UDC 004.02
DOI: 10.31891/CSIT-2021-3-3

RUSLAN PETROSIAN, OLEKSANDR KUZMENKO, ARSEN PETROSIAN
Zhytomyr Polytechnic State University, Zhytomyr, Ukraine

METHOD FOR CALCULATING THE FIR FILTER BASED ON GENETIC ALGORITHM

The paper considers the analysis of the genetic algorithm with the aim of its application to optimize the coefficients of FIR filters with a linear phase. FIR filters are more preferable for some tasks, because they have the following advantages: the group delay of the filter is continual; FIR filters are always stable. A genetic algorithm is a heuristic method, which is a variation of evolutionary algorithms. It solves optimization problems using natural evolution methods similar to natural selection in real world. When solving a problem with a genetic algorithm, the phenotype was identified that determines the real object. As a chromosome, which stores a list of genes (real filter coefficients), a class implemented in the Python programming language is used. As a fitness function, the standard deviation between the approximating and approximated functions of the amplitude-frequency characteristics of the FIR filter (filter with finite impulse response) is used. FIR filter design includes both optimization of filter coefficients and consideration of quantization effects, which can also be implemented using a genetic algorithm. Modeling was carried out on the example of a FIR filter of the first type using the Python programming language. The simulation results showed the effectiveness of the genetic algorithm for the synthesis of FIR filters (the results are comparable with the least squares method). This method can be successfully used in the design of FIR filters with a linear phase in the creation of technical means. It should be noted that the efficiency of the genetic algorithm depends on the generated initial population, as well as on the setting of hyperparameters, therefore, further analysis involves the study of the formation of the initial population, as well as optimization of the hyperparameters of the genetic algorithm.
Keywords: genetic algorithm; FIR filter; amplitude-frequency characteristic; optimization; standard deviation.

РУСЛАН ПЕТРОСЯН, ОЛЕКСАНДР КУЗЬМЕНКО, АРСЕН ПЕТРОСЯН
Державний університет «Житомирська політехніка»

МЕТОД РОЗРАХУНКУ КІХ-ФІЛЬТРА З ВИКОРИСТАННЯМ ГЕНЕТИЧНОГО АЛГОРИТМУ

У статті розглядається аналіз генетичного алгоритму з метою його застосування для оптимізації коефіцієнтів КІХ-фільтрів з лінійної фазою. Застосування КІХ-фільтрів більш переважно при вирішенні деяких задач, тому що мають наступні переваги: групова затримка фільтра постійна; КІХ-фільтри завжди стійкі. Генетичний алгоритм – евристичний алгоритм, що є різновидом еволюційних алгоритмів, за допомогою яких вирішуються оптимізаційні задачі з використанням методів природної еволюції, аналогічних природному відбору в природі. При вирішенні задачі генетичним алгоритмом виділено фенотип, який визначає реальний об'єкт. Як хромосоми використовується клас, реалізований на мові програмування Python, що зберігає список генів (дійсних коефіцієнтів фільтра). Як фітнес-функція використовується середньоквадратичне відхилення між апроксимованою та апроксимуючою функціями амплітудно-частотних характеристик КІХ-фільтра (фільтр з кінцевою імпульсною характеристикою). Проектування КІХ-фільтрів включає одночасно оптимізацію як коефіцієнтів фільтра, так і облік ефектів квантування, що можна виконати також з використанням генетичного алгоритму. Моделювання проводилося на прикладі КІХ-фільтра першого типу з використанням мови програмування Python. Результати моделювання показали ефективність застосування генетичного алгоритму для синтезу КІХ-фільтрів (результати ідентичні до методу найменших квадратів). Даний метод може з успіхом використовуватися при проектуванні КІХ-фільтрів з лінійною фазою при створенні технічних засобів. Слід врахувати, що ефективність генетичного алгоритму залежить від згенерованої початкової популяції, а також від настроювання гіперпараметрів, тому подальший аналіз передбачає дослідження формування початкової популяції, а також оптимізацію гіперпараметрів генетичного алгоритму.
Ключові слова: генетичний алгоритм; КІХ-фільтр; АЧХ; оптимізація; середньоквадратичне відхилення.

Introduction

Today the digital processing methods in the life of modern society play a significant role and their importance continues to increase rapidly. Digital processing is used in many areas of technology: process control systems, medical diagnostic equipment, communication facilities, measuring and diagnostic technical means of power engineering, etc. [1].

Digital signal processing is used in the fields, where it is necessary to perform such tasks as filtering, compressing, recovering, controlling, measuring a signal: audio, video processing, or any signal coming from any source.

Filtering is the most common digital processing task. Its implementation is based on using digital filters: filters with a finite impulse response (FIR filters), filters with infinite impulse response (IIR filters). In many applications with digital signal processing, the use of FIR filters is preferable because it has the following advantages:

- filter group delay is continual;
- FIR filters are always stable.

Although many methods for digital filter design were already been developed in the 60s and 70s, there is still a high interest in digital filter development. Today algorithms based on intelligent data processing are widely used. And the genetic algorithm is one of them [2].

A genetic algorithm is a heuristic method, which is a variation of evolutionary algorithms. It solves optimization problems using natural evolution methods similar to natural selection in real world.

Filter design involves at the same time optimizing both filter coefficients and accounting for quantization effects, which can be done using a genetic algorithm. It should also be noted that an increase in the efficiency of filter approximation in the passband and stopband leads to an increase in the transition band and vice versa. So, when solving many problems, a compromise is required. It can be provided by using multiobjective optimization based on a genetic algorithm [2]. Another advantage of the genetic algorithm is the use of the parallelization mechanism during the search for a solution [3]. Considering the above, the synthesis of FIR filters using a genetic algorithm is an actual task.

Analysis of recent research and publications

In the scientific researches, digital filters and methods for their design have been deeply analyzed [1]. Among the most widely used classical methods for calculating FIR filters are window method, frequency sampling method, least squares method, best uniform approximation method. The first two are not optimization methods, but are simple to use. The third method requires the use of numerical methods, and, therefore, the calculation accuracy will depend on the accuracy of those methods. The fourth method allows obtaining the best results, but, as a rule, it is impossible to determine analytically the function of the best uniform approximation. One of the most effective iterative methods for determining the best uniform approximation is the Remez exchange algorithm (modified Remez algorithm).

The use of modern heuristic algorithms for data processing [2] makes possible to get some advantages: to increase the efficiency of searching an optimal solution; perform multi-criteria optimization.

In [4], the FIR filter synthesis using a genetic algorithm is considered. The work presents a diagram (fig. 1), which explains the principle of calculating the coefficients.

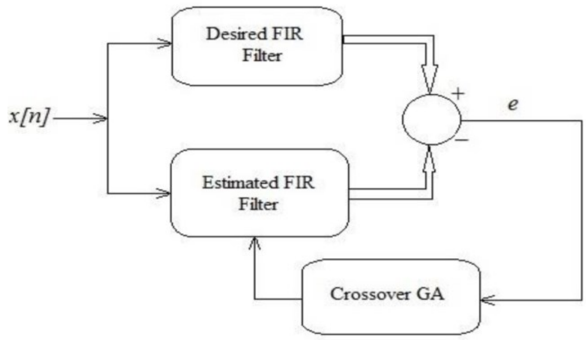

Fig.1. Diagram for estimating FIR filter coefficients using a genetic algorithm

The FIR filter belongs to the class of linear discrete systems. The relationship between input x (n) and output y (n) digital signals is determined by the following difference equation (1):

$$y(n) = \sum_{i=0}^{N-1} b_i \cdot x(n-i) \qquad (1)$$

where *y(n)*, *x(n)* - output and input digital signals respectively; b_i - coefficients of the FIR filter.

When designing a digital filter, two main synthesis methods can be distinguished: synthesis of digital filters in the frequency domain; synthesis of digital filters in the time domain.

In accordance with fig. 1 to optimize the coefficients of the FIR filter the following objective function (2) is proposed in [4]:

$$MSE = \sum_{n=0}^{T} (y(n) - \hat{y}(n))^2 \qquad (2)$$

where $y(n)$, $\hat{y}(n)$ are output digital signals of the approximated and approximating filters respectively, T is the number of signal samples.

The nature of the input signal x(n) is not clear from the work, so it is not clear in which domain the filter is synthesized: frequency or time? It can be easily shown that if the input signal *x(n)* is harmonic, then the optimization of the synthesized filter will occur in the frequency domain and only at one point of the amplitude-frequency characteristic.

The purpose of this paper is the analyze the genetic algorithm for applying it to optimizing the coefficients of FIR filters.

The research method. The structure of a digital FIR filter in accordance with expression (1) is shown in fig. 2.

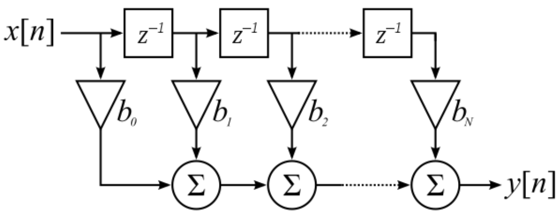

Fig.2. Direct form of the FIR filter implementation

We will synthesize the FIR filter in the frequency domain. Therefore, for synthesis, it is necessary to know the amplitude-frequency characteristic of the filter, which in general case will have the form of the expression (3) for the approximating filter:

$$\boldsymbol{H(\omega)=\sum_{n=0}^{N-1} h(n)\cdot e^{-j\omega n}} \quad (3)$$

where *h(n)* is the final impulse response of the FIR filter, *ω* is the angular frequency.

However, as already mentioned, linear phase FIR filters are widely used. This requires the impulse response to be symmetric or antisymmetric [5]. In this case, four types of FIR filters are possible (Table 1).

Table 1

Linear Phase Filter Types

Filter type	**Impulse response**	**Number of impulse response coefficients**	**Amplitude-frequency characteristic**
I	symmetrical	odd	$H(\omega)=\sum_{n=0}^{(N-1)/2} a(n)\cdot cos\,(\omega n)$
II	symmetrical	even	$H(\omega)=\sum_{n=1}^{N/2} b(n)\cdot cos\,(\omega(n-1/2))$
III	antisymmetric	odd	$H(\omega)=\sum_{n=1}^{(N-1)/2} c(n)\cdot sin\,(\omega n)$
IV	antisymmetric	even	$H(\omega)=\sum_{n=1}^{N/2} d(n)\cdot sin\,(\omega(n-1/2))$

To optimize filter coefficients, an objective function (fitness function) is required. As an objective function, we will use the standard deviation between the approximated and approximating frequency response functions of the FIR filter (4).

$$e=\sqrt{\int_0^{\pi} W(\omega)(H(\omega)-\widehat{H}(\omega))^2 d\omega} \rightarrow min \quad (4)$$

where $W(\omega)$ is the weight function.

When solving a problem with a genetic algorithm, it is necessary to identify the phenotype that determines the real object. In our case, the filter coefficients will be a phenotype, which for our task will be genes. These genes will form a chromosome (fig. 3).

a0	a1	a2	a3	. . .	aN-1

Fig.3. FIR filter chromosome structure

As an implementation of the genetic algorithm, a chromosome is a Python class that stores a list of real filter coefficients:

```
class Fir1T:
    """ Non-recursive digital filter of the first type"""

    def __init__(self, fmax, a):
        self._N = 2 * len(a) - 1   # number of FIR coefficients
        self._fmax = fmax          # sampling rate / 2 (maximum)
```

```
        self._a = a                  # list of coefficients AFC
(chromosome)

    def __del__(self):

    def getOrder(self):

    def getNumCoeff(self):

    def getGain(self, fi):

    def getCoeffFIR(self):
```

When solving optimization problems, the genetic algorithm consists of the following sequence of the steps:

1. Creation an initial population;
2. Calculation of the chromosomes fitness;
3. Selection of initial chromosomes (solutions) with the best fitness values for creating a new population;
4. Performing the operation of the chromosomal crossover;
5. Performing the mutation operation;
6. If the stop condition is triggered, we return the chromosome with the best fitness value, otherwise go to step 2 to process the new population.

We will implement modeling for the first type filter (table 1). The approximated frequency response function will have the form shown in fig. 4.

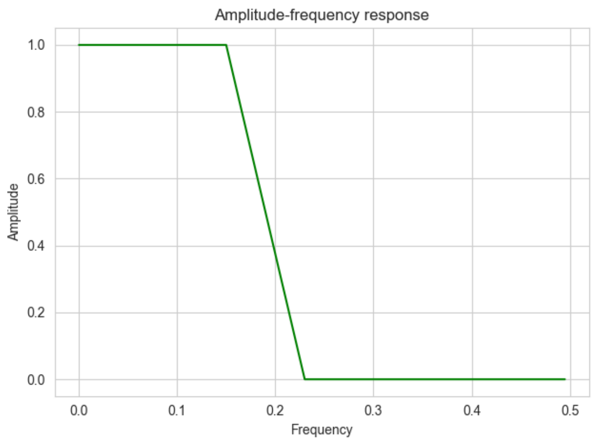

Fig.4. Frequency response of the desired FIR filter

To run the genetic algorithm, you need to setup some hyperparameters. In our case, it will look like this:

```
ORDER = 24            # the filter order
POPULATION = 150      # the number of individuals in the population
SURVIVOR = 0.25       # the survival probability
MUTATION = 0.1        # the probability of an individual mutating
GENERATIONS = 100     # the maximum number of generations
```

The simulation results are shown in the next two figures: fig. 5 – synthesized FIR filter of the 24th order; fig. 6 shows the dependence of fitness on the chromosomes population.

Calculated FIR filter coefficients using a genetic algorithm:

a0 =0.005091678516838307,
a1 =0.0028879984601902814,
a2 =-0.008090511277044445,
a3 =-0.014439652055667128,
a4 =0.00013690865912870708,
a5 =0.026388498063331992,
a6 =0.02700867633181265,
a7 = -0.01882295463803626,
a8 = -0.06886025153816047,
a9 = -0.037724044224033935,
a10 =0.10832130045736052,
a11 =0.2915719791165157,
a12 =0.37570415749595354,
a13 =0.2915719791165157,
a14 =0.10832130045736052,
a15 =-0.037724044224033935,
a16 =-0.06886025153816047,
a17 =-0.01882295463803626,
a18 =0.02700867633181265,
a19 =0.026388498063331992,
a20 =0.00013690865912870708,
a21 =-0.014439652055667128,
a22 =-0.008090511277044445,
a23 =0.0028879984601902814,
a24=0.005091678516838307.

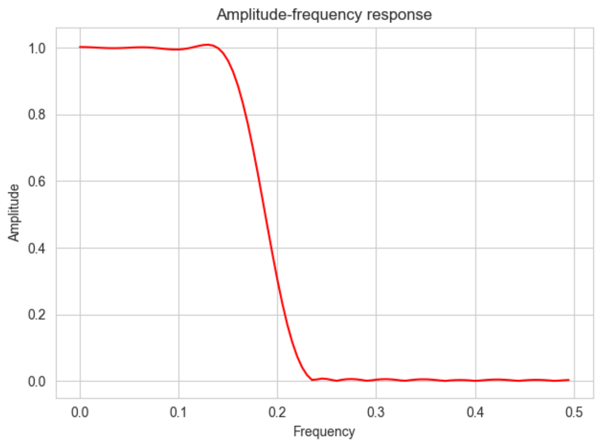

Fig.5. Frequency response of the synthesized FIR filter

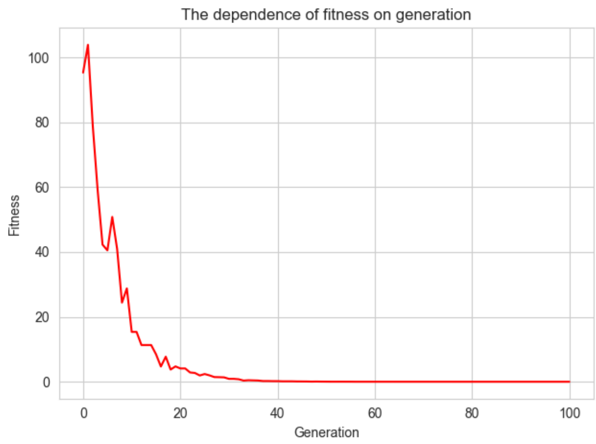

Fig.6. The fitness change when searching a solution

Conclusions and prospects for further research

The paper analyzes the genetic algorithm for the purpose of its application to optimize the coefficients of FIR filters with a linear phase

When solving a problem with a genetic algorithm, a chromosome was created, which is formed as a list of genes (real filter coefficients); a fitness function has been chosen to optimize the FIR filter coefficients.

Modeling was carried out on the example of an FIR filter of the first type using the Python programming language. The simulation results showed the effectiveness of the genetic algorithm for the synthesis of FIR filters (the results are comparable with the least squares method). This method can be successfully used in the design of FIR filters with a linear phase in the creation of technical means. It should be noted that the efficiency of the genetic algorithm depends on the generated initial population, as well as on the setting of hyperparameters. Therefore, further analysis involves the study of the initial population formation, as well as optimization of the genetic algorithm hyperparameters.

REFERENCES

1. Solonina A.I. Digital signal processing in MATLAB. SPb.: BHV-Peterburg, 2018. 560 p.
2. Mutingi M., Mbohwa C. Grouping Genetic Algorithms. Advances and Applications. Switzerland: Springer International Publishing, 2017. 243 p.
3. Knysh D.S., Kureichyk V.M. Parallel genetic algorithm with fuzzy migration operator. Artificial Intelligence. 2010. № 3. P. 73-80.
4. Thakare V.V., Sahu R.K. A Genetic Algorithm for Optimization of MSE & Ripples in Linear Phase Low Pass FIR Filter & Also Compare with Cosine Window Techniques. International Journals Digital Communication and Analog Signals. 2015. №1. P. 9–15.
5. Proakis J., Manolakis D. Digital Signal Processing : 4th Edition. London: Pearson, 2013. 1019 p.

Ruslan Petrosian **Руслан Петросян**	senior lecturer, Computer Science Department, Zhytomyr Polytechnic State University, Zhytomyr, Ukraine, e-mail: e_rvs@ukr.net orcid.org/0000-0002-0388-8821 https://scholar.google.com/citations?user=K2CKTncAAAAJ	старший викладач кафедри комп'ютерних наук, Державний університет "Житомирська Політехніка", Житомир, Україна.

Oleksandr Kuzmenko **Олександр Кузьменко**	senior lecturer, Computer Science Department, Zhytomyr Polytechnic State University, Zhytomyr, Ukraine? e-mail: kkn_kov@ztu.edu.ua orcid.org/0000-0002-4937-3284, Scopus Author ID: 57216845002 https://scholar.google.com/citations?user=6WRuJ0gAAAAJ	
Arsen Petrosian **Арсен Петросян**	master's degree student, Zhytomyr Polytechnic State University, Zhytomyr, Ukraine https://orcid.org/0000-0003-0960-8461 e-mail: xckaytz@gmail.com https://scholar.google.com.ua/citations?user=ySfCqTYAAAAJ	студент першого курсу магістратури, Державний університет "Житомирська Політехніка", Житомир, Україна.