Automatic FIR Filter Design Method and Tool based on Genetic Algorithms

André M. Barros, Heitor S. Lopes and Álvaro L. Stelle
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial, CPGEI.
andre_barros@superig.com.br, hslopes@cpgei.cefetpr.br, stelle@cpgei.cefetpr.br
Universidade Tecnológica Federal do Paraná, UTFPR.
Av. Sete de Setembro, 3165. Curitiba-PR. Brazil. 82.230-901. Phone: 55(41)-3310-4679.

***Abstract*—A FIR Filter design method and its respective tool using genetic algorithms were developed. The main feature of this method is to offer a transparent mode for the user who doesn't know evolutionary computation, as well as its parameters. The user inputs the filter specifications and gets a sub-optimal result in an average number of four attempts. The sub-optimal criterion was based on the Rabiner, Parks and McClellan algorithm and the implemented software was built using the GALOPPS tool.**

***Index Terms*—FIR filter design, genetic algoritms, Galopps, Rabiner Parks and McClellan algorithm, Remez.**

I. Introduction

The design of FIR digital filters using techniques of DSP is an automatic procedure. There are computer programs like MATLAB and DSPLAY that offer this facility. Some of these techniques use methods like window, remez and frequency sampling [1]. The only thing the user needs to concern is the the FIR filter specification which can involve some few additional parameters related to the method chosen.

FIR filter design using genetic algorithms has been studied at least for 16 years [2]. But choosing this way to get a digital filter, the user usually has to know a considerable number of additional parameters besides the filter specifications. They are the evolutionary parameters. Some of them need to be adjusted to each new filter specification. Another feature is the stochastic behavior of this kind of application: the user previously needs to know that it can be necessary to run more than one execution to get an interesting solution. And this solution can be different at each time the application is run. These two features of a FIR filter design tool based on genetic algorithms tend to restrict it for people that have some knowledge of evolutionary computation. This work is an attempt to hide the evolutionary complexity offering sub-optimal FIR filter responses with a compromise of an average number of four executions. A sub-optimal pattern was created and the reference for this was the Parks-McClellan method implemented in MATLAB through the remez command [3].

II. FIR filter design

A FIR digital filter frequency response can be calculated from [1]:

$$H(k)=\sum_{n=0}^{M-1} h(n)\times e^{\left(\frac{-j2\pi}{N}\right)\times kn} \tag{1}$$

Where: *H(k)* is the Discrete Fourier Transform complex vector. This is the FIR digital filter frequency response; *N* is the number of collected points during the sampling process; *k* is an index varying from zero to *N-1*; *h(n)* is the FIR filter response vector to the unit impulse. This vector corresponds to the FIR filter coefficients; and *M* is the number of the FIR filter coefficients.

A digital filter gives a realizable version of a desired frequency response that was specified as part of the filter specifications. This happens because an ideal digital filter response is unrealizable.

To express *H(k)* as a function of the normalized frequency, it can be used [1]:

$$f=\frac{k}{N-1} \tag{2}$$

Where *f* is the normalized frequency ranging from 0.0 to 1.0 cicles/sample.

Depending on the number of coefficients and the simmetry of h(n), the FIR filters can be classified in four categories (types I to IV), three of them requiring some restrictions to give a specific frequency response. This work covers the four categories.

The complex vector *H(k)* is more useful when viewed as separated in magnitude and phase frequency responses. There is a symmetry in the FIR filters coefficients that guarantees a linear phase frequency response. Because of this, it is more frequent to found FIR filter magnitude frequency responses.

Table I presents some parameters used in a FIR filter desired frequency response specification.

TABLE I

FIR FILTER FREQUENCY RESPONSE USUAL PARAMETERS.

Parameter	Description
A_1 , A_2 ,A_3	Amplitudes of bands #1, #2, #3
f_{11} , f_{12}	Initial and final frequency of band #1
f_{21} , f_{22}	Initial and final frequency of band #2
f_{31} , f_{32}	Initial and final frequency of band #3
δ_1	Maximum ripple allowed for band #1
δ_2	Maximum ripple allowed for band #2
δ_3	Maximum ripple allowed for band #3
Δt_{12}	Transition width between bands #1 and #2
Δt_{23}	Transition width between bands #2 and #3
M	Number of FIR filter coefficients

FIR filters are Linear and Time Invariant systems. So, the filtering process can be made trough the convolution of the FIR filter coefficients and the signal *x(n)* to be filtered:

$$y(n) = x(n) * h(n) = \sum_{k=0}^{U} x(k)h(n-k) \tag{3}$$

Where: *y(n)* is the filtered signal; *h(n)* is the FIR filter unit response; *x(n)* is the signal to be filtered; * is the convolution operator symbol; *U* is the sum of the number of samples founded in *h(n)* and in *x(n)*; and *n* varies from zero to *U-1*.

It is possible to get from eq. (3) the following conclusion: the larger number of *h(n)* coefficients (*M*), the more precise will be the filtering process. Larger values of *M* offer a better quality of filtering. Better here can be understood as a FIR filter frequency response with a minimum transition width and a minimum value of ripple. But also through eq. (3), it is possible to conclude that for a same signal *x(n)* to be filtered, the number of products between *x(n)* and *h(n)* is regulated by the number of FIR filter coefficients, *M*. And, because of this, larger values of *M* cause larger delays during the filtering process. This can be unacceptable in some real-time applications.

The present work also tries to enforce the following area of study: the fetch for an acceptable FIR filter response in terms of quality with a minimum value of FIR filter coefficients.

III. GENETIC ALGORITHMS AND THE PROPOSED MODELING

A GA is based on a sequence of actions that, among others, can be represented by Fig. 1 [4]:

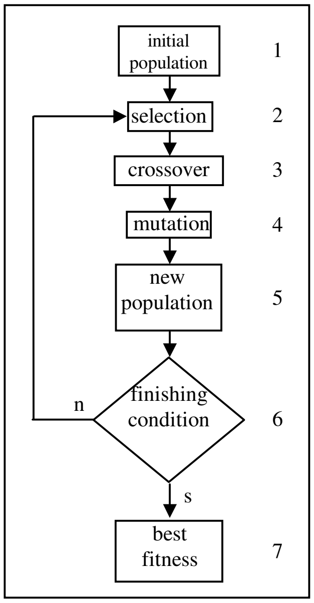

Figure 1: A GA basic flow.

A brief description of these actions with the model adopted by this work is:

a) a possible numerical solution of the problem is codified as an individual. Such representation usually adopts symbols to codify the numerical solution. The vector of the individual corresponds to the called chromosome. In the present work, an individual is the *h(n)* vector represented by the binary alphabet either in Gray or binary positional codification, as will be described in the metodology;

b) a set of individuals is generated at random. This set is called population and corresponds to the step 1 in Fig. 1. It is called the search space, *S*, that corresponds to all possible solutions that can be formed with the chosen alphabet. In the present work, the search space is variable with the number of FIR filter coefficients according to:

$$S = 2^l \tag{4}$$

Where: *S* is the search space; and *l* is the size in bits of the chromosome: 77 for *M*=13 or 14, 88 for *M*=15 or 16, 99 for *M*=17 or 18 and so on. One floating point coefficient with a precision of 1×10^{-3} precision corresponds to 11 bits. Because of the simmetry, only half of the coefficients are mapped into the chromosome;

c) this population is submitted to an evaluation. Each individual is tested according to how good it is as an optimal or suboptimal numerical solution for the proposed problem. In this way, each individual receives a score. The name of the mathematical function that does this job is the fitness function. In the present work, the fitness used is:

$$fitness = \sum_{i=0}^{N-1} \frac{1}{\left(\left|H_d(f_i)\right| - \left|H_i(f_i)\right|\right)^2} \qquad (5)$$

Where: *fitness* is the fitness function; $|H_d(f)|$ is the desired (given by user specification) magnitude frequency response; and $|H_i(f)|$ is the magnitude frequency response from each individual;

$|H_i(f)|$ is calculated through Eq. (1), because an individual corresponds to one vector *h(n).* This fitness function is one kind of a MSE function that numerically indicates how close an answer is from the desired answer. This happens in step one of Fig. 1 in the first time;

d) it is applied in this evaluated population some mechanism of selection. This is an attempt to exclude from the next steps the individuals with low values of fitness. In the present work, it was used the following selection processes: Roullete Wheel, Stochastic Tournament, and the Stochastic Universal Sampling, as will be described in the methodology. This happens in step two of Fig. 1;

e) the next step, with the selected population, is a recombination mechanism called crossover. A crossover consists basically of changes in portions of the chromosome between two individuals. The crossover is applied in this selected population with a probability *pcross* between 0.0 to 1.0. In the present work the following crossover techniques were used: one point and two point crossover, as will be described in the methodology. This happens in step three of Fig. 1;

f) after this, another operation over the chromosomes is applied. It is the mutation, that consists basically of a change in one or more bits in an individual at random. This operation is applied with a probability *pmutation* also ranging from 0.0 to 1.0. In the present work the following mutation operations were used: single bit mutation and multiple field mutation, as will be described in the methodology. This is step four in Fig. 1;

g) this new population is evaluated as the same way as made in the step described in item *c*. This happens in step five of Fig. 1;

h) it is called *generation* to the steps covered by the items *d* to *g* inclusive. Every time a generation is concluded, a finishing condition is tested to finish the GA run. In the present work the finish condition is a max generation number. This is step six in Fig. 1;

i) if the finish condition is not true, there is a return to the step that corresponds to item *d* and a new generation cycle is executed. This happens in the decision structure 6 in Fig. 1;

j) if the finish condition is true, the last population is obtained and the most fit individual in this population is the best solution the GA can give.

IV. FIR FILTER DESIGN WITH GENETIC ALGORITHMS

Counting from reference [1] until these days, more than one hundred FIR Filter Design methods using GAs are already published, e.g., [5], [6], and [7]. In these works, the prior goal was not a look for a method or software based on GAs for give FIR filter coefficients without the need to adjust evolutionary parameters at each new specification and with a predetermined number of mean trials to get an acceptable answer.

Two related works were found.

The first is a MATLAB toolbox [8]. Some differences between that tool and the proposed method are: a) The GA parameters specifications. In that tool the GA parameters are configurable, that is, the user must know GAs. And here, the proposed work, this kind of knowledge is not necessary; b) The platform: to run that application, it is necessary to have MATLAB. And here, the proposed work, the final version of the tool runs over Windows directly.

The second work, also a MATLAB toolbox, is a tool called CSDFIR [9]. The final version of this tool is automatic, but today it owned by a private company.

V. METHODOLOGY

To achieve the proposed goals, the following strategy was taken: a) two suboptimal conditions related to a well know specialist method was specified; and b) a bank of FIR filter frequency response specifications was specified. These filter specifications tried to cover all the possible kinds of hard and easy to solve filters. All of them were specified with a zero and one amplitudes. Arbitrary levels were not proposed.

With the conditions and the filter bank, a three-phase group of tests was proposed, differing slightly from a previous work [10]: a) Phase 1: the filter bank was tested with a fixed number of coefficients changing and verifying the results of the following GA parameters: chromosome binary representation, selection process, crossover operator, mutation operator, probabilities of mutation and crossover, and a auxiliary technique of selection called elitism [4]. A score based on the quality of the results for each configuration was proposed. If the configuration passed to all the proposed filters, it received a score based on the quality of the results. If the configuration did not pass through all the proposed filters, its score was zero; b) Phase 2: with the best score-approved configuration of Phase 1, the same filter bank was tested, but with a variable number of coefficients, to check if the configuration is robust to support different search spaces. The GA parameters to be changed in this phase were the population size and the maximum number of generations; and c) Phase 3: with the most robust configuration of Phase 2, several variable amplitude and variable number of coefficients FIR filter specifications were tested. Also in this phase, the LTI superposition propriety was tested for hard to solve filters or filters that could not be solved when specified directly.

During phase two and three, it was looked for some pattern behavior in some GA parameters with the variation of the FIR filter coefficients. This was done as an attempt to find some mathematical relationship between the number of coefficients and them.

The limitations of the proposed method and GA modeling were: a) do not cover more than three amplitude levels between 0.0 and 0.5 cycles/sample; b) do not cover any frequency range smaller than 0.01 cycles/sample; c) do not cover all the possibilities of arbitrary level response frequencies; and d) depending of the number of coefficients (which determine the search space), the execution cannot be processed in usual machines.

With this strategy the expectation was to find a configuration that covered the requisites of any FIR filter specification, through tests.

A. *The Specialist Method to be Compared*

To make the quality response comparisons, it was chosen the MATLAB implemented version of the Parks-McClellan method, with equal weights for pass and reject bands.

B. *First and Second Sub-Optimal Conditions*

A FIR filter frequency response given by the GA that is considered accepted must satisfy two conditions.

The condition here called the First Suboptimal Condition, FSC is:

$$\left|20\times\log(A_i+\delta_{iAG}/2)-20\times\log(A_i+\delta_{iR}/2)\right|\le 5 \tag{6}$$

Where: δ_{iAG} are the GA ripples of bands #1, #2, #3; δ_{iR} are the Parks-McClellan ripples of bands #1, #2, #3; and A_i are the amplitude specifications of bands #1, #2, #3.

And the Second Suboptimal Condition, SSC, is:

$$\left|f_{iAG}-f_{iR}\right|\le 0.05 \tag{7}$$

Where: f_{iAG} are the GA transition frequency edges between the bands; f_{iR} are the Parks-McClellan transition frequencies edges between the bands.
with equal weights for pass and reject bands.

C. *The FIR Filter Bank*

Some FIR filter specifications of the filter bank are presented in Table II.

TABLE II
SOME FIR FILTER SPECIFICATIONS

#	f_{12}	f_{21}	f_{22}	f_{31}	A_1	A_2	A_3
1	0.35	0.42	0.45	0.45	1	0	0
5	0.15	0.19	0.30	0.30	1	0	0
7	0.04	0.04	0.05	0.12	0	0	1
8	0.35	0.35	0.40	0.47	0	0	1
10	0.40	0.40	0.44	0.47	0	0	1
12	0.05	0.11	0.39	0.45	0	1	0
16	0.05	0.10	0.40	0.45	1	0	1
18	0.15	0.16	0.44	0.45	1	0	1

For all filter specifications: f_{11} = 0.0 and f_{32} = 0.5 cicles/sample.

VI. RESULTS

A GA filter response that does not satisfy the FSC, i.e., it is not ok with Eq. (6), is presented in Fig. 2:

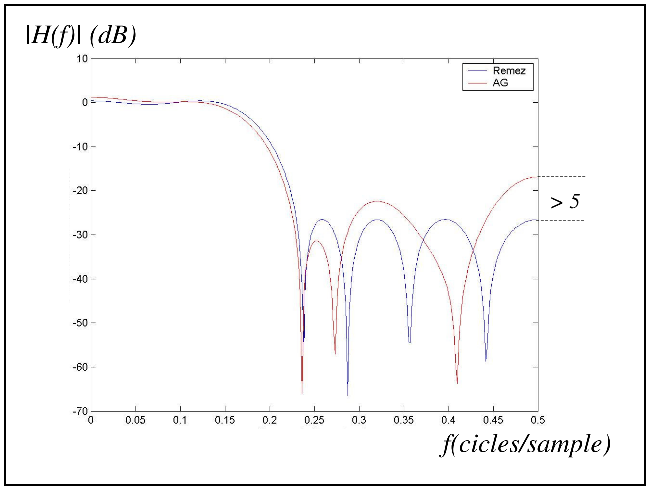

Figure 2: GA filter response that does not satisfy the FSC.

An example of a FIR filter frequency response that does not satisfy the SSC, i.e, Eq. (7), is presented in Fig. 3:

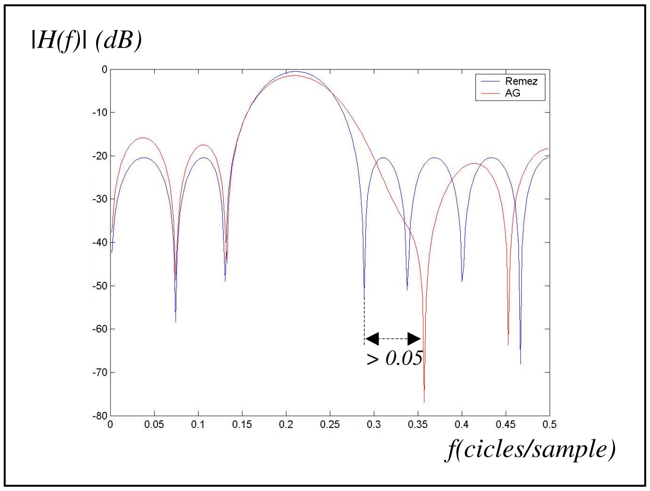

Figure 3: GA filter response that does not satisfy the SSC.

Figure 5 presents one of the good results of Phase 1 and Table III presents some scores obtained by the proposed configurations for Phase 1. For this table, it was used a population size of 160 individuals, a number of FIR filter coefficients M of 15 and the computer used was a Pentium notebook, 1.7 GHz with 512 Mb of RAM.

The number of coefficients (*M*) was varied in Phase 2. This was done with the only approved configuration from Phase 1. For each new value of *M*, new other values for population size and maximum number of generations were needed.

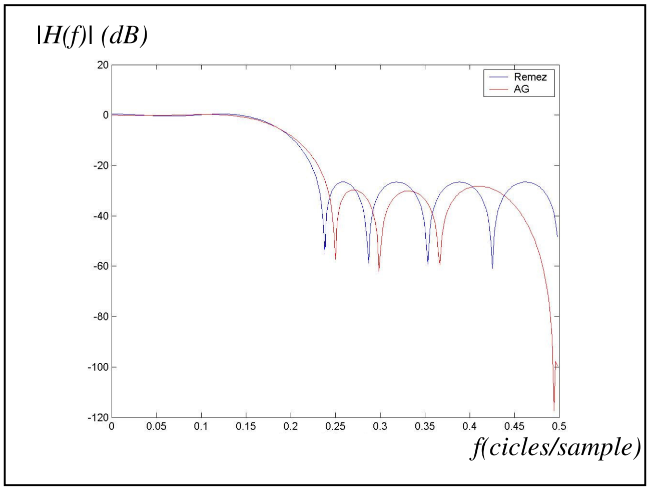

Figure 5: One result that satisfy the FSC and the SSC.

TABLE III
SOME CONFIGURATION SCORES FOR PHASE 1

config #.	*1*	*2*	*3*	*4*
type	I	I	I	I
selection	suselect	rselect	tselect/4	tselect/4
maxgen	500	500	500	500
cross	twoptx	oneptx	twoptx	oneptx
pcross	0.85	0.90	0.85	0.90
mutation	bitmutat	multimut	multimut	bitmutat
pmut	0.10	0.01	0.01	0.10
bin rep	BinGray	BinPos	BinPos	BinGray
elitism	Yes	Yes	No	Yes
t_{AG}	11 s	11 s	11 s	11 s
score	5.00	0.00	0.00	0.00
config. #	*5*	*6*	*7*	*8*
type	II	II	III	IV
selection	suselect	suselect	suselect	suselect
maxgen	500	1000	500	500
cross	twoptx	twoptx	twoptx	twoptx
pcross	0.85	0.85	0.85	0.85
mutation	bitmutat	bitmutat	bitmutat	bitmutat
pmut	0.10	0.10	0.10	0.10
bin rep	BinGray	BinGray	BinGray	BinGray
elitism	Yes	Yes	Yes	Yes
t_{AG}	11 s	11 s	11 s	11 s
score	0.00	5.00	7.81	5.42

Conventions: type = filter type, selection = selection process, maxgen = max generation number, crossover = crossover operator, mutation = mutation operator, bin rep = binary representation, t_{AG} = time spent, suselect = Stochastic Universal Sampling process, rselect = Roullete Whell process, tselect = tournament process, twoptx = 2-point crossover, oneptx = one-point crossover, bitmutat = single bit mutation operator, multmut = multiple field mutation operator, BinGray = Gray binary representation, BinPos = Positional binary representation. pcross/pmut = probabilities of cross-over and mutation.

One of the good results obtained in Phase 2 can be seen in Figure 6.

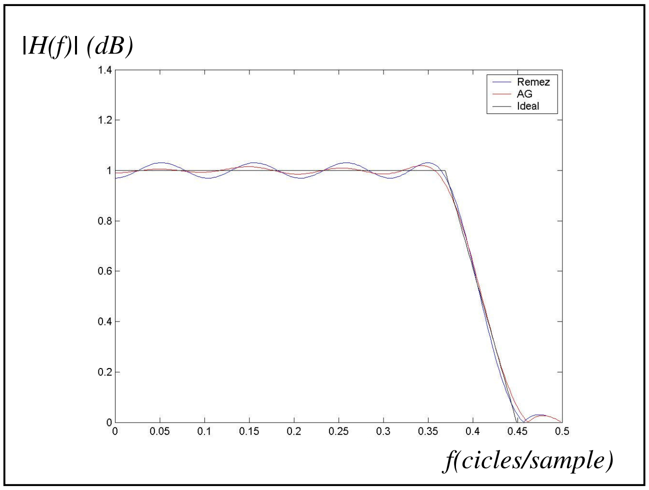

Figure 6: One FSC and SSC compliant result in Phase 2.

This result was obtained for a filter specification with the configuration approved from Phase 1 (configuration numbers 1, 5, 6, 7 and 8) except for: M = 20, population size = 2310, and maxgen = 1500.

For hard to solve and arbitrary levels specifications, the LTI superposition principle is also valid. Among other tests, this was verified in Phase 3. The AG FIR filter frequency response presented in Figure 7 was obtained directly as well as trhough the sum of a low-pass and a high-pass FIR filter specification.

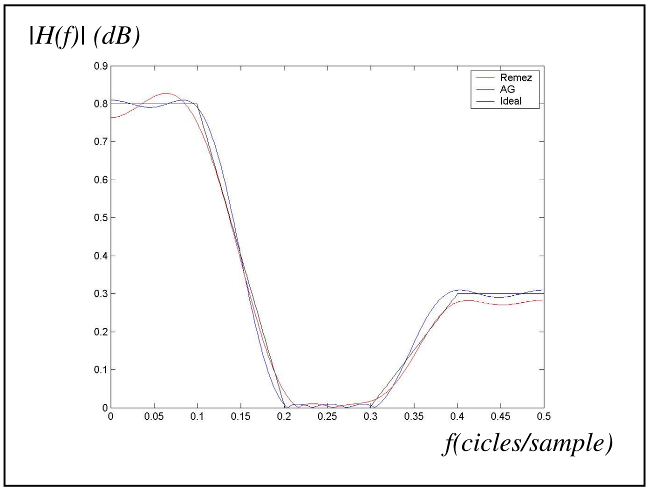

Figure 7: Arbitrary level filter response obtained in Phase 3.

VII. CONCLUSIONS

It can be seen in configuration #2 of Table III, that the canonic version of the model could not satisfy the proposed goals, requiring more advanced processes.

It was observed that the elitism was always present with the configuration that passed through the filter bank.

The binary Gray representation presented more resolution than the conventional binary positional representation. This was expected, because in Gray representation, only one bit can change from one number and its next or prior.

The Stochastic Tournament selection process did not pass in Phase 1, but it presented very interesting results (more successful results than the Roullete Wheel selection process). The Stochastic Universal Sampling selection method presented the following useful behavior: at each attempt, one different suboptimal result was obtained. Comparing it to the Stochastic Tournament, this one presented always the same suboptimal result for different attempts.

A GA with predefined parameters was obtained to solve variable coefficients, variable filter type, and fixed in zero or one amplitude FIR digital filter specifications.

The fixed configuration obtained was: *binary representation* = Gray, *fitness function* = Eq. (5), *selection* = Stochastic Universal Sampling, *crossover* = two-point crossover, *mutation* = one-bit mutation, *elitism* = yes, *probability of crossover* = 0.85, and *probability of mutation* = 0.10. For all the filter specifications of the filter bank proposed and for more than fifty others, this GA did run on in an average number of four times to satisfy the conditions specified in Eq. (6) and Eq. (7).

These results used prior tests [10] to stablish the following relationships:

$$posize = 160 \times 3.8^{L_2 - 8} \tag{8}$$

$$maxgen = 1000 + (L_2 - 8) \times 50 \quad , L_2 \geq 8 \tag{9}$$

Where: popsize is the population size; L_2 = M/2 for M even and (M+1)/2 for M odd, maxgen is the maximum number of generations.

With equations (8) and (9) it was possible to get an automatic procedure to obtain FIR filter responses for a GA. And this was done changing another prior work [11].

Acknowledgment

The first author thanks to Professor Álvaro Luiz Stelle and to Professor Heitor Silvério Lopes for their strong support for the development of the work.

References

[1] J. Proakis and D. Manolakis, *Digital Signal Processing: principles, algorithms and applications*, 3rd ed. , Upper Saddle River: Prentice-Hall, 1996.

[2] D. Sucley, "Genetic algorithm in the design of FIR filters", *IEEE Proceedings G: Circuits, Devices and Systems*, vol. 138, n.2, pp 234-238, 1991.

[3] L.R. Rabiner, J.H. McClellan, and T. W. Parks. "FIR digital filter design techniques using weighted Chebyshev approximation", *Aproceedings of the IEEE*. New York: vol. 63, n. 4, pp 595-610, 1975.

[4] D. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Reading: Addison-Wesley, 1989.

[5] R. Cemes and D. Ait-Boudaoud, "Multiplier-less FIR filter design with power-of-two coefficients", In: Digital and Analogue Filters and Filtering Systems, London, *Colloquium on*, IEE, Digest No. 199, pp. 6/1-6/4, 1993.

[6] A. Lee, M. Ahmadi, G.A. Julien, R.S. Lashkari, W.C. Miller, "Design of 1-D FIR filters with genetic algorithms", In: 5th International Symposium on Signal Processing and its Applications, ISSPA´99, Brisbane. *Proceedings*, New York: IEEE, 1998, vol. 2, pp. 955-958.

[7] N. Karaboğa, and B. Çetinkaya, "Performance comparison of genetic algorithm based methods of digital filters with optimal magnitude response and minimum phase", In: 46th IEEE International Midwest Symposium on Circuits and Systems, MWSCAS 2003, Cayro. *Proceedings*, New York: vol. 2, pp 644-647.

[8] S. M. Ahmed, "Design of FIR filters with arbitrary amplitude and phase specifications using genetic algorithms", In: 46th IEEE International Midwest Symposium on Circuits and Systems, MWSCAS 2003, Cayro. *Proceedings*, New York: vol. 2, pp 648-651.

[9] Y.M. Hasan, L.J. Karam, M. Falkinburg, A. Helwig, and M. Ronning, "Canonic signed digit Chebyshev FIR filter design", *IEEE Signal Processing Letters*, vol. 8, n. 6, pp 167-169, 2001.

[10] A.M. Barros, A.L. Stelle, and H.S. Lopes, "FIR filter design using genetic algorithms with a minimum number of parameter adjustments", In: XXVII Iberian Latin-American Congress on Computational Methods in Engineering, CILAMCE 2006, Brazilian Assoc. for Comp. Mechanics & Latin American Assoc. of Comp. Methods in Engineering, *Proceedings*, Belém, 2006, in press.

[11] A.M. Barros, A.L. Stelle, and H.S. Lopes, "A FIR filter design tool using genetic algorithms", In: XXXIV Congresso Brasileiro de Ensino de Engenharia, COBENGE 2006, *Proceedings*, Passo Fundo: Ed. Universidade de Passo Fundo, pp. 1799-1811, 2006.