

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**  
Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial

---

**DISSERTAÇÃO**  
apresentada à UTFPR  
para obtenção do grau de

**MESTRE EM CIÊNCIAS**

por

**ANDRÉ MACÁRIO BARROS**

---

**PROJETO DE FILTROS FIR ATRAVÉS DE ALGORITMOS  
GENÉTICOS**

---

Banca Examinadora:

Presidente e Orientador:

**PROF. ÁLVARO LUIZ STELLE, PHD**

**UTFPR**

Examinadores:

**PROF. DR. HEITOR SILVÉRIO LOPES**

**UTFPR**

**PROF. DR. ANSELMO RODRIGUES**

**UNICENP**

Curitiba, Abril de 2006.

# **Livros Grátis**

<http://www.livrosgratis.com.br>

Milhares de livros grátis para download.

**ANDRÉ MACÁRIO BARROS**

**PROJETO DE FILTROS FIR ATRAVÉS DE ALGORITMOS GENÉTICOS**

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial da Universidade Tecnológica Federal do Paraná, como requisito parcial para a obtenção do grau de “Mestre em Ciências” – Área de Concentração: Informática Industrial.

Orientador: Prof. Álvaro Luiz Stelle, PhD.

Curitiba

2006

Ficha catalográfica elaborada pela Biblioteca da UTFPR – Campus Curitiba

B277p Barros, André Macário

Projeto de filtros fir através de algoritmos genéticos / André Macário Barros . Curitiba. UTFPR, 2006

XII, 112 p. : il. ; 30 cm

Orientador: Prof. Dr. Álvaro Luiz Stelle

Dissertação (Mestrado) – Universidade Tecnológica Federal do Paraná. Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Curitiba, 2006

Bibliografia: p. 105-112

1. Processamento de sinais. 2. Algoritmos genéticos. 3. Matlab. I. Stelle, Álvaro Luiz, orient. II. Universidade Tecnológica Federal do Paraná. Curso de Pós-Graduação em Engenharia Elétrica e Informática Industrial. III. Título.

CDD: 621.3822



## AGRADECIMENTOS

Aos meus pais pela criação, pela formação moral que tive e por acreditarem incondicionalmente nos meus sonhos.

Aos Professores Paulo Brero de Campos e Alfranci Freitas Santos pelas suas cartas de recomendações.

Ao meu orientador, Professor Álvaro Luiz Stelle, pela sua segurança transmitida, seu amplo domínio da área, sua confiança, seu apoio constante e uma paciência infinita durante todo este período.

Ao Professor Heitor Silvério Lopes, pela sua orientação em diversos momentos importantes desta pesquisa, pelas suas idéias fornecidas, sua visão da área, suas recomendações e pelo seu pronto atendimento sempre em que foi solicitado.

A minha esposa Izabel, a minha eterna namorada, por sua obstinada determinação em me acompanhar e apoiar incondicionalmente neste e em todos os desafios que apareceram em nosso caminho.

A minha filha Andréa, o meu eterno neném, por ter se contentado com tão pouco tempo para brincar com seu pai que tanto a ama durante este período.

A todos os meus colegas que direta ou indiretamente contribuíram para o desenvolvimento deste trabalho.

E agradeço a Deus, por iluminar meu caminho, possibilitando-me chegar até aqui.

## SUMÁRIO

<b>LISTA DE FIGURAS</b> .....	vii
<b>LISTA DE TABELAS</b> .....	ix
<b>LISTA DE ABREVIATURAS E SIGLAS</b> .....	x
<b>RESUMO</b> .....	xi
<b>ABSTRACT</b> .....	xii
<b>1 INTRODUÇÃO</b> .....	1
1.1 MOTIVAÇÕES .....	1
1.2 OBJETIVOS .....	3
1.3 ESTRUTURA DA DISSERTAÇÃO .....	4
<b>2 FUNDAMENTAÇÃO TEÓRICA</b> .....	5
2.1 PROCESSAMENTO DIGITAL DE SINAIS .....	5
2.1.1 Introdução .....	5
2.1.2 Filtros Digitais .....	7
2.1.3 Sistemas LIT .....	8
2.1.4 A Transformada Discreta de Fourier .....	10
2.1.5 A Transformada-z .....	14
2.1.6 Tipos de filtros FIR .....	16
2.1.7 Projeto de filtros FIR .....	20
2.1.8 Projeto de filtros FIR pelo método de Parks-McClellan .....	21
2.1.9 Projeto de filtros FIR por outros métodos .....	26
2.1.10 A transformada rápida de Fourier .....	27
2.1.11 Filtros digitais sem multiplicação .....	28
2.1.12 Quantização dos coeficientes na implementação .....	29
2.2 ALGORITMOS GENÉTICOS .....	30
2.2.1 Introdução .....	31
2.2.2 Outros métodos numéricos e aplicabilidade .....	33
2.2.3 Representação cromossômica .....	34
2.2.4 Função-objetivo, função-penalidade e função de <i>fitness</i> .....	35
2.2.5 Convergência prematura, diversidade genética e pressão seletiva .....	35
2.2.6 Métodos de seleção .....	35

2.2.7 Operadores de cruzamento .....	37
2.2.8 Operadores de mutação .....	37
2.2.9 Escalonamentos de <i>fitness</i> .....	38
2.3 REVISÃO DA LITERATURA .....	39
2.3.1 Trabalhos correlatos .....	39
2.3.2 Trabalhos na área .....	43
3 METODOLOGIA .....	45
3.1 MATERIAIS E INFRA-ESTRUTURA .....	45
3.1.1 Computador .....	45
3.1.2 <i>Softwares</i> utilizados .....	46
3.2 O MÉTODO .....	47
3.2.1 Descrição geral do cenário dos experimentos .....	47
3.2.2 Variáveis do experimento .....	48
3.2.3 Condições de resposta sub-ótima .....	51
3.2.4 Formulação do problema e da hipótese .....	53
3.2.5 Operacionalização das variáveis e critérios de redução .....	53
3.2.6 Os testes propostos .....	58
3.2.6.1 Primeira fase .....	58
3.2.6.2 Segunda fase .....	61
3.2.6.3 Terceira fase .....	61
3.2.7 A modelagem do AG .....	62
3.2.8 Limitações .....	68
3.2.9 Os <i>softwares</i> .....	69
3.2.9.1 O <i>software Especificador</i> .....	69
3.2.9.2 O <i>software AG</i> .....	71
3.2.9.3 O <i>software RespFreq</i> .....	73
4 RESULTADOS .....	77
4.1 PRIMEIRA FASE .....	77
4.1.1 Configurações utilizadas .....	77
4.1.2 Resumo dos resultados .....	78
4.1.3 Exemplos de resultados NOK, SS e OK .....	80
4.2 SEGUNDA FASE .....	83
4.2.1 Resumo dos resultados .....	83

4.2.2 A influência dos parâmetros sobre os resultados .....	84
4.2.3 Resultado relevante de possível aplicação do AG .....	86
4.3 TERCEIRA FASE .....	86
4.3.1 Teste da superposição .....	87
4.3.2 Teste da normalização da <i>fitness</i> .....	89
4.3.3 Evolução das curvas de <i>fitness</i> .....	89
4.3.4 Teste do escalonamento de <i>fitness</i> .....	91
4.3.5 Teste da ordem do filtro .....	92
4.3.6 Teste com filtros de amplitudes arbitrárias .....	93
4.3.7 Teste com outras probabilidades de mutação .....	93
4.3.8 Resultado relevante de provável máximo local .....	95
5 DISCUSSÃO E CONCLUSÕES .....	97
5.1 ANÁLISE DOS RESULTADOS .....	97
5.1.1 Análise da primeira fase .....	97
5.1.2 Análise da segunda fase .....	98
5.1.3 Análise da terceira fase .....	99
5.2 CONCLUSÕES .....	102
5.2.1 Conclusões gerais .....	102
5.2.1.1 AG automático .....	102
5.2.1.2 AG não automático .....	103
5.2.2 Conclusões específicas .....	103
5.3 LINHAS DE AÇÃO FUTURAS .....	104
REFERÊNCIAS BIBLIOGRÁFICAS .....	105

## LISTA DE FIGURAS

1	Método de obtenção de um filtro digital via AGs .....	2
2	Um sinal de ECG .....	5
3	A Forma Direta I .....	7
4	TDF de um sinal de ECG .....	13
5	$h(n)$ de um filtro FIR, Tipo II, com $M = 16$ .....	18
6	$ H(f) $ do filtro FIR da Figura 5 .....	18
7	$h(n)$ de um filtro FIR, Tipo III, com $M = 17$ .....	18
8	$ H(f) $ do filtro FIR da Figura 7 .....	19
9	$h(n)$ de um filtro FIR, Tipo IV, com $M = 14$ .....	19
10	$ H(f) $ do filtro FIR da Figura 9 .....	19
11	Parâmetros de especificação de um filtro digital .....	20
12	Alternâncias em filtros gerados pelo MPM .....	23
13	Fluxo básico de um AG .....	31
14	Visão geral do experimento .....	47
15	<i>Ripples</i> envolvidos no experimento .....	51
16	Distorções causadas por arredondamentos nos coeficientes dos filtros .....	63
17	Diferença entre $H(f) _{3\text{ casas}}$ e $H(f) _{15\text{ casas}}$ .....	64
18	Fluxograma do <i>software Especificador</i> .....	69
19	Fluxograma da rotina de geração de arquivos .....	70
20	Fluxograma da rotina do cálculo do $M$ (recomendado) .....	70
21	Interface do <i>software Especificador</i> .....	71
22	Interface do <i>software AG</i> .....	72
23	Fluxograma do <i>software RespFreq</i> .....	74
24	Fluxograma da rotina <i>Captura</i> .....	74
25	Fluxograma da rotina <i>Resposta</i> .....	75
26	Fluxograma da rotina <i>Resposta dB</i> .....	75
27	Interface do <i>software RespFreq</i> .....	76
28	Primeiro exemplo de um resultado NOK .....	81
29	Segundo exemplo de um resultado NOK .....	81
30	Exemplo de resultado SS .....	81
31	Um exemplo de resultado OK .....	82

32	Resposta da Figura 31 em decibéis .....	82
33	Resposta NOK com $maxgen = 500$ .....	84
34	Resposta SS com $maxgen = 750$ .....	85
35	Resposta OK com $maxgen = 1500$ .....	85
36	Filtro passa-faixa estreito que o MPM não conseguiu gerar .....	86
37	$ H_1(f) $ do teste da subseção 4.3.1 .....	88
38	$ H_2(f) $ do teste da subseção 4.3.1 .....	88
39	$ H(f) $ do teste da subseção 4.3.1 .....	88
40	Curvas de <i>fitness</i> para um resultado do registro 2 da Tabela 6 .....	90
41	Número de indivíduos com <i>fitness</i> acima da média por geração .....	90
42	<i>Fitness</i> dos indivíduos da última população .....	90
43	Filtro NOK obtido via escalonamento de <i>fitness</i> linear .....	91
44	Curvas de <i>fitness</i> com escalonamento linear .....	91
45	Filtro em OK confrontado com MPM equivalente .....	92
46	Filtro via MPM com mesmos <i>ripples</i> que o filtro AG .....	93
47	Desempenho de <i>fitness</i> com $pmutation = 0,01$ .....	94
48	Uma solução do tipo NOK .....	95
49	Melhor indivíduo de uma geração intermediária .....	95

## LISTA DE TABELAS

1	Parâmetros de especificação de um filtro digital .....	21
2	Relação entre o nº de alternâncias $N_e$ e o nº de coeficientes $M$ .....	23
3	Perfil dos métodos, suas deficiências e características .....	34
4	Banco de filtros propostos .....	57
5	Planilha do teste principal .....	60
6	Notas para a primeira fase .....	78
7	Planilha do teste principal preenchida .....	79
8	Notas para os testes da segunda fase .....	84

## LISTA DE ABREVIATURAS E SIGLAS

AE	- Algoritmo Evolucionário.
AG	- Algoritmo Genético.
AIS	- <i>Artificial Immune System</i> .
AM	- Algoritmo Memético.
AS	- Anelamento Simulado.
CE	- Computação Evolucionária.
ECG	- Eletrocardiograma.
ED	- Evolução Diferencial.
EE	- Estratégia Evolucionária.
FIR	- <i>Finite Impulse Response</i> .
FORTTRAN	- <i>Formula Translation</i> .
GEP	- <i>Genetic Expression Programming</i> .
IC	- Inteligência Computacional.
IIR	- <i>Infinite Impulse Response</i> .
HE	- <i>Hardware</i> Evolutivo.
LIT	- Linear e Invariável com o Tempo.
MPM	- Método de Parks-McClellan.
MSE	- <i>Mean Square Error</i> .
PDS	- Processamento Digital de Sinais.
PE	- Programação Evolucionária.
PG	- Programação Genética.
POT	- <i>Power of Two</i> .
PCRSO	- Primeira Condição de Resposta Sub-Ótima.
PSCRSO	- Primeira e Segunda Condições de Resposta Sub-Ótima.
PSO	- <i>Particle Swarm Optimization</i> .
RNA	- Redes Neurais Artificiais.
SC	- Sistema Classificador.
SCRSO	- Segunda Condição de Resposta Sub-Ótima.
TDF	- Transformada Discreta de Fourier.
TRF	- Transformada Rápida de Fourier.

## RESUMO

Um método baseado em Algoritmos Genéticos foi desenvolvido e implementado em *software* para fornecer os coeficientes de um filtro FIR, cuja resposta em frequência é classificada como quase ótima em relação a um método especialista escolhido. Resultados comparativos evidenciam a contribuição do trabalho para os casos de frequências de transição abaixo de determinado valor. O *software* desenvolvido oferece a possibilidade de gerar filtros FIR com: três faixas de frequências com amplitudes arbitrárias, um número variável de coeficientes (pares e ímpares) e simetria positiva ou negativa. O método que originou o *software* busca a diminuição do número de parâmetros do Algoritmo Genético que requeiram uma configuração caso a caso. Portanto, a segunda contribuição do trabalho é a de oferecer uma forma de ser utilizada uma ferramenta baseada em Algoritmos Genéticos sem a necessidade de serem conhecidas as complexidades que envolvem sua utilização, por exemplo, operadores de recombinação, probabilidades de cruzamento e representação cromossômica. O resultado desta contribuição é uma ferramenta que proporciona ao usuário que só entenda de Processamento Digital de Sinais um meio de obtenção de filtros FIR com uma qualidade útil para casos onde o método especialista que foi confrontado não atenda. O método especialista a partir do qual foi feita a classificação de respostas ótimas, quase ótimas ou insatisfatórias, foi a versão implementada no *software* MATLAB do Segundo Algoritmo de Remez, o de múltiplas trocas, adaptado à síntese de filtros FIR por Rabiner, Parks e McClellan. O *software* foi implementado a partir da ferramenta *GALOPPS* para Algoritmos Genéticos.

## ***ABSTRACT***

A genetic algorithm based method was developed and implemented. Its main task is to receive from the user a FIR filter specification (including the desired frequency response) and to give a set of FIR filter coefficients. This set of coefficients must be the best approximation the proposed genetic algorithm can reach for the desired frequency response specification. The software implementation offers the following specification possibilities: three arbitrary amplitude levels, positive and negative symmetry, and variable size order. The second scientific contribution of the proposed method is to offer a minimum number of genetic algorithm parameters needed to be changed from one to another different run (considering a different FIR filter specification). The specialist method used for comparison was the Parks-McClellan adaptation for FIR filters of the Second Remez Multiple Exchange Algorithm implemented in MATLAB. The software was implemented using the GALOPPS tool for genetic algorithms. Results were obtained which can establish a suboptimal condition compared with a specialist method. One scientific contribution of the proposed method is for the cases of short transition bands: the frequency response, even suboptimal, offers smaller ripples compared to the specialist method.

# CAPÍTULO 1

## INTRODUÇÃO

### 1.1 MOTIVAÇÕES

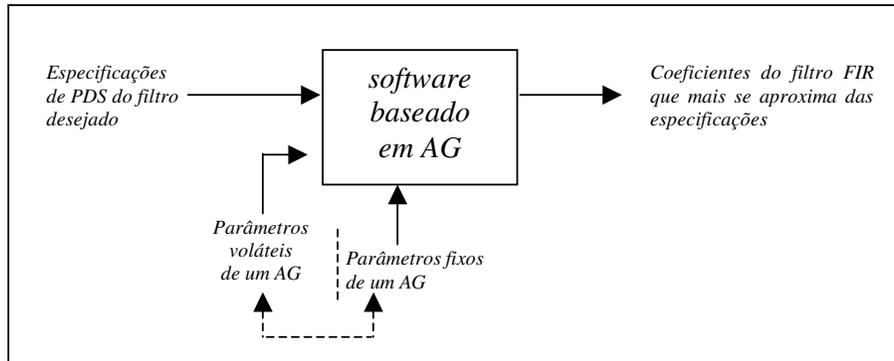
O projeto de filtros digitais envolve técnicas matemáticas e de PDS (Processamento Digital de Sinais), algumas das quais consagradas e implementadas em *softwares* gratuitos ou comerciais.

A obtenção dos coeficientes de um filtro digital através de uma determinada técnica de PDS que esteja disponível em *software* é normalmente simples: são passados os parâmetros de especificação do filtro desejado e o *software* fornece os coeficientes do filtro que mais se aproxima do que foi solicitado. Logo, na maioria das vezes, sequer há a necessidade de serem conhecidos detalhes a respeito do funcionamento interno do *software* ou do método que forneceu tal filtro.

Quando não há um *software* pronto que realize este trabalho por meio de uma determinada técnica, podem estar disponíveis na literatura os algoritmos computacionais ou ainda os códigos-fonte correspondentes à técnica procurada.

O projeto de filtros digitais, especificamente filtros do tipo FIR (*Finite Impulse Response*), por meio de AGs (Algoritmos Genéticos), é uma técnica alternativa, cujas linhas de pesquisa já iniciaram e estão em atividade no mínimo há dezesseis anos. São áreas, em sua maioria, com um alto grau de especificidade, voltadas à discussão de métodos que foram descobertos. Não há em tais áreas uma preocupação específica neste momento a respeito da forma através da qual um método baseado em AGs seja utilizado. Na maioria dos casos, é pressuposto que o utilizador de tal método esteja ciente de que irá precisar não só especificar a resposta em frequência do filtro desejado, assim como, para cada caso, ajustar algum tipo de parâmetro específico de AGs. Tais parâmetros são classificados neste trabalho como voláteis, pois precisam de ajustes caso a caso. São exemplos: tamanho de população, probabilidades de cruzamento e de mutação, número de gerações e semente pseudo-aleatória. Há ainda parâmetros de AGs que podem permanecer fixos e por este motivo classificados como tal, por exemplo, o método de seleção, o tipo de cruzamento e o tipo de mutação. Porém, esta relação entre parâmetros supostamente voláteis e fixos é dinâmica, podendo um parâmetro volátil

passar a ser fixo em uma outra modelagem escolhida. Uma possível representação deste cenário é a da Figura 1:



**Figura 1:** Método de obtenção de um filtro digital via AGs.

É pressuposto ainda, em um cenário como o da Figura 1: que o utilizador compreenda o perfil estocástico que uma execução de um AG tem, requerendo normalmente várias execuções para encontrar uma solução  $s_1$  quase ótima, podendo ainda ser diferente de outra quase ótima  $s_2$ , obtida em outra execução e ambas possam ser inferiores a uma solução ótima  $s$ , obtida por um método especialista.

Não foi localizado um *software* ou um código-fonte disponível gratuitamente para a comunidade acadêmica, no qual seja possível:

- a) Serem inseridas as especificações de PDS de um filtro FIR desejado;
- b) Serem inseridos um conjunto mínimo de parâmetros de AG, sejam eles fixos ou voláteis;
- c) Ser executado o AG com um mínimo de compromisso com resultados úteis (sucesso em 4 tentativas, por exemplo);
- d) Ser obtido um filtro FIR que ofereça uma motivação mínima para a utilização do *software* no lugar de outro que contenha um método especialista.

Portanto, esta lacuna observada é a motivação deste trabalho.

## 1.2 OBJETIVOS

O principal objetivo deste trabalho foi desenvolver um método para a síntese de filtros FIR por meio de AGs que busque minimizar o número de parâmetros voláteis do AG, visando proporcionar uma transparência ao método para o utilizador.

Como a abordagem escolhida foi experimental, o objetivo principal se estende para a sua implementação por meio de um *software* com as seguintes características:

- a) Ter uma interface gráfica para receber a especificação dos parâmetros de PDS;
- b) Ser capaz de processar especificações de filtros com um número de coeficientes variável, simetria positiva ou negativa, três faixas de frequência com amplitudes arbitrárias;
- c) Proporcionar um ajuste mínimo de parâmetros de AGs a cada vez que as especificações do filtro a ser buscado sejam alteradas;
- d) Fornecer o número de coeficientes recomendado para o método especialista atender às especificações passadas; e
- e) Oferecer, no máximo em quatro tentativas, uma resposta em frequência justificável para seu emprego, em comparação com os resultados oferecidos pelo MPM (Método de Parks e McClellan) (RABINER, MCCLELLAN e PARKS, 1975), que está implementado no programa MATLAB (MATHWORKS, 2002). Este é o método especialista que servirá de referência para a avaliação de resultados e será estudado no caso particular de pesos iguais para a faixa passante e a de rejeição.

Como objetivos específicos há:

- a) a elaboração de um arquivo `leia-me.txt` para facilitar a utilização do *software*;
- b) o fornecimento da versão executável do *software* para uso gratuito pela comunidade científica;
- c) a descrição sucinta das linhas de ação futuras que possam utilizar este método como base; e
- d) o desenvolvimento de um artigo que descreva o método utilizado e forneça detalhes gerais de sua implementação.

### 1.3 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação está organizada em cinco capítulos.

No Capítulo 2 há a fundamentação teórica, tanto de PDS quanto de AGs e há a apresentação de alguns trabalhos relevantes na área e correlatos.

No Capítulo 3 é feita a exposição dos materiais, do método e dos critérios empregados.

No Capítulo 4, há os resultados (gráficos e tabelas) obtidos.

E no Capítulo 5 há a discussão sobre tais resultados e a que conclusões foram chegadas.

## CAPÍTULO 2

### FUNDAMENTAÇÃO TEÓRICA

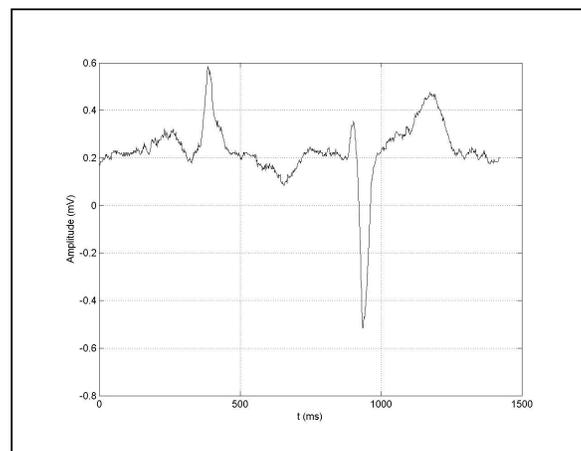
Este capítulo é dividido em três seções. A primeira e a segunda são destinadas a oferecerem a fundamentação teórica de PDS e de AGs. A terceira seção apresenta os trabalhos correlatos.

#### 2.1 PROCESSAMENTO DIGITAL DE SINAIS

As subseções 2.1.1 a 2.1.7, partes da 2.1.9, 2.1.10 e 2.1.12, foram elaboradas com base em Proakis e Manolakis (1996).

##### 2.1.1 Introdução

Um sinal é qualquer quantidade física que varie com o tempo, espaço ou qualquer outra variável (ou variáveis), sendo portanto uma descrição de como um parâmetro varia de acordo com outro parâmetro, por exemplo, a alteração de uma tensão em um circuito em função do tempo. Alguns sinais podem ser expressos matematicamente por meio de funções. Um sinal elétrico como o ECG (eletrocardiograma), é um exemplo cuja representação pode ser vista na Figura 2.



**Figura 2:** Um sinal de ECG

O sinal de ECG da Figura 2 corresponde a uma porção do conteúdo do arquivo *aami3a* de Moody, Mark e Goldberger (2001). Nesta figura, o eixo dos tempos demonstra que toda esta excursão do sinal se deu em aproximadamente um segundo.

Um sistema pode ser definido como sendo um dispositivo físico que pode ser um *software* ou um *firmware*<sup>1</sup>, que realize uma operação sobre um sinal. Um filtro que elimina ruído de um sinal, por exemplo, é um sistema.

Os sinais na natureza são contínuos e para que um *software* ou um *firmware* consiga processá-los, é necessário convertê-los para um formato digital. Esta conversão se dá através da coleta de instantes do sinal, normalmente a intervalos regulares de tempo. É a chamada amostragem. No sinal de ECG apresentado na Figura 2, por exemplo, foram ali apresentadas 1024 amostras. O número de amostras coletadas de um sinal será aqui simbolizado por  $N$ . Porém, para que a representação seja condizente com o sinal real, é necessário que seja realizada esta coleta com uma frequência denominada frequência de amostragem,  $F_s$ , a qual deverá igual ou superior ao dobro da máxima frequência existente no sinal que esteja sob amostragem. No exemplo do sinal de ECG, a frequência de amostragem utilizada foi de 720 Hz, o que fornece um período de amostragem  $T_s$  de  $1/720 \approx 1,39$  ms. Desta forma é possível serem determinados, a cada 1,39 ms, pontos no gráfico da Figura 2 para representarem os 1024 instantes do sinal de ECG. A frequência máxima típica de um sinal de ECG é aproximadamente 100 Hz. Logo o valor de  $F_s$  escolhido foi satisfatório.

Portanto, uma das formas de se representar um sinal em PDS é através de um vetor  $x(n)$ , uni ou multidimensional, onde cada componente do mesmo corresponde uma das amostras do sinal analógico coletadas durante o método da amostragem:

$$x(n) = [x(0), x(1), x(2), \dots, x(N-1)] \quad (1)$$

Onde:

- $x(n)$  é o sinal  $x(t)$  após o processo de amostragem; e
- $N$  é o número de amostras utilizado no processo da amostragem.

Um filtro digital é um tipo de sistema que processa um sinal recebido  $x(n)$  deixando passar ou atenuando determinadas componentes de frequência deste, fornecendo um sinal de saída  $y(n)$ , resultado deste processamento.

---

<sup>1</sup> *Firmware*: também conhecido como *software* embarcado, trata-se de um *software* que controla o *hardware* diretamente. É armazenado permanentemente em um chip de memória de *hardware*.

### 2.1.2 Filtros digitais

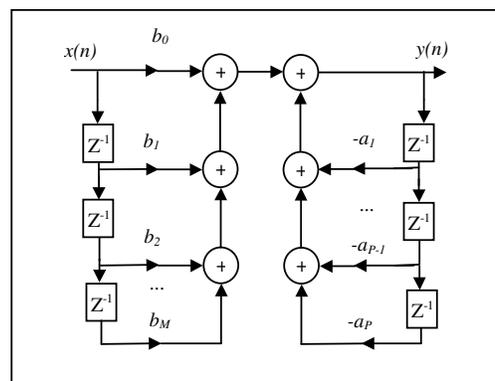
O sinal resultante de uma filtragem digital pode ser obtido através da seguinte equação:

$$y(n) = - \sum_{k=1}^P a_k y(n-k) + \sum_{k=0}^M b_k x(n-k) \quad (2)$$

Onde:

- $y(n)$  é o sinal  $x(n)$  filtrado;
- $a_k$  são as  $P$  constantes características que cada filtro digital tem. Estas constantes ponderam as amostras de instantes anteriores do sinal de saída, por exemplo,  $y(n-1)$  e  $y(n-2)$ , para formarem a amostra presente do sinal  $y(n)$ ; e
- $b_k$  são as  $M$  constantes características que cada filtro digital tem. Estas constantes ponderam as amostras de instantes anteriores do sinal de entrada, por exemplo,  $x(n-1)$  e  $x(n-2)$ , para formarem a amostra presente do sinal  $y(n)$ .

Uma das formas de ser implementada a equação (2) pode ser a Forma Direta I, apresentada na Figura 3:



**Figura 3:** A Forma Direta I.

Na Figura 3:

- as constantes  $a_1, a_2, \dots, a_P$  e  $b_0, b_1, b_2, \dots, b_M$  representam blocos multiplicadores do sinal e são representadas por pontas de setas no meio de cada linha. Quando uma

linha não possui uma ponta de seta no meio de seu caminho, significa que o sinal não está sendo multiplicado por nenhuma constante;

- b) os blocos que contêm um  $Z^{-1}$ , são responsáveis pela geração de um atraso na amostra do sinal, isto é, recebem, por exemplo,  $x(n-1)$  e entregam  $x(n-2)$ ;
- c) os blocos com um sinal de + representam os somadores do sinal; e
- d) o vetor do sinal  $x(n)$  é filtrado sendo passado por esta estrutura e sendo formado na saída desta o sinal filtrado  $y(n)$ .

### 2.1.3 Sistemas LIT

Dependendo do filtro, a obtenção das constantes  $a_k$  e  $b_k$  nem sempre é simples. Porém, há sistemas que são classificados como LITs (Linear e Invariável com o Tempo) e conseguem fornecer o sinal de resposta  $y(n)$  por um caminho alternativo ao da equação (2).

Um sistema é linear quando admite o princípio da superposição:

$$\begin{aligned}
 & \text{se} \\
 & y_1(n) = \mathfrak{F}[x_1(n)] \quad e \quad y_2(n) = \mathfrak{F}[x_2(n)] \\
 & \text{então} \\
 & y(n) = y_1(n) + y_2(n) = \mathfrak{F}[x_1(n) + x_2(n)]
 \end{aligned} \tag{3}$$

Onde:

- $x_1(n)$  e  $x_2(n)$  são dois sinais de entrada que são aplicados ao sistema linear;
- $y_1(n)$  e  $y_2(n)$  são os sinais de saída individuais correspondentes às entradas  $x_1(n)$  e  $x_2(n)$ , respectivamente;
- $\mathfrak{F}$  é a transformação matemática que o sistema aplica ao sinal de entrada para fornecer o sinal de saída; e
- $y(n)$  é, nesta equação, a saída do sistema a qual irá corresponder a uma entrada que seja a soma dos sinais  $x_1(n)$  e  $x_2(n)$ .

Um sistema é invariável com o tempo quando obedece à seguinte condição:

$$\begin{aligned}
 & \text{se} \\
 & y(n) = \mathfrak{F}[x(n)] \\
 & \text{então} \\
 & y(n-a) = \mathfrak{F}[x(n-a)]
 \end{aligned} \tag{4}$$

Onde:

- $y(n)$  é o sinal de resposta do sistema;
- $x(n)$  é o sinal de entrada do sistema;
- $\mathfrak{F}$  é a transformação matemática que o sistema linear aplica ao sinal de entrada para fornecer o sinal de saída; e
- $a$  é uma constante que é responsável pelo deslocamento no tempo do sinal.

Portanto, um sistema LIT é aquele que atende às equações (3) e (4) e pode ser representado pela sua resposta a um sinal chamado impulso unitário. Um impulso unitário é um sinal que atenda à seguinte equação:

$$\delta(n) = \begin{cases} 1, & n = 0 \\ 0, & n \neq 0 \end{cases} \tag{5}$$

Onde:

- $\delta(n)$  é o sinal impulso unitário.

Será chamado de  $h(n)$  o sinal de resposta de um sistema LIT a um impulso unitário.

Portanto:

$$\begin{aligned}
 & \text{se} \\
 & y(n) = \mathfrak{F}[x(n)] \\
 & \text{então} \\
 & y(n) = h(n) = \mathfrak{F}[\delta(n)]
 \end{aligned} \tag{6}$$

O caminho alternativo ao oferecido à equação (2) para a obtenção do sinal de saída  $y(n)$  com sistemas LIT é através da convolução entre o sinal de entrada e a resposta ao impulso unitário do filtro:

$$y(n) = x(n) * h(n) = \sum_{k=0}^{\infty} x(k)h(n-k) \tag{7}$$

Onde:

- $y(n)$  é o sinal filtrado pelo filtro LTI;
- $h(n)$  é a resposta ao impulso unitário do filtro digital LTI;
- $x(n)$  é o sinal de entrada do sistema; e
- $*$  é o operador matemático que representa a convolução entre dois sinais.

A simplificação da obtenção do sinal filtrado  $y(n)$  através da equação (7) em relação à equação (2) acontece devido ao seguinte fato: basta ser aplicado à entrada do filtro um impulso unitário e serem feitos os registros de  $h(n)$ ; em seguida, através da equação (7), é possível ser feita a convolução de  $x(n)$  com  $h(n)$  sendo obtido  $y(n)$ . Isto evita a busca pelos coeficientes  $a_k$  e  $b_k$  necessários diretamente através da equação (2).

#### 2.1.4 A Transformada Discreta de Fourier

Há a necessidade também de ser conhecida a resposta em frequência de um filtro digital, independentemente de qual sinal de entrada o mesmo esteja processando. Para isto, é possível ser utilizada a TDF (Transformada Discreta de Fourier):

$$H(k) = \sum_{n=0}^{N-1} h(n) \times e^{\left(\frac{-j2\pi}{N}\right) \times kn}, \quad k, n = 0, 1, 2, \dots, N-1 \quad (8)$$

Onde:

- $H(k)$  é o vetor de números complexos que expressa a resposta em frequência de um dado filtro LTI;
- $N$  é o número de pontos coletados durante o processo da amostragem;
- $k$  é um índice que varia entre zero a  $N-1$ ; e
- $h(n)$  é a resposta do filtro digital ao impulso unitário.

A equação (8) também pode ser utilizada para sinais quaisquer onde se queira saber o perfil de frequências dos mesmos. Para isto, basta substituir nesta equação  $h(n)$  por  $x(n)$

Porém, através da equação (8), somente será obtido um vetor complexo, indexado por  $k$ , que ainda não esclarece quais as regiões de frequências o filtro irá atenuar ou deixar passar.

Para ser atingido este objetivo, primeiramente estabelece-se uma relação entre a faixa de frequências que foi o alvo da filtragem e  $k$ , de acordo com:

$$f = \frac{k}{N-1} \quad (9)$$

Onde:

- $f$  é a frequência normalizada que varia de 0,0 a 1,0 e é medida em ciclos por amostra;
- $N$  é o número de pontos coletados durante o processo da amostragem;
- $k$  é um índice que varia entre zero a  $N-1$ .

A frequência normalizada expressa frequências quaisquer através da seguinte relação:

$$f = F / F_s \quad (10)$$

Onde:

- $f$  é a frequência normalizada que varia de 0,0 a 1,0 e é medida em ciclos por amostra;
- $F$  é a frequência a ser normalizada em Hz;
- $F_s$  é a frequência de amostragem em Hz.

É neste ponto que deve ser salientado o que foi citado anteriormente a respeito da frequência de amostragem, que é ser igual ou superior ao dobro da máxima frequência existente no sinal a ser processado. Com base nisto, é possível ser determinado que a faixa que contém informações relevantes sobre a atuação da resposta em frequência do filtro está compreendida entre 0,0 e 0,5.

O que ocorre entre 0,5 e 1,0 é a repetição do que foi apresentado. Uma explicação pode ser:

- a) a TDF pode ser vista também como uma combinação linear de senos e cossenos representativos do sinal. Como o formato de um sinal senoidal é do tipo  $A \times \text{sen}(2\pi f)$ , é possível serem extraídas as diversas contribuições de diferentes frequências que um sinal pode ter;
- b) as frequências destes termos em seno e co-seno também não deverão exceder à metade da frequência de amostragem, conforme citado anteriormente e indiretamente através da equação (9);

- c) logo, o que é apresentado de 0,5 a 1,0 é a repetição do conteúdo que vai de 0,0 a 0,5.

Ainda, para prover uma compreensão da resposta em frequência, há a necessidade do vetor complexo  $H(k)$  ser processado. Um caminho para isto poderia ser retornar à equação (8) e separar  $H(k)$  em um vetor real e outro imaginário:

$$H(f) = H_R(f) + jH_I(f) \quad (11)$$

Onde:

- $H(f)$  é o vetor complexo, indexado pela frequência normalizada e obtido através da equação (8);
- $f$  é a frequência normalizada que pode ser obtida a partir de  $k$  na equação (9);
- $H_R(f)$  é o vetor de números reais indexado pela frequência normalizada que representa a parte real de  $H(f)$ ;
- $H_I(f)$  é o vetor de números reais indexado pela frequência normalizada que representa a parte imaginária de  $H(f)$ ; e
- $j$  é o número imaginário  $(-1)^{1/2}$ .

$H_R(f)$  e  $H_I(f)$  podem ser calculados a partir de  $H(k)$  na equação (8) adaptada à fórmula de Euler<sup>2</sup>:

$$H_R(f) = \sum_{n=-\infty}^{\infty} h(n) \cos(2\pi f n) \quad (12)$$

$$H_I(f) = - \sum_{n=-\infty}^{\infty} h(n) \sen(2\pi f n) \quad (13)$$

A resposta em frequência do módulo e da fase pode ser obtida a partir de:

$$|H(f)| = \sqrt{H_R^2(f) + H_I^2(f)} \quad (14)$$

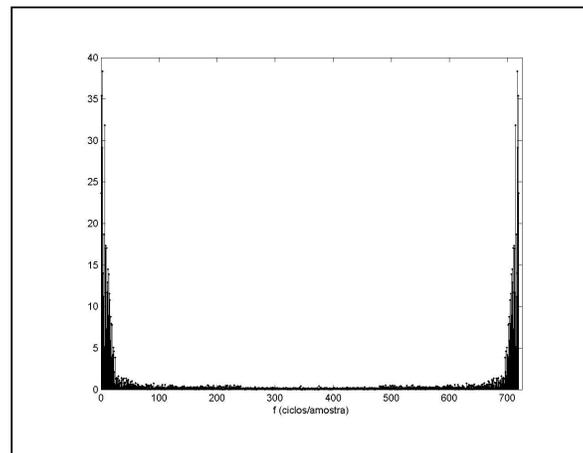
<sup>2</sup> Fórmula de Euler:  $e^x = \cos(x) + j \sen(x)$

$$\Theta(f) = \tan^{-1} \left( \frac{H_I(f)}{H_R(f)} \right) \quad (15)$$

Onde:

- $|H(f)|$  é módulo da resposta em frequência do filtro digital; e
- $\Theta(f)$  é resposta da fase de  $H(f)$  em função da frequência.

O sinal de ECG da Figura 2, por exemplo, possui um perfil de frequências,  $|X(f)|$ , obtido a partir da equação (14), como o que está apresentado na Figura 4:



**Figura 4:** TDF de um sinal de ECG.

Na Figura 2:

- foi apresentada a escala de frequências não normalizada, portanto, variando de zero a 700 Hz (que foi a frequência de amostragem adotada);
- a repetição de  $|X(f)|$  a partir dos 350 Hz já era prevista (repetição da TDF a partir de 0,5 devido à frequência máxima do sinal processado ser menor que a metade da frequência de amostragem); e
- É confirmado pela figura o dado informativo de que a componente de frequência máxima típica de um sinal de ECG não é superior aos 100 Hz.

As equações (14) e (15) costumam expressar vetores para as frequências entre 0,0 e 0,5 devido ao que foi exposto anteriormente nesta subseção (vide repetição de  $|X(f)|$  na Figura 4).

E finalmente, a partir das equações (14) e (15), é obtida a informação procurada: qual é a atuação do filtro digital em função da frequência sendo fornecida  $h(n)$ .

Algumas propriedades da TDF serão destacadas a seguir.

Uma propriedade dos sinais que trata relaciona a convolução com a TDF é:

$$y(n)=x(n) * h(n) \leftrightarrow Y(f) = X(f) \times H(f) \quad (16)$$

Outra propriedade da TDF que será destacada é:

$$\begin{aligned} \text{se} \quad & y_1(n)=x_1(n) * h(n) \\ \text{e} \quad & y_2(n)=x_2(n) * h(n), \\ \text{então} \quad & y(n)=y_1(n)+y_2(n)=[x_1(n)+x_2(n)] \times h(n) \end{aligned} \quad (17)$$

Há ainda duas propriedades que envolvem deslocamentos.

A propriedade do deslocamento no tempo:

$$x(n) \leftrightarrow X(f), \text{ assim como } x(n - k) \leftrightarrow e^{-j2\pi f k} X(f) \quad (18)$$

E a propriedade do deslocamento em frequência:

$$x(n) \leftrightarrow X(f), \text{ assim como } x(n) \times e^{j2\pi a n} \leftrightarrow X(f-a) \quad (19)$$

### 2.1.5 A Transformada -z

O produto entre números complexos pode ser obtido a partir do produto de seus módulos e da soma de seus argumentos. Com base nisto e na equação (16), pode surgir um problema. Se  $H(f)$  possuir uma fase que não for linear,  $y(n)$  poderá receber uma deformidade que não seja a desejada em algum tipo de aplicação. Portanto, para serem investigados e projetados filtros tendo-se a informação de que os mesmos tenham ou não uma fase linear, é utilizada uma outra ferramenta matemática em conjunto com a TDF, chamada Transformada-z:

$$H(z) = \sum_{n=-\infty}^{\infty} h(n)z^{-n} \quad (20)$$

Onde:

- $H(z)$  é a transformada-z de  $h(n)$ ; e
- $z$  é um número complexo.

A Transformada-z consiste de uma somatória de termos de uma progressão geométrica que quando converge, pode ser escrita em um formato total do tipo numerador-denominador:

$$H(z) = \frac{N(z)}{1+D(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \dots + b_{N_1} z^{-N_1}}{1 + a_1 z^{-1} + a_2 z^{-2} + \dots + a_{N_2} z^{-N_2}} \quad (21)$$

Onde:

- $N(z)$  é o polinômio em  $z$  que é encontrado para o numerador;
- $D(z)$  é o polinômio em  $z$  que é encontrado para o denominador;
- $b_1$  a  $b_{N_1}$  são os coeficientes do polinômio de  $N(z)$ ;
- $a_1$  a  $a_{N_2}$  são os coeficientes do polinômio de  $D(z)$ ;
- $N_1$  é o grau máximo do módulo do expoente de  $z$  em  $N(z)$ ; e
- $N_2$  é o grau máximo do módulo do expoente de  $z$  em  $D(z)$ .

Em (21), é dado o nome de zeros de  $H(z)$  às raízes de  $N(z)$  e pólos de  $H(z)$  às raízes de  $1+D(z)$ .

O número  $z$  é um número complexo pode ser expresso a partir de:

$$z = r e^{j2\pi f} \quad (22)$$

Onde:

- $r$  é um número real positivo.

Um sistema é chamado de causal quando depende somente de amostras, tanto da saída quanto da entrada, que sejam pertencentes ao estado presente ou a estados anteriores.

A equação (21) escrita na equação (20), permite determinar:

$$H(z)|_{z=r e^{j2\pi f}} = \sum_{n=-\infty}^{\infty} [h(n) r^{-n}] e^{-j2\pi f n} \quad (23)$$

Há uma relação, com base em (23), entre  $r$  e a viabilidade prática do filtro, isto é, se o mesmo é ou não causal: os pólos de  $H(z)$  devem estar contidos no interior de um círculo de raio  $r$  unitário no plano complexo  $z$ .

Filtros digitais causais que tenham todos os seus pólos na origem do plano  $z$  são chamados de filtros FIR. E filtros digitais causais que tenham pelo menos um pólo fora da origem são chamados de filtros IIR (*Infinite Impulse Response*).

Com relação aos filtros FIR:

- a) são também caracterizados por terem uma resposta em frequência linear de sua fase;
- b) como  $D(z) = 0$ , filtros FIR podem ser expressos por uma seqüência polinomial em  $z$ , tal polinômio apresenta uma simetria em relação ao seu centro e os termos em  $a_k$  na equação (2) tornam-se iguais a zero, assim como seus respectivos blocos na Figura 3.

Com base na resposta de fase, dada pela equação (15), é determinado que:

- a) Filtros FIR têm sua resposta de fase linear; e
- b) Filtros IIR têm sua resposta de fase não linear.

Portanto, fica desta forma estabelecido o apoio que a Transformada- $z$  oferece ao estudo dos filtros digitais: permite a avaliação direta dos pólos e zeros, permitindo sua classificação entre filtros FIR e IIR, além de outras finalidades, por exemplo, a determinação se o filtro FIR tem fase mínima (todos os zeros estão no interior do círculo unitário do plano- $z$ ).

### 2.1.6 Tipos de filtros FIR

A simetria dos filtros FIR, porém, os divide em quatro tipos, que são os de simetria positiva e negativa agrupados em um número de coeficientes par ou ímpar.

Os de simetria positiva atendem à seguinte relação:

$$h(n) = h(M - 1 - n) \quad (24)$$

Onde:

- $M$  é o número de coeficientes que o filtro FIR tem.

Os de simetria negativa atendem à seguinte relação:

$$h(n) = -h(M - 1 - n) \quad (25)$$

Para cada estes dois tipos, ainda há a possibilidade dos mesmos terem um número  $M$  par ou ímpar de coeficientes.

São encontrados ainda na literatura outros nomes para estes tipos de filtros:

- a) Tipo I para simetria positiva e  $M$  ímpar;
- b) Tipo II para simetria positiva e  $M$  par;
- c) Tipo III para simetria negativa e  $M$  ímpar;
- d) Tipo IV para simetria negativa e  $M$  par.

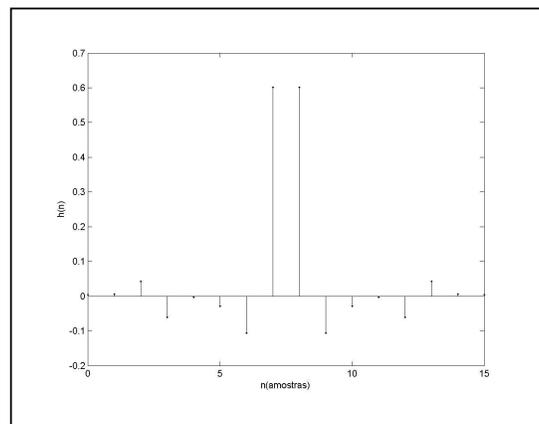
Filtros dos Tipos II ao IV possuem restrições em termos de sua resposta em frequência, que surgem da substituição das equações (24) e (25) na equação (8). Tais restrições são:

- a) Tipo II requer  $|H(f)|=0$  para  $f=0,5$ . Logo, filtros como os passa-baixas e os corta-faixa não podem ser projetados com este perfil;
- b) Tipo III requer  $|H(f)|=0$  para  $f=0,0$  e  $f=0,5$ . Logo, filtros do tipo passa-baixas, passa-altas e corta-faixa não podem ser projetados com este perfil; e
- c) Tipo IV requer  $|H(f)| = 0$  para  $f=0,0$ . Logo, filtros do tipo passa-baixas e corta-faixa não podem ser projetados com este perfil.

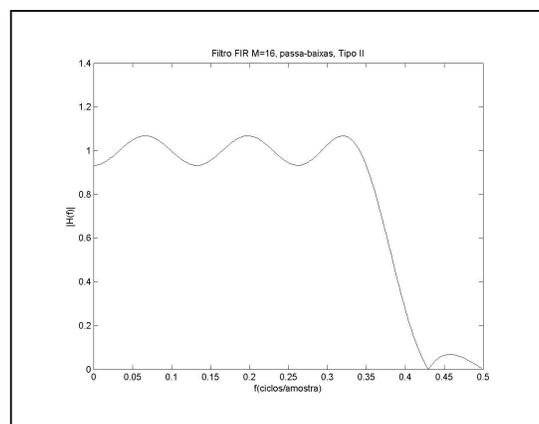
Filtros do Tipo III, para satisfazerem à equação (16) têm de ter sua amostra central igual a zero.

Portanto, o único tipo de filtro que não tem restrições quanto ao perfil de resposta em frequência a ser apresentada é o Tipo I.

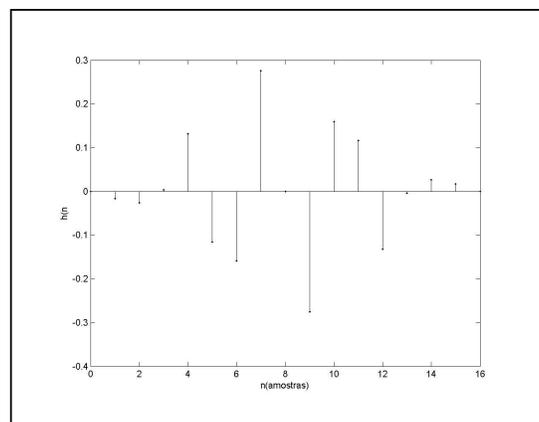
As Figuras 5 a 10 apresentam as respostas ao impulso unitário de exemplos de filtros Tipo II ao IV e suas respectivas respostas em frequência. Os gráficos das respostas em frequência já estão no formato de 0,0 a 0,5, como explicado na subseção 2.1.4.



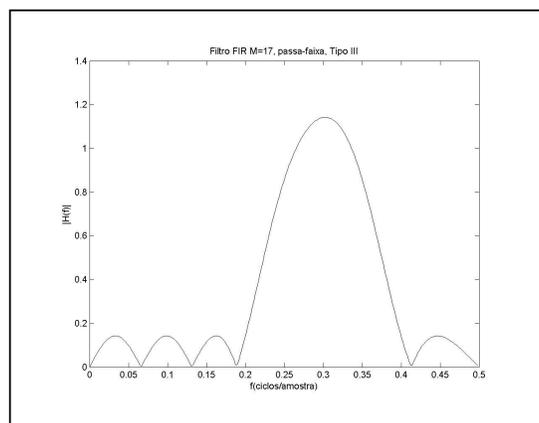
**Figura 5:**  $h(n)$  de um filtro FIR, Tipo II, com  $M=16$ .



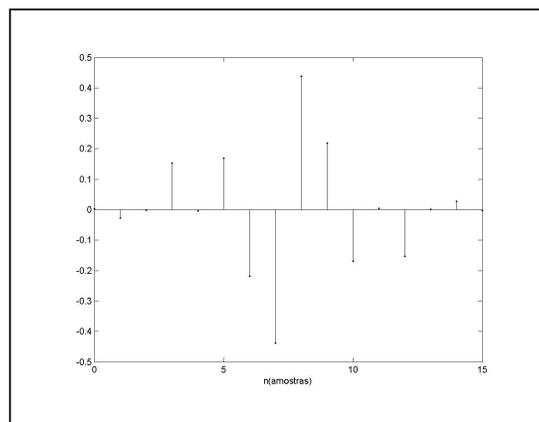
**Figura 6:**  $|H(f)|$  do filtro FIR da Figura 5.



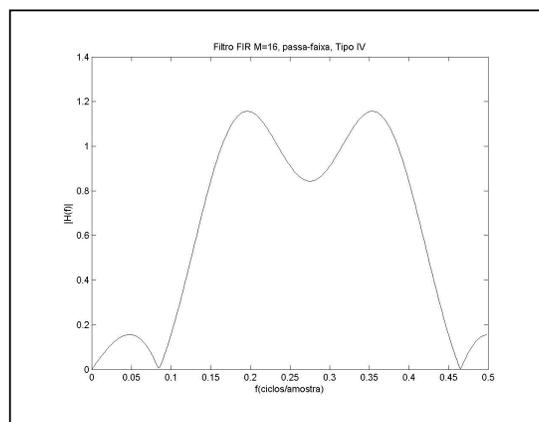
**Figura 7:**  $h(n)$  de um filtro FIR, Tipo III, com  $M=17$ .



**Figura 8:**  $|H(f)|$  do filtro FIR da Figura 7.



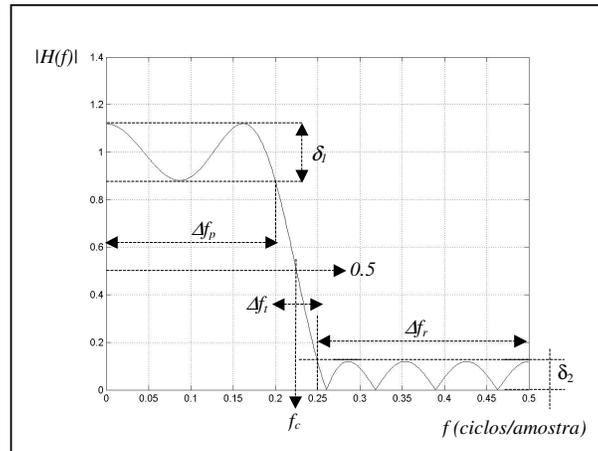
**Figura 9:**  $h(n)$  de um filtro FIR, Tipo IV, com  $M=14$ .



**Figura 10:**  $|H(f)|$  do filtro FIR da Figura 9.

### 2.1.7 Projeto de filtros FIR

A primeira etapa em um projeto de um filtro digital é a sua especificação, que é uma forma de serem fornecidos dados a respeito de pontos que devam ser atingidos pela resposta em frequência do filtro a ser projetado. Um exemplo pode ser observado na Figura 11.



**Figura 11:** Parâmetros de especificação de um filtro digital.

Estes parâmetros e outros podem também ser vistos na Tabela 1, com suas respectivas descrições. Os valores que podem ser alcançados pelo parâmetro chamado *ripple* atualmente são baixos. Devido a isto eles podem também ser encontrados na literatura expressos em decibéis através da relação  $20 \times \log(\delta_2)$  (SMITH, 1999).

Para métodos mais eficazes de projetos de filtros FIR, é costume não ser encontrado o valor da frequência de corte  $F_c$  e sim os valores da faixa passante e da faixa de rejeição, devido ao fato de que quando se informa simplesmente a  $F_c$ , é possível existir uma faixa passante e uma faixa de rejeição com valores que não possam ser aproveitados (devido a seus ganhos já não estarem valendo 1.0 ou 0.0).

Um exemplo de especificação de frequências pode ser  $F_s = 1000$  Hz, faixa passante de 0 a 200 e faixa de rejeição de 300 a 500 Hz. Efetuando a normalização estes valores passam para uma faixa passante que vai de  $fn_{11} = 0/1000 = 0,0$  ciclos/amostra até  $fn_{12} = 200/1000 = 0,2$  ciclos/amostra, uma faixa de rejeição que vai de  $fn_{21} = 300/1000 = 0,3$  ciclos/amostra a  $fn_{31} = 500/1000 = 0,5$  ciclos/amostra.

**Tabela 1:** Parâmetros de especificação de um filtro digital

<i>Parâmetro</i>	<i>Descrição</i>
M	Número de coeficientes
M-1	Ordem do filtro
$\Delta f_p$	Faixa passante
$\Delta f_r$	Faixa de rejeição
$\Delta f_t$	Faixa de transição
$\delta_1$	<i>Ripple</i> da faixa passante
$\delta_2$	<i>Ripple</i> da faixa de rejeição
F <sub>c</sub>	Frequência de corte
F <sub>s</sub>	Frequência de amostragem
f <sub>c</sub>	Frequência de corte normalizada

### 2.1.8 Projeto de filtros FIR pelo método de Parks-McClellan

Para a apresentação deste método, será tomado um filtro FIR Tipo I, ou seja, com simetria positiva e com um número de coeficientes  $M$  ímpar (PARKS e MCCLELLAN, 1972a). A sua TDF, a partir das equações (8) e (9), pode ser tomada como sendo:

$$H(f) = \sum_{n=-\infty}^{\infty} h(n) \times e^{-j2\pi f n} \quad (26)$$

Sendo aplicada a simetria dos filtros Tipo I e sendo considerado um número  $L$ , a equação (26) pode ser reescrita como sendo:

$$H(f) = \sum_{n=-L}^L h(n) \times e^{-j2\pi f n} \quad (27)$$

Onde:

- $L$  é um número tal que  $L=(M-1)/2$ .

Para um caso com um número ímpar de coeficientes,  $M=7$ , os coeficientes serão tais que  $h(-3)=h(3)$ ,  $h(-2)=h(2)$  e  $h(-1)=h(1)$ . Sendo aplicadas estas igualdades na equação (8) combinada com a (9), sendo colocados os termos em evidência através da fórmula de Euler e sendo feita uma generalização para casos quaisquer de simetria positiva e  $M$  ímpar, é obtida a seguinte equação:

$$H(f) = h(0) + \sum_{n=1}^L 2h(n) \times \cos(2\pi fn) \quad (28)$$

Esta equação pode ser generalizada para:

$$H(f) = \sum_{k=0}^L a_k \times [\cos(2\pi fn)]^k \quad (29)$$

Onde:

- $a_k$  é cada um dos coeficientes dos termos da combinação linear dos co-senos resultantes.

A equação (29) está no formato em que se pode aplicar a teoria da aproximação de Chebyshev (PRESS et. al., 2002), que basicamente consiste num procedimento matemático para encontrar uma combinação linear de co-senos que mais se aproxime de uma determinada curva que precise ser expressada numericamente por meio deste recurso. Através desta teoria, é aplicado o Teorema da Alternância, que basicamente consiste em ser determinado um polinômio de ordem  $L$ , que minimize o máximo erro existente em:

$$E(f) = W(f)[H_d(f) - H(f)] \quad (30)$$

Onde:

- $E(f)$  é um vetor que contém os erros, ponto a ponto, entre os valores dos vetores de resposta em frequência desejada e a que está sendo obtida;
- $W(f)$  é uma função do tipo peso;

- $H_d(f)$  é o vetor de resposta em frequência desejado (aquele que contém o formato da resposta em frequência que se quer);
- $H(f)$  é o vetor de resposta em frequência que o método numérico está fornecendo.

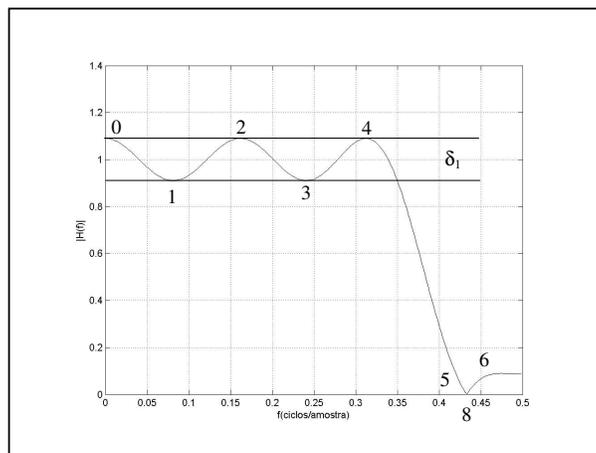
Quando o polinômio  $H(f)$  exibir  $L+2$  alternâncias, este será o que irá oferecer o erro máximo,  $\max[E(f)]$ , que terá o mínimo valor. Porém, há um subcaso, denominado filtro *extraripple* (PARKS e MCCLELLAN, 1972a), onde é possível também serem encontradas  $L+3$  alternâncias, dependendo das especificações do filtro, ou seja, não ocorre sempre. Portanto, o algoritmo de Parks-McClellan pode tanto oferecer  $L+2$  quanto  $L+3$  alternâncias.

Em Rabiner, McClellan e Parks (1975), há a generalização para os quatro casos. A Tabela 2 apresenta um resumo da relação que deve existir entre o número de coeficientes do filtro e seu número máximo de alternâncias. Os extremos, 0,0 e 0,5, não devem ser contados.

**Tabela 2:** Relação entre nº de alternâncias  $N_e$  e nº de coeficientes  $M$

<i>Tipo de filtro</i>	<i>Relação com nº de alternâncias</i>
I	$N_e \leq (M+1)/2$
II	$N_e \leq M/2$
III	$N_e \leq (M-1)/2$
IV	$N_e \leq M/2$

Por exemplo: um filtro com 13 coeficientes, uma faixa passante  $\Delta f_p$  de 0,0 a 0,35 e uma faixa de rejeição  $\Delta f_r$  de 0,42 a 0,5, gerado a partir do MPM, pode ser visto na Figura 12.



**Figura 12:** Alternâncias em filtros gerados pelo MPM.

Na Figura 12:

- a) há seis alternâncias, sendo que a sexta não foi contada em  $f=0,5$ , mas sim antes, no intervalo  $0,45 < f < 0,5$ ;
- b) a relação para filtro Tipo I da Tabela 2 foi atendida, pois  $6 \leq (13+1)/2$ ;
- c) foi enfatizado o ripple  $\delta_1$  para demonstrar que filtros obtidos pelo MPM oferecem ripples de iguais amplitudes, sejam nas faixas passante ou de rejeição;

Os exemplos das Figuras 6, 8 e 10 foram também obtidos pelo MPM e podem confirmar a relação apresentada na Tabela 2.

Os valores dos ripples  $\delta_1$  e  $\delta_2$  podem ainda ter seus valores individualizados por meio da função-peso  $W(f)$  da equação (30).

O algoritmo de Remez (1957) adaptado para a síntese de filtros FIR por Parks e McClellan consiste basicamente nos seguintes passos:

- a) um conjunto  $L+2$  pontos de frequências candidatas a serem os pontos extremos procurados para serem o polinômio de Chebyshev é aleatoriamente estabelecido;
- b) um valor chamado de  $\rho$  é calculado a partir destes pontos;
- c) tal conjunto de pontos determina uma curva através de uma interpolação de Lagrange específica;
- d) a curva resultante desta interpolação é utilizada para calcular o erro entre a mesma e a curva desejada de acordo com a equação (30) e tal erro deverá ser menor ou igual a  $\rho$ ;
- e) se existirem  $L+2$  ou  $L+3$  pontos extremos, os pontos deverão ser testados se mudaram de posição em relação à iteração anterior;
- f) se os pontos não mudaram, a solução foi encontrada;
- g) se os pontos mudaram, há um retorno à etapa *b* do algoritmo e é feita a repetição do processo até serem encontrados pontos que não mudem de valor na etapa *e*.

A apresentação e descrição do método estão em Parks e McClellan (1972a) e o fluxograma com o código em FORTRAN (*Formula Translation*) estão em Parks e McClellan (1972b).

A relação entre os nomes envolvidos neste método é a que se segue:

- a) Pafnuty Lvovich Chebyshev foi um matemático russo que viveu entre 1821 a 1894 e criou a teoria dos mínimos e máximos, com ou sem pesos, de aproximações matemáticas a partir de seu polinômio, que se baseia em uma combinação linear de co-senos;
- b) Evgeny Yakovlevich Remez (1957), outro matemático russo que viveu entre 1896 e 1975, desenvolveu dois algoritmos de aproximação por Chebyshev, um chamado “Primeiro Algoritmo de Remez” e o outro “Segundo Algoritmo de Remez”. O primeiro desempenha a aproximação de Chebyshev por trocas simples dos pontos e o segundo por múltiplas trocas de pontos;
- c) Thomas Parks passou para seu então aluno de PhD, James McClellan um trabalho de turma de verão, que era adaptar o trabalho de Remez para a síntese de filtros FIR (MCCLELLAN e PARKS, 2005). Em tal trabalho, foi tomado como base o Segundo Algoritmo de Remez e nele feitas três modificações. A primeira foi a de converter um intervalo genérico onde a aproximação era calculada,  $[a, b]$ , para o intervalo de frequências  $[0,0; 0,5]$ . A segunda foi a habilitação para a movimentação dos prováveis pontos de extremos que correspondam às frequências onde ocorram as alternâncias. E a terceira foi a utilização de uma interpolação de Lagrange denominada Baricêntrica (BERRUT e TREFETHEN, 2004) para unir os pontos. Esta interpolação tem a finalidade de prover um melhor formato da curva para filtros com altas ordens, quando comparada com a interpolação de Lagrange convencional.

Para que o método também contemplasse os outros tipos de filtros, com suas ressalvas em termos de resposta em frequência em função do número de coeficientes e simetria escolhida, um arranjo trigonométrico foi necessário para que todos os casos pudessem recair sobre um formato do polinômio de Chebyshev análogo ao obtido na equação (29). Para tratar dos filtros de simetria negativa, foi utilizado o conceito dos filtros de Hilbert (que mantém o ganho unitário em toda a faixa e invertem a fase a partir  $f > 0,5$ ). Esta generalização foi feita em McClellan e Parks (1973) e a adaptação no fluxograma e no programa, já com a colaboração do Professor Lawrence Rabiner, em McClellan, Parks e Rabiner (1973). Após isto, em Rabiner, McClellan e Parks (1975) foi apresentado um trabalho no qual eram detalhados aspectos do método e trabalhos correlatos, dentre eles a fórmula de Herrmann,

Rabiner e Chan (1973) para previsão do número  $M$  de coeficientes a partir dos *ripples*  $\delta_1$  e  $\delta_2$  e da faixa de transição  $\Delta f_i$ :

$$M \cong \frac{D_\infty(\delta_1, \delta_2) - f(\delta_1, \delta_2) \times (\Delta f_i)^2}{\Delta f_i} + 1$$

sendo

$$D_\infty(\delta_1, \delta_2) = [0,005309 \times (\log \delta_1)^2 + 0,07114 \times (\log \delta_1) - 0,4761] \times (\log \delta_2) - [0,00266 \times (\log \delta_1)^2 + 0,5491 \times (\log \delta_1) + 0,4278]$$

$$e$$

$$f(\delta_1, \delta_2) = 11,012 + 0,51244 \times (\log \delta_1 - \log \delta_2)$$
(31)

Onde:

- $\delta_1$  e  $\delta_2$  são os *ripples* da faixa passante e da faixa de rejeição, respectivamente; e
- $\Delta f_i$  é a frequência de transição, apresentada na Figura 11.

O MPM foi o método mais difundido, principalmente pelos seguintes fatores:

- a) ser possível a obtenção de um filtro que atenda rigorosamente às especificações de faixa passante ( $\Delta f_p$ ), faixa de rejeição ( $\Delta f_r$ ) e *ripples* ( $\delta_1$  e  $\delta_2$ ), desde que seja empregado o  $M$  calculado previamente pela equação (31);
- b) ter seu detalhamento público tanto do método quanto de sua implementação (fluxograma e código); e
- c) o código-fonte ser simples (foi um código em FORTRAN, feito em algumas semanas, composto somente por atribuições, estruturas de decisão e *gotos*).

Pela trajetória histórica apresentada, o MPM também é encontrado com os seguintes nomes: Método de Rabiner, Parks e McClellan (ou estes mesmos nomes em seqüências alternadas), Algoritmo de Remez ou ainda um dos algoritmos de aproximação por Chebyshev.

### 2.1.9 Projeto de filtros FIR por outros métodos

O método dos mínimos quadrados: segundo Lang (1999), enquanto o método de Parks e McClellan procura minimizar o máximo erro existente no vetor  $E(f)$  da equação (30), o método dos mínimos quadrados procura minimizar a energia do erro deste mesmo vetor:

$$E(f) = \int_{f \in [0, 2\pi]} |W(f) \times [H_d(f) - H(f)]|^2 df \quad (32)$$

Os termos desta expressão são os mesmos que os da equação (30).

Há ainda outros métodos como o das janelas e o da amostragem de frequência, ambos sem controle sobre a faixa passante e a de rejeição, não podendo ter seus formatos previstos.

### 2.1.10 A transformada rápida de Fourier

A equação (8) pode ser reescrita, fornecendo:

$$H(k) = \sum_{n=0}^{N-1} h(n) \times W_N^{kn} \quad (33)$$

Onde:

$$W_N = e^{-\frac{j2\pi}{N}} \quad (34)$$

Os demais termos desta equação são os mesmos descritos para a equação (8).

Ao ser desenvolvida a equação (33), aparecem  $N^2$  multiplicações para um conjunto de  $N$  pontos. Estas multiplicações podem ser reduzidas através de duas propriedades da TDF, que são a periodicidade:

$$W_N^{k+N} = W_N^k \quad (35)$$

E a simetria:

$$W_N^{k+\frac{N}{2}} = -W_N^k \quad (36)$$

Baseado nestas propriedades surge um conjunto de algoritmos que foi chamado de TRF (Transformada Rápida de Fourier). Em sua maioria, estes algoritmos se baseiam na fatoração do número  $N$  em números menores, não obrigatoriamente primos. Um caso

particular é aquele em que assume  $N$  como sendo sempre uma potência de dois. Para este caso, há uma redução de  $N^2$  multiplicações para  $N/2 \times \log_2 N$ . Há ainda outros métodos que conseguem prover uma redução maior do número de operações, por exemplo, *radix-4*, *radix-8* e *split-radix*.

### 2.1.11 Filtros digitais sem multiplicação

A partir de um sistema implementado em uma estrutura como a Forma Direta I da Figura 3, o sinal  $x(n)$  ao ser processado por um sistema  $h(n)$  com  $M$  coeficientes  $a$  e  $P$  coeficientes  $b$ , gerará uma saída  $y(n)$ , que passará por um número de multiplicações igual a  $M+N+1$ .

Uma multiplicação em *hardware* entre dois números de ponto flutuante consome tempo de processamento e com um  $M$  e  $N$  altos, o tempo de resposta que o sistema levará para gerar  $y(n)$ , poderá comprometer sua viabilidade em um projeto.

Visando resolver este problema, um artifício foi empregado e encontra-se disponível na literatura da área. Em Ye e Chang (2004), por exemplo, encontra-se uma apresentação deste recurso: para filtros FIR, com base na Figura 3, com os termos  $b_i$  iguais a zero (por ser FIR), os valores de  $a_i$ , que são os coeficientes do sistema  $h(n)$  responsáveis pelas multiplicações, podem ter seus valores restritos a expressarem somente potências de dois, POT (*Power of Two*).

Um número em ponto flutuante, ao ser multiplicado por um outro que seja POT, é simplesmente repetido com seus *bits* deslocados para a esquerda ou para a direita a depender isto do expoente da potência de dois, se for positivo ou negativo, respectivamente. Desta forma o número de multiplicações existentes será substituído por um igual número de deslocamentos de *bits*, o que oferece agilidade ao processamento. Tais filtros, desta forma implementados, são chamados de filtros sem multiplicações (ou do inglês *Multiplierless Filters*).

### 2.1.12 Quantização dos coeficientes na implementação

Um problema pode ocorrer durante a implementação dos filtros FIR por meio dos métodos clássicos: uma vez obtidos os coeficientes de um filtro FIR que melhor se aproxime das especificações desejadas por meio de algum método numérico, é chegada a hora de tais coeficientes serem colocados em algum *firmware* na fase da implementação do filtro. Porém, nem sempre há 15 casas decimais destinadas a cada coeficiente, como normalmente é calculado. Devido a isto, os coeficientes são arredondados para valores com menos casas decimais. Porém isto ocasiona o deslocamento dos pólos e zeros do filtro, podendo distorcer a resposta em frequência real.

## 2.2 ALGORITMOS GENÉTICOS

Os processos naturais que regem os seres vivos, como evolução, reprodução, extinção, entre outros mecanismos, assim como a dinâmica dos organismos vivos e da natureza em geral, tornaram-se fonte de inspiração para estudos, entre os quais, aqueles que buscam imitar-lhes. Cientistas procuraram estudar algoritmos genéticos para resolver problemas computacionais complexos, os quais não vinham sendo resolvidos de uma forma satisfatória por outros métodos.

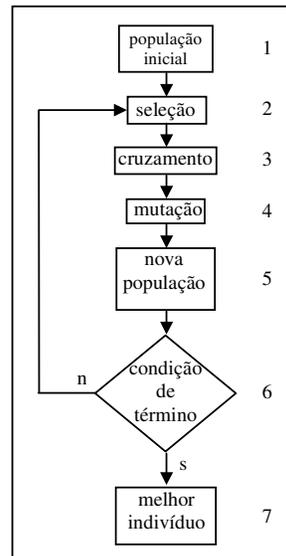
Segundo Tanomaru (1995), processos como o anelamento que ocorre nos metais, deu origem à área da ciência que adota este paradigma, denominada AS (Anelamento Simulado) (KIRKPATRICK, GELATT e VECCHI, 1983), o estudo do sistema nervoso central dos animais superiores deu origem às RNAs (Redes Neurais Artificiais) (BARRETO, 2004) e o processo estocástico da evolução natural deu origem à CE (Computação Evolucionária) (BÄCK, HAMMEL e SCHWEFEL, 1997), a qual um de seus ramos são os AGs.

Dentro da CE há ainda as seguintes áreas: PE (Programação Evolucionária) (BÄCK, ULRICH E HAMMEL, 1997), EE (Estratégias Evolucionárias) (BÄCK, HAMMEL e SCHWEFEL, 1997), AMs (Algoritmos Meméticos) (KRASGONOR, 2002), ED (Evolução Diferencial) (STORN e PRICE, 1995), HE (*Hardware* Evolutivo) (HOUNSELL, ARSLAN e THOMSON, 2004), SC (Sistemas Classificadores) (BÄCK, HAMMEL e SCHWEFEL, 1997) e PG (Programação Genética) (KOZA, 1992).

A CE está enquadrada dentro de uma grande área chamada IC (Inteligência Computacional), que compreende outras áreas, como por exemplo, RNA, AIS (*Artificial Immune Systems*) (DASGUPTA, 1999), PSO (*Particle Swarm Optimization*) (SHI, 2004) e Sistemas Fuzzy (ZADEH, 1965).

### 2.2.1 Introdução

Um AG baseia-se em uma seqüência de ações, que podem ser representadas pela Figura 13:



**Figura 13:** Fluxo básico de um AG.

Um detalhamento do AG apresentado nesta figura, segundo uma interpretação feita a partir de Goldberg (1989) e Tanomaru (1995) pode ser a que se segue:

- a) uma possível solução de um problema, isto é, um candidato-resposta, é representado como sendo um indivíduo. Normalmente o indivíduo não é representado pelos números reais ou inteiros do problema a ser otimizado. Este conjunto de números do mundo real, que representa a informação numérica do problema em si, é chamado de fenótipo. Um possível alfabeto no qual um indivíduo possa ser codificado é o binário. É dado o nome de genótipo a um valor específico de uma palavra binária que represente um indivíduo. Um exemplo de um indivíduo codificado por meio de genótipos pode ser uma cadeia de *bits*, com os quatro primeiros *bits* representando uma resposta à pergunta 1 do problema, os seis *bits* que se seguem representando uma resposta à pergunta 2 e assim sucessivamente. A todo o indivíduo (aos 20 *bits* do exemplo), é dado o nome de cromossomo. Ao conjunto de bits que representem um fenótipo é dado o nome de gene. No caso do exemplo um gene seria os quatro primeiros bits, outro gene os seis seguintes e assim

por diante. Esta etapa, denominada representação cromossômica, antecede à execução do AG e faz parte da modelagem do problema;

- b) um conjunto de indivíduos é gerado aleatoriamente (ou a partir de algum critério inicial) formando o que se chama de uma população. Normalmente, esta população tem um tamanho,  $\lambda$ , fixo. Em AGs, é chamado de espaço de busca,  $S$ , ao conjunto de todas as possíveis combinações de soluções de genótipos considerados como satisfatoriamente representativos. A relação entre os fenótipos e seus respectivos genótipos dependerá da codificação adotada e pode não ser de um para um. Esta é a etapa um na Figura 13;
- c) uma avaliação inicial de cada indivíduo da população é feita de acordo com a resposta que se está procurando. Quanto mais próximos da resposta, maiores serão as notas dos indivíduos em tal avaliação. Dá-se o nome de *fitness* a esta medida e este cálculo normalmente é feito em função do cromossomo decodificado de cada indivíduo. Esta ação ainda faz parte da etapa um na Figura 13;
- d) é aplicado sobre a população algum mecanismo de seleção, visando a exclusão da próxima etapa do processo os indivíduos com valores de *fitness* baixos. Um exemplo de seleção pode ser basicamente o ato de serem tomados aleatoriamente um par de indivíduos da  $P$  e submetê-los aos processos de cruzamento e mutação. Esta é a etapa dois na Figura 13;
- e) em seguida é aplicado, com uma probabilidade *pcross*, um cruzamento (ou do inglês *crossover*) sobre uma dupla de indivíduos selecionados, ou seja, tal dupla poderá ou não efetuar o cruzamento, que consiste na geração de dois indivíduos, chamados descendentes, com base na troca de bagagem genética entre os indivíduos que foram selecionados no item *d*. Este é a etapa três na Figura 13;
- f) após isto, os descendentes são submetidos a uma mutação, com uma probabilidade *pmutation*. A mutação consiste basicamente em ser aplicado uma mudança aleatória de um ou mais *loci* do cromossomo. Este é a etapa 4 na Figura 13;
- g) concluídas as etapas *d*, *e* e *f* os descendentes gerados são destinados à nova população. Os indivíduos desta população irão passar neste estágio por uma avaliação. Esta é a etapa cinco na Figura 13;

- h) chama-se de geração ao ciclo compreendido entre as etapas *d* à *g*, inclusive. Concluída uma geração, é feito um teste se a condição de término foi atendida. A condição de término pode ser, por exemplo, um valor buscado de *fitness*, um determinado número máximo de gerações. Esta é a etapa seis na Figura 13;
- i) se a condição de término não foi atendida, há um retorno à etapa que corresponde ao item d e um novo ciclo de geração é executado. Esta ação corresponde à seta que conecta o bloco 6 ao 2 na Figura 13;
- j) se a condição de término foi atendida, há a passagem para o próximo bloco, o sete, que é a determinação do indivíduo pertencente à última geração tenha a maior *fitness*.

Há variações deste modelo.

Segundo Beasley, Bull e Martin (1993), há problemas de AGs que não enfocam o valor do gene e sim seu possível *locus* dentro do cromossomo, isto é, os genes permanecem com seus alelos fixos. Tais problemas são chamados de problemas de análise combinatória ou de ordenação. Um exemplo de problema deste tipo é o do Caixeiro Viajante (TANOMARU, 1995).

### 2.2.2 Outros métodos numéricos e aplicabilidade

Os algoritmos que utilizam o paradigma da CE são chamados de AEs (Algoritmos Evolucionários) e estão enquadrados dentro dos métodos heurísticos baseados em transições probabilísticas. Há outros métodos, como se pode ver na Tabela 3 (GOLDBERG, 1989). Nesta tabela, também estão resumidos os perfis dos demais métodos de busca e otimização e suas deficiências. Foi atendendo a estas deficiências, que os AEs conquistaram espaço, mesmo fornecendo uma resposta quase ótima na maioria das vezes.

**Tabela 3:** Perfil dos métodos, suas deficiências e características

<i>Método</i>	<i>Descrição</i>	<i>Deficiência</i>
Enumerativo	Busca Sequencial	Lento, não opera com $S$ grande
Baseado em Cálculo Numérico	Gradiente/Simplex Newton	Dificuldades com $f(x)$ multimodal e $S$ descontinuado
Analítico	Cálculo Analítico (derivadas)	Exige que $f(x)$ seja Derivável
Aleatório	Baseado na Sorte	Pode encontrar ou não a resposta
Baseados em Transições Probabilísticas	Métodos de CE	Pode encontrar a solução quase ótima

### 2.2.3 Representação cromossômica

A interpretação a seguir tomou como base as seguintes obras: Goldberg (1989), Tanomaru (1995) e Bäck, Hammel e Schwefel (1997).

Para se representar um conjunto de números com uma cadeia de *bits*, tendo sido estabelecidas as faixas dos valores válidos mínimo  $Val_{mín}$  e máximo  $Val_{máx}$ , assim como sua precisão  $prc$ , pode-se empregar a fórmula a seguir para determinar-se o tamanho  $l$  do cromossomo (IZIDORO, 2005):

$$prc = \frac{Val_{máx} - Val_{mín}}{2^l - 1} \quad (37)$$

#### 2.2.4 Função-objetivo, função-penalidade e função de *fitness*

Além das obras da subseção anterior foi tomada também como base para esta subseção a obra de Silva (2002).

A função-objetivo é o cálculo matemático que expressa o quão perto do ótimo está um indivíduo. A função-penalidade opera atribuindo decréscimos à função-objetivo para diferentes casos, por exemplo, indivíduos que não representem um fenótipo válido (seja um número pertencente a uma faixa inválida). E a função de *fitness*, é o valor que é destinado ao AG e, por este motivo, normalmente procura por definição ser sempre positiva e estar dentro de uma faixa de valores controlados, como 0,0 a 1,0.

Portanto, as funções-objetivo e função-penalidade operam em conjunto para atribuírem valores para a função de *fitness*.

#### 2.2.5 Convergência prematura, diversidade genética e pressão seletiva

Um indivíduo que esteja com uma *fitness* ligeiramente acima da média da população, poderá espalhar seu material genético para as próximas gerações. Porém, se tal material genético for o caminho da convergência para um máximo local, pouco poderá ser feito a partir deste ponto. Em um AG que esteja sendo executado nas suas primeiras gerações, tal indivíduo, caso possua uma *fitness* acima dos demais, poderá conduzir a população a uma convergência prematura.

Diversidade genética pode ser entendida como o grau de dissimilaridade entre indivíduos.

Pressão seletiva pode ser entendida como a consequência da discrepância entre os valores de *fitness* dos indivíduos da população atual.

#### 2.2.6 Métodos de seleção

A seleção proporcional (GOLDBERG, 1989) pode ser interpretada como sendo a escolha dos indivíduos através do giro de uma roleta: cada indivíduo está encaixado no seu local dentro da roleta e o tamanho deste local corresponde ao valor individual de *fitness* que cada um tem.

É dado um giro na roleta e no ponto em que a mesma parar, é selecionado o indivíduo que estiver naquele ponto. Porém, o indivíduo não sai da roleta, sendo colocado uma cópia dele em  $P_i$ . São repetidos os giros até serem completados um número de  $\lambda$  (número de indivíduos da população  $P$ ) repetições. Logo,  $P_i$  poderá estar constituída mais com elementos de alta *fitness* do que de indivíduos com *fitness* menores. Por este motivo, métodos que tenham como base este mecanismo, são chamados de métodos de seleção baseados em roleta. A equação a seguir indica como as probabilidades (os espaços na roleta) são determinadas:

$$p_{sel}(s) = \frac{f(s)}{\sum_{i=1}^N f(s_i)} \quad (38)$$

Onde:

- $p_{sel}(s)$  é a probabilidade de seleção do indivíduo  $s$ ;
- $f(s)$  é o valor de *fitness* do indivíduo  $s$ ; e
- o denominador corresponde à somatória dos valores de *fitness* de todos os indivíduos da população.

Outro método de seleção é o da Amostragem Estocástica Universal (do inglês *Stochastic Universal Sampling*) (BAKER, 1987), que será apresentado a partir de uma interpretação feita com base em Chipfield et al. (1994) comparando-o com o método da roleta:

- a) os indivíduos são acomodados na roleta, cada um dentro de um espaço determinado pela equação (38);
- b) a roleta é girada e ao parar, um ponto de retirada é estabelecido;
- c) são somados a este ponto de retirada, mais outros pontos de retirada que são equidistantes até ser completada uma volta sobre a roleta; e
- d) são repetidas as etapas *b* e *c* até o processo estar concluído.

Há ainda outros métodos de seleção como o Resto Estocástico (TANOMARU, 1995), Torneio Estocástico e Ordenação Linear (SILVA, 2002). Tais métodos podem passar por técnicas que atuam sobre a seleção como o Elitismo e o *crowding* (BEASLEY, BULL E MARTIN, 1993).

### 2.2.7 Operadores de cruzamento

No cruzamento de um ponto:

- a) é determinada aleatoriamente uma determinada posição de um gene nos dois cromossomos que farão o cruzamento;
- b) são gerados dois cromossomos descendentes cujos genes serão uma cópia de seus cromossomos ascendentes, exceto a partir do *locus* determinado, que deste *locus* até o final do cromossomo descendente, terá o conteúdo de seus genes composto pelo outro cromossomo ascendente.

No cruzamento multiponto:

- a) são determinadas  $m$  faixas para segmentar os cromossomos ascendentes  $A1$  e  $A2$ , respectivamente;
- b) são gerados os cromossomos descendentes,  $D1$  e  $D2$  respectivamente, a partir da cópia dos alelos de seus respectivos ascendentes;
- c) são copiados em  $D1$  e  $D2$ , a partir da segunda faixa, os alelos de  $A2$  e  $A1$  respectivamente;
- d) são mantidos em  $D1$  e  $D2$ , para a faixa que se segue, o conteúdo original da etapa  $b$ ;
- e) são repetidas as etapas  $c$  e  $d$  até o final do comprimento dos cromossomos.

Os dois cruzamentos citados e outros cruzamentos, como o cruzamento uniforme, podem ser encontrados em Goldberg (1989). Os cruzamentos são os operadores de recombinação responsáveis pela busca local (nas vizinhanças próximas) no espaço de busca.

### 2.2.8 Operadores de mutação

Segundo Goldberg (1989), a mutação altera, com uma probabilidade  $p_{mutation}$ , o alelo de um cromossomo. É importante ressaltar que o operador de mutação atua sobre cada gene e devido a isto, um cromossomo pode sair de uma mutação com mais de um gene alterado. O operador que desempenha esta operação é o chamado operador de mutação *bit a bit* (GOODMAN, 1996). Com base nisto, por exemplo, um cromossomo descendente que

passa por uma mutação bit a bit, pode entrar 0100110 e sair 0101110 (somente o gene do *locus* 3 foi alterado), 1100010 (os genes dos *loci* 0 e 4 foram alterados), etc.

Segundo Goodman (1996) e as linhas de comentários do arquivo *multimut.c* de Goodman e Punch (2002), foi criado também um outro operador de mutação denominado operador de mutação de múltiplos campos (do inglês *Multiple Field Mutation Operator*), o qual atua não somente sobre um gene, mas sim sobre uma faixa de genes adjacentes, isto é, um determinado número de *loci*. A faixa de genes adjacentes que pode sofrer esta mutação é estipulada e pode ir de um até *avg\_mut\_width* (com comprimento máximo igual ao tamanho do cromossomo), que é um parâmetro determinado por quem esteja utilizando este operador. O número de *loci* a serem alterados é determinado de acordo com a distribuição de Poisson utilizada como uma aproximação à distribuição Normal (MEIER, 1983), a partir de uma probabilidade  $1/avg\_mut\_width$ . Para o mesmo cromossomo que foi usado de exemplo no parágrafo anterior, um cromossomo resultante possível neste caso poderia ser 1010110 (genes dos *loci* 0 ao 2, com *avg\_mut\_width*=5, isto é, não é preciso que 5 genes sejam alterados).

A mutação é responsável pela busca global dentro do Espaço de busca.

### 2.2.9 Escalonamentos de *fitness*

Há ferramentas de AGs que aceitam receber os valores de *fitness* brutos, podendo excursionar em faixas além do 0,0 a 1,0, assim como receber valores negativos. Para tal, estas ferramentas desempenham escalonamentos nos valores de tais *fitness*. São formas de escalonamentos: a) o linear; b) o por janela; c) por corte  $\sigma$ ; e d) de Boltzmann.

Os escalonamentos dos itens *a* ao *c* podem ser encontrados em Tanomaru (1995) e o *d* em Goodman (1996).

## 2.3 REVISÃO DA LITERATURA

Não foram encontrados *softwares* automatizados, disponibilizados de forma gratuita ou comercializáveis, para o projeto de filtros FIR por meio de AGs, que possam ser utilizados por usuários que desconheçam o paradigma da CE.

Dentro desta seção encontram-se alguns métodos de projetos de filtros FIR que utilizam AGs e outros métodos, com a finalidade de servirem para mostrar um estado da arte que possa ser traçado em paralelo à investigação de *softwares* prontos que desempenhem a geração de filtros FIR.

### 2.3.1 Trabalhos correlatos

O primeiro registro localizado de um trabalho de projetos de filtros digitais usando AGs, foi o de Etter, Hicks e Cho (1982), onde é apresentado um sistema de identificação no qual a busca pelos coeficientes do filtro digital que deve ser encontrado é por AGs. O filtro estudado é do tipo IIR.

Para se apresentar o primeiro trabalho específico de projeto de filtros FIR, é necessário ser apresentado primeiramente o trabalho de Wade, Van-Eetvelt e Darwen (1990), o qual propôs o seguinte:

- a) uma biblioteca de seções de estruturas, parecidas com a Forma Direta II, de filtros básicas era pré-determinada; cada uma das estruturas já era pré-otimizada para não terem multiplicações e somente deslocamentos e somas;
- b) quando do recebimento de uma nova especificação, um método numérico buscava uma combinação linear da melhor disposição em cascata destas estruturas que atendessem à resposta em frequência desejada.

O primeiro registro localizado e específico sobre projeto filtros FIR, foi este mesmo trabalho, porém com um AG ao invés do método numérico que estava sendo usado, devido terem ocorrido problemas com altos tempos de execução (SUCLEY, 1991). Como se pode observar, o grau de hibridismo deste trabalho é alto e direcionado para serem construídos filtros sempre a partir de uma biblioteca de estruturas de filtros fixa.

Seguidas a este primeiro trabalho, durante o ano de 1993, surgiu uma série de pesquisas relatadas, tendo inclusive ocorrido uma conferência sobre processos naturais

aplicados à área de PDS. Uma pesquisa desta época foi a de Cemes e Ait-Boudaoud (1993a), que empregou o seguinte método:

- a) a partir das especificações do filtro, era determinado o número de coeficientes do filtro segundo a equação (31);
- b) para este número de coeficientes, um AG era iniciado para buscar um filtro que atendesse às especificações. Tais coeficientes eram pertencentes ao conjunto de números em potências de dois, também a fim de se prover um melhor desempenho para o filtro;
- c) caso não se encontrasse uma solução aceitável, um coeficiente a mais era adicionado, até ser encontrada a solução.

Este método foi melhorado em Cemes e Ait-Boudaoud (1993b), através do seguinte artifício: o coeficiente do filtro podia agora não mais somente receber um único valor de potência de dois, mas também dois valores de potência de dois somados ou subtraídos, aumentando-se a possibilidade de valores a serem expressos em potências de dois que anteriormente era restrito a um só e nem por isso aumentando o tempo de processamento significativamente, pois a soma era ainda mais vantajosa do que a multiplicação.

Há um livro (MAN, TANG e KWONG, 1999) que traz uma abordagem específica para a síntese de filtros digitais através de AEs. Mesmo com um estudo restrito aos filtros IIR, é possível nesta literatura serem aproveitados os esquemas de codificação (representação cromossômica).

Em Hounsell, Arslan e Thomson (2004), é apresentado um método para serem projetados filtros FIR por meio de HE que gera coeficientes em potências de dois. Portanto, é possível atualmente serem implementados diretamente em estruturas (como a Forma Direta I da Figura 3), caso seja necessário, filtros FIR.

Em Chen e Yu (2000), é utilizada PE para serem obtidos dois pontos de frequência. Estes dois pontos obtidos pelo método evolucionário irão ser utilizados em um método especialista de síntese de filtros FIR chamado amostragem de frequência. Tais pontos de transição são normalmente consultados em uma tabela (RABINER, GOLD, MCGONEGAL, 1970), a qual força o projetista a adequar o perfil do filtro aos valores apresentados na tabela e que não obrigatoriamente oferece os *ripples* mínimos. O que se busca otimizar neste método híbrido é a amplitude dos *ripples*. Analogamente, em Huang, Zhou e Qian (2004), é relatado um trabalho onde foi utilizado PSO para projetar-se filtros FIR, fornecendo-se as amostras de

transição para o método da amostragem de frequência. Um trabalho análogo foi feito, porém para AIS, em Xiaoping, Bo e Gang, (2003).

Em Benevenuto, Marchesi e Uncini (1992), é apresentado um método utilizando AS para gerar coeficientes em potências de dois para filtros FIR empregando AS. Cabe aqui a observação de que há diversos trabalhos com AS para o projeto de filtros FIR.

Em Ng et.al. (1999) é relatado um método para o projeto de filtros FIR via o método dos mínimos quadrados mesclado com EE.

Em Zhang e Iwakura (1994), é empregada uma RNA em um método para serem gerados filtros FIR baseados no módulo do erro quadrático entre a frequência desejada e a frequência que está sendo gerada. Filtros FIR e RNA são também assuntos amplamente pesquisados.

Em ED, não há somente um método (STORN, 2005), assim como há um *software* (STORN, 2003): em 01/02/2006 seu preço estava em US\$ 500,00 e trata-se de uma aplicação que sob o ponto de vista de projetos de filtros, não apresenta rigorosamente algum tipo de inovação em termos de características que a diferencie dos *softwares* comerciais com métodos especialistas, porém, por ser um AE, o limite de coeficientes de filtros FIR ser de 200, é um ponto relevante. O código-fonte não é revelado e pelo que se pôde observar no manual, é necessário que seu usuário conheça parâmetros evolucionários (*population vector*, por exemplo).

Em PG, foi também encontrada uma dissertação (MUÑHOZ, 2005) que apresenta uma derivação deste paradigma, denominada GEP (*Genetic Expression Programming*), para o projeto de filtros FIR.

Embora tenha sido um estudo sobre filtros IIR, em Nambiar e Mars (1992) é feito um estudo comparativo entre AS e AGs, tendo-se chegado à conclusão de que cada uma das técnicas é melhor em situações diferentes durante a execução. Por este motivo, foi proposto neste trabalho um algoritmo híbrido mesclando as duas técnicas.

Já na área dos métodos especialistas, somente para ser citado um deles, por exemplo, há um método que é capaz de fornecer filtros digitais com um número de coeficientes que pode ir até 1500 (EVANGELISTA, 2002).

O próprio método especialista alvo da comparação neste trabalho já recebeu melhoramentos, sendo o trabalho de Antoniou (1983) um dos representantes.

Na grande maioria, atualmente em um número que se aproxima dos 100 (levando-se em consideração os periódicos de circulação internacional), os trabalhos desenvolvidos para o projeto de filtros FIR com AGs é voltado à obtenção de coeficientes expressos em potências

de dois. Foram tomados alguns exemplos somente para ser oferecida uma noção do que se tem à disposição da comunidade científica.

Em Gentili, Piazza e Uncini (1994), é apresentado um AG para o projeto de filtros FIR, modelado para fornecer coeficientes expressos em potências de dois. Neste trabalho é feito um confronto entre o AG proposto e AS, um método especialista baseado em programação linear e outros métodos relatados. O AG só perde para o método especialista, empata com AS e ganha dos demais. O empate com o AS ocorre, porém com a ressalva de que foi necessária mais de uma execução via AS para atingir a qualidade de resposta obtida pelo AG em somente uma execução.

Os trabalhos de Karaboğa e Çetinkaya (2003) e Öner (1998) podem ser considerados representantes de diversos trabalhos que adotaram funções de *fitness* no formato:

$$f(\cdot) = \frac{a}{b + x(op)y} \quad (39)$$

Onde:

- $f(\cdot)$  é o valor da função de *fitness*;
- $a$  e  $b$  são números ajustáveis caso a caso;
- $x$  são valores dependentes de diferenças entre a resposta em frequência desejada e a obtida por cada indivíduo;
- $y$  seja um valor de peso dependente de cada perfil de filtro a ser projetado (configurado caso a caso); e
- $op$  seja uma operação matemática do tipo produto, soma, diferença, multiplicação ou divisão.

Funções de *fitness* neste formato provêm uma convergência com uma boa aceleração e ao mesmo tempo podem ainda ser controladas para ser evitada a convergência prematura.

Em Lee et. al. (1999), é proposta uma modificação no formato padrão do AG da Figura 13, sendo sacrificados os indivíduos que não atendam às restrições, com fins a se operar somente com potências de dois.

### 2.3.2 Trabalhos na área

Em Hasan et.al. (2001), é apresentado um método e uma *toolbox* de MATLAB (MATHWORKS, 2002) chamada CSDFIR. É uma ferramenta automática e que atualmente pertence à MOTOROLA.

Foi encontrado um *software* que gera os coeficientes de filtros FIR a partir de especificações nele inseridas. É outra *toolbox* para MATLAB (MATHWORKS, 2002) desenvolvida pelo Professor Sabah M. Ahmed (2003), apresentada em um simpósio. As diferenças deste trabalho para o trabalho aqui proposto são:

- a) para este *software* ser executado é necessário o MATLAB. O *software* aqui proposto é executado diretamente a partir do Windows; e
- b) é um *software* que mantém os parâmetros de AG em aberto para serem configurados pelo usuário, isto é, necessita-se conhecer AGs para utilizá-lo. O *software* aqui proposto deixa em aberto ao usuário um ou no máximo dois parâmetros simples de AGs para que o usuário necessite eventualmente configurá-los, sendo desnecessário o conhecimento de AGs para ser utilizado.

Alguns problemas foram observados com relação aos resultados relatados: o filtro proposto no exemplo da seção 5 está invertido (a amostra central é negativa) e sua resposta em frequência, quando testada, não proporciona o formato anunciado no artigo. Foi encaminhada uma mensagem eletrônica para o autor, Professor Ahmed, em 20/01/2006, no endereço constante no artigo, solicitando do mesmo uma cópia da *toolbox* para que fossem feitos testes, não tendo sido obtida nenhuma resposta. Foram também procurados registros referentes ao Professor Ahmed para verificar se o mesmo tinha alguma página pessoal na qual existissem mais dados a respeito de seu trabalho. Não foi encontrada a referida página nem alguma página da Faculdade de Engenharia de Assuit, local de procedência do trabalho.

Todos os trabalhos citados nesta seção não apresentaram explicitamente os passos adotados na modelagem. Em todos, por exemplo, não é encontrada a codificação adotada ou ainda não é informado quantas vezes foram executados o AG para serem obtidos os resultados anunciados. Em alguns não é possível ser compreendida com clareza o cálculo da função de *fitness* que foi adotado.



## CAPÍTULO 3

### METODOLOGIA

Conforme citado na revisão da literatura, não foi encontrado um método e seu respectivo *software* que enfatizasse a atual proposta do trabalho proposto: gerar filtros FIR via AGs sem que o usuário necessite conhecer algum tipo de paradigma de CE e que um ofereça um resultado relevante em um número pré-determinado de tentativas.

Para a metodologia a ser empregada dentro deste contexto, foi escolhido o seguinte caminho: para qualificar os resultados como relevantes, foi escolhido para fins comparativos um método consagrado, o MPM, resultando em uma pesquisa experimental. Porém, não foram encontrados na literatura padrões pré-estabelecidos de comparação entre respostas em frequência. Portanto, tais padrões foram criados, o que resultou em uma pesquisa com enfoque experimental tendo como base critérios de referência de uma pesquisa exploratória (GIL, 1996).

#### 3.1 MATERIAIS E INFRA-ESTRUTURA

Foi um experimento simples em termos de materiais e infra-estrutura: computador, referências bibliográficas e *softwares* de apoio à depuração dos resultados.

##### 3.1.1 Computador

Foi utilizado um computador pessoal do tipo *notebook*, da marca *Hewlett-Packard*, modelo *Pavillion ze2000*, cujas características relevantes ao trabalho foram a de ter um processador da marca *Intel*, modelo *celeron* com um *clock* de 1.4 GHz, memória RAM de 256 Mb, interface de rede *Fast-Ethernet* e sistema operacional *Windows* (MICROSOFT, 2002) versão *XP-Home-Eddition*, release 5.1.2600.

O critério de escolha do sistema operacional foi o *software* desenvolvido poder atingir uma comunidade maior de acadêmicos. Os tempos de execução apresentados nas tabelas de resultados foram obtidos neste computador.

### 3.1.2 *Softwares* utilizados

O MATLAB (MATHWORKS, 2002) foi o software de validação de códigos e análise de resultados durante o experimento.

O GALOPPS (GOODMAN e PUNCH, 2002) foi utilizado como ferramenta de apoio ao desenvolvimento do algoritmo genético. Para que uma aplicação de AGs seja desenvolvida no GALOPPS, é necessário um compilador da linguagem C. O escolhido para o experimento foi o GCC (MITCHEL, 2005). Os critérios de escolha desta ferramenta foram:

- a) ser gratuito;
- b) ter o seu código-fonte disponível (isto possibilitou, por exemplo, a análise do mecanismo de geração aleatória de indivíduos da população); e
- c) oferecer internamente o fluxograma da Figura 13, requerendo do projetista somente a modelagem da função de *fitness*, a especificação dos parâmetros de execução, dos operadores de cruzamento e de mutação e do método de seleção.

O BORLAND C++ BUILDER (BORLAND, 2000) foi utilizado para desenvolver partes do *software*. Este ambiente de desenvolvimento oferece uma forma de programação denominada Programação Baseada em Componentes (PRESSMAN, 2002). É possível, através deste conceito, ser desenvolvida toda uma programação estruturada, no padrão ANSI-C, sendo utilizados os recursos gráficos e *multi-thread* que o Windows oferece, aumentando desta forma a portabilidade códigos escritos anteriormente no modo Dos/Console. Basta ao programador que domina a linguagem C conhecer um conjunto proprietário de comandos da BORLAND responsável pela interface gráfica.

Dentro do *software* desenvolvido, foram utilizados três códigos disponíveis na *Internet*:

- a) um para o cálculo da TRF (NIELSEN, 2000);
- c) outro para a geração do filtro FIR pelo MPM (JANOVETZ, 1998), porém na linguagem C ao invés de FORTRAN como fora lançado em 1972; e
- d) e uma pequena função que converte de binário posicional para Gray Refletido (PRESS, 2002).

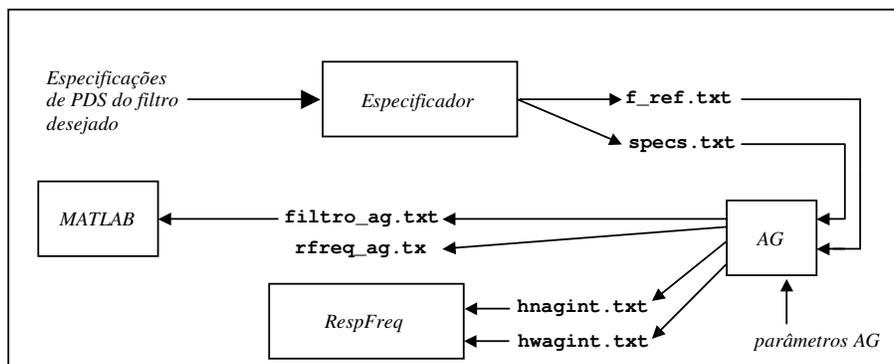
### 3.2 O MÉTODO

O método proposto é apoiado em uma pesquisa experimental: são testadas diversas possibilidades de configuração do AG para serem verificadas a qualidade da resposta e qual configuração mais permanece constante diante das mais diferentes possibilidades de especificações que forem passadas. A qualidade baseia-se em oferecer uma resposta em frequência com um perfil equiparado ao do MPM, senão melhor.

#### 3.2.1 Descrição geral do cenário dos experimentos

Os experimentos foram realizados no cenário representado pela Figura 14. A dinâmica do experimento consiste dos seguintes passos:

- há um programa, chamado de *Especificador*, que recebe do usuário as especificações desejadas de filtro a ser gerado;
- recebendo os dados, o Especificador gera dois arquivos. Um é chamado de *f\_ref.txt*, com  $N$  amostras, que é a resposta em frequência desejada do filtro, na forma ideal, isto é, com os intervalos entre as faixas um dois e três unidos com segmentos de reta, fazendo com que  $|H_d(f)|$  tenha um formato trapezoidal. O formato desta resposta em frequência ideal pode ser visto nos gráficos do capítulo de Resultados, com a legenda correspondente. O outro arquivo, *specs.txt*, contém informações de controle para o funcionamento do AG;



**Figura 14:** Visão geral do experimento.

- c) o programa que executa o AG é implementado dentro do GALOPPS e recebe como dados de entrada os dois arquivos gerados pelo programa *Especificador*, juntamente com todo o conjunto de parâmetros AG que o GALOPPS recebe. Estes parâmetros serão descritos adiante;
- d) durante a execução do AG, dois arquivos criados vão recebendo informações, que são o `hnagint.txt`, o qual contém os coeficientes do melhor filtro obtido por geração e o `hwagint.txt` que contém a respectiva resposta em frequência de tal filtro;
- e) ao final da execução do AG, um arquivo, com o nome `filtro_ag.txt`, contém os coeficientes do filtro gerado, o qual é confrontado com o método especialista no MATLAB. Para quaisquer propósitos, também é fornecida a resposta em frequência deste filtro em um arquivo chamado `rfreq_ag.txt`; e
- f) durante ou depois da execução, é possível ser executado um outro programa denominado *RespFreq*, que permite monitorar a qualidade da resposta em frequência que esteja sendo obtida por meio da leitura do arquivo `hwagint.txt`. Caso uma determinada resposta em frequência intermediária de uma geração *g-ésima* seja considerada satisfatória, basta serem coletados os *g-ésimos* coeficientes correspondentes no arquivo `hnagint.txt`.

### 3.2.2 Variáveis do experimento

São variáveis independentes as pertencentes ao conjunto  $A$ , que são as especificações do filtro desejado:

$$A = \{fn_{11}, fn_{12}, fn_{21}, fn_{22}, fn_{31}, fn_{32}, A_1, A_2, A_3, Tipo, M\} \quad (40)$$

Onde:

- $fn_{11}, fn_{12}, fn_{21}, fn_{22}, fn_{31}$  e  $fn_{32}$  são as frequências iniciais e finais das faixas um dois e três, respectivamente, da resposta em frequência desejada, assumindo valores entre 0,0 e 0,5 ciclos/amostra;

- $A_1$ ,  $A_2$  e  $A_3$  são as amplitudes nas faixas um dois e três, respectivamente, da resposta em frequência desejada, assumindo valores entre 0,0 e 1,0;
- *Tipo* é o tipo de filtro desejado a ser gerado, assumindo valores do I ao IV; e
- $M$  é o número de coeficientes que o filtro a ser gerado deva ter, assumindo valores inteiros positivos.

O MPM, método a ser confrontado, exige algumas restrições que deverão ser seguidas por este motivo:

$$fn_{12} \neq fn_{21} \quad (41)$$

$$fn_{22} \neq fn_{31} \quad (42)$$

São variáveis independentes, porém indiretas, as pertencentes ao conjunto B:

$$B = \{fn_{R11}, fn_{R12}, fn_{R21}, fn_{R22}, fn_{R31}, fn_{R32}, A_{R1}, A_{R2}, A_{R3}, \delta_{R1}, \delta_{R2}, \delta_{R3}\} \quad (43)$$

Onde:

- $fn_{R11}$ ,  $fn_{R12}$ ,  $fn_{R21}$ ,  $fn_{R22}$ ,  $fn_{R31}$  e  $fn_{R32}$  são as frequências iniciais e finais das faixas um dois e três, respectivamente, da resposta em frequência do filtro FIR obtido via MPM, assumindo valores entre 0,0 a 0,5 ciclos/amostra;
- $A_{R1}$ ,  $A_{R2}$  e  $A_{R3}$  são as amplitudes nas faixas um, dois e três, respectivamente, da resposta em frequência do filtro FIR obtida via MPM, assumindo valores entre 0,0 e 1,0;
- $\delta_{R1}$ ,  $\delta_{R2}$  e  $\delta_{R3}$  são, respectivamente, os *ripples* nas faixas um dois e três na resposta em frequência do MPM, assumindo valores entre 0,001 a 0,500.

São variáveis independentes, porém indiretas, as pertencentes aos conjuntos C, D e E:

$$C = \{\text{seleção, cruzamento, mutação, rc, fitness}\} \quad (44)$$

$$D = \{\text{elitismo, maxgen, popsize, pcross, pmut, semente}\} \quad (45)$$

$$E=\{par\_sel, par\_escal\} \quad (46)$$

Onde:

- *seleção* é o método de seleção escolhido, assumindo os seguintes valores: *suselect* (amostragem estocástica universal), *rselect* (roleta estocástica) e *tselect* (torneio estocástico);
- *cruzamento* é o operador de cruzamento escolhido, assumindo os seguintes valores: *oneptx* (cruzamento de um ponto), *twoptx* (cruzamento de dois pontos) e *unifx* (cruzamento uniforme);
- *mutação* é o operador de mutação escolhido, assumindo os seguintes valores: *bitmutat* (mutação bit a bit) ou *multimut* (mutação de múltiplos campos);
- *rc* é a representação cromossômica escolhida, assumindo os seguintes valores: *BinPos* (binário posicional) e *BinGray* (Binário Gray Refletido);
- *fitness* é a função de *fitness* escolhida, ser detalhada na subseção de modelagem;
- *elitismo* é o parâmetro que determina se foi escolhido elitismo ou não, assumindo os valores *sim* ou *não*;
- *maxgen* é o número máximo de gerações, assumindo valores inteiros;
- *popsize* é o tamanho da população, que assumirá valores inteiros;
- *pcross* e *pmut* são as probabilidades na quais ocorrerão os cruzamentos e as mutações, respectivamente, assumindo valores de 0,0 a 1,0; e
- *par\_sel* e *par\_escal* são, respectivamente, parâmetros que dependam do método de seleção e do tipo de escalonamento de *fitness*.

O conjunto *C* corresponde aos parâmetros de AG que normalmente ficam fixos dentro do GALOPPS, requerendo uma nova compilação do código-fonte, caso precisem de alteração.

O conjunto *D* corresponde aos parâmetros de AG que normalmente são alterados caso a caso. No GALOPPS, podem tanto receber valores iniciais, quanto serem alterados ao longo da execução.

Os elementos do conjunto *E* dependem dos parâmetros dos conjuntos *C* e *D* e, por este motivo, foram deixados em separado.

São variáveis dependentes as pertencentes ao conjunto *F*:

$$F=\{fn_{AG11}, fn_{AG12}, fn_{AG21}, fn_{AG22}, fn_{AG31}, fn_{AG32}, A_{AG1}, A_{AG2}, A_{AG3}, \delta_{AG1}, \delta_{AG2}, \delta_{AG}, t_{AG}\} \quad (47)$$

Onde:

- $fn_{AG11}$ ,  $fn_{AG12}$ ,  $fn_{AG21}$ ,  $fn_{AG22}$ ,  $fn_{AG31}$  e  $fn_{AG32}$  são as frequências iniciais e finais das faixas um dois e três, respectivamente, da resposta em frequência do filtro FIR gerado pelo AG, assumindo valores de 0,0 a 0,5 ciclos/amostra;
- $A_{AG1}$ ,  $A_{AG2}$  e  $A_{AG3}$  são as amplitudes nas faixas um dois e três, respectivamente, da resposta em frequência do filtro FIR gerado pelo AG, assumindo valores de 0,0 a 0,5;
- $\delta_{AG1}$  e,  $\delta_{AG2}$  e  $\delta_{AG3}$  são, respectivamente, os *ripples* nas faixas um dois e três presentes na resposta em frequência do filtro FIR gerado pelo AG, assumindo valores de 0,001 a 0,500; e
- $t_{AG}$  é o tempo consumido pelo AG para gerar os coeficientes do filtro FIR.

### 3.2.3 Condições de resposta sub-ótima

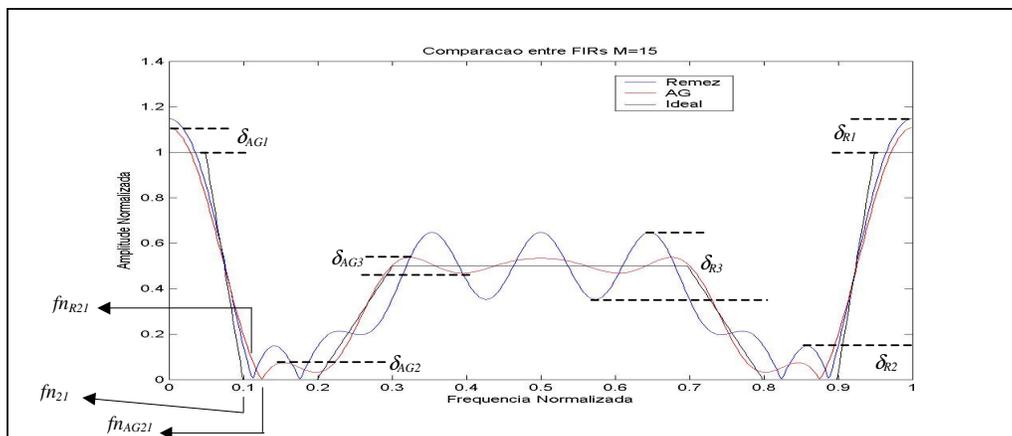
O confronto entre os métodos consiste de duas comparações: entre *ripples* e entre pontos onde ocorrem as transições da resposta em frequência. Destas duas comparações surgem duas condições que foram classificadas como sub-ótimas a serem procuradas.

Para apresentá-las, será utilizado um exemplo. Seja um filtro de resposta arbitrária Tipo I, com 15 coeficientes, a ser projetado pelas especificações constantes no conjunto a seguir:

$$A = \{fn_{11}, fn_{12}, fn_{21}, fn_{22}, fn_{31}, fn_{32}, A_1, A_2, A_3, Tipo, M\}$$

$$= \{0,00; 0,05; 0,10; 0,20; 0,30; 1,00; 0,00; 0,50; I, 15\}$$

As respostas em frequência dos filtros gerados pelo MPM e pelo AG, com seus respectivos *ripples* correspondentes aos conjuntos *B* e *E* estão mostrados na Figura 15:



**Figura 15:** *Ripples* envolvidos no experimento.

Na Figura 15 e demais a serem apresentadas, uma legenda com o nome *Remez* corresponde ao mesmo que MPM. Serão ainda apresentadas respostas em frequência tanto de 0,0 a 0,5 ciclos por amostra, quanto de 0,0 a 1,0 ciclos/amostra (veja subseção 2.1.4).

Com relação à Figura 15, são observados todos os *ripples* gerados pelo AG de valor inferior aos gerados pelo MPM. Quanto menores forem os *ripples*, maior será a qualidade do filtro. Será chamada de PCRSO (Primeira Condição de Resposta Sub-Ótima) a que estabelece a seguinte inequação:

$$\left| 20 \times \log(\delta_{iAG}) - 20 \times \log(\delta_{iR}) \right| \leq 5, \quad i = 1, 2, 3 \quad (48)$$

Onde:

- $\delta_{iAG}$  são os *ripples* das faixas um dois e três da resposta em frequência do AG; e
- $\delta_{iR}$  são os *ripples* das faixas um dois e três da resposta em frequência do MPM.

Por exemplo:  $\delta_{1AG} = 0,0694$  e  $\delta_{2R} = 0,0470$ , têm entre si uma distância de 0.0224, a qual está em conformidade com a PCRSO. Este valor é mais facilmente visto nas respostas em frequência cuja amplitude está em decibéis.

É ainda observado na Figura 15 que  $fn_{21AG} > fn_{21R} > fn_{21}$ . Já na faixa três, a relação passa a ser  $fn_{31R} > fn_{31AG} > fn_{31}$ . Quanto mais próximas estas frequências estiverem do valor especificado (do ideal), melhor será a qualidade do filtro. Será chamada de SCRSO (Segunda Condição de Resposta Sub-Ótima): a que estabelece a seguinte inequação:

$$\left| f_{iAG} - f_{iR} \right| \leq 0,05 \quad (49)$$

Onde:

- $f_{iAG}$  são as frequências  $fn_{AG21}$  e  $fn_{AG31}$  da resposta em frequência do AG; e
- $f_{iR}$  são as frequências  $fn_{R21}$  e  $fn_{R31}$  da resposta em frequência do MPM.

Um filtro que obedeça às inequações (48) e (49) será classificado como um filtro que tenha atendido às PSCRSO (Primeira e Segunda Condições de Resposta Sub-Ótima).

As medidas que deram origem às PSCRSO tiveram como base resultados obtidos por trabalhos correlatos (HASAN et. al., 2001), (CEMES e AIT-BOUDAUD, 1993a), (CEMES e AIT-BOUDAUD, 1993b).

#### 3.2.4 Formulação do problema e da Hipótese

O problema a ser resolvido é: como reduzir o número de variáveis ajustáveis dos conjuntos  $C$ ,  $D$  e  $E$ , de maneira a maximizar o valor dos elementos do conjunto  $A$ , gerando resultados correspondentes no conjunto  $F$  que atendam às PSCRSO.

A hipótese defendida é a de que exista pelo menos uma combinação de parâmetros dos conjuntos  $C$ ,  $D$  e  $E$  que possa satisfazer às PSCRSO. Tal combinação poderá, se necessário, ajustar o tamanho da população de indivíduos, a semente aleatória e o número de gerações. Com relação ao tamanho da população de indivíduos e o número máximo de gerações, serão procuradas relações destas com as variáveis do conjunto  $A$ , para serem encontradas formas de automatização, dispensando seu ajuste manual caso a caso. Com relação à semente aleatória, serão procuradas relações com os demais parâmetros dos conjuntos  $C$ ,  $D$  e  $E$ , de maneira a não ser necessário mais do que quatro tentativas de execução para ser encontrada uma resposta que atenda às PSCRSO.

#### 3.2.5 Operacionalização das variáveis e critérios de redução

Há três linhas de ação que disponíveis para a condução do trabalho:

- a) testes baseados na combinação de todas as possibilidades de valores para as variáveis dos conjuntos apresentados;
- b) testes baseados na escolha de determinados valores para as variáveis dos conjuntos apresentados por critérios intuitivos; e
- c) testes baseados em combinações selecionadas com critérios técnicos para os valores das variáveis dos conjuntos apresentados.

Para operacionalizar as variáveis com base em testes através da primeira alternativa, um exemplo simples pode fornecer uma estimativa da magnitude de tal experimento: 20 ordens de filtro, 20 perfis de filtros diversificados, 5 valores de população, 5 valores de

geração, 5 métodos de seleção, 3 tipos de cruzamento, 2 tipos de mutação, 4 probabilidades de cruzamento, 4 probabilidades de mutação, 3 tipos de escalonamento de *fitness*, 3 possíveis combinações de valores que parâmetros associados a tais tipos de escalonamento possam receber, 4 sementes aleatórias, 2 representações cromossômicas e 3 funções de *fitness*. Sendo executados uma só vez, seriam  $20 \times 20 \times 5 \times 5 \times 5 \times 3 \times 2 \times 4 \times 4 \times 3 \times 3 \times 4 \times 2 \times 3 = 1.036.000.000$  testes.

A outra alternativa de operacionalização seria a de não haver um critério definido para serem testadas as possibilidades. Um exemplo disto seria o projetista adotar um arquivo de configuração de um outro problema que contivesse uma combinação de valores definidos para os conjuntos *C*, *D* e *E*, adequando-o ao seu problema. Esta alternativa foi experimentada em uma fase de testes informais e não conseguiu se manter para cinco tipos de filtros, todos com um mesmo número de coeficientes.

Face ao exposto, restou a operacionalização das variáveis pautada em critérios que conseguissem reduzir o total de combinações para um número realizável.

Para reduzir os métodos de seleção a serem testados, três conceitos serão apresentados. James Edward Baker teve seu trabalho citado em Goldberg (1989) e em Chipfield et. al. (1994), como tendo criado três conceitos relativos aos métodos de seleção:

- a) o desequilíbrio (do inglês *bias*), que é a diferença entre a probabilidade atual e a esperada, com base na equação (38);
- b) o espalhamento (do inglês *spread*), que é o número de ocorrências que determinado indivíduo poderá ter na população intermediária resultante da seleção; e
- c) a eficiência, que está relacionada com tempo de processamento, resultante da complexidade computacional.

Na seleção pela roleta estocástica, não há um desequilíbrio em favor de nenhuma parte, inteira ou fracionária, porém o espalhamento para qualquer indivíduo com uma probabilidade não-nula é ilimitado.

Na seleção pelo Resto Estocástico, há o aparecimento de um desequilíbrio em favor de indivíduos que tenham partes fracionárias decorrentes dos cálculos da equação (38), porém restringe o espalhamento.

E na Amostragem Estocástica Universal, não há desequilíbrio, o espalhamento é mínimo e a complexidade computacional é menor que os métodos de seleção anteriores.

Encontra-se em Goodman (1996) uma recomendação para ser preferida a Amostragem Estocástica Universal à Roleta Estocástica e ao Resto Estocástico.

Portanto, serão descartados do processo os métodos da roleta estocástica e do resto estocástico, para ficarem somente participantes dos testes: Amostragem Estocástica Universal e Torneio Estocástico.

A função de *fitness* foi mantida fixa nos padrões da equação (39). Esta escolha foi feita a partir de comparações em testes preliminares com outras funções de *fitness* baseadas em diferenças normalizadas do tipo  $1 - x$ , onde  $x$  seja a diferença entre a resposta em frequência desejada e a resposta em frequência de cada indivíduo. O formato  $1/x$  demonstrou ser capaz de atender a uma variedade maior de especificações.

Com relação à representação cromossômica, será adotada a binária. Através desta representação é esperada uma maximização do número de esquemas, aumentando as chances de surgir indivíduos com esquemas contendo um bloco constituinte da resposta. Com o aumento do número de esquemas, é esperada uma compensação sobre o efeito destrutivo dos cruzamentos e das mutações sobre seu número (GOLDBERG, 1989). Em Dianati, Song e Treiber (2002), são citados quem é contra e a favor da Teoria dos Esquemas. Devido aos contras apoiarem-se em questões envolvendo ruído e neste trabalho o conjunto  $A$  não contemplar tal característica, a opção pela Teoria dos Esquemas será mantida. Detalhes da codificação serão apresentados na subseção da modelagem.

Com relação aos cruzamentos, a partir de Beasley, Bull e Martin (1993), foi preferido o cruzamento de dois pontos ao de um ponto, sem ainda a preocupação de serem perdidos os blocos constituintes de esquemas. E o cruzamento uniforme foi também preterido em relação ao de dois pontos, com base na mesma literatura, por ter ocorrido um relato de um experimento com um tamanho grande de populações.

Embora seja possível encontrar em Goodman (1996) uma recomendação de uso do operador de mutação de múltiplos campos ao invés do operador de mutação *bit a bit* para problemas enganadores<sup>3</sup>, foram mantidos os dois tipos de mutação dentro da seqüência de avaliação para serem testados. Isto foi feito, devido ao problema poder revelar indícios de interação entre os genes.

Independentemente dos critérios, será colocado, para efeitos de comparação, o modelo de AG canônico (DIANATI, SONG e TREIBER, 2002) a fim ser proporcionado uma noção de qual seria seu desempenho para resolver um problema como estes.

---

<sup>3</sup> Um problema enganador é aquele em que acontece o seguinte fenômeno (*deception*): os indivíduos gerados em uma população, embora resultantes da combinação de pais cujos blocos constituintes tenham bons valores de *fitness*, a sua própria *fitness* que deveria ser maior, é mais baixa que a deles (GOLDBERG, 1989).

No intuito de serem diminuídos o número de testes a serem realizados, as seguintes considerações foram feitas:

- a) manterem-se em valores fixos a probabilidade de mutação *pmutation* em 0,10 e a probabilidade de cruzamento *pcross* em 0,85; e
- b) não serem realizados testes com escalonamentos de *fitness* (os valores de *fitness* ficarão na forma bruta) nem com *crowding*.

As possíveis combinações de especificações do conjunto *A* que possam surgir foram representadas por um banco de filtros-teste que está apresentado na Tabela 4.

Através destes filtros são excursionadas sistematicamente faixas de transição que vão de 0,08 até 0,05 ciclos/amostra, assim como larguras da faixa passante ou de rejeição que vão de 0,04 até 0,01 ciclos/amostra.

Com os critérios adotados, o número de testes ficou reduzido para: 3 métodos de seleção, 3 operadores de cruzamento, 2 operadores de mutação, 2 tipos de representação cromossômica, 2 possibilidades de elitismo, perfazendo um total de 144 testes para uma especificação de filtro. Logo, para os 19 tipos de filtros-teste a serem analisados, o número de testes passa para 2736. Se os testes tiverem de ser repetidos com quatro diferentes sementes aleatórias este valor será de 10944, o que ainda é alto.

Para reduzir este número de testes ainda alto foi feita uma divisão em três fases, que serão detalhadas na próxima subseção:

- a) a primeira fase mantém alguns parâmetros fixos e varia outros, de acordo com seus possíveis valores apresentados na subseção 3.2.2 e reduzidos segundo os critérios apresentados anteriormente nesta subseção;
- b) a segunda fase toma os melhores resultados obtidos da fase anterior e realiza testes invertendo o cenário para os parâmetros: quem fica fixo agora são aqueles que melhor se classificaram na primeira fase e são alterados os parâmetros que ficaram anteriormente fixos;
- c) a terceira fase receberá resultante da primeira e segunda fases pelo menos uma configuração de parâmetros fixados em seus melhores valores, que atenda às PSCRSO em pelo menos quatro tentativas. Esta fase é responsável por uma compensação: irá confrontar esta configuração com variantes da mesma contemplando valores e técnicas que foram desconsiderados do plano de ação principal das fases anteriores. Escalonamentos de *fitness*,

assim como outros valores para as probabilidades de mutação e de cruzamento são exemplos de testes realizados nesta fase.

**Tabela 4:** Banco de filtros propostos.

$N^{\circ}$	Ação	$fn_{11}$	$fn_{12}$	$A_1$	$fn_{21}$	$fn_{22}$	$A_2$	$fn_{31}$	$fn_{32}$	$A_3$	Nome1	Nome2
1	fpb	0,00	0,35	1	0,42	0,45	0	0,45	0,50	0	fpb-apresentação	fpba
2	fpb	0,00	0,15	1	0,23	0,35	0	0,35	0,5	0	fpb-padrão	fpb1
3	fpb	0,00	0,37	1	0,45	0,46	0	0,46	0,5	0	fpb-largo	fpb2
4	fpb	0,00	0,04	1	0,12	0,20	0	0,20	0,5	0	fpb-estreito	fpb3
5	fpb	0,00	0,15	1	0,19	0,30	0	0,30	0,5	0	fpb-transição	fpb4
6	fpa	0,00	0,20	0	0,20	0,33	0	0,40	0,5	1	fpa-padrão	fpa1
7	fpa	0,00	0,04	0	0,04	0,05	0	0,12	0,5	1	fpa-largo	fpa2
8	fpa	0,00	0,35	0	0,35	0,40	0	0,47	0,5	1	fpa-estreito	fpa3
9	fpa	0,00	0,20	0	0,20	0,27	0	0,30	0,5	1	fpa-transição	fpa4
10	fpa	0,00	0,40	0	0,40	0,44	0	0,47	0,5	1	fpa-estreito-transição	fpa5
11	fpf	0,00	0,14	0	0,20	0,30	1	0,36	0,5	0	fpf-padrão	fpf1
12	fpf	0,00	0,05	0	0,11	0,39	1	0,45	0,5	0	fpf-largo	fpf2
13	fpf	0,00	0,14	0	0,20	0,22	1	0,28	0,5	0	fpf-estreito	fpf3
14	fpf	0,00	0,18	0	0,20	0,38	1	0,40	0,5	0	fpf-transição	fpf4
15	fcf	0,00	0,20	1	0,25	0,35	0	0,40	0,5	1	fcf-padrão	fcf1
16	fcf	0,00	0,05	1	0,10	0,40	0	0,45	0,5	1	fcf-largo	fcf2
17	fcf	0,00	0,11	1	0,16	0,17	0	0,22	0,5	1	fcf-estreito	fcf3
18	fcf	0,00	0,15	1	0,16	0,44	0	0,45	0,5	1	fcf-transição	fcf4
19	fcf	0,00	0,28	1	0,29	0,30	0	0,31	0,5	1	fcf-estreito-transição	fcf5

Convenções: fpb = passa-baixas, fpa = passa-altas, fpf = passa-faixa e fcf = corta-faixa.

### 3.2.6 Os testes propostos

As fases dos testes apresentadas na subseção anterior são aqui descritas em detalhes.

#### 3.2.6.1 Primeira fase

É a etapa que visa encontrar uma configuração de parâmetros de AG que melhor resolva o subproblema de se projetar filtros FIR para um número fixo de coeficientes.

Para tal, serão testadas as combinações dos conjuntos *C* e *D*, reduzidas de acordo com os critérios apresentados na subseção 3.2.5. Os testes serão realizados com o banco de filtros da Tabela 4 (que representa o conjunto *A*) e uma nota é associada aos resultados de cada combinação de parâmetros AG que for testada.

O teste e sua avaliação de desempenho das configurações será conforme segue:

- a) uma configuração de parâmetros de AG é proposta;
- b) com esta configuração busca-se gerar cada um dos filtros do banco de filtros. O número de filtros do banco que serão testados dependerá do seu tipo devido às suas restrições (veja subseção 2.1.6): 19 para filtros Tipo I, 9 para filtros Tipo II, 4 para filtros Tipo III e 10 para filtros Tipo IV;
- c) o número máximo de tentativas por filtros para ser obtida pelo menos uma resposta em frequência que atenda às PSCRSO é 4;
- d) se a combinação de parâmetros AG sob teste não conseguir em quatro tentativas gerar uma resposta que atenda às PSCRSO, o teste é interrompido. A nota que será atribuída a esta combinação de parâmetros AG será zero;
- e) se a combinação de parâmetros AG sob teste conseguir passar por todos os filtros gerando pelo menos uma resposta que atenda às PSCRO, concluindo o teste, uma nota será estipulada. Esta nota dependerá de três classificações atribuídas aos resultados de cada uma das execuções: NOK para os casos em que a resposta do filtro AG não atenda às PSCRSO, SS para os casos em que tenha atendido e uma classificação especial, denominada OK, que corresponde a uma resposta fornecida pelo AG que ofereça todos os seus *ripples* com valores inferiores aos fornecidos pelo MPM e que também atenda à SCRSO.

A nota será calculada de acordo com:

$$Nota = \frac{N(Oks) + N(SS) / 2}{D} \times 10 \quad (50)$$

Onde:

- *Nota* é a nota que a configuração irá receber;
- *N(Oks)* é a quantidade de OKs recebidos;
- *N(SS)* é a quantidade de SSs recebidos;
- *D* é o número total de testes que será 76 (19×4) para filtros Tipo I, 36 (9×4) para filtros Tipo II, 16 (4×4) para filtros Tipo III e 40 (10×4) para filtros Tipo IV.

A planilha que atende às condições do teste principal pode ser vista na Tabela 5.

Nesta planilha são preenchidos todos os dados referentes às configurações dos conjuntos *C* e *D*, assim como qual tipo de filtro e qual número de coeficientes (*M*) estão sob avaliação. O campo das observações será destinado principalmente para o registro da *fitness* máxima.

Realizados os testes com uma dada configuração de AG conforme estipulado na Tabela 5, a próxima configuração de parâmetros AG é testada, até se chegar ao final das possíveis configurações.

Resultantes deste teste principal, as combinações de AG que não ficarem com zero, estarão organizadas por nota, podendo desta forma ser escolhida a melhor configuração para prosseguir na segunda fase. Convém ressaltar que o número fixo de coeficientes, para contemplar todos os tipos de filtros deverá pelo menos ter um número de coeficientes par e outro ímpar (veja subseção 2.1.6).

**Tabela 5:** Planilha do teste principal.

	<i>popsi</i>	<i>M</i>	<i>Tipo</i>	<i>maxgen</i>	
	<i>tempo</i>				
	<i>seleção</i>		<i>cruzamento</i>	<i>Mutação</i>	
	<i>fitness</i>		<i>rc</i>	<i>Elitismo</i>	
<i>Nº</i>	<i>Filtro</i>	0,123	0,456	0,789 0,012	<i>Observações</i>
1	fpba				
2	fpb1				
3	fpb2				
4	fpb3				
5	fpb4				
6	fpa1				
7	fpa2				
8	fpa3				
9	fpa4				
10	fpa5				
11	fpf1				
12	fpf2				
13	fpf3				
14	fpf4				
15	fcf1				
16	fcf2				
17	fcf3				
18	fcf4				
19	fcf5				
	<i>Nº OK</i>	<i>Nº SS</i>	<i>Nota</i>		

### 3.2.6.2 Segunda fase

A primeira fase visa estabelecer e determinar o conjunto de parâmetros AG que conseguem resolver o subproblema de se gerar filtros FIR com tamanhos fixos de coeficientes.

A segunda fase toma a melhor configuração da fase anterior e a faz ser confrontada novamente com o banco de filtros, agora com valores variáveis para o número de coeficientes. Esta variação, conforme será apresentado na subseção da modelagem, altera o espaço de busca do problema.

Ainda na segunda fase serão testados os valores dos parâmetros a fim de se observar o surgimento de algum padrão comportamental dos resultados que possa relacionar o número de coeficientes com algum valor de parâmetro da configuração do AG obtida na primeira fase.

Ao final da segunda fase, a expectativa é ter-se uma configuração de parâmetros AG estabelecida e com alguns de seus parâmetros regulados por alguma relação numérica que dependa do número de coeficientes que o filtro deva ter.

### 3.2.6.3 Terceira fase

Este grupo de testes corresponderá a confrontos entre a configuração de AG obtida e algumas configurações que foram excluídas do processo. Por exemplo, as que usariam, escalonamentos de *fitness*, testes com filtros de amplitudes arbitrárias, testes com especificações de filtros via superposição. Neste último caso, trata-se de ser testado a propriedade de sistemas LTI (dentro do qual filtros FIR estão incluídos) apresentada na equação (17). Através desta propriedade, por exemplo, dados três filtros específicos, a resposta em frequência de um filtro passa-baixas somada à de um filtro passa-altas pode ser igual à que um filtro corta-faixa forneça diretamente.

### 3.2.7 A modelagem do AG

O primeiro passo, que corresponde a ser avaliada a necessidade de ser usado um AG para resolver este problema, é justificada através dos trabalhos apresentados pela comunidade científica desde 1991, conforme apresentado na seção da Revisão da Literatura. Porém, é importante ser repetido aqui o motivo: aproximar cada vez mais a resposta em frequência do filtro gerado (conjuntos  $B$  e  $F$ , por exemplo) da resposta em frequência especificada (conjunto  $A$ ).

A natureza do problema: é um problema de otimização. São passados como dados de entrada as especificações para o AG incluindo o vetor da resposta em frequência desejada  $|H_d(f)|$ , para que seja encontrado um indivíduo com um conjunto de coeficientes que tenha um vetor com uma resposta em frequência  $|H_i(f)|$  tal, que a operação  $1/(|H_i(f)|-|H_d(f)|)^2$  resulte em um valor máximo. Foi, portanto, feita uma adaptação de um problema que originalmente era de minimização para se tornar de maximização para adequá-lo à ferramenta de AG escolhida, que foi o GALOPPS, que requer valores de *fitness* crescentes.

As variáveis que afetam a otimização são os coeficientes da resposta ao impulso  $h_{AG}(n)$  do indivíduo, pois são estes coeficientes que são aplicados na TDF da equação (8) combinada com a equação (9). O resultado deste cálculo é o vetor  $|H_i(f)|$ , que é a resposta em frequência do indivíduo usada no cálculo da função de *fitness* a ser apresentada nesta subseção.

A representação das variáveis que afetam a otimização consiste basicamente no vetor  $h_{AG}(n)$  representado em uma cadeia de *bits*.

Os valores extremos que cada coeficiente de  $h_{AG}(n)$  pode receber são 0,5 e -0,5. Estes extremos vêm da teoria de PDS ainda referente a sistemas não causais. Um filtro digital basicamente é uma função do tipo retângulo no domínio da frequência, a qual corresponde a uma função *sinc* no tempo.

A função *sinc*( $x$ ) é definida como sendo:

$$\text{sinc}(x) = \begin{cases} \frac{\text{sen}(\pi x)}{\pi x}, & x \neq 0 \\ 1, & x = 0 \end{cases}, \quad (51)$$

Se um sinal, no domínio do tempo, tiver uma forma tal que:

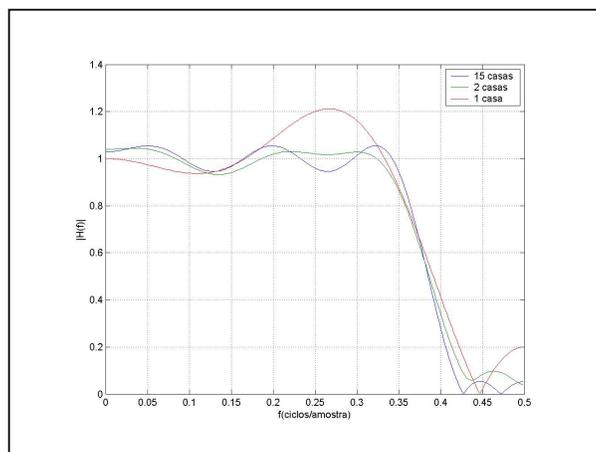
$$h(n) = \begin{cases} \frac{2\pi f_c}{\pi} & , & n = 0 \\ 2f_c \operatorname{sinc}(2f_c n), & n \neq 0 \end{cases} \quad (52)$$

Sua TDF será:

$$H(f) = \begin{cases} 1, & |f| < f_c \\ 0, & f_c < |f| < 0.5 \end{cases} \quad (53)$$

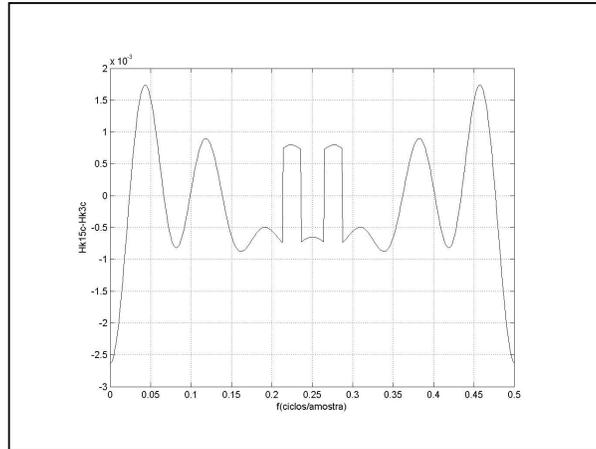
Onde  $f_c$  é a chamada frequência de corte do filtro digital. A partir da equação (53) um filtro digital do tipo passa-baixas, por exemplo, com a maior largura de banda possível será aquele que ofereça uma  $f_c = 0,5$  ciclos/amostra. Usando este valor de  $f_c$  na equação (52), é possível ser observado que a maior amplitude do  $h(n)$  correspondente será  $2 \times \pi \times f_c / \pi = 1,0$ . Como o cálculo da magnitude é feito em módulo, o outro extremo também deve ser contemplado, proporcionando a excursão de  $-1,0$  a  $+1,0$ .

A precisão adotada para a representação de  $h_{AG}(n)$  foi de três casas decimais. Quanto mais casas decimais forem utilizadas, maior será o tamanho do cromossomo resultante e mais tempo o AG levará para processar a resposta. O método a ser confrontado utiliza 15 casas decimais. Quanto menos casas decimais forem utilizadas, maior será a diferença obtida entre os resultados se fossem feitos com 15 casas decimais. Através de ensaios preliminares com um número de casas variáveis, chegou-se a resultados apresentados nas Figuras 16 e 17.



**Figura 16:** Distorções causadas por arredondamentos nos coeficientes de filtros.

Observou-se que os resultados com 1 e 2 casas decimais deformam a resposta em frequência quando comparados com 15 casas decimais, com pode ser visto na Figura 16.



**Figura 17:** Diferença entre  $H(f)|_{3\text{ casas}}$  e  $H(f)|_{15\text{ casas}}$ .

Resultados a partir de três casas decimais apresentaram um formato idêntico, somente variando a precisão do erro. Na Figura 17, a excursão do erro para três casas decimais é da ordem de  $10^{-3}$ . Um gráfico análogo para quatro casas decimais, por exemplo, pode ser obtido, porém com uma excursão da ordem de  $10^{-4}$ .

Porém, esta é uma característica que foi observada somente para os filtros FIR. Para filtros IIR, os pólos e zeros mudam de posição drasticamente e este critério não pode ser aplicado.

De posse da precisão  $prc$  valendo  $1 \times 10^{-3}$  e dos valores máximo  $Val_{máx} = +1,0$  e mínimo  $Val_{mín} = -1,0$ , é possível ser determinado através da equação (37), o tamanho do gene  $l'$  que é necessário para representar somente um coeficiente: *11 bits*.

Para ser obtido o tamanho  $l$  total do cromossomo que será usado no AG, uma variável  $L_2$  será definida:

$$L_2 = \begin{cases} \frac{M}{2} & , \quad M \text{ par} \\ \frac{M+1}{2} & , \quad M \text{ ímpar} \end{cases} \quad (54)$$

Onde:

- $L_2$  é uma variável auxiliar usada na obtenção do tamanho total do cromossomo;

- $M$  é o número de coeficientes especificados para o filtro.

Uma vez determinado o valor de  $L_2$ , o tamanho total do cromossomo poderá ser obtido a partir de:

$$l = l' \times L_2 \quad (55)$$

Onde:

- $l$  é o tamanho final do cromossomo em *bits*;
- $L_2$  é obtido a partir da equação (54);
- $l'$  ficou fixo neste trabalho em *11 bits* devido à precisão escolhida e aos valores máximo e mínimo que  $h_{AG}(n)$  poderá receber.

Foi utilizado o  $L_2$  devido às propriedades de simetria apresentadas nas equações (24) e (25). Portanto, o AG só precisa determinar metade dos coeficientes, pois os demais são obtidos pela simples repetição simétrica para os tipos I e II e repetição simétrica com troca de sinal para os tipos III e IV. Com isto o espaço de busca é reduzido à metade. O cromossomo que será utilizado pelo AG será então uma cadeia de  $l$  bits que representam, a cada onze deles, um coeficiente de  $h_{AG}(n)$ .

O espaço de busca pode ser determinado a partir da equação (55):

$$S = 2^l \quad (56)$$

Onde:

- $S$  é o espaço de busca; e
- $l$  é obtido a partir da equação (55).

Com base em (56), é constatado que a cada acréscimo de  $L_2$ , o espaço de busca é multiplicado por 2048. São possíveis espaços de busca que vão de, por exemplo,  $4.194 \times 10^6$  para  $M=4$ , a  $2.658 \times 10^{36}$  para  $M=21$ . Portanto, o método contempla um espaço de busca variável.

O mapeamento genótipo-fenótipo ficou:

- a) fenótipo variando entre  $-1,0$  a  $+1,0$  em passos de  $1 \times 10^{-3}$ , gerando 2000 números;
- b) quanto ao genótipo, é adotado o seguinte procedimento:
  - a. é tomado um gene do cromossomo, 11 bits. Destes 11 bits vem uma decodificação de um número que vai de zero a 2047 ( $2^{11}$ );
  - b. tal número decodificado recebe uma subtração de 1024, passando a representar um número entre  $-1024$  a  $+1023$ ;
  - c. este número é em seguida dividido por 1024, o que irá gerar números entre  $-1,0$  e  $+1,0$  com uma resolução de  $0,977 \times 10^{-3}$ ;
  - d. a decodificação prossegue para os demais genes até a conclusão do cromossomo.

Ao final, metade dos coeficientes de um filtro  $h_{AG}(n)$  terá sido obtida. A outra metade virá das propriedades de simetria proporcionada pelas equações (24) e (25).

Duas representações cromossômicas foram utilizadas para ser determinada qual a que oferecia melhores resultados: binário posicional e binário em Gray Refletido.

Será empregada a seguinte função-objetivo, a chamada MSE (*Mean Square Error*):

$$fobj = \sum_{i=0}^{N-1} \frac{1}{\left( |H_d(f_i)| - |H_i(f_i)| \right)^2} \quad (57)$$

Onde:

- $N$  é número de pontos do vetor resposta em frequência;
- $fobj$  é a função-objetivo;
- $|H_d(f)|$  é o módulo da resposta em frequência desejada; e
- $|H_i(f)|$  é o módulo da resposta em frequência do indivíduo que esteja sendo avaliado;

Esta função se baseia no erro quadrático para indicar quão próximo um indivíduo está do perfil de resposta em frequência que se procura. Em Öner (1998), por exemplo, foi empregada uma função parecida com esta, porém não era elevada ao quadrado. Em Karaboğa e Çetinkaya (2003) são empregadas duas funções de *fitness*: uma para a faixa de transição e outra para as faixas passante e de rejeição.

A função de *fitness*: o GALOPPS é uma ferramenta que aceita valores de *fitness* brutos, isto é, não normalizados entre 0,0 e 1,0. Da mesma forma, esta ferramenta também

provê técnicas de escalonamento de *fitness* brutas. Por este motivo, a função de *fitness* empregada no teste principal foi diretamente a função-objetivo.

Foram deixadas para a terceira fase duas verificações:

- a) uma tentativa de normalização da função de *fitness* para que os valores recaiam sempre dentro de uma faixa entre 0,0 e 1,0; e
- b) testes com escalonamentos da *fitness* bruta.

O mecanismo que será empregado na tentativa de normalização nos testes posteriores será baseado no que o MPM fornece para cada especificação. Isto é possível devido ao código-fonte do MPM ter sido adaptado para a linguagem C e também ter sido disponibilizado à comunidade científica (JANOVETZ, 1998). Isto também foi possível devido a existirem centenas de códigos na linguagem C gratuitos que calculam a resposta em frequência de um dado vetor através da TRF, sendo um deles a versão de Nielsen (2000). A partir destes dois códigos:

- a) é obtido o filtro de Remez para o caso da especificação que foi colocada;
- b) de posse deste filtro, é obtida a TRF do mesmo;
- c) com o valor da resposta em frequência de Remez,  $|H_r(f)|$ , é utilizada a equação (57) para ser obtido o valor de MSE que o filtro de Remez oferece para o caso específico que foi calculado;
- d) Este número que representa o erro quadrático de *Remez*,  $E_R$  será um parâmetro que servirá para normalizar a função de *fitness*.

Com base no exposto, a função de *fitness* que será utilizada em um dos testes pertencentes à terceira fase será:

$$fitness = \frac{fobj}{E_R} \quad (58)$$

Onde:

- *fitness* é a função de *fitness*; e
- $E_R$  é o erro quadrático do MPM obtido através da substituição da sua resposta em frequência  $|H_R(f)|$  no lugar de  $|H_i(f)|$  na equação (57).

Para a modelagem proposta para o AG principal foi feita a opção por não serem empregadas penalidades.

### 3.2.8 Limitações

Embora o experimento esteja preparado para resolver casos de filtros com amplitudes arbitrárias, como pode ser observado pelo banco de filtros da Tabela 4 proposto para o teste principal, estão sendo investigados em detalhe somente os filtros com amplitudes fixas em 0,0 ou 1,0.

Foi mantido fixo em 512, para este estudo, o número de pontos coletados,  $N$ . O código-fonte desenvolvido, no entanto, tem este valor sob a forma de constante de pré-processador, podendo ser alterada. Como a equação (57) é sensível ao número de pontos, isto significa que os resultados podem ser alterados para casos diferentes deste valor.

Não foi equacionado no conjunto de variáveis independentes o desempenho do *software*, isto é, face ao caráter exploratório o que foi priorizado a ser descoberto foi um método que conduza a um *software* que consiga suportar um número máximo de especificações que não requeira interação com o usuário.

A proposta do experimento apóia-se também em uma hipótese secundária que defende a idéia da equação (31) não precisar ser obedecida obrigatoriamente para ser atingida a qualidade especificada pelos *ripples* e largura da faixa de transição que lhe serviram de argumentos. Certamente, se o MPM for executado com um número de coeficientes recomendado pela equação (31), sua qualidade será praticamente imbatível. A motivação de serem investigados valores de  $M$  abaixo da recomendação da equação (31), vem do fato de que, quanto maior o número de coeficientes, maior será o atraso para o filtro processar um sinal. Logo, soluções que forneçam valores de  $M$  abaixo do recomendado e que mantenham uma qualidade, são úteis. E é esta a grande região de operação onde é esperada a contribuição do AG.

A interface do *software* proposto e sua forma de trabalho são mais apropriadas a fins de depuração do que de uso final.

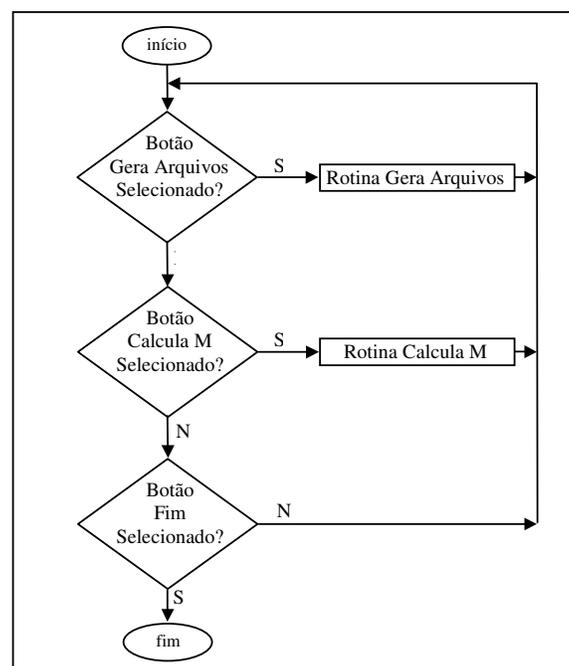
O limite mínimo de intervalo de freqüências estudado ficou em 0,01 ciclos por amostra.

### 3.2.9 Os softwares

São três *softwares* que são executados em WINDOWS, todos com interfaces gráficas.

#### 3.2.9.1 O software Especificador

O *software Especificador* apresentado na subseção 3.2.1 segue um conjunto de ações que está representado no fluxograma da Figura 18.

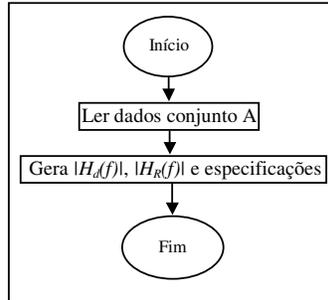


**Figura 18:** Fluxograma do *software Especificador*.

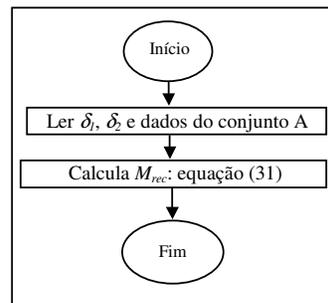
O fluxograma da Figura 18 faz referência a duas rotinas cujos fluxogramas estão nas Figuras 19 e 20.

Das figuras 18, 19 e 20:

- a) a leitura dos dados do conjunto A consiste na aquisição simples (via teclado) da especificação da *respost* em frequência desejada e demais parâmetros do conjunto A apresentado na subseção 3.2.2, que o filtro FIR a ser gerado pelo AG deve buscar aproximar-se;



**Figura 19:** Fluxograma da rotina de geração de arquivos.

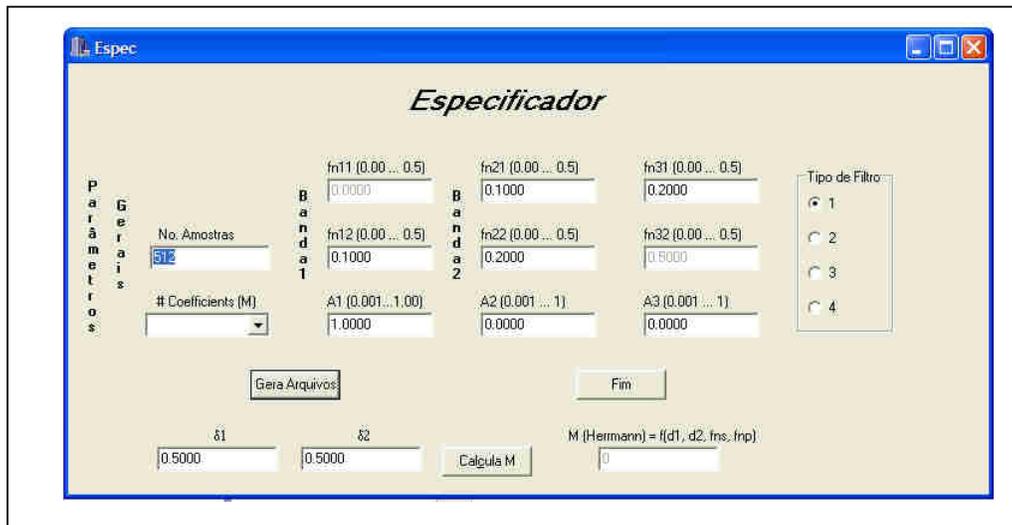


**Figura 20:** Fluxograma da rotina do cálculo do M (recomendado).

- b) o teste das consistências é a verificação se a especificação solicitada condiz com as restrições relativas aos tipos de filtros apresentada na subseção 2.1.6;
- c) a geração da resposta em frequência ideal desejada  $|H_d(f)|$  consiste basicamente na geração de um vetor com N (fixado em 512) pontos unindo as faixas 1, 2 e 3 através de segmentos de retas;
- d) para fins comparativos (no AG e no *software RespFreq*), é gerada uma resposta em frequência  $|H_R(f)|$ , que é a resposta do MPM. Para tal, foi utilizado um código pronto (JANOVETZ, 1998) que fornece este vetor a partir das especificações (conjunto A) recebidas;
- e) dados e especificações auxiliares gerados são as demais informações a serem coletadas pelo AG para processamentos adicionais, por exemplo, o  $E_R$  da equação (58);
- f) a resposta em frequência ideal, os dados auxiliares e a resposta em frequência do MPM são, respectivamente, salvos nos arquivos `f_ref.txt` e `specs.txt` e `remez.txt`.
- g) caso seja solicitado, o número de coeficientes recomendado  $M_{rec}$ , pode também ser calculado. Isto é feito através da rotina apresentada na Figura

20, que contém a equação (31) apresentada na subseção 2.1.8, usando como argumentos a diferença entre as faixas passante e de rejeição (que pode ser obtida a partir das especificações recebidas) e de dois valores de ripples,  $\delta_1$  e  $\delta_2$ , que sejam informados.

A implementação do *software Especificador* foi em BORLAND C++ BUILDER (BORLAND, 2000) e sua interface está apresentada na Figura 21:



**Figura 21:** Interface do *software Especificador*.

### 3.2.9.2 O *software* AG

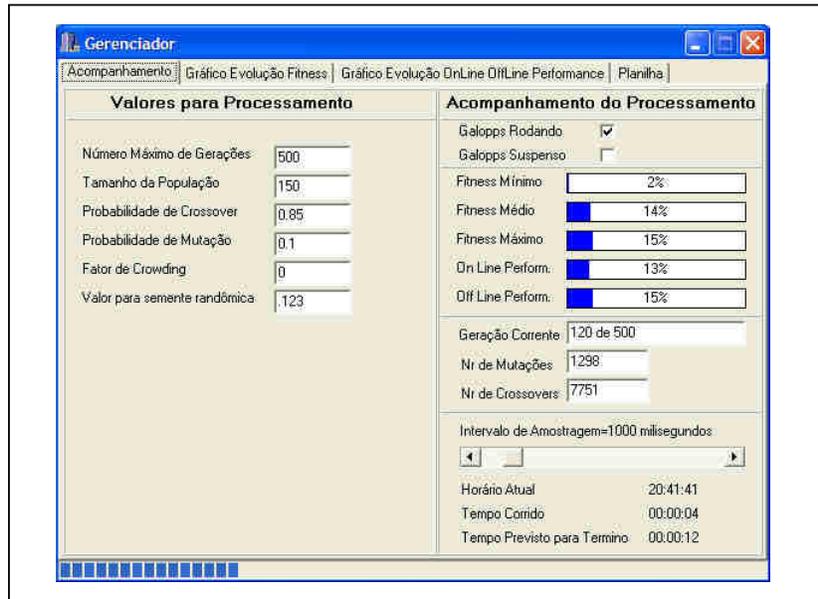
Um *software* para AG que possibilite implementar o método proposto deve conter o que o fluxograma da Figura 13 apresenta, acrescido de uma série de variações, recursos e conceitos atualizados sobre AGs. Entre tais recursos, são indispensáveis:

- ler arquivos de entrada (antes da execução);
- gerar informações via arquivo entre cada geração e no final da execução;
- receber as diversas combinações propostas pelos conjuntos *C*, *D* e *E*; e
- informar dados estatísticos sobre o andamento dos resultados a cada geração.

A implementação deste *software* foi feita com o GALOPPS (MSU GARAGE, 2002). A versão disponível para distribuição é para ser executada no modo Dos/Console. Esta foi a

versão utilizada na fase de desenvolvimento. Porém para a versão final, foi empregada uma versão do GALOPPS adaptada para o WINDOWS, com uma interface gráfica. Esta adaptação foi feita por Geraldo Correia e Edson Bobel disponível no *link* de Recursos da Disciplina de CE da página do Prof. Dr. Heitor Silvério Lopes<sup>4</sup>.

Uma das telas de interface desta versão pode ser vista na Figura 22.



**Figura 22:** Interface do *software* AG.

O GALOPPS é uma ferramenta de apoio ao desenvolvimento de AGs, gratuita, com o seu código-fonte em aberto e escrito na linguagem C. Esta ferramenta permite ao programador concentrar-se em desenvolver as seguintes ações:

- a) decodificar o cromossomo de *bits* (podendo ainda ser outro alfabeto) para seus valores numéricos correspondentes; e
- b) calcular a função de *fitness*, que é chamada de *fitness* bruta.

Estas duas ações são feitas para cada indivíduo da população sendo processada a cada geração. Isto é feito em uma função chamada `objfunc()`, que fica dentro de um arquivo cujo nome original de distribuição é `appxxxxx.c`, mas que pode ser alterado.

O restante das ações, determinação do tamanho da população, do número máximo de gerações a serem processadas, qual operação de cruzamento, qual operação de mutação, qual

<sup>4</sup> <http://bioserver.cpegei.cefetpr.br/disciplinas/ce>

método de seleção, quais mecanismos para que a *fitness* receba escalonamentos, entre diversos outros recursos, são fornecidos e com várias opções a escolher. Basta selecionar o que se quer e utilizar dentro de arquivos que vêm com as opções de escolha. Estas opções estão em dois arquivos, que são o MAKEFILE e outro cujo nome original de distribuição é `appone.in`. Caso não se queira trabalhar com este segundo arquivo, é possível serem passados os para AG em modo interativo.

A decodificação do cromossomo de *bits* para o  $h(n)$  correspondente foi realizada através de uma função também pronta e fornecida pelo GALOPPS, a `ithruj2int`, que foi parametrizada para fornecer, a cada 11 *bits*, um número inteiro, de zero a 2048, que será um coeficiente de  $h(n)$ , de acordo com a seqüência de mapeamento genótipo-fenótipo apresentada na subseção 3.2.7 (modelagem).

Para os casos em que está configurada a representação em binário Gray, é preciso aplicar-se uma lógica XOR ao inteiro convertido pela função citada no parágrafo anterior. Esta lógica é realizada com dois operadores da linguagem C: `^` e `>>` (PRESS, 2002).

A função de *fitness*, apresentada na equação (57), é calculada para cada indivíduo, decodificado em um  $h(n)$  correspondente, a cada geração que vai sendo executada.

Se a equação (8) da TDF fosse aplicada como está, o tempo de processamento iria ultrapassar 8 horas no computador apresentado na subseção 3.1.1 para um filtro com somente 19 coeficientes. Para resolver este problema, foi utilizado em seu lugar um algoritmo que calcula a TRF (NIELSEN, 2000) no lugar da TDF.

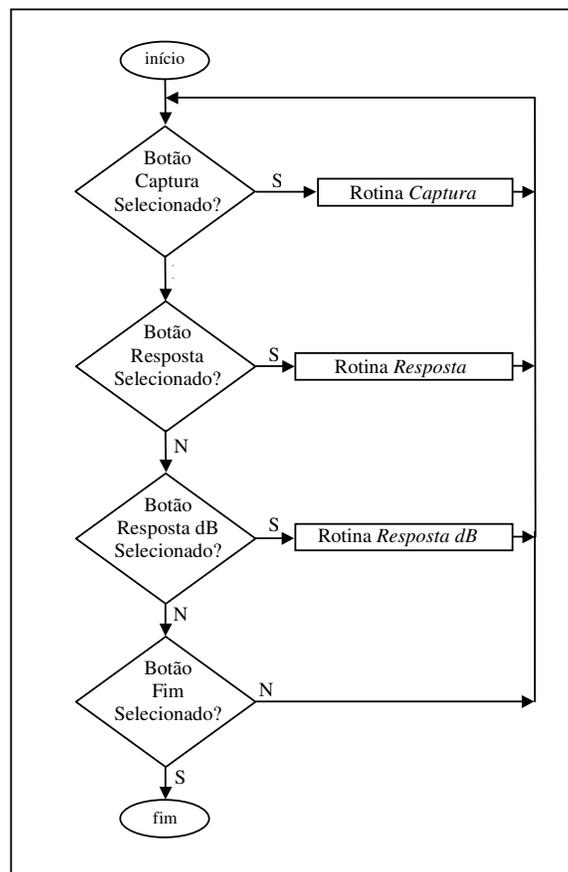
### 3.2.9.3 O software *RespFreq*

O software *RespFreq* contém um algoritmo que está representado através do fluxograma da Figura 23. Nesta figura, são chamadas rotinas cujos fluxogramas estão apresentados nas figuras 24, 25 e 26.

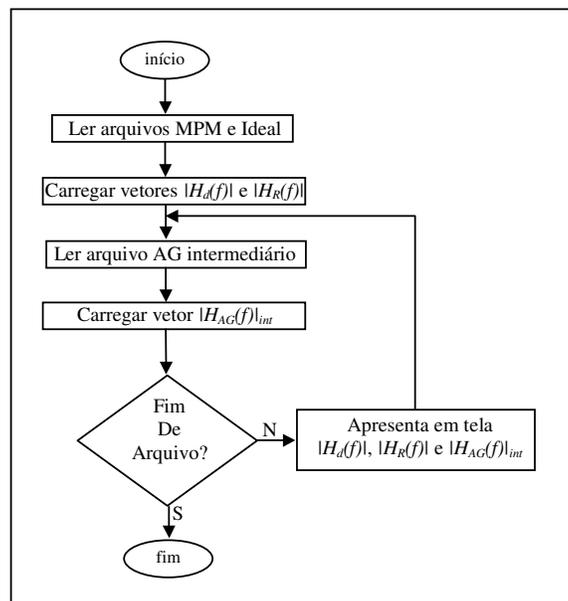
O software funciona de acordo com o botão que for selecionado, podendo apresentar resultados durante a execução ou após o seu término.

O botão responsável por executar tarefas durante o momento em que o software AG esteja sendo executado é o *Captura*, que faz aparecer na tela, a cada geração, o melhor indivíduo até o momento encontrado pelo AG.

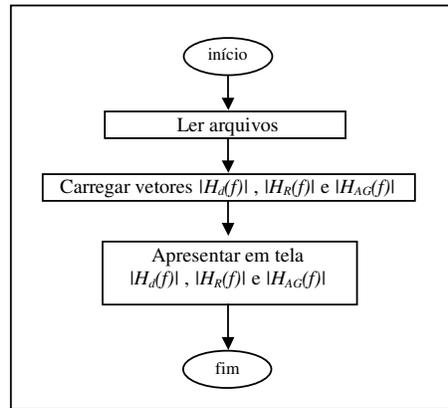
Os demais botões são destinados a apresentarem a resposta em frequência do melhor indivíduo encontrado, comparando sua qualidade com a obtida pelo MPM.



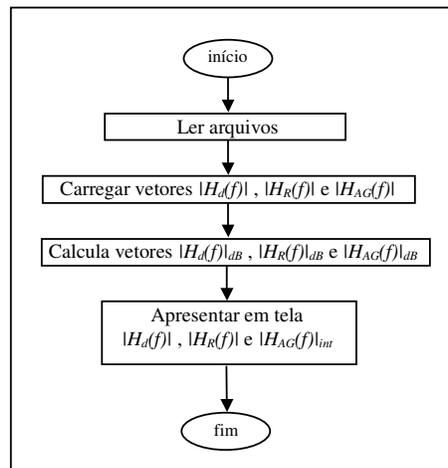
**Figura 23:** Fluxograma do *software RespFreq*.



**Figura 24:** Fluxograma da rotina *Captura*.



**Figura 25:** Fluxograma da rotina *Resposta*.



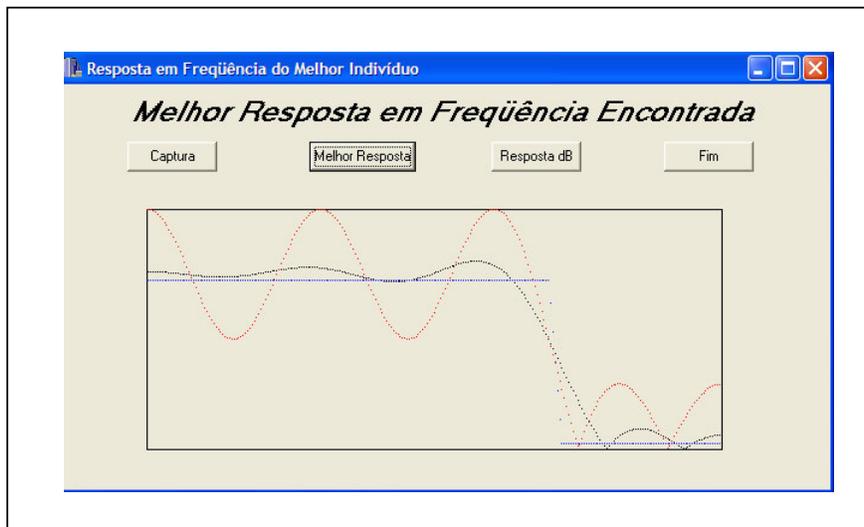
**Figura 26:** Fluxograma da rotina *Resposta dB*.

Das figuras 23 à 26:

- a) ao ser selecionado o botão *Captura*, são lidos os arquivos contendo a resposta em frequência do MPM e a resposta em frequência do filtro ideal gerados pelo *software Especificador*;
- b) os conteúdos de tais arquivos são transferidos para seus respectivos vetores,  $|H_R(f)|$  e  $|H_d(f)|$ ;
- c) em seguida é aberto o arquivo de resposta em frequência intermediária que é gerado pelo *software AG* e é constantemente por ele atualizado;
- d) é carregado um vetor de resposta em frequência intermediária,  $|H_{AG}(f)|_{int}$ , com a  $i$ -ésima linha deste arquivo;
- e) são apresentados em tela os três vetores obtidos nos itens b) e d);
- f) é novamente carregado o vetor  $|H_{AG}(f)|_{int}$  com o próximo conjunto de resposta em frequência intermediária do arquivo citado no item c);

- g) repete-se a etapa citada no item e);
- h) repetem-se as etapas f) e g) até que seja encontrado o final do arquivo de resposta em frequência intermediária;
- i) quando isto ocorrer, a rotina *Captura* é encerrada;
- j) ao ser selecionado o botão *Resposta*, são lidos os arquivos citados no item a) e também é lido o arquivo com a resposta em frequência do melhor indivíduo;
- k) os conteúdos de tais arquivos são transferidos para seus respectivos vetores,  $|H_R(f)|$ ,  $|H_d(f)|$  e  $|H_{AG}(f)|$ ;
- l) são apresentados em tela os vetores obtidos na etapa anterior;
- m) ao ser selecionado o botão *Resposta dB*, são processadas de forma semelhante às etapas j) k);
- n) são obtidos os vetores em decibéis através da relação  $|H(f)|_{dB} = 20 \times \log(|H(f)|)$ ; e
- o) são apresentados em tela os vetores obtidos na etapa anterior.

A implementação do *software* RespFreq foi também feita em Borland C++ (BORLAND, 2000) e sua interface pode ser vista na Figura 27. Nesta figura  $|H_d(f)|$  está em azul,  $|H_R(f)|$  está em vermelho e  $|H_{AG}(f)|$  em preto.



**Figura 27:** Interface do *software* RespFreq.

## CAPÍTULO 4

### RESULTADOS

Serão apresentadas neste capítulo amostras representativas dos testes realizados. As siglas que referenciam os filtros correspondem aos seus respectivos nomes na Tabela 4 do Banco de Filtros.

#### 4.1 PRIMEIRA FASE

É o conjunto de testes que foram propostos na subseção 3.2.6.1.

##### 4.1.1 Configurações utilizadas

O conjunto  $A$  ficou disposto conforme segue:

- a)  $f_{n11}$ ,  $f_{n12}$ ,  $f_{n21}$ ,  $f_{n22}$ ,  $f_{n31}$ ,  $f_{n32}$ ,  $A_1$ ,  $A_2$  e  $A_3$  variando seus valores de acordo com as especificações do banco de filtros apresentado na Tabela 4;
- b) Tipos = I, II, III e IV;
- c)  $M = 15$  para os tipos I e III e  $M=16$  para os tipos III e IV. Estes números de coeficientes fornecem, de acordo com a equação (54), um  $L_2=8$ , o que levado à equação (55) fornece um comprimento do cromossomo  $l$  de 88 *bits*, que por sua vez, se levado à equação (56), fornece um espaço de busca  $S$  de aproximadamente  $3,095 \times 10^{26}$ .

O conjunto  $C$  ficou disposto conforme segue:

- a) *seleção*, *cruzamento*, *mutação* e  $rc$  variando seus valores de acordo com a subseção 3.2.2;
- b) função de *fitness* = equação (57);

O conjunto  $D$  ficou disposto conforme segue:

- a) *elitismo* = sim/não;
- b) *pcross* = 0,85; *pmutation* = 0,10 (conforme critérios de redução apresentados na subseção 3.2.5);
- c) *maxgen* = 500, 750 e 1000;
- d) *popsiz*e = 160;
- e) *semente* = valores aleatórios entre 0,000 e 0,999;

O conjunto  $E$  ficou com seu parâmetro de seleção *par\_sel* = 4 quando o método *tselect* for selecionado e *par\_scal* = vazio (sem escalonamento, conforme critérios de redução apresentados na subseção 3.2.5).

#### 4.1.2 Resumo dos resultados

A Tabela 6 resume algumas notas obtidas por configurações do AG variando-se os possíveis valores dos conjuntos  $C$ ,  $D$  e  $E$  conforme apresentado nesta subseção.

**Tabela 6:** Notas para a primeira fase.

$N^o$	Tipo	seleção	maxgen	cruzamento	mutação	rc	elitismo	$t_{AG}$	Nota
1	I	rselect	500	oneptx	bitmutat	BinPos	não	11s	0,00
2	I	suselect	500	twoptx	bitmutat	BinGray	sim	11s	5,00
3	II	suselect	500	twoptx	bitmutat	BinGray	sim	11s	0,00
4	II	suselect	750	twoptx	bitmutat	BinGray	sim	17s	0,00
5	II	suselect	1000	twoptx	bitmutat	BinGray	sim	24s	5,00
6	III	suselect	500	twoptx	bitmutat	BinGray	sim	11s	7,81
7	IV	suselect	500	twoptx	bitmutat	BinGray	sim	11s	5,42
8	I	suselect	500	Twoptx	bitmutat	BinGray	não	24s	0,00
9	I	tselect/4	500	Twoptx	multimut	BinPos	sim	11s	0,00
10	I	tselect/4	500	Twoptx	bitmutat	BinGray	sim	11s	0,00
11	I	tselect/4	500	Twoptx	multimut	BinGray	sim	11s	0,00

Cada um dos registros da Tabela 6 corresponde a uma planilha preenchida de acordo com a Tabela 5. A Tabela 7 apresenta as avaliações feitas para a geração do registro N<sup>o</sup> 4 da Tabela 6.

**Tabela 7:** Planilha do teste principal preenchida.

	<i>popsize</i>	750	<i>M</i>	16	<i>Tipo II</i>	<i>maxgen</i>	750
	<i>tempo</i>	17 s					
	<i>seleção</i>	<i>suselect</i>		<i>cruzamento</i>	<i>twoptx</i>	<i>Mutação</i>	<i>bitmutat</i>
	<i>fitness</i>	<i>MSE</i>		<i>rc</i>	<i>BinGray</i>	<i>Elitismo</i>	<i>Sim</i>
N <sup>o</sup>	<i>Filtro</i>	0.123	0.456	0.789	0.012	<i>Observações</i>	
1	fpba	NOK	SS	NOK	NOK		
2	fpb1	NOK	NOK	NOK	OK		
3	fpb2	NOK	NOK	NOK	NOK	fpb que causou a nota zero	
4	fpb3	NOK	SS	SS	SS		
5	fpb4	OK	OK	NOK	NOK		
6	fpa1	NA	NA	NA	NA	NA = Não Aplicável	
7	fpa2	NA	NA	NA	NA	Não aplicável por ser filtro Tipo II	
8	fpa3	NA	NA	NA	NA	(veja subseções 2.1.6 e 3.2.6.1)	
9	fpa4	NA	NA	NA	NA		
10	fpa5	NA	NA	NA	NA		
11	fpf1	OK	OK	OK	SS		
12	fpf2	NOK	SS	NOK	OK		
13	fpf3	OK	OK	OK	OK		
14	fpf4	NOK	OK	NOK	OK		
15	fcf1	NA	NA	NA	NA		
16	fcf2	NA	NA	NA	NA		
17	fcf3	NA	NA	NA	NA		
18	fcf4	NA	NA	NA	NA		
19	fcf5	NA	NA	NA	NA		
	N <sup>o</sup> OK	13	N <sup>o</sup> SS	7	Nota	Zero	

#### 4.1.3 Exemplos de resultados NOK, SS e OK

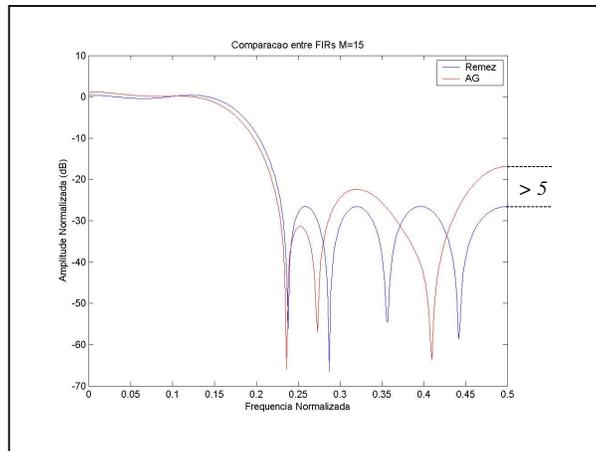
Serão apresentados nesta subseção três exemplos de resultados de resposta em frequência para ser feita a demonstração dos qualificadores NOK e SS apresentados nas subseções 3.2.3 e 3.2.6.1.

A Figura 28 apresenta um exemplo de resultado NOK por não ter atendido à PCRSO. Trata-se do resultado gerado para o filtro *fpb1* (do Banco de Filtros – Tabela 4), com semente 0,789 durante os testes correspondentes ao registro N<sup>o</sup> 2 da Tabela 6.

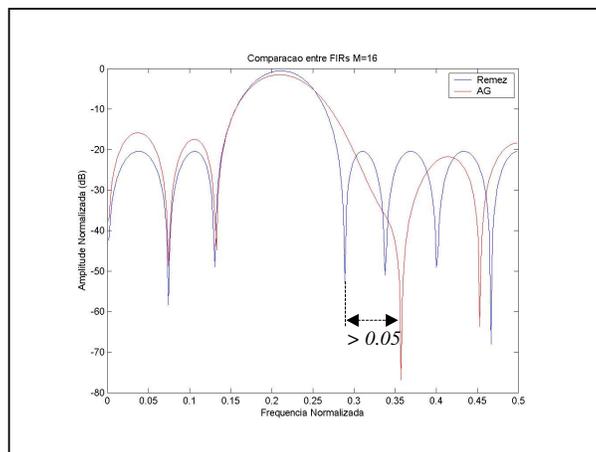
A Figura 29 apresenta um exemplo de resultado NOK por não ter atendido à SCRSO. Trata-se do resultado gerado para o filtro *fpb3*, com semente 0,123 durante os testes correspondentes ao registro N<sup>o</sup> 7 da Tabela 6.

A Figura 30 refere-se ao mesmo registro do parágrafo anterior, porém com uma semente 0,456 gerando um resultado que atendeu às PSCRSO, sendo um exemplo de resultado SS. Devido ao filtro AG ter apresentado um valor de *ripple* superior ao oferecido pelo filtro MPM (próximo a 0,05 ciclos/amostra), este exemplo não chega a ser um resultado do tipo OK.

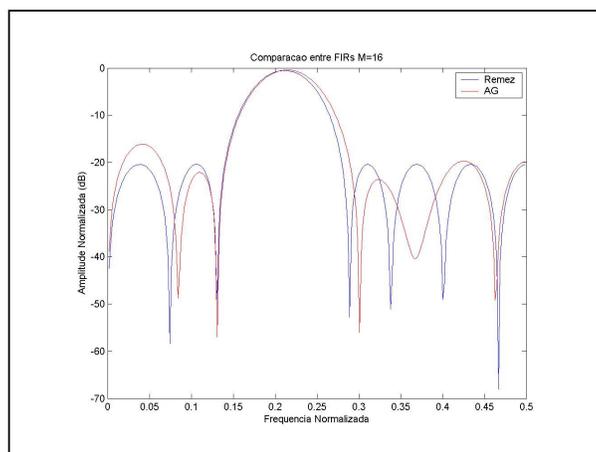
As Figuras 31 e 32 apresentam um exemplo de resultado OK por ter atendido às PSCRSO e também ter apresentado todos os *ripples* menores do que os *ripples* do MPM. Trata-se do resultado gerado para o filtro *fpb1*, com semente 0,012 durante os testes correspondentes ao registro N<sup>o</sup> 2 da Tabela 7.



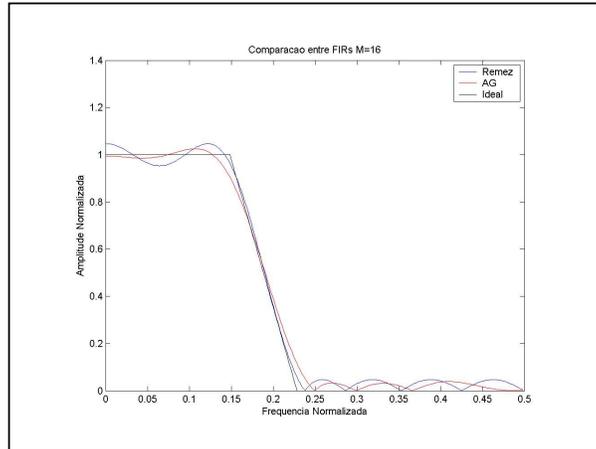
**Figura 28:** Primeiro exemplo de um resultado NOK.



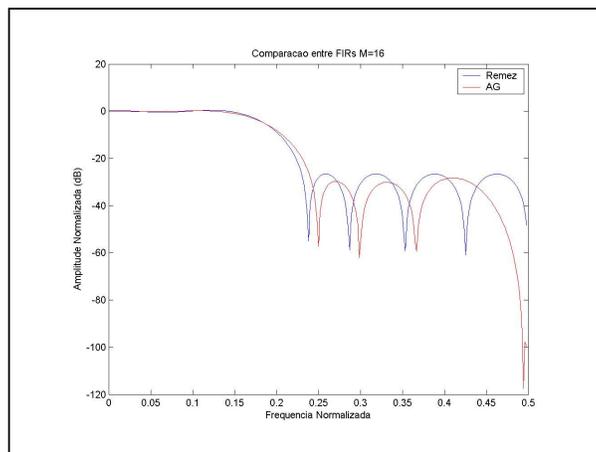
**Figura 29:** Segundo exemplo de resultado NOK.



**Figura 30:** Exemplo de resultado SS.



**Figura 31:** Um exemplo de resultado OK.



**Figura 32:** Resposta da Figura 31 em decibéis.

## 4.2 SEGUNDA FASE

São os testes que foram propostos na subseção 3.2.6.2. Para que tais testes fossem realizados, deveria restar pelo menos uma configuração de parâmetros AG definida pelos conjuntos C, D e E, que conseguisse passar pela primeira fase, isto é, obtivesse uma nota diferente de zero. A configuração que restou foi a seguinte:

Conjunto C:

- a) *seleção* = *suselect*, *cruzamento* = *twoptx*, *mutação* = *bitmutat* e *rc* = *BinGray*;
- b) função de *fitness* = equação (57);

Conjunto D:

- a) *elitismo* = *sim*;
- b) *pcross* = 0,85; *pmut* = 0,10;
- c) *maxgen* = 500, 750 e 1000;
- d) *popsiz*e = 160;
- e) *semente* = valores aleatórios entre 0,000 e 0,999;

O conjunto E ficou com seu parâmetro de seleção *par\_sel* = *vazio* (sem parâmetros para o método de seleção) e *par\_scal* = *vazio* (sem escalonamento, conforme critérios de redução apresentados na subseção 3.2.5).

Portanto, todos os testes realizados nesta fase foram feitos com configurações de parâmetros AG dispostos conforme apresentado nesta subseção.

Como esta fase visa resolver casos com um número variável de coeficientes do filtro, o espaço de busca irá também ser variável da mesma forma, de acordo com a equação (56). Com esta variação, os parâmetros que foram escolhidos da configuração restante para receberem ajustes são: o número máximo de gerações, *maxgen*, e o tamanho da população, *popsiz*e.

### 4.2.1 Resumo dos resultados

A Tabela 8 resume algumas notas obtidas pela configuração que foi classificada na primeira fase.

**Tabela 8:** Notas para a os testes da segunda fase.

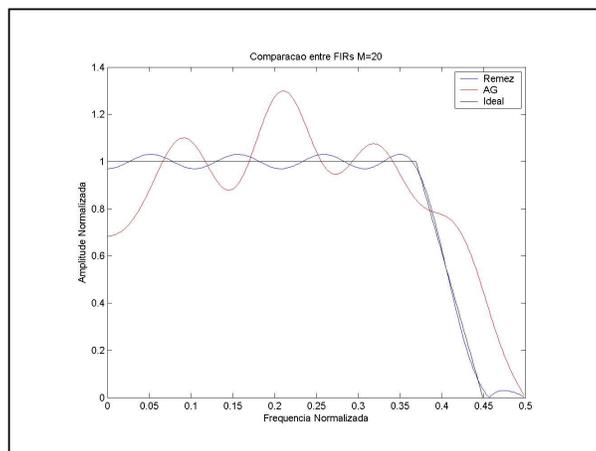
Nº	popsiz	maxgen	Tipo	M	S	t <sub>AG</sub> (s)	Nota
1	44	500	I	13	151,1×10 <sup>21</sup>	4	0.00
2	50	500	I	13	151,1×10 <sup>21</sup>	4	5.46
3	480	500	II	18	633,8×10 <sup>27</sup>	29	3.06
4	608	500	I	17	633,8×10 <sup>27</sup>	36	4.54
5	608	500	II	18	633,8×10 <sup>27</sup>	36	3.75
6	2310	1500	II	20	1,298×10 <sup>33</sup>	410	4.86

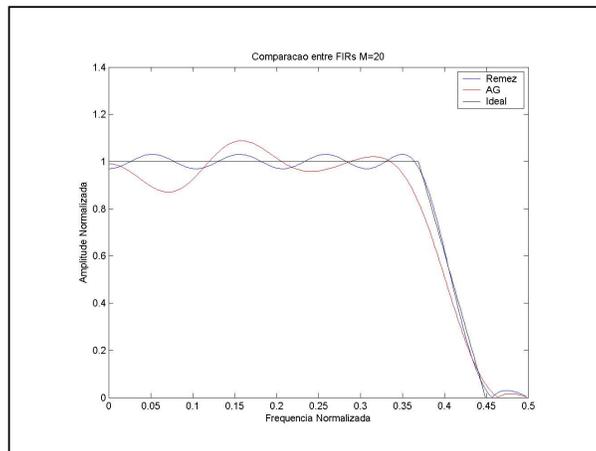
#### 4.2.2 A influência dos parâmetros sobre os resultados

As figuras 33 e 34 correspondem a testes que antecederam a um dos testes que forneceram dados para o registro N<sup>o</sup> 6 da Tabela 8, isto é, com as mesmas especificações. Tais testes foram feitos com um valor de *maxgen* abaixo do utilizado no referido registro.

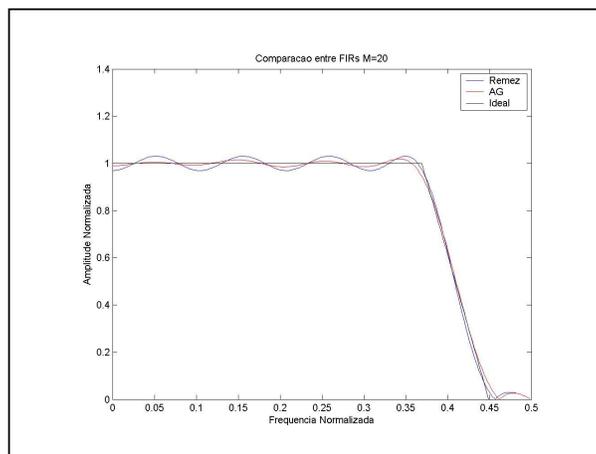
A Figura 35 corresponde ao resultado de um dos testes obtidos para a formação do registro N<sup>o</sup> 6 da Tabela 8.

Testes preliminares também foram feitos com valores de *popsiz* abaixo do que foi utilizado durante os testes para a formação do registro N<sup>o</sup> 8 da Tabela 8. Todos receberam nota zero.

**Figura 33:** Resposta NOK com *maxgen* = 500.



**Figura 34:** Resposta SS com  $maxgen = 750$ .

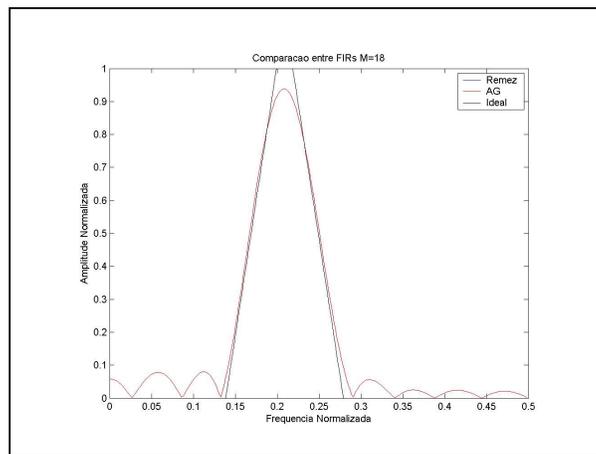


**Figura 35:** Resposta OK com  $maxgen = 1500$ .

### 4.2.3 Resultado relevante de possível aplicação do AG

Durante a realização dos testes da segunda fase, um tipo de resultado se destacou por sua característica: o AG conseguiu fornecer respostas para determinados tipos de especificações em que MPM não conseguiu.

A Figura 36 apresenta um filtro passa-faixa estreito, o *fpf3*, em uma configuração com  $M=18$ ,  $maxgen = 1500$ ,  $popsize = 200$ ,  $semente = 0,123$  e  $t_{AG} = 24 s$ , que o MPM não conseguiu fornecer uma solução. Por este motivo não é possível ser visualizada a curva em azul na figura.



**Figura 36:** Filtro passa-faixa estreito que o MPM não conseguiu gerar.

### 4.3 TERCEIRA FASE

São os testes que foram propostos na subseção 3.2.6.3. A configuração que serviu de base para os teste foi a mesma da segunda fase com os respectivos ajustes de tamanho de população e número máximo de gerações. Esta configuração será chamada de configuração básica nos testes que serão realizados.

### 4.3.1 Teste da superposição

Este teste verifica a capacidade do AG em atender ao princípio da superposição devido a filtros FIR serem sistemas do tipo LTI, conforme apresentado na subseção 2.1.3.

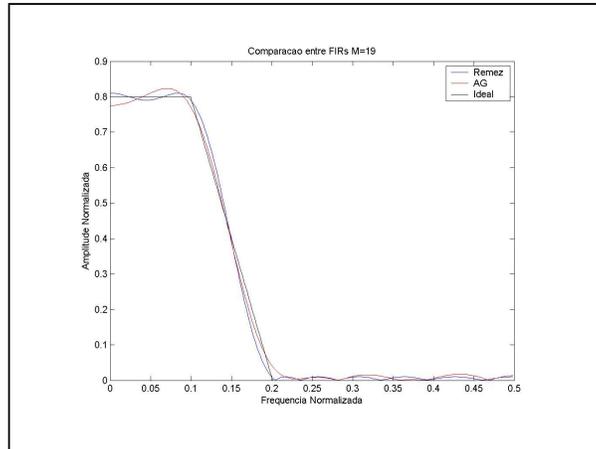
Uma interpretação do teorema da superposição para filtros pode ser a que se segue:

- a) considerar um filtro do tipo passa-baixas,  $h_1(n)$ , com uma resposta em frequência  $|H_1(f)|$ ;
- b) considerar um filtro do tipo passa-altas,  $h_2(n)$ , com uma resposta em frequência  $|H_2(f)|$ ;
- c) considerar um filtro do tipo corta-faixa,  $h(n)$ , com uma resposta em frequência  $|H(f)|$ ;
- d) se  $|H(f)| = |H_1(f)| + |H_2(f)|$ , então  $h(n) = h_1(n) + h_2(n)$ .

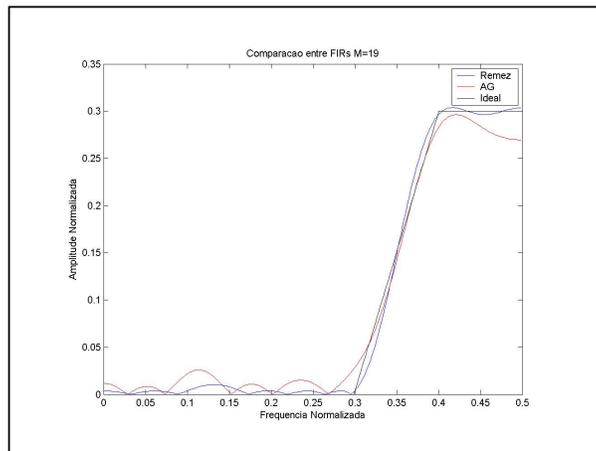
Portanto, este teste irá fazer com que o AG forneça os vetores  $h_1(n)$  e  $h_2(n)$ . Em seguida tais vetores serão somados e será obtida a TDF do vetor-soma. Este resultado deverá corresponder a um filtro corta-faixa cujas especificações sejam a união das especificações dos dois filtros que o originaram. Foram empregados nos testes filtros Tipo I com  $M=19$  coeficientes. A configuração do AG utilizada, conjuntos C, D e E, foi a adotada na segunda fase com  $maxgen = 500$  e  $popsiz = 2132$ .

As especificações, conjunto A, do filtro que deverá originar o  $h_1(n)$  são: a)  $f_{n11} = 0,0$ ;  $f_{n12} = 0,1$ ;  $A_1 = 0,8$ ; b)  $f_{n21} = 0,2$ ;  $f_{n22} = 0,3$ ;  $A_2 = 0,0$ ; e c)  $f_{n31} = 0,4$ ;  $f_{n32} = 0,5$ ;  $A_3 = 0,0$ . E as especificações, conjunto A, do filtro que deverá originar o  $h_2(n)$  são: a)  $f_{n11} = 0,0$ ;  $f_{n12} = 0,1$ ;  $A_1 = 0,0$ ; b)  $f_{n21} = 0,2$ ;  $f_{n22} = 0,3$ ;  $A_2 = 0,0$ ; e c)  $f_{n31} = 0,4$ ;  $f_{n32} = 0,5$ ;  $A_3 = 0,3$ . Observar que estas especificações podem ser unidas em uma especificação de um filtro corta-faixa.

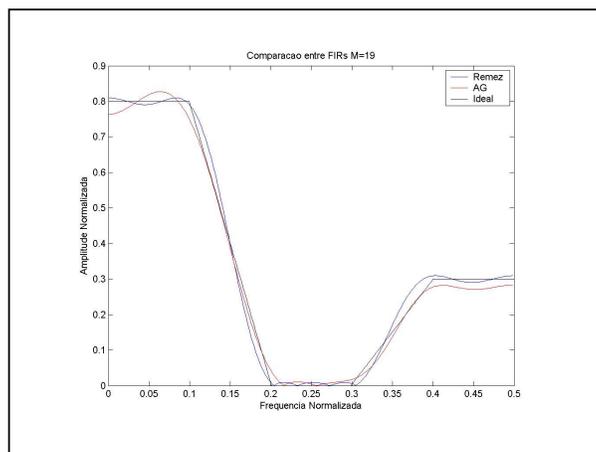
As figuras 37 e 38 apresentam, respectivamente as respostas  $|H_1(f)|$  e  $|H_2(f)|$  que o AG forneceu. A Figura 39 apresenta a TDF, obtida via MATLAB, de  $|H(f)|$ .



**Figura 37:**  $|H_1(f)|$  do teste da subseção 4.3.1.



**Figura 38:**  $|H_2(f)|$  do teste da subseção 4.3.1.



**Figura 39:**  $|H(f)|$  do teste da subseção 4.3.1.

#### 4.3.2 Teste da normalização da *fitness*

Este teste substitui a função de *fitness* da equação (57), na configuração básica, pela equação (58), com o objetivo de normalizar os resultados entre 0,0 e 1,0. Isto foi idealizado tendo-se levado em consideração a suposição de que o erro quadrático fornecido pelo MPM fosse sempre menor do que o fornecido pelo AG.

Foram realizados alguns testes e o que se observou foi que, para alguns deles, o erro quadrático médio do AG foi menor do que o do MPM. Isto fez com que o valor da função de *fitness* continuasse apresentando valores superiores a 1,0. Um exemplo deste tipo de resultado foi a repetição do teste correspondente à obtenção do filtro *fpf3*, com semente 0,789, dentro dos testes realizados para formar o registro N<sup>o</sup> 8 da Tabela 8, porém com a equação (58) ao invés da (57).

#### 4.3.3 Evolução das curvas de *fitness*

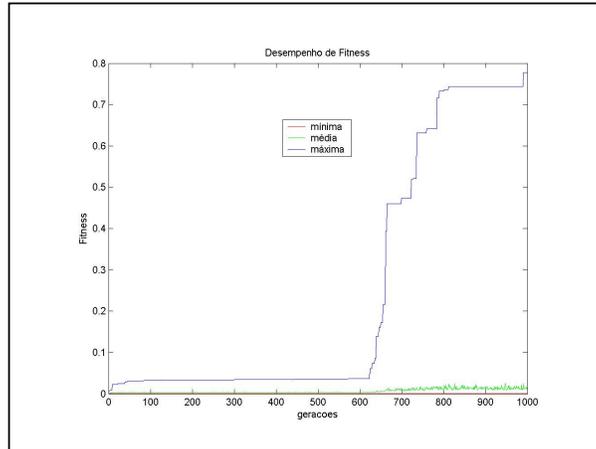
Esta subseção apresenta o perfil típico das curvas de *fitness* quando o AG encontra soluções OK ou SS.

As figuras 40, 41 e 42 referem-se ao resultado OK obtido para o filtro *fpba* (do Banco de Filtros – Tabela 4), com semente 0,123 durante os testes que geraram o registro N<sup>o</sup> 2 da Tabela 6.

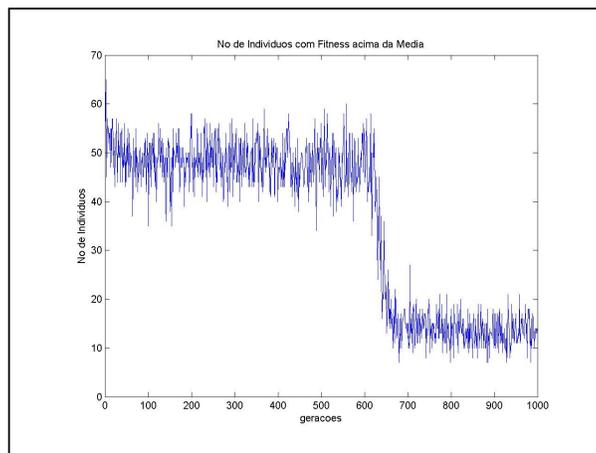
A Figura 40 é a evolução dos valores de *fitness* máximos, médios e mínimos.

A Figura 41 apresenta o número de indivíduos com *fitness* acima da média a cada geração.

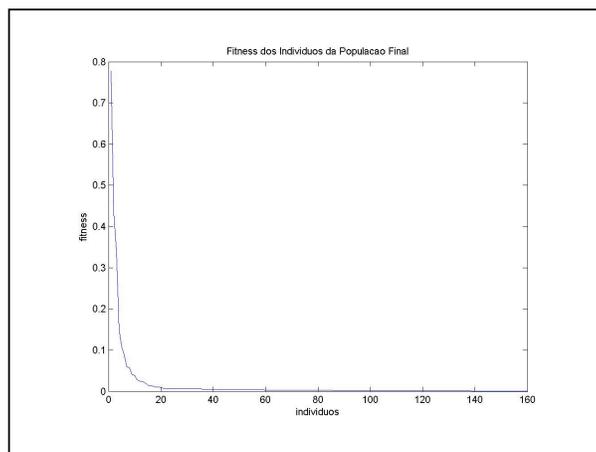
E a Figura 42, apresenta os valores de *fitness* para cada um dos 160 indivíduos da população final em ordem decrescente.



**Figura 40:** Curvas de *fitness* para um resultado do registro 2 da Tabela 6.



**Figura 41:** Número de indivíduos com *fitness* acima da média por geração.

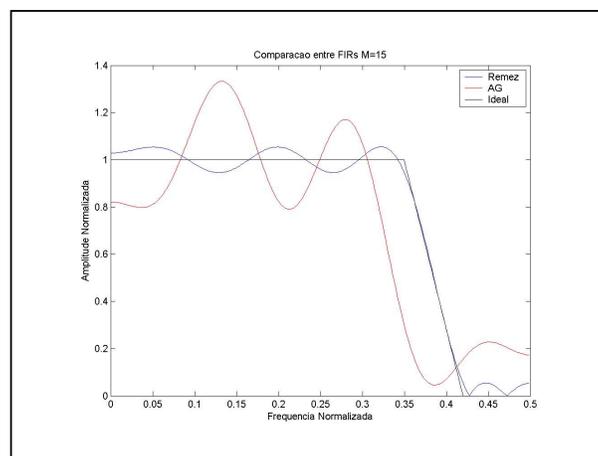


**Figura 42:** Fitness dos indivíduos da última população.

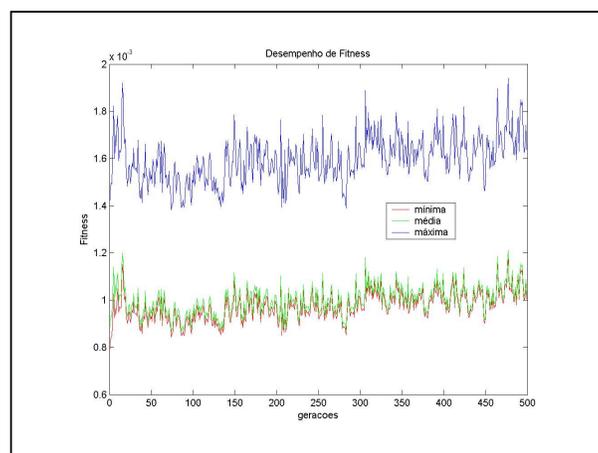
#### 4.3.4 Teste do escalonamento de *fitness*

Estes testes visam aplicar mecanismos de escalonamentos sobre a *fitness* bruta calculada pelo AG através da equação (57).

O primeiro teste foi aplicando à configuração básica um escalonamento da *fitness* do tipo linear. A relação escolhida foi *fitness* máxima / *fitness* média = 1,6. O filtro escolhido para os testes foi o *fpba* do Banco de Filtros (Tabela 4), *Tipo* = 1, *M* = 15, *maxgen* = 500, *popsize* = 160, com a configuração básica alterada para o escalonamento de *fitness* linear proposto. O melhor resultado foi ainda um filtro NOK apresentado na Figura 43 e com um desempenho de *fitness* apresentado na Figura 44.



**Figura 43:** Filtro NOK obtido via escalonamento de *fitness* linear.



**Figura 44:** Curvas de *fitness* com escalonamento linear.

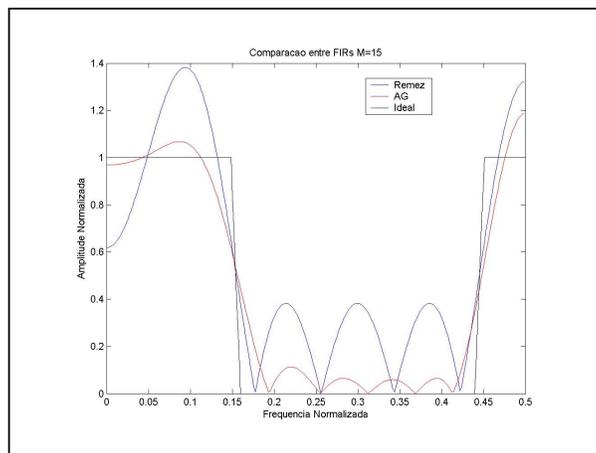
Como o teste não passou das quatro tentativas (foram feitas em média mais que quatro tentativas) no primeiro filtro, que é o *fpba*, a nota da configuração sob teste foi 0,0. Mesmo com este resultado, buscou-se dar continuidade ao teste, primeiramente com outros filtros do Banco de Filtros, além do *fpba* e em seguida com outras relações como 1,5 e 1,2. A nota permaneceu em 0,0.

Testes adicionais foram também feitos com escalonamento de *fitness* por janela e escalonamento através da constante  $\beta$  de Boltzmann com e sem controle automático de população (GOODMAN, 1996). Todas receberam 0,0 para o mesmo perfil de teste realizado com o escalonamento de *fitness* linear.

#### 4.3.5 Teste da ordem do filtro

O objetivo deste teste é o de se verificar se o AG pode fornecer resultados equivalentes aos apresentados pelo MPM através de um número de coeficientes menor.

A Figura 45 apresenta um caso destes: trata-se do filtro *fcf4* do Banco de Filtros (Tabela 4), em configuração básica, *Tipo = I*,  $M = 15$ ,  $maxgen = 1000$ ,  $popsiz = 160$ .

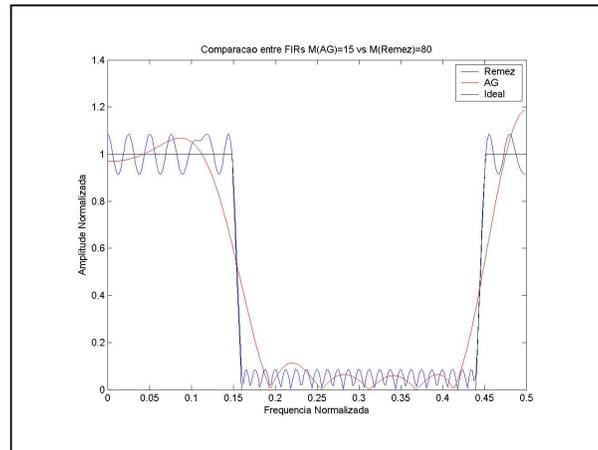


**Figura 45:** Filtro em OK confrontado com MPM equivalente.

Para o resultado obtido na Figura 45, foram medidos os *ripples* da resposta em frequência fornecida pelo AG. Tais *ripples*, juntamente com as especificações do filtro *fcf4* (faixas 1, 2 e 3, inicial, final e amplitude) constantes na Tabela 4, foram inseridas na equação (31). O valor recomendado para o número de coeficientes foi de 80, isto é, o MPM, para

atender a tais especificações, que o AG atendeu com 15 coeficientes, necessitou de 80 para fazer a mesma filtragem.

A Figura 46 apresenta o filtro MPM com  $M = 80$  e o filtro AG com  $M = 15$ , obtido no teste da Figura 45.



**Figura 46:** Filtro via MPM com mesmos *ripples* que o filtro AG.

#### 4.3.6 Teste com filtros de amplitudes arbitrárias

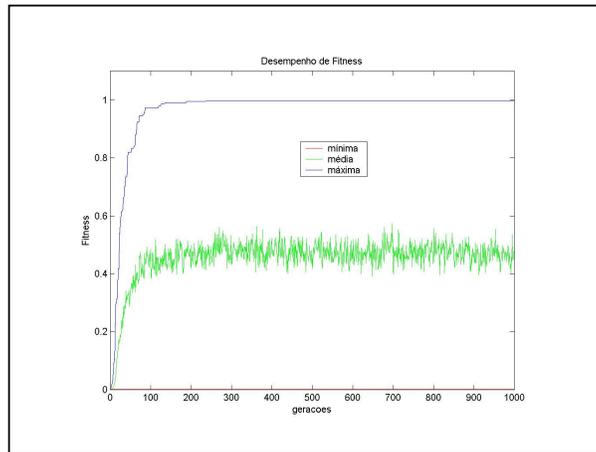
Foram realizados testes com filtros cujas especificações envolviam amplitudes arbitrárias, isto é, com valores de amplitudes diferentes de 0,0 e 1,0. O filtro corta-faixa resultante do teste da superposição apresentado na subseção 4.3.1, é um exemplo de filtro cuja resposta em frequência pôde ser obtida diretamente, porém com outros valores para *maxgen*, *popsize* e número máximo de tentativas. A resposta em frequência obtida pelo processo direto é exatamente a mesma que foi obtida na Figura 39.

#### 4.3.7 Teste com outras probabilidades de mutação

Foram realizados testes com a configuração básica recebendo diferentes valores para a probabilidade de mutação, além do valor que foi fixado para as fases 1 e 2, que foi de 10%.

Foi escolhido um deles como um dos principais representantes do efeito causado sobre os resultados que o AG forneceu:  $pmutation = 0,01$ , isto é de 1%.

O principal impacto foi sobre o desempenho das curvas de *fitness*. Foram repetidos os testes análogos aos obtidos no registro N<sup>o</sup> 2 da Tabela 6, porém com esta probabilidade de mutação alterada. Um dos resultados está apresentado na Figura 47.



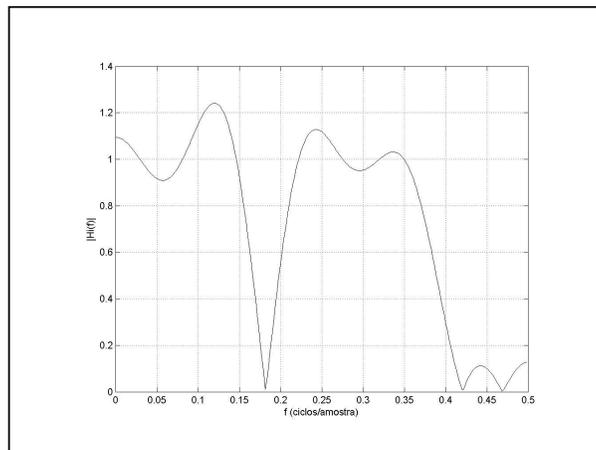
**Figura 47:** Desempenho de *fitness* com  $p_{mutação} = 0,01$ .

O valor máximo de 1,00 para a *fitness* máxima é mera coincidência nada tendo a ver com algum processo de normalização.

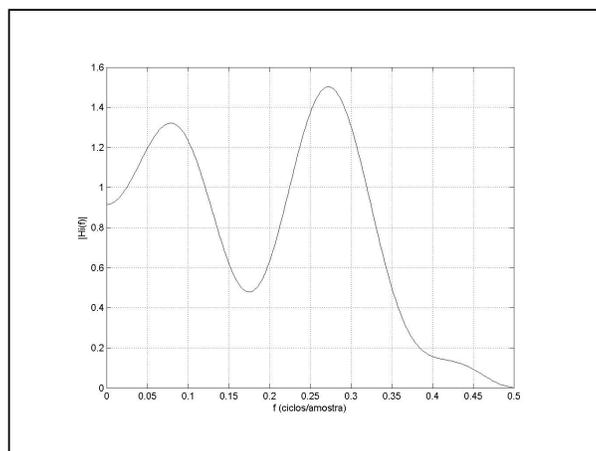
Os resultados do teste para esta probabilidade de mutação e outros próximos a estes valores, 1 a 5%, geraram uma nota final zero, por terem falhado em fornecerem resultados que atendessem às PSCRSO em quatro tentativas.

#### 4.3.8 Resultado relevante de provável máximo local

Vários resultados NOK foram obtidos com um perfil assemelhado ao da Figura 48. Porém, em tais testes, indivíduos com valores de *fitness* inferiores pertencentes a gerações intermediárias, apresentaram respostas assemelhadas às da Figura 49.



**Figura 48:** Uma solução do tipo NOK.



**Figura 49:** Melhor indivíduo de uma geração intermediária.



## CAPÍTULO 5

### DISCUSSÃO E CONCLUSÕES

#### 5.1 ANÁLISE DOS RESULTADOS

##### 5.1.1 Análise da primeira fase

O AG canônico, isto é uma configuração dos parâmetros dos conjuntos C, D e E no formato conhecido como canônico (roleta, mutação simples, cruzamento de um ponto, probabilidades de mutação e de cruzamento fixas, sem elitismo ou outras técnicas específicas), que corresponde ao registro N<sup>o</sup> 1 da Tabela 6, não conseguiu atender a um número superior a três filtros do Banco de Filtros. Configurações adicionais (com elitismo, com representação cromossômica em Binário Posicional e com cruzamento de dois pontos) com o AG canônico foram também testadas, também gerando poucos resultados válidos para os filtros propostos no Banco de Filtros. Estes resultados comprovaram a necessidade de um AG parametrizado sob configurações de valores e operadores mais tecnicamente aprimorados.

Configurações do AG com o método de seleção Torneio Estocástico, o *tselect*, apresentaram o seguinte sintoma: ou geravam um resultado OK ou um NOK. Poucos resultados SS foram gerados. Observando-se os valores finais de *fitness* que os resultados OK geravam (foram registrados na coluna de observações da Tabela 5), foi possível constatar-se que tais resultados OK eram a mesma solução para diferentes valores de sementes pseudo-aleatórias. Configurações de AG que continham este método falharam para alguns filtros cujo grau de dificuldade era maior (filtros com faixas de transição inferiores a 0,05 ciclos/amostra). Por este motivo, todas as configurações que empregaram este método receberam nota zero na Tabela 6.

Configurações do AG com o método de seleção da Amostragem Estocástica Universal, o *suselect*, apresentaram o seguinte comportamento: foram gerados resultados OK, NOK e SS, a depender do grau de dificuldade do filtro a ser encontrado. Observando-se os valores finais de *fitness* que os resultados OK geravam, foi possível constatar-se que eram sempre diferentes, isto é, encontravam sempre diferentes soluções sub-ótimas. Devido a esta característica, os registros de números 2, 5, 6 e 7 conseguiram receber notas diferentes de

zero. Os registros de números 3, 4 e 8 ficaram com zero devido ao tamanho da população ou o número de gerações não ser o que proporcionava pelo menos um resultado SS para algum dos filtros do Banco de Filtros. Porém, estes resultados NOK que desclassificavam a configuração não eram respostas que haviam evoluído para um máximo local (como na Figura 47). Eram respostas em que se aumentando o número máximo de gerações, a qualidade da resposta evoluía para SS ou OK. É o que aconteceu com a seqüência de registros de números 3, 4 e 5 da Tabela 6. Um exemplo de registro que recebeu esta evolução foi o Nº 3 da Tabela 7.

A combinação dos operadores que mais ofereceu resultados OK foi o cruzamento de dois pontos, o *twoptx*, com a mutação simples *bit a bit*, *bitmutat*. Todos os resultados que receberam notas diferentes de zero na Tabela 6 empregaram esta combinação.

O vetor  $h(n)$  tem seus maiores valores de coeficientes em torno da amostra central e seus menores valores nos extremos, havendo uma gradual diminuição dos mesmos de acordo com a equação (51), como se pode ver nas Figuras 5, 7 e 9. Portanto, pequenas variações no cromossomo precisam representar pequenas variações no valor do coeficiente decodificado. A representação cromossômica que melhor conseguiu acompanhar estas variações foi a que utilizou o Binário Gray, como se pode constatar pelos resultados que receberam notas diferentes de zero.

Da primeira fase restou somente uma configuração que conseguiu passar pelo Banco de Filtros, com todos os tipos de filtros (I ao IV), com uma nota diferente de zero, que foi a apresentada na subseção 4.2.

### 5.1.2 Análise da segunda fase

Na segunda fase, a configuração que foi aprovada na primeira foi tomada como base, porém os valores do tamanho da população, (*popsize*) e número máximo de gerações (*maxgen*) foram recebendo diferentes valores conforme a análise que se segue:

- a) o número de coeficientes determina, através das equações (54), (55) e (56), o espaço de busca do problema;
- b) com o aumento do número de coeficientes era necessário um aumento no valor de *popsize*. Se isto não fosse feito, os resultados do tipo NOK apresentavam soluções em máximos locais, como na Figura 47;
- c) com o aumento de *popsize*, surgia pelo menos um resultado, ainda em NOK, porém sem ter parado em um máximo local;

- d) nesta configuração, com este valor de *popsiz*e aumentado, aumentou-se o valor de *maxgen*, fazendo com que o resultado NOK do item c passasse para SS ou para OK, que é o caso das seqüência de figuras 33, 34 e 35.

E seguindo este princípio foram obtidos os registros 1 e 2, 3 e 5 da Tabela 8. O registro 6 desta tabela foi obtido por observação aos resultados que estavam sendo obtidos, permitindo ser estabelecida as seguinte relações:

$$popsiz = 160 \times 3.8^{L_2 - 8} \quad (59)$$

Onde:

- *popsiz* é o tamanho da população; e
- $L_2$  é obtido através da equação (54).

$$maxgen = 1000 + (L_2 - 8) \times 50, L_2 \geq 8 \quad (60)$$

Onde:

- *maxgen* é o número máximo de gerações; e
- $L_2$  é obtido através da equação (54).

Ainda na segunda fase um resultado, o que foi apresentado na Figura 36, demonstrou uma primeira aplicação alternativa onde o AG pode ser útil: em gerar filtros onde o MPM não o consiga fazer. Esta situação foi observada quando o número de coeficientes da especificação está acentuadamente abaixo do número de coeficientes recomendado pela equação (31).

### 5.1.3 Análise da terceira fase

Através do primeiro teste onde as figuras 37, 38 e 39 foram apresentadas, foi constatado que o AG atende ao princípio da superposição. A utilidade prática disto corresponde à possibilidade de serem obtidos filtros com especificações que tenham um grau de dificuldade tal não permita ao AG obtê-los de forma direta. Para estes casos, pode-se entrar com partes da especificação do filtro e depois somar seus resultados, como feito neste teste.

No teste da normalização da *fitness* bruta, ficou constatado que o erro quadrático do MPM não é o menor que se pode obter para todos os casos. O próprio AG obteve erros

menores. Isto fez com que a tentativa de normalização proposta pela equação (58) não fosse útil para manter a *fitness* excursionando de 0,0 a 1,0.

Os registros das evoluções de *fitness* para respostas OK e SS foi atípico. Com base unicamente na Figura 40, uma primeira análise poderia conduzir a se pensar que fosse somente um indivíduo que, através de mutações favoráveis, conseguisse aumentar sua *fitness*, devido ao elitismo estar sendo usado. Porém, através das figuras 41 e 42, constata-se que não é somente um indivíduo, mas aproximadamente 20 de 160 que efetivamente trocavam bagagem genética contribuindo para o aumento da *fitness* do melhor indivíduo, porém não chegando a aumentar efetivamente o valor da *fitness* média.

Os testes de escalonamentos da *fitness* bruta apresentaram resultados que conseguiram estabelecer uma melhor relação entre os valores de *fitness* máxima, média e mínima, como se pode observar na Figura 44. Porém o AG não evoluiu para os escalonamentos propostos, como se pode ver na Figura 43. E por este motivo foi mantida a *fitness* bruta que é obtida através da equação (57).

O teste da ordem do filtro demonstrou, através dos resultados das figuras 45 e 46, uma segunda aplicação alternativa onde o AG pode ser útil: em gerar filtros com um número de coeficientes menor do que o MPM gera, para mesmos valores de *ripples*. A única ressalva é a faixa de transição do filtro AG, apresentada na Figura 46, pode ser inaceitável para algumas aplicações. Isto ocorre quando o número de coeficientes especificado está abaixo do número recomendado pela equação (31) com o MPM ainda gerando resultados, porém com *ripples* inaceitáveis para qualquer tipo de aplicação, como o apresentado pela Figura 45.

O teste com filtros de amplitudes arbitrárias não evidenciou uma relação entre o conjunto A e os conjuntos C, D e E, que permitisse ser estabelecido algum padrão para *maxgen*, *popsiz*e e número máximo de tentativas. Os filtros de amplitudes arbitrárias precisaram ser tratados caso a caso. Porém, em conjunto com a capacidade do AG em operar com superposição, em menos de 10 tentativas em média, consegue-se encontrar o filtro procurado.

O teste com outras probabilidades de mutação revelou um aspecto interessante: a capacidade que o método da Amostragem Estocástica Universal tinha de oferecer diferentes respostas sub-ótimas para diferentes sementes pseudo-aleatórias, comentada na subseção 5.1.1, diminuiu com a diminuição da probabilidade de mutação. Isto fez com que os resultados fornecidos pelo AG apresentassem um perfil assemelhado ao obtido quando estava sendo usado o método do Torneio Estocástico, isto é, aumentaram os resultados OK, diminuíram os resultados SS e apareceram resultados NOK em maior quantidade. Não foi

possível ao AG resolver alguns filtros difíceis propostos pelo Banco de Filtros. Porém, ao contrário de outros métodos o método da roleta Estocástica, no máximo três filtros não conseguiam ser obtidos em quatro tentativas. Um fator positivo foi ter-se conseguido melhorar o aspecto da evolução das curvas de *fitness*, como se pode ver na Figura 47 quando comparada à Figura 40.

A função de *fitness* adotada, a equação (57), apresentou uma ineficácia. As figuras 48 e 49 demonstram o problema. O indivíduo apresentado Figura 48 tem um valor de *fitness* superior ao apresentado pelo indivíduo da Figura 49. Estes dois indivíduos pertencem a execuções distintas. O indivíduo da Figura 49 evoluiu para uma resposta SS, pois ainda correspondia a uma geração intermediária. E o indivíduo da Figura 48 é o melhor indivíduo de uma execução do AG. Portanto, numericamente, a somatória dos erros, ponto a ponto, que a função de *fitness* processa, não faz o AG ser informado de que um ponto em específico está deformando a resposta. Foram feitas algumas tentativas envolvendo estratégias de penalidades, porém sem resultados relevantes.

Foi ainda observado que, quanto maior for o número de coeficientes da especificação, mais o AG tende a encontrar respostas NOK com o perfil assemelhado ao da Figura 48. Foi buscada na Fundamentação Teórica, uma possível explicação para este fato e para os ajustes de tamanho de população: quanto maior for o número de coeficientes, maior será o número de oscilações na faixa passante e na faixa de rejeição (veja contraste entre a Figura 45 com  $M=15$  e a Figura 46 com  $M=80$ ). Isto pode ser observado na Tabela 2, independentemente de tal tabela estar relacionada ao MPM exclusivamente, uma vez que a resposta em frequência obtida pelo AG também é resultado de uma combinação linear de co-senos, que é a equação (8). Portanto, a cada nova oscilação encontrada na faixa passante do filtro, um novo ponto de máximo local surge onde o AG possa parar de evoluir. Isto fez com que o grau de dificuldade do AG aumentasse com o aumento do número de coeficientes, implicando na necessidade de ser aumentado o tamanho da população observado nas Tabelas 6 e 8.

O *software RespFreq* exerceu fundamental importância no descobrimento destes pontos apresentados nos dois parágrafos anteriores, por permitir que fosse visto em tela, geração a geração por meio do botão *Captura*, os melhores indivíduos que estavam sendo obtidos.

## 5.2 CONCLUSÕES

Através da metodologia proposta e dos resultados obtidos, foi possível a obtenção de dois AGs minimamente parametrizados e de algumas conclusões específicas que serão descritas.

### 5.2.1 Conclusões gerais

#### 5.2.1.1 AG automático

Para filtros cuja especificação determine amplitudes em 0,0 ou 1,0, foi obtido um AG automático, que fornece respostas sub-ótimas em comparação aos resultados oferecidos pelo Método de Parks-McClellan, em um número máximo de quatro tentativas. Seus parâmetros são:

##### Conjunto C:

- a) seleção = Amostragem Estocástica Universal;
- b) cruzamento = dois pontos;
- c) mutação = simples;
- d) representação cromossômica = Binário Gray; e
- e) função de *fitness* = equação (57).

##### Conjunto D:

- a) elitismo = sim;
- b) número máximo de gerações (*maxgen*) = equação (60);
- c) tamanho da população (*popsize*) = equação (59);
- d) probabilidade de cruzamento (*pcross*) = 0,85;
- e) probabilidade de mutação (*pmutation*) = 0,10; e
- f) semente = aleatória entre 0,001 e 0,999.

##### Conjunto E:

- a) *par\_sel* = nenhum parâmetro; e
- b) *par\_escal* = sem parâmetro ou técnica de escalamento de *fitness*.

### 5.2.1.2 AG não automático

Para filtros cuja especificação determine amplitudes arbitrárias, foi obtido um AG que fornece respostas sub-ótimas em comparação aos resultados oferecidos pelo Método de Parks-McClellan, em um número médio de tentativas inferior a 10, com parâmetros análogos ao AG automático apresentado na subseção 5.2.1.2, porém com os parâmetros *maxgen* e *popsiz* em aberto, a serem definidos caso a caso.

### 5.2.2 Conclusões específicas

O método proposto não emprega outras técnicas de outros métodos, como os que utilizaram a amostragem por frequência. Em tais métodos, tanto com AGs, PSO ou AIS, as estratégias de IC encontram dois ou no máximo quatro dos  $M$  coeficientes que devem ser encontrados do  $h(n)$ . Encontrados tais coeficientes, o método especialista ainda tem de ser empregado para encontrar os  $M-2$  ou  $M-4$  coeficientes restantes. Este AG entrega o  $h(n)$  pronto para ser utilizado, não requerendo outros recursos.

Não há problemas no método diante da quantização dos coeficientes, pois todo o confronto foi realizado com três casas decimais do AG contra as quinze do MPM. Basta mudar a codificação para o método gerar um número diferente de casas decimais.

O método é orientado à faixa, isto é, pode ser estabelecido um compromisso entre faixa passante e faixa de rejeição desejadas e obtidas, ao contrário do método das janelas ou do método de amostragem por frequência.

A estratégia de se ter deixado variar as probabilidades de mutação em um teste da terceira fase foi equivocada. Resultados com relevância possivelmente maior poderiam ser obtidos se mais testes variando-se a probabilidade de mutação e a de cruzamento fossem feitos na primeira fase.

O critério rígido das notas, desclassificando a configuração sob teste se em quatro tentativas não atendessem a um dos filtros da Tabela 4, também exerceu um impacto negativo sobre os resultados, pois acabou por desclassificar resultados interessantes como os proporcionados pelo Torneio Estocástico e pelo próprio método da Amostragem Estocástica Universal.

### 5.3 LINHAS DE AÇÃO FUTURAS

O método está, como especificado nos conjuntos A, C, D, E e F, pronto para ser utilizado para gerar resultados para amplitudes arbitrárias. Basta ser determinado um novo banco de filtros que representem uma amostra significativa do novo universo de possibilidades do conjunto A. Três refinamentos podem ser aplicados: a) trocar o número máximo de tentativas por um número médio de tentativas; b) colocar na primeira fase testes envolvendo variações das probabilidades de mutação e de cruzamento; e c) só operar com a representação cromossômica em Binário Gray.

Uma linha de ação futura relevante é a seguinte: codificar também dentro do cromossomo os parâmetros do conjunto D (que podem ser alterados em tempo de execução). Desta forma, o próprio AG pode encarregar-se de encontrar valores ótimos para parâmetros como as probabilidades de mutação, que exerceram neste experimento um impacto significativo.

A adaptação deste trabalho para encontrar coeficientes expressos em potências de dois é praticamente imediata devido à modelagem usada. Uma grande vantagem existe em se trabalhar com potências de dois: o que vai para o cromossomo é um pequeno conjunto de símbolos, que são os expoentes de dois. Portanto, o número de símbolos que serão representados dependerá da precisão desejada. Se a precisão fosse a utilizada neste trabalho,  $10^{-3}$ , os expoentes a serem representados iriam do  $2^{-1}$  ao  $2^{-9}$ , pois o  $2^{-10}$  não é mais representável em três casas decimais (0,0009766). Isto fornece nove expoentes que se tornam nove símbolos. Logo, 4 bits os representam com sobra. Reduzir o tamanho do cromossomo de  $L_2 \times 11$  bits para  $L \times 4$  bits proporciona um aumento significativo no desempenho do AG, com duas vantagens: a) operar com coeficientes de alto desempenho; e b) poder gerar filtros com ordens cada vez maiores em tempos de processamento tratáveis.

Uma vez obtida a versão com amplitudes arbitrárias, será um bom momento para ser iniciada uma modelagem através de Engenharia de Software (PRESSMAN, 2002) de uma versão de um produto propriamente dito, assemelhado ao FT3D (MARCON, 2002). Esta versão deverá oferecer uma interface no idioma Inglês e ser um software único, no intuito de ser oferecida à comunidade uma versão mais amigável, sem precisar entrar em um programa, sair, chamar outro e monitorar com outro. Haverá ainda a compatibilidade com o GALOPPS, a fim de serem feitos testes de depuração.

## REFERÊNCIAS BIBLIOGRÁFICAS

- AHMED, S.M. Design of FIR filters with arbitrary amplitude and phase specifications using genetic algorithm. In: IEEE, **Proceedings of the 46<sup>th</sup> IEEE International Midwest Symposium on Circuits and Systems MWSCAS 2003**, v. 2, p. 648-651, 2003.
- ANTONIOU, A. New improved method for the design of weighted-Chebyshev, nonrecursive, digital filters. **IEEE Transactions on Circuit and Systems**. v. CAS-30, p. 740-750, 1983.
- BARRETO, J.M. Redes Neurais Artificiais e Aplicações In: Cláudio César de Sá, Cléber Ângelo Capellari (Org.), **XII Escola Regional de Informática da SBC, ERI 2004**. Joinville, v. 1, p. 1-61, 2004.
- BEASLEY, D., BULL, D.R., MARTIN, R.R. Overview of genetic algorithms: part 2, research topics. **University Computing**, v. 15, n. 4, p. 170-181, 1993.
- BAKER, J.E. Reducing bias and inefficiency in the selection algorithm In: Grefenstette, J.J. (Ed.), **Proceedings of the Second International Conference on Genetic Algorithms**, Cambridge (MA). Lawrence Erlbaum Associates, Mahwah: p. 14-21, 1987.
- BÄCK, T., HAMMEL, U., SCHWEFEL, H.P. Evolutionary computation: comments on the history and current state. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 3-17, 1997.
- BENEVENUTO, N., MARCHESI, M., UNCINI, A. Applications of simulated annealing for the design of special digital filters. **IEEE Transactions on Signal Processing**. v. 40, n. 2, p. 323-332, 1992.
- BERRUT, J.P., TREFETHEN, L.N. Barycentric Lagrange Interpolation. **SIAM Review**, v. 46, n. 3, p. 501-517, 2004.
- BORLAND. **Borland C++ Builder Professional, versão 5.0**. Cupertino, USA, 2000. 1 CD-ROM. Borland Software Corporation. Ambiente integrado de desenvolvimento de programas de computador.

- CEMES, R., AIT-BOUDAUD, D. Multiplier-less FIR filter design with power-of-two coefficients. In: IEE, **Colloquium on Digital and Analogue Filters and Filtering Systems (Digest No. 1993/199)**. London: IEE, p. 6/1-6/4, 2 Nov. 1993a.
- CEMES, R., AIT-BOUDAUD, D. Genetic Approach to design of multiplierless FIR filters. **Electronic Letters**. Stevenage: IEE, v. 29, n. 24, p. 2090-2091, 25 Nov. 1993b.
- CHIPPERFIELD, A., FLEMING, P., POHLHEIM, H., FONSECA, C. **Genetic Algorithm Toolbox for use with MATLAB Version 1.2 User's Guide**. Department of Automatic Control and Systems Engineering. University of Sheffield, UK, 1994. Disponível em: [http://www.pohlheim.com/papers\\_pohlheim.html](http://www.pohlheim.com/papers_pohlheim.html). Acessado em 28/01/2006.
- CHEN, X.P., YU, S.L. FIR filter design: frequency-sampling method based on evolutionary programming. In: IEEE Neural Networks Council, **Proceedings of the Congress on Evolutionary Computation, CEC00**, v. 1, p. 575-579, 2000.
- DASGUPTA, D. **Artificial Immune Systems and Their Applications**. Berling: Springer-Verlag, Inc., 1999.
- DEB, K. **Practical Optimization Using Evolutionary Methods**. Kanpur: Kanpur Genetic Algorithms Laboratory (KanGAL), Department of Mechanical Engineering, Indian Institute of Technology Kanpur, 2005. Relatório Técnico 2005008. 20 p. Disponível em: <http://www.iitk.ac.in/kangal/reports.shtml>. Acessado em 02/02/2006.
- DIANATI, M., SONG, I. TREIBER, M. **An Introduction to Genetic Algorithms and Evolution Strategies**. Waterloo: Department of Electrical and Computer Engineering, University of Waterloo, 2002. Relatório Técnico. sn. 11 p. Disponível em: <http://bcr.uwaterloo.ca/~mdianati/Publications.htm>. Acessado em 02/02/2006.
- EVANGELISTA, G. Design of optimum high-order finite-wordlength digital FIR filters with linear phase. **Signal Processing**. v. 82, n. 2, p. 187-194, 2002.
- ETTER, D.M., HICKS, M.J., CHO, K.H. Recursive adaptive filter design using an adaptive genetic algorithm. In: IEEE **Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing ICASSP 82**, v. 7, p. 635-638, 1982.

GENTILI, P., PIAZZA, F., UNCINI, A. Evolutionary design of FIR digital filters with power-of-two coefficients. In: IEEE, **Proceedings of the 1<sup>st</sup> IEEE Conference on Evolutionary Computation, IEEE World Congress on Computational Intelligence**, v. 1, p. 110-114, 1994.

GIL, A.C. **Como elaborar projetos de pesquisa**. São Paulo: Editora Atlas S.A., 1996.

GOLDBERG, D. **Genetic Algorithms in Search, Optimization, and Machine Learning**. Reading: Addison-Wesley Publishing Company, Inc, 1989.

GOODMAN, E.D. **An introduction to GALOPPS, the “Genetic Algorithm Optimized for portability and parallelism” system release 3.2 (July 31, 1996)**. East Lansing: Department of Computer Science, and Case Center for Computer-Aided Engineering and Manufacturing, Michigan State University, USA, 1996. 90 p. Relatório Técnico 96-07-01. Disponível em: <http://garage.cps.msu.edu/software/galopps>. Acessado em 31/01/2006.

GOODMAN, E., PUNCH, B. **GALOPPS - the “Genetic Algorithm Optimized for portability and parallelism” versão 3.2.4, 25/Ago/2002**. East Lansing, 2002. 601 kbytes (compactados). Genetic Algorithms Research and Applications Group (GARAGe). *Software* de apoio ao desenvolvimento de algoritmos genéticos. Disponível em: <http://garage.cps.msu.edu/software/galopps>. Acessado em 31/01/2006.

HASAN, Y.M., KARAM, L.J., FALKINBURG, M., HELWIG, A., RONNING, M. Canonic signed digit Chebyshev FIR filter design. **IEEE Signal Processing Letters**, v. 8, n. 6, p. 167-169, 2001.

HERRMANN, O., RABINER, L.R., CHAN, D.S.K. Practical design rules for optimum finite impulse response lowpass digital filters. **Bell Systems Technical Journal**, v. 52, p. 769-799, 1973.

HOUNSELL, B.I., ARSLAN, T., THOMSON, R. Evolutionary design and adaptation of high performance digital filters within an embedded reconfigurable fault tolerant hardware platform. **Soft Computing – A Fusion of Foundations, Methodologies and Applications**, v. 8, n. 5, p. 307-317, 2004.

- HUANG, W.P., ZHOU, L.F., QIAN, J.X. FIR filter design: frequency sampling filters by particle swarm optimization algorithm In: IEEE: **Proceedings of the 3<sup>rd</sup> International Conference on Machine Learning and Cybernetics**, v. 4, p. 2322-2327, 2004.
- IZIDORO, S.C. Determinação do número de agrupamentos em conjuntos de dados multidimensionais utilizando algoritmos genéticos. **Infocomp Journal of Computer Science**, v. 4, n. 4, p. 63-72, 2005.
- JANOVETZ, J. **McClellan-Parks Program**. Illinois, 1998. 23.9 kbytes (compactados). Código na linguagem de programação C para gerar filtros FIR pelo método de Parks-McClellan. Disponível em: <http://www.janovetz.com/jake>. Acessado em 15/03/2006.
- KARABOĞA, N., ÇETINKAYA, B. Performance comparison of genetic algorithm based design methods of digital filters with optimal magnitude response and minimum phase. In: IEEE, **The 46<sup>th</sup> IEEE Midwest Symposium on Circuits and Systems, IEEE 2003**. Cairo: v.2, p. 644-647, 2003.
- KIRKPATRICK, S., GELATT, C.D., VECCHI, M.P. Optimization by simulated annealing. **Science**, v. 220, n. 4598, p. 671-680, 1983.
- KOZA, J.R. **Genetic Programming: on the programming of computers by means of natural selection**. Cambridge: MIT Press, 1992.
- KRASNOGOR, N. Memetic algorithms. In: **The Seventh International Conference on Parallel Problem Solving from Nature (PPSN VII)**, tutorial, 127 páginas, 2002.
- LANG, M. **Algorithms for the Constrained Design of Digital Filters with Arbitrary Magnitude and Phase Responses**. Viena, 1999. 223 f. Dissertação (Doutorado em Engenharia Elétrica) – Faculdade de Eletrotécnica. Universidade de Viena.
- LEE, A., AHMADI, M., JULIEN, G.A., LASHKARI, R.S., MILLER, W.C. Design of 1-D FIR filters with genetic algorithms. In: IEEE Signal Processing Research Centre (Org.), **Proceedings of the Fifth International Symposium on Signal Processing and its Applications ISSPA '99**. Brisbane: v. 2, p. 955-958, 1999.
- MAN, K.F., TANG, K.S., KWONG, S. **Genetic Algorithms**. 1 ed. Londres: Springer-Verlag, 1999.

- MARCON, F. **FT3D: Software Didático para Ensino Básico de Processamento de Sinais**. Curitiba, 2002. 146 f. Dissertação (Mestrado em Ciências) – Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial. Centro Federal de Educação Tecnológica do Paraná, CEFET-PR.
- MATHWORKS INC, THE. **MATLAB versão 6.5.0.180913a release 13**. Natick, USA, 2002. 1 CD-ROM. Ferramenta de apoio e desenvolvimento de tarefas numéricas computacionais.
- MCCLELLAN, J.H., PARKS, T.W. A unified approach to the design of optimum FIR linear-phase digital filters. **IEEE Transactions on Circuit Theory**, v. CT-20, n. 6, p. 697-701, 1973.
- MCCLELLAN, J.H., PARKS, T.W., RABINER, L.R. A computer program for designing optimum FIR linear phase digital filters. **IEEE Transactions on Audio and Electroacustics**, v. AU-21, n. 6, p. 506-526, 1973.
- MCCLELLAN, J.H., PARKS. A personal history of the Parks-McClellan algorithm. **IEEE Signal Processing Magazine**, v. 22, n. 2, p. 82-86, 2005.
- MEIER, P.L., **Probabilidade: Aplicações à Estatística**. 2 ed. Rio de Janeiro: Livros Técnicos e Científicos Editora S.A., 1987.
- MICROSOFT. **Microsoft Windows XP, Home Edition, versão 5.1.2600**. Redmond, USA, 2002. 1 CD-ROM. Microsoft Corporation. Sistema operacional para computador pessoal.
- MITCHEL, M. **GNU Compiler Collection, versão 4.0.2, 11/Out/2005**. 30.3 Mbytes (compactados). The Free Software Foundation, Inc. Compilador para programas em C ou C++. Disponível em: <http://directory.fsf.org/GNU/gcc.html>. Acessado em 31/01/2006.
- MOODY, G.B., MARK, R.G., GOLDBERGER, A.L. Physionet: a web-based resource for the study of physiologic signals. **IEEE Engineering in Medicine and Biology**, v. 20, n. 3, p. 70-75, 2001.
- MUÑOZ, D.G. **Discovering Unknown Equations that Describe Large Data Sets using Genetic Programming Techniques**. Linköping, 2005. 126 f. Dissertação (Mestrado em Engenharia Eletrônica) – Linköpings Universitet.

- NAMBIAR, R., MARS, P. Genetic and annealing approaches to adaptive digital filtering In: Avtar Singh (Ed), **Conference Record of the 26<sup>th</sup> Asilomar Conference on Signals, Systems and Computers**, v. 2, p. 871-875, 1992.
- NG, S.C., CHUNG, C.Y., LEUNG, S.H., LUK, A. A variable step size algorithm using evolution strategies for adaptive filterign. In: IEEE, **Proceedings of the 1999 Congress on Evolutionary Computation CEC 99**, v. 1, p. 542-545, 1999.
- NIELSEN, J.J. **MIXFFT, versão 0.5**. Hoersholm, 2000. 40.7 kbytes (compactados). Código na linguagem de programação C para calcular a Transformada Rápida de Fourier. Disponível em: <http://hjem.get2net.dk/jjn/fft.htm>. Acessado em 15/03/2006.
- ÖNER, M. A genetic algorithm for optimisation of linear phase FIR filter coefficients In: IEEE, **Conference Record of the 32<sup>th</sup> Asilomar Conference on Signals, Systems and Computers**. Pacific Grove: v. 2, p. 1397-1400, 1998.
- PARKS, T.W., MCCLELLAN, J.H. Chebyshev approximation for nonrecursive digital filters with linear phase. **IEEE Transactions on Circuit Theory**, v. CT-19, n. 2, p. 189-194, 1972a.
- PARKS, T.W., MCCLELLAN, J.H. A Program for the design of linear phase finite impulse response digital filters. **IEEE Transactions on Audio and Electroacustics**, v. AU-20, n. 3, p. 195-199, 1972b.
- PRESS, W.H., TEUKOLSKI, S.A., VETTERLING, W.T., FLANNERY, B.P. **Numerical Recipes in C: the Art of Scientific Computing**. 2 ed. Cambridge: Cambridge University Press, 2002.
- PRESSMAN, R.S. **Engenharia de Software**. 5 ed. Rio de Janeiro: McGraw-Hill, 2002.
- PROAKIS, J., MANOLAKIS, D. **Digital Signal Processing: Principles, Algorithms, and Applications**. 3 ed. Upper Saddle River: Prentice-Hall, 1996.
- RABINER, L.R., GOLD, B., MCGONEGAL, C.A. An approach to the approximation problem for nonrecursive digital filters. **IEEE Transactions on Audio and Electroacustics**, v. AU-18, n. 2, p. 83-106, 1970.

- RABINER, L.R., MCCLELLAN, J.H., PARKS, T.W. FIR digital filter design techniques using weighted Chebyshev approximation. **Proceedings of the IEEE**, v. 63, n. 4, p. 595-610, 1975.
- REMEZ, E.Y. General computational methods of Tchebycheff approximation. Kiev: **Atomic Energy Comission**. Versão traduzida para o Inglês N<sup>o</sup> 4491. p.1-85, 1957.
- SHAPIRO, J.L., PRÜGEL-BENNET, A. Maximum entropy analysis of genetic algorithm operators. **Lecture Notes in Computer Science**, v. 993, p. 14-24, 1995.
- SHI, Y., Particle Swarm optimization. **IEEE Connections**, v. 2, n. 1, p. 8-13, 2004.
- SILVA, A.P.A. Tutorial: genetic algorithms. **Learning and Nonlinear Models**, v. 1, n. 1, p. 45-60, 2002.
- SMITH, S.W. **The Scientist and Engineer's Guide to Digital Signal Processing**. 2 ed. San Diego: California Technical Publishing, 1999.
- STORN, R., PRICE, K. **Differential Evolution – a Simple and Efficient Adaptive Scheme for Global Optimization over continuous spaces**. Berkeley: International Computer Science Institute, 1995. 12 p. Relatório técnico TR-95-012.
- STORN, R. **Filter Design Wizard FIWIZ User Manual version 2.3, May 18, 2003**. Berkeley, 2003. Ferramenta de implementação de algoritmos por evolução diferencial. Disponível em: <http://www.icsi.berkeley.edu/~storn/fiwiz.html>. Acessado em 01/02/2006.
- STORN, R. Designing nonstandard filters with differential evolution. **IEEE Signal Processing Magazine**, v. 22, n. 1, p. 103-106, 2005.
- SUCLEY, D. Genetic algorithm in the design of FIR filters. **IEE Proceedings G: Circuits, Devices & Systems**, v. 138, n. 2, p. 234-238, 1991.
- TANOMARU, J. Motivação, fundamentos e aplicações de algoritmos genéticos. In: **II Congresso Brasileiro de Redes Neurais**. Curitiba, Brasil, 1995.
- WADE, G., VAN-EETVELT, P., DARWEN, H. Synthesis of efficient low-order FIR filters from primitive sections. **IEE Proceedings G: Circuits, Devices & Systems**, v. 137, n. 5, p. 367-372, 1990.

XIAOPING, C., BO, Q., GANG, L. An application of immune algorithm FIR filter design. In: IEEE, **International Conference on Neural Networks & Signal Processing**, Nanjing, v.1, p. 473-475-2, 2003.

YE, Z., CHANG, C.H. Local search method for FIR filter coefficients synthesis. In: Adam Osseiran, Edith Cowan U (Eds), **Proceedings of the Second IEEE Workshop on Electronic Design, Test and Applications DELTA'04**, Perth: IEEE Computer Society, p. 255-260, 2004.

ZADEH, L., Fuzzy Sets. **Journal of Information and Control**, v.8, n. 3, p. 338-353, 1965.

ZHANG, X., IWAKURA, H. Design of FIR linear phase filters with discrete coefficients using Hopfield neural networks. **Electronic Letters**, v. 30, n. 13, p. 1039-1040, 1994.

**RESUMO:**

Um método baseado em Algoritmos Genéticos foi desenvolvido e implementado em *software* para fornecer os coeficientes de um filtro FIR, cuja resposta em frequência é classificada como quase ótima em relação a um método especialista escolhido. Resultados comparativos evidenciam a contribuição do trabalho para os casos de frequências de transição abaixo de determinado valor. O *software* desenvolvido oferece a possibilidade de gerar filtros FIR com: três faixas de frequências com amplitudes arbitrárias, um número variável de coeficientes (pares e ímpares) e simetria positiva ou negativa. O método que originou o *software* busca a diminuição do número de parâmetros do Algoritmo Genético que requeiram uma configuração caso a caso. Portanto, a segunda contribuição do trabalho é a de oferecer uma forma de ser utilizada uma ferramenta baseada em Algoritmos Genéticos sem a necessidade de serem conhecidas as complexidades que envolvem sua utilização, por exemplo, operadores de recombinação, probabilidades de cruzamento e representação cromossômica. O resultado desta contribuição é uma ferramenta que proporciona ao usuário que só entenda de Processamento Digital de Sinais um meio de obtenção de filtros FIR com uma qualidade útil para casos onde o método especialista que foi confrontado não atenda. O método especialista a partir do qual foi feita a classificação de respostas ótimas, quase ótimas ou insatisfatórias, foi a versão implementada no *software* MATLAB do Segundo Algoritmo de Remez, o de múltiplas trocas, adaptado à síntese de filtros FIR por Rabiner, Parks e McClellan. O *software* foi implementado a partir da ferramenta *GALOPPS* para Algoritmos Genéticos.

**PALAVRAS-CHAVE**

FIR, Algoritmos Genéticos, Remez, MATLAB, Parks-McClellan.

**ÁREA/SUB-ÁREA DE CONHECIMENTO**

3.04.00.00 – 7 Engenharia Elétrica

Processamento de Sinais

Processamento Digital de Sinais

Ano 2006

N<sup>o</sup>: 400

# Livros Grátis

( <http://www.livrosgratis.com.br> )

Milhares de Livros para Download:

[Baixar livros de Administração](#)

[Baixar livros de Agronomia](#)

[Baixar livros de Arquitetura](#)

[Baixar livros de Artes](#)

[Baixar livros de Astronomia](#)

[Baixar livros de Biologia Geral](#)

[Baixar livros de Ciência da Computação](#)

[Baixar livros de Ciência da Informação](#)

[Baixar livros de Ciência Política](#)

[Baixar livros de Ciências da Saúde](#)

[Baixar livros de Comunicação](#)

[Baixar livros do Conselho Nacional de Educação - CNE](#)

[Baixar livros de Defesa civil](#)

[Baixar livros de Direito](#)

[Baixar livros de Direitos humanos](#)

[Baixar livros de Economia](#)

[Baixar livros de Economia Doméstica](#)

[Baixar livros de Educação](#)

[Baixar livros de Educação - Trânsito](#)

[Baixar livros de Educação Física](#)

[Baixar livros de Engenharia Aeroespacial](#)

[Baixar livros de Farmácia](#)

[Baixar livros de Filosofia](#)

[Baixar livros de Física](#)

[Baixar livros de Geociências](#)

[Baixar livros de Geografia](#)

[Baixar livros de História](#)

[Baixar livros de Línguas](#)

[Baixar livros de Literatura](#)  
[Baixar livros de Literatura de Cordel](#)  
[Baixar livros de Literatura Infantil](#)  
[Baixar livros de Matemática](#)  
[Baixar livros de Medicina](#)  
[Baixar livros de Medicina Veterinária](#)  
[Baixar livros de Meio Ambiente](#)  
[Baixar livros de Meteorologia](#)  
[Baixar Monografias e TCC](#)  
[Baixar livros Multidisciplinar](#)  
[Baixar livros de Música](#)  
[Baixar livros de Psicologia](#)  
[Baixar livros de Química](#)  
[Baixar livros de Saúde Coletiva](#)  
[Baixar livros de Serviço Social](#)  
[Baixar livros de Sociologia](#)  
[Baixar livros de Teologia](#)  
[Baixar livros de Trabalho](#)  
[Baixar livros de Turismo](#)