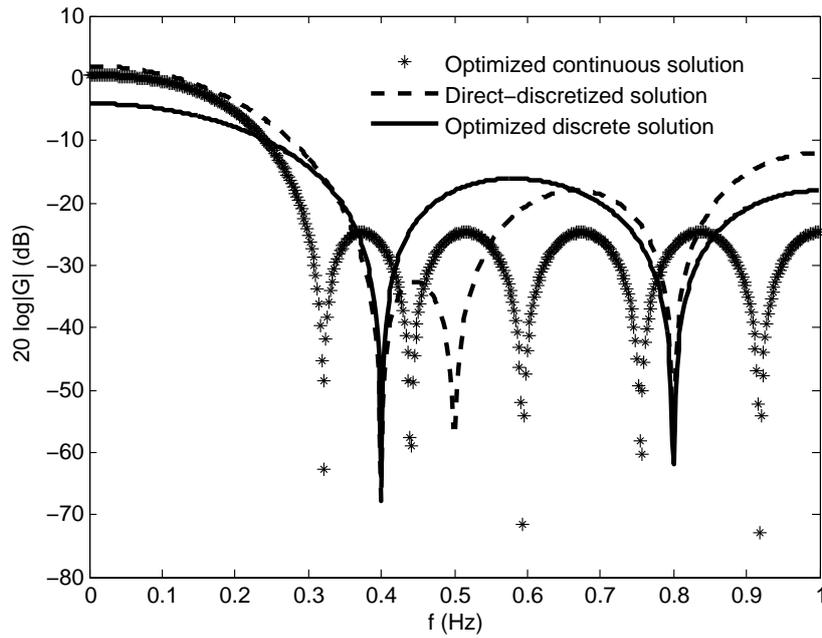


Fig. 4. The comparison of the initial and designed low-pass filters

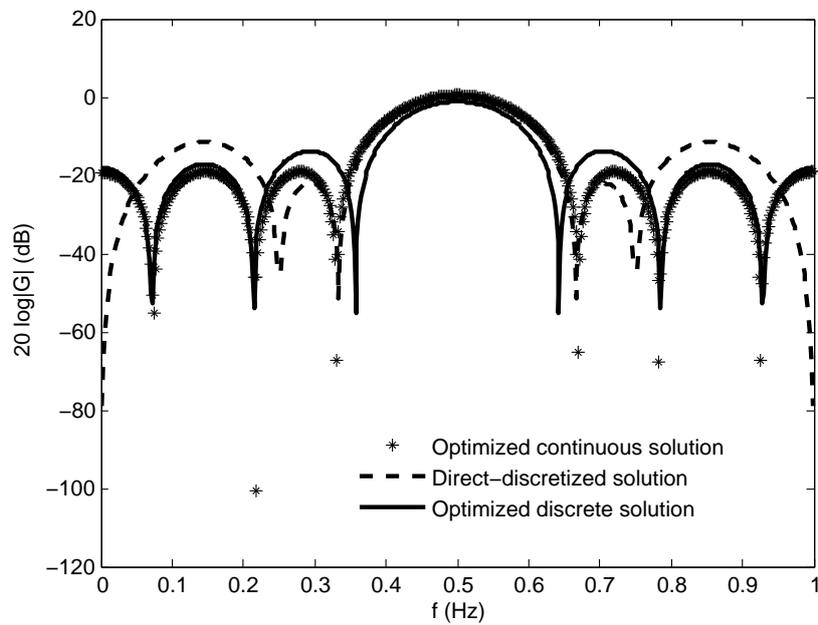


Fig. 5. The comparison of the initial and designed band-pass filters