

Dissertação de Mestrado

Inatel

Instituto Nacional de Telecomunicações

**IMPLEMENTAÇÃO DE
EQUALIZADORES ALLPASS PARA
CORREÇÃO DE ATRASO DE
GRUPO EMPREGANDO TÉCNICAS
DE OTIMIZAÇÃO**

RODRIGO BARBOSA BRITO

OUTUBRO / 2005

Implementação de equalizadores *allpass* para correção de atraso de grupo empregando técnicas de otimização

RODRIGO BARBOSA BRITO

Dissertação apresentada ao Instituto Nacional de Telecomunicações, como parte dos requisitos para obtenção do título de Mestre em Telecomunicações.

Orientador: PROF. DR. SANDRO ADRIANO FASOLO

Santa Rita do Sapucaí
2005

Este trabalho foi realizado com o apoio financeiro da
Financiadora de Estudos e Projetos – FINEP



Dissertação defendida e aprovada em 07/10/2005, pela comissão julgadora:

(Prof. Dr. Sandro Adriano Fasolo/INATEL)

(Prof. Dr. Geraldo Gil Ramundo Gomes/INATEL)

(Prof. Dr. Carlos Aurélio Faria da Rocha/UFSC)

Prof. Dr. Adonias Costa da Silveira
Coordenador do Curso de Mestrado

“We seek meaning, even in random numbers”. (Carl Seagan)

Agradecimentos

Agradeço a Deus pela oportunidade e condições para se chegar a este objetivo. À minha mãe, Romeu, minhas irmãs, avós e tia que também fizeram parte desta conquista. Em especial à minha namorada Mariana, pelo carinho, afeto e compreensão em todos os momentos. Ao Prof. Sandro, pela orientação e amizade. Aos amigos Carlos, Marcelo, Antônio, Sebastião, Juliano e colegas de trabalho pela valiosa ajuda e companheirismo. A todos os professores e alunos do mestrado do INATEL, pela amizade e incentivo. Ao apoio financeiro proporcionado pelo convênio nº 22.02.0431.00, celebrado entre INATEL, Linear Equipamentos Eletrônicos S/A e a FINEP (Financiadora de Estudos e Projetos).

Índice

Lista de Figuras	vi
Lista de Tabelas	viii
Lista de Abreviaturas e Siglas	ix
Lista de Símbolos	x
1 Introdução	1
1.1 Introdução e Objetivos	1
1.2 Teoria de Filtros Digitais	3
1.2.1 Sistemas Lineares	3
1.2.2 Sistemas Invariantes no Tempo	4
1.2.3 Sistemas Lineares e Invariantes no Tempo - LIT	4
1.2.4 Causalidade	6
1.2.5 Estabilidade	6
1.3 Sistemas FIR e IIR	7
1.3.1 Análise no Domínio do Tempo	7
1.3.2 Análise no Domínio Z	9
1.3.3 Análise da estabilidade no Domínio Z	11
1.4 Filtros <i>allpass</i>	13
1.4.1 Estabilidade de filtros <i>allpass</i>	16
1.5 Estrutura da Dissertação	19

2	Técnicas e Métodos de Otimização	20
2.1	Introdução	20
2.2	Formulando o problema	21
2.3	<i>Minimax</i>	23
2.4	Método de Newton	25
2.5	Método Quasi-Newton	27
2.6	<i>Line Search</i>	31
2.7	Restrições	36
2.8	Conclusão	37
3	Algoritmo Final	39
3.1	Introdução	39
3.2	Relação entre os métodos e técnicas	39
3.3	Implementação do Algoritmo Final	40
3.4	Diagrama de Blocos	51
3.5	Conclusão	51
4	Resultados	54
4.1	Introdução	54
4.2	Resultados	54
4.2.1	Simulação A	55
4.2.2	Simulação B	61
4.2.3	Simulação C	64
4.3	Duração das Simulações	69
5	Itens Conclusivos	70
5.1	Conclusões	70
5.2	Principais Contribuições	72
5.3	Sugestão para Futuras Investigações	73
5.4	Publicações	73

Bibliografía

74

Lista de Figuras

1.1	Sistema Linear.	4
1.2	Sistema Invariante no Tempo.	4
1.3	Sistemas IIR e FIR.	10
1.4	Diagrama de pólos e zeros.	12
1.5	Região de Estabilidade.	18
2.1	Resposta desejada X Resposta obtida.	21
2.2	Exemplo de Função Objetiva com vários pontos estacionários. . .	23
2.3	Técnica <i>Line Search</i>	33
2.4	Variação do parâmetro ρ	34
3.1	Diagrama dos Algoritmos.	40
3.2	Diagrama de Blocos do Algoritmo Final.	52
4.1	Máscara de Atraso de Grupo.	55
4.2	Equalizador com uma seção.	56
4.3	Equalizador com duas seções.	57
4.4	Equalizador com quatro seções.	58
4.5	Equalizador com seis seções(Resposta de Atraso de Grupo).	58
4.6	Equalizador com seis seções (Plano Z).	59
4.7	Equalizador com seis seções (Resposta Impulsiva).	59
4.8	Coefficientes sobre a Região de Estabilidade.	60
4.9	Equalizador com duas seções.	62

4.10	Equalizador com sete seções.	62
4.11	Coeficientes sobre a Região de Estabilidade.	63
4.12	Máscara de atraso de grupo com descontinuidades.	64
4.13	Equalizador com uma seção.	65
4.14	Equalizador com duas seções.	65
4.15	Equalizador com dez seções.	66
4.16	Equalizador com vinte seções.	66
4.17	Equalizador com trinta e cinco seções.	67
4.18	Coeficientes sobre a Região de Estabilidade.	67

Lista de Tabelas

3.1	Pontos (c_{01}, c_{11}) pertencentes a região de estabilidade.	41
3.2	Pontos (c_{02}, c_{12}) para inicialização da segunda seção do equalizador.	48
4.1	Resultados Simulação A.	61
4.2	Resultados Simulação B.	63
4.3	Resultados Simulação C.	68
4.4	Duração das Simulações.	69

Lista de Abreviaturas e Siglas

TV	<i>Television</i> - Televisão
DSP	<i>Digital Signal Processing</i> - Processamento Digital de Sinais
ANATEL	Agência Nacional de Telecomunicações
LIT	<i>Linear Time-Invariant</i> - Sistema Linear Invariante no Tempo
FIR	<i>Finite Impulse Response</i> - Resposta Impulsiva Finita
IIR	<i>Infinite Impulse Response</i> - Resposta Impulsiva Infinita
DFP	<i>Davidon-Fletcher-Powell</i> - Referência à fórmula proposta por <i>Davidon-Fletcher-Powell</i>
BFGS	<i>Broyden-Fletcher-Goldfarb-Shanno</i> - Referência à fórmula proposta por <i>Broyden-Fletcher-Goldfarb-Shanno</i>

Lista de Símbolos

$x(n)$	Sinal de entrada de um sistema
$y(n)$	Resposta de um sistema quando excitado com $x(n)$
$T[\cdot]$	Operador matemático que representa a passagem de um sinal através de um sistema
$\delta(n)$	Delta de Dirac ou função impulso
$h(n)$	Resposta impulsiva de um sistema
a_k e b_k	Coefficientes dos sistemas <i>IIR</i> e <i>FIR</i>
c_k e d_k	Pólos e zeros de um sistema
$H_{ap}(z)$	Função de Sistema de um filtro <i>all-pass</i>
c_{0l} e c_{1l}	Coefficientes do filtro <i>all-pass</i>
\mathbf{c}	Vetor contendo os coefficients de todas as seções do filtro <i>all-pass</i>
ω	Frequência analógica
$\tau_{ap}(\mathbf{c}, \omega)$	Resposta de atraso de grupo do filtro <i>all-pass</i> em função dos coefficients e da frequência
$\theta_{ap}(\mathbf{c}, \omega)$	Resposta de fase do filtro <i>all-pass</i> em função dos coefficients e da frequência
$\arg \{ \cdot \}$	Argumento de um vetor
\mathbf{x}	Variável do sistema que representa o vetor \mathbf{c}
$\mathbf{e}_i(\mathbf{x})$	Erro do sistema em ω_i
τ_0	Resposta de atraso de grupo de referência
$\underset{\cup}{\mathbf{x}}$	Valor ótimo de \mathbf{x} que minimiza o erro do sistema
K	Número de amostras tomadas para se calcular o erro
L_p	Norma utilizada para se calcular a função objetivo do sistema a partir de $\mathbf{e}_i(\mathbf{x})$
$\Psi(\mathbf{x})$	Função objetiva do sistema dada pela norma L_p
$E(\mathbf{x})$	Vetor contendo todos as amostras da função erro do sistema
$\overset{\cap}{E}(\mathbf{x})$	Máximo valor do vetor $E(\mathbf{x})$
\mathbf{x}_0	Valor inicial para \mathbf{x}
δ	Vetor correção de \mathbf{x}

\mathbf{g}	Vetor gradiente da função objetivo $\Psi\mathbf{x}$
∇	Operador que representa a diferenciação de um vetor
\mathbf{H}	Matriz de <i>Hessian</i>
k	Índice das iterações
α_k	Passo de equalização na iteração k
\mathbf{S}	Matriz que se aproxima da matriz de <i>Hessian</i> inversa
A	Matriz <i>rank one</i> simétrica
γ	Parâmetro que representa a variação do vetor gradiente entre duas iterações sucessivas
$\text{sgn}\{\cdot\}$	Função sinal de um vetor
$\overset{\cup}{\alpha}$	Passo de equalização ótimo
α_0	Valor inicial do passo de adaptação
\mathbf{d}	Vetor correção de \mathbf{x} , sem a influência do passo de adaptação
ρ e σ	Parâmetros que controlam a exatidão da técnica <i>line search</i>
α_L	Valor mínimo de referência pra α
α_U	Valor máximo de referência pra α
$\overset{\cup}{\alpha}_0$	Valor do passo de adaptação obtido após processos de extrapolação e intrapolação
Ψ_L	Valor da função objetivo para α_L , quando analisada em função de α
Ψ_0	Valor da função objetivo para α_0
Ψ'_L	Inclinação da função objetivo em α_L
Ψ'_0	Inclinação da função objetivo em α_0
p e μ	Parâmetros utilizados pra calcular a função objetivo através da norma L_p , durante a técnica <i>minimax</i>
ε	Mínima variação permitida pra se finalizar o algoritmo <i>minimax</i>
ε_1 e ε_2	Parâmetros relacionados ao critério de parada do algoritmo Quasi-Newton
m	Contabiliza o número de operações envolvendo $\Psi(\mathbf{x})$
$\overset{\cup}{M}$	Número máximo de operações permitidas envolvendo $\Psi(\mathbf{x})$
τ e χ	Parâmetros utilizados para correção de possíveis discrepâncias no valor de $\overset{\cup}{\mathbf{x}}_0$
$\Delta\Psi(\mathbf{x})$	Variação no valor de $\Psi(\mathbf{x})$ entre iterações sucessivas
D	Parâmetro utilizado para garantir que \mathbf{S} seja positiva definida
\bar{c}_{01} e \bar{c}_{11}	Coefficientes do filtro <i>all-pass</i> obtidos após a minimização de $\Psi(\mathbf{x})$ para uma seção

ε_3	Pequena constante utilizada para obter os valores de inicialização da segunda seção
P_{jM}	Coefficientes utilizados pra inicialização da M -ésima seção do filtro, onde $j = 0, 1, \dots, 4$
Q	Parâmetro que analisa a ondulação em $e_i(\mathbf{x})$
$\overset{\cap}{e}(\mathbf{x})$	Máximo valor encontrado em $e_i(\mathbf{x})$
$\overset{\cup}{e}(\mathbf{x})$	Mínimo valor encontrado em $e_i(\mathbf{x})$
Q_{max}	Máximo valor de Q permitido para se obter equalizadores de qualidade

Resumo

Esta dissertação apresenta o desenvolvimento de um algoritmo, que utiliza métodos de otimização com o objetivo de obter filtros passa-tudo (*allpass*) estáveis para atuarem como equalizadores de atraso de grupo. O algoritmo obtém os coeficientes do filtro *allpass* a partir de uma resposta de atraso de grupo pré-definida. O projeto deste algoritmo envolve métodos de otimização com e sem restrições. Inicialmente, foi implementado o algoritmo *minimax* usando um algoritmo Quasi-Newton prático, proposto por Andreas Antoniou [Ant93], para minimização do erro. Após esta etapa, um outro algoritmo que aplica restrições foi implementado para garantir a estabilidade do filtro.

Extensivos testes realizados com o algoritmo final, utilizando diferentes dados de entrada e diferentes características de atraso de grupo, apresentam bons resultados de minimização do erro e filtros recursivos com poucos coeficientes. A dissertação detalha todas as etapas, técnicas e equações utilizadas na obtenção do algoritmo final. Também são mostrados alguns resultados e simulações que comprovam sua praticabilidade e eficiência. A motivação para este trabalho surgiu com a necessidade de se obter filtros *allpass* estáveis a partir de uma pré-determinada máscara atraso de grupo, necessária para pré-corriger distorções causadas por receptores de TV.

Palavras-chave: *DSP*, Filtros Digitais, Otimização com e sem restrições, algoritmo *minimax*.

Abstract

This dissertation presents the development of an algorithm using optimization methods to obtain stable allpass filters to act as group delay equalizer. The algorithm returns the coefficients of an allpass filter with a prescribed group delay response. The design of this algorithm involved constrained and unconstrained optimization methods. Initially, was implemented the minimax algorithm using a practical Quasi-Newton algorithm, proposed by Andreas Antoniou [Ant93], for error minimization. After that, a constrained algorithm was used to guarantee the filter's stability.

Extensive tests in the final algorithm with different input data, and different group delay characteristic show that it reaches good results of error minimization and return recursive filters with few coefficients, proving to be a successful design. The dissertation details all the steps, techniques and equations used to obtain the final algorithm and it still shows some results and simulations proving its feasibility and efficiency. The motivation for this work was the necessity to obtain a stable allpass filter with a prescribed group delay mask, necessary to pre-correct the delay distortion caused by TV receivers.

Keywords: DSP, Digital Filter, Constrained and Unconstrained Optimization, Minimax Algorithm.

Capítulo 1

Introdução

1.1 Introdução e Objetivos

A idéia deste trabalho surgiu com a necessidade de implementar um filtro digital com uma específica resposta de atraso de grupo, que atenda a curva definida pela Resolução 284 da ANATEL. Esta resolução é responsável pelo regulamento técnico para a prestação do serviço de radiodifusão de sons e imagens e do serviço de retransmissão de televisão. A implementação deste filtro, na transmissão de sinais de televisão, se torna necessária para pré-corrigir diferenças de atrasos entre sinais de crominância e luminância, dado que estes sinais são processados separadamente nos receptores de televisão. Devido a banda reduzida do sinal de crominância, esse sinal sofre um atraso maior que o de luminância, portanto o sinal responsável pela cor atinge o tubo de imagem atrasado em relação ao sinal de luminância. Com isso gera-se imagens onde o sinal de cor está deslocado do sinal de luminância. Mais detalhes a respeito de sinais e sistemas de televisão podem ser encontrados em [Ber84].

A partir de uma análise mais abrangente, pode-se dizer que filtros equalizadores de atraso de grupo são essenciais para corrigir degradações provocadas pela não linearidade na resposta de fase dos sistemas, dado que a resposta de atraso de grupo é a derivada da resposta de fase. Um sinal qualquer, ao passar por um sistema com resposta de fase não linear, tem suas componentes de frequência

atrasadas de forma desigual. Em sistemas com resposta de fase linear, ou seja, resposta de atraso de grupo constante, todas as componentes de frequência do sinal aplicado na entrada do sistema sofrem o mesmo atraso. Logo, tem-se na saída uma réplica atrasada do sinal aplicado na entrada. Mais detalhes a respeito das respostas de fase e atraso de grupo de filtros serão vistos nas próximas seções.

A partir destas necessidades, surgiu o interesse em estudar métodos de otimização, com e sem restrição, [Fle87], com o intuito de implementar um algoritmo para obtenção de equalizadores de atraso de grupo. Mais detalhadamente, este trabalho visa a implementação de um algoritmo baseado em técnicas e métodos de otimização, para obter os coeficientes de filtros passa-tudo (do inglês, *allpass*) estáveis. Ou seja, deseja-se filtros com respostas de magnitude plana e atraso de grupo dado por uma curva de referência.

Algoritmos para obtenção de filtros *allpass* têm sido um tema bastante pesquisado nas últimas décadas, onde o critério *Least Square*, algoritmo de *Remez*, método de Newton, dentre outros foram largamente estudados e empregados para solucionar este problema de otimização [Lan98], [Ko97]. Atualmente, algumas publicações visam uma melhora na complexidade computacional e na flexibilidade dos algoritmos, como pode ser comprovado com os resultados obtidos em [Had01] e [Mat04]. Como o objetivo deste trabalho é a determinação dos coeficientes de filtros *allpass*, para posterior implementação, que atendam uma determinada resposta de atraso de grupo, não há a preocupação em demasia com a complexidade do algoritmo, mas sim, com a precisão da resposta de atraso de grupo e a estabilidade do filtro obtido.

Procurou-se neste trabalho aplicar métodos que garantam a obtenção de equalizadores estáveis e com respostas de atraso de grupo bastante fiéis à máscara proposta. A partir deste cenário, a escolha dos métodos e técnicas utilizados foi baseada nos resultados apresentados em [Ko97] e na teoria detalhada em [Ant93]. Os métodos e técnicas de otimização utilizados na obtenção dos equalizadores são

Quasi-Newton, *Minimax*, uma técnica de busca de linha (mais conhecido como *line search*), além de restrições para garantir resultados estáveis. Após implementado o algoritmo, vários testes com diferentes curvas de atraso de grupo foram realizados. Os resultados obtidos mostram que o algoritmo pode ser utilizado para obtenção de filtros digitais com as mais diversas características de resposta de atraso de grupo.

Nas próximas seções serão apresentadas algumas definições básicas a respeito de filtros digitais, dando ênfase à filtros *allpass*, estabilidade e alguns conceitos relacionados a resposta de atraso de grupo. Esse pequeno resumo foi elaborado com o intuito de auxiliar a compreensão de todo o restante desta dissertação.

1.2 Teoria de Filtros Digitais

1.2.1 Sistemas Lineares

Considerando um sistema com sinais de entrada $x_1(n)$ e $x_2(n)$, tendo como respostas $y_1(n)$ e $y_2(n)$, respectivamente, esse sistema será linear, se e somente se:

$$\text{i) } T[x_1(n) + x_2(n)] = T[x_1(n)] + T[x_2(n)] = y_1(n) + y_2(n)$$

$$\text{ii) } T[\alpha x_1(n)] = \alpha T[x_1(n)] = \alpha y_1(n)$$

onde T representa a operação realizada pelo sistema sobre o sinal de entrada. Em sistemas não lineares, as duas condições acima não se verificam, ou seja, ao excitar a entrada de um sistema não linear com um sinal igual a $x_1(n) + x_2(n)$, o resultado será um sinal diferente de $y_1(n) + y_2(n)$.

A Figura 1.1 é um diagrama ilustrativo, onde se evidencia a característica de linearidade. O princípio da superposição, dado pela junção dos princípios da aditividade, i), e da homogeneidade, ii), se aplica a sistemas lineares.

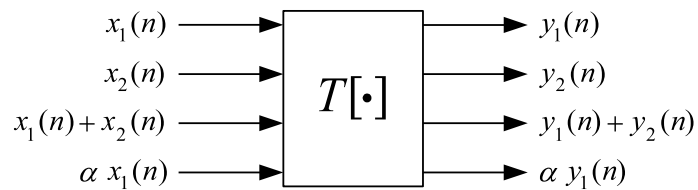


Figura 1.1: *Sistema Linear.*

1.2.2 Sistemas Invariantes no Tempo

Sistemas invariantes no tempo são aqueles que aplicando um deslocamento temporal ao sinal de entrada, causará o mesmo deslocamento de tempo ao sinal de saída. Supondo uma entrada $x(n)$ que corresponde a uma saída $y(n)$, caso um novo sinal, $x_1(n) = x(n - n_0)$, for aplicado na entrada do sistema e a resposta a esta excitação for $y_1(n) = y(n - n_0)$ diz-se que o sistema é invariante no tempo.

A Figura 1.2 traz um esquema demonstrando a característica de invariância no tempo de um sistema qualquer.

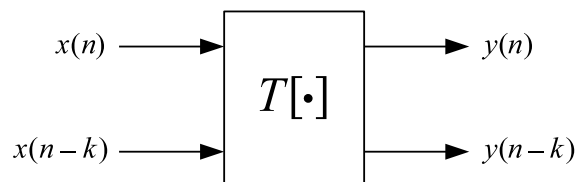


Figura 1.2: *Sistema Invariante no Tempo.*

1.2.3 Sistemas Lineares e Invariantes no Tempo - LIT

Após estas definições, pode-se unir essas duas propriedades e caracterizar os sistemas LIT. Tais sistemas são de grande importância e têm significantes aplicações na área de processamento digital de sinais. Antes de apresentar algumas considerações a respeito de sistemas LIT, será feita uma abordagem matemática sobre seqüências discretas, mais especificamente como representar um sinal discreto no

tempo através da somatória de impulsos deslocados no tempo, ou seja:

$$x(n) = \sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \quad (1.1)$$

onde $\delta(n)$ é a função amostra unitária. A Equação (1.1) diz que qualquer sinal discreto no tempo pode ser representado por uma somatória de impulsos deslocados e ponderados pela amplitude de $x(k)$. Ao excitar a entrada de um sistema linear com o sinal $x(n)$, a saída será $y(n) = T[x(n)]$, utilizando este conceito juntamente com (1.1), tem-se:

$$y(n) = T[x(n)] = T \left[\sum_{k=-\infty}^{\infty} x(k)\delta(n-k) \right] \quad (1.2)$$

Como esta abordagem se refere a sistemas lineares, pode-se aplicar o princípio da superposição, com isso pode-se escrever:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)T[\delta(n-k)] = \sum_{k=-\infty}^{\infty} x(k)h(n-k) \quad (1.3)$$

onde $h(n-k)$ é a resposta do sistema à entrada $\delta(n-k)$. A seqüência $h(n) = T[\delta(n)]$ é chamada de resposta impulsiva do sistema.

Todo este desenvolvimento, até chegar em (1.3), só é válido para sistemas lineares e invariantes no tempo. Tal abordagem permite afirmar que um sistema LIT é completamente caracterizado por sua resposta impulsiva, ou seja, dado a resposta impulsiva de um sistema LIT é possível usar (1.3) para calcular a saída, $y(n)$, a partir de qualquer entrada $x(n)$. A Equação (1.3) é conhecida como somatória de convolução e representada por:

$$y(n) = x(n) * h(n) \quad (1.4)$$

1.2.4 Causalidade

Uma outra característica importante de sistemas LIT é a causalidade, a partir deste conceito pode-se determinar se o sistema será realizável na prática. Um sistema causal é aquele onde a saída não depende de valores futuros da entrada, ou seja, a saída em tempo discreto para um dado instante n_0 , só depende das entradas com índices menores ou iguais à n_0 . Um sistema LIT é dito causal, se e somente se, sua resposta impulsiva for igual a zero para índices menores que zero, ou seja:

$$h(n) = 0, \text{ para } n < 0 \quad (1.5)$$

Com isso, pode-se representar a somatória de convolução da seguinte maneira:

$$y(n) = \sum_{k=-\infty}^{\infty} x(k)h(n-k) = \sum_{k=0}^{\infty} h(k)x(n-k) \quad (1.6)$$

1.2.5 Estabilidade

Os sistemas estáveis são imprescindíveis a implementações práticas. A razão básica para o estudo da estabilidade é evitar que se tenha na saída sinais com níveis que extrapolem um limite máximo permitido. Sistemas instáveis fazem com que uma determinada excitação produza na saída sinais com níveis tendendo ao infinito.

Um sistema LIT será estável, se e somente se, qualquer sinal de entrada com amplitude limitada produzir um sinal na saída também com amplitude limitada. Ou seja, para a entrada

$$|x(n)| < \infty \quad \text{para todo } n.$$

O sistema estável deve produzir

$$|y(n)| < \infty \quad \text{para todo } n.$$

A partir destes conceitos é possível determinar uma condição para análise da estabilidade de sistemas LIT através da resposta impulsiva, como pode ser visto em [Opp99]. Para um determinado sistema ser estável deve-se ter:

$$\sum_{k=-\infty}^{\infty} |h(k)| < \infty \quad (1.7)$$

Isto quer dizer que um sistema LIT será estável, se e somente se, sua resposta impulsiva for absolutamente somável. Posteriormente, será feita uma análise da estabilidade de sistemas empregando o conceito de Transformada Z.

1.3 Sistemas FIR e IIR

1.3.1 Análise no Domínio do Tempo

Nas seções anteriores caracterizou-se os sistemas quanto à linearidade, estabilidade, causalidade e invariância no tempo, após estas definições pode-se analisar dois diferentes tipos de sistemas LIT, trata-se dos sistemas com resposta impulsiva finita (*FIR*) e os com resposta impulsiva infinita (*IIR*). Os sistemas *FIR* são caracterizados por terem uma resposta impulsiva diferente de zero dentro de um intervalo de tempo finito. Como o foco deste trabalho está em sistemas aplicáveis na prática, a caracterização de um sistema *FIR* causal é:

$$h(n) = 0 \quad \text{para} \quad n < 0 \quad \text{e} \quad n \geq M$$

onde M é o comprimento da seqüência $h(n)$. Por esta razão, pode-se mostrar a somatória de convolução de um sistema *FIR* da seguinte maneira:

$$y(n) = \sum_{k=0}^{M-1} h(k)x(n-k)$$

Por este motivo, é comum encontrar em algumas referências a denominação de sistemas com memória finita, ou seja, precisa-se de memória finita com tamanho de M amostras para implementar um sistema *FIR*.

Em contrapartida, os sistemas *IIR* por terem uma resposta impulsiva infinita causal, sua saída é baseada na somatória de convolução dada por:

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

Até este ponto os sistemas LIT foram caracterizados através de sua resposta impulsiva, o que permite obter a saída de um sistema LIT para uma dada entrada, a partir do somatório de convolução. Para o caso de sistemas *FIR*, onde se tem uma resposta impulsiva finita, esta realização envolve manipulação de um número finito de amostras, conseqüentemente, um sistema *FIR* pode ser implementado diretamente através da somatória de convolução.

Por outro lado, a implementação de um sistema *IIR* através da somatória de convolução é claramente impossível, pois necessitaria de uma memória de tamanho infinito. Porém, um sistema *IIR* pode ser descrito e implementado na prática, através de uma equação de diferenças, comumente utilizada para representar este tipo de sistema [Opp99]. Um sistema LIT de tempo discreto pode ser descrito através de uma equação de diferenças da forma:

$$\sum_{k=0}^N a_k y(n-k) = \sum_{k=0}^M b_k x(n-k) \quad (1.8)$$

Para $a_0 = 1$, uma forma equivalente seria:

$$y(n) = \sum_{k=0}^M b_k x(n-k) - \sum_{k=1}^N a_k y(n-k) \quad (1.9)$$

onde a_k e b_k são conhecidos como coeficientes dos sistemas *IIR* e *FIR*.

1.3.2 Análise no Domínio Z

Nesta seção, sistemas LIT serão abordados utilizando definições e propriedades de Transformada Z [Opp99], cujo aprofundamento foi considerado dispensável.

Foi visto em seções anteriores que a saída de um sistema LIT pode ser obtida convoluindo uma entrada arbitrária com a resposta impulsiva do sistema. Esta análise no domínio do tempo e no domínio Z, se equivalem da seguinte maneira:

Domínio do Tempo	Domínio Z
$y(n) = x(n) * h(n)$	$Y(z) = X(z)H(z)$

Onde $z = |z|e^{j\Omega}$, sendo Ω a frequência discreta, e $H(z)$ é chamado de Função de Sistema, que equivale à Função de Transferência quando se trabalha no Domínio da Frequência (Analogico). Por definição a Função de Transferência se equivale a Função de Sistema quando $z = e^{j\Omega}$. Sinais instáveis podem ser representados na Função de Sistema, caso que não ocorre na Função de Transferência. A Função de Sistema é por definição a Transformada Z da resposta impulsiva, o que pode se comprovar com a relação anterior.

Aplicando a Transformada Z de ambos os lados de (1.8) e aplicando as propriedades de linearidade e deslocamento no tempo, [Opp99], chega-se na representação da equação de diferenças no Domínio Z.

$$\left(\sum_{k=0}^N a_k z^{-k} \right) Y(Z) = \left(\sum_{k=0}^M b_k z^{-k} \right) X(Z) \quad (1.10)$$

Para sistemas LIT onde as entradas e as saídas respeitam (1.8), a Função de Sistema é expressa por:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} \quad (1.11)$$

A Figura 1.3 é uma representação da Função de Sistema para sistemas *IIR* e *FIR*.

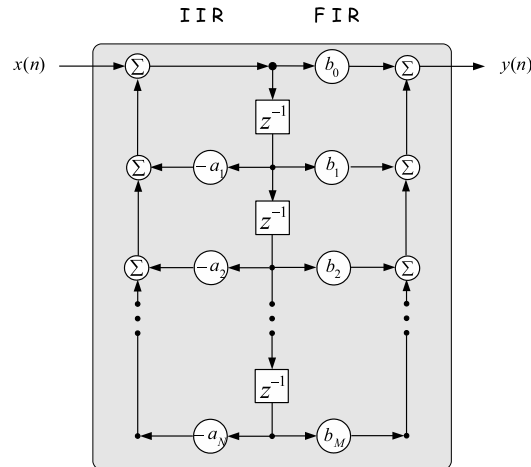


Figura 1.3: *Sistemas IIR e FIR.*

Na análise no domínio do tempo foi visto que sistemas *FIR* possuem resposta impulsiva finita e sistemas *IIR* resposta impulsiva infinita. No domínio Z , a análise é feita levando em conta a existência ou não do ramo de realimentação com a saída, ou seja, sistemas *IIR* possuem ramo de realimentação com a saída o que lhe dá a característica de sistema com memória infinita. Analisando a Figura 1.3 e (1.11) chega-se a conclusão de que sistemas *IIR* possuem tanto o numerador quanto o denominador na Função de Sistema, em outras palavras sistemas *IIR* possuem pólos e zeros. Já sistemas *FIR* possuem apenas o numerador na Função de Sistema, o que lhe garante possuir zeros e todos os pólos posicionados na origem ou no infinito. Esta propriedade garante que sistemas *FIR* sejam sempre estáveis assunto que será tratado com mais detalhes na seção seguinte.

$$\text{Sistemas } IIR \longrightarrow H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}}$$

$$\text{Sistemas } FIR \longrightarrow H(z) = \sum_{k=0}^M b_k z^{-k}$$

Com um pouco de manipulação objetivando salientar as raízes dos polinômios do denominador e do numerador, pólos e zeros respectivamente, chega-se na equação:

$$H(z) = \left(\frac{b_0}{a_0} \right) \frac{\prod_{k=1}^M (1 - c_k z^{-1})}{\prod_{k=1}^N (1 - d_k z^{-1})} \quad (1.12)$$

onde c_k representam as raízes do numerador (zeros do sistema) e d_k as raízes do denominador (pólos do sistema) da Função de Sistema. A análise desses pólos e zeros irá determinar as características do sistema no que se refere a estabilidade, resposta de fase e resposta de magnitude no domínio da frequência.

1.3.3 Análise da estabilidade no Domínio Z

Na Seção 1.2 foi analisada a estabilidade de sistemas LIT a partir da resposta impulsiva, outra forma bastante empregada é através da abordagem no domínio Z. Esse último método analisa a estabilidade de um sistema através do diagrama de pólos e zeros. Partindo de (1.12) e considerando que os pólos e zeros da Função de Sistema podem assumir valores complexos, sabe-se que o sistema será estável se a magnitude dos pólos for menor que a unidade [Ant93][Opp99]:

$$\begin{aligned} |d_k| < 1 & \quad \text{Sistema estável} \\ |d_k| \geq 1 & \quad \text{Sistema instável} \end{aligned}$$

onde $d_k = |d_k|e^{j\phi_k}$. Sistemas estáveis possuem todos os seus pólos posicionados dentro do círculo unitário, que é o lugar geométrico dos pontos onde $|z| = 1$. Enquanto sistemas instáveis possuem pelo menos um pólo com magnitude maior que um, ou seja, posicionado fora do círculo unitário. No diagrama de pólos e zeros, representa-se os pólos com “×” e os zeros com “o”.

Exemplo:

Dado um sistema com resposta:

$$H(z) = \frac{1 - z^{-1} + 0,8z^{-2}}{1 - 0,5z^{-1} + 0,9z^{-2}}$$

Reescrevendo a equação anterior de maneira a salientar os pólos e zeros do sistema, chega-se:

$$\begin{aligned} H(z) &= \frac{(1 - 0,89e^{j0,31\pi}z^{-1})(1 - 0,89e^{-j0,31\pi}z^{-1})}{(1 - 0,95e^{j0,42\pi}z^{-1})(1 - 0,95e^{-j0,42\pi}z^{-1})} \\ &= \frac{[1 - (0,50 + j0,74)z^{-1}][1 - (0,50 - j0,74)z^{-1}]}{[1 - (0,25 + j0,92)z^{-1}][1 - (0,25 - j0,92)z^{-1}]} \end{aligned}$$

Onde:

	Forma Retangular	Forma Exponencial
1° Zero	$0,50 + j0,74$	$0,89e^{j0,31\pi}$
2° Zero	$0,50 - j0,74$	$0,89e^{-j0,31\pi}$
1° Pólo	$0,25 + j0,92$	$0,95e^{j0,42\pi}$
2° Pólo	$0,25 - j0,92$	$0,95e^{-j0,42\pi}$

O diagrama de pólos e zeros deste exemplo pode ser visualizado pela Figura 1.4:

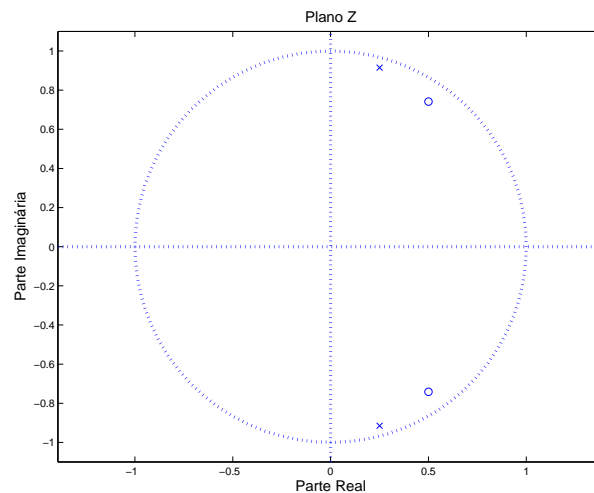


Figura 1.4: Diagrama de pólos e zeros.

Observa-se que este exemplo trata-se de um sistema estável, pois os pólos estão dentro do círculo unitário.

Foi visto anteriormente que sistemas *FIR* são sempre estáveis, este é um dos motivos para a sua larga utilização em projetos de filtros digitais. Em sistemas *IIR*, para garantir um sistema estável, há a necessidade de que todos os pólos estejam dentro do círculo unitário, tarefa esta que nem sempre é fácil de ser realizada. Apesar desta desvantagem, existem inúmeras aplicações onde há a necessidade de utilizar filtros *IIR*, esta dissertação é um exemplo onde houve a necessidade de implementar filtros *allpass* para correção de atraso de grupo, tal filtro só é conseguido, à principio, utilizando conceitos de filtros *IIR*.

1.4 Filtros *allpass*

Um filtro para atuar como equalizador de atraso de grupo é necessário que ele apresente algumas características essenciais. Esse filtro deve possuir resposta de magnitude plana (0 dB) e uma resposta de atraso de grupo específica, determinada pela aplicação e ajustada através de seus coeficientes. Sistemas com estas características são denominados filtros *allpass*. Estes filtros são representados por sistemas *IIR* com uma particularidade na Função de Sistema, os coeficientes do denominador e do numerador da Função de Sistema são os mesmos, porém com a ordem invertida. Por questões de implementação, tais como, obtenção de sistemas estáveis e rapidez no processo de obtenção de equalizadores que atendam a máscara proposta, o filtro *allpass* foi implementado como o produto de M seções de segunda ordem, dado por:

$$H_{ap}(z) = \prod_{l=1}^M \frac{1 + c_{1l}z + c_{0l}z^2}{c_{0l} + c_{1l}z + z^2} \quad (1.13)$$

Esta particularidade na Função de Sistema de filtros *allpass* faz com que seja possível reescrever (1.13), em função das raízes do denominador da seguinte

maneira.

$$\begin{aligned}
 H_{ap}(z) &= \prod_{l=1}^M \frac{(1 - r_l e^{j\phi_l} z)(1 - r_l e^{-j\phi_l} z)}{(z - r_l e^{j\phi_l})(z - r_l e^{-j\phi_l})} \\
 &= \prod_{l=1}^M z^{-1} \frac{(1 - r_l e^{j\phi_l} z)(1 - r_l e^{-j\phi_l} z)}{(1 - r_l e^{j\phi_l} z^{-1})(1 - r_l e^{-j\phi_l} z^{-1})}
 \end{aligned} \tag{1.14}$$

onde r_l representa o módulo e $\pm\phi_l$ a fase das raízes do denominador (pólos) de (1.13), determinados pelos coeficientes do filtro *allpass*. Para analisar as características da resposta de magnitude e fase do filtro, pode-se reescrever (1.14), fazendo $z = e^{j\omega T}$, onde ω representa frequência analógica e T o tempo de amostragem.

$$\begin{aligned}
 H_{ap}(e^{j\omega T}) &= \prod_{l=1}^M e^{-j\omega T} \frac{(1 - r_l e^{j\phi_l} e^{j\omega T})(1 - r_l e^{-j\phi_l} e^{j\omega T})}{(1 - r_l e^{-j\phi_l} e^{-j\omega T})(1 - r_l e^{j\phi_l} e^{-j\omega T})} \\
 &= \prod_{l=1}^M e^{-j\omega T} \frac{[1 - r_l e^{j(\omega T + \phi_l)}][1 - r_l e^{j(\omega T - \phi_l)}]}{[1 - r_l e^{-j(\omega T + \phi_l)}][1 - r_l e^{-j(\omega T - \phi_l)}]}
 \end{aligned} \tag{1.15}$$

Utilizando a Identidade de Euler, $e^{j\omega T} = \cos(\omega T) + j \sin(\omega T)$, chega-se:

$$\begin{aligned}
 H_{ap}(e^{j\omega T}) &= \prod_{l=1}^M e^{-j\omega T} \left\{ \frac{[1 - r_l \cos(\omega T + \phi_l)] - j [r_l \sin(\omega T + \phi_l)]}{[1 - r_l \cos(\omega T + \phi_l)] + j [r_l \sin(\omega T + \phi_l)]} \right\} \cdot \\
 &\quad \cdot \left\{ \frac{[1 - r_l \cos(\omega T - \phi_l)] - j [r_l \sin(\omega T - \phi_l)]}{[1 - r_l \cos(\omega T - \phi_l)] + j [r_l \sin(\omega T - \phi_l)]} \right\}
 \end{aligned} \tag{1.16}$$

Nota-se que o numerador e o denominador dos termos entre parênteses formam um par conjugado complexo, ou seja, possuem mesma magnitude e fase oposta.

Expressando (1.16) na forma exponencial, tem-se:

$$\begin{aligned}
 H_{ap}(e^{j\omega T}) &= \prod_{l=1}^M \exp\{-j\omega T\} \cdot \exp\left\{-j2 \arctan\left[\frac{r_l \sin(\omega T + \phi_l)}{1 - r_l \cos(\omega T + \phi_l)}\right]\right\} \cdot \\
 &\quad \cdot \exp\left\{-j2 \arctan\left[\frac{r_l \sin(\omega T - \phi_l)}{1 - r_l \cos(\omega T - \phi_l)}\right]\right\}
 \end{aligned} \tag{1.17}$$

Com isso, comprova-se que (1.17), e conseqüentemente (1.13), trata-se de um filtro com resposta de magnitude plana ($|H_{ap}(e^{j\omega T})| = 1$) e resposta de fase definida pelos coeficientes c_{0l} e c_{1l} .

O objetivo proposto neste trabalho é a obtenção de filtros *allpass* com uma determinada resposta de fase, que proporcione uma específica resposta de atraso de grupo. Por este motivo, a expressão que determina a resposta de atraso de grupo do filtro *allpass*, dado por (1.13), é de extrema importância para o desenvolvimento deste trabalho. Assumindo $z = e^{j\omega T}$ em (1.13), tem-se:

$$H_{ap}(e^{j\omega T}) = \prod_{l=1}^M \frac{1 + c_{1l}e^{j\omega T} + c_{0l}e^{j2\omega T}}{c_{0l} + c_{1l}e^{j\omega T} + e^{j2\omega T}} \quad (1.18)$$

Aplicando a identidade de Euler em (1.18), obtém-se:

$$\begin{aligned} H_{ap}(e^{j\omega T}) &= \prod_{l=1}^M \frac{1 + c_{1l} \cos(\omega T) + jc_{1l} \sin \omega T + c_{0l} \cos(2\omega T) + jc_{0l} \sin(2\omega T)}{c_{0l} + c_{1l} \cos(\omega T) + jc_{1l} \sin(\omega T) + \cos(2\omega T) + j \sin(2\omega T)} \\ &= \prod_{l=1}^M \frac{[1 + c_{1l} \cos(\omega T) + c_{0l} \cos(2\omega T)] + j [c_{1l} \sin(\omega T) + c_{0l} \sin(2\omega T)]}{[c_{0l} + c_{1l} \cos(\omega T) + \cos(2\omega T)] + j [c_{1l} \sin(\omega T) + \sin(2\omega T)]} \end{aligned} \quad (1.19)$$

Por definição, o atraso de grupo de um filtro é dado por:

$$\tau_{ap}(\mathbf{c}, \omega) = -\frac{d\theta_{ap}(\mathbf{c}, \omega)}{d\omega} \quad (1.20)$$

onde

$$\theta_{ap}(\mathbf{c}, \omega) = \arg \{ H_{ap}(e^{j\omega T}) \} \quad (1.21)$$

é a resposta de fase do filtro. Nota-se que (1.19) é o produtório de seções de segunda ordem, conseqüentemente $\theta_{ap}(\mathbf{c}, \omega)$ é a somatória do argumento destas

seções. A partir de (1.19) e (1.21) chega-se em:

$$\begin{aligned} \theta_{ap}(\mathbf{c}, \omega) = & \sum_{l=1}^M \arctan \left[\frac{c_{1l} \sin(\omega T) + c_{0l} \sin(2\omega T)}{1 + c_{1l} \cos(\omega T) + c_{0l} \cos(2\omega T)} \right] - \\ & - \arctan \left[\frac{c_{1l} \sin(\omega T) + \sin(2\omega T)}{c_{0l} + c_{1l} \cos(\omega T) + \cos(2\omega T)} \right] \end{aligned} \quad (1.22)$$

A partir de (1.22) e (1.20), chega-se na resposta de atraso de grupo do filtro *allpass*.

$$\begin{aligned} \tau_{ap}(\mathbf{c}, \omega) = & T \sum_{l=1}^M - \left[\frac{c_{1l}^2 + 2c_{0l}^2 + c_{1l} \cos(\omega T) + 2c_{0l} \cos(2\omega T)}{(1 - c_{0l})^2 + c_{1l}^2 + 2c_{1l}(1 + c_{1l}) \cos(\omega T) + 4c_{0l} \cos^2(\omega T)} \right] + \\ & + \left[\frac{2 + c_{1l}^2 + c_{1l}(3 + c_{0l}) \cos(\omega T) + 2c_{0l} \cos(2\omega T)}{(1 - c_{0l})^2 + c_{1l}^2 + 2c_{1l}(1 + c_{1l}) \cos(\omega T) + 4c_{0l} \cos^2(\omega T)} \right] \\ = & 2T \sum_{l=1}^M \frac{1 - c_{0l}^2 + c_{1l}(1 - c_{0l}) \cos(\omega T)}{(1 - c_{0l})^2 + c_{1l}^2 + 2c_{1l}(1 + c_{1l}) \cos(\omega T) + 4c_{0l} \cos^2(\omega T)} \end{aligned}$$

Com isso, chega-se na equação final para o atraso de grupo do filtro *allpass*.

$$\tau_{ap}(\mathbf{c}, \omega) = 2T \sum_{l=1}^M \frac{\tilde{P}_l(\omega)}{P_l(\omega)} \quad (1.23)$$

onde

$$\begin{aligned} \tilde{P}_l(\omega) &= 1 - c_{0l}^2 + c_{1l}(1 - c_{0l}) \cos(\omega T) \\ P_l(\omega) &= (1 - c_{0l})^2 + c_{1l}^2 + 2c_{1l}(1 + c_{0l}) \cos(\omega T) + 4c_{0l} \cos^2(\omega T) \end{aligned}$$

e

$$\mathbf{c} = [c_{01} \quad c_{11} \quad c_{02} \quad c_{12} \quad \dots \quad c_{0M} \quad c_{1M}]^T$$

1.4.1 Estabilidade de filtros *allpass*

Como foi dito em seções anteriores, para se obter filtros estáveis é necessário que todos os seus pólos estejam dentro do círculo unitário, ou seja, as raízes do denominador de (1.13) devem ser, em módulo, menores que a unidade. Para

garantir a estabilidade do filtro é necessário determinar qual faixa de valores que c_{0l} e c_{1l} pode assumir, ou melhor, determinar a região de estabilidade. Para isto foi utilizado o Critério de Estabilidade Jury-Marden [Jur64][Mar49]. Considerando que $D(z)$ seja o polinômio do denominador de (1.13), definido como:

$$D(z) = \sum_{k=0}^N a_k z^{N-k} \quad (1.24)$$

onde N é a ordem do polinômio. O critério de estabilidade Jury-Marden diz que para $D(z)$ possuir raízes dentro do círculo unitário, é necessário que as seguintes condições sejam satisfeitas:

- (i) $D(1) > 0$
- (ii) $(-1)^N D(-1) > 0$
- (iii) $a_0 > |a_N|$
 $|c_0| > |c_{N-1}|$
 $|d_0| > |d_{N-2}|$

onde

$$c_i = \begin{vmatrix} a_0 & a_{N-i} \\ a_N & a_i \end{vmatrix} \quad \text{para } i = 0, 1, \dots, N-1$$

$$d_i = \begin{vmatrix} c_0 & c_{N-1-i} \\ c_{N-1} & c_i \end{vmatrix} \quad \text{para } i = 0, 1, \dots, N-2$$

Aplicando o critério Jury-Marden para $D(z) = z^2 + c_{1l}z + c_{0l}$, de onde tira-se que $a_0 = 1$, $a_1 = c_{1l}$, $a_2 = c_{0l}$ e obviamente $N = 2$, chega-se nas condições que garantem a estabilidade do filtro *allpass* proposto em (1.13).

- (i) $D(1) = 1 + c_{1l} + c_{0l} > 0$, de onde tira-se $c_{1l} + c_{0l} > -1$

(ii) $(-1)^2 D(-1) = 1 - c_{1l} + c_{0l} > 0$, de onde tira-se $c_{1l} - c_{0l} < 1$

(iii) $1 > |c_{0l}|$, de onde tira-se $-1 < c_{0l} < 1$

Com a condição $|c_0| > |c_{N-1}|$ chega na mesma restrição obtida em (ii). Com isso conclui-se que o filtro será estável, se e somente se, os coeficientes da Função de Sistema obedecerem as seguintes restrições:

$$-1 < c_{0l} < 1, \quad c_{1l} - c_{0l} < 1, \quad c_{1l} + c_{0l} > -1 \quad (1.25)$$

para $l = 0, 1, 2, \dots, M$. A região de estabilidade é representada pela Figura 1.5, no plano $c_0 \times c_1$.

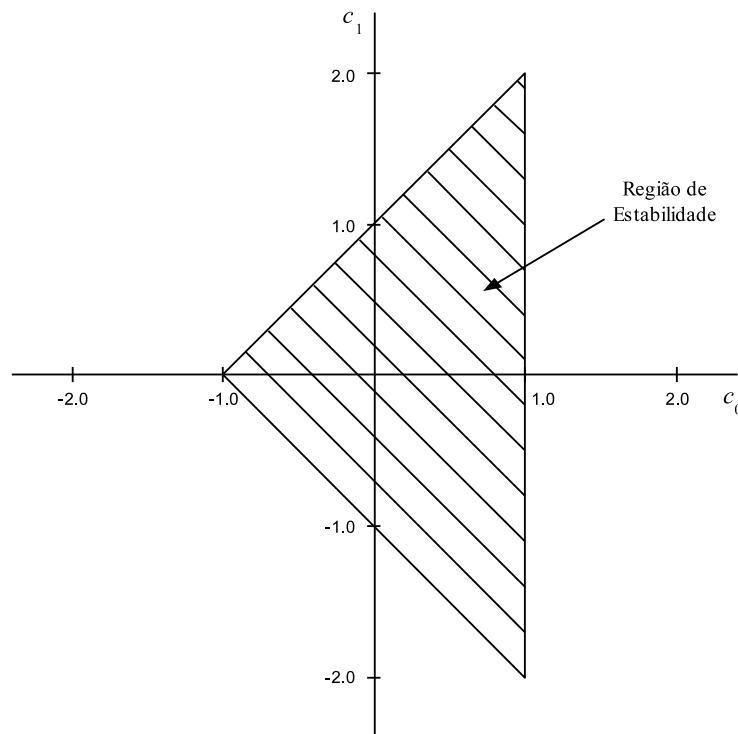


Figura 1.5: Região de Estabilidade.

Para que um filtro com Função de Sistema expressa por (1.13) seja estável, é necessário que os coeficientes de cada seção de segunda ordem estejam dentro da região de estabilidade definida pela Figura 1.5. Os filtros projetados são implementados através da combinação de seções de segunda ordem, por isso a

necessidade de detalhar seu comportamento.

1.5 Estrutura da Dissertação

O objetivo deste trabalho, após um estudo prévio de alguns métodos e técnicas de otimização, é implementar um algoritmo que a partir de uma resposta de atraso de grupo de referência, retorne os coeficientes de um filtro *allpass* estável. Neste primeiro Capítulo foi abordado alguns conceitos que fundamentam o desenvolvimento deste trabalho.

O Capítulo 2 apresenta todas as técnicas e métodos utilizados na obtenção do algoritmo, são eles *line search*, Quasi-Newton, *Minimax* e uma técnica que garante a estabilidade do resultado final. A finalidade, os princípios e a teoria necessária para o bom entendimento dos métodos são apresentados, bem como todo o equacionamento necessário para realizar sua implementação.

O Capítulo 3 traz de uma forma detalhada os algoritmos que implementam cada um dos métodos discutido no Capítulo 2. Também, é apresentado como os quatro métodos são interligados para formar o algoritmo final. Todos os passos e equações utilizados nos algoritmos são comentados para o perfeito entendimento e compreensão.

No Capítulo 4, os resultados obtidos com a implementação do algoritmo são apresentados e discutidos. As respostas de atraso de grupo dos filtros, para as diversas curvas de referência, são apresentadas.

O Capítulo 5 traz as conclusões finais deste trabalho, principais contribuições, bem como idéias para futuras investigações, além das publicações geradas.

Capítulo 2

Técnicas e Métodos de Otimização

2.1 Introdução

Este Capítulo abordará alguns métodos e técnicas de otimização que podem ser utilizados na implementação de filtros digitais e equalizadores. O equacionamento, aplicações, vantagens e desvantagens serão analisados, bem como, as características essenciais para o bom entendimento das técnicas e métodos aplicados nesta dissertação.

O método Quasi-Newton, uma variação do tradicional método de Newton, foi utilizado para minimização da função objetivo, função que modela o erro do sistema, juntamente com uma técnica *line search* para obtenção de um passo de adaptação otimizado. A técnica *minimax* foi utilizada para obter um erro uniformemente distribuído, evitando picos de erro na resposta final do sistema, além da aplicação de um algoritmo que implementa restrições necessárias para garantir a estabilidade dos filtros.

2.2 Formulando o problema

O objetivo final deste trabalho é implementar um algoritmo para obtenção dos coeficientes de filtros *allpass*, auxiliado por técnicas e métodos de otimização. O primeiro passo é determinar a função objetivo, formulada a partir do erro do sistema que é formado basicamente pela diferença entre o resposta de atraso de grupo atual do sistema e a resposta de referência que se deseja obter. Considerando que τ_0 seja a resposta de atraso de grupo que se deseja obter e $\tau_{ap}(\mathbf{c}, \omega)$, expresso por (1.23), o atraso de grupo do filtro *allpass*, a função erro será dada por:

$$e_i(\mathbf{x}) = \tau_{ap}(\mathbf{x}, \omega_i) - \tau_0 \quad (2.1)$$

onde $\mathbf{x} = \mathbf{c} = [c_{01} \ c_{11} \ c_{02} \ c_{12} \ \dots \ c_{0M} \ c_{1M}]^T$ e o índice i indica as amostras de frequência $\omega_1, \omega_2, \omega_3, \dots, \omega_K$ onde é calculado o erro do sistema, como mostrado na Figura 2.1.

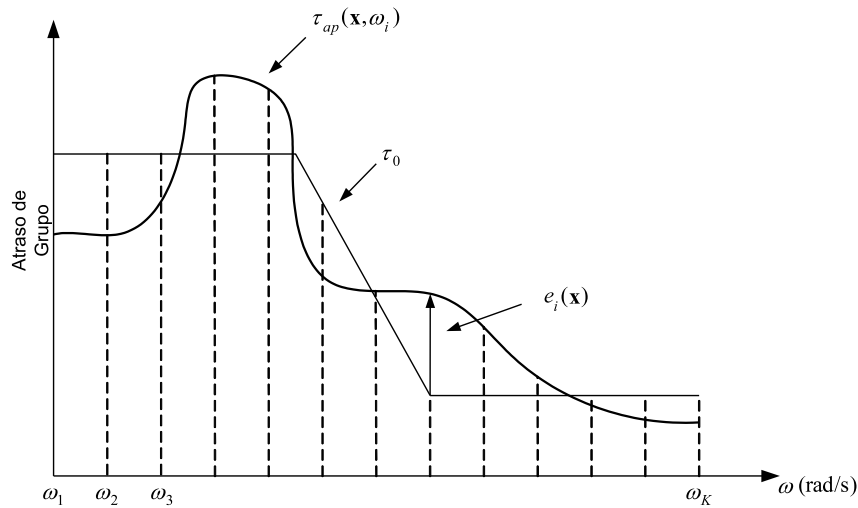


Figura 2.1: Resposta desejada X Resposta obtida.

Pode-se dizer que o objetivo dos métodos de otimização é a obtenção de um valor ótimo, $\hat{\mathbf{x}}$, que resulte em $e_i(\hat{\mathbf{x}}) \approx 0$ para $i = 1, 2, 3, \dots, K$. De posse do vetor

erro e aplicando a norma L_p , [Fle87], chega-se na função objetivo do sistema:

$$\Psi(\mathbf{x}) = L_p = \|E(\mathbf{x})\|_p = \left[\sum_{i=1}^K |e_i(\mathbf{x})|^p \right]^{1/p} \quad (2.2)$$

onde $E(\mathbf{x}) = [e_1(\mathbf{x}) \ e_2(\mathbf{x}) \ \dots \ e_K(\mathbf{x})]^T$. Essa função deve satisfazer alguns requisitos, tais como ser diferenciável, retornar apenas um escalar e que sua aplicação reflita na minimização de todos os elementos do vetor $E(\mathbf{x})$, essas características são conseguidas com a utilização da norma L_p .

Dado a função objetivo de um sistema, a principal função dos métodos de otimização é obter as coordenadas de \mathbf{x} onde se tem o mínimo valor para essa função, ou melhor, o menor erro possível para o sistema. Para isto, utiliza-se técnicas de convergência baseadas em informações obtidas das derivadas primeira e segunda da função. As informações extraídas da derivada primeira irão indicar a localização dos pontos estacionários, tais pontos têm que satisfazer a condição:

$$\frac{d\Psi(\mathbf{x})}{d\mathbf{x}} = 0 \quad (2.3)$$

e com a informação obtida da derivada segunda é possível determinar se esse ponto estacionário é de mínimo ou máximo.

$$\text{Ponto de mínimo} \quad \frac{d^2\Psi(\mathbf{x})}{d\mathbf{x}^2} > 0$$

$$\text{Ponto de máximo} \quad \frac{d^2\Psi(\mathbf{x})}{d\mathbf{x}^2} < 0$$

A Figura 2.2 mostra a curva de uma função objetivo de apenas uma variável, com diversos pontos estacionários. Curvas com vários pontos estacionários são indesejadas em sistemas de otimização, pois a minimização do erro se torna uma tarefa ainda mais complexa. Para obter o ponto de mínimo global da curva deve-se aplicar métodos que contornem mínimos locais e pontos de inflexão, como pode ser visto pela Figura 2.2.

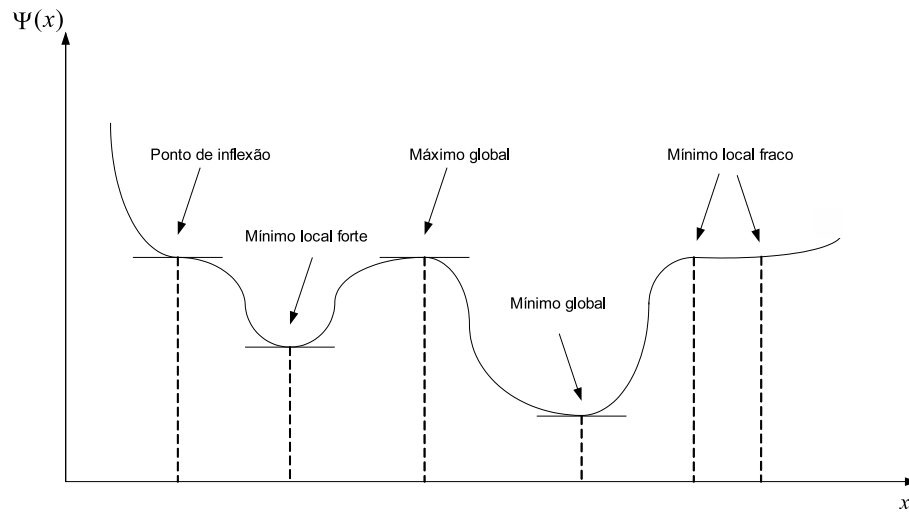


Figura 2.2: Exemplo de Função Objetiva com vários pontos estacionários.

2.3 *Minimax*

De posse da função objetivo do sistema, o próximo passo é a solução do seguinte problema de otimização:

$$\text{minimizar}_{\mathbf{x}} \Psi(\mathbf{x}) \quad (2.4)$$

ou seja, minimizar $\Psi(\mathbf{x})$ com relação a \mathbf{x} .

Existe uma grande variedade de métodos para solucionar este problema de minimização. A escolha do método ideal dependerá única e exclusivamente da aplicação e da característica da função objetivo, ou seja, cada aplicação exige determinadas características da função erro do sistema. Por exemplo, alguns projetos necessitam de uma alta precisão na busca pelo mínimo erro do sistema, outros já priorizam a velocidade em que se chega a esse valor. Para esta aplicação em específico, o enfoque está em obter equalizadores com respostas de atraso de grupo bastante fieis às curvas propostas. Para obter um resultado com estas características, se faz necessário minimizar a função objetivo de uma maneira que se obtenha um erro uniformemente distribuído sobre as amostras de frequência. Esse objetivo é alcançado com a implementação da técnica *minimax* juntamente com o método Quasi-Newton.

Com uma manipulação matemática e fazendo com que $p \rightarrow \infty$ na função objetivo baseada na norma L_p dado por (2.2), os objetivos da técnica *minimax* são alcançados.

$$\Psi(\mathbf{x}) = L_\infty = \lim_{p \rightarrow \infty} \left\{ \sum_{i=1}^K |e_i(\mathbf{x})|^p \right\}^{1/p} \quad (2.5)$$

A partir de (2.5) e aplicando as modificações previstas pela técnica, chega-se na função objetivo referenciada pela norma L_∞ . Dado que

$$\hat{E}(\mathbf{x}) = \max_{1 \leq i \leq K} |e_i(\mathbf{x})| \neq 0$$

seja o máximo valor de erro do vetor $E(\mathbf{x})$, pode-se chegar na função objetivo que agrega os princípios da técnica *minimax* da seguinte maneira.

$$\begin{aligned} L_\infty &= \frac{\hat{E}(\mathbf{x})}{\hat{E}(\mathbf{x})} \lim_{p \rightarrow \infty} \left\{ \sum_{i=1}^K |e_i(\mathbf{x})|^p \right\}^{1/p} \\ &= \hat{E}(\mathbf{x}) \lim_{p \rightarrow \infty} \left\{ \sum_{i=1}^K \left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})} \right]^p \right\}^{1/p} \end{aligned} \quad (2.6)$$

Estas alterações fazem com que cada termo da somatória em (2.6) seja menor ou igual a unidade, logo, quando $p \rightarrow \infty$ tem-se $L_\infty \rightarrow \hat{E}(\mathbf{x})$. Através dessa manipulação matemática, ao solucionar o problema de otimização proposto em (2.4), consegue-se além de minimizar todos os elementos de $E(\mathbf{x})$ também se consegue a minimização do máximo erro encontrado nesse vetor, dado por $\hat{E}(\mathbf{x})$.

A técnica *minimax* é implementada através de um algoritmo que incrementa o valor de p e analisa se as condições previstas pelo projeto foram atendidas a cada iteração. Considerando esse incremento gradativo no valor de p , pode-se desconsiderar o limite aplicado em (2.6), chegando-se na equação que determina a função objetivo para a técnica *minimax*.

$$\Psi_k(\mathbf{x}) = L_\infty = \hat{E}(\mathbf{x}) \left\{ \sum_{i=1}^K \left[\frac{|e_i(\mathbf{x})|}{\hat{E}(\mathbf{x})} \right]^p \right\}^{1/p} \quad (2.7)$$

onde k indica as iterações. Como já foi dito, após o processo de minimização da função objetivo, a técnica *minimax* tem a função de garantir um erro uniformemente distribuído na frequência. Nas próximas seções será apresentado o método utilizado para obter esta minimização, que é o método Quasi-Newton fundamentado pela teoria do tradicional método de Newton.

2.4 Método de Newton

O método de Newton, bem como o método do gradiente, comumente conhecido como *steepest descent* [Din97], fundamentam a grande variedade de algoritmos e métodos de otimização existentes. Esse método é aplicado a funções onde se tem a primeira e a segunda derivada contínuas, ou seja, aplicado à minimização de funções convexas quadráticas.

Em métodos de otimização, um dos primeiros passos é determinar o vetor correção, ou vetor direcional, $\boldsymbol{\delta}$, que a partir de um posição inicial, \mathbf{x}_0 , chega-se na posição, $\overset{\cup}{\mathbf{x}}$, que fornece o mínimo valor para a função objetivo. A cada iteração o algoritmo deve calcular o valor desse vetor que indica a direção de convergência para a posição ótima:

$$\overset{\cup}{\mathbf{x}} = \mathbf{x}_0 + \boldsymbol{\delta} \quad (2.8)$$

Considerando a função objetivo $\Psi(\mathbf{x})$, como sendo uma função de n variáveis, onde $\mathbf{x} = \mathbf{c} = [x_1 \ x_2 \ x_3 \ \dots \ x_n]^T$ e $\boldsymbol{\delta} = [\delta_1 \ \delta_2 \ \delta_3 \ \dots \ \delta_n]^T$ como sendo o vetor correção de \mathbf{x} . Se $\Psi(\mathbf{x})$ for uma função quadrática com derivadas de primeira e segunda ordem contínuas, a sua expansão por séries de Taylor fica:

$$\Psi(\overset{\cup}{\mathbf{x}}) = \Psi(\mathbf{x}_0 + \boldsymbol{\delta}) = \Psi(\mathbf{x}_0) + \sum_{i=1}^n \frac{\partial \Psi(\mathbf{x}_0)}{\partial x_i} \delta_i + \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \frac{\partial^2 \Psi(\mathbf{x}_0)}{\partial x_i \partial x_j} \delta_i \delta_j + o(\|\boldsymbol{\delta}\|_2^2) \quad (2.9)$$

Diferenciando (2.9) em relação à variável x_k , onde $k = 1, 2, \dots, n$ com o objetivo

de se determinar pontos estacionários, desprezando-se o termo $o(\|\boldsymbol{\delta}\|_2^2)$, tem-se:

$$\frac{\partial \Psi(\mathbf{x})}{\partial x_k} = \frac{\partial \Psi(\mathbf{x}_0)}{\partial x_k} + \sum_{i=1}^n \frac{\partial^2 \Psi(\mathbf{x}_0)}{\partial x_i \partial x_k} \delta_i \quad (2.10)$$

Como se trata de uma função quadrática, as derivadas de ordem superior a 2 são nulas, por isso foram desconsideradas em (2.10). Aplicando os conceitos envolvidos em (2.3), chega-se aos pontos estacionários:

$$\frac{\partial \Psi(\mathbf{x}_0)}{\partial x_k} + \sum_{i=1}^n \frac{\partial^2 \Psi(\mathbf{x}_0)}{\partial x_i \partial x_k} \delta_i = 0 \quad (2.11)$$

escrevendo em notação vetorial

$$\mathbf{g} + \mathbf{H}\boldsymbol{\delta} = 0 \quad (2.12)$$

onde

$$\mathbf{g} = \nabla \Psi(\mathbf{x}) = \left[\frac{\partial \Psi(\mathbf{x})}{\partial x_1} \quad \frac{\partial \Psi(\mathbf{x})}{\partial x_2} \quad \cdots \quad \frac{\partial \Psi(\mathbf{x})}{\partial x_n} \right]^T$$

e

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 \Psi(\mathbf{x})}{\partial x_1^2} & \frac{\partial^2 \Psi(\mathbf{x})}{\partial x_1 \partial x_2} & \cdots & \frac{\partial^2 \Psi(\mathbf{x})}{\partial x_1 \partial x_n} \\ \frac{\partial^2 \Psi(\mathbf{x})}{\partial x_2 \partial x_1} & \frac{\partial^2 \Psi(\mathbf{x})}{\partial x_2^2} & \cdots & \frac{\partial^2 \Psi(\mathbf{x})}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 \Psi(\mathbf{x})}{\partial x_n \partial x_1} & \frac{\partial^2 \Psi(\mathbf{x})}{\partial x_n \partial x_2} & \cdots & \frac{\partial^2 \Psi(\mathbf{x})}{\partial x_n^2} \end{bmatrix}$$

são o vetor *gradiente* e a matriz *Hessian* de $\Psi(\mathbf{x})$ respectivamente. A partir da matriz \mathbf{H} , que traz informações da derivada segunda da função objetivo, e do vetor \mathbf{g} , que representa a derivada primeira, chega-se no vetor $\boldsymbol{\delta}$ que indica a direção para o mínimo erro do sistema.

$$\boldsymbol{\delta} = -\mathbf{H}^{-1}\mathbf{g} \quad (2.13)$$

O vetor $\boldsymbol{\delta}$ somente indicará a direção de convergência do sistema se a matriz

Hessian for positiva definida [Gol96]. A cada iteração é necessário verificar e garantir que \mathbf{H} seja positiva definida. O resultado obtido em (2.13) será aplicado em (2.8) para chegar em:

$$\overset{\cup}{\mathbf{x}} = \mathbf{x}_0 - \mathbf{H}^{-1}\mathbf{g} \quad (2.14)$$

Esta equação representa a base do método de Newton. Uma matriz *Hessian* positiva definida é associada a um local de mínimo forte, ou melhor ainda, o vetor $\overset{\cup}{\mathbf{x}}$ representará a posição do mínimo erro do sistema. A Equação (2.14) pode ser levemente modificada para representar uma fórmula geral para um algoritmo iterativo

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k[-\mathbf{H}_k^{-1}\mathbf{g}_k] \quad (2.15)$$

O termo entre colchetes é uma “direção de busca” à posição de mínimo da função objetivo e o escalar α_k , comumente chamado de passo de adaptação, é obtido através de alguma técnica *line search* que será estudada posteriormente.

O método de Newton, apesar de ser base para outros métodos, apresenta algumas características que o torna não atrativo para aplicações práticas. Isso deve-se principalmente a necessidade de calcular a cada iteração o *gradiente*, a matriz *Hessian*, verificar e garantir que ela seja positiva definida, além de calcular a sua inversa. Tarefas essas, que dependendo do comprimento do vetor \mathbf{x} , se tornam impraticáveis e sem aplicações práticas. Um outro método que utiliza os princípios do tradicional método de Newton, que com algumas alterações consegue reduzir o esforço computacional e é mais indicado para minimização de funções não quadráticas, é o método Quasi-Newton.

2.5 Método Quasi-Newton

O método Quasi-Newton foi proposto para solucionar problemas inerentes ao método de Newton. Nesse algoritmo apenas a derivada primeira (*gradiente*) precisa ser calculada, não há necessidade de trabalhar com a matriz de *Hessian*, nem

mesmo sua inversa. Um dos princípios deste algoritmo é utilizar uma matriz \mathbf{S} como direção de busca do ponto de mínimo, essa matriz é uma aproximação da inversa da matriz de *Hessian*. A fórmula geral do algoritmo Quasi-Newton é a mesma do método de Newton, dada por (2.8), apenas com as alterações citadas:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\delta}_k \quad (2.16)$$

onde

$$\boldsymbol{\delta}_k = -\alpha_k \mathbf{S}_k \mathbf{g}_k$$

Em meados da década de 60, algumas fórmulas para obtenção iterativa da matriz \mathbf{S} foram propostas. As fórmulas *rank one* e *rank two* merecem destaque dentre todas as outras propostas. O objetivo destas fórmulas iterativas é conseguir que \mathbf{S} se aproxime de \mathbf{H}^{-1} , à medida que se aumenta o número de iterações, conseqüentemente, elas se igualam para algoritmos com elevado número de iterações [Fle87]. Uma conseqüência bastante importante a respeito desta equivalência é que o método Quasi-Newton não necessita de uma técnica *line search* exata para conseguir convergência. A fórmula *rank one* é obtida através de um método iterativo, onde uma matriz *rank one* simétrica, \mathbf{A}_k , é adicionada à \mathbf{S}_k para se obter \mathbf{S}_{k+1} .

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \mathbf{A}_k \quad (2.17)$$

Pode-se assumir que $\mathbf{S}_{k+1} [\mathbf{g}_{k+1} - \mathbf{g}_k] \approx \boldsymbol{\delta}_k$, dado que $\mathbf{S} \approx \mathbf{H}^{-1}$ para elevado número de iterações, [Fle87]. A partir desta consideração, chega-se na fórmula *rank one* proposta independentemente por Broyden(1967), Davidon(1968), Fiacco e McCormick(1968), Murtagh e Sargent(1969) e Wolfe(1968).

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \frac{(\boldsymbol{\delta}_k - \mathbf{S}_k \boldsymbol{\gamma}_k)(\boldsymbol{\delta}_k - \mathbf{S}_k \boldsymbol{\gamma}_k)^T}{(\boldsymbol{\delta}_k - \mathbf{S}_k \boldsymbol{\gamma}_k)^T \boldsymbol{\gamma}_k} \quad (2.18)$$

onde $\boldsymbol{\gamma}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$. Alguns inconvenientes da fórmula *rank one*, dada por

(2.18), são a não garantia de se obter uma matriz \mathbf{S}_{k+1} positiva definida a cada iteração e a possibilidade do denominador de (2.18) se aproximar de zero, com isso a equação é considerada não definida.

As fórmulas *rank two* são mais flexíveis, pois se preocupam em obter matrizes \mathbf{S}_{k+1} positiva definida a cada iteração. As fórmulas *rank two* mais importantes foram sugeridas por Davidon-Fletcher-Powell (DFP), no início da década de 60, e por Broyden-Fletcher-Goldfarb-Shanno (BFGS) em 1970, [Fle87]. Uma propriedade importantíssima destas fórmulas é que elas garantem que se \mathbf{S}_k for positiva definida, a matriz \mathbf{S}_{k+1} também será. As fórmulas de recorrência para os métodos DFP e BFGS são, respectivamente:

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k} - \frac{\mathbf{S}_k \boldsymbol{\gamma}_k \boldsymbol{\gamma}_k^T \mathbf{S}_k}{\boldsymbol{\gamma}_k^T \mathbf{S}_k \boldsymbol{\gamma}_k} \quad (2.19)$$

e

$$\mathbf{S}_{k+1} = \mathbf{S}_k + \left(1 + \frac{\boldsymbol{\gamma}_k^T \mathbf{S}_k \boldsymbol{\gamma}_k}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k}\right) \frac{\boldsymbol{\delta}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k} - \frac{\boldsymbol{\delta}_k \boldsymbol{\gamma}_k^T \mathbf{S}_k + \mathbf{S}_k \boldsymbol{\gamma}_k \boldsymbol{\delta}_k^T}{\boldsymbol{\gamma}_k^T \boldsymbol{\delta}_k} \quad (2.20)$$

Geralmente, \mathbf{S}_k é iniciada como sendo uma matriz identidade [Gol96]. As fórmulas *rank two* demonstraram bons resultados na prática e vem sendo largamente utilizadas. A fórmula BFGS apresentou bons resultados em situações práticas, melhores até mesmo que a DFP. Além do mais, foi provado que a fórmula BFGS consegue realizar convergência global com técnicas *line search* inexatas, resultado que ainda não foi apresentado para a fórmula DFP, [Fle87]. Neste trabalho foi utilizado para atualização da matriz \mathbf{S} a fórmula BFGS.

Para garantir que \mathbf{S}_{k+1} seja positiva definida, deve-se verificar a seguinte condição em ambas as equações [Fle87]:

$$\boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k = \boldsymbol{\delta}_k^T \mathbf{g}_{k+1} - \boldsymbol{\delta}_k^T \mathbf{g}_k > 0 \quad (2.21)$$

A matriz \mathbf{S} e o vetor \mathbf{g} agregam informações que indicam a direção para a convergência do algoritmo, ou seja, a direção para se chegar no ponto de mínimo

da função objetivo. A matriz \mathbf{S} é obtida através de um método iterativo e definida por (2.20) ou (2.19), enquanto o vetor *gradiente* é determinado através da diferenciação da função objetivo, dada por (2.7), proposta pela técnica *minimax*.

$$\mathbf{g} = \nabla \Psi_k(\mathbf{x}) = \left\{ \sum_{i=1}^K \left[\frac{|e_i(\mathbf{x})|}{\bar{E}(\mathbf{x})} \right]^p \right\}^{(1/p)-1} \sum_{i=1}^K \left[\frac{|e_i(\mathbf{x})|}{\bar{E}(\mathbf{x})} \right]^{p-1} \nabla |e_i(\mathbf{x})| \quad (2.22)$$

O termo $\nabla |e_i(\mathbf{x})|$ é definido por:

$$\nabla |e_i(\mathbf{x})| = \text{sgn } e_i(\mathbf{x}) \nabla e_i(\mathbf{x}) \quad (2.23)$$

onde

$$\text{sgn } e_i(\mathbf{x}) = \begin{cases} 1 & \text{se } e_i(\mathbf{x}) \geq 0 \\ -1 & \text{caso contrário} \end{cases}$$

O vetor $\nabla e_i(\mathbf{x})$ é determinado pela diferenciação da função erro em relação à \mathbf{x} , ou seja, diferenciação em relação aos coeficientes do filtro *allpass*, já que $\mathbf{x} = [c_{01} \ c_{11} \ c_{02} \ c_{12} \ \dots \ c_{1M}]^T$.

$$\begin{aligned} \frac{\partial e_i(\mathbf{x})}{\partial c_{0l}} &= \frac{U_{0l} + U_{1l} \cos(w_i T) + U_{2l} \cos^2(w_i T) + U_{3l} \cos^3(w_i T)}{[P_l(w_i)]^2} \\ \frac{\partial e_i(\mathbf{x})}{\partial c_{1l}} &= \frac{V_{0l} + V_{1l} \cos(w_i T) + V_{2l} \cos^2(w_i T) + V_{3l} \cos^3 w_i T}{[P_l(w_i)]^2} \end{aligned}$$

onde

$$\begin{aligned} U_{0l} &= 4[(1 - c_{0l})^2 - c_{0l}c_{1l}^2], \\ U_{1l} &= -2c_{1l}(1 + 6c_{0l} + c_{0l}^2 + c_{1l}^2), \\ U_{2l} &= -8(1 + c_{0l}^2 + c_{1l}^2), \\ U_{3l} &= -8c_{1l}, \\ V_{0l} &= -4c_{1l}(1 - c_{0l})(1 + c_{0l}), \\ V_{1l} &= -2(1 - c_{0l})(1 + 6c_{0l} + c_{0l}^2 + c_{1l}^2), \\ V_{2l} &= 0, \\ V_{3l} &= 8(1 - c_{0l})c_{0l}. \end{aligned}$$

para $l = 1, 2, \dots, M$ e $i = 1, 2, \dots, K$.

Por reunir todas as características mencionadas nesta seção, o método Quasi-Newton será o responsável por encontrar o valor de $\overset{U}{\mathbf{x}}$ na tarefa de minimizar o erro do sistema. Após detalhar todo o equacionamento necessário para implementar algoritmos que realizam as tarefas do método Quasi-Newton, se torna necessário analisar a técnica *line search*, sem a qual não se consegue a convergência na etapa de minimização.

2.6 *Line Search*

Será abordado agora uma técnica bastante empregada em sistemas de otimização e que, geralmente, é uma etapa do processo onde se tem grande, senão o maior esforço computacional. O objetivo da técnica *line search* é a obtenção de um passo de adaptação ótimo, $\overset{U}{\alpha}$, necessário para se chegar no mínimo valor possível para a função objetivo. Uma característica fundamental desta técnica é analisar a função objetivo como sendo uma função de α e, a partir disso, obter o valor de $\overset{U}{\alpha}$. Quanto maior a precisão da técnica *line search*, maior o esforço computacional dedicado à tarefa, por isso busca-se maneiras de obter o valor de $\overset{U}{\alpha}$ com o menor esforço possível. A partir desse cenário, geralmente, utiliza-se técnicas *line search* inexatas, o que quer dizer que não se chega no valor exato do passo de adaptação ótimo, mas sim em um valor aproximado, que em muitos métodos de minimização é o suficiente para se conseguir a convergência. O método Quasi-Newton se mostrou bastante robusto às imprecisões da técnica *line search*, mesmo com alguma imprecisão no valor de $\overset{U}{\alpha}$, conseguiu-se a convergência do mesmo.

O princípio da técnica *line search* é extrair informações da própria função objetivo e de sua derivada em relação a α com a finalidade de determinar o valor máximo, α_{max} , e o mínimo, α_{min} , que $\overset{U}{\alpha}$ pode assumir. A determinação de α_{max} e α_{min} está relacionado ao grau de precisão da técnica *line search*.

Todo o equacionamento necessário para se aplicar a técnica *line search*, pode

ser obtido partindo da expressão:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha \mathbf{d}_k$$

O termo $\mathbf{d}_k = -\mathbf{S}_k \mathbf{g}_k$ indica a direção para a minimização da função objetivo, expressa em função de α por $\Psi(\mathbf{x}_{k+1}) = \Psi(\mathbf{x}_k + \alpha \mathbf{d}_k)$. O objetivo da técnica *line search* é a partir de um valor inicial, α_0 , obter o valor de $\hat{\alpha}$. Analisando a aproximação linear de $\Psi(\mathbf{x}_{k+1})$ por Séries de Taylor, Equação (2.9), e aplicando uma notação vetorial tem-se:

$$\Psi(\mathbf{x}_{k+1}) = \Psi(\mathbf{x}_k) + \alpha \mathbf{g}_k^T \mathbf{d}_k \quad (2.24)$$

A partir de (2.24), chega-se na expressão que determina a inclinação de $\Psi(\mathbf{x}_{k+1})$:

$$\alpha \mathbf{g}_k^T \mathbf{d}_k = \Psi(\mathbf{x}_{k+1}) - \Psi(\mathbf{x}_k)$$

$$\mathbf{g}_k^T \mathbf{d}_k = \frac{\Psi(\mathbf{x}_{k+1}) - \Psi(\mathbf{x}_k)}{\alpha}$$

Para uma variação de α tendendo a zero tem-se:

$$\mathbf{g}_k^T \mathbf{d}_k = \left. \frac{d\Psi(\mathbf{x}_k + \alpha \mathbf{d}_k)}{d\alpha} \right|_{\alpha=0} \quad (2.25)$$

A Equação (2.25) representa a inclinação de $\Psi(\mathbf{x}_{k+1})$ na origem do eixo α . Essa inclinação é representada pela linha A mostrada na Figura 2.3. A inclinação da função objetivo na origem, $\alpha = 0$, será utilizada como referência para a obtenção dos valores máximo e mínimo que $\hat{\alpha}$ pode assumir.

Para se determinar o valor de α_{max} , utiliza-se o parâmetro ρ para variar a inclinação de $\Psi(\mathbf{x}_{k+1})$ na origem, dado por $\mathbf{g}_k^T \mathbf{d}_k$. Variando-se o valor de ρ de 0 a 0,5, pode-se chegar a uma inclinação que varia de 0 a $0,5 \mathbf{g}_k^T \mathbf{d}_k$, essa nova inclinação está exemplificada na Figura 2.3 pela linha B. O valor de α_{max} é de-

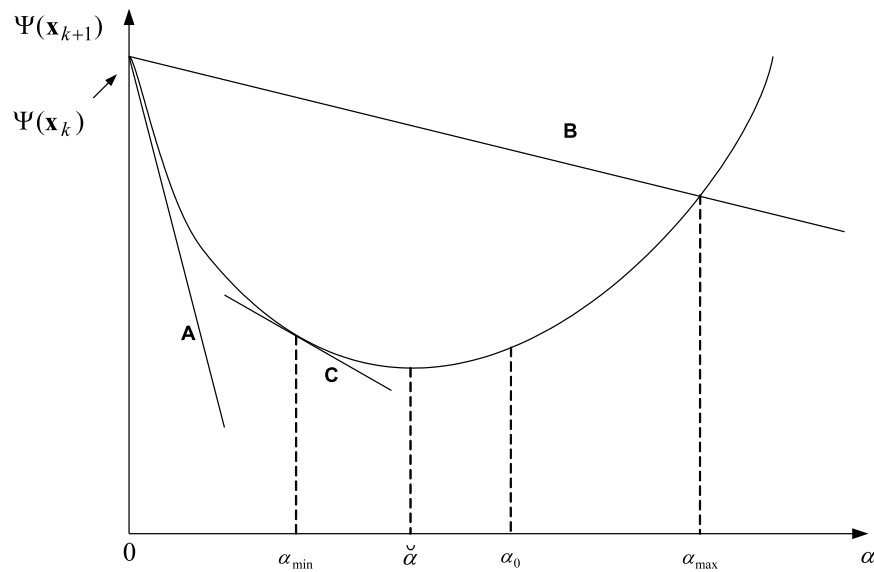


Figura 2.3: Técnica Line Search.

terminado pelo ponto de cruzamento da linha B com a função objetivo, analisada aqui em função de α . A equação que relaciona o valor da função objetivo a cada valor de α obtido com a inclinação da curva, é dada por:

$$\Psi(\mathbf{x}_{k+1}) = \Psi(\mathbf{x}_k) + \rho\alpha\mathbf{g}_k^T\mathbf{d}_k \quad (2.26)$$

O valor do parâmetro ρ está indiretamente relacionado ao valor de α_{max} . Através da Figura 2.4 pode-se notar que quanto menor o valor de ρ maior será o valor de α_{max} .

A técnica *line search* em estudo utiliza informações da inclinação de $\Psi(\mathbf{x}_{k+1})$ na origem para a obtenção do limitante inferior de $\overset{U}{\alpha}$. O valor de α_{min} deve ser maior que zero e menor que $\overset{U}{\alpha}$, com isso, a inclinação da função objetivo no ponto α_{min} deve ser mais positiva (ou menos negativa) que a inclinação na origem, $\mathbf{g}_k^T\mathbf{d}_k$, e mais negativa que a inclinação em $\overset{U}{\alpha}$, como pode ser analisado pelo exemplo da Figura 2.3, onde a linha C representa a inclinação no ponto α_{min} . Para garantir que as condições mencionadas acima sejam atendidas, utiliza-se o parâmetro σ para analisar a inclinação no ponto α_{min} , dado a inclinação na origem. Essa constante assume valores $0 < \sigma < 1$, o que garante $0 < \alpha_{min} < \overset{U}{\alpha}$. A Equação

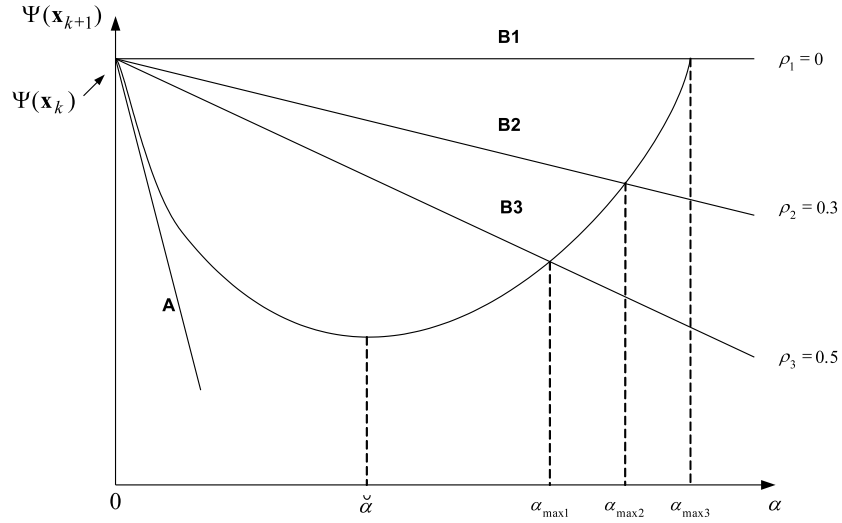


Figura 2.4: Variação do parâmetro ρ .

(2.27) relaciona a inclinação da função em um determinado ponto com a inclinação de $\Psi(\mathbf{x}_{k+1})$ na origem.

$$\mathbf{g}_{k+1}^T \mathbf{d}_k = \sigma \mathbf{g}_k^T \mathbf{d}_k \quad (2.27)$$

Como pode ser visto, o valor de σ está diretamente relacionado ao valor de α_{min} , é fácil notar que para $\sigma \rightarrow 0$ tem-se $\alpha_{min} \rightarrow \overset{\cup}{\alpha}$ e para $\sigma \rightarrow 1$ tem-se $\alpha_{min} \rightarrow 0$.

As Equações (2.26) e (2.27) definem o intervalo $[\alpha_{min}, \alpha_{max}]$, que é o intervalo de valores onde se tem uma aceitável aproximação de $\overset{\cup}{\alpha}$. Pela teoria apresentada nota-se que a precisão da técnica pode ser configurada alterando-se os valores das constantes σ e ρ , com isso, consegue-se controlar os limites do intervalo $[\alpha_{min}, \alpha_{max}]$. Os valores de σ e ρ devem ser especificados atendendo exigências do método de minimização empregado, devendo analisar-se a robustez do método frente a imprecisões da técnica *line search*. Para garantir que $\alpha_{min} < \alpha_{max}$ deve-se ter $\sigma \geq \rho$.

Na determinação de um valor para α , que minimize a função objetivo, primeiramente deve-se através de alguma técnica, estimar o valor inicial de $\overset{\cup}{\alpha}$, isto é, α_0 . A partir deste valor deve-se checar a condição estabelecida por (2.28), ou seja checar se o valor de $\Psi(\mathbf{x}_{k+1})$ para $\alpha = \alpha_0$ é menor que o valor estimado por

(2.26) para α_0 :

$$\Psi(\mathbf{x}_{k+1}) \leq \Psi(\mathbf{x}_k) + \rho\alpha_0 \mathbf{g}_k^T \mathbf{d}_k \quad (2.28)$$

Se a condição expressa por (2.28) for respeitada, garante-se que $\alpha_0 \leq \alpha_{max}$. Para verificar se $\alpha_{min} \leq \alpha_0$ a relação imposta por (2.29) deve ser satisfeita:

$$\mathbf{g}_{k+1}^T \mathbf{d}_k \geq \sigma \mathbf{g}_k^T \mathbf{d}_k \quad (2.29)$$

ou seja, a inclinação em $\alpha = \alpha_0$ deve ser mais positiva que a inclinação da curva no ponto α_{min} , exemplificada pela linha C exposta na Figura 2.3. Caso as condições dadas por (2.28) e (2.29) sejam satisfeitas, garante-se que $\alpha_{min} \leq \alpha_0 \leq \alpha_{max}$, com isso pode-se considerar que α_0 é uma aproximação aceitável de $\overset{\cup}{\alpha}$.

Se apenas a condição expressa por (2.28) não for atendida, $\alpha_0 > \alpha_{max}$, alguma fórmula de interpolação deve ser empregada para chegar a um novo valor de α_0 que satisfaça a condição. Caso apenas a condição expressa por (2.29) não for atendida, tem-se que $\alpha_0 < \alpha_{min}$, com isso alguma formula de extrapolação deve ser aplicada. Caso as condições expressas por (2.28) e (2.29) não forem respeitadas, o valor de α_0 está fora do intervalo $[\alpha_{min}, \alpha_{max}]$ e ele não é considerado uma boa aproximação para $\overset{\cup}{\alpha}$. Dado isso, fórmulas de interpolação e extrapolação devem ser utilizadas até que se consiga satisfazer as condições mencionadas acima, quando essas condições forem garantidas a busca por $\overset{\cup}{\alpha}$ está terminada.

Assumindo que $\Psi(\mathbf{x}_{k+1})$ tenha uma representação quadrática quando analisado em função de α , pode-se chegar a fórmulas de extrapolação e interpolação, [Fle87], a partir do valor da função objetivo e de sua derivada primeira nos pontos α_0 e α_L , onde $\alpha_0 > \alpha_L$. Para o caso de $\alpha_0 > \alpha_{max}$ a seguinte fórmula de interpolação pode ser aplicada:

$$\overset{\cup}{\alpha}_0 = \alpha_L + \frac{(\alpha_0 - \alpha_L)^2 \Psi'_L}{2[\Psi_L - \Psi_0 + (\alpha_0 - \alpha_L) \Psi'_L]} \quad (2.30)$$

Caso $\alpha_0 < \alpha_{min}$, a fórmula de extrapolação pode ser usada:

$$\overset{u}{\alpha}_0 = \alpha_0 + \frac{(\alpha_0 - \alpha_L)\Psi'_0}{(\Psi'_L - \Psi'_0)} \quad (2.31)$$

onde $\overset{u}{\alpha}_0$ é o novo valor estimado de α_0 e

$$\begin{aligned} \Psi_L &= \Psi(\mathbf{x}_k + \alpha_L \mathbf{d}_k) \\ \Psi'_L &= \Psi'(\mathbf{x}_k + \alpha_L \mathbf{d}_k) = \mathbf{g}(\mathbf{x}_k + \alpha_L \mathbf{d}_k)^T \mathbf{d}_k \\ \Psi_0 &= \Psi(\mathbf{x}_k + \alpha_0 \mathbf{d}_k) \\ \Psi'_0 &= \Psi'(\mathbf{x}_k + \alpha_0 \mathbf{d}_k) = \mathbf{g}(\mathbf{x}_k + \alpha_0 \mathbf{d}_k)^T \mathbf{d}_k \end{aligned}$$

2.7 Restrições

Até esta seção foram apresentadas técnicas e métodos para solucionar o problema da minimização da função objetivo. A implementação em conjunto das técnicas *minimax* e *line search* e do método Quasi-Newton é suficiente para solucionar o problema de otimização proposto em (2.4). Porém, por se tratar da implementação de filtros digitais *IIR*, não basta apenas minimizar o erro do sistema, deve-se chegar a coeficientes que proporcionem filtros estáveis. Para esta aplicação em específico, tem-se que chegar em coeficientes que pertençam à região de estabilidade definida pela Figura 1.5 e pelas condições apresentadas em (1.25). Para contornar este problema há a necessidade de implementar um algoritmo que estabeleça algumas restrições, ou seja, um algoritmo que a cada minimização de $\Psi(\mathbf{x})$ teste se o resultado encontrado fornece um filtro estável.

O princípio deste algoritmo é, primeiramente, considerar o equalizador com apenas uma seção, fazendo $M = 1$ em (1.13), e inicializar o algoritmo de minimização com os valores de $\mathbf{x}_0 = [c_{01} \quad c_{11}]^T$ extraídos de uma Tabela contendo pontos pertencentes a região de estabilidade. Esse processo é repetido até que se obtenha um resultado bem sucedido na etapa de minimização, ou seja, até que

se obtenha um equalizador estável. A cada processo de minimização bem sucedido, incrementa-se o número de seções do equalizador e utiliza-se os coeficientes obtidos na mais recente etapa de minimização bem sucedida para se chegar nos valores que inicializarão a nova seção do filtro. Com o aumento do número de seções do equalizador tem-se uma melhora na qualidade do mesmo, esse algoritmo é encerrado quando se obtém um filtro *allpass* que atenda as exigências da resposta do atraso de grupo, precisão e estabilidade do projeto.

O princípio do algoritmo que implementa as restrições foi proposto por Andreas Antoniou em [Ant93], a versão utilizada neste trabalho é ligeiramente modificada da versão original, onde houve a necessidade de se alterar as condições para o incremento do número de seções. Para as duas primeiras seções do filtro, o único critério utilizado para incrementar o número de seções foi a obtenção de resultados estáveis, não se importando com a qualidade do equalizador obtido. Extensivos testes com o algoritmo mostram que os coeficientes, obtidos após a minimização de $\Psi(\mathbf{x})$, tendem a formar uma célula em torno da região de estabilidade, com isso consegue-se uma aproximação forçada que resulta em filtros estáveis. O princípio de incrementar gradativamente o número de seções do equalizador e a necessidade de garantir que as condições propostas em (1.25) sejam respeitadas, justificam a implementação do filtro *allpass* em sua forma cascadeada com M seções de 2^{a} ordem, dado por (1.13).

Detalhes referente à implementação deste algoritmo serão apresentados no próximo Capítulo, bem como o detalhamento de todos os parâmetros, equações e dados necessários para o seu entendimento.

2.8 Conclusão

Com a utilização da técnica *minimax* consegue-se um erro uniformemente distribuído na banda de interesse do filtro, neste trabalho foi utilizado o método Quasi-Newton juntamente com a técnica *line search* para a minimização da função

objetivo do sistema. O método Quasi-Newton se mostra robusto a minimização de funções não quadráticas e a imprecisões do método *line search*, o que proporciona um algoritmo com menos esforço computacional. Para finalizar, utilizou-se um algoritmo que implementa restrições necessárias para se obter um equalizador estável a cada etapa de minimização da função objetivo.

Neste Capítulo foi apresentado as técnicas e métodos utilizados, bem como os objetivos que se pretende alcançar com cada um, além de todo o equacionamento necessário para a implementação. No Capítulo 3 será apresentado detalhes dos algoritmos implementados para alcançar os objetivos propostos por cada método e técnica.

Capítulo 3

Algoritmo Final

3.1 Introdução

Este Capítulo mostrará como as técnicas e métodos, apresentados no Capítulo 2, foram implementados e interligados para compor o algoritmo final. Este algoritmo é responsável por retornar os coeficientes do filtro *allpass* estável que atenda a uma específica máscara de atraso de grupo. Todos os parâmetros e constantes utilizados serão detalhados, bem como a apresentação de um diagrama de blocos para facilitar o entendimento global do algoritmo.

3.2 Relação entre os métodos e técnicas

O algoritmo final pode ser entendido como a combinação de quatro algoritmos, onde cada um implementa um dos quatro métodos descritos no Capítulo 2. Analisando as etapas do algoritmo final, pode-se dizer que o bloco responsável por implementar as restrições (necessárias para se garantir estabilidade) realiza também a inicialização e o término do algoritmo final. Isto se deve ao fato, deste método indicar quais são os valores de \mathbf{x} necessários para a inicialização do algoritmo. E também determinar se o mesmo já atingiu resultados satisfatórios de ondulação na resposta de atraso de grupo.

Toda vez que se incrementa uma seção do filtro deve-se realizar a minimização da função objetivo, a fim de aproximar o atraso de grupo do filtro atual à máscara especificada para o projeto. Essa minimização é feita de acordo com implicações da técnica *minimax* e realizada pelo Algoritmo Quasi-Newton Prático proposto por Andreas Antoniou em [Ant93]. Este algoritmo prático nada mais é do que a implementação do tradicional método Quasi-Newton juntamente com uma técnica *line search* inexata. Esta última técnica é implementada através de rotinas baseadas em um algoritmo proposto por Fletcher, [Fle87], com algumas pequenas modificações, como por exemplo o valor inicial para o passo de adaptação α e o valor dos parâmetros ρ e σ , que determinam a precisão do método. A Figura 3.1 mostra um diagrama indicando como os quatro métodos se relacionam. Pelo diagrama nota-se que o algoritmo responsável por implementar as

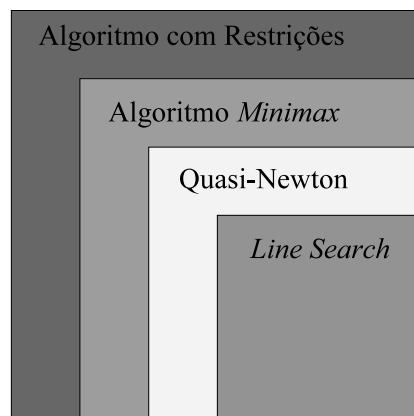


Figura 3.1: Diagrama dos Algoritmos.

restrições é o responsável pela inicialização do algoritmo e também por acionar o algoritmo *minimax*, que por sua vez irá determinar o início e término do processo de minimização, através do método Quasi-Newton e da técnica *line search*.

3.3 Implementação do Algoritmo Final

Todas as técnicas e parâmetros utilizados na implementação do algoritmo final serão apresentadas nesta seção. A apresentação do algoritmo será feita na maneira

com que foi implementado, ou seja, mostrando exatamente como as diferentes técnicas e métodos se relacionam, tentando assim facilitar o entendimento de todo o processo.

A primeira consideração a ser levantada é a respeito da forma com que foi implementado o filtro *allpass*, nota-se em (1.13) que o filtro é analisado em sua forma cascadeada com M seções de 2^{a} ordem. Esta consideração é importante para o bom entendimento do algoritmo final, pois o seu princípio básico é incrementar gradativamente o número de seções do equalizador até que se obtenha resultados estáveis e satisfatórios.

Primeiramente, é preciso especificar a resposta de atraso de grupo que se deseja obter, τ_0 , bem como as frequências que determinam as amostras desta resposta, $\omega_1, \omega_2, \dots, \omega_K$. Assumindo o filtro *allpass* com apenas uma seção ($M = 1$), tem-se

$$H_{ap}(z) = \frac{1 + c_{11}z + c_{01}z^2}{c_{01} + c_{11}z + z^2}$$

onde $\mathbf{x}_0 = [c_{01} \quad c_{11}]^T$ representa o valor de \mathbf{x} que iniciará o algoritmo. Para a inicialização, carrega-se pontos da Tabela 3.1 até que se obtenha um resultado bem sucedido após a etapa de minimização da função objetivo.

Tabela 3.1: Pontos (c_{01}, c_{11}) pertencentes a região de estabilidade.

No.	Ponto	No.	Ponto	No.	Ponto
1	(0,30;0,30)	1A	(0,25;0,50)	1B	(0,50;0,25)
2	(0,70;0,70)	2A	(0,50;0,75)	2B	(0,75;0,50)
3	(0,70;1,30)	3A	(0,50;1,25)	3B	(0,75;1,50)
4	(-0,30;0,30)	4A	(-0,25;0,50)	4B	(-0,50;0,25)
5	(0,30;-0,30)	5A	(0,25;-0,50)	5B	(0,50;-0,25)
6	(0,70;-0,70)	6A	(0,50;-0,75)	6B	(0,75;-0,50)
7	(0,70;-1,30)	7A	(0,50;-1,25)	7B	(0,75;-1,50)
8	(-0,30;-0,30)	8A	(-0,25;-0,50)	8B	(-0,50;-0,25)

Esta Tabela contém pontos pertencentes a região de estabilidade do sistema dado pela Figura 1.5. O processo de minimização é realizado através da implementação de algoritmos que aplicam as definições das técnicas *minimax* e *line*

search e do método Quasi-Newton. Esse processo de minimização nada mais é do que obter o valor de $\hat{\mathbf{x}}$ que minimize o erro do sistema. Os objetivos da técnica *minimax* são conseguidos com a implementação do seguinte algoritmo:

Algoritmo *Minimax*

1. Entre com os valores de \mathbf{x}_0 , valores extraídos da Tabela 3.1, e ε .
 2. Assuma $k = 1$, $p = 2$, $\mu = 2$ e $\hat{E}_o = 10^{99}$.
 3. Utilizando \mathbf{x}_0 como valor inicial, minimize, em relação a \mathbf{x} , a função objetivo dada por (2.7) para se obter $\hat{\mathbf{x}}_k$.
 4. Assuma $\hat{E}_k = \hat{E}(\hat{\mathbf{x}}_k)$.
 5. Se $|\hat{E}_{k-1} - \hat{E}_k| \geq \varepsilon$ ou $\hat{\mathbf{x}}_k$ não atender as condições estabelecidas em (1.25), então
 - Se $k < K_{max}$, então
 - Assuma $p = p + \mu$, $k = k + 1$ e volte ao passo 3;
 - Senão,
 - Atualize o valor de \mathbf{x}_0 com valores da Tabela 3.1 e volte ao Passo 2.
- Senão,
- Finalize o processo *minimax*, armazene $\hat{\mathbf{x}}_k$ e \hat{E}_k e incremente o número de seções do equalizador.

Nos passos 1 e 5 do algoritmo *minimax* o valor de ε deve ser especificado de acordo com as características do projeto. O valor desta constante estabelece qual a variação do máximo erro, \hat{E}_k , pode ser considerada aceitável quando se incrementa o valor de p . Neste trabalho, para o filtro com uma e duas seções, o

valor desta constante foi considerado $\varepsilon = 10^{99}$, ou seja, o seu valor não interfere no término do algoritmo *minimax*, o parâmetro mais importante nestas duas etapas é obter filtros estáveis. Para o filtro com mais de duas seções, o valor de ε se torna importante para o processo de refinamento do equalizador. Foram realizados testes e o valor de $\varepsilon = 10^{-2}$ garantiu bons resultados. As constantes p e μ são utilizadas para obter a função objetivo baseada na norma L_p , dado por (2.7). A constante K_{max} é o número máximo de iterações permitidas com um valor da Tabela 3.1, caso não se obtenha um resultado estável até K_{max} iterações, deve-se iniciar o algoritmo com outro valor da Tabela até que um resultado estável seja obtido, na implementação foi utilizado $K_{max} = 50$.

O processo de minimização da função $\Psi(\mathbf{x})$, referenciado no passo 3 do algoritmo *minimax*, é conseguido utilizando-se o algoritmo Prático de Quasi-Newton. Neste algoritmo foi implementado o método tradicional de Quasi-Newton, juntamente com a técnica *line search* inexata. O algoritmo Prático de Quasi-Newton é exposto da seguinte maneira.

Algoritmo Prático de Quasi-Newton, proposto por Andreas Antoniou

1. Inicializando o algoritmo

- (a) A partir de \mathbf{x}_0 , p e ε_1 .
- (b) Assuma $k = m = 0$.
- (c) Assuma $\rho = 0.1$, $\sigma = 0.7$, $\tau = 0.1$, $\chi = 0.75$, $\hat{M} = 600$ e $\varepsilon_2 = 10^{-10}$.
- (d) Assuma $\mathbf{S}_0 = \mathbf{I}_n$.
- (e) Calcule $\Psi_0 = \Psi(\mathbf{x}_0)$ e $\mathbf{g}_0 = \nabla\Psi_0$ utilizando (2.7) e (2.22) respectivamente. Assuma $m = m + 2$.
- (f) Assuma $\Psi_{00} = \Psi_0$ e $\Delta\Psi_0 = \Psi_0$.

2. Inicializando *line search*.

- (a) Assuma $\mathbf{d}_k = -\mathbf{S}_k \mathbf{g}_k$.
- (b) Assuma $\alpha_L = 0$ e $\alpha_U = 10^{99}$.
- (c) Assuma $\Psi_L = \Psi_0$ e calcule $\Psi'_L = \mathbf{g}(\mathbf{x}_k + \alpha_L \mathbf{d}_k)^T \mathbf{d}_k$.
- (d) Estimando α_0 .
 Se $|\Psi'_L| > \varepsilon_2$, então
 - $\alpha_0 = \frac{-2\Delta\Psi_0}{\Psi'_L}$;
 Senão,
 - $\alpha_0 = 1$.
- (e) Se $\alpha_0 \leq 0$ ou $\alpha_0 > 1$, então
 - $\alpha_0 = 1$.

3. Assuma $\delta_k = \alpha_0 \mathbf{d}_k$ e calcule $\Psi_0 = \Psi(\mathbf{x}_k + \delta_k)$. Assuma $m = m + 1$.

4. Interpolação do *line search*

Se $\Psi_0 > \Psi_L + \rho(\alpha_0 - \alpha_L)\Psi'_L$ e $|(\Psi_L - \Psi_0)| > \varepsilon_2$ e $m < \hat{M}$, então

- (a) Se $\alpha_0 < \alpha_U$, então $\alpha_U = \alpha_0$.
- (b) Calcule $\overset{\cup}{\alpha}_0$ utilizando (2.30).
- (c) Calcule $\overset{\cup}{\alpha}_{0L} = \alpha_L + \tau(\alpha_U - \alpha_L)$.
 Se $\overset{\cup}{\alpha}_0 < \overset{\cup}{\alpha}_{0L}$, então $\overset{\cup}{\alpha}_0 = \overset{\cup}{\alpha}_{0L}$.
- (d) Calcule $\overset{\cup}{\alpha}_{0U} = \alpha_U - \tau(\alpha_U - \alpha_L)$.
 Se $\overset{\cup}{\alpha}_0 > \overset{\cup}{\alpha}_{0U}$, então $\overset{\cup}{\alpha}_0 = \overset{\cup}{\alpha}_{0U}$.
- (e) Assuma $\alpha_0 = \overset{\cup}{\alpha}_0$ e volte ao passo 3.

5. Calcule $\Psi'_0 = \mathbf{g}(\mathbf{x}_k + \alpha_0 \mathbf{d}_k)^T \mathbf{d}_k$. Assuma $m = m + 1$.

6. Extrapolação do *line search*.

Se $\Psi'_0 < \sigma \Psi'_L$ e $|(\Psi_L - \Psi_0)| > \varepsilon_2$ e $m < \hat{M}$, então

(a) Calcule $\Delta\alpha_0 = \frac{(\alpha_0 - \alpha_L)\Psi'_0}{(\Psi'_L - \Psi'_0)}$, (2.31).

(b) Se $\Delta\alpha_0 \leq 0$, então

- $\overset{\cup}{\alpha}_0 = 2\alpha_0$;

Senão,

- $\overset{\cup}{\alpha}_0 = \alpha_0 + \Delta\alpha_0$.

(c) Calcule $\overset{\cup}{\alpha}_{0U} = \alpha_0 + \chi(\alpha_U - \alpha_0)$;

Se $\overset{\cup}{\alpha}_0 > \overset{\cup}{\alpha}_{0U}$, então $\overset{\cup}{\alpha}_0 = \overset{\cup}{\alpha}_{0U}$.

(d) Assuma $\alpha_L = \alpha_0$, $\alpha_0 = \overset{\cup}{\alpha}_0$, $\Psi_L = \Psi_0$, $\Psi'_L = \Psi'_0$ e volte ao passo 3.

7. Critério de Parada e saída de resultados.

(a) Assuma $\mathbf{x}_{k+1} = \mathbf{x}_k + \boldsymbol{\delta}_k$.

(b) Assuma $\Delta\Psi_0 = \Psi_{00} - \Psi_0$.

(c) Se $\|\delta_k\|_2 < \varepsilon_1$ e $|\Delta\Psi_0| < \varepsilon_1$ ou $m > \overset{\cap}{M}$, então

- $\overset{\cup}{\mathbf{x}} = \mathbf{x}_{k+1}$, $\Psi(\overset{\cup}{\mathbf{x}}) = \Psi(\mathbf{x}_{k+1})$ e volte ao algoritmo *minimax*;

Senão,

(d) Assuma $\Psi_{00} = \Psi_0$.

8. Preparando para a próxima iteração.

(a) Calcule \mathbf{g}_{k+1} e $\boldsymbol{\gamma}_k = \mathbf{g}_{k+1} - \mathbf{g}_k$.

(b) Calcule $D = \boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k$;

Se $D \leq 0$, então

- $\mathbf{S}_{k+1} = \mathbf{I}_n$;

Senão,

- Calcule \mathbf{S}_{k+1} através de (2.20).

(c) Assuma $k = k + 1$ e volte ao passo 2.

No passo 1 do algoritmo Prático de Quasi-Newton ocorre a inicialização de vários parâmetros e constantes para o desenvolvimento do algoritmo. O valor de ε_1 foi adotado como 10^{-3} , este parâmetro está diretamente relacionado ao critério de parada do algoritmo Quasi-Newton e com o grau de minimização que se pretende obter. O parâmetro m contabiliza o número de operações relacionadas às funções Ψ_0 e Ψ'_0 , este é incrementado a cada vez que se calcula o valor dessas funções nos passos 1(e) da etapa de inicialização e passos 3 e 5 na execução do algoritmo, o valor máximo de operações permitida é determinado por \hat{M} . Para evitar que o algoritmo realize infinitas operações sem surtir resultado, o programa termina quando m se torna maior que \hat{M} . Cada algoritmo contabiliza o número de iterações separadamente dos outros, o parâmetro k é responsável por esta função em todos os algoritmos.

O algoritmo foi implementado exatamente com os mesmo parâmetros que a versão proposta por Andreas Antoniou em [Ant93]. As variáveis ρ e σ determinam o grau de precisão da técnica *line search*, como descrito na Seção 2.6. Os parâmetros χ e τ são utilizados na correção de possíveis discrepâncias no valor de $\hat{\alpha}_0$ obtido após os processos de interpolação e extrapolação, realizados nos passos 4 e 6, respectivamente. Estes testes são necessários para se evitar que a técnica *line search* obtenha um valor para o passo de adaptação que não promova a minimização da função objetivo do sistema. Nos passos 4(c) e 4(d), o parâmetro τ é utilizado para garantir que $\hat{\alpha}_0$ não seja menor que o mínimo, α_L , e nem maior que o máximo valor, α_U , de um intervalo onde possivelmente se situa o valor ótimo de α . O processo de extrapolação quadrática realizada no passo 6, possivelmente, retorna valores negativos que indicam um ponto de máximo na função objetivo, por isso a necessidade de realizar o teste do passo 6(b) no algoritmo. Para se evitar que o processo de extrapolação resulte em um valor de $\hat{\alpha}_0$ muito elevado, a implementação da rotina presente em 6(c) se torna necessário.

A estimativa realizada no passo 2(d) para determinar um valor inicial de α_0 ,

utiliza-se da redução de $\Psi(\mathbf{x})$ quando α varia de uma iteração para a outra, esta variação é representada por $\Delta\Psi_0$. Esta estimativa pode levar a valores inexatos e até mesmo negativos, por isso há a necessidade de realizar testes para encontrar um valor inicial de α_0 que não comprometa o processo.

O critério de parada do algoritmo leva em consideração a norma L_2 do vetor $\boldsymbol{\delta}_k$, dada por

$$\|\boldsymbol{\delta}_k\|_2 = \left[\sum_{k=1}^K |\boldsymbol{\delta}_k|^2 \right]^{1/2} \quad (3.1)$$

e a variação no valor da função objetivo entre iterações sucessivas, $\Delta\Psi(\mathbf{x})$. Caso o valor de $\Psi(\mathbf{x})$ esteja tendo uma variação menor que o valor de ε_1 de uma iteração para outra, o algoritmo é finalizado pois não está conseguindo minimizar o erro do sistema ou esse erro já atingiu seu valor mínimo para esta seção.

Caso os valores de $\Delta\Psi(\mathbf{x})$ e $\|\boldsymbol{\delta}_k\|_2$ não satisfaçam as condições estabelecidas pelo critério de parada, deve-se atualizar alguns parâmetros para a próxima iteração. Neste processo, uma das tarefas que merece maior atenção é a atualização da matriz \mathbf{S} , essa matriz juntamente com o vetor gradiente, \mathbf{g} , irão determinar a direção de convergência para a minimização da função objetivo. Foi dito na Seção 2.5 que a fórmula BFGS garante que a cada iteração se tenha uma matriz positiva definida caso o valor anterior também o seja, mas em raríssimas exceções isto pode não acontecer. Para evitar que se tenha uma matriz não-positiva definida, antes da atualização de \mathbf{S} através de (2.20), no passo 8(b) é feito um teste para garantir que \mathbf{S} atenda as características exigidas. Sabendo que $D = \boldsymbol{\delta}_k^T \boldsymbol{\gamma}_k$, $\boldsymbol{\delta}_k = \alpha_k \mathbf{d}_k$ e a partir de (2.21) obtém-se:

$$\begin{aligned} D &= \boldsymbol{\delta}_k^T \mathbf{g}_{k+1} - \boldsymbol{\delta}_k^T \mathbf{g}_k \\ &= \alpha_k (\mathbf{g}_{k+1}^T \mathbf{d}_k - \mathbf{g}_k^T \mathbf{d}_k) \end{aligned}$$

aplicando esta relação a (2.28) chega-se:

$$\begin{aligned} D &\geq \alpha_k(\sigma \mathbf{g}_k^T \mathbf{d}_k - \mathbf{g}_k^T \mathbf{d}_k) \\ &\geq \alpha_k(\sigma - 1) \mathbf{g}_k^T \mathbf{d}_k \end{aligned}$$

Caso \mathbf{d}_k indique a direção de minimização, tem-se $\mathbf{g}_k^T \mathbf{d}_k < 0$ e, além disso, sabe-se que $\alpha_k > 0$ e como $0 < \sigma < 1$, conclui-se que $D > 0$. Se esta condição for atendida, garante-se que \mathbf{S}_{k+1} será positiva definida aplicado-se a fórmula BFGS.

Após o processo minimização da função objetivo e a obtenção de um resultado estável para o equalizador com apenas 1 seção, deve-se incrementar o número de seções do filtro, $M = 2$, e iniciar todo o processo de minimização novamente aplicando os mesmos algoritmos. Considere que o resultado obtido com a minimização da função objetivo, para o filtro com apenas uma seção, seja \bar{c}_{01} e \bar{c}_{11} . Estes valores serão utilizados na inicialização da primeira seção e na obtenção dos coeficientes para a inicialização da segunda seção do filtro, formando assim o novo vetor $\mathbf{x}_0 = [\bar{c}_{01} \quad \bar{c}_{11} \quad c_{02} \quad c_{12}]^T$, que será utilizado como parâmetro de entrada para os algoritmos de minimização. Os valores de c_{02} e c_{12} utilizados para a inicialização dos algoritmos *minimax* e Quasi-Newton Prático são obtidos a partir de cada um dos pontos da Tabela 3.2.

Tabela 3.2: Pontos (c_{02}, c_{12}) para inicialização da segunda seção do equalizador.

No.	Ponto
1	$([1 - \varepsilon_3] \bar{c}_{01}; [1 - \varepsilon_3] \bar{c}_{11})$
2	$([1 + \varepsilon_3] \bar{c}_{01}; [1 - \varepsilon_3] \bar{c}_{11})$
3	$([1 + \varepsilon_3] \bar{c}_{01}; [1 + \varepsilon_3] \bar{c}_{11})$
4	$([1 - \varepsilon_3] \bar{c}_{01}; [1 + \varepsilon_3] \bar{c}_{11})$

Estes valores são carregados sucessivamente até que se obtenha um resultado estável após os processos de minimização. O parâmetro ε_3 é uma pequena constante utilizada para garantir que os pontos c_{02} e c_{12} estejam próximos de \bar{c}_{01} e \bar{c}_{11} , tentando assim forçar com que os coeficientes se situem dentro da região de

estabilidade do filtro *allpass*. Essa constante foi inicializada com o valor de 10^{-1} , caso se utilize todas as quatro alternativas e mesmo assim não se obtenha um resultado estável, deve-se alterar o valor de ε_3 e repetir todo o processo até que um resultado estável seja encontrado. Esta alteração no valor de ε_3 foi implementado dobrando-se o valor da constante a cada processo de minimização mal sucedido.

Diferentemente do algoritmo proposto por Andreas Antoniou, neste trabalho, apenas a obtenção de um filtro estável já é motivo para se incrementar o número de seções, não houve a preocupação de se analisar a qualidade do equalizador com apenas duas seções. A qualidade do filtro só começa ser verificada pelo parâmetro Q , a partir da terceira seção.

De posse dos coeficientes que garantam um equalizador estável com duas seções, os próximos passos do algoritmo que implementa as restrições são obter os coeficientes que inicializarão as próximas seções do filtro. Incrementando o número de seções do filtro e utilizando os coeficientes obtidos nas mais recentes e bem sucedidas etapas de minimização para a inicialização das seções $1, 2, \dots, M-1$ e o ponto

$$P_{0M} = \left[\frac{1}{2}(\hat{c}_{0(M-1)} + \check{c}_{0(M-1)}), \frac{1}{2}(\hat{c}_{1(M-1)} + \check{c}_{1(M-1)}) \right] \quad (3.2)$$

para a inicialização da M -ésima seção, onde $\hat{c}_{0(M-1)}\check{c}_{0(M-1)}$ e $\hat{c}_{1(M-1)}\check{c}_{1(M-1)}$ são o maior e o menor coeficiente, respectivamente, obtidos na última etapa de minimização bem sucedida. Alternativamente, caso o ponto dado por (3.2) não retorne filtros estáveis, os pontos

$$P_{1M} = (\check{c}_{0(M-1)}, \check{c}_{1(M-1)}) \quad P_{2M} = (\hat{c}_{0(M-1)}, \hat{c}_{1(M-1)})$$

$$P_{3M} = (\hat{c}_{0(M-1)}, \hat{c}_{1(M-1)}) \quad P_{4M} = (\check{c}_{0(M-1)}, \check{c}_{1(M-1)})$$

podem ser utilizados para a inicialização da M -ésima seção. Após obter os valores que inicializarão as seções $1, 2, \dots, M-1, M$ do equalizador, deve-se novamente

minimizar o erro do sistema através dos algoritmos *minimax* e Quasi-Newton Prático. Lembrando novamente que para o equalizador com mais de 2 seções o valor de ε no passo 1 do algoritmo *minimax* foi considerado igual a 10^{-2} . O número de seções do filtro é incrementado a cada vez que se obtém um resultado estável nas etapas de minimização, até que se obtenha um equalizador que atenda os requisitos de qualidade do projeto. Para finalizar o processo deve-se checar a cada etapa de minimização bem sucedida algum parâmetro que indique a qualidade do filtro, este parâmetro deve analisar a variação de τ_{ap} sobre a banda de frequência de interesse. Uma fórmula que pode ser utilizada para qualificar o equalizador de atraso de grupo é:

$$Q = \frac{100[\overset{\cap}{e}(\mathbf{x}) - \overset{\cup}{e}(\mathbf{x})]}{[\overset{\cap}{e}(\mathbf{x}) + \overset{\cup}{e}(\mathbf{x})]} \quad (3.3)$$

onde

$$\overset{\cap}{e}(\mathbf{x}) = \max_{1 \leq i \leq K} e_i(\mathbf{x})$$

$$\overset{\cup}{e}(\mathbf{x}) = \min_{1 \leq i \leq K} e_i(\mathbf{x})$$

O algoritmo é finalizado quando se obtém um valor de $Q < Q_{\max}$, para este projeto foi utilizado $Q_{\max} = 2$. Este valor representa uma relação de 2% entre a variação relativa do erro sobre a média desse erro, ou uma relação de 4% entre a variação absoluta do erro pela média. O valor de Q_{\max} , bem como a maioria dos parâmetros utilizados no algoritmo, foram obtidos empiricamente, ou seja, a partir de uma análise minuciosa de todos os passos do algoritmo implementado. Para cada projeto deve-se especificar o valor destes parâmetros atendendo às exigências de qualidade do equalizador a ser projetado.

3.4 Diagrama de Blocos

Nesta seção será apresentado um diagrama de blocos mostrando de uma outra maneira como os algoritmos que implementam os referidos métodos e técnicas se relacionam para formar o algoritmo final. O diagrama de blocos, dado pela Figura 3.2, mostra de uma maneira resumida as etapas do algoritmo utilizado para projetar equalizadores de atraso de grupo.

3.5 Conclusão

O algoritmo para obtenção de equalizadores, com uma mascarada específica de atraso de grupo, foi obtido com a combinação de quatro algoritmos. Foi mostrado primeiramente a implementação de um algoritmo que aplica as definições previstas pela técnica *minimax*, onde se pretende obter um resultado sem picos de erro na resposta. A cada iteração da técnica *minimax* o algoritmo Prático de Quasi-Newton, proposto e apresentado por Andreas Antoniou em [Ant93], realiza a aproximação da curva de atraso de grupo do filtro com a curva de referência proposta no projeto. Este algoritmo de minimização aplica juntamente com as definições do método Quasi-Newton as implicações da técnica *line search*, sem a qual não se consegue a convergência do algoritmo. A técnica *line search* foi aplicada por Andreas Antoniou a partir de uma ligeira modificação no algoritmo *line search* inexato proposto por Fletcher em [Fle87]. Este algoritmo implementa a técnica *line search* com uma certa imprecisão na obtenção do passo de adaptação ótimo. Gerenciando esses métodos tem-se um algoritmo que aplica restrições para garantir que se tenha resultados estáveis, ele também é responsável pela inicialização das seções do equalizador, tentando assim garantir que os coeficientes das M seções se concentrem dentro da região de estabilidade do filtro. O algoritmo que implementa as restrições foi baseado em uma técnica de aproximação forçada, proposto por Andreas Antoniou em [Ant93].

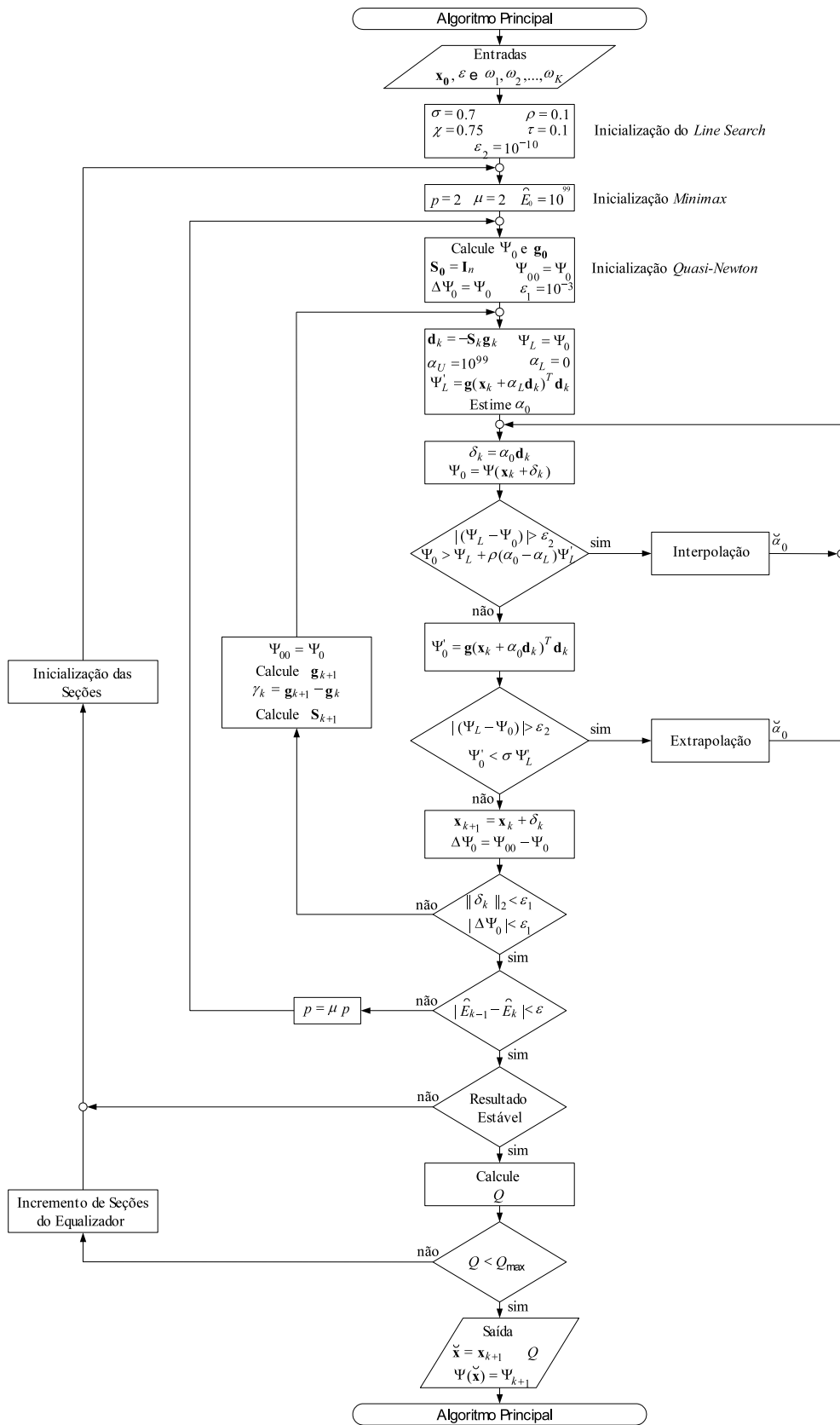


Figura 3.2: Diagrama de Blocos do Algoritmo Final.

Após a apresentação das definições teóricas de todas as técnicas e métodos utilizados na implementação do algoritmo, vista no Capítulo 2, o atual Capítulo apresentou considerações práticas sobre implementação e relacionamento das referidas técnicas e métodos. Foi apresentado todos os passos necessários à implementação dos quatro algoritmos, esses responsáveis pela aplicação dos conceitos relacionados às técnicas e métodos detalhados no Capítulo 2.

Capítulo 4

Resultados

4.1 Introdução

Toda a teoria necessária para o bom entendimento desta dissertação foi abordada nos Capítulos 1 e 2, bem como os passos para implementação do algoritmo, visto no Capítulo 3. Neste Capítulo, os resultados obtidos com as simulações realizadas em MATLAB serão apresentados e discutidos.

4.2 Resultados

O objetivo inicial deste trabalho é a obtenção de um equalizador de atraso de grupo com uma específica resposta, utilizado para realizar uma pré-correção no sinal de TV. A partir desta necessidade, surgiu a idéia de estudar métodos e técnicas de otimização para implementar um algoritmo, que de posse de uma máscara de atraso de grupo qualquer, pudesse obter os coeficientes de um filtro estável, para efetuar esta pré-correção.

O algoritmo foi testado utilizando-se várias curvas de atraso de grupo, com o objetivo de analisar sua eficiência frente a curvas com as mais diversas características. Os resultados que serão apresentados neste Capítulo mostram as respostas de atraso de grupo e o diagrama de pólos e zeros obtidos, à medida que se incrementa

o número de seções do equalizador. Também será apresentados gráficos e tabelas mostrando a progressão do algoritmo, indicando os coeficientes obtidos e o valor do parâmetro Q a cada seção incrementada. Finalizando a etapa de resultados, é apresentada uma tabela indicando a duração das simulações realizadas nesta dissertação. Não há o interesse em analisar as respostas de magnitude dos filtros, pois se tratam de filtros *allpass* onde se tem resposta de magnitude plana.

4.2.1 Simulação A

O primeiro resultado a ser apresentado é o equalizador obtido quando se inicializou o algoritmo com a máscara de atraso de grupo dada pela Resolução 284 da ANATEL. A Figura 4.1 apresenta a máscara de atraso de grupo citada acima.

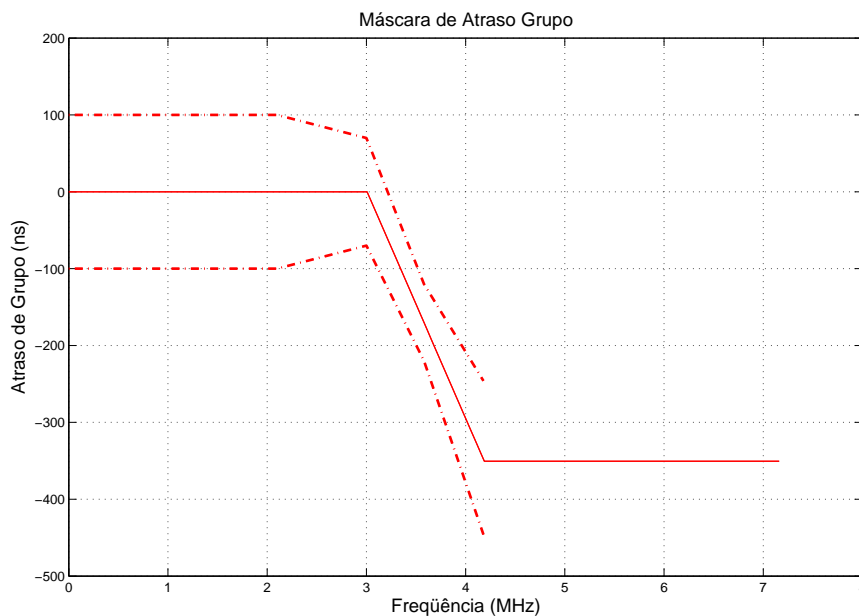


Figura 4.1: *Máscara de Atraso de Grupo.*

O algoritmo deve obter os coeficientes de um filtro estável com uma resposta de atraso de grupo semelhante à especificada pela norma. A resposta do equalizador deve se situar dentro dos limites representados pelas duas linhas tracejadas.

Iniciou-se o equalizador com apenas uma seção e o algoritmo com as amostras da resposta de atraso de grupo de referência, nas frequências $\omega_1, \omega_2, \omega_3, \dots, \omega_K$, e

com os valores iniciais de \mathbf{x}_0 , extraídos da Tabela 3.1. Em todas as simulações apresentadas neste Capítulo, utilizou-se 201 amostras, espaçadas uniformemente, para representar a resposta de atraso de grupo de referência e uma frequência de amostragem de $F_s = 14.32 \text{ MHz}$. Obtém-se, após a execução do algoritmo, a resposta de atraso de grupo do filtro. A Figura 4.2 apresenta a curva obtida para o equalizador com apenas uma seção, também é apresentado o diagrama de pólos e zeros da Função de Sistema do equalizador, mostrando assim que se trata de um filtro estável.

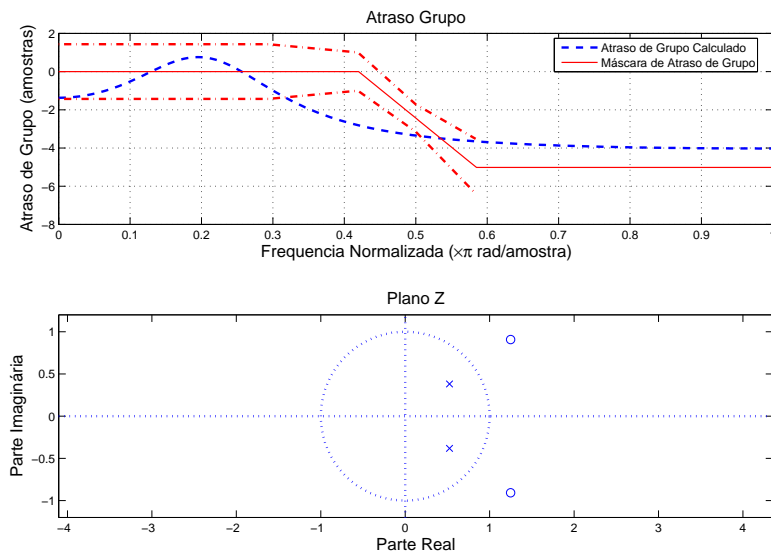


Figura 4.2: *Equalizador com uma seção.*

Os coeficientes obtidos nesta primeira etapa, $\bar{c}_{01} = 0,419862$ e $\bar{c}_{11} = -1,048494$, são a base para a inicialização da segunda seção do equalizador, a partir dos valores extraídos da Tabela 3.2. A resposta de atraso de grupo obtida para o equalizador com duas seções é apresentada pela Figura 4.3. Nota-se que a resposta do equalizador já atende os limites da norma, mas ainda pode ser melhorada com o incremento do número de seções, sem trazer complicações para o processo de implementação do filtro. O número de seções será incrementado até se atingir um nível mais baixo de ondulação na banda de interesse, analisado pelo parâmetro Q . É evidente que à medida que se incrementa o número de seções, as duas curvas

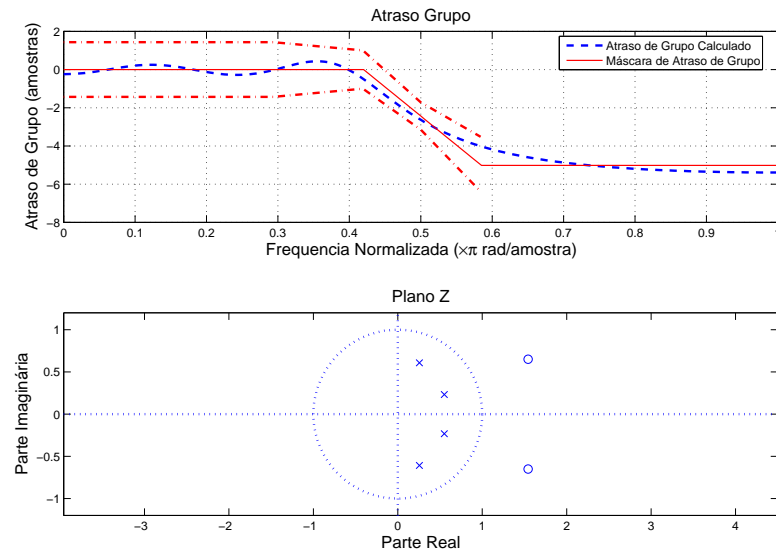


Figura 4.3: *Equalizador com duas seções.*

se tornam mais semelhantes, com isso tem-se um equalizador de melhor qualidade. O processo termina quando a resposta de atraso de grupo do equalizador é considerada próxima o suficiente da máscara de referência e quando a relação entre o máximo erro e o seu valor médio, medido pelo parâmetro Q , ficar abaixo do valor máximo, definido por Q_{max} .

A Figura 4.4 apresenta a resposta de atraso de grupo obtida quando se tem um filtro com quatro seções, percebe-se visualmente que a resposta de atraso de grupo do equalizador já é bem próxima da máscara desejada. Mesmo com essa semelhança, o parâmetro Q ainda é maior que o valor máximo estabelecido, por isso, esta resposta ainda não é suficiente para finalizar o processo.

Analisando fatores referentes à implementação prática de filtros digitais, pode-se dizer que filtros com até dez seções, vinte e um coeficientes, são considerados filtros curtos, portanto pode-se refinar ainda mais o resultado sem comprometer a etapa de implementação. O valor de $Q_{max} = 2$ foi estipulado para se ter um compromisso entre fidelidade do equalizador e quantidade de coeficientes do filtro. Com o incremento do número de seções nota-se facilmente pelos resultados obtidos que o qualidade do equalizador também aumenta. Como dito anterior-

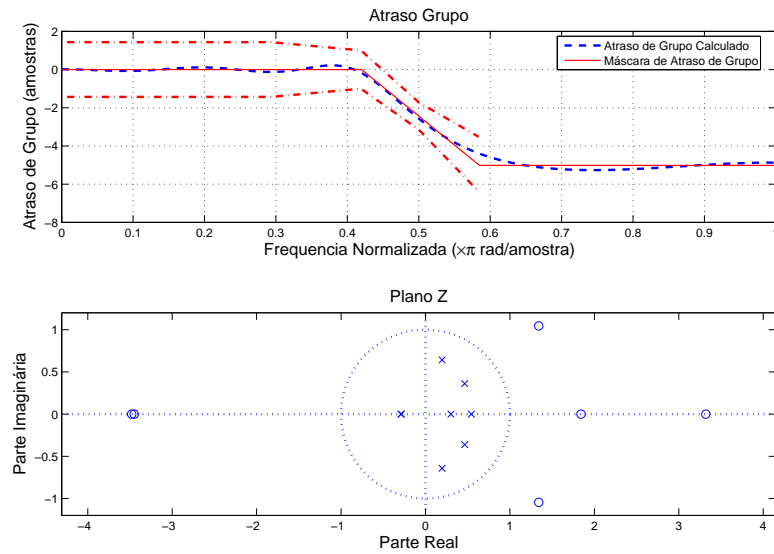


Figura 4.4: *Equalizador com quatro seções.*

mente, o algoritmo é finalizado quando se obtém equalizadores que atendam as condições mínimas de qualidade propostas pelo projeto. Para esta aplicação e para esta máscara de atraso de grupo em específico, conseguiu-se atender todos os critérios estabelecidos anteriormente com um filtro de seis seções, este resultado é apresentado através da resposta de atraso de grupo, diagrama de pólos e zeros e também da resposta impulsiva do filtro, dado pelas Figuras 4.5, 4.6, 4.7 respectivamente.

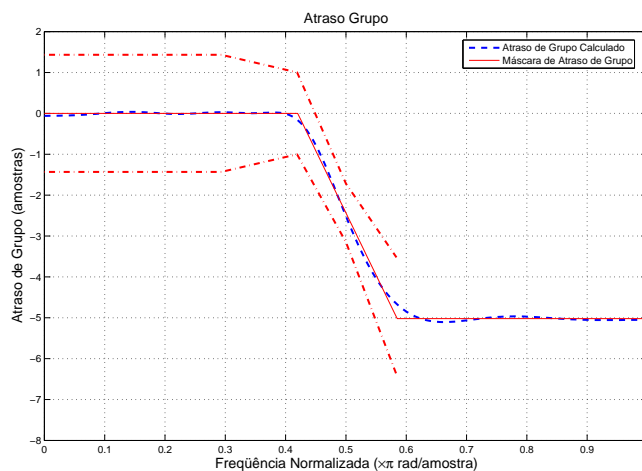


Figura 4.5: *Equalizador com seis seções(Resposta de Atraso de Grupo).*

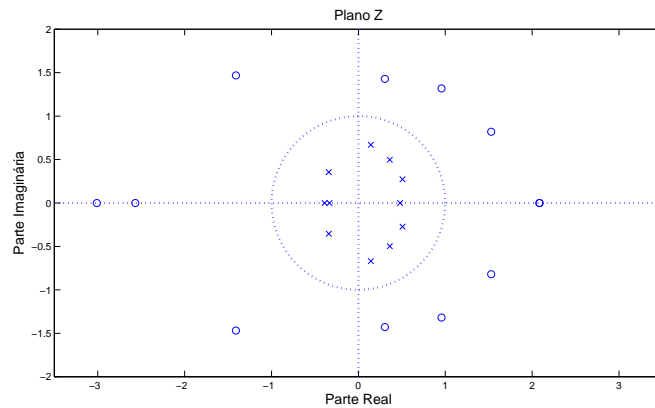


Figura 4.6: Equalizador com seis seções (Plano Z).

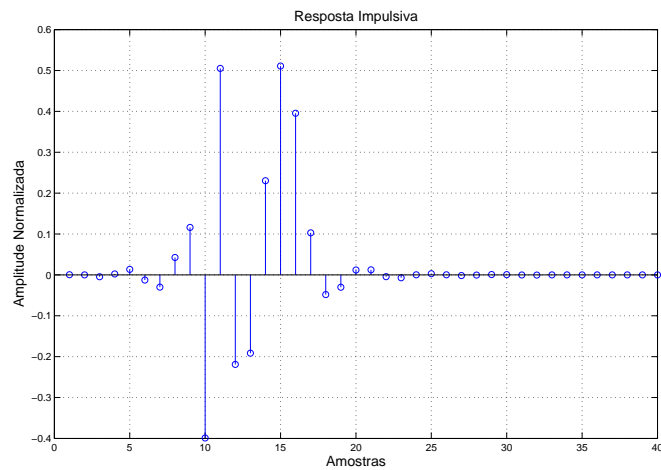


Figura 4.7: Equalizador com seis seções (Resposta Impulsiva).

Visualmente, é possível notar que o atraso de grupo conseguido com o equalizador de seis seções é muito próximo do atraso de grupo da máscara apresentada pela Figura 4.1 e o valor de Q comprova este resultado. Além das Figuras 4.7 e 4.6, uma outra maneira de comprovar a estabilidade do equalizador obtido é analisar a disposição dos coeficientes c_{0l} e c_{1l} , obtidos em cada seção, sobre a região de estabilidade dada pelas condições (1.25) e pela Figura 1.5.

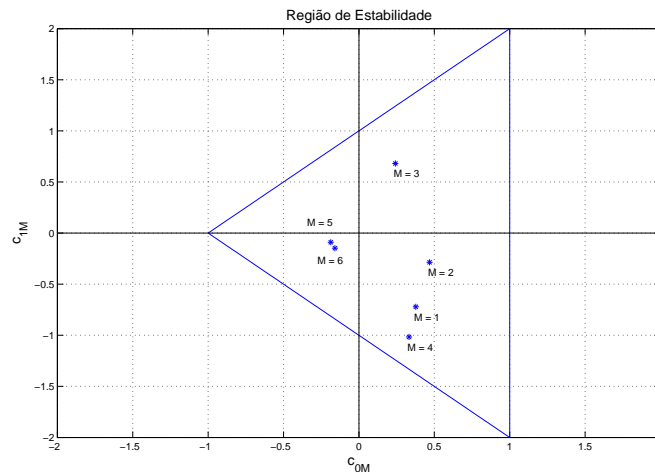


Figura 4.8: Coeficientes sobre a Região de Estabilidade.

O progresso do algoritmo mostra que o equalizador com somente duas seções, ou um filtro IIR de quarta ordem, já é suficiente para se obter um atraso de grupo que satisfaz a norma, necessária para “pré-corriger” o erro de fase em receptores. Por questões de implementação, foi escolhido o equalizador com seis seções, pois um filtro com treze coeficientes ainda é considerado um filtro curto, além de apresentar um nível de ondulação bem mais baixo se comparado com o filtro de duas seções.

Os coeficientes obtidos a cada incremento no número de seções é apresentado pela Tabela 4.1, bem como o valor do parâmetro Q e o máximo valor de p que atendeu os critérios do algoritmo *minimax*. Além da curva de referência dada pela Figura 4.1, várias outras curvas foram utilizadas como referência, τ_0 , para analisar o desempenho do algoritmo frente as mais diversas características de respostas de atraso de grupo. Algumas curvas utilizadas possuem características que, teoricamente, causariam dificuldades aos processos de otimização, como por exemplo curvas que apresentam descontinuidades. Nas próximas seções serão apresentados os resultados obtidos com a inicialização do algoritmo com algumas dessas curvas de referência.

Tabela 4.1: *Resultados Simulação A.*

Seções	l	c_{0l}	c_{1l}	Q	p_{max}
0	0	—	—	100	—
1	1	0,419862	-1,048494	55,40	6
2	1	0,434410	-0,514658	11,19	6
	2	-0,514658	0,357457		
3	1	0,458554	-0,416420	5,04	8
	2	0,374199	-0,990318		
	3	-0,233092	-0,193503		
4	1	0,452510	-0,394478	4,00	8
	2	0,352339	-0,938476		
	3	-0,080025	0,017775		
	4	-0,157991	-0,272546		
5	1	0,471971	-0,301871	2,12	10
	2	0,388837	-0,772746		
	3	0,236865	0,708312		
	4	0,359169	-1,087294		
	5	-0,252588	-0,126608		
6	1	0,469045	-0,282025	1,74	10
	2	0,379484	-0,712388		
	3	0,246558	0,678899		
	4	0,341462	-1,018144		
	5	-0,183754	-0,128670		
	6	-0,188067	-0,118155		

4.2.2 Simulação B

O segundo resultado a ser apresentado é a resposta de atraso de grupo obtida inicializando o algoritmo com uma máscara ainda suave, ou seja, sem descontinuidades. Com o objetivo de analisar o comportamento do algoritmo frente a uma referência com praticamente as mesmas características da curva da Figura 4.1. A curva de referência e a resposta do algoritmo para o equalizador com duas seções é apresentado pela Figura 4.9.

Nota-se que essa curva de referência também não possui descontinuidades. De posse disto, espera-se obter um resultado próximo ao obtido na seção anterior, no que diz respeito a número de seções necessárias para se atingir um valor de $Q < 2$. A partir do resultado encontrado para o filtro estável com duas seções, pode-se incrementar o número de seções do equalizador e dar continuidade ao processo de otimização. Incrementando o número de seções do filtro até chegar em um resultado que atenda as mesmas exigências de qualidade do modelo anterior.

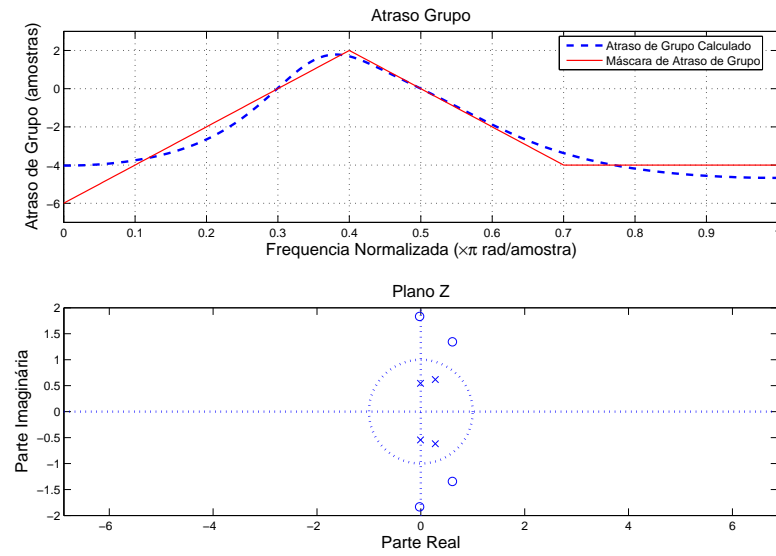


Figura 4.9: *Equalizador com duas seções.*

Conseguiu-se obter um resultado que atenda as exigências do projeto com apenas sete seções, o resultado final pode ser comprovado pela Figura 4.10.

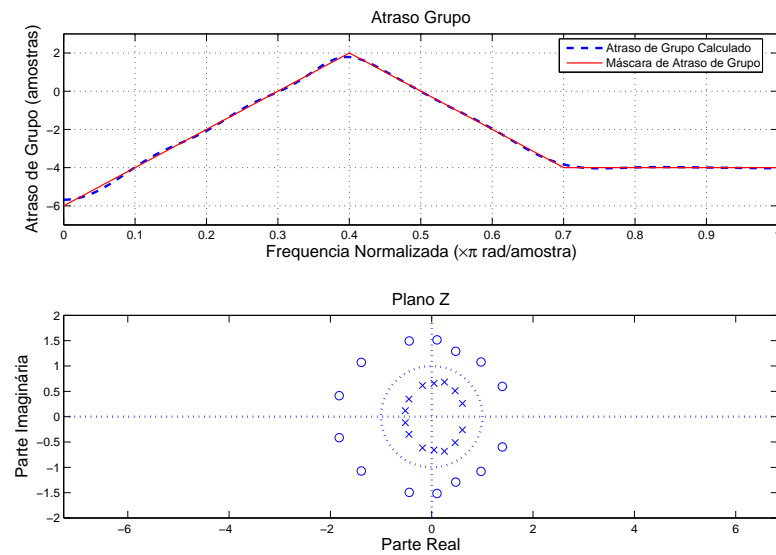


Figura 4.10: *Equalizador com sete seções.*

Analisando o resultado obtido, confirma-se a previsão de se obter um filtro com aproximadamente o mesmo número de seções da Simulação A. A disposição dos coeficientes na região de estabilidade pode ser visto pela Figura 4.11.

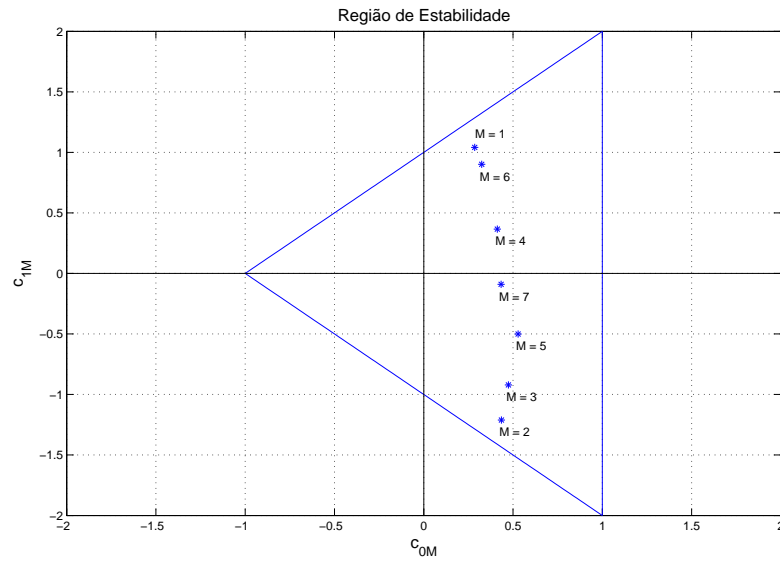


Figura 4.11: Coeficientes sobre a Região de Estabilidade.

A Tabela 4.2 apresenta os coeficientes, os valores do parâmetro Q e o valor de máximo de p obtidos durante a execução do algoritmo.

Tabela 4.2: Resultados Simulação B.

Seções	l	c_{0l}	c_{1l}	Q	p_{max}
0	0	—	—	100	—
1	1	0,551983	-0,376732	41,78	4
2	1	0,458875	-0,558641	19,01	4
	2	-0,558641	0,297854		
4	1	0,419030	-0,969193	5,29	12
	2	0,237233	0,885677		
	3	0,518524	-0,470227		
	4	0,402353	0,159639		
5	1	0,399939	-1,021471	3,61	14
	2	0,307909	1,029404		
	3	0,452206	-0,508895		
	4	0,322742	0,284099		
	5	0,347026	-0,196904		
7	1	0,435324	-1,212497	1,60	20
	2	0,288345	1,046282		
	3	0,528415	-0,500842		
	4	0,324171	0,899772		
	5	0,474301	-0,922292		
	6	0,412495	0,365451		
	7	0,433641	-0,090774		

4.2.3 Simulação C

O próximo resultado a ser apresentado é a resposta de atraso de grupo obtida quando utilizou-se uma curva de referência com descontinuidades. Esse tipo de resposta foi utilizada como um caso hipotético, apenas para analisar o desempenho do algoritmo frente a sinais com essas características. A princípio, esperase que o algoritmo tenha um maior esforço para se chegar ao resultado final, que atenda as especificações de qualidade propostas anteriormente. A curva de referência que se deseja obter é dada pela Figura 4.12.

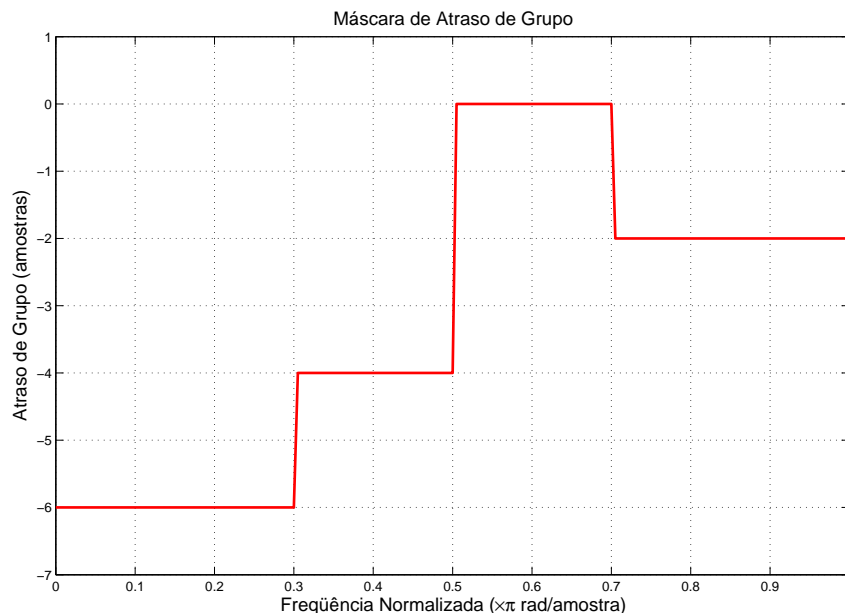


Figura 4.12: *Máscara de atraso de grupo com descontinuidades.*

Já nos primeiros passos do algoritmo percebe-se que não se tratava de um caso simples de minimização. Durante a execução do programa percebeu-se uma demora considerável na busca por coeficientes estáveis e também não se obtinha um grande avanço na qualidade do equalizador quando se incrementava o número de seções. O resultado para o equalizador com apenas uma seção é apresentado pela Figura 4.13. Ao se incrementar o número de seções para dois, obteve-se um resultado bastante semelhante ao do filtro com uma seção. Nota-se que a resposta

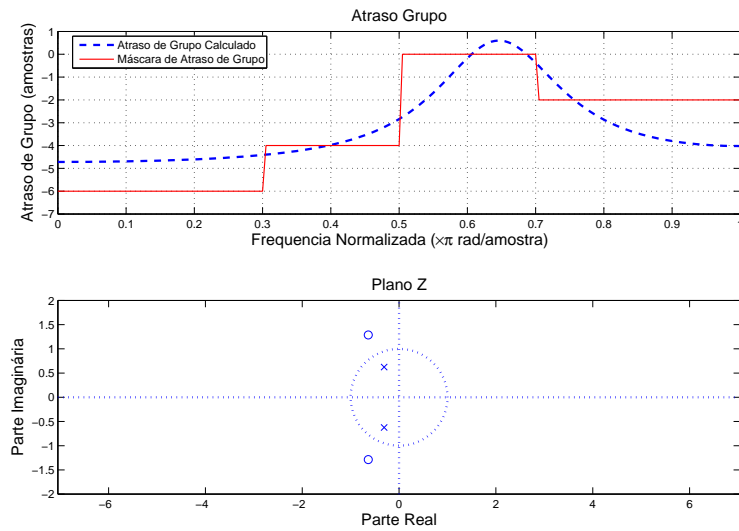


Figura 4.13: *Equalizador com uma seção.*

obtida com o filtro consegue apenas se aproximar da curva de referência fora das descontinuidades, o que pode ser comprovado pela Figura 4.14

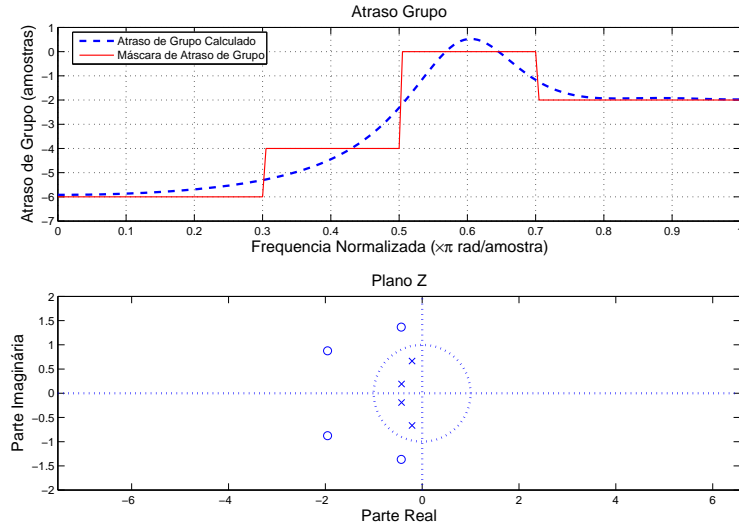


Figura 4.14: *Equalizador com duas seções.*

Os resultados obtidos apenas comprovam o que já era esperado, é uma missão um tanto quanto difícil querer que um filtro com apenas duas seções, Função de Sistema de quarta ordem, represente de maneira aproximada uma função com tantas descontinuidades. Analisando o progresso do algoritmo nota-se que a

partir da décima seção, a resposta obtida com o filtro começa a tomar a forma da curva de referência, o que pode ser observado na Figura 4.15.

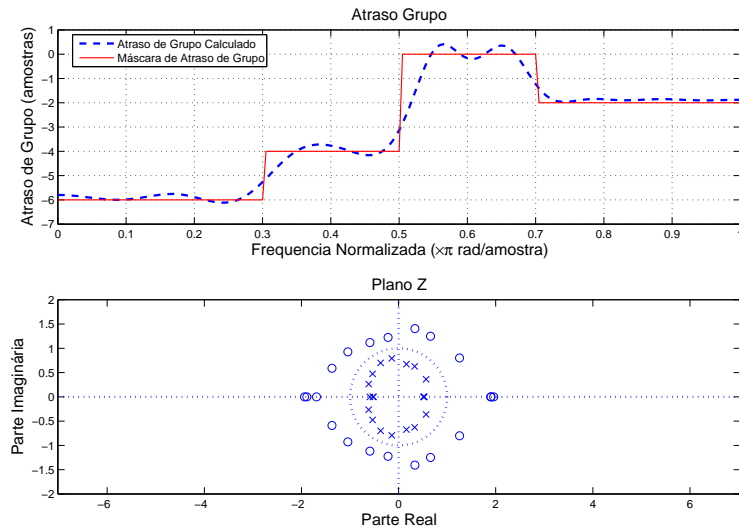


Figura 4.15: *Equalizador com dez seções.*

Após a décima seção o algoritmo ajusta a resposta do filtro às discontinuidades da curva proposta, conforme Figura 4.16.

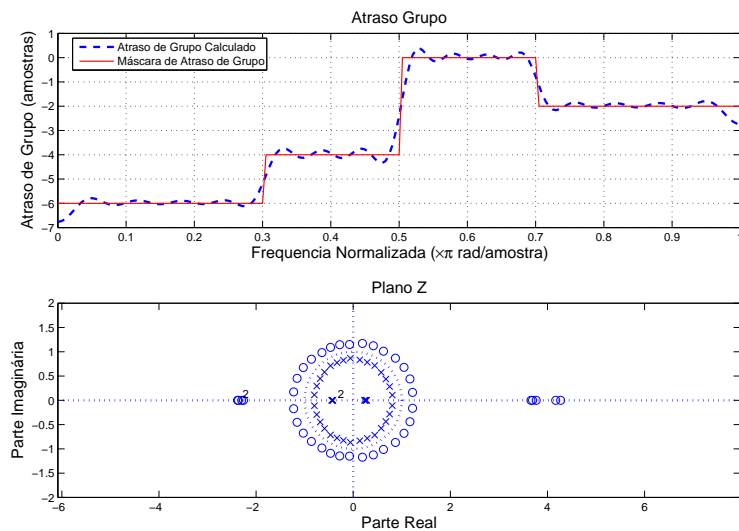


Figura 4.16: *Equalizador com vinte seções.*

A partir da vigésima seção, observa-se que o algoritmo tenta apenas otimizar o resultado, ou seja, melhorar cada vez mais a qualidade do equalizador. Obteve-se

um equalizador com trinta e cinco seções, que atende os requisitos de qualidade propostos anteriormente. A Figura 4.17 mostra o resultado final para a referida curva de referência.

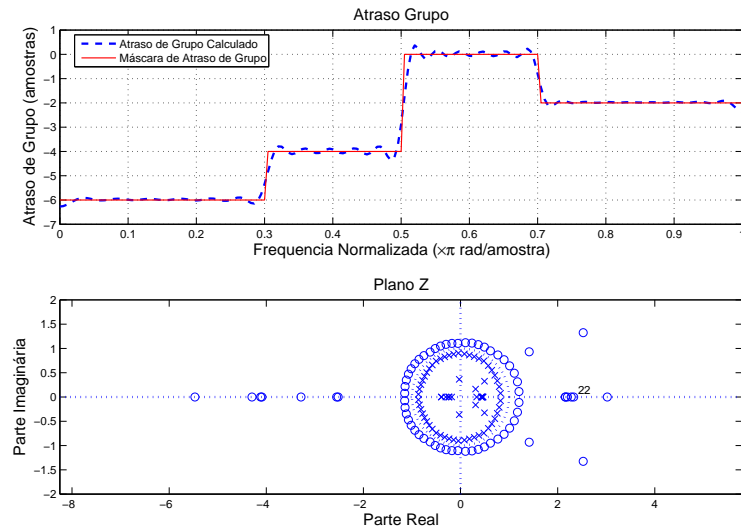


Figura 4.17: *Equalizador com trinta e cinco seções.*

A Figura 4.18 traz os coeficientes das trinta e cinco seções dispostos sobre a região de estabilidade, comprovando mais uma vez a estabilidade do equalizador.

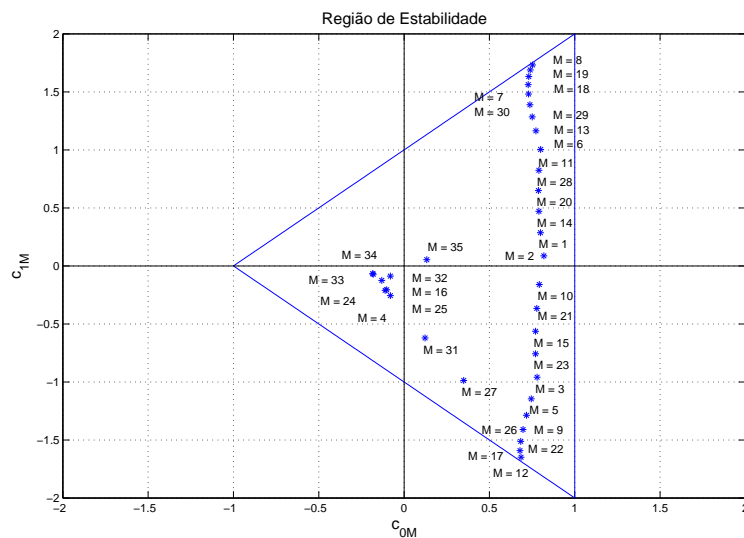


Figura 4.18: *Coefficientes sobre a Região de Estabilidade.*

A Tabela 4.3 apresenta os resultados obtidos em algumas etapas do algoritmo. Nota-se que a cada incremento no número de seções, não se consegue uma redução tão acentuada no valor do parâmetro Q , como nos casos anteriores.

Tabela 4.3: *Resultados Simulação C.*

Seções	l	c_{0l}	c_{1l}	Q	p_{max}
0	0	—	—	100	—
1	1	0,485094	0,616834	46,82	6
2	1	0,218220	0,852321	27,52	6
	2	0,852321	0,486951		
3	1	0,332509	1,068761	23,77	56
	2	0,519511	0,506557		
	3	0,018616	-0,281735		
35	1	0,818499	0,091069	1,96	12
	2	-0,268100	0,105327		
	3	0,779523	-0,950631		
	4	-0,179966	-0,219917		
	5	-0,275241	0,061764		
	6	0,737439	1,300063		
	7	0,737156	-1,129869		
	8	-0,275122	0,061592		
	9	0,606463	1,501798		
	10	0,764217	1,168716		
	11	-0,275452	0,066676		
	12	0,795566	-0,155026		
	13	-0,275662	0,063744		
	14	0,539282	-1,425930		
	15	0,791928	0,996629		
	16	-0,274596	0,061391		
	17	0,698814	-1,259346		
	18	0,778825	0,640760		
	19	0,781839	-0,359422		
	20	0,795480	0,285633		
	21	-0,274067	0,062164		
	22	0,783898	0,466100		
	23	0,775044	-0,556084		
	24	-0,275583	0,063187		
	25	-0,276316	0,059445		
	26	0,672362	1,487105		
	27	0,616922	-1,431596		
	28	-0,275366	0,060374		
	29	0,709855	1,409579		
	30	0,773406	-0,749865		
	31	-0,273528	0,059265		
	32	0,780407	0,813982		
	33	-0,242252	-0,056863		
	34	0,661898	-1,362670		
	35	-0,308044	0,050049		

4.3 Duração das Simulações

Nesta seção serão apresentados dados referentes às durações das simulações realizadas nesta dissertação. A inclusão destas informações foram julgadas importantes, caso o leitor queira comprovar algum resultado, ou até mesmo ter uma noção da complexidade e tempo de processamento dos algoritmos implementados. As simulações foram realizadas utilizando-se um Processador Pentium M 1,5 GHz, 1 MB de Memória Cache, 512 MB de Memória RAM, utilizando o programa MATLAB 7.0. A Tabela 4.4 traz os tempos gastos para obter equalizadores com resposta de atraso de grupo dado pelas três curvas de referência apresentadas neste Capítulo.

Tabela 4.4: *Duração das Simulações.*

SIMULAÇÃO	DURAÇÃO
Simulação A	5,30 (segundos)
Simulação B	10,60 (segundos)
Simulação C	15,90 (minutos)

Capítulo 5

Itens Conclusivos

5.1 Conclusões

O algoritmo Quasi-Newton prático se mostrou bem robusto à minimização de funções não-quadráticas, provenientes das funções objetivas obtidas pela norma L_p . Testes realizados com o algoritmo mostraram que sem a utilização da técnica *line search*, o método Quasi-Newton indica a direção de minimização da função objetivo, mas não consegue a convergência sem a obtenção de um passo de adaptação ótimo. Pequenas variações nos valores de cada coeficiente mudam drasticamente a resposta de atraso de grupo do equalizador.

Pôde-se notar através dos resultados que a implementação do algoritmo *minimax* também é essencial à obtenção de uma resposta de atraso de grupo fiel à máscara proposta. Com simulações adicionais, sem a utilização da técnica *minimax*, observou-se que para a obtenção de equalizadores com praticamente a mesma qualidade de resposta de atraso de grupo são necessários filtros com número maior de seções. Pode-se dizer que o algoritmo *minimax* é o responsável pelo refinamento do resultado obtido, ou seja, reduz a ondulação na resposta de atraso de grupo na faixa de interesse do equalizador.

A implementação e utilização de cada método, abordado nesta dissertação, teve sua necessidade comprovada após testes e simulações. Após implemen-

tado os algoritmos Quasi-Newton prático e *minimax*, uma tentativa natural era obter resultados satisfatórios apenas com a utilização destes métodos. Contudo, observou-se que o algoritmo consegue a minimização da função objetivo mas resultando em coeficientes que distam muito da região de estabilidade, assim proporcionando filtros instáveis. Deste teste houve a necessidade de utilizar alguma técnica que garantisse a estabilidade dos filtros, isso foi conseguido com a implementação de uma versão ligeiramente modificada de um algoritmo proposto por Andreas Antoniou em [Ant93]. O algoritmo implementado neste trabalho utiliza os mesmos métodos para identificar os pontos que iniciarão cada seção, mas aplica critérios diferentes tanto para incremento do número de seções bem como para o término do algoritmo, como pode ser visto no Capítulo 3. Implementadas estas restrições e após obter resultados satisfatórios nas duas primeiras seções, o algoritmo se mostrou bastante eficiente e rápido na obtenção de coeficientes estáveis para as próximas seções. Com o decorrer do algoritmo observou-se que os coeficientes tendem a se situar próximos aos valores previamente obtidos, com isso facilitando a obtenção de novos valores estáveis.

Pôde-se observar a partir das respostas de atraso de grupo obtidas, que o algoritmo consegue excelentes resultados de minimização tanto para curvas de referência com transições suaves como para sinais com altos índices de descontinuidades. Notou-se que para obter um equalizador com resposta de atraso de grupo que apresente descontinuidades é necessário um filtro com elevado número de seções, quando comparado a resultados para respostas suaves. Isso se deve, principalmente, ao fato desses sinais serem representados apenas por funções de ordem elevada, o que faz com que o processo de minimização seja mais árduo e demorado. Outra virtude, de necessário comentário, é a robustez do algoritmo responsável por restringir resultados instáveis, já que em todo o processo tem-se a necessidade de se obter coeficientes que minimizem o erro e que também pertençam à região de estabilidade, resultados estes obtidos com sucesso.

O algoritmo apresentou excelentes resultados na obtenção de equalizadores atraso de grupo. O método Quasi-Newton demonstrou ser eficiente na minimização de soluções não-quadráticas, uma matriz \mathbf{S}_{k+1} positiva definida mostrou ser obtida facilmente pela fórmula BFGS. Testes com o algoritmo principal, com diferentes dados de entrada, diferentes características de atraso de grupo mostraram bons resultados de minimização bem como a obtenção de filtros recursivos com poucos coeficientes, com isso comprovando a eficiência e robustez do algoritmo.

5.2 Principais Contribuições

No Capítulo 1 foi analisado uma teoria básica relacionada a filtros digitais e sistemas, necessária para determinar as características do projeto, no que diz respeito à escolha das técnicas e métodos necessários para chegar ao objetivo final. Também foi analisada toda a teoria necessária para implementar filtros *allpass* estáveis. Este 1º Capítulo foi de fundamental importância para analisar e interpretar os resultados obtidos.

Após definidos alguns conceitos básicos, o Capítulo 2 trouxe uma descrição detalhada das técnicas utilizadas na implementação do algoritmo. A partir deste Capítulo pôde-se analisar melhor todo o processo e entender as características e finalidades de cada método. Ainda foi apresentado todo o equacionamento necessário para implementar algoritmos que apliquem os conceitos das técnicas e métodos utilizados neste trabalho.

A principal contribuição do Capítulo 3 foi a implementação dos algoritmos para cada método e técnica apresentados nesta dissertação, estes algoritmos poderão ser utilizados separadamente em futuras investigações nas mais diversas áreas. Este Capítulo ainda traz a uma descrição detalhada de todas as linhas dos algoritmos, bem como a forma com que os métodos se relacionam para formar o algoritmo final, este responsável por obter equalizadores de atraso de grupo a

partir de uma máscara pré-definida.

Finalizando este trabalho, os Capítulos 4 e 5 trazem os resultados obtidos com o algoritmo e as conclusões finais à respeito destes resultados. Para uma melhor interpretação dos resultados, o Capítulo 4 traz os gráficos mostrando o progresso do algoritmo e também os coeficientes obtidos em cada etapa do processo de minimização.

5.3 Sugestão para Futuras Investigações

A partir do conhecimento adquirido com o estudo e implementação destes métodos e técnicas de otimização, pode-se analisar o desempenho dos algoritmos frente a novas aplicações nas mais diferentes áreas possíveis. Dentre elas pode-se citar:

- Estudar uma possível utilização dos algoritmos implementados em sistemas de equalização adaptativa.
- Analisar o emprego destas técnicas e métodos para outras finalidades, como por exemplo, na obtenção de circuitos de linearização.
- Fazer a mesma análise matemática para filtros com seções de 3^a ou 4^a ordem, para investigar agilidade de todo o processo.

5.4 Publicações

- BRITO, RODRIGO BARBOSA e FASOLO, SANDRO ADRIANO, *Robust and Efficient Algorithm for the Design of Stable Group Delay Equalizers using Constrained Optimization Methods*, 15th Annual MPRG/ Virginia Tech Symposium on Wireless Personal Communications. pp 83- 89, 8-10 de Junho de 2005, Blacksburg-VA, USA.

Referências Bibliográficas

- [Ber84] - Bernard Grob, “Basic Television and Video Systems”, 1984, Fifth Edition, MacGraw-Hill;
- [Fle87] - R. Fletcher, “Practical Methods of Optimization”, 1987, Second Edition, John Wiley & Sons.
- [Ko97] - Ko, N.; Shpak, D.; Antoniou, A.; “Design of Recursive Delay Equalizers Using Constrained Optimization”, IEEE Pacific Rim Conf., Comm., Comp. And Signal Processing, 1997, Pages: 173 - 177.
- [Lan98] - Lang, M.; “Allpass filter design and applications”, IEEE Transactions on Volume 46, Issue 9, 1998, Pages:2505 - 2514.
- [Mat04] - Matsunaga, T.; Ikehara, M.; “Group delay approximation of allpass digital filter by transforming the desired response”, The 2004 47th Midwest Symposium on Volume 2, 2004, Pages: II-45 - II-48.
- [Had01] - Haddad, K.C.; Yongyi Yang; Galatsanos, N .P.; Stark, H.; “Allpass Filter Design Using Projection-Based Method Under Group Delay Constraints”, IEEE International Conference on Volume 6, 2001, Pages: 3833 - 3836.
- [Ant93] - A. Antoniou, “Digital Filters: Analysis, Design, and Applications”, 1993, Second Edition, McGraw Hill, New York;

- [Opp99] - A. V. Oppenheim e R. W. Schaffer, “Discrete-Time Signal Processing”, 1999, Second Edition, Prentice Hall;
- [Jur64] - E. I. Jury, “Theory and Application of the z-Transform Method”, 1964, Wiley, New York;
- [Mar49] - M. Marden, “The Geometry of the Zeros of a Polynomial in a Complex Variable”, 1949, American Mathematical Society, New York;
- [Pro96] - J. G. Proakis e D. G. Manolakis, “Digital Signal Processing”, 1996, Third Edition, Prentice Hall;
- [Gol96] - Golub, GENE H. e Loan, Charles F. V., “Matrix computations”, 1996, Third Edition, London University Press;
- [Din97] - P. S. R. Diniz, “Adaptive Filtering: Algorithms and Practical Implementation”, 1997, Second Edition, Kluwer Academic Publishers.