# Fractional Delay FIR Filters Design with Enhanced Differential Evolution

Krzysztof Walczak

*Abstract*—Fractional delay FIR filters design method based on the differential evolution algorithm is presented. Differential evolution is an evolutionary algorithm for solving a global optimization problems in the continuous search space. In the proposed approach, an evolutionary algorithm is used to determine the coefficients of a fractional delay FIR filter based on the Farrow structure. Basic differential evolution is enhanced with a restricted mating technique, which improves the algorithm performance in terms of convergence speed and obtained solution. Evolutionary optimization is carried out by minimizing an objective function which is based on the amplitude response and phase delay errors. Experimental results show that the proposed algorithm leads to a reduction in the amplitude response and phase delay errors relative to those achieved with the Least-Squares method.

*Keywords*—Fractional Delay Filters, Farrow Structure, Evolutionary Computation, Differential Evolution.

## I. INTRODUCTION

**F**RACTIONAL delay digital filters are used in many applications to interpolate the new sample value of arbitrary position, which is defined as the fractional delay, between the current input sample and the previous sample. Furthermore, it is desirable that the fractional delay be tunable online without redesigning the filter coefficients. The applications of such filters include arbitrary sampling rate converters [3], speech coding and synthesis [2], symbol timing recovery in digital communication systems [4]-[7]. The area of interests is still growing, in [8] a multimode transmultiplexer structure utilizing fractional delay filter has been presented. Summary of applications as well as design methods is presented in [2].

A variable fractional delay FIR filter can be implemented by Farrow structure [1], which is based on a parallel connection of FIR subfilters. Farrow structure coefficients can be designed with the Least-Squares (LS) technique [2]. The other design methods make use of nonlinear optimization algorithms. Such approach has been presented in [11], where the coefficients are obtained from the optimal solutions to the minimax problem. However, the filter coefficients design is a difficult optimization task and hence the standard optimization methods may not provide a global optimum solution. In such cases, the solution can be found with evolutionary computation, which has been proved to be a powerful, global optimization method. Evolutionary computation comprises a group of evolutionary algorithms, such as genetic algorithm, evolution strategies, evolutionary programming. These algorithms have a common fundamental idea: all of them are based on natural evolution

Krzysztof Walczak is a PhD student at the Institute of Electronics, Silesian University of Technology, Akademicka 16, 44-100 Gliwice, Poland; email: krzysztof.walczak@yahoo.com

and during the optimization process they involve the reproduction, random variation, competition and selection. In [9] and [10], genetic algorithm has been applied to the design of variable fractional delay FIR filters. A lot of attention is also paid to the differential evolution, which is a relatively new evolutionary algorithm. It has been shown in many publications that the differential evolution outperforms other evolutionary algorithms in a wide range of applications.

In this paper, a differential evolution based optimization approach for the design of fractional delay Farrow structure filters is presented. A basic differential evolution is enhanced with the restricted mating technique, which leads to the improved algorithm performance in terms of convergence speed and the obtained solution. The presented algorithm is flexible regarding the Farrow structure subfilters length and the subfilters symmetry. The design process can result with a symmetric impulse response of FIR subfilters as well as the symmetry requirement can be relaxed. The paper is organized as follows. In Section II, the problem formulation is presented. Section III is devoted to the proposed filter design algorithm. Design example is presented in Section IV. Finally, conclusion is drawn in section V.

## II. PROBLEM FORMULATION

The Farrow structure provides a convenient and efficient way to implement a variable fractional delay digital filter which is supposed to have a tunable delay without changing the filter coefficients. Furthermore, such filter should satisfy the general requirement for constant amplitude response of unity and linear phase response in a prescribed passband, $0 \leq \omega \leq \omega_p$, where $\omega_p$ is the passband edge. The Farrow structure consists of parallel connection of $P+1$ FIR subfilters, each of length $N$, as depicted on Fig. 1. Transfer function of such structure can be written as

$$G(z, \mu) = \sum_{k=0}^{P} \mu^k H_k(z) \tag{1}$$

where

$$H_k(z) = \sum_{n=0}^{N-1} b_{kn} z^{-n} \tag{2}$$

and $\mu$ is the fractional delay control parameter.
From (1) and (2) the transfer function can be defined as

$$G(z, \mu) = \sum_{n=0}^{N-1} \left( \sum_{k=0}^{P} \mu^k b_{kn} \right) z^{-n} \tag{3}$$

World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:7, 2008

$$G(z,\mu) = \sum_{n=0}^{N-1} g_\mu(n) z^{-n} \qquad (4)$$

where

$$g_\mu(n) = \sum_{k=0}^{P} b_{kn}\mu^k \qquad (5)$$

A Farrow structure coefficients $b_{kn}$ can be a result of optimization task, e.g, evolutionary algorithm. The aim of such optimization is that the frequency response of the filter, $G(e^{j\omega},\mu)$, satisfies with a given precision the desired frequency response

$$G_{des}(e^{j\omega},\mu) = e^{-j\omega(D+\mu)}, \text{ for } 0 \le \omega \le \omega_p \qquad (6)$$

and, thus, the desired phase response

$$\Phi_{des}(\omega,\mu) = -\omega(D+\mu), \text{ for } 0 \le \omega \le \omega_p \qquad (7)$$

where

$$D = \begin{cases} \frac{N-1}{2} & \text{for } N \text{ odd} \\ \frac{N}{2} & \text{for } N \text{ even} \end{cases}$$

and

$$\mu \in [-0.5, 0.5]$$

are a fixed delay and adjustable fractional delay, respectively. Based on (6) and (7), the amplitude response error $\delta_a$ and phase delay error $\delta_d$ can be defined as

$$\delta_a = max \left| 1 - |G(e^{j\omega},\mu)| \right| \qquad (8)$$

$$\delta_d = max \left| D + \mu - \frac{argG(e^{j\omega},\mu)}{\omega} \right| \qquad (9)$$

In the evolutionary optimization process, which is proposed in this paper, Farrow structure coefficients $b_{kn}$ are optimized to achieve a minimum of the objective function $F_{obj}$, which is defined as

$$F_{obj} = S_a\delta_a + S_d\delta_d \qquad (10)$$

where $S_a$ and $S_d$ are positive scaling factors which control an impact of the amplitude and delay errors on the objective function.

## III. PROPOSED ALGORITHM

### A. Basic Differential Evolution

Differential evolution (DE) is an evolutionary algorithm (EA) proposed by Storn and Price [12] for solving global optimization problems in the continuous search space. Like other EAs, DE is a population based stochastic optimizer that explores search space of potential solutions, starting at multiple, randomly chosen initial points [13]. Each population member represents a search point in the space of potential solutions to a given problem. During the simulated evolution process, genetic operations like mutation, crossover and selection are applied to the population members in order to create population for next generation. Evolution process is performed until optimal solution is found or predefined maximum number of generations is reached. DE algorithm maintains a population of pramaters vectors which represent the problem to be solved.
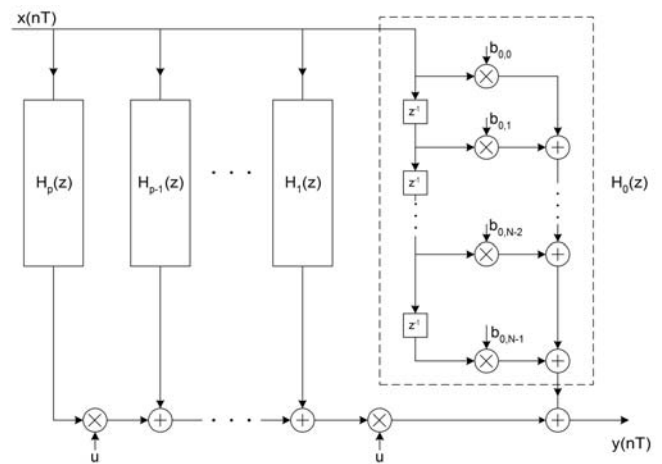


Fig. 1. Farrow structure

The current population $P_x$ consists of $N_p$ $D$-dimensional vectors:

$$P_{x,g} = (\mathbf{x}_{i,g}), i = 0,...,N_p - 1, g = 0,...,g_{max} \qquad (11)$$
$$\mathbf{x}_{i,g} = (x_{j,i,g}), j = 0,...,D \qquad (12)$$

The index $g$ indicates current generation number, index $i$ denotes vector parameters number within population and index $j$ indicates parameter number within vectors which are optimized. For each generation, DE algorithm creates an intermediate population $P_{v,g}$ which is a result of mutation operator:

$$P_{v,g} = (\mathbf{v}_{i,g}), i = 0,...,N_p - 1, g = 0,...,g_{max} \qquad (13)$$
$$\mathbf{v}_{i,g} = (v_{j,i,g}), j = 0,...,D \qquad (14)$$

Although DE algorithm follows the common concepts of evolutionary computation, there are some features which make it different. One of such unique characteristic is the way the recombination is carried out. In most of the EAs, offspring are generated from randomly chosen parents and not all of the population members participate in the reproduction process. In the DE algorithm, each population member is recombined with a mutant vector to produce a trial population $P_u$ of $N_p$ trial vectors $\mathbf{u}_{i,g}$:

$$P_{u,g} = (\mathbf{u}_{i,g}), i = 0,...,N_p - 1, g = 0,...,g_{max} \qquad (15)$$
$$\mathbf{u}_{i,g} = (u_{j,i,g}), j = 0,...,D \qquad (16)$$

Next characteristic, which distinguish DE from classical EA, is the differential mutation. This operation produces mutant vector $\mathbf{v}_{i,g}$ by adding scaled, randomly sampled, vector difference to a third vector. Basic DE mutation operator is shown by (17).

$$\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) \qquad (17)$$

The base vector index $r0$ is randomly chosen and different from target vector index $i$. The difference vector indices $r1$ and $r2$ are also randomly chosen and distinct from each other and from the base and target vector indices. According to [12], the factor $F$, which controls the amplification of the difference

World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:7, 2008

vector, should be positive, real number, less than 1. Besides the basic mutation schema (17), there are a many variants of DE which have been proposed, (18)-(20) show the most popular.

$$\mathbf{v}_{i,g} = \mathbf{x}_{best,g} + F(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) \quad (18)$$

$$\mathbf{v}_{i,g} = \mathbf{x}_{r0,g} + F(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) + F(\mathbf{x}_{r3,g} - \mathbf{x}_{r4,g}) \quad (19)$$

$$\mathbf{v}_{i,g} = \mathbf{x}_{best,g} + F(\mathbf{x}_{r1,g} - \mathbf{x}_{r2,g}) + F(\mathbf{x}_{r3,g} - \mathbf{x}_{r4,g}) \quad (20)$$

The DE algorithm offers two kinds of crossover operation, exponential and binary. In the crossover operation, trial vectors are built out of parameter values that have been copied from two different vectors. Binary crossover is implemented as follows:

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{if } rand_j(0,1) < CR \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (21)$$

The crossover probability, $CR \in [0,1]$ is user defined parameter. In the exponential crossover, an integer $n$ is randomly chosen from $[0, D-1]$ and integer $L$ from the interval $[1, D]$. Next, the offspring vector is created as follows:

$$u_{j,i,g} = \begin{cases} v_{j,i,g} & \text{for } j = <n>_D, <n+1>_D, ..., <n-L+1>_D \\ x_{j,i,g} & \text{otherwise} \end{cases} \quad (22)$$

where the angular brackets $<>_D$ denote a modulo function with modulus D. Selection process in the DE algorithm is carried out by comparing each trial vector with the target vector from which it inherits parameters. DE involves in its selection the Darwinian principle of "Survival of the fittest" [14]. If the trial vector, $\mathbf{u}_{i,g}$, has an objective function value equal or lower than its target vector $\mathbf{x}_{i,g}$, then the trial vector replaces the target in the next generation. Otherwise, the target vector remains a population member at least for one more generation.

$$\mathbf{x}_{i,g+1} = \begin{cases} \mathbf{u}_{i,g} & \text{if } f(\mathbf{u}_{i,g}) < f(\mathbf{x}_{i,g}) \\ \mathbf{x}_{i,g} & \text{otherwise} \end{cases} \quad (23)$$

The above process of mutation, crossover and selection is repeated until the optimum is located or the maximum number of generations is reached.

### B. Restricted Mating Technique

Although DE algorithm has been found to be very efficient evolutionary optimization method, for some problems it explores too many points before locating the optimum. A lot of enhancements to the basic DE algorithm have been proposed. Works [14]-[19] are good examples of modifications introduced into the DE algorithm. In this paper, the restricted mating technique is proposed as a method for improving DE performance in terms of convergence speed. The idea of restricted mating introduced into the DE algorithm is to enforce a recombination between the members with a similar objective function value. To achieve that, population is sorted with regards to the objective function value. After sorting operation, population is divided into the subpopulations and

TABLE I
POPULATION SORTING EXAMPLE.

| Population from previous generation | | Population after sorting | | |
|---|---|---|---|---|
| Member No. | Object. func. | Member No. | Object. func. | Subpop. |
| 1 | 5 | 11 | 1 | |
| 2 | 12 | 14 | 2 | |
| 3 | 14 | 5 | 3 | 1 |
| 4 | 6 | 8 | 4 | |
| 5 | 3 | 1 | 5 | |
| 6 | 9 | 4 | 6 | |
| 7 | 10 | 13 | 7 | 2 |
| 8 | 4 | 16 | 8 | |
| 9 | 15 | 6 | 9 | |
| 10 | 13 | 7 | 10 | |
| 11 | 1 | 15 | 11 | 3 |
| 12 | 16 | 2 | 12 | |
| 13 | 7 | 10 | 13 | |
| 14 | 2 | 3 | 14 | |
| 15 | 11 | 9 | 15 | 4 |
| 16 | 8 | 12 | 16 | |

individuals are allowed to mate only within a one subpopulation. Since the sorting operation is performed for each generation, information between subpopulations is exchanged each time any of individuals become a member of a different subpopulation relative to the previous generation. Table I shows the example how the subpopulations are created after population sorting, e.g, member number 10, from subpopulation 3, with an objective function value 13, becomes a member of subpopulation 4. In the next generation, this member can recombine only with members 3, 19 and 2. Proposed approach has been applied to the filter design algorithm and verified that such modification of the basic DE algorithm leads to the improved convergence speed and produces better solution. Fig. 2 shows the convergence speed of the basic DE and DE with the restricted mating technique applied to the coefficients optimization for the fractional delay FIR filter.
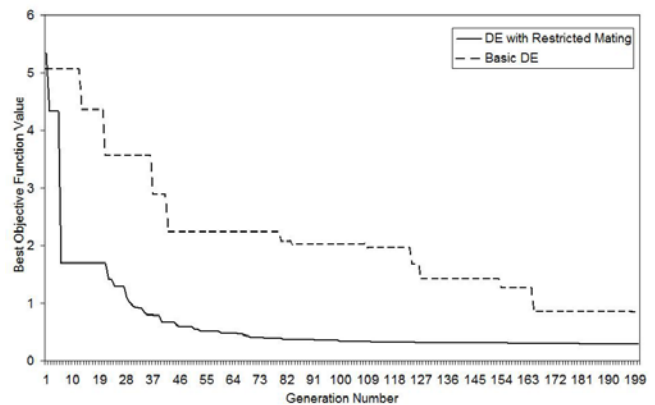


Fig. 2. Convergence Speed comparison - Basic DE vs. DE with Restricted Mating

### C. Filter Design Algorithm

This subsection describes the proposed filter design algorithm. The objective of applied evolutionary algorithm is to find Farrow structure coefficients $b_{kn}$, which results in a fractional delay filter with a minimum amplitude and phase delay

World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:7, 2008

errors, and thus, in minimum value of function defined by (10). Presented in previous sections, differential evolution enhanced with restricted mating technique was adopted to the filter design task. Farrow structure coefficients are encoded into parameters vector, which represents the population member in DE algorithm. Therefore, the population consists of potential solutions to the filter design problem. Dimension of parameters vector depends on subfilters length $N$ and number of subfilters $P$ in Farrow structure. These filter attributes are assumed to be an input parameters for the algorithm and can be defined on the basis of required amplitude and phase delay errors. Other input data, which are supposed to be provided to the algorithm are required passband and a range of tunable fractional delay. In real applications, fractional delay is expected to be tunable within some predefined discrete domain, e.g, $\mu \in \{0; 0.1; 0.2; 0.3; 0.4; 0.5\}$. Then, the approximation errors are computed for each value of $\mu$ and the errors are sampled at discrete frequency points $\omega_m$. According to (8) and (9), amplitude and phase delay errors are defined as the maximum difference between desired and obtained values. The other way the error functions can be calculated is to sum up each violation of desired amplitude and phase responses sampled at frequency points $\omega_m$. It was verified that good results can be also achieved with the error functions as defined by (24) and (25).

$$\Delta_a = \sum_l \sum_m \left| 1 - |G(e^{j\omega_m}, \mu_l)| \right| \qquad (24)$$

$$\Delta_d = \sum_l \sum_m \left| D + \mu - \frac{argG(e^{j\omega_m}, \mu_l)}{\omega_m} \right| \qquad (25)$$

where

$$m = 1, ..., M \text{ and } 0 \leq \omega_m \leq \omega_p$$
$$l = 1, ..., L \text{ and } \mu_{min} \leq \mu_l \leq \mu_{max}$$

According to the theory of evolutionary computation, each algorithm should be able to find a solution to a given problem regardless of the way the population is initialized. In most cases, population is initialized randomly. However, the initial population has an impact on the convergence speed of the algorithm. It has been found that the performance of the filter design algorithm can be increased when the population is initialized with a filters coefficients obtained from other standard design methods such as LS. In the proposed algorithm, half of the population is initialized with the vectors in the neighborhood of filters coefficients obtained from LS method. To increase diversity, the remaining part of the population is initialized randomly. Similar method of population initialization has been applied in [10].

The advantage of DE algorithm over the other EA is that it has only a few control parameters. Generally, there are only two control parameters which are specific to DE algorithm: $F$ that controls the amplification of difference vector and $CR$ which is the crossover probability. Studies performed in the area of DE control parameters show that choosing the best values is not an obvious task and can be application specific. In [20], study on DE parameter setting has been presented. Liu and Lampien [21] proposed a new version of DE, where the mutation control parameter and the crossover

control parameter are adaptive. In the algorithm presented in this paper, DE control parameters were chosen to be 0.85 and 1 for mutation control and crossover probability, respectively. Population size was set to $5 \times D$, where $D$ is the size of vector being optimized, which corresponds to the number of coefficients in the Farrow structure. The following steps summarize the proposed filter design algorithm:

1) *Define the input data for the algorithm:*
   - *required passband $\omega_p$*
   - *fractional delay values $\mu_l$*
   - *number of subfilters $P$ and subfilters length $N$*
   - *maximum number of generations $G_{max}$*
   - *acceptable amplitude response error $\epsilon_a$ and phase delay error $\epsilon_d$*

2) *Using LS method, compute the Farrow coefficients which satisfy the requirements from Step 1.*

3) *Generate an initial population, half of the population from the points in the neighborhood of coefficients. obtained from LS method, remaining part randomly*

4) *Evaluate the initial population - compute the objective function as defined by (10). Amplitude response and phase delay errors compute according to (8) and (9).*

5) *Store the best population member*

6) *Perform the population sorting with regards to the objective function value*

7) *Divide the population into four subpopulations as shown in Table I*

8) *For each of the population members perform the following genetic operations:*
   - *differential mutation as defined by (18)*
   - *exponential crossover as defined by (22)*
   - *selection mechanism as defined by (23)*

9) *Store the best population member*

10) *If current generation number $G = G_{max}$ or ($\delta_a \leq \epsilon_a$ and $\delta_d \leq \epsilon_d$) then the end of the algorithm, otherwise, go to Step 6.*

## IV. DESIGN EXAMPLE

In this example, a fractional delay FIR filter is designed using the proposed evolutionary method. The desired filter specifications are defined as follows: passband $\omega_p = 0.5$, subfilters length $N = 7$ and three subfilters in the Farrow structure. The design is carried out for the following fractional delay values: $\mu \in \{0; 0.1; 0.2; 0.3; 0.4; 0.5\}$. The results obtained show that the design method based on the DE provides a better solution compared to LS method. As can be seen in

TABLE II
DESIGN EXAMPLE RESULTS.

| $\mu$ | $\delta_d$ | | $\delta_a$ | |
|---|---|---|---|---|
| | LS | DE | LS | DE |
| 0 | $1.01 \times 10^{-5}$ | $0.19 \times 10^{-5}$ | $3.93 \times 10^{-5}$ | $2.20 \times 10^{-5}$ |
| 0.1 | $11.85 \times 10^{-5}$ | $5.89 \times 10^{-5}$ | $30.04 \times 10^{-5}$ | $29.23 \times 10^{-5}$ |
| 0.2 | $19.85 \times 10^{-5}$ | $7.69 \times 10^{-5}$ | $56.32 \times 10^{-5}$ | $39.14 \times 10^{-5}$ |
| 0.3 | $18.02 \times 10^{-5}$ | $7.13 \times 10^{-5}$ | $78.91 \times 10^{-5}$ | $61.30 \times 10^{-5}$ |
| 0.4 | $12.58 \times 10^{-5}$ | $4.74 \times 10^{-5}$ | $79.65 \times 10^{-5}$ | $56.09 \times 10^{-5}$ |
| 0.5 | $1.05 \times 10^{-5}$ | $0.11 \times 10^{-5}$ | $95.64 \times 10^{-5}$ | $82.98 \times 10^{-5}$ |

World Academy of Science, Engineering and Technology
International Journal of Electronics and Communication Engineering
Vol:2, No:7, 2008

Table II, DE based method reduces amplitude response and delay errors for each desired values of fractional delay. The amplitude response error has been reduced in the range of 13.23% to 43.86% and the phase delay error in the range of 50.33% 89.15%. The maximum values of amplitude response and delay errors are shown on Fig. 3 and Fig. 4.
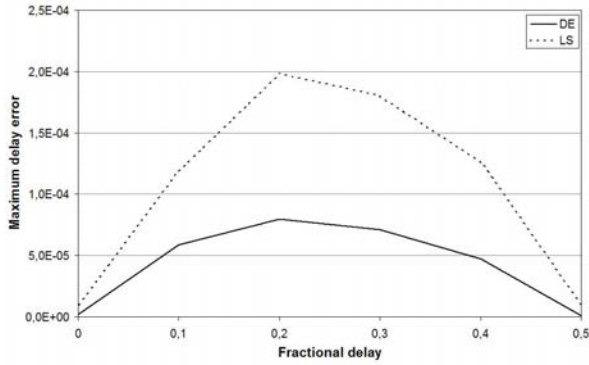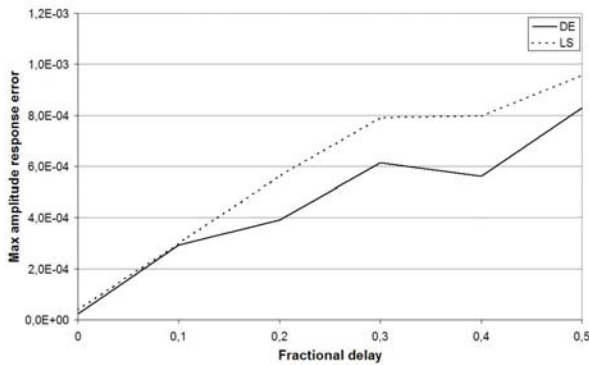


Fig. 3.    Maximum delay error



Fig. 4.    Maximum amplitude response error

## V. CONCLUSION

A Differential Evolution enhanced with the restricted mating technique for the design of fractional delay Farrow structure has been presented. Design example shows that the proposed method leads to reductions in the maximum amplitude response and phase delay errors. The proposed algorithm make use of the LS method as an initial design and then over the consecutive generations filter coefficients are optimized to obtain the desired solution. The drawback of evolutionary design method is that it requires a large amount of computation, which is common for all evolutionary algorithms. The proposed method can be used in conjunction with other novel design methods which are based on the Farrow structure.

## REFERENCES

[1]  C. W. Farrow, "A continuously variable digital delay element", *Proc. IEEE Int. Symp. Circuits Syst.*, vol. 3, pp. 2641-2645, Espoo, Finland, Jun 7-9, 1988
[2]  T. I. Laakso, V. Valimiki, M. Karjalainen, and U. K. Laine, "Splitting the unit delay", *IEEE Signal Precessing Magazine*, pp. 30-60, Jan 1996
[3]  S. C. Chan, K. M. Tsui, K. S. Yeung, and T. I. Yuk, "Design and complexity optimization of a new digital IF for software radio receivers with prescribed output accuracy", *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 2, pp. 351-366, Feb 2007
[4]  G. Watkins, "Optimal Farrow coefficients for symbol timing recovery", *IEEE Communications Letters*, vol. 5, no. 9, pp. 381-383, Seb 2001
[5]  M. T. Shiue and C. L. Wey, "Efficient Implementation of Interpolation Technique for Symbol Timing Recovery in DVB-T Transceiver Design", *IEEE International Conference on Electro. Information Technology*, pp. 427-431, 7-10 May 2006
[6]  P. Savazzi, P. Gamba, and L. Favalli, "Joint Symbol Timing Recovery and Equalization for Short Burst Transmissions", *IEEE 66th Vehicular Technology Conference*, pp. 1396-1400, Sep 30 2007-Oct 3 2007
[7]  T. G. B. Wilson, "Baud Rate Symbol Timing Synchronization for 8-VSB ATSC DTV Receivers", *IEEE International Symposium on Consumer Electronics, ISCE 2007*, pp. 1-6, 20-23 Jun 2007
[8]  A. Eghbali, H. Johansson, and P. Lowenborg "A farrow-structure-based multi-mode transmultiplexer", *IEEE International Symposium on Circuits and Systems, ISCAS 2008*, pp. 3114-3117, 18-21 May 2008
[9]  K. E. Khamei, A. Nabavi, and S. Hessabi, "Design of variable fractional delay FIR filters using genetic algorithm", *IEEE International Conference on Electronics, Circuits and Systems, ICECS 2003*, vol. 1, pp. 48-51, 14-17 Dec 2003
[10]  S. U. Ahmad and A. Antoniou, "A genetic algorithm approach for fractional delay FIR filters", *IEEE International Symposium on Circuits and Systems, ISCAS 2006*, 21-24 May 2006
[11]  H. Johansson and P. Lowenborg, "On the Design of Adjustable Fractional Delay FIR Filters", *IEEE Trans. Circuits Syst. II*, vol. 50, pp. 164-169, Apr 2005
[12]  R. Storn and K. Price, "Differential Evolution - a Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces" , *Technical Report TR-95-012*, ICSI, March 1995
[13]  K. Price, R. Storn and J. Lampien, "Differential Evolution - A Practical Approach to Global Optimization", Springer, 2005
[14]  S. Das, A. Konar and U. K. Chakraborty, "Annealed Differential Evolution", *IEEE Congress on Evolutionary Computation*, 2007
[15]  D. Davendra and G. Onwubolu, "Enhanced Differential Evolution hybrid Scatter Search for Discrete Optimization", *IEEE Congress on Evolutionary Computation*, 2007
[16]  S. Rahnamayan, H. Tizhoosh and M. Salama, "Opposition-Based Differential Evolution", *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, February 2008
[17]  O. Olorunda and A. Engelbrecht, "Differential Evolution in High-Dimensional Search Spaces", *IEEE Congress on Evolutionary Computation*, 2007
[18]  Z. Yang, K. Tang and X. Yao, "Differential Evolution in High-Dimensional Function Optimization", *IEEE Congress on Evolutionary Computation*, 2007
[19]  N. Noman and H. Iba, "Accelerating Differential Evolution Using an Adaptive Local Search", *IEEE Transactions on Evolutionary Computation*, vol. 12, no. 1, February 2008
[20]  J. Liu and J. Lampien, "On setting the control parameters of the differential evolution method", in *Proc. 8th Int. Conf. Soft Computing (MENDEL 2002)*, 2002, pp. 11-18
[21]  J. Liu and J. Lampien, "A fuzzy adaptive differential evolution algorithm" *Soft Computing - A Fusion of Fundations, Methodologies and Applications*, vol. 9, no. 6, pp. 448-462, 2005