EXTENSÕES PERCEPTUAIS DO ALGORITMO MMP APLICADO À CODIFICAÇÃO DE VOZ

Leonardo dos Anjos Chaves

Dissertação de Mestrado apresentada ao Programa de Pós-graduação em Engenharia Elétrica, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Mestre em Engenharia Elétrica.

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Rio de Janeiro
Junho de 2013

EXTENSÕES PERCEPTUAIS DO ALGORITMO MMP APLICADO À CODIFICAÇÃO DE VOZ

Leonardo dos Anjos Chaves

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

Prof. Eduardo Antônio Barros da Silva, Ph.D.

Prof. Sergio Lima Netto, Ph.D.

Prof. José Antonio Apolinário Junior, D.Sc.

Prof. Murilo Bresciani de Carvalho, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
JUNHO DE 2013

Chaves, Leonardo dos Anjos

Extensões perceptuais do algoritmo MMP aplicado à codificação de voz/Leonardo dos Anjos Chaves. – Rio de Janeiro: UFRJ/COPPE, 2013.

XVI, 99 p.: il.; 29, 7cm.

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Dissertação (mestrado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2013.

Referências Bibliográficas: p. 87 – 89.

1. Compressão. 2. MMP. 3. Codificação de Voz. 4. Recorrência de padrões. I. Silva, Eduardo Antônio Barros da *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

A Deus e aos meus pais,
Abelardo e Zuleica.

Agradecimentos

Agradeço a todos que contribuíram para este trabalho, em especial aos professores Eduardo e Sergio, fiéis orientadores que atuaram sempre com transparência, clareza e esbanjando a capacidade de ensinar. “Mestres” é a palavra que indiscutivelmente deve ser atribuída a essas pessoas.

Aos professores do Departamento de Eletrônica e Computação (DEL) da UFRJ e do Programa de Engenharia Elétrica da COPPE/UFRJ. Todos, sem exceção, têm relevante parcela no conhecimento que adquiri ao longo dos anos e são responsáveis pelo incentivo à progressão da minha vida acadêmica.

Aos companheiros do LPS, sempre dispostos a ajudar e participar de discussões esclarecedoras.

Aos amigos da Globo, agradeço pelas motivações, pelas referências, parcerias e, claro, pelo convívio diário, sempre irreverente e muito profissional.

A minha família, meus pais e irmã, pela compreensão em todos os momentos, os ausentes inclusive que foram importantes e essenciais para a dedicação e produção deste trabalho.

A Julia: sem a nossa parceria e entrega tudo seria mais difícil. Nosso relacionamento sempre foi meu porto seguro. Estar ao seu lado traz conforto e o desejo de superar os desafios.

Todos colaboraram muito para essa conclusão deste trabalho.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

EXTENSÕES PERCEPTUAIS DO ALGORITMO MMP APLICADO À CODIFICAÇÃO DE VOZ

Leonardo dos Anjos Chaves

Junho/2013

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Programa: Engenharia Elétrica

Este trabalho investiga o processo de codificação de sinais de voz a partir de técnicas de casamento de padrões recorrentes. O algoritmo MMP (*Multidimensional Multiscale Parser*) foi utilizado como base nessa dissertação por proporcionar uma codificação em blocos muito eficiente. O MMP, originalmente desenvolvido para compressão de imagens e vídeo, desponta como um algoritmo poderoso na compressão de sinais de diferentes naturezas, como eletrocardiograma, informações de textura e profundidade de imagens 3D, radares meteorológicos, entre outros. Sua característica universal reside no fato que nenhuma informação prévia do sinal é necessária para a codificação e sua eficiência está diretamente relacionada à velocidade de aprendizado dos padrões existentes no sinal de interesse.

Contribuições significativas foram feitas à estrutura básica do MMP, que incluem novos algoritmos de predição linear baseados no método dos mínimos quadrados, a introdução de características perceptuais no processo de codificação e novas análises de qualidade a partir de métricas objetivas que permitem avaliar o comportamento do algoritmo e indicam campos de pesquisa ainda pouco explorados. Este trabalho priorizou o ajuste no conjunto de parâmetros de codificação que produzem o melhor resultado perceptual para a taxa alvo de 8 kbps, permitindo assim, uma comparação direta com o algoritmo CELP, descrito no padrão ITU-T G.729. Os resultados experimentais foram obtidos a partir do banco de 40 frases descrito no Apêndice A.

Por último, um codificador e um decodificador de voz baseado em MMP foram inteiramente desenvolvidos de forma estruturada para apoiar futuras pesquisas e se destacam como outra contribuição relevante deste trabalho.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

PERCEPTUAL EXTENSIONS OF MMP ALGORITHM APPLIED TO SPEECH CODING

Leonardo dos Anjos Chaves

June/2013

Advisors: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Department: Electrical Engineering

This dissertation considers the topic of speech coding using recurrent pattern matching techniques. In that context, the performance of the multidimensional multiscale parser (MMP) algorithm as a block codec was evaluated. The MMP, originally developed for application on image and video compression, emerges as a powerful tool for compressing different source signals such as electrocardiogram, 3D images, radars, and many others. The MMP universal nature relies on the fact that it does not require any a priori information about the signal at hand and thus, its compression efficiency results from how fast the patterns are learned along the coding process.

Considerable contributions to the basic MMP framework are suggested here, including new variations of the linear prediction speech pre-processing, the introduction of perceptual characteristics on the coding process, and the establishment of new quality evaluation procedures based on objective metrics which lead to an underexplored novel field for research. The entire methodology considered the adjustment of several MMP parameters in order to optimize its perceptual performance when targeting an 8 kbps coding rate and hence, enabling a straight comparison to the ITU-T G.729 speech codec. All experimental results were based on a 40-signal speech database, as detailed in Appendix A.

Finally, a complete and optimized MMP-based speech codec was developed in a structured framework, standing as another important contribution of the present work to the current knowledge on speech coding research area.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

"Watson, if I can get a mechanism which will make a current of electricity vary its intensity as the air varies in density when sound is passing through it, I can telegraph any sound, even the sound of speech" - A. G. Bell.

Em 1877, o objetivo de Alexander Graham Bell era de aperfeiçoar o telégrafo. Entretanto, ele não imaginava que com essa simples ideia estabeleceria os princípios básicos do telefone, um aparelho que revolucionaria as comunicações à distância. Esse evento marca o início do envolvimento da ciência em desempenhar um papel importante no processamento do sinal de voz.

Na segunda metade do século XX, muitos estudos foram publicados para analisar a natureza da voz, as diferentes formas de representação e suas características temporais e espectrais. Os principais achados da época podem ser encontrados no artigo escrito por Schafer e Rabiner de 1975 [4] . De fato, pesquisas como essa auxiliaram no desenvolvimento das comunicações modernas e impulsionaram a criação de novas aplicações. Isso fez com que a transmissão do sinal de voz através da telefonia se popularizasse muito ao longo do tempo, ocupando um espaço importante na vida das pessoas de forma tal que, possivelmente, nem A. G. Bell previa.

1.1 Histórico

A história dos sistemas telefônicos se inicia ainda no século XIX, mais precisamente entre 1850-1900 quando os transdutores de voz para sinais elétricos foram desenvolvidos. Um dos precursores desta tecnologia foi A. G. Bell. Ainda na forma analógica, o serviço de telefonia já despertava um grande interesse das empresas de correios e telégrafos. Os primeiros telefones não pertenciam a nenhuma rede e eram basicamente para uso particular. Ainda nessa época, cada telefone era ligado a uma central responsável pela comutação entre chamadas. E mesmo com as limitações das primeiras redes, ainda muito restritas e isoladas, companhias como a *Bell Telephone Company*, fundada em 1877 nos Estados Unidos, desenvolviam um novo mercado

no setor de telecomunicações. Anos depois, a partir da união de outras empresas do setor, a já conhecida *American Bell Telephone Company* deu origem em 1880 à *American Telephone & Telegraph Company* (AT&T), que por muitos anos foi a maior companhia de telefonia do mundo. Um dos desejos da AT&T era de expandir o serviço para um número cada vez maior de cidades para atender o novo hábito da população, como pode ser visto na Figura 1.1.

Figura 1.1: Folheto da AT&T sobre oferta de ligações de longa distância divulgado em 1891. Imagem reproduzida de [1]

Entretanto, o interesse no processo de digitalização só foi alavancado pelas companhias de telecomunicações a partir do interesse comum de expansão das PSTNs (*Public Switch Telephony Network*) como eram conhecidas as diferentes redes que permitiam a comutação entre chamadas de diversas localidades. A indústria de telefonia se apoiava num projeto global de modernização e padronização que interligaria as principais redes de comunicação. De forma geral, essa iniciativa foi dividida em duas etapas. Primeiramente, realizou-se a expansão da infraestrutura de redes, com *links* mais longos e mais capilares. Em seguida, o setor incentivou e estimulou o desenvolvimento de técnicas de transmissão do sinal de voz, cuja representação digital mais simples da forma de onda no domínio do tempo tem sido até hoje o formato PCM (*Pulse Code Modulation*).

Nos anos 70, a UIT (União Internacional de Telecomunicações), através da publicação do documento ITU-T G.711, padronizou técnicas de quantização não uniforme conhecidas como Lei-μ e Lei-A no sinal PCM para representar as amostras com número reduzido de níveis. Nos dois casos, apenas 8 bits são necessários para representar os índices das amostras quantizadas garantindo a qualidade subjetiva original do sinal de voz. Desta forma, a partir de um sinal de voz com banda limitada entre 300 - 3400 Hz, usando a frequência de amostragem de 8 kHz, um canal de voz codificado em G.711 alcança uma taxa de 64 kbps [5].

A partir de novas sinalizações e técnicas de multiplexação no tempo, começariam

a ser formados os canais de telefonia digital. O primeiro nível ficou conhecido como DS0 (*Digital Signal 0*) com exatos 64 kbps e, de forma hierárquica, foram propostos os canais T1 (24 x DS0) e E1 (32 x DS0) com 1536 kbps e 2048 kbps de capacidade, respectivamente. Estabelecia-se assim a padronização da telefonia digital a ser empregada por muitas décadas nas redes de comunicações.

1.2 Motivação

Nos últimos anos já temos observado que a comunicação de voz tem se estabelecido de maneira bastante diversificada. Neste novo cenário, as chamadas telefônicas através de redes tradicionais de acesso fixo deram lugar à comunicação de voz via IP (VoIP) e tecnologias de acesso móvel. Com o advento desses novos meios foram criados novos requisitos de transmissão, traduzidos muitas vezes em restrições e limitações na banda de transmissão para aumentar o número de canais e a oferta de serviço. No entanto, as mesmas características que demandam a expansão da capacidade, trazem consigo o desafio de superar a qualidade no transporte. O reflexo é que parte do progresso que temos observado está associado ao processamento digital da voz e técnicas mais eficientes de compressão. Indiscutivelmente, o advento das aplicações móveis e seu rápido crescimento em escala mundial impulsionam novas linhas de pesquisa e desenvolvimentos na busca de mais qualidade. Além disso, continuamente vemos que os dispositivos móveis são dotados de mais capacidade de processamento, memória e sistemas inteligentes embarcados, o que faz com que o telefone se torne item fundamental para a disseminação da tecnologia permitindo acréscimo de complexidade dos algoritmos de compressão, na universalização das comunicações, bem como no processo de inclusão digital.

Mesmo com a expansão dos acessos às redes de banda larga fixa ou móvel com taxas maiores de transferências, atualmente as redes se caracterizam por trafegar aplicações multi-serviços cada vez mais personalizadas, com diferentes taxas e requisitos. Nesse intuito, diferentes técnicas têm sido aplicadas na etapa de compressão dos sinais de voz. Em trabalho recente, [6], foram propostas muitas contribuições relevantes ao tema usando como base a técnica conhecida como MMP.

A concepção do algoritmo de compressão de sinais multidimensionais usando recorrência de padrões multiescala - MMP (do inglês: *Multidimensional Multiscale Parser*) [7], deu origem a uma nova linha de pesquisa na área de codificação de sinais. Desde então, diversas contribuições aperfeiçoaram o método e ampliaram os campos de aplicação do MMP [3], [8], [9], [10], que originalmente foi desenvolvido com foco na compressão de imagens. Em essência, esse algoritmo possui uma característica universal para sinais multidimensionais, uma vez que ele prevê casamentos de padrões com palavras código de diferentes dimensões, semelhante ao processo de

quantização vetorial aplicado a diferentes escalas.

Atualmente, o MMP é um método de codificação de fonte de imagens e vídeos bastante eficiente que supera a qualidade de padrões tradicionais como JPEG, JPEG2000 e H.264 AVC Intra. O algoritmo também se demonstra muito competitivo se aplicado a outros tipos de sinais como eletrocardiograma, eletromiograma [11] e radares meteorológicos [12].

A aplicação do algoritmo MMP para compressão de sinais de voz possui um potencial subestimado ainda pouco explorado. Pela característica adaptativa inerente à técnica, o MMP é um forte candidato a obter os ganhos de compressão para sinais de voz semelhantes aos produzidos por outras fontes de sinal. Portanto, temos à frente um objetivo bastante atual e um cenário bastante favorável.

1.3 Proposta de Trabalho

Este trabalho tem como principal objetivo desenvolver um sistema de compressão baseado no algoritmo MMP aplicado a sinais de voz e avaliar a qualidade do sinal comprimido à taxa alvo de 8 kbps, comumente utilizada nos sistemas modernos de comunicações. A avaliação de qualidade é feita a partir de uma métrica de distorção perceptual, conhecida como PESQ [13].

O sistema proposto neste trabalho incorpora todas as ferramentas empregadas com sucesso em trabalhos predecessores a este. Demos ao MMP uma visão estruturada do algoritmo que auxilia bastante na compreensão de suas etapas de processamento. Essa nova organização que descrevemos com diagramas em blocos ao longo da dissertação é importante para aplicarmos novas funcionalidades à estrutura básica. Além disso, introduzimos novas análises de dados quadro-a-quadro para inferir o comportamento de codificação do algoritmo MMP quanto à perceptualidade. Com isso, visamos adicionar critérios que levem em conta fenômenos do sistema auditivo humano no casamento de padrões.

Adicionalmente, um codificador e um decodificador de voz baseado em MMP foram inteiramente desenvolvidos para apoiar futuras pesquisas e se destacam como outra contribuição relevante deste trabalho. Todas as funções foram desenvolvidas utilizando a linguagem de programação `C ANSI`, no ambiente Microsoft Windows®. Todo o código tem característica multiplataforma para permitir a compilação em ambientes UNIX/Linux. Os dados que conseguimos gerar a partir de funções adicionais contribuíram muito para entendermos melhor o funcionamento do MMP Voz.

1.4 Organização da tese

No capítulo 2 apresentamos as características das duas categorias clássicas de sistemas de compressão que incluem as abordagens com e sem perdas. Apresentamos também os diferentes grupos de codificadores de voz existentes e introduzimos o MMP nesse contexto.

No capítulo 3 descrevemos a estrutura básica do algoritmo MMP. Destacamos cada ferramenta sua e ilustramos o processo de compressão através de diagramas em bloco.

No capítulo 4 fazemos uma rápida revisão sobre as técnicas de predição linear que têm sido empregadas em sistema de compressão de voz há algum tempo. No entanto, esse é um capítulo que nos permite formular a predição em blocos, o que é bastante útil para comparar com os diferentes algoritmos propostos no capítulo seguinte.

No capítulo 5, descrevemos as estruturas complementares do MMP que incluem, por exemplo, os estágios de filtragem. Além disso, propomos novos algoritmos de predição e algumas inovações à estrutura básica do MMP. Apresentamos os resultados experimentais que obtivemos com base nos parâmetros definidos em trabalhos predecessores e destacamos os ganhos obtidos pouco a pouco a partir das novas funcionalidades que incorporamos ao processo de codificação.

No capítulo 6 introduzimos as análises perceptuais no MMP. Novas propostas no casamente do padrões que levam em consideração as características do sistema auditivo humano são apresentadas. Além disso, outras contribuições perceptuais são apresentadas nas etapas de pré e pós processamento que melhoram a qualidade média do nosso banco de frases.

No Apêndice A apresentamos a lista dos sinais de testes com suas características. Todos os resultados médios foram produzidos a partir dessa lista. No Apêndice B, descrevemos as diferentes formas de acesso à árvore binária. Os detalhes de implementação do codificador e do decodificador e o dicionário de funções são apresentados no Apêndice C. No Apêndice D, incluímos os scritps auxiliares que compõem o codificador e o decodificador.

Capítulo 2

Codificação de Voz

Esse capítulo compara as duas abordagens clássicas dos sistemas de compressão. Apresentamos as formulações matemáticas da compressão, com perdas e sem perdas, e mostramos um contexto dos grupos de codificadores de voz. Fornecemos também uma visão geral dos algoritmos e padrões que ainda são bastante utilizados nos sistemas de comunicação.

2.1 Sistemas de Compressão

A história da telefonia se confunde muitas vezes com a história da codificação do sinal de voz, seja no processo de digitalização dos primeiros sistemas analógicos ou na criação de novos padrões de transmissão mais robustos e mais eficientes. Em sistemas digitais, parte dessa eficiência está diretamente relacionada com as técnicas de compressão. Em essência, essas técnicas se baseiam no princípio de representar amostras de um sinal fonte por um conjunto de símbolos com a menor quantidade de bits possível, quando usamos um alfabeto binário. Matematicamente, podemos representar a transformação do sinal fonte $\mathcal{X}$ em $\mathcal{X}_c$ por uma função $\mathfrak{T}$ que leva as amostras de um domínio ao outro: $\mathcal{X} \xrightarrow{\mathfrak{T}[.]} \mathcal{X}_c$.

Idealmente, desejamos que o processo de compressão seja inversível, ou seja, $\mathfrak{T}^{-1}$ deve existir. As técnicas aplicadas neste processo podem ser classificadas em duas categorias: *sem perdas*, onde a reconstrução $\mathcal{Y}$ a partir de $\mathcal{X}_c$ é idêntica a fonte $\mathcal{X}$ e, *com perdas*, onde a reconstrução não é idêntica. Em outras palavras, se intuitivamente pensarmos em uma medida de diferença ou distância entre o sinal fonte e o sinal reconstruído $D(\mathcal{X}, \mathcal{Y})$, concluiremos que:

$$D(\mathcal{X}, \mathcal{Y}) = 0, \text{para o caso } \textbf{sem perdas,} \tag{2.1}$$

$$D(\mathcal{X}, \mathcal{Y}) \neq 0, \text{para o caso } \textbf{com perdas} \tag{2.2}$$

2.1.1 Compressão sem perdas

Definição: Procuramos um codificador C de fonte reversível que minimiza o tamanho médio $L(C)$ do código $C(x)$ de uma variável aleatória X com alfabeto $\mathcal{X}$.

Se $l(x)$ é o comprimento da palavra de código $C(x)$, então

$$L(C) = \sum_{x \in \mathcal{X}} p(x) l(x), \tag{2.3}$$

onde $p(x)$ é a probabilidade associada à realização x em X. A descoberta de Shannon em [14] encontrou o limite teórico para o tamanho médio mínimo do código: a entropia $H(X)$ definida por

$$H(X) = -\sum_{x \in \mathcal{X}} p(x) \log_2 p(x). \tag{2.4}$$

Portanto, a entropia de uma variável aleatória discreta representa o limite de compressão desta fonte, ou seja, $H(X) \leq L(C)$. O algoritmo de Huffman e o codificador aritmético são exemplos de códigos que se aproximam do limite de Shannon, já implementados em diversos padrões de compressão. Um aprofundamento maior neste tema pode ser obtido em [15] e [16].

2.1.2 Compressão com perdas

Definição: Dada uma métrica de distorção D, procuramos um processo de codificação $\mathfrak{T}_d$ e decodificação $\mathfrak{T}_d^{-1}$, tal que para cada $d \in \mathbb{R}$ o tamanho médio das representações $x_d = \mathfrak{T}_d(x)$ seja mínimo, ou seja, $l(x_d) < l(x)$. Este processo está sujeito à restrição $D(x, \hat{x}) \leq d$, onde $\hat{x} = \mathfrak{T}_d^{-1}(x_d)$.

De uma forma geral, as técnicas de compressão exploram as características temporais, espectrais, estatísticas e espaciais (quando processamos imagens e vídeo) do sinal fonte. Algumas técnicas têm por objetivo remover toda a informação redundante para evitar a transmissão ou armazenamento excessivos de dados. Outras se propõem a extrair parâmetros do sinal fonte que possam ser usados na síntese durante a decodificação. E ainda há aquelas que utilizam uma combinação de ambos. Em abordagens mais recentes os métodos de compressão tendem a explorar também as condições perceptuais seja do sistema auditivo ou do sistema visual humano. Neste caso, o objetivo é evitar a transmissão de dados "irrelevantes" para a compreensão humana, ou em outras palavras, que as distorções inseridas pela etapa de compressão sejam imperceptíveis. Dessa forma, podemos atribuir um critério de distorção subjetivo D_{sub} entre o sinal fonte $\mathcal{X}$ e o sinal reconstruído $\hat{\mathcal{X}}$ para que aliado às condições perceptuais tenhamos uma diferença aproximadamente nula, ou seja, uma semelhança maior. Matematicamente, podemos representar essa afirmação por:

$$D_{sub}(\mathcal{X}, \hat{\mathcal{X}}) \approxeq 0. \tag{2.5}$$

De fato, a variação dessa distorção subjetiva está diretamente relacionada à variação de qualidade do sinal reconstruído, pois quanto menor a diferença perceptual, mais fiel é a representação codificada.

2.2 Métricas de Distorção

Originalmente, os testes de qualidade em sinais de voz eram feitos através de testes subjetivos, onde um grupo de pessoas participavam das sessões de avaliação. Cada avaliador era convidado a dizer sua própria opinião, sugerindo uma nota para cada sentença. Com a intenção de criar um sistema padronizado, a União Internacional de Telecomunicações (UIT) publicou a recomendação ITU-T REC. P.800 que descreve as condições de realização dos testes, de como devem ser conduzidos e de como calcular uma nota média da qualidade. Isso fez com que diferentes sistemas pudessem ser avaliados pela mesma referência tornando a comparação de qualidade válida e universal.

Com o objetivo de reduzir prazos, custos e tornar o processo de avaliação mais prático muitas pesquisas foram desenvolvidas com o objetivo de mapear as características do sistema auditivo humano. Os resultados desses estudos permitiram a criação de métodos objetivos de análise cada vez mais precisos. Esses métodos se propõem a estimar uma nota média de qualidade, conhecida como MOS, do inglês *Mean Opinion Square*, cuja escala de 1 a 5 indica a qualidade do sinal reconstruído, como visto na tabela 2.1. Quanto maior a nota, mais semelhante o sinal em teste está do sinal original, ou seja, maior é a qualidade.

Tabela 2.1: Tabela MOS para sinais de voz

Nota	**Qualidade**	**Nível de Distorção**
5	Excelente	Imperceptível
4	Boa	Perceptível mas não incomoda
3	Regular	Perceptível e incomoda pouco
2	Ruim	Perceptível e incomoda
1	Péssima	Perceptível e incomoda muito

Uma medida de fidelidade adequada deve, por princípio, se basear na qualidade percebida e, portanto, considerar fenômenos importantes da nossa perceptualidade como, por exemplo, o efeito de mascaramento. Por definição, medidas de distorção como *Mean Square Error* - MSE - (definido na Equação (2.6)) e *Mean Absolute*

Error - MAE - (definido na Equação (2.7)) desconsideram tais fenômenos. No entanto, MSE e MAE ganharam muita popularidade como medida de distorção pela simplicidade de implementação e por serem intuitivamente um valor de proximidade. Uma discussão rica sobre o assunto pode ser encontrada em [17].

$$D(x, y) = (x - y)^2 \tag{2.6}$$

$$D(x, y) = |x - y| \tag{2.7}$$

Em 2001, a UIT publicou a recomendação ITU-T REC. P.862 que atualmente é o algoritmo mais adequado para avaliar sistemas de telefonia, considerando em suas etapas de cálculos, influências de erro no canal de transmissão, como inserção de ecos, variação dos quadros de voz ou *jitter* de rede, mas, principalmente, influências dos sistemas de compressão de sinais de voz com banda limitada (300 - 3400 Hz). Também conhecido como PESQ (*Perceptual Evaluation of Speech Quality*) [13], esse algoritmo é uma evolução do PSQM (*Perceptual Speech Quality Measure*) [18], recomendação também desenvolvida pelo mesmo subgrupo da UIT. No entanto, a nova recomendação inclui funcionalidades importantes para detectar também distorções localizadas, pois a comparação do sinal de referência com o sinal degradado é feita basicamente através de duas modelagens, a saber:

- Psicoacústica: é a representação interna do sinal de voz no sistema auditivo, ou seja, uma série de transformações são aplicadas tanto no sinal de referência quanto no sinal degradado para mapear as componentes espectrais na escala de Bark[1] e descobrir as bandas críticas que, por definição, variam de acordo com a intensidade percebida (*loudness*) do sinal.

- Cognitiva: calcula as distorções entre os dois sinais como funções no tempo e na frequência e, por fim, mapeia esses valores em notas objetivas.

Essas notas, por sua vez, têm um sentido subjetivo, pois reflete mais fielmente a qualidade da voz à luz das características perceptuais. Atualmente, o PESQ é o algoritmo mais indicado para comparação de sistemas de compressão devido à alta correlação entre as notas produzidas por ele e as notas obtidas por avaliadores em sessões subjetivas.

[1] A escala de Bark é aquela em que cada banda ou faixa de frequência pode ser interpretada como um "canal" da cóclea, onde as ondas sonoras são percebidas com a mesma intensidade [19] [20]. A função que mapeia frequência em Hertz na escala de Bark geralmente usada é $Z_b(f) = 13 \arctan(0.00076f) + 3.5 \arctan\left[\left(\frac{f}{7500}\right)^2\right]$.

2.3 Codificação de Voz

De uma forma geral, as técnicas de compressão de voz podem ser divididas em três principais grupos: codificadores de forma de onda, codificadores de fonte e codificadores híbridos [21]. Os codificadores de forma de onda usam características temporais e espectrais do sinal associando as próprias amostras originais a códigos que traduzidos no decodificador geram uma forma de onda muito semelhante ao sinal fonte. Esses codificadores usam a taxa de bits suficiente para reproduzir a voz da forma mais fiel que conhecemos atualmente. Técnicas como o PCM (*Pulse Code Modulation*), DPCM (*Differential Pulse Code Modulation*) e ADPCM (*Adaptative Pulse Code Modulation*) incluem-se no primeiro grupo e conseguem reproduzir sinais de voz com alta qualidade e baixa complexidade a taxas de 64 kbps, 32 kbps e 16 kbps, respectivamente.

Os codificadores fonte, por sua vez, conseguem atingir os mais altos graus de compressão. Isso é feito através da extração de parâmetros do sinal original que modelam o sistema do trato vocal humano através de uma predição linear. A grande desvantagem desta família de codificadores é a qualidade sintética do sinal reconstruído que produz sinais de baixa qualidade se comparados ao sinal de referência, porém inteligíveis com taxas entre 1 e 2,4 kbps. O codificador de fonte mais conhecido atualmente é o *vocoder* LPC (*Linear Predictive Coding*) [22]. Em 1976, para comunicações militares seguras o governo dos Estados Unidos, sob a regulamentação FS-1015, padronizou a codificação LPC-10 que utiliza ordem 10 no filtro de predição.

No entanto, é na terceira categoria dos codificadores de voz que estão as técnicas mais eficientes conhecidas até o momento e que estão reunidas num processo de análise por síntese [23] [24]. Nesse grupo destacam-se o codificador por excitação de multi-pulsos (MPE - *Multi-Pulse Excitation*) com boa qualidade a 9,6 kbps, o codificador por excitação de pulsos regulares (RPE - *Regular Pulse Excitation*), cuja variação RPE-LTP (*Long-term Prediction*) é empregado no padrão GSM de telefonia móvel na Europa a 13 kbps. Cabe mencionar ainda o codificador por excitação de pulsos com predição linear (CELP - *Code Excited Linear Prediction*), largamente utilizado a 8 kbps em sistemas de voz por IP (VoIP).

Os codificadores híbridos oferecem uma boa qualidade no sinal de voz reconstruído a taxas intermediárias. Percebemos este fato a partir do gráfico da Figura 2.1. Nesta ilustração, observa-se que os codificadores de forma de onda não têm boa qualidade a taxas abaixo de 16 kbps, enquanto os codificadores de fonte não melhoram a qualidade mesmo a taxas altas.

A técnica baseada no MMP, por sua vez, quebra o paradigma de análise por síntese na codificação do sinal de voz. Na verdade, o MMP é um método disruptivo e pode ser interpretado por três diferentes aspectos. Primeiramente, vemos o

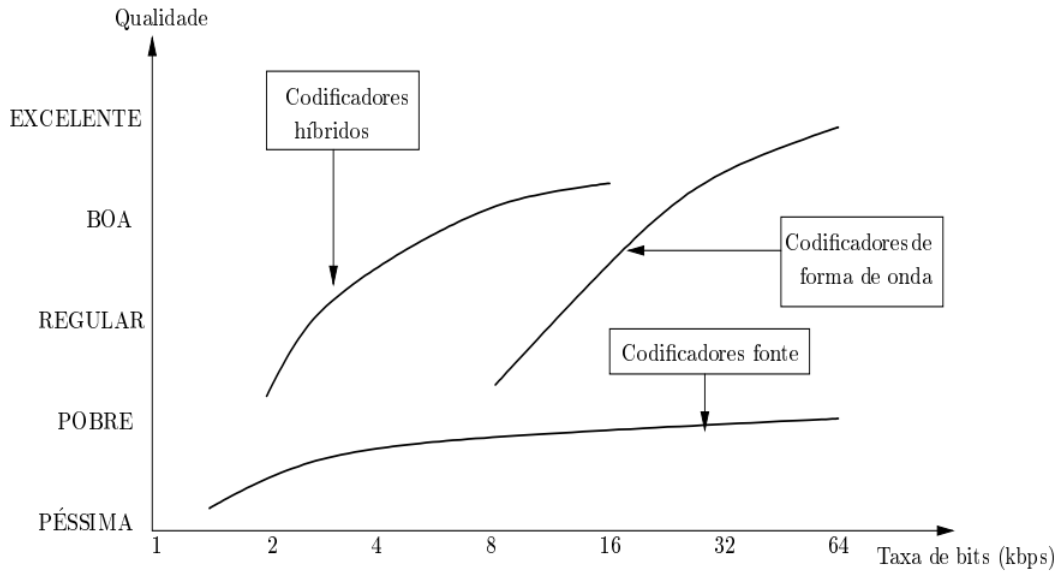


Figura 2.1: Comparação Taxa x Qualidade dos codificadores de voz. Imagem reproduzida de [2].

MMP como um quantizador vetorial adaptativo se não incluirmos os procedimentos de atualização do dicionário através de transformações de escala, uma vez que ele procura aproximar segmentos de diferentes dimensões a partir de um *codebook* adaptativo. Outra interpretação que pode ser associada ao MMP é a mesma dos algoritmos da família Lempel-Ziv, cuja codificação e feita a partir de concatenações de elementos recorrentes. E, por fim, se entendermos que inicialmente os elementos do dicionário original compõem um conjunto de pulsos com diferentes amplitudes e escalas, a medida que o processamento avança, novas funções são incluídas no dicionário, geradas a partir da concatenação de pulsos iniciais, ou em outras palavras, coeficientes. Então, as novas palavras do dicionário podem ser vistas como funções base e o sinal é codificado pela contração e expansão de pulsos. A técnica MMP será discutida com maiores detalhes no capítulo a seguir.

Capítulo 3

A Estrutura Básica do Algoritmo MMP

O MMP é um algoritmo de compressão em blocos de taxa variável. Ele divide o sinal original em segmentos que são aproximados a partir de vetores de diferentes comprimentos provenientes de dicionários adaptativos, cujos índices são submetidos a um codificador de entropia aritmético para alocação ótima de bits. Como o MMP processa o sinal em blocos, o atraso mínimo previsto no processo entrada-saída é proporcional ao tamanho do bloco. Um grande destaque do algoritmo consiste na rapidez no aprendizado dos padrões existentes no sinal de interesse, que são atualizados durante o processo de codificação e inseridos no dicionário. Isso permite que padrões recorrentes, transformados ou não, sejam usados na codificação de blocos subsequentes.

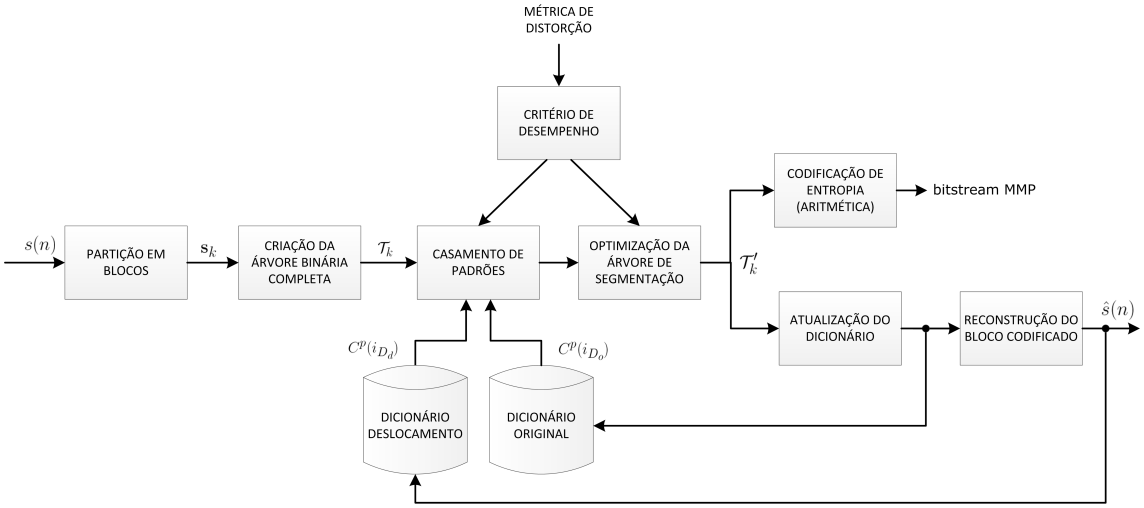
A julgar pela característica universal do MMP, uma vez que nenhum conhecimento prévio do sinal é necessário, ele pode ser aplicado às duas diferentes abordagens de compressão: com e sem perdas. Uma revisão detalhada das formulações matemáticas relativas às duas abordagens pode ser encontrada nos Apêndices E e F de [7] e no Apêndice D de [9].

O codificador MMP com predição linear para sinais de voz é composto fundamentalmente pelas seguintes etapas:

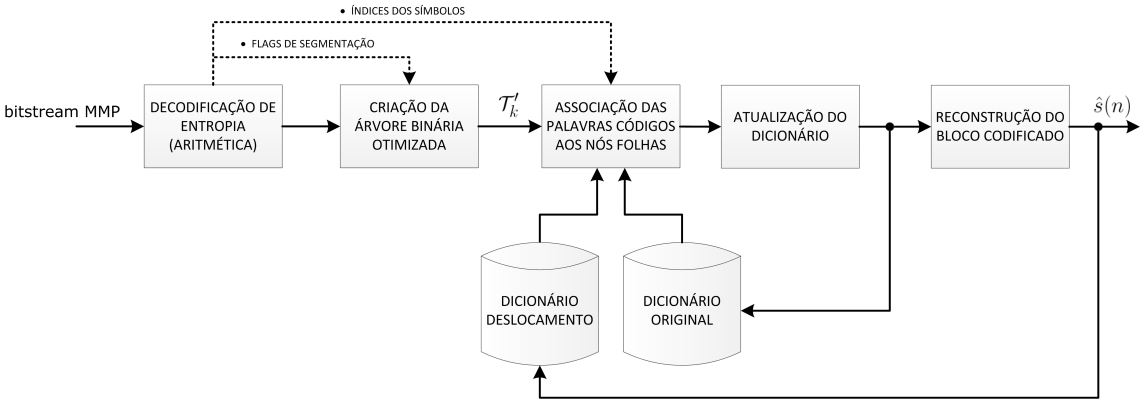
1. Partição em blocos do sinal de entrada;
2. Associação das amostras com a estrutura de dados em árvore;
3. Casamento de padrões recorrentes;
4. Otimização da árvore de segmentação;
5. Codificação de entropia para representação dos nós folha da árvore;

6. Atualização do dicionário com transformações de escala, equalização de norma e controle de redundância.

As Figuras em 3.1 (3.1a e 3.1b) apresentam os diagramas em blocos do codificador e decodificador, respectivamente. Cada uma destas etapas será detalhada nas seções seguintes.



(a) Diagrama em blocos do codificador MMP.



(b) Diagrama em blocos do decodificador MMP.

Figura 3.1: Em (a) e (b) $s(n)$ representa o sinal de entrada e $\hat{s}(n)$ o sinal reconstruído (decodificado). $\mathcal{T}_k$ identifica a árvore binária completa, construída a partir da representação em blocos $\mathbf{s}_k$ do sinal de entrada. $\mathcal{T}'_k$ representa a árvore binária após o processo de otimização. O decodificador inicia o processamento ao receber o *bitstream* e recupera $\mathcal{T}'_k$ a partir dos *flags* de segmentacão. Cada nó folha é associado ao vetor selecionado pelo codificador identificado pelo índice do dicionário.

3.1 Partição em blocos

O sinal de entrada $s\,(n)$ é particionado em segmentos lineares com comprimento N, cujo valor é uma potência de 2. Portanto, cada bloco de índice k possui $N = 2^p$

amostras, onde p representa a ordem do bloco. O sinal de entrada pode, então, ser representado como uma sequência de blocos, ou seja:

$$s(n) = [\mathbf{s}_{k-M} \quad \mathbf{s}_{k-M+1} \ \cdots \ \mathbf{s}_{k-1} \quad \mathbf{s}_k \quad \mathbf{s}_{k+1} \ \cdots \ \mathbf{s}_{k+M-2} \quad \mathbf{s}_{k+M-1}] \tag{3.1}$$

onde $2M$ é o total de partições e,

$$\mathbf{s}_k(n) = s(kN + n), \quad n = 0, 1, 2, ..., N - 1 \tag{3.2}$$

Matematicamente, podemos representar o processo de particionamento do sinal de entrada aplicando a janela retangular finita deslocada $w(n - m)$, se $m = kN$.

$$\mathbf{s}_k(n) = s(n)w(n - kN) \tag{3.3}$$

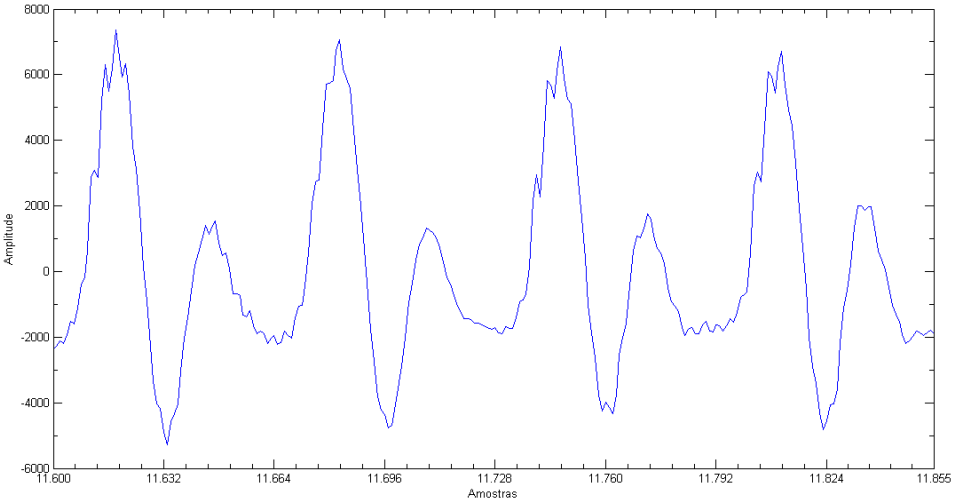
$$w(n) = \begin{cases} 1, & kN \leq n \leq k(N-1) \\ 0, & \text{outro } n \end{cases} \tag{3.4}$$

O sinal de voz é um processo não estacionário, por natureza. No entanto, se olharmos apenas para trechos entre 10 - 30 ms de duração, percebemos que tais amostras são realizações de um processo mais bem comportado, cujas propriedades estatísticas tendem a variar pouco [25]. Isto significa que podemos interpretar e aproximar o sinal de voz como um sinal estacionário por partes, de tal forma que as características temporais e espectrais das amostras pertencentes ao mesmo quadro podem ser utilizadas para classificá-lo como sonoro, fricativo (surdo) ou de silêncio. No MMP, o tamanho do segmento N pode ser tal que um bloco $\mathbf{s}_k$ tenha amostras suficientes para representar um desses três tipos de quadro ilustrados nas Figuras 3.2a, 3.2b e 3.2c a seguir.

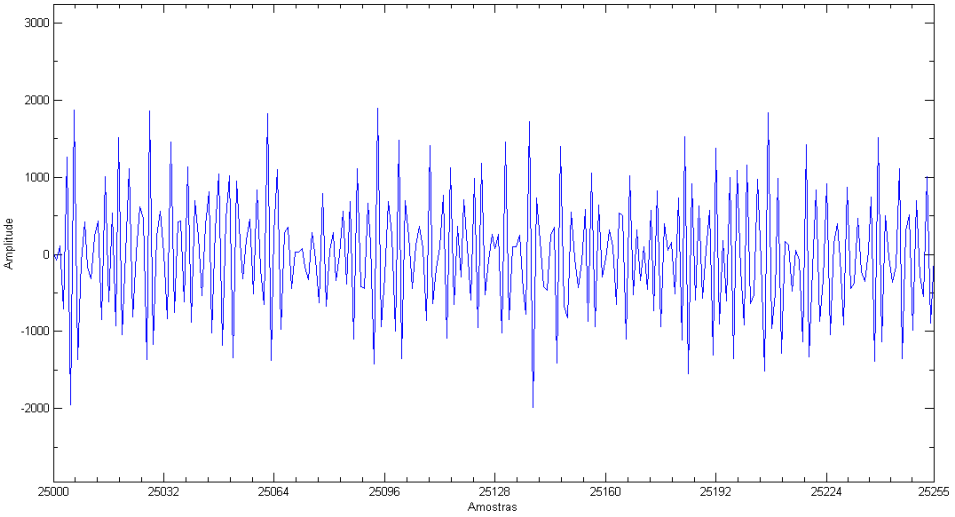
A máxima eficiência do algoritmo será atingida quando o codificador conseguir representar fielmente e com o menor número de bits um bloco com o maior número de amostras possíveis. Por outro lado, note que quanto menor a dimensão do bloco, mais taxa será exigida para codificar uma quantidade maior de blocos que compõem a entrada. Portanto, sugere-se que N seja escolhido de acordo com a natureza do sinal. Alguns testes experimentais foram realizados com diferentes valores de N e serão apresentados na seção de resultados desta Dissertação.

3.2 A árvore binária completa

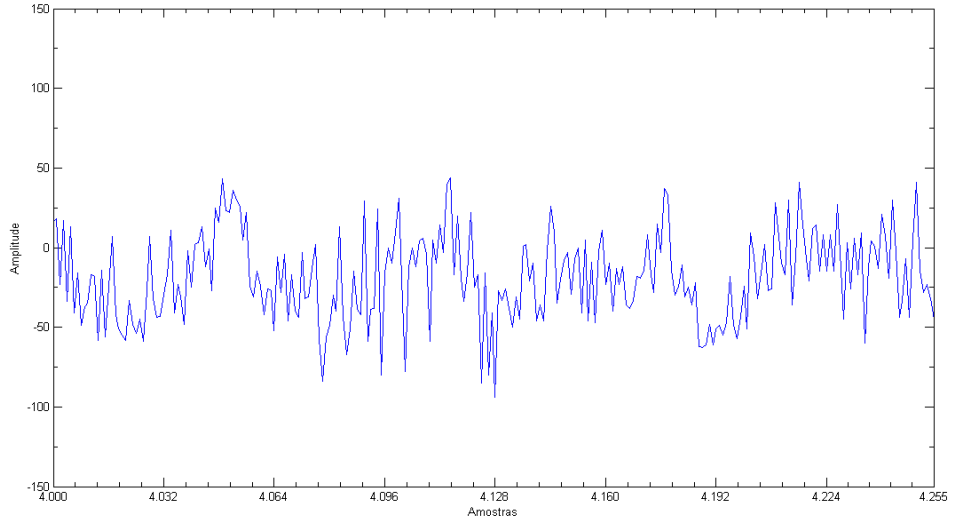
A árvore binária é a base da estrutura de dados de codificação do algoritmo MMP onde cada nó possui no máximo dois nós filhos, comumente conhecidos como es-



(a) O som da vogal /e/ em *near*



(b) O som fricativo do fonema /sh/ em *fresh*



(c) Silêncio que antecede a frase us39.wav

Figura 3.2: Representação de trechos sonoros, surdos e de silêncio.

querda e direita. São chamados de nós pai os nós que possuem filhos, ou seja, nós que tenham ramos não nulos. Por outro lado, os nós que não possuem filhos são chamados de nós folha. O nó raiz é o nó que não possui pai e pode existir apenas um nó raiz por árvore. Outra definição importante nesta estrutura de dados é a profundidade ou nível, cujo valor representa a distância percorrida do nó raiz até o nó filho ou folha de interesse. Na sua forma completa, todos os nós folha da árvore binária estão na mesma profundidade e o nó raiz sempre terá profundidade 0 (zero). A árvore binária completa possui as seguintes características:

- Quantidade de níveis ou profundidade $\mathcal{P}$ da árvore igual a:

$$\mathcal{P} = \log_2(N) + 1 \tag{3.5}$$

- Quantidade de nós $\mathcal{N}$ pertencentes à arvore completa igual a:

$$\mathcal{N} = 2^{\mathcal{P}} - 1 \tag{3.6}$$

- Quantidade de ramos $\mathcal{B}$ igual a:

$$\mathcal{B} = \mathcal{N} - 1 \tag{3.7}$$

- Quantidade de nós folhas $\mathcal{N}_L$ igual a:

$$\mathcal{N}_L = N \tag{3.8}$$

Inicialmente, cada nó deve ser associado a um segmento ou partição do bloco de entrada, de forma que a concatenação dos nós folhas represente o próprio bloco de entrada, como pode ser visto em

$$\hat{X}_0 = (\hat{X}_7 : \hat{X}_8 : \cdots : \hat{X}_{14}) \tag{3.9}$$

e

$$\mathcal{L}\left[\hat{X}_0\right] = \mathcal{L}\left[(\hat{X}_7 : \hat{X}_8 : \cdots : \hat{X}_{14})\right], \tag{3.10}$$

onde a simbologia $(a : b)$ indica a concatenação de a e b e $\mathcal{L}[.]$ indica o tamanho do vetor. Isto significa que o nó raiz também representa o mesmo bloco de entrada, conforme ilustrado na Figura 3.3, que representa uma árvore de ordem (profundidade) 4. Recomenda-se que a árvore seja criada com as seguintes propriedades:

- Ser transversa com a capacidade de ser percorrida no sentido pós-ordem e pré-ordem (ver Apêndice B). Portanto, cada nó deve permitir o acesso aos nós filhos.

- Cada nó η deve possuir um índice i ordenado de identificação, η_i.
- Nó raiz, η_0 tem profundidade $p = 0$ e pertence à escala $l = \log_2(N)$.
- Os nós folhas, $\eta_{2^{\mathcal{P}-1}-1}, \eta_{2^{\mathcal{P}-1}}, ..., \eta_{2^{\mathcal{P}}-2}$ têm profundidade $p = \log_2(N)$ e escala $l = 0$.
- Cada nó intermediário η_i tem escala $l_i = \log_2(\mathcal{L}[\eta_i])$ e profundidade $p_i = \mathcal{P} - l_i - 1$.

A árvore binária também pode ser criada de forma bem eficiente usando uma estrutura implícita de vetor. Os filhos do nó que possui índice i podem ser acessados nos índices $2i + 1$ (esquerda) e $2i + 2$ (direita), enquanto o pai pode ser visitado no índice $\lfloor (i - 1)/2 \rfloor$, onde $\lfloor a \rfloor$ representa o maior inteiro menor ou igual a a. Vale ressaltar que o nó raiz tem índice 0 (zero). A Figura 3.4 ilustra este caso.

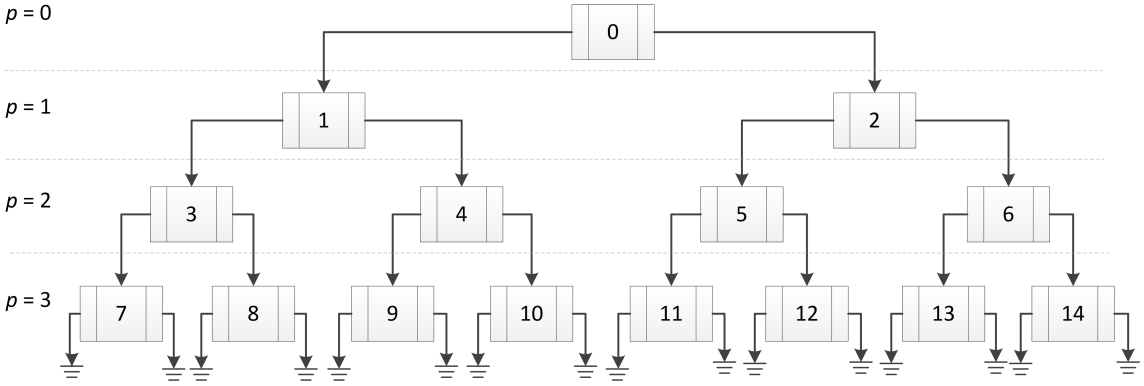


Figura 3.3: Árvore binária completa

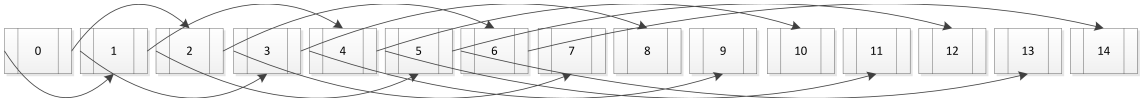


Figura 3.4: Mapeamento dos nós da árvore na forma de vetor.

3.3 Casamento de Padrões

Nesta etapa, todos os nós da árvore binária são visitados com o objetivo de selecionar palavras código do dicionário com o menor custo segundo um critério de desempenho. Portanto, após a realização das buscas, cada nó possuirá a melhor aproximação do sub-bloco original, respeitando-se a escala, ou seja, a mesma dimensão. Nesta implementação empregamos a busca pela força bruta, percorrendo todas as possíveis entradas dos dicionários para obtermos o casamento com o menor custo. As estruturas dos dicionários envolvidos no processo de casamento serão discutidas em detalhes nas Seções 3.6 e 3.7.

3.3.1 Critério de desempenho

Originalmente, as pesquisas desenvolvidas para o algoritmo MMP aplicavam um critério de desempenho que, por princípio, forçava o algoritmo a buscar um valor alvo de distorção d^*. Esse valor pode ser interpretado como uma medida de fidelidade ou proximidade de um bloco ou sub-bloco reconstruído. Na prática, a distorção de cada nó é calculada a partir do erro quadrático

$$\begin{aligned} d_{\eta,i} &= ||X^p - \hat{X}_i^p||^2 \\ &= \sum_{l=0}^{L-1} \left(X^p(l) - \hat{X}_i^p(l) \right)^2, i \in \mathcal{D}, \end{aligned} \tag{3.11}$$

e caso a aproximação por $\hat{X}_i^p$ produza uma distorção superior a d^* o sub-bloco é segmentado e uma nova análise é feita para X^{2p+1} e X^{2p+2} até que todos os nós sejam percorridos de forma recursiva.

No entanto, o MMP foi aperfeiçoado quando um novo critério de desempenho foi incluído durante a etapa de casamento de padrões. Esse critério considera a estimativa da taxa de representação da palavra-código durante o processo de quantização [7]. Uma revisão bibliográfica mais extensa sobre este tema pode ser encontrada nos estudos de Chou *et al* [26]. Até o momento, o uso do critério taxa-distorção R(D) computando o custo lagrangiano J tem produzido os melhores resultados do algoritmo MMP considerando diferentes fontes de sinal. Ele permite ponderar entre a menor distorção escolhida e a taxa para representar a palavra código, uma vez que calculamos um custo J conforme

$$J = D + \lambda R, \tag{3.12}$$

onde λ é o multiplicador de Lagrange.

O valor de λ é definido como parâmetro no codificador e permanece inalterado durante todo o processo de codificação. Para cada λ usado, uma taxa diferente de compressão é obtida (novo ponto na curva $R(D)$). Valores altos de lambda geram taxas altas de compressão, uma vez que a taxa de representação R passa a ser relevante na composição do custo J e um alto nível de distorção. Por outro lado, valores baixos de λ geram baixas taxas de compressão e um baixo nível de distorção.

No cálculo do custo $J(\eta)$ do nó η, consideramos uma métrica de distorção D_i associada à aproximação $\hat{X}_i^p$, cuja escala (dimensão) p é a mesma do sub-bloco X^p de entrada e uma taxa $R(i)_{\mathcal{D}}^p$ necessária para representar o índice i da palavra selecionada. Portanto, temos que:

$$\begin{aligned} J(\eta) &= D_i + \lambda R(i)_{\mathcal{D}}^p \\ &= ||X^p - \hat{X}_i^p|| + \{\lambda - \log_2[Pr(i|\mathcal{D}, p)]\}, \end{aligned} \tag{3.13}$$

onde $Pr(i|\mathcal{D}, p)$ é a probabilidade do índice i condicionada à profundidade p e ao tipo de dicionário $\mathcal{D}$, original ou deslocamento, descritos nas seções 3.6 e 3.7,respectivamente.

3.4 Processo de otimização da árvore

Nesta etapa tomamos a decisão de podar ou não os ramos dos nós pertencentes à árvore binária $\mathcal{T}_k$ de acordo com o critério de desempenho. Portanto, após esta análise de segmentação, a árvore resultante conterá os nós-folhas que, concatenados, representarão o bloco X_k com o menor custo global $J(\mathcal{T}_k)$. No processo de decisão avaliamos se o sub-bloco poderá ser representado apenas pelo nó-pai X^j ou pela concatenação dos nós-filhos $(X^{2j+1} : X^{2j+2})$, de acordo com o critério de menor custo lagrangeano.

Na análise de segmentação, adicionamos ao custo de representação da palavra código o custo de codificação do *flag* de segmentação. Assim como nas implementações passadas, estabelecemos a mesma convenção de valores: o *flag* '0' é usado para indicar segmentação, ou seja, é uma referência a um nó pai que não será podado e conterá seus respectivos filhos. Por outro lado, usamos valor '1' para indicar poda, ou seja, diz respeito a um nó folha. Se os custos lagrangeanos $J(\eta^j)$, associados com cada aproximação $\hat{X}^j$ dos nós são independentes, então os custos de duas sub-árvores $J(\mathcal{T}^f)$ e $J(\mathcal{T}^g)$ também são independentes. Então, tomando como exemplo a sub-árvore $\mathcal{T}^4$ que contém o nó pai X^4 e os respectivos filhos, X^9 e X^{10}, opta-se pelo menor custo lagrangeano, calculado da seguinte forma:

$$J(\mathcal{T}^4) = min\left\{J(\eta^4)_{\text{non\_seg}}, J(\eta^4)_{\text{seg}}\right\} \tag{3.14}$$

onde

$$J(\eta_4)_{\text{non\_seg}} = J(\eta_4) + \underbrace{\lambda R(1|p=2)}_{\text{Custo do flag de poda na profundidade p=2}} \tag{3.15}$$

e

$$J(\eta_4)_{\text{seg}} = J(\eta_4) + \underbrace{\lambda R(0|p=2)}_{\text{Custo do flag de segmentação na profundidade p=2}} . \tag{3.16}$$

Ou seja, se o custo do nó pai sem filhos for menor ou igual ao custo deste nó segmentado, forçamos a poda. Isto significa que:

$$J(\eta_m) \leq J(\eta^{2m+1}) + J(\eta^{2m+2}) + \lambda R(0|p=p_m), \tag{3.17}$$

e a sub-árvore $\mathcal{T}^m$ será podada. Caso contrário o custo do nó pai é atualizado, sendo

composto pela soma dos custos dos nós filhos mais o custo do *flag* de segmentação, isto é,

$$J(\eta_m) = D_i + \lambda \left\{ R(i|p = p_m, \mathcal{D}) + R(1|p = p_m) \right\}. \tag{3.18}$$

A cada análise de segmentação este processo é repetido de forma recursiva no percurso de pós-ordem da árvore binária. Como cada nó contém o custo agregado da sua própria sub-árvore, ao fim do processo de otimização de cada bloco completo teremos $J(\eta_{\text{raiz}}) = J(\mathcal{T}^0)$. Esta otimização é conhecida como algoritmo RD Aproximado e, como visto acima, ela considera que o dicionário original permanece inalterado durante todo o processo de otimização da árvore. Portanto, os sub-blocos são aproximados por palavras do dicionário na etapa de Casamento de Padrões de forma independente do processo de otimização. Desta forma, assumimos que não há dependência entre os custos dos nós de uma mesma árvore.

O algoritmo RD Aproximado é sub-ótimo no sentido que ele assume que os custos dos nós de uma mesma árvore não são vinculados. Uma forma de otimização alternativa, conhecida como algoritmo RD Modificado, foi proposto em [7] que além de realizar todas as tarefas do RD Aproximado, realiza uma análise de dependência entre os nós de uma mesma árvore. O RD Modificado explora o impacto da codificação do nó à direita do atual caso um dado elemento não seja incluído no dicionário, devido a uma poda. Este elemento ausente no dicionário poderia representar com um custo menor um ou mais nós à sua direita, e a sua falta acaba ocasionando um resultado de maior custo, reduzindo o desempenho do sistema.

Alguns estudos comprovaram que a análise de dependência entre os nós da árvore de segmentação melhora o desempenho do sistema na codificação de imagens ao preço de um alto custo computacional, uma vez que todos os nós à direita do atual devem ser analisados com e sem o elemento que seria incluído no dicionário. Além disso, a queda no desempenho da codificação fica bastante evidente apenas quando usamos blocos de grandes dimensões ou quando o tamanho do dicionário ainda é pequeno [7] [8].

Com o objetivo de contornar esse problema uma versão intermediária do RD Modificado foi proposto em [8]. Conhecido como RD Intermediário, o método consiste em realizar as etapas de casamento de padrões e otimização da árvore de forma simultânea com a atualização do dicionário. Para isso, ao iniciar o processo de otimização, cria-se um dicionário rascunho $\mathcal{D}_R$ como uma cópia do dicionário original. O dicionário rascunho conterá elementos complementares aos do dicionário original. Além disso, criam-se também contadores rascunhos de todas as estatísticas do sistema, que inicialmente também serão cópias dos contadores originais. Esses contadores serão atualizados dinamicamente durante o processo de otimização

armazenando as estatísticas complementares.

Através do percurso de pré-ordem ($\eta_0 \rightarrow \eta_1 \rightarrow \eta_3 \rightarrow \cdots$), semelhante ao que é feito no processo de codificação de entropia que veremos adiante, visitamos os nós da árvore e realizamos o casamento de padrões respeitando as dimensões de cada escala. À medida que as aproximações são feitas nó a nó a partir das buscas nos dicionários rascunho, original e deslocamento que serão detalhados nas seções a seguir, os custos lagrangeanos são computados. Simultaneamente, as estatísticas associadas a cada casamento são atualizadas nos contadores rascunhos como, por exemplo, dos índices das palavras código utilizadas, do tipo de dicionário escolhido e a estatísticas dos *flags* de segmentação. Assumimos que os nós visitados até então são segmentados, ou seja, calculamos e armazenamos o custo de não-segmentação de cada nó da escala p computando o valor de $\lambda R(1|p)$ para uso futuro.

Esse processo é feito até se alcançar os nós filhos que são do tipo folha para finalmente retornar-se ao nó pai. Nesse momento, avaliamos se o custo para a representação do nó pai $J(\eta^m)$ é menor que o custo dos nós filhos $J(\eta^{2m+1})$ e $J(\eta^{2m+2})$. Caso isso seja verdade e os ramos do nó pai sejam podados, os nós filhos são retirados da árvore e, conjuntamente as estatísticas associadas aos nós filhos são restauradas ao estado anterior. O nó pai, portanto, torna-se um nó temporário do tipo folha que contém um índice i ou i_R. Essa condição simula a etapa de codificação, uma vez que as estatísticas também são atualizadas nos contadores rascunhos. No entanto, caso o nó pai não seja segmentado, os nós-filhos tornam-se os nós-folhas e a sub-árvore $\mathcal{T}(\eta^m)$ é mantida. Em seguida o custo do nó pai é atualizado bem como o dicionário rascunho $\mathcal{D}_R$, que receberá novas palavras código a partir da concatenação dos filhos (η^{2m+1} e η^{2m+2}) e suas respectivas transformações de escala. Esse processo é repetido recursivamente até que todos os nós sejam visitados. É importante ressaltar que a decisão de poda dos ramos só é tomada quando retornamos ao nó pai. Por ter uma eficiência superior, optamos neste trabalho por empregar o algoritmo RD Intermediário porque ele apresenta uma maior eficiência.

3.5 Codificação de entropia

O fluxo dos bits gerados pelo MMP é composto pelos dados que representam a árvore de segmentação de forma ordenada. Essa ordenação é feita através do percurso pré-ordem visto no Apêndice B. Esses dados são compostos por:

1. *Flag* de segmentação
    - '0', se o nó for segmentado.
    - '1', se o nó for podado. Neste caso, cada *flag* de poda precede o tipo de dicionário.

2. Índice do tipo de dicionário;

    - ‘0’, Dicionário de deslocamento (ver 3.7).
    - ‘1’, Dicionário original.

3. Se o *flag* do tipo de dicionário for 0, transmitimos o índice i do símbolo usado na aproximação do sub-bloco correspondente.

4. Se o *flag* do tipo de dicionário for 1, transmitimos o índice da partição (contexto) i_c do dicionário original, seguido do índice i do símbolo da palavra código.

Fica evidente, portanto, que a informação da escala p não é transmitida. Ela é obtida de forma implícita ao processo, uma vez que esse dado é uma propriedade dos nós da árvore binária e está disponível no decodificador durante o percurso entre os diferentes níveis. Outro ponto importante é que o *flag* de segmentação ‘1’ não é transmitido e nem contabilizado no custo J quando a aproximação se referir a nós que pertençam à escala ‘0’, cuja dimensão é 1x1, já que estes não podem ser repartidos. Sendo assim, uma representação da árvore segmentada da Figura 3.5 pode ser ordenada como:

$$0 \quad 1 \quad i_{D_d} \quad i_1 \quad 0 \quad 1 \quad i_{D_o} \quad i_c \quad i_5 \quad 0 \quad i_{D_d} \quad i_{13} \quad i_{D_o} \quad i_c \quad i_{14}.$$

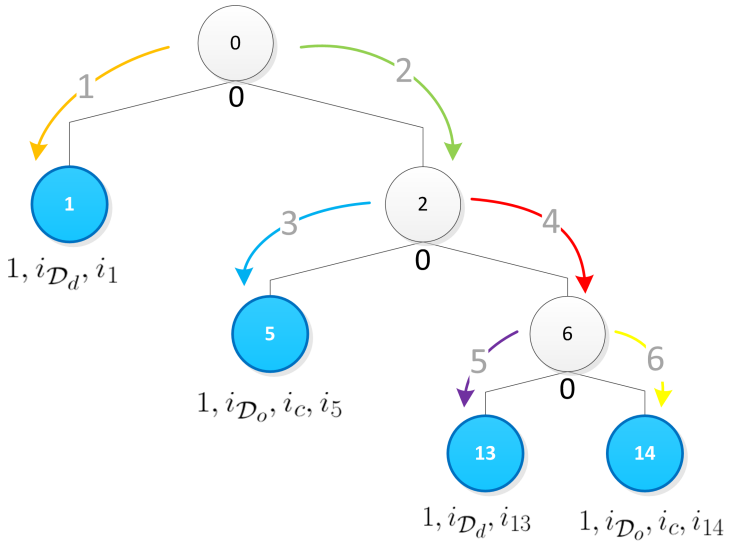


Figura 3.5: Exemplo da árvore segmentada com os flags e índices dos dicionários, contextos e palavras código ordenados de acordo com a bitstream MMP.

A alocação final de bits é feita a partir do codificador aritmético adaptativo [27] nos dados MMP, cujas distribuições de probabilidade dependem dos contextos de cada símbolo. Originalmente, nas primeiras implementações do MMP, apenas as informações de escalas eram utilizadas como contexto para o codificador aritmético.

Por exemplo, é natural esperar que as distribuições de probabilidade do *flag* de segmentação sejam diferentes entre escalas e, portanto, o codificador aritmético pode se beneficiar uma vez que permitimos que cada escala tenha sua distribuição de probabilidade. De fato, essa técnica melhora a eficiência do codificador aritmético, pois reduz a entropia do símbolo $H(i|p) < H(i)$ condicionando as probabilidades dos símbolos sem que nenhum dado adicional seja transmitido. No entanto, novos estudos comprovaram que uma segmentação eficiente dentro da própria escala do dicionário melhora o desempenho do sistema, mesmo ao preço da transmissão de novos símbolos referentes às partições. Essas partições são vistas como sub-contextos para o codificador aritmético. Desta forma, cada escala do dicionário possui sub-contextos que são definidos pela escala da palavra de origem [3].

3.6 O Dicionário Original

No MMP, o dicionário original é uma das estruturas mais importantes do algoritmo. Ele armazena todas as palavras-código usadas na etapa de casamento de padrões e o processo de atualização deve ser executado igualmente tanto no codificador quanto no decodificador. Os parâmetros de inicialização do dicionário devem ser escolhidos adequadamente de acordo com a distribuição de probabilidade do sinal fonte e da faixa dinâmica de suas amostras. Nesse momento, leva-se em consideração que a etapa de predição permite que os padrões existentes no dicionário sejam elementos com amostras residuais, e de maneira geral, com baixa energia. A predição envolvida na codificação será vista em detalhes no Capítulo 4.

De fato, os ganhos observados com o uso de preditor ganharam destaque e relevância nas pesquisas do MMP [9]. Mesmo assim, algumas características prejudiciais ao desempenho do sistema durante o processo de atualização ainda deveriam ser contornadas, como, por exemplo, a inclusão de elementos desnecessários, o que será visto na seção a seguir. Uma representação do dicionário é feita na Figura 3.6, enfatizando que existe apenas um dicionário por escala.

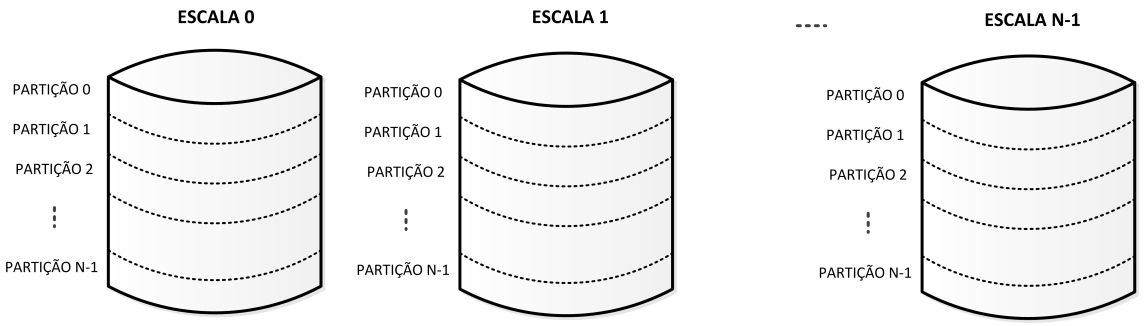


Figura 3.6: Representação das diferentes escalas do dicionário original e sua estrutura de sub contextos (partições) destacada pelas linhas pontilhadas.

3.6.1 Atualização do dicionário

É intuitivo desejarmos a maior velocidade no processo de aprendizagem do algoritmo, já que isso permitiria aproximações com distorções ainda mais baixas. Isto significa incluir o maior número de palavras-código possível. Na prática, entretanto, precisamos que este aprendizado também seja eficiente, uma vez que o custo lagrangeano é composto pela estimação de taxa de bits e novos elementos incluídos no dicionário aumentam a entropia dos símbolos. Algumas técnicas descritas abaixo foram propostas para contribuir com a velocidade de aprendizado de forma controlada [7] [3], dentre elas merecem destaque:

1. O conceito da hiper-esfera para evitar a similaridade de palavras no dicionário;

2. Limites superior e inferior de escala para inclusões de palavras transformadas;

Durante a atualização, percorremos a árvore binária otimizada no sentido pós-ordem e para cada nó folha submetemos a palavra associada às etapas descritas a seguir de transformação, equalização e controle de redundância. Ao acessarmos o nó pai, concatenamos as palavras dos nós filhos e resubmetemos a nova palavra formada às mesmas etapas. Este processo se repete de forma recursiva até que todos os nós tenham sido acessados.

3.6.2 Transformação de Escala

A transformação de escala, $\mathbb{T} : \mathbb{R}^{N_0} \to \mathbb{R}^N$, realiza contrações e expansões dos vetores utilizados em sub-blocos passados. Elas são operações simples de mudança de taxas onde, basicamente, aplicam-se a decimação e a interpolacão linear conforme descrito em [28].

Para a dilatação, onde $N > N_0$, mudamos o tamanho do vetor para $N_0 N$ usando um interpolador linear como filtro. Depois aplicamos a decimação por N_0, cujo procedimento é descrito por:

$$\begin{aligned} m_n^0 &= \left\lfloor \frac{n\,(N_0 - 1)}{N} \right\rfloor \\ m_n^1 &= \begin{cases} m_n^0 + 1, & m_n^0 < N_0 - 1, \\ m_n^0, & m_n^0 = N_0 - 1, \end{cases} \\ \alpha_n &= n\,(N_0 - 1) - N m_n^0, \\ C_n &= \left\lfloor \alpha_n \frac{\left(C_{m_n^1} - C_{m_n^0}\right)}{N} \right\rfloor + C_{m_n^0}, \\ n &= 0, 1, \ldots, N-1, \end{aligned} \tag{3.19}$$

onde C_n é a palavra expandida.

Quando $N < N_0$, aplicamos a contração. Novamente, escalonamos o vetor para $N_0 N$ usando um interpolador linear. Depois aplicamos um filtro de média com tamanho $N_0 + 1$ para só depois decimarmos pelo fator N_0. O procedimento de contração pode ser descrito por

$$\begin{aligned} m_{n,k}^0 &= \left\lfloor \frac{n\left(N_0 - 1\right) + k}{N} \right\rfloor \\ m_{n,k}^1 &= \begin{cases} m_{n,k}^0 + 1, & m_{n,k}^0 < N_0 - 1, \\ m_{n,k}^0, & m_{n,k}^0 = N_0 - 1, \end{cases} \\ \alpha_n &= n\left(N_0 - 1\right) + k - N m_n^0, \\ C_n &= C_{m_{n,k}^0} + \frac{1}{N_0 + 1} \sum_{k=0}^{N_0} \left\lfloor \alpha_{n,k} \frac{\left(C_{m_{n,k}^1} - C_{m_{n,k}^0}\right)}{N} \right\rfloor, \\ n &= 0, 1, \ldots, N - 1. \end{aligned} \tag{3.20}$$

Após o processo de otimização, cada nó folha é concatenado com o seu nó irmão. Esse vetor resultante é submetido ao processo de transformação de escala, incluindo o nó raiz, cujas amostras representam inteiramente a aproximação do bloco $\hat{X}$. Todos esses vetores, contraídos e expandidos, portanto, são incluídos no dicionário caso não haja nenhuma palavra código igual de mesma escala. Isso significa que o dicionário contém padrões referentes a blocos, sub-blocos e suas versões contraídas e dilatadas. Um detalhe importante de implementação é que todas as escalas que são computadas também são mantidas para evitar o custo computacional de transformação durante o processo de busca seguinte.

Além disso, para explorar a similaridade das normas entre as palavras originais (C^{p_o}) do dicionário e suas versões escaladas usadas na aproximação dos nós, aplicamos uma normalização nas palavras transformadas. Isto garante regularidade nas normas das novas palavras C_α^p incluídas no dicionário

$$C_\alpha^p = s_\alpha^{p_o,p} \mathbb{T}_{p_o}^p \left(C^{p_o}\right), \tag{3.21}$$

onde

$$s_{\alpha=1}^{p_o,p} = \frac{|C^{p_o}|_{\alpha=1}}{|C^p|_{\alpha=1}} \tag{3.22}$$

3.6.3 Controle de Redundância

O controle de redundância implementado inicialmente neste trabalho se baseia na quantização das amostras de cada palavra código utilizada para aproximar os sub-blocos da árvore binária e suas versões escaladas. Caso a palavra código resultante

$\hat{C}_{\alpha}^{p}$, após ter passado pela transformação de escala, equalização da norma e quantização escalar, não exista no dicionário (em nenhuma partição), ela será, então, incluída na partição referente à sua escala de origem p_o. O procedimento de atualização é ilustrado na Figura 3.7:

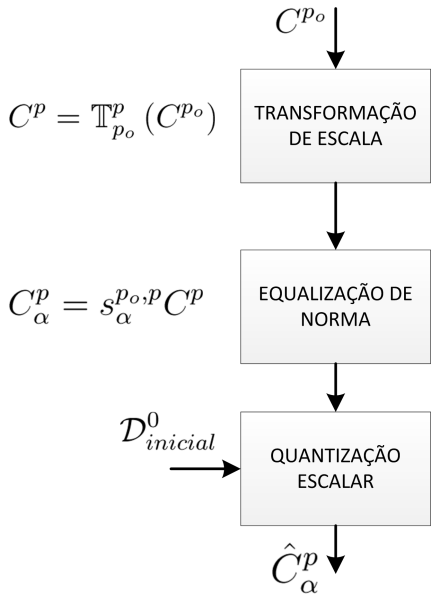


Figura 3.7: Processo de atualização do dicionário.

3.7 O Dicionário de Deslocamento

O dicionário de deslocamento é composto pelas palavras utilizadas na codificação de amostras anteriores ($[\hat{s}(n-L) \ \cdots \ \hat{s}(n-2) \ \hat{s}(n-1)]$), o que significa que em trechos periódicos como nos quadros vozeados e de silêncio, o uso deste dicionário auxiliar pode favorecer o desempenho do MMP. Define-se um valor máximo L para o deslocamento e, quando ocorre o melhor casamento do bloco no índice $i = \delta$, codifica-se o valor deslocado. Nesta implementação, utilizamos deslocamentos inteiros δ com passo $\kappa = 1$, conforme ilustrado na Figura 3.9.

3.8 Conclusão

Neste capítulo, descrevemos a estrutura básica do MMP e, com essa nova organização, conseguimos identificar o núcleo do algoritmo e as estruturas de dados que o compõem. Ainda neste capítulo, destacamos cada ferramenta e ilustramos o processo de compressão através de diagramas em blocos. No capítulo seguinte, introduzimos o conceito de predição linear que tem sido empregado em alguns codificadores de voz nos últimos anos e, principalmente, como adaptamos essa nova técnica ao processo de codificação através do MMP.

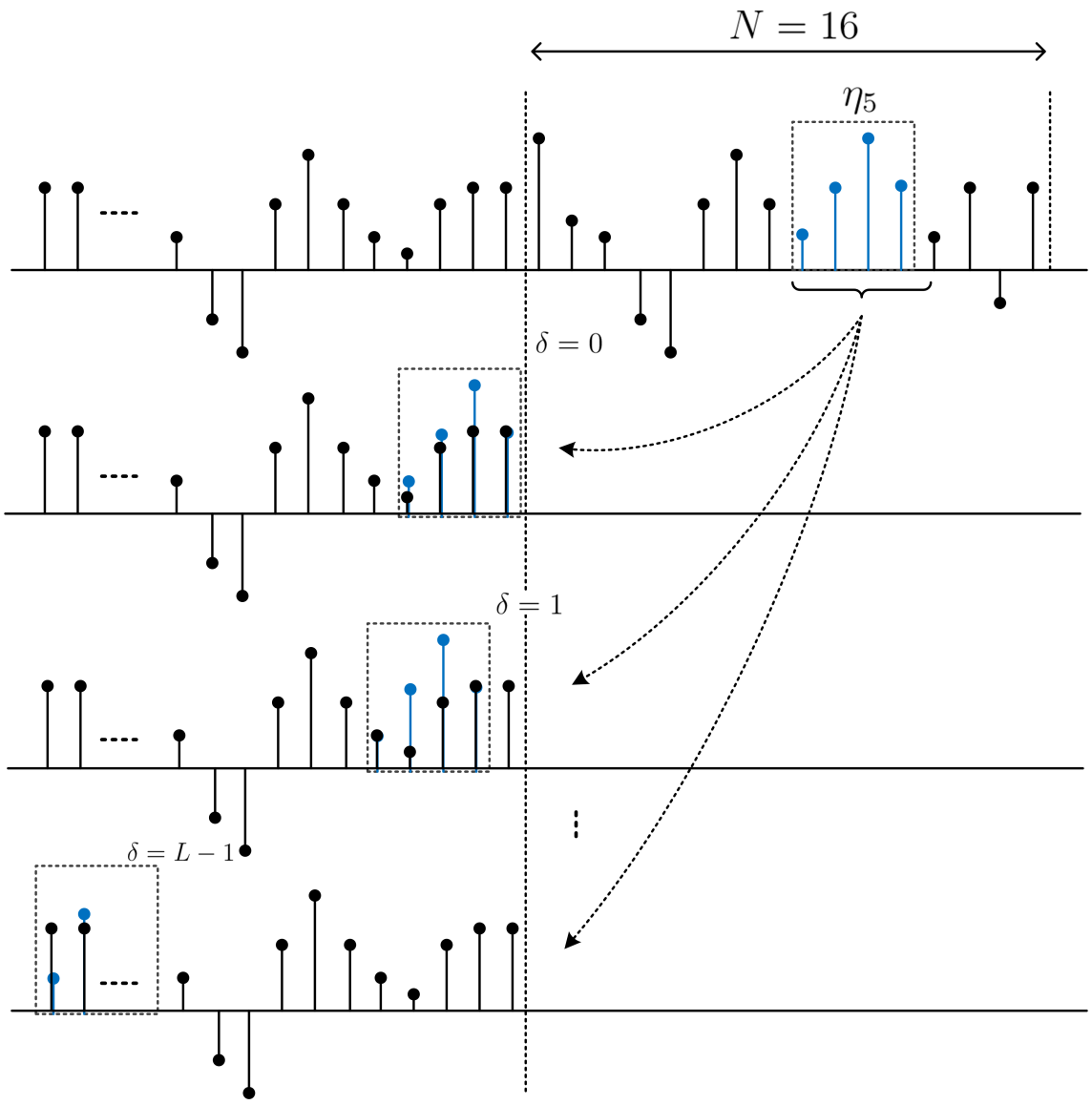


Figura 3.8: Procedimento do dicionário de deslocamento.

Capítulo 4

Predição Linear no MMP

A técnica de predição linear (LP, do inglês *Linear Prediction*) na codificação de sinais de voz tem sido empregada há bastante tempo. Este capítulo se propõe a fazer uma revisão breve do assunto e descrever a formulação para o processamento em blocos que utilizamos na codificação MMP. Essa contextualização é importante na comparação entre diferentes algoritmos de predição propostos nesse trabalho.

4.1 Descrição do Preditor

A ideia por trás da predição linear é estimar amostras futuras a partir de combinações lineares de amostras passadas. Matematicamente, podemos representar esse processo pela Equação (4.1).

$$\tilde{s}(n) = a_1 s(n-1) + a_2 s(n-2) + \cdots + a_M s(n-M) = \sum_{m=1}^{M} a_m s(n-m), \qquad (4.1)$$

onde $a_i, i = 1, 2, ..., M$ são conhecidos como coeficientes LP e M define a ordem do modelo LP. Comparando a estimação LP com o sinal de origem, encontramos o erro de estimação

$$e(n) = s(n) - \tilde{s}(n) = s(n) - \sum_{m=1}^{M} a_m s(n-m), \qquad (4.2)$$

e o erro médio quadrático (MSE) é dado por

$$\xi = E\left[e^2(n)\right]. \qquad (4.3)$$

Do ponto de vista de sistemas lineares, a predição pode ser interpretada como um processo de filtragem (AR), cujos coeficientes LP compõem o filtro $A(z)$ que modela do trato vocal humano:

$$A(z) = \frac{1}{1 - a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{M-1} z^{-M+1}} \tag{4.4}$$

Uma forma de encontrar os coeficientes LP que minimizam o erro médio quadrático é tomarmos a derivada parcial de (4.3) com respeito aos coeficientes a_m e igualarmos a zero. Desenvolvendo essa ideia, obtemos:

$$\begin{aligned}
\frac{\partial \xi}{\partial a_m} &= \frac{\partial E\left[e^2(n)\right]}{\partial a_m} \\
&= E\left[\frac{\partial e^2(n)}{\partial a_m}\right] \\
&= E\left[2e(n)\frac{\partial e(n)}{\partial a_m}\right] \\
&= E\left[2e(n)\frac{\partial\left(s(n) - \sum_{m=1}^{M} a(m)s(n-m)\right)}{\partial a_m}\right] \\
&= -2E\left[e(n)s(n-m)\right]
\end{aligned} \tag{4.5}$$

Substituindo $e(n)$ por (4.2) na equação (4.5), temos que:

$$\begin{aligned}
\frac{\partial \xi}{\partial a_m} &= -2E\left[\left(s(n) - \sum_{j=1}^{M} a(j)s(n-j)\right)s(n-m)\right] \\
&= -2\left\{E\left[s(n)s(n-m)\right] - E\left[\sum_{j=1}^{M} a(j)s(n-j)s(n-m)\right]\right\} \\
&= -2R_{ss}(m) + 2\sum_{j=1}^{M} a(j)R_{ss}(m-j)
\end{aligned} \tag{4.6}$$

onde

$$R_{ss}(k) = \frac{1}{M}\sum_{n=n_0}^{n_0+M-1} \hat{s}(n)\hat{s}(n-k) \tag{4.7}$$

e assumindo que o processo $\{S\}$ é WSS.

Portanto, para determinar os valores do vetor $\mathbf{a} = [a_1 \quad a_2 \quad \cdots \quad a_M]^T$ a partir do gradiente $\nabla_{\mathbf{a}}\xi$ com menor MSE do processo $\{S\}$, igualamos (4.6) a zero e obtemos um conjunto de M equações lineares:

$$\begin{bmatrix} R_{ss}(0) & R_{ss}(1) & \cdots & R_{ss}(M-1) \\ R_{ss}(1) & R_{ss}(0) & \cdots & R_{ss}(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{ss}(M-1) & R_{ss}(M-2) & \cdots & R_{ss}(0) \end{bmatrix} \begin{bmatrix} a1 \\ a2 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} R_{ss}(1) \\ R_{ss}(2) \\ \vdots \\ R_{ss}(M) \end{bmatrix}, \tag{4.8}$$

também conhecidas com equações de Wiener-Hopf, que podem ser representadas de forma mais compacta como:

$$\mathbf{R}_S\mathbf{a} = \mathbf{p}_S, \tag{4.9}$$

onde $\mathbf{R}_S$ é a matriz de autocorrelação do processo $\{S\}$.

A solução que procuramos para encontrar os coeficientes LP é da forma:

$$\mathbf{a} = \mathbf{R}_S^{-1}\mathbf{p}_S \tag{4.10}$$

Existem duas formas clássicas na estimação dos coeficientes $\mathbf{a}$ utilizados na literatura, cujos métodos são conhecidos como autocorrelação e método da autocovariância (este último não será abordado neste trabalho). No método da autocorrelação, assume-se que a sequência s pertence a um processo estacionário e, além disso, considera-se que as amostras são nulas fora da janela de treinamento. Por se tratar de uma matriz com forma *Toeplitz*, onde além de $\mathbf{R}$ ser simétrica cada diagonal consiste no mesmo elemento, algumas soluções recursivas e robustas podem ser aplicadas neste caso como, por exemplo, o algoritmo de Levinson-Durbin ([29] - [30]).

O diagrama em blocos do MMP com predição linear pode ser visto na Figura 4.1.

A predição funciona como uma inferência estatística a partir de realizações passadas que constituem um conjunto de treinamento. Na prática, este conjunto é composto por amostras decodificadas e, como essas realizações estão disponíveis tanto no codificador quanto no decodificador, os coeficientes são computados em ambos os processos sem fazer uso de transmissões extras de dados. Isto significa que os coeficientes LP não são transmitidos, mas sim calculados de forma adaptativa. Neste caso, assumimos que as amostras do conjunto de treinamento e do bloco estimado são realizações de um mesmo processo estocástico. Consideramos, portanto, que as amostras estimadas se baseiam na mesma lei de formação que as amostras de uma vizinhança causal. Neste momento é necessário definirmos o conceito de "janela de treinamento" composta pelas L últimas amostras reconstruídas, conforme ilustrado pela Figura 4.2. Em outras palavras, a vizinhança causal é formada pelas amostras reconstruídas $\hat{s}(n-i), i = 1, 2, ..., L$.

Os estudos [6] e [9] investigaram a técnica dos mínimos quadrados para estimação dos coeficientes do preditor na codificação de imagem e voz, respectivamente e, observaram que a eficiência do MMP aumenta quando a codificação é aplicada ao sinal resíduo $r(n) = s(n) - \tilde{s}(n)$. Nesta abordagem, assumimos que esses coeficientes LP $\hat{\mathbf{a}}$ modelam o processo do sinal reconstruído $\hat{s}(n)$ e, portanto, se aproximam do modelo LP do sinal original ($\hat{\mathbf{a}} \approx \mathbf{a}$). Isto significa que essa aproximação será tão melhor quanto menor for o erro de quantização do sistema $\varepsilon(n)$, pois de forma

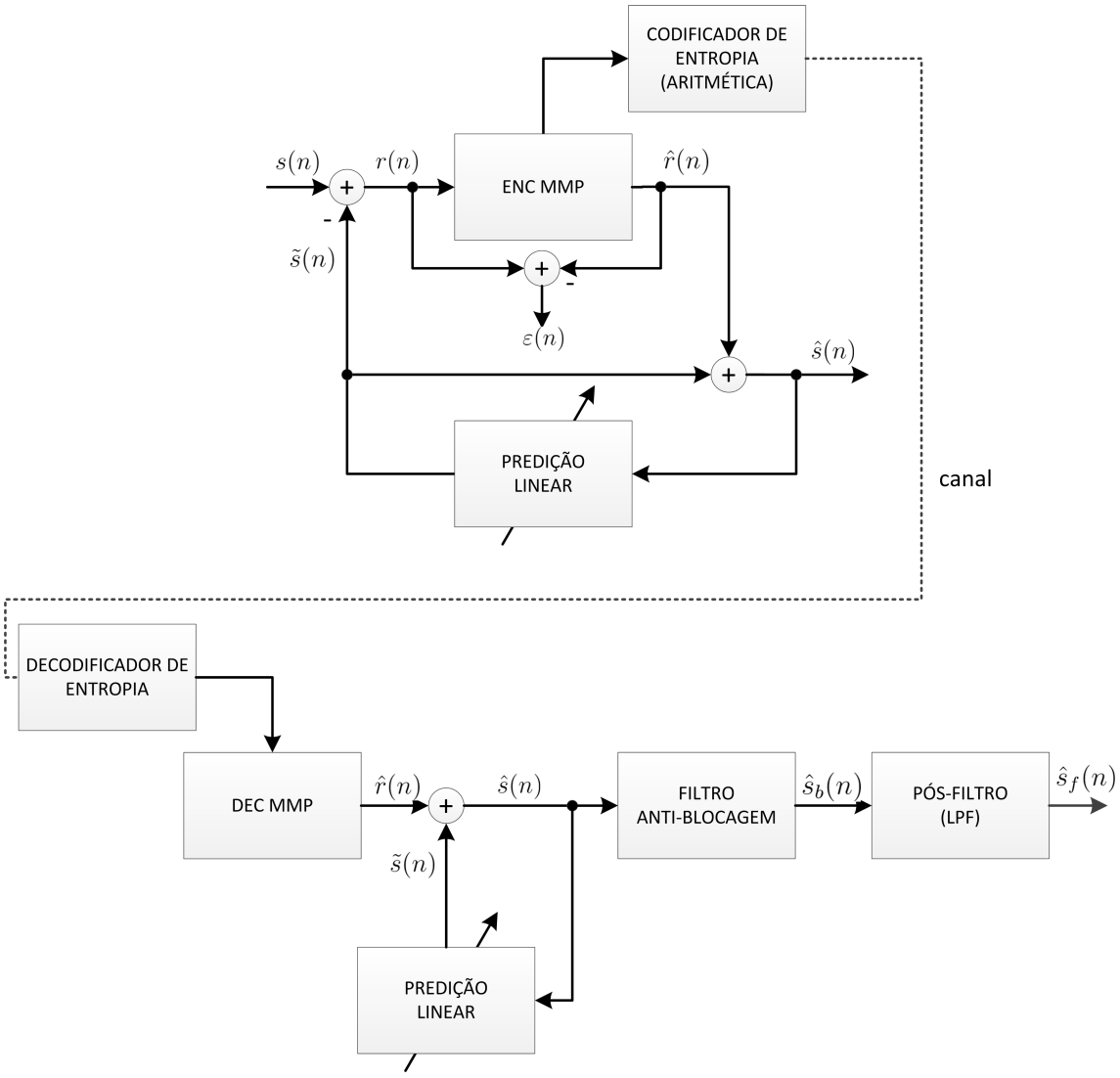


Figura 4.1: Diagrama em blocos do MMP com predição linear.

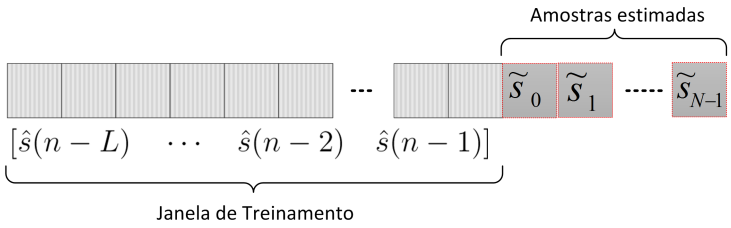


Figura 4.2: As amostras reconstruídas pertencem à janela de treinamento.

intuitiva temos:

$$\begin{aligned}\varepsilon(n) &= r(n) - \hat{r}(n) \\ &= s(n) - \tilde{s}(n) - \hat{r}(n) \\ &= s(n) - \tilde{s}(n) - \hat{s}(n) + \tilde{s}(n) \\ &= s(n) - \hat{s}(n),\end{aligned} \tag{4.11}$$

onde $\varepsilon(n)$ é o erro de quantização, cuja energia média $E[\varepsilon^2(n)]$ tende a ser minimizada devido ao critério de desempenho adotado no MMP (MSE) na aproximação dos resíduos. A Figura 4.3 representa uma simplificação do modelo discutido.

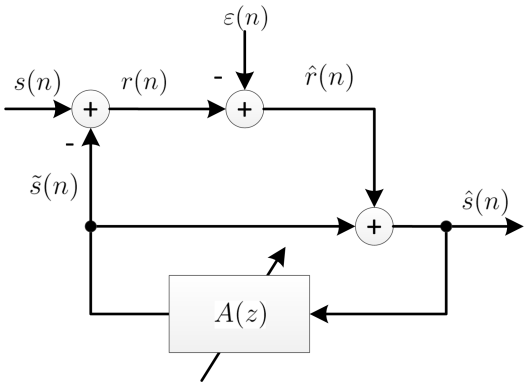


Figura 4.3: Modelo simplificado do processo de codificação MMP com predição linear.

A predição linear tem o objetivo de reduzir a correlação entre as amostras, a faixa dinâmica do sinal e, consequentemente, a energia da fonte a ser codificada. Portanto, a variância do sinal resultante, ou energia do sinal erro, será tão menor do que o sinal original quanto mais eficiente for a predição. Isso permite que o dicionário contenha palavras que representem apenas resíduos, cuja função de distribuição de probabilidade seja mais parecida às estatísticas do sinal erro [3]. Com a faixa dinâmica menor, as amostras residuais têm probabilidade de ocorrência mais concentrada resultando numa entropia menor o que, de maneira geral, faz com que os padrões existentes no dicionário sejam usados mais eficientemente.

Nos estudos mencionados, a distribuição de probabilidade dos resíduos pode ser modelada como uma DGG (Distribuição Gaussiana Generalizada) com base na função (4.12), cujo histograma está representado na Figura 4.4 e é descrito por:

$$p(x) = \left[\frac{\alpha\eta\left(\alpha,\beta\right)}{2\Gamma\left(\frac{1}{\alpha}\right)}\right] e^{-(\eta(\alpha,\beta)|x|)^{\alpha}}, \tag{4.12}$$

onde

$$\eta\left(\alpha,\beta\right) = \beta^{-1}\left[\frac{\Gamma\left(\frac{3}{\alpha}\right)}{\Gamma\left(\frac{1}{\alpha}\right)}\right]^{\frac{1}{2}}, \tag{4.13}$$

e $\Gamma(.)$ é a função de gamma. Neste modelo, o valor de α define a taxa de decaimento da distribuição e β é o desvio padrão correspondente. Neste trabalho, os parâmetros utilizados foram $\alpha = 0,43$ e $\beta = 1,1031 \times 10^3$.

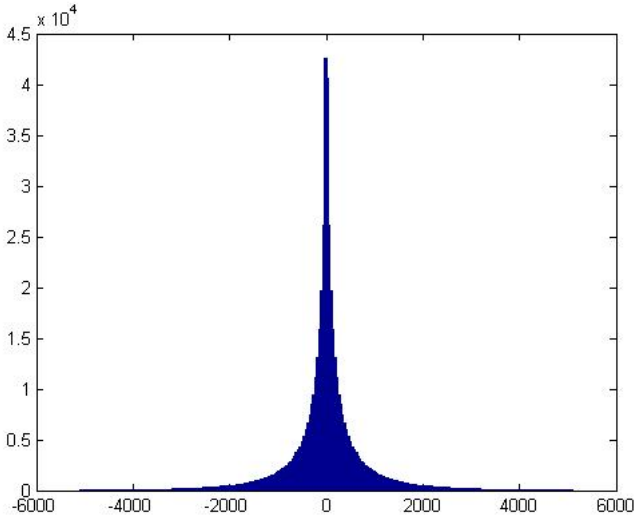


Figura 4.4: Histograma do processo definido pela DGG com os parâmetros α e β definidos acima.

Os níveis do dicionário inicial foram gerados a partir do script Matlab definido no Apêndice D.1. Através das raias na Figura 4.5 é possível observar a distribuição não uniforme dos valores encontrados. Estes níveis são submetidos ao mesmo processo de atualização do dicionário, visto no capítulo anterior. O destaque aqui é que após a etapa de dilatação e se a inclusão for permitida, a palavra transformada é incluída na partição 0 de escala relacionada. Neste trabalho, o tamanho do dicionário inicial é 256.

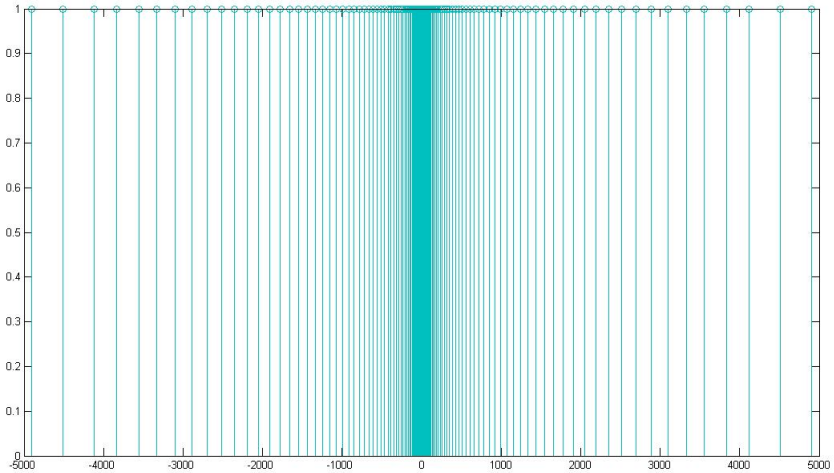


Figura 4.5: Distribuição não uniforme dos níveis do dicionário inicial.

4.2 Predição em Blocos

Para se adequar ao processamento em blocos do MMP, a predição linear precisa de uma formulação própria. Tendo um bloco de interesse a ser codificado pelo MMP $\mathbf{s}_k = [s_k(0)\, s_k(1)\, \cdots\, s_k(N-1)]^T$, calculam-se os coeficientes LP $\hat{\mathbf{a}} = [\hat{a}_1\, \hat{a}_2\, \cdots\, \hat{a}_M]^T$ apenas uma vez a partir de uma janela de treinamento fixa composta por amostras decodificadas $\hat{s}(n-i), i = 1, 2, ..., L$. Essas amostras são utilizadas para gerar a matriz de autocorrelação $\mathbf{R}$ e calcular o vetor de correlação cruzada $\mathbf{p}$. Em seguida fazemos uso do algoritmo de Levinson-Durbin para encontrar os coeficientes LP.

A estimativa é feita num processo amostra-a-amostra e o vetor $\hat{\mathbf{a}}$ não é atualizado até que o bloco termine. Ou seja, as amostras estimadas $\tilde{s}(n+i), i = 0, 1, 2, ..., N-1$ são utilizadas no cálculo das amostras seguintes:

$$\tilde{s}(n+k) = \sum_{j=1}^{k} \hat{a}(j)\tilde{s}(n+k-j) + \sum_{i=k+1}^{M} \hat{a}(i)\hat{s}(n+k-i) \qquad (4.14)$$

$$\text{para, } k = 0, 1, 2, ..., N-1.$$

Chamaremos este algoritmo de *Least Squares A* (LS_A), representado pela Figura 4.6.

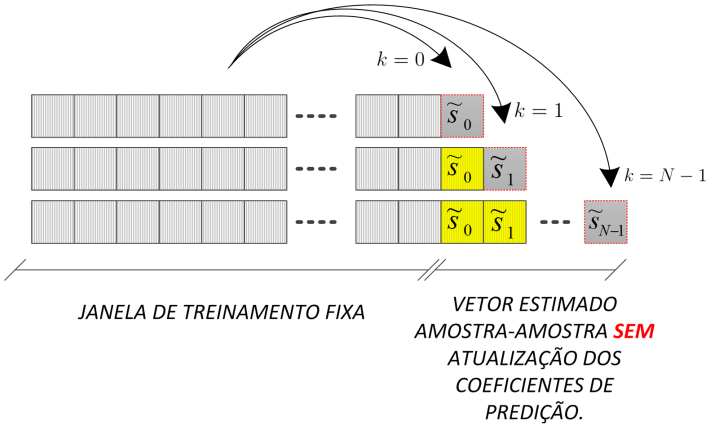


Figura 4.6: Algoritmo de predição em blocos *Least Squares A* proposto baseado na técnica dos mínimos quadrados. Os blocos hachurados representam as amostras reconstruídas e os blocos em amarelo representam as amostras estimadas utilizadas no processo de estimação da amostra seguinte.

É importante mencionar que antes do cálculo da matriz de autocorrelação, aplicamos a janela de Hamming às amostras do conjunto de treinamento.

4.3 Conclusões

Neste capítulo apresentamos o modelo de predição linear baseado na técnica dos mínimos quadrados que chamamos de LS_A, a formulação matemática para o pro-

cessamento em blocos, as vantagens de codificação do sinal resíduo e o procedimento de estimação dos coeficientes LP. No capítulo seguinte veremos as etapas de filtragem no decodificador que melhoram a performance do MMP. Apresentaremos também os resultados experimentais do que conhecemos como o estado da arte do MMP Voz incorporando todas as ferramentas discutidas.

Capítulo 5

Complementos da Estrutura Básica do MMP Voz

Neste capítulo vamos apresentar as ferramentas adicionais de filtragem na etapa de decodificação nas Seções 5.1 e 5.2. Com elas, complementamos a estrutura básica do algoritmo e atingimos e o estado da arte do MMP Voz. Em seguida, apresentaremos os resultados experimentais com os parâmetros que produziram a melhor nota PESQ-MOS. Novas ferramentas propostas derivam de algumas análises que serão feitas ao longo do texto e, por fim, destacamos os ganhos de desempenho obtidos com as contribuições inseridas no MMP Voz.

5.1 Filtro *Anti-Blocking*

As ferramentas descritas nos Capítulos 3 e 4 compõem a essência do MMP com predição linear. No entanto, como a qualidade perceptual em sinais de voz é bastante afetada por descontinuidades artificialmente criadas no codificador, adicionamos um estágio de filtragem para reduzir o efeito de blocos. Essas descontinuidades são consequência comum da codificação independente dos blocos que tendem a ser mais evidentes em taxas mais baixas. Um forma de combater o efeito *blocking* é considerar uma interdependência entre blocos adjacentes no critério de desempenho durante etapa de casamento de padrões. Em nosso caso, estudamos uma forma alternativa de minimizar as descontinuidades: consideramos a sobreposição de segmentos vizinhos, minimizando as diferenças abruptas que eventualmente possam existir. Desta forma, evitamos que segmentos adjacentes sejam apenas concatenados. Aplicamos esta técnica numa etapa de filtragem do decodificador, chamado de filtro *anti-blocking*, conforme descrito na Seção 7.2 de [3]. O filtro é definido a partir de uma base gaussiana

$$g^{p_k}(n) = e^{-\frac{\left(n-\frac{L-1}{2}\right)^2}{2(\alpha L)^2}}, n = 0, 1, 2, ..., \mathcal{G}^{p_k} - 1, \tag{5.1}$$

onde $\mathcal{G}^{p_k}$ representa o tamanho da resposta ao impulso (IR), definido por

$$\mathcal{G}^{p_k} = \mathcal{L}[g^{p_k}]. \tag{5.2}$$

Acima, $\mathcal{G}^{p_k}$ é ajustado de acordo com o tamanho do segmento codificado, digamos 2^{p_k}. A variância σ^2 do filtro gaussino é determinada por $\sigma^2 = (\alpha L)^2$, onde α controla o decaimento da função (5.1).

Quando α tende a zero, a função base torna-se um impulso fazendo que o efeito anti-blocagem não atue. Em nossas simulações diferentes valores de α foram testados no processo de decodificação e o valor ótimo α_o foi encontrado para cada frase quando alcançamos a maior nota PESQ-MOS.

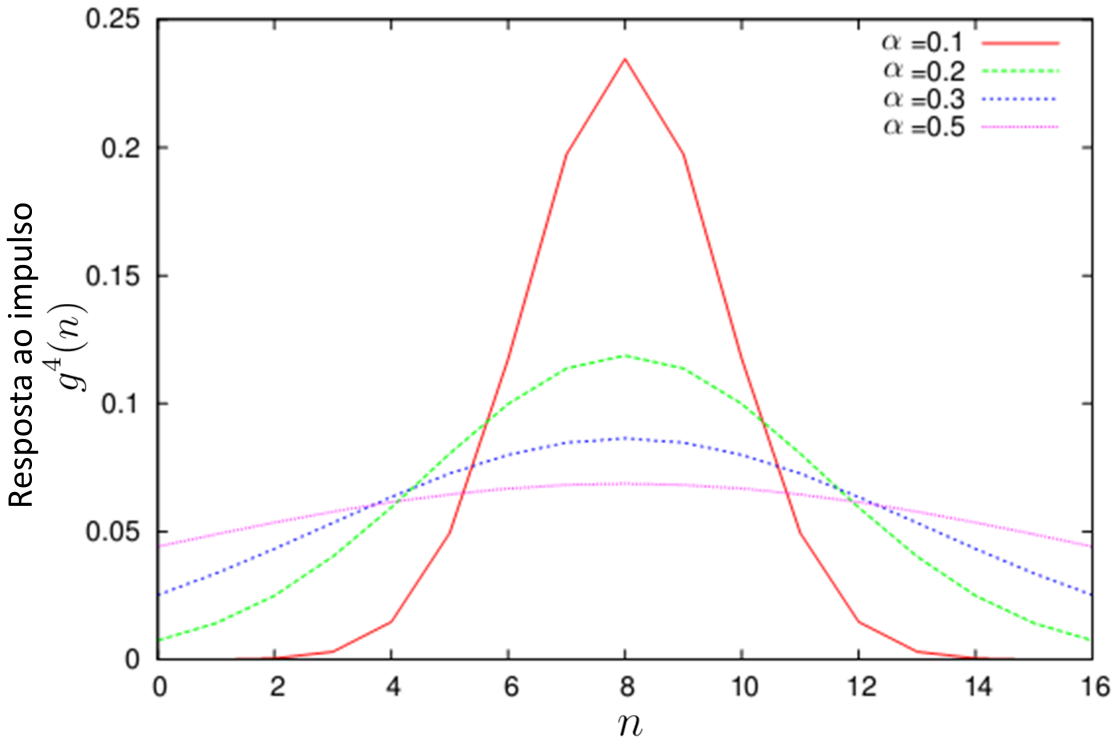


Figura 5.1: Exemplo da base gaussiana do filtro que minimiza o efeito blocos para diferentes valores de α. Neste caso, o tamanho do filtro é definido para função $g^4(n)$ com $\mathcal{G} = 17$. Figura reproduzida de [3]

.

Em implementações anteriores, a dimensão do filtro era determinada apenas com base no tamanho do segmento atual, o que causava artefatos indesejáveis na amostra filtrada, uma vez que a base do filtro incluía amostras de segmentos vizinhos não imediatos. Para evitar tal efeito, controlamos a dimensão do filtro de acordo com o segmento de menor tamanho que contribui para o efeito de borda. Isso significa

que escolhemos o filtro representado pela linha pontilhada da Figura 5.2 ao invés da linha contínua utilizada em versões anteriores. Neste trabalho, inicializamos as funções base g^{p_k} para filtrar segmentos elementares de qualquer dimensão possível no conjunto definido pelo tamanho máximo do bloco N, suas versões contraídas $(N/2, N/4, ..., 1)$ e deslocadas. Na verdade, ao longo da codificação, armazenamos o tamanho do segmento elementar (original) que deu origem à palavra-código selecionada. Se essa palavra foi criada através da concatenação de nós adjacentes, guardamos também o tamanho de cada componente elementar que constitui o novo vetor. O tamanho do componente é definido pela dimensão original daquele subbloco. Isso significa que os segmentos deslocados podem assumir tamanhos que não são potência de 2, mas garantimos que após a geração das bases gaussianas temos funções com dimensão ímpar. Quando o tamanho do componente elementar do vetor for ímpar, mantemos a base gaussiana com a mesma dimensão. Do contrário, quando for potência de 2, definimos

$$\mathcal{G}_{l_i} = 2^{l_i} + 1, \quad l_i = 0, 1, 2, ..., \log_2(N) \tag{5.3}$$

e as amostras filtradas $\hat{s}_b(n)$, pelo exemplo da figura 5.2, são calculadas como

$$\hat{s}_b(n - 2^{(l_1-1)}) = \sum_{i=0}^{\mathcal{G}^{l_1}} g^{l_1}(i)\hat{s}(n - 2^{(l_1-1)} - 1 + \mathcal{G}^{l_1} - i) \tag{5.4}$$

5.2 O Pós-Filtro

Tabela 5.1: Tabela com os coeficientes do filtro FIR passa-baixas usado na etapa de pós-filtragem.

Coeficiente FIR	Valor
h(0)	0,0444
h(1)	0,0463
h(2)	0,0479
h(3)	0,0491
h(4)	0,0498
h(5)	0,0500
h(6)	0,0498
h(7)	0,0491
h(8)	0,0479
h(9)	0,0463
h(10)	0,0444

Na saída do decodificador, o sinal de voz ainda pode apresentar componentes

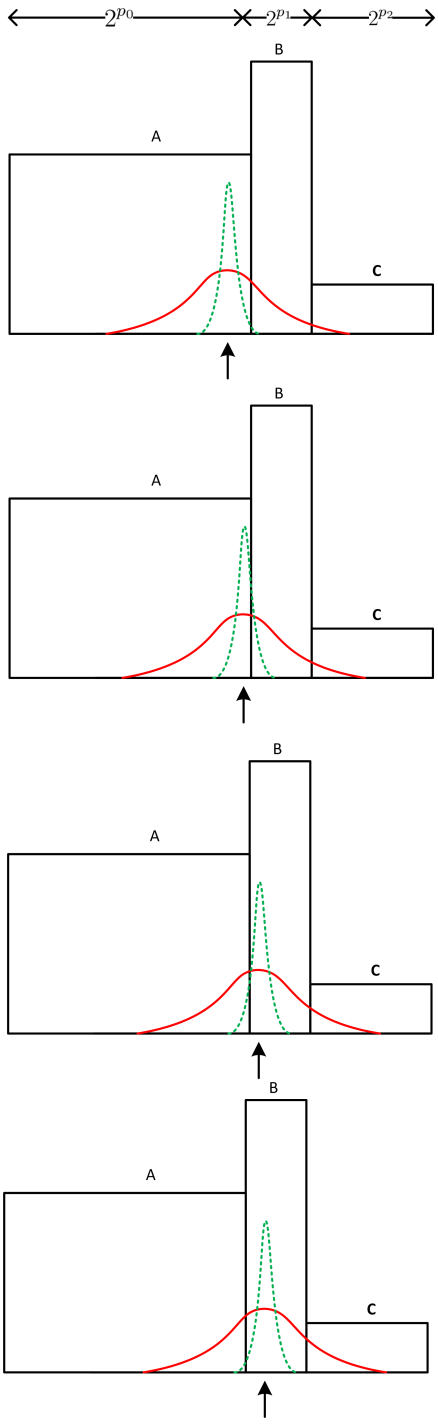


Figura 5.2: Quando filtramos o segmento A, desejamos minimizar a descontinuidade AB sem que haja influência das amostras do bloco C. Para isso, determinamos que o tamanho do filtro *anti-blocking* seja determinado da forma $\mathcal{G} = min[2^{p_0}, 2^{p_1}] + 1$. Neste exemplo, $\mathcal{L}[A] = 16$, $\mathcal{L}[B] = 4$. A seta indica onde o filtro está centrado a cada iteração. A linha pontilhada representa a escolha correta do filtro, enquanto que a linha contínua representa a escolha errada, cujo filtro leva em consideração componentes de sub-blocos vizinhos não-imediatos (segmento C)

de alta frequência, correspondentes a artefatos da codificação. Para minimizar o efeito dessas componentes espectrais, inserimos o filtro FIR passa-baixas de melhor desempenho [6], cujos coeficientes estão definidos na Tabela 5.1. A Figura 5.3 ilustra a resposta em frequência (magnitude e fase) do filtro selecionado.

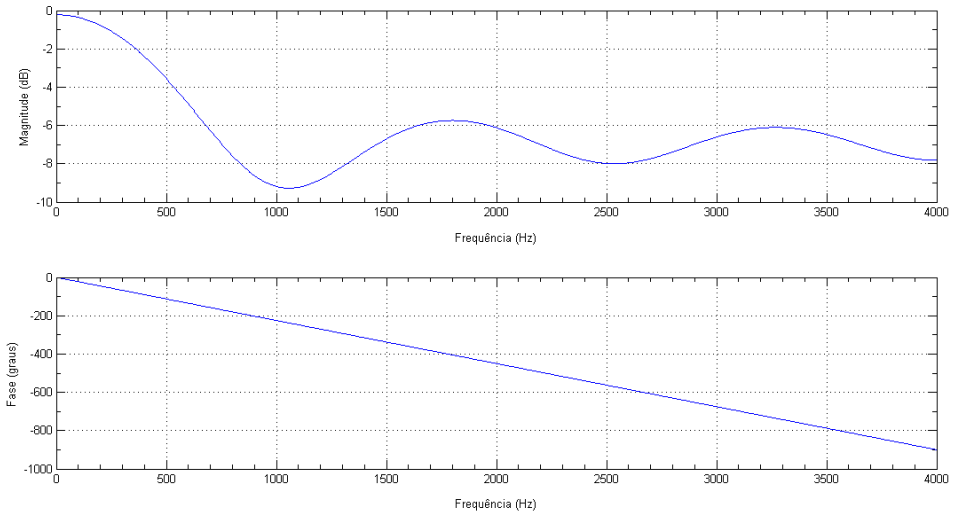


Figura 5.3: Resposta em frequência (magnitude e fase) do filtro FIR passa baixas aplicado na saída do decodificador para remover componentes espectrais de alta frequência indesejáveis.

5.3 Resultados Experimentais

As Figuras 5.4 e 5.5 contém os diagramas do estado da arte do codificador e do decodificador MMP, respectivamente. A performance do MMP Voz será sempre avaliada a partir de duas métricas: a relação sinal-ruído (SNR) e nota PESQ-MOS, cujos valores médios representarão todo o banco de frases do Apêndice A. Mais ainda, apresentaremos os valores de SNR e PESQ-MOS para três sinais diferentes com o objetivo de registrar e analisar as contribuições de cada etapa de processamento, a saber:

1. $\hat{s}$: saída do codificador;
2. $\hat{s}_b$: saída do decodificador com *anti-blocking* e α ótimo orientado à melhor nota PESQ-MOS;
3. $\hat{s}_f$: saída do decodificador depois do pós-filtro.

Na primeira simulação, os parâmetros do MMP seguem as definições de [6], onde:

- Sinal de entrada para o MMP: Resíduo com amostras quantizadas pelo dicionário inicial

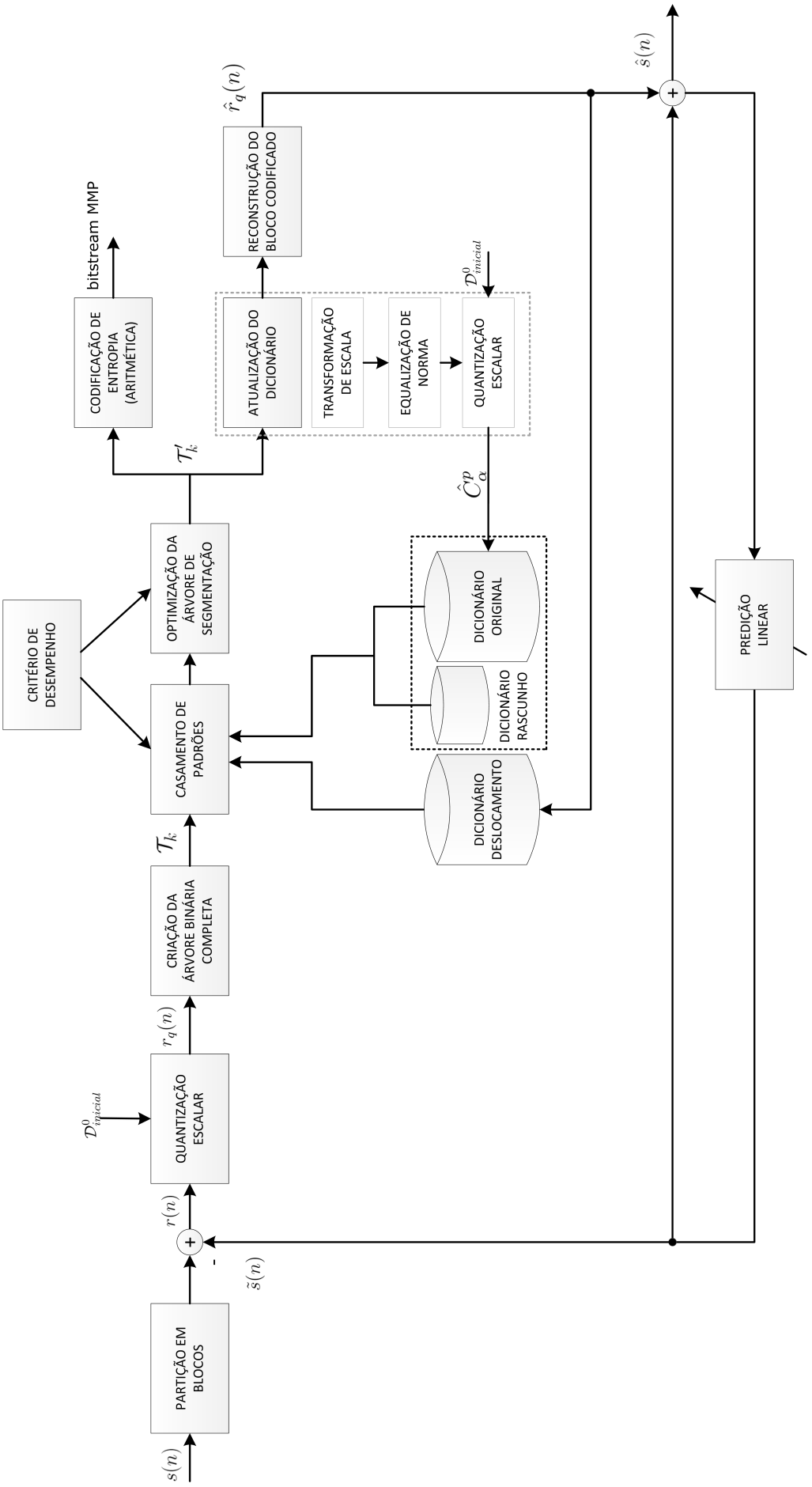


Figura 5.4: Diagrama em blocos do codificador MMP.

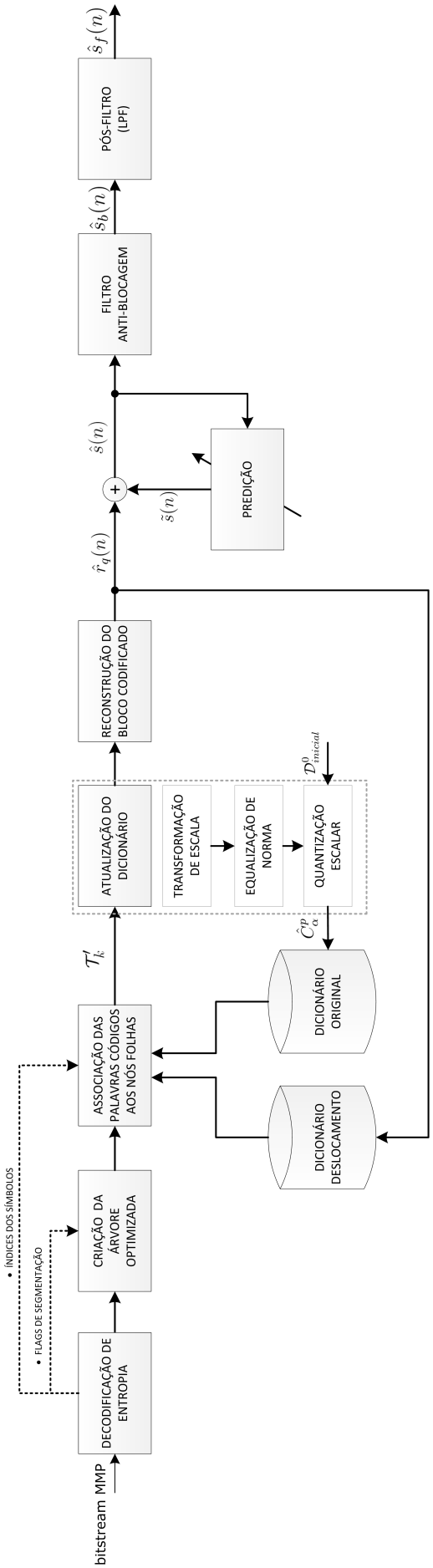


Figura 5.5: Diagrama em blocos do decodificador MMP.

- Dimensão do bloco: 16
- Algoritmo LP: LS_A
- Ordem do preditor: 40
- Tamanho do conjunto de treinamento: 128
- Função da janela: Hamming
- Cardinalidade do Dicionário inicial: 256
- Casamento de Padrões do Dicionário Original: Com análise de dependência (Dicionário Rascunho)
- Casamento de Padrões do Dicionário de Deslocamento: Sem análise de dependência.
- Métrica de Distorção (D): Erro quadrático (SE)
- Critério de Desempenho: $J = D + \lambda R$
- Otimização da Árvore Binária: Algoritmo RDI (RD Intermediário)
- Atualização do Dicionário
  - Controle de Redundância: palavras quantizadas pelo dicionário inicial
- Filtro *Anti-blocking*: aplicado às amostras reconstruídas ($\hat{s}(n)$).

Para tais parâmetros, encontramos valores de λ que geraram taxas de compressão superiores e inferiores à taxa alvo. E, assim como em todas as outras simulações, calculamos os desempenhos de SNR e PESQ-MOS para a taxa referência de 1 bit/amostra, ou seja, 8 kbps, o que nos remete aos resultados apresentados na Tabela 5.2.

Tabela 5.2: Resultados da primeira configuração (MMP Voz - I) codificado à taxa de 8 kbps (1 bit/amostra)

Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	19,76	2,96
$\hat{\mathbf{s}}_\mathbf{b}$	19,41	2,98
$\hat{\mathbf{s}}_\mathbf{f}$	12,58	3,03
G.729 (main)	5,58	3,74

5.4 Contribuições à Estrutura Básica

De forma geral, a etapa de quantização do resíduo antes da codificação MMP beneficia o algoritmo por aproximar as amostras do sinal erro aos mesmos níveis do padrão inicial do dicionário. Esse artifício tende a favorecer o casamento de padrões uma vez que forçamos a existência de apenas 256 níveis diferentes, limitando o comportamento do sinal.

No entanto, a consequência desta técnica é a introdução de uma pré-distorção $(r(n) - r_q(n))$. Na verdade, as modificações inseridas degradam o sinal e contribuem para reduzir a qualidade perceptual como podemos comprovar na nota PESQ-MOS média da Tabela 5.3. Para alcançarmos esse último resultado, aplicamos apenas a quantização escalar nas amostras residuais e desligamos o processamento MMP (Figura 5.6). Isso nos fez perceber que a pré-distorção reflete uma redução na nota PESQ-MOS, que atinge o valor médio de $4,17$ conforme podemos ver na Tabela 5.3.

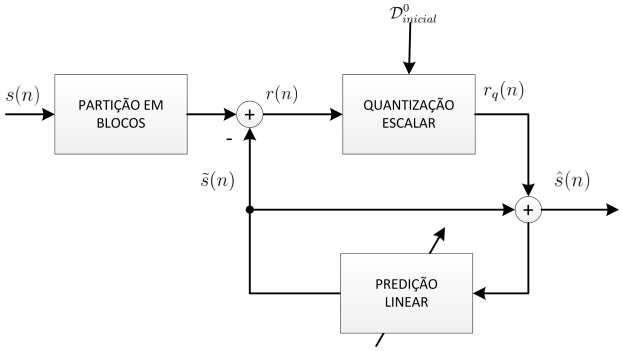


Figura 5.6: Diagrama em blocos das ferramentas que avaliaram o impacto do sinal resíduo quantizado a partir do dicionário inicial.

Tabela 5.3: Avaliação da pré-distorção inserida durante o processo de quantização escalar das amostras residuais cujos níveis se baseiam na versão inicial do dicionário. A Tabela apresenta os valores médios SNR e nota PESQ-MOS das amostras recuperadas a partir de $r_q(n)$.

Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	30,79	4,17

Em nossa próxima avaliação, retiramos a quantização do sinal resíduo e submetemos ao processamento MMP o erro de estimação original, sem qualquer pré-distorção. Nossa expectativa de melhora foi comprovada conforme observamos na tabela 5.4. Obtivemos um ganho de 0,43 dB na relação sinal-ruído para o sinal $\hat{s}$ na saída do codificador e, no domínio perceptual, mesmo que marginalmente, melhoramos a nota média PESQ-MOS do sinal $\hat{s}_f$ atingindo o valor de 3,07 depois do pós-filtro.

Tabela 5.4: Resultados do MMP Voz - II codificado à taxa de 8 kbps (1 bit/amostra)

Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	20,19	2,96
$\hat{\mathbf{s}}_\mathbf{b}$	19,35	2,98
$\hat{\mathbf{s}}_\mathbf{f}$	12,59	3,07
G.729 (main)	5,58	3,74

Mesmo assim, as palavras que são inseridas no dicionário durante o processo de atualização são submetidas às mesmas etapas de quantização escalar com os níveis do dicionário inicial descritas anteriormente. Nossa proposta a seguir é remover essa quantização e aplicar um controle de redundância simples, apenas para garantir que palavras estritamente iguais não sejam adicionadas.

Tabela 5.5: Resultados do MMP Voz - III codificado à taxa de 8 kbps (1 bit/amostra)

Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	22,76	2,98
$\hat{\mathbf{s}}_\mathbf{b}$	21,50	3,00
$\hat{\mathbf{s}}_\mathbf{f}$	13,03	3,05
G.729 (main)	5,58	3,74

Embora tenhamos uma pequena variação da nota média PESQ-MOS na saída do pós-filtro ($\hat{s}_f$), melhoramos em 2,57 dB a relação sinal-ruído na saída do codificador, atingindo o valor de 22,76 dB.

5.4.1 Dicionário de Deslocamento Rascunho

O dicionário de deslocamento, visto na Seção 3.7, até então era limitado ao casamento de padrões de palavras codificadas originárias de blocos anteriores e deslocadas até um limite máximo $\delta = L$. Quando $\delta = 0$ significava que o casamento era realizado com o sub-bloco de mesma dimensão $\hat{r}(n - \mathcal{L}[X^{p_k}])$[1], ou seja, primeiro sub-bloco imediatamente anterior ao início do bloco atual. Neste caso p define a profundidade do sub-bloco a ser aproximado e k indica a posição do nó em uma mesma profundidade.

Nossa proposta é ampliar este conceito, considerando que a otimização da árvore binária é feita pelo algoritmo RD Intermediário. Isto significa que podemos assumir que as amostras anteriores ao sub-bloco de interesse que pertencem à mesma árvore

[1] $\mathcal{L}[X^{p_k}] = 2^{\mathcal{P}-p-1}$

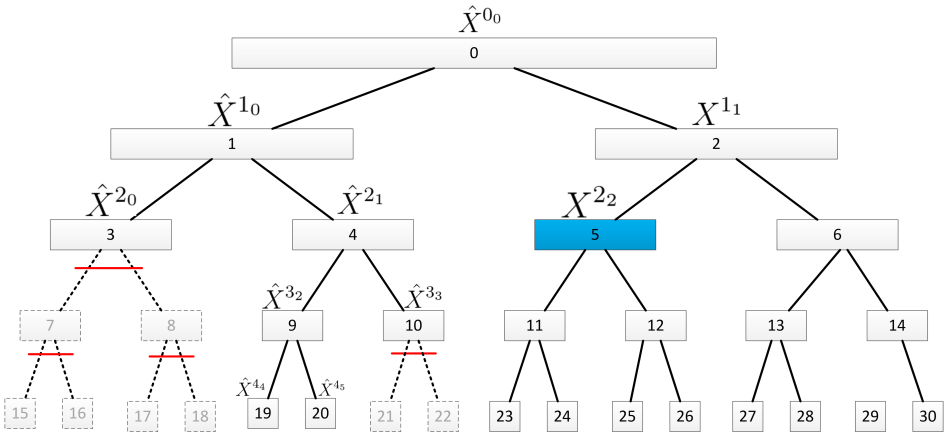


Figura 5.7: Representação do processo de otimização da árvore binária. Os sub-blocos identificados como $\hat{X}^{p_k}$ já foram aproximados. A linha vermelha indica poda dos nós filhos já que análise do custo foi realizada e, em destaque, o nó η_5 de azul representando o sub-bloco X^{2_2}, submetido à busca pelo melhor casamento.

binária já foram codificadas temporariamente e, portanto, as concatenações de nós à esquerda do nó atual, já constituem novas palavras disponíveis no dicionário. Por exemplo, se utilizarmos a árvore binária da Figura 5.7 o sub-bloco X^{2_2} pertencente ao nó η_5 , concluímos que a primeira tentativa de casamento é no sub-bloco representado por $\hat{X}^{2_1}$ (nó η_4) que já foi previamente codificado. A partir de então, variamos o deslocamento δ para procurar o melhor casamento. Podemos ilustrar esse processo de busca no dicionário rascunho de deslocamento pela Figura 5.8. Com esta configuração, não obtivemos ganhos significativos na saída do pós-filtro. Apenas identificamos variações residuais no valor médio de SNR e PESQ-MOS na saída do codificador, como podemos comprovar na Tabela 5.6.

Tabela 5.6: Resultados do MMP Voz - IV codificado à taxa de 8 kbps (1 bit/amostra)

Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	22,80	2,99
$\hat{\mathbf{s}}_\mathbf{b}$	21,45	3,00
$\hat{\mathbf{s}}_\mathbf{f}$	13,04	3,05
G.729 (main)	5,58	3,74

Uma hipótese para que a variação tenha sido pouco relevante reside no tamanho do bloco MMP, mantido em 16. Outro destaque é que não houve uma mudança na utilização dos tipos de dicionário, pois percentualmente as diferenças entre as versões III e IV são irrelevantes (21,8% e 21,2% para a frase us39.wav). Embora essa nova técnica não tenha privilegiado o uso de palavras deslocadas, ela é útil para incrementar a velocidade do aprendizado do algoritmo. A verdade por trás desta afirmação é que, de fato, quando ocorre o casamento com algum sub-bloco

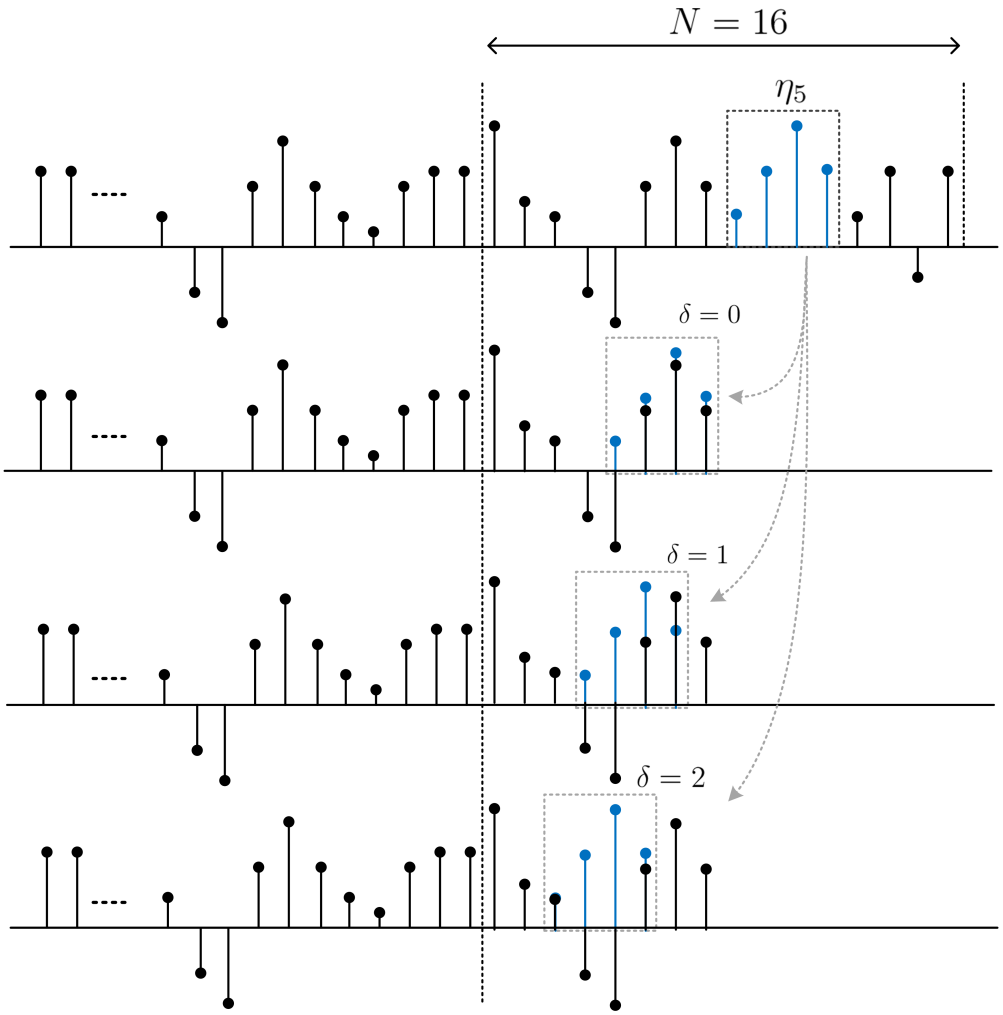


Figura 5.8: Representação do casamento de padrões com conceito de rascunho. Os nós da árvore binária que representam os sub-blocos a esquerda do nó atual já concluíram o casamento de padrões com menor custo. Essas palavras codificadas agora são utilizadas para o casamento do nó atual e $\delta = 0$ representa a palavra imediatamente anterior. No exemplo, queremos aproximar o nó η_5, cuja amostras são $X_{\eta_5} = [r(n+8)\, r(n+9)\, r(n+10)\, r(n+11)]^T$. Se $\delta = 0$ for escolhido, queremos representar o vetor $[r(n+4)\, r(n+5)\, r(n+6)\, r(n+7)]^T$. como melhor aproximação de acordo com o critério de desempenho adotado.

deslocado dentro da mesma árvore, o dicionário pode receber palavras novas, provenientes de partições quaisquer de segmentos, cujas dimensões originais são múltiplas de 2. Outra observação importante é que em trechos periódicos, o dicionário de deslocamento rascunho tende a incrementar as frequências de ocorrência dos índices de deslocamento respeitando uma tendência. Nesse caso tais índices tendem a se aproximar do valor de pitch do segmento sonoro para a mesma escala (Figura 5.9). Tomamos como exemplo o arquivo us39.wav e calculamos a utilização de cada tipo de dicionário para taxa de 8 kbps nas duas versões, III e IV. Os resultados estão nas tabelas 5.7 e 5.8, respectivamente.

Tabela 5.7: Porcentagem de utilização do tipo de dicionário para o arquivo us39.wav codificado pelo MMP Voz III a 8 kbps ($\lambda = 34000$).

Escala	Dic. Deslocamento	Dic. Original
0	0% (0)	100% (68)
1	0,66% (4)	99,34% (602)
2	55,07% (576)	44,93% (470)
3	37,06% (202)	62,94% (343)
4	1,94% (28)	98,06% (1418)
Total	21,80% (810)	78,20% (2901)

Tabela 5.8: Porcentagem de utilização do tipo de dicionário para o arquivo us39.wav codificado pelo MMP Voz IV a 8 kbps ($\lambda = 34100$).

Escala	Dic. Deslocamento	Dic. Original
0	0% (0)	100% (72)
1	7,82% (43)	92,18% (507)
2	39,32% (418)	60,68% (645)
3	50,78% (262)	49,22% (254
4	3,69% (54)	96,31% (1409)
Total	21,20% (777)	78,80% (2887)

No total, as diferenças entre a quantidades de palavras utilizadas que vieram do dicionário original e deslocamento não são relevantes. Observamos algumas alterações apenas entre escalas. Mas vale destacar alguns pontos:

- 100% das palavras com escala 0 são provenientes do dicionário original e,
- Vetores de maior dimensão (escala 4) provém majoritariamente do dicionário original. Isto significa que o comportamento pseudo-estacionário dos quadros

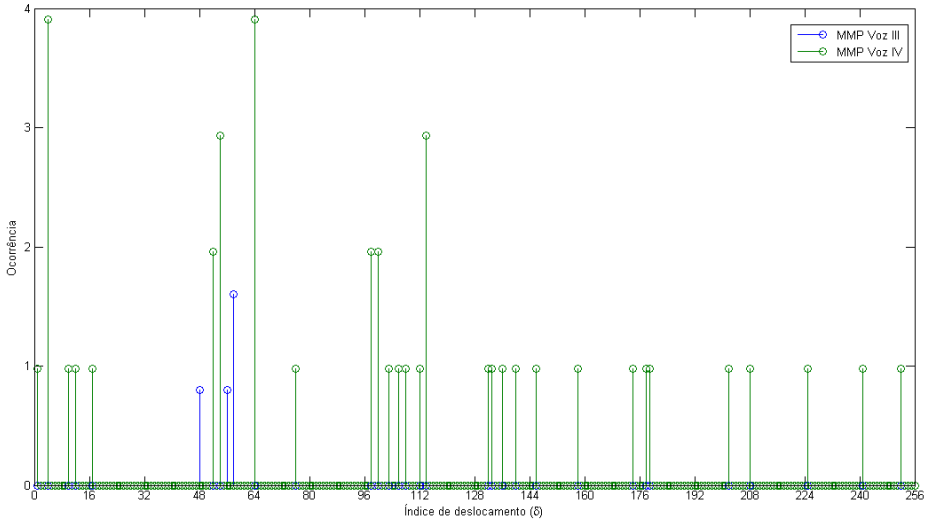


Figura 5.9: O gráfico ilustra a frequência de ocorrência dos índices δ para a dimensão 2x1 do dicionário de deslocamento. Considerando a versão MMP Voz IV, percebemos que os índices mais prováveis tendem a se concentrar em diferenças de aproximadamente 60 amostras, como é o caso de $\delta = 4$ e $\delta = 64$, indicando um pitch estimado em 133 Hz.

sonoros não é considerado quando usamos o critério do erro quadrático no o casamento de palavras deslocadas.

5.4.2 Novos Algoritmos de Predição

Com o objetivo de avaliar o desempenho do MMP a partir de diferentes algoritmos de predição, fizemos duas novas propostas, adicionais ao *Least Squares A*, que chamaremos *Least Squares B* (LS_B) e *Least Squares C* (LS_C).

5.4.3 Least Squares B

Até agora, utilizamos o algoritmo LS_A para estimar amostras $\tilde{s}(n), n = 0, 1, 2, ..., N-1$. A partir delas, geramos o bloco de resíduos que submetemos ao processamento MMP, como vimos na seção 4.2. A fragilidade do algoritmo *Least Squares A* reside no fato que as propriedades temporais, espectrais e estatísticas de um sinal de voz podem variar bastante dependendo do tamanho do bloco N. A proposta do algoritmo LS_B é atualizar os coeficientes à medida que as amostras sejam estimadas. Essa atualização pode corrigir a aproximação que fazemos quando assumimos estacionariedade por partes do sinal de voz. Portanto, a janela de treinamento $\mathbf{s_{w,m}}$ passa a se deslocar e incluir as amostras estimadas. Novos coeficientes são calculados cada vez que uma nova amostra é estimada, pois geramos uma nova matriz de autocorrelação $\mathbf{R}_{S,m}$ e um novo vetor de correlação cruzada $\mathbf{p}_{S,m}$. Vamos

utilizar a notação $\hat{a}_m(i)$ para representar o coeficiente LP de índice i calculado para a amostra deslocada $s(n+m)$. Portanto, a amostra é estimada pelo algoritmo LS_B conforme

$$\begin{aligned} \tilde{s}(n+m) &= \sum_{j=1}^{m} \hat{a}_m(j)\tilde{s}(n+m-j) + \sum_{i=m+1}^{M} \hat{a}_m(i)\hat{s}(n+m-i) \\ &\text{para, } m = 0, 1, 2, ..., N-1 \end{aligned} \tag{5.5}$$

onde $\hat{\mathbf{a}}_m = \mathbf{R}_{S,m}^{-1}\mathbf{p}_{S,m}$ com

$$R_{S,m}(k) = \frac{1}{L}\sum_{i=n-L+m}^{n+m-1} s_{w,m}(i)s_{w,m}(i-k), \quad k = 0, 1, 2, ..., M-1 \tag{5.6}$$

e $\mathbf{p}_{S,m} = [R_{S,m}(1)\ R_{S,m}(2)\ \cdots R_{S,m}(M)]^T$

A atualização da janela de treinamento é feita de acordo com:

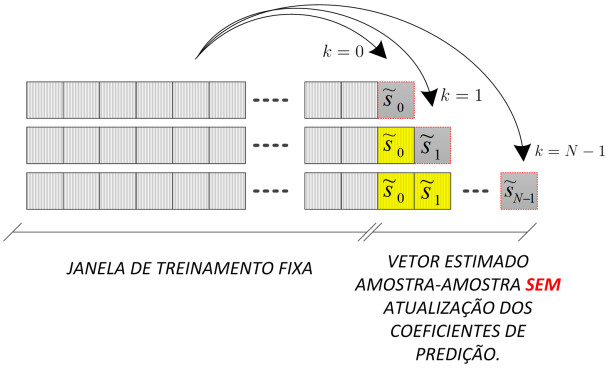
$$\begin{aligned} \mathbf{s_{w}}_0 &= [\hat{s}(n-L)\ \hat{s}(n-L+1)\ \cdots\ \hat{s}(n-2)\ \hat{s}(n-1)] \\ \mathbf{s_{w}}_1 &= [\hat{s}(n-L+1)\ \hat{s}(n-L+2)\ \cdots\ \hat{s}(n-1)\ \tilde{s}(n)] \\ \mathbf{s_{w}}_2 &= [\hat{s}(n-L+2)\ \hat{s}(n-L+3)\ \cdots\ \tilde{s}(n)\ \tilde{s}(n+1)] \\ &\vdots \\ \mathbf{s_{w}}_{N-1} &= [\tilde{s}(n-L+N-1)\ \hat{s}(n-L+N-1)\ \cdots\ \tilde{s}(n+N-3)\ \tilde{s}(n+N-2)] \end{aligned} \tag{5.7}$$

5.4.4 Least Squares C

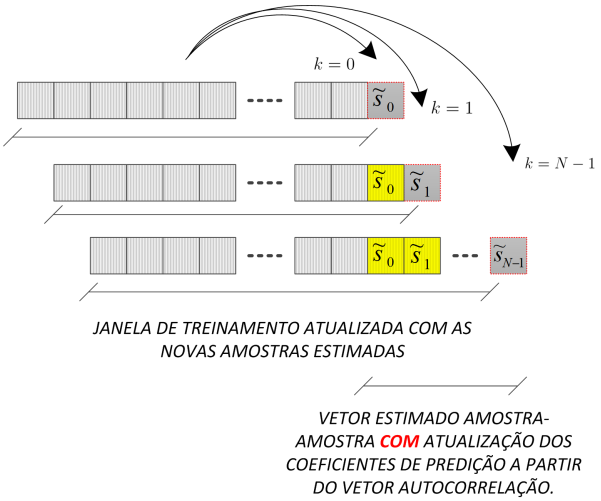
Para um mesmo bloco, o terceiro algoritmo define que a matriz de autocorrelação se mantenha constante, ou seja, assumimos o mesmo modelo estocástico durante a predição de todas as amostras do bloco. No entanto, a cada amostra estimada, atualizamos os coeficientes LP e o vetor de correlação cruzada, uma vez que a predição, agora, é feita para qualquer amostra futura $s(n+i), i = 0, 1, 2, ...N-1$. Por exemplo, quando $i = 1$, a solução é dada por (4.8) e (4.9). Porém, generalizando a equação, quando $i = m$, a solução é dada por:

$$\begin{bmatrix} a1 \\ a2 \\ \vdots \\ a_M \end{bmatrix} = \begin{bmatrix} R_{ss}(0) & R_{ss}(1) & \cdots & R_{ss}(M-1) \\ R_{ss}(1) & R_{ss}(0) & \cdots & R_{ss}(M-2) \\ \vdots & \vdots & \ddots & \vdots \\ R_{ss}(M-1) & R_{ss}(M-2) & \cdots & R_{ss}(0) \end{bmatrix}^{-1} \begin{bmatrix} R_{ss}(m) \\ R_{ss}(m+1) \\ \vdots \\ R_{ss}(m+M) \end{bmatrix} \tag{5.8}$$

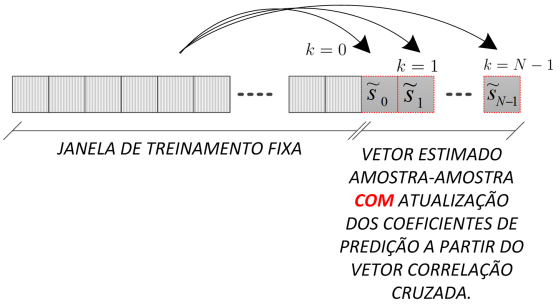
Assim, o algoritmo LS_C se preocupa em não usar amostras futuras que eventualmente tenham sido mal estimadas para evitar a propagação do erro de predição.



(a) *Least Squares A*



(b) *Least Squares B*



(c) *Least Squares C*

Figura 5.10: Representação dos três algoritmos de predição baseados na técnica dos mínimos quadrados. Repetimos a figura do *Least Squares A* apenas por conveniência para facilitar a comparação.

5.4.5 Resultados Experimentais

A comparação entre os três algoritmos foi realizada para diferentes tamanhos de bloco N e tamanhos do conjunto de treinamento L através do Ganho de Predição(GP), cujo valor é definido como

$$GP_{dB} = 10\log_{10}\left(\sum s^2(n)\right) - 10\log_{10}\left(\sum r^2(n)\right). \tag{5.9}$$

Na comparação, encontramos o valor médio para cada algoritmo considerando todo banco de frases listados no Apêndice A. Além disso, submetemos o conjuntos das amostras estimadas $\tilde{s}(n)$ à análise PESQ para obtermos uma indicação de "qualidade perceptual" do preditor. Variamos também a ordem M do preditor, pois calculamos o modelo LP tanto no codificador quanto no decodificador a partir das amostras reconstruídas. Se desconsideramos o custo computacional deste cálculo, isso nos permite avaliar a eficiência do preditor de diferentes ordens, diferente dos codificadores híbridos.

Os codificadores de voz que pertencem ao grupo Análise-por-Síntese estimam o modelo LP do quadro corrente e buscam o melhor conjunto de excitação que minimizam as diferenças perceptuais entre as amostras geradas e as originais. Com o modelo estimado, versões alternativas dos coeficientes LP são quantizadas e transmitidas ao decodificador e, portanto, quanto maior a ordem do modelo LP, maior a taxa de bits para representá-lo. Um valor bastante comum e muito praticado pelos sistemas de compressão deste tipo para sinais de voz de banda estreita que oferecem um bom compromisso taxa *versus* qualidade é de ordem 10. Com essa ordem, o modelo LP consegue prever as redundâncias de curta duração (*short-term*), ou seja, na vizinhança próxima a amostra de interesse. Esta é a razão pela qual os codificadores de voz também aplicam a predição de longa duração (*long-term*) para estimar o valor de pitch, cuja característica temporal é bem evidente nos trechos vozeados. Em geral, é comum que este filtro seja de primeira ordem da forma

$$P(z) = \frac{1}{1 - \beta z^{-T}} \tag{5.10}$$

onde T é o período de *pitch* e β é o coeficiente que minimiza o erro médio quadrático, calculado da forma $E\left[(\tilde{s}(n) - \beta\hat{s}(n-T))^2\right]$. Na prática, o período de *pitch* está compreendido entre 2 ms e 20 ms, ou seja, para uma taxa de amostragem de 8000 amostras por segundo, isto significa que T está entre 20 e 160. É possível encontrar na literatura muitos algoritmos que estimam valores de T, mas não serão abordados neste trabalho.

Como nosso modelo de predição linear é aplicado às amostras reconstruídas, não há limitações relacionadas a ordem do filtro que impliquem diretamente na taxa de

transmissão, pois os coeficientes LP são calculados tanto no codificador quanto no decodificador. O único ponto que consideramos é a complexidade computacional que incrementa a medida que aumentamos a ordem do preditor.

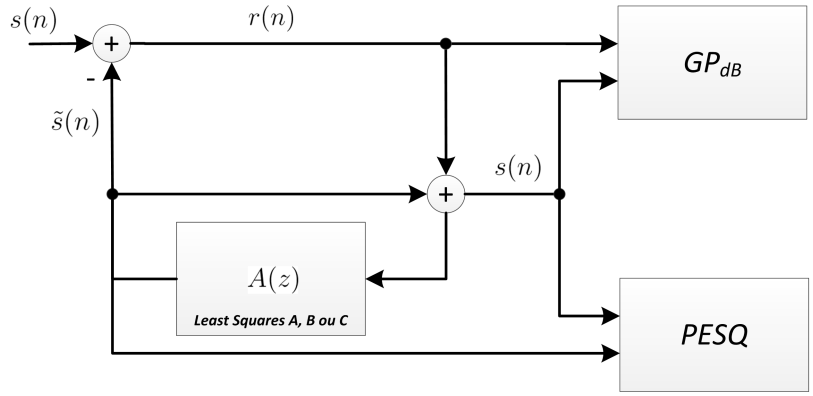


Figura 5.11: Diagrama em blocos do método de comparação entre os algoritmos LS_A, LS_B e LS_C.

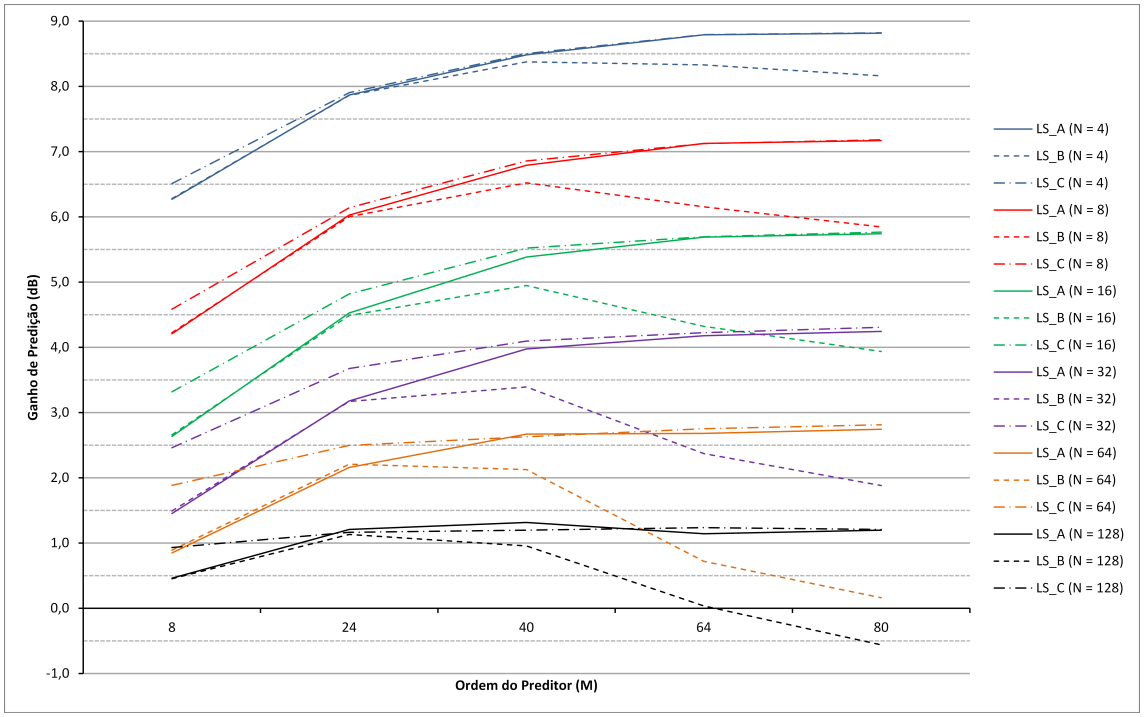
Os resultados das avaliações estão nos gráficos das Figuras 5.12 e 5.13 para duas dimensões de conjunto de treinamento, 128 e 256 respectivamente. Esses testes foram executados de acordo com o diagrama em blocos representado pela Figura 5.11, sem influência do processamento MMP.

É de se esperar que a cada vez que o tamanho do bloco aumente, a predição fique menos precisa, já que o algoritmo busca estimar amostras mais distantes, o que justifica as diferenças acentuadas entre cada valor de N. Observamos também que, a medida que a ordem do preditor aumenta, o desempenho do algoritmo LS_B piora, indicando que o erro de predição é propagado no cálculo das estimações seguintes.

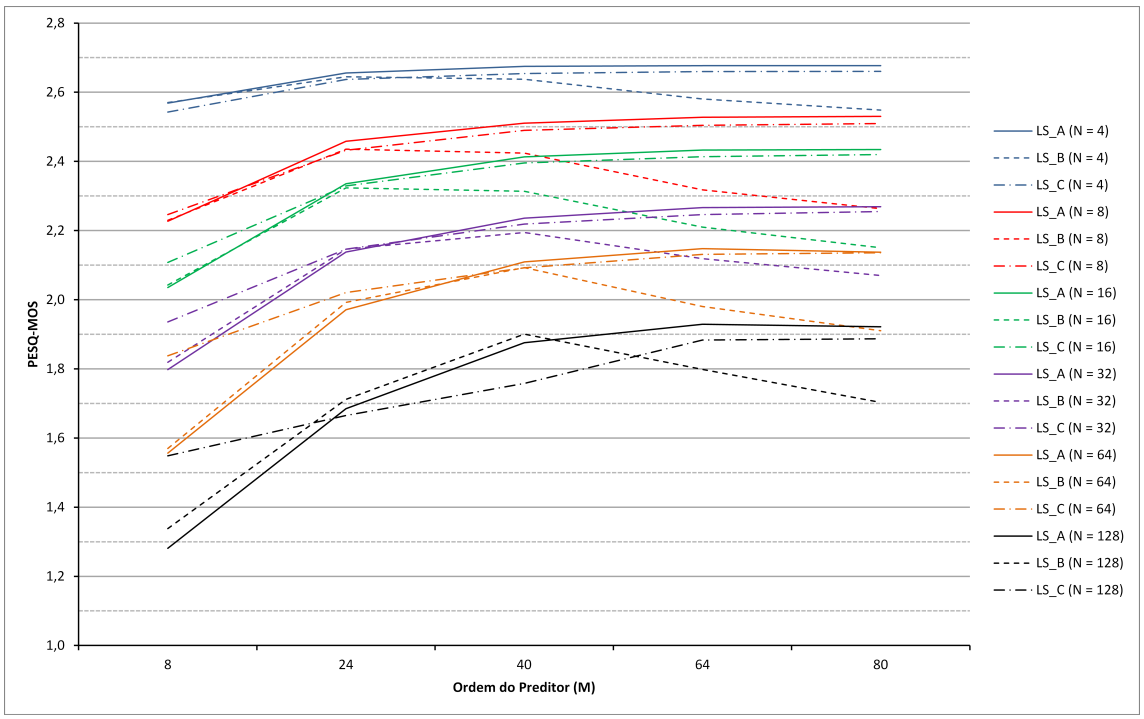
Como os gráficos indicam, o algoritmo que apresentou o melhor desempenho foi LS_C, para os diferentes tamanhos de bloco, superando o algoritmo LS_A em muitos casos. Outro destaque reside no tamanho do conjunto de treinamento. Com 256 amostras obtivemos resultados superiores ao conjunto de 128.

A partir deste novo cenário, incluímos o novo algoritmo de predição LS_C ao processamento MMP. Realizamos novas simulações de acordo com diagrama em blocos do MMP Voz - V representado pela Figura 5.14, e, os parâmetros que produziram a melhor qualidade perceptual, segundo a métrica PESQ, são:

- Sinal de entrada para o MMP: Resíduo original (sem quantização)
- Dimensão do bloco: 16
- Algoritmo LP: LS_C
- Ordem do preditor: 40
- Tamanho do conjunto de treinamento: 128

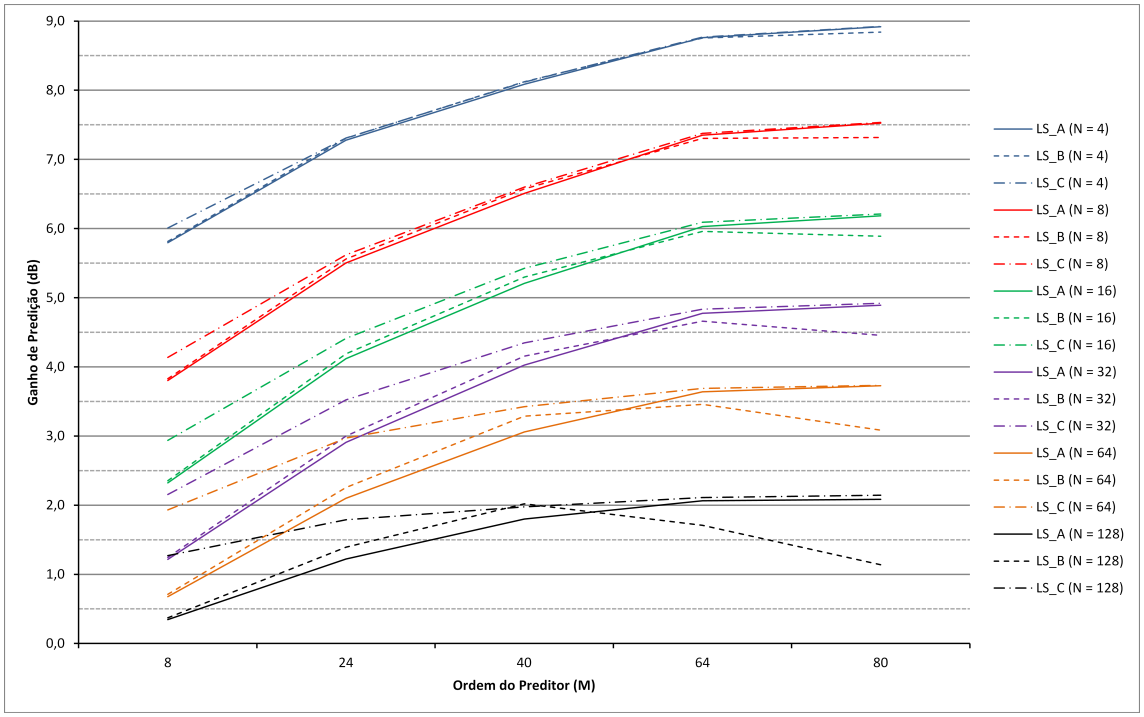


(a) Comparação a parir do Ganho de Predição (dB) médio, calculado para todo o banco de frases.

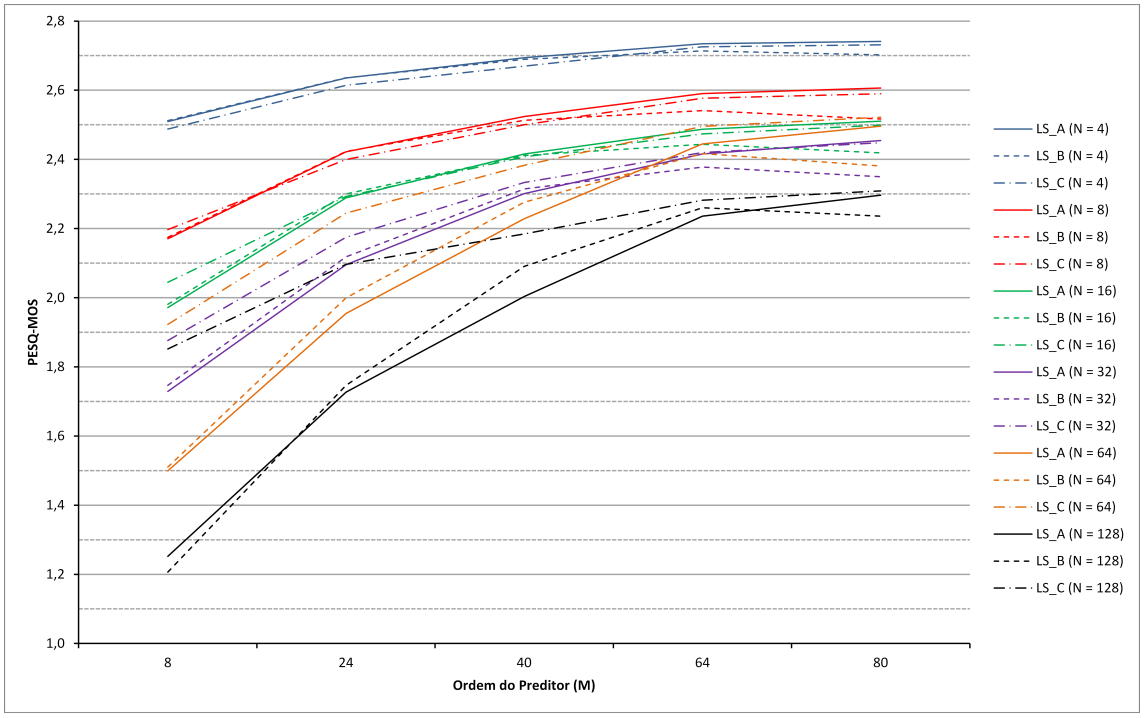


(b) Comparação a partir da nota média PESQ-MOS, calculado para todo o banco de frases.

Figura 5.12: Comparação entre os algoritmos *Least Squares A, B e C*. O conjunto de treinamento está limitado à **128** amostras.



(a) Comparação a parir do Ganho de Predição (dB) médio, calculado para todo o banco de frases.



(b) Comparação a partir da nota média PESQ-MOS, calculado para todo o banco de frases.

Figura 5.13: Comparação entre os algoritmos *Least Squares A, B e C*. O conjunto de treinamento está limitado à **256** amostras.

- Função da janela: Retangular
- Cardinalidade do Dicionário inicial: 256
- Casamento de Padrões do Dicionário Original: Com análise de dependência (Dicionário Rascunho)
- Casamento de Padrões do Dicionário de Deslocamento: Com análise de dependência (Dicionário de Deslocamento Rascunho).
- Métrica de Distorção (D): Erro quadrático (SE)
- Critério de Desempenho: $J = D + \lambda R$
- Otimização da Árvore Binária: Algoritmo RDI (RD Intermediário)
- Atualização do Dicionário
  - Controle de Redundância: palavras estritamente iguais não são incluídas
- Filtro *Anti-blocking*: aplicado às amostras reconstruídas ($\hat{s}(n)$).

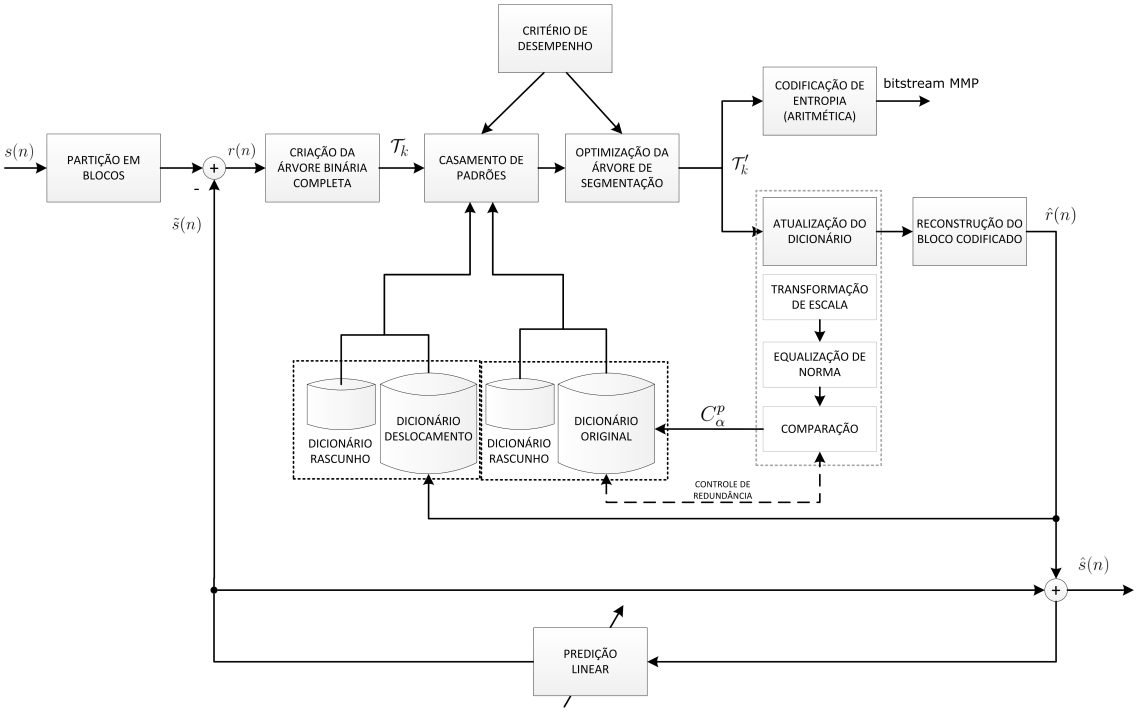


Figura 5.14: Diagrama em blocos do codificador MMP Voz - V.

Na Tabela 5.9 apresentamos a condição que alcançamos a partir dos parâmetros descritos.

Embora ainda bastante distante do padrão de qualidade do G.729, alteramos o tamanho do conjunto de treinamento para 256 e obtivemos melhoras em todas métricas, como pode ser visto na Tabela 5.10.

Tabela 5.9: Resultados do MMP Voz - V codificado à taxa de 8 kbps (1 bit/amostra).

Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	23,01	3,01
$\hat{\mathbf{s}}_\mathbf{b}$	20,94	3,02
$\hat{\mathbf{s}}_\mathbf{f}$	13,05	3,07
G.729 (main)	5,58	3,74

Tabela 5.10: Resultados do MMP Voz - VI codificado à taxa de 8 kbps (1 bit/amostra).

Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	23,34	3,05
$\hat{\mathbf{s}}_\mathbf{b}$	22,00	3,06
$\hat{\mathbf{s}}_\mathbf{f}$	13,11	3,11
G.729 (main)	5,58	3,74

5.5 Novo Controle de Redundância

O desempenho do MMP está diretamente relacionado à velocidade de aprendizado dos padrões recorrentes no sinal fonte. No entanto, como visto no Capítulo 3, esse aprendizado deve ser eficiente para evitar que palavras inúteis com pouca representatividade de um novo padrão do sinal fonte. Podemos contornar esse problema se aplicarmos o conceito de similaridade, como visto em [3], evitando a inclusão de vetores redundantes. Em outras palavras, permitimos que novas palavras sejam incluídas no dicionário apenas se a diferença entre elas e as palavras existentes respeitarem uma distância mínima, definida por um valor d. Neste caso, a distância pode ser interpretada como uma medida de distorção entre o novo vetor C_α^p e as palavras existentes $\mathcal{C}_i^p, i = 0, 1, ..., \mathcal{L}[\mathcal{D}^p]$ de mesma escala p. Novas palavras serão incluídas apenas se

$$\sum_m (C_\alpha^p(m) - \mathcal{C}_i^p)^2 > d^2. \tag{5.11}$$

O parâmetro d controla a redundância entre a palavras do dicionário e, para determinar seu valor, precisamos considerar que, quanto menor, maior tenderá ser a cardinalidade do dicionário. Por outro lado, quanto maior for o valor de d, mais diferentes as palavras serão entre si e, consequentemente, podemos fazer as aproximações com um grau de distorção muito alto, que talvez seja não desejável. Isto nos dá a indicação que para taxas altas de compressão (λ alto) , d tende a ser grande por reduzir a quantidade de padrões no dicionário. Ou de forma oposta, quando d

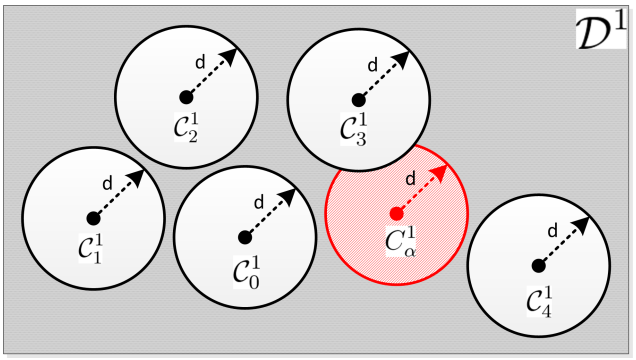


Figura 5.15: Análise da distância mínima para inclusão no dicionário na dimensão dois.

for pequeno, buscamos taxas de compressão menores (λ pequeno). A Figura 5.11 ilustra o caso da dimensão dois. Para encontrar o melhor valor de d, rodamos várias simulações, cujos resultados estão apresentados na Tabela 5.11

Tabela 5.11: Resultados do MMP Voz VII para diferentes distâncias d aplicado ao controle de redundância do dicionário (valores calculados para taxa de codificação de 8 kbps - 1 bit/amostra)

	$d = 2$		$d = 32$		$d = 256$	
Sinal	SNR (dB)	PESQ-MOS	SNR (dB)	PESQ-MOS	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	23,40	3,05	23,63	3,06	23,29	3,13
$\hat{\mathbf{s}}_\mathbf{b}$	22,16	3,06	22,10	3,07	22,28	3,14
$\hat{\mathbf{s}}_\mathbf{f}$	13,12	3,11	13,13	3,12	13,16	3,18

Conseguimos alcançar a melhor performance do MMP Voz para 8 kbps com $d = 256$, resultando na maior nota PESQ-MOS: 3,18 na saída do pós-filtro. Além desses valores de d, não observamos variações relevantes na relação sinal-ruído.

5.6 Incremento na Velocidade de Aprendizado

Até agora a atualização do dicionário está limitada a processar apenas os padrões recorrentes do sinal de interesse, suas partições deslocadas e, claro, suas versões expandidas e contraídas. No entanto, podemos incrementar a velocidade de aprendizado se aplicarmos uma transformação geométrica na palavra codificada. Isso permitiria que o MMP já pudesse se preparar para codificar padrões que ainda não tivessem ocorrido. Em nosso caso, submetemos também ao processo de atualização do dicionário com todo o rigor no controle de redundância discutido até aqui, a

palavra simétrica $\bar{C}_{\alpha}^{p}$ definida da forma

$$\bar{C}_{\alpha}^{p}(i) = (-1)C_{\alpha}^{p}(i), i = 0, 1, 2, ..., 2^{\mathcal{P}-p-1} \tag{5.12}$$

Na Figura 5.16, que tomamos como exemplo a frase us39.wav, observamos a velocidade da cardinalidade do dicionário aumentando, nas diferentes escalas, à medida que o processamento avança quando incluímos também as versões simétricas das palavras codificadas.

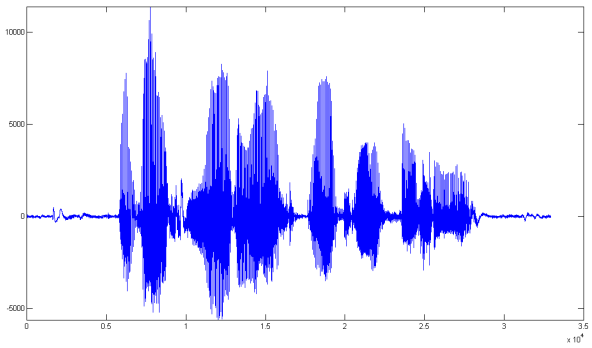
Com essas últimas modificações elevamos a qualidade média do MMP Voz, conforme observamos na Tabela 5.12.

Tabela 5.12: Resultados do MMP Voz - VIII codificado à taxa de 8 kbps (1 bit/amostra)

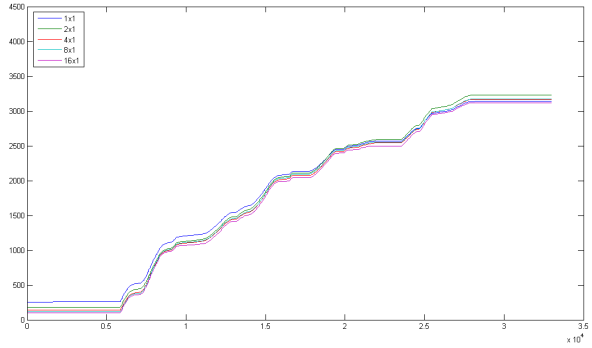
Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	23,41	3,15
$\hat{\mathbf{s}}_{\mathbf{b}}$	21,70	3,16
$\hat{\mathbf{s}}_{\mathbf{f}}$	13,18	3,20
G.729 (main)	5,58	3,74

5.7 Conclusões

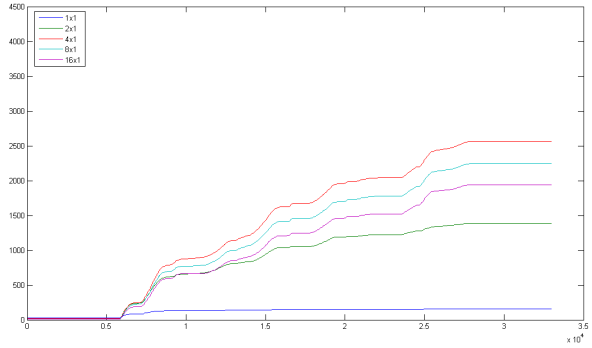
Neste capítulo apresentamos o estado da arte do MMP Voz e diversas ferramentas que contribuíram para aperfeiçoar o método, como por exemplo o filtro redutor de efeito blocos e um pós-filtro. Outros dois algoritmos de predição linear baseados no método dos mínimos quadrados foram propostos com o objetivo de reduzir a energia média do sinal resíduo e analisados. Comprovamos que o uso do *Least Squares* C melhora a performance do codificador. Um método inteligente de controle de redundância baseado no conceito de similaridade entre os vetores foi apresentado e discutido. Por último, sugerimos incrementar a velocidade de aprendizado do dicionário a partir da transformação simétrica da palavra codificada, preparando o MMP para codificar padrões que ainda não ocorreram, respeitando a distância d entre palavras existentes. No capítulo seguinte veremos algumas abordagens perceptuais no processamento do MMP.



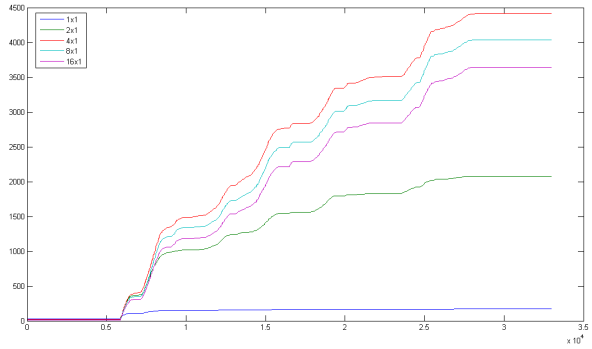
(a) Sentença us39.wav: *"A vent near the edge brought in fresh air"*.



(b) Crescimento do dicionário por escala com $d = 0$.



(c) Crescimento do dicionário por escala com $d = 256$.



(d) Crescimento do dicionário por escala com $d = 256$ com atualização de palavras simétricas $\bar{C}_{\alpha}^{p}$.

Figura 5.16: Comparação entre crescimento do dicionário na frase us39.wav codificada à 8 kbps ($\lambda_a = 31300$, $\lambda_b = 31300$, $\lambda_c = 29500$)

Tabela 5.13: Resultados consolidados do MMP Voz

Versão MMP Voz	Sinal	SNR (dB)	PESQ-MOS
MMP Voz I	$\hat{\mathbf{s}}$	19,76	2,96
	$\hat{\mathbf{s}}_\mathbf{b}$	19,41	2,98
	$\hat{\mathbf{s}}_\mathbf{f}$	12,58	3,03
MMP Voz II	$\hat{\mathbf{s}}$	20,19	2,96
	$\hat{\mathbf{s}}_\mathbf{b}$	19,35	2,98
	$\hat{\mathbf{s}}_\mathbf{f}$	12,59	3,07
MMP Voz III	$\hat{\mathbf{s}}$	22,76	2,98
	$\hat{\mathbf{s}}_\mathbf{b}$	21,50	3,00
	$\hat{\mathbf{s}}_\mathbf{f}$	13,03	3,05
MMP Voz IV	$\hat{\mathbf{s}}$	22,80	2,99
	$\hat{\mathbf{s}}_\mathbf{b}$	21,45	3,00
	$\hat{\mathbf{s}}_\mathbf{f}$	13,04	3,05
MMP Voz V	$\hat{\mathbf{s}}$	23,01	3,01
	$\hat{\mathbf{s}}_\mathbf{b}$	20,94	3,02
	$\hat{\mathbf{s}}_\mathbf{f}$	13,05	3,07
MMP Voz VI	$\hat{\mathbf{s}}$	23,34	3,05
	$\hat{\mathbf{s}}_\mathbf{b}$	22,00	3,06
	$\hat{\mathbf{s}}_\mathbf{f}$	13,11	3,11
MMP Voz VII	$\hat{\mathbf{s}}$	23,29	3,13
	$\hat{\mathbf{s}}_\mathbf{b}$	22,28	3,14
	$\hat{\mathbf{s}}_\mathbf{f}$	13,16	3,18
MMP Voz VIII	$\hat{\mathbf{s}}$	23,41	3,15
	$\hat{\mathbf{s}}_\mathbf{b}$	21,70	3,16
	$\hat{\mathbf{s}}_\mathbf{f}$	13,18	3,20
G.729 (main)		5,58	3,74

Capítulo 6

Características Perceptuais do MMP Voz

Neste capítulo, introduzimos as análises perceptuais no MMP Voz. Novas propostas no casamento de padrões que levam em consideração as características do sistema auditivo humano são apresentadas. Além disso, outras contribuições perceptuais são feitas nas etapas de pré e pós processamento que melhoram a qualidade média do nosso banco de frases.

De fato, temos atuado na estrutura básica do MMP e nos parâmetros de codificação com o objetivo de alcançar a versão de compressão mais eficiente, respeitando o compromisso taxa-distorção. No entanto, a medida de distorção que aplicamos na codificação dos vetores, o erro médio quadrático, não reflete as características do sistema auditivo humano, nem tão pouco fenômenos como o mascaramento. Nossa dificuldade em incluir qualquer análise perceptual no *loop* de codificação do MMP é obter tais informações de um conjunto pequeno de amostras que, em geral, é não-representativo no domínio espectral.

Muitas pesquisas têm estudado as particularidades da nossa audição e da produção do sinal de voz propondo modelos que se aproximam muito do sistema real. E, em sua grande parte, recomendam que as análises perceptuais sejam realizadas em conjuntos de amostras entre 10 e 30 ms. Pequenos vetores com amostras do sinal de voz, com em nosso caso (16x1, 8x1, ..., 1x1), não carregam informações suficientes para indicar qualquer fenômeno no domínio perceptual.

O próprio algoritmo PESQ, indicado como a métrica objetiva com maior correlação com as avaliações subjetivas de sinais de voz de banda estreita, em essência, calcula as distorções quadro-a-quadro, cuja duração é de 32 ms. Para um sinal com taxa de amostragem de 8 KHz, cada quadro equivale à 256 amostras. Na verdade, o algoritmo PESQ executa um processamento bastante rebuscado entre o sinal de referência e o sinal degradado de alinhamento temporal, equalização de nível, identificação de atividade e cálculo de distorções. As distorções são computadas após o

mapeamento tempo-frequência do quadro e consideram a sobreposição de 50% de quadros adjacentes. Em seguida essas distorções são integradas ao longo da duração do sinal, restritos apenas aos trechos que o algoritmo elege como válidos. A função de integração mapeia as degradações em um modelo cognitivo para assim produzir um nota global, cujo sentido físico é representar a nota MOS. Mais detalhes do algoritmo podem ser encontrados em [13].

O importante dessa análise é considerar que o algoritmo PESQ pode indicar a qualidade de codificação do MMP Voz ao longo do sinal, em trechos vozeados ou surdos, ao invés de produzir apenas uma avaliação global. Portanto, em nossos exemplos, vamos incluir o comportamento da nota PESQ quadro a quadro, pois mesmo sem um sentido físico, a análise individual pode ajudar na evolução do algoritmo.

6.1 Análise do Silêncio

Alguns codificadores de voz, a exemplo do algoritmo G.729 Anexo B, identificam trechos de silêncio para codificá-los de maneira eficiente. O principal objetivo é representar esse conjunto de amostras com um número reduzido de bits sem comprometer a qualidade do sinal codificado. Normalmente, esses algoritmos ativam a geração de ruído de fundo no decodificador, uma técnica conhecida como *comfort noise*, cujo controle é feito através de um *flag* transmitido pelo codificador. Isso significa que o ruído de fundo artificialmente gerado substitui o conjunto de amostras classificado como silêncio o que, para nossa audição, é mais confortável do que o silêncio absoluto.

Se analisarmos a forma de onda do sinal de voz, podemos identificar visualmente os trechos de silêncio. Matematicamente, também é fácil identificar tais quadros a partir do cálculo de energia,

$$E_k = \sum_{m=-\infty}^{\infty} [s(m)w(k-m)]^2 \tag{6.1}$$

uma simples técnica de processamento no domínio do tempo. Uma característica evidente dos quadros de silêncio é a baixa energia se comparados a trechos sonoros ou surdos. No entanto, se quisermos ser mais precisos na detecção do início e fim de quadros ativos, ou seja, com locução em uma frase, tema importante nos problemas de reconhecimento de voz, podemos analisar também a taxa de cruzamento por zero, definida por

$$Z_k = \sum_{m=-\infty}^{\infty} |sgn[s(m)] - sgn[s(m-1)]|\, w(k-m) \tag{6.2}$$

onde

$$sgn[s(m)] = \begin{cases} 1 & s(m) \geqslant 0 \\ -1 & s(m) < 0 \end{cases} \tag{6.3}$$

que em muitas vezes é usada de forma combinada com o valor de energia. Esse problema se torna mais difícil quando detectamos sons fricativos fracos (/f/,/th/,/h/) no início ou no fim do quadro, sons nasalados nas amostras finais, sons explosivos fracos (/t/,/p/,/k/) no início ou no fim do quadro, entre outros. Mesmo assim, de forma bem clássica, utilizamos os valores de E_k e Z_k para classificação conforme demonstrado na Figura 6.1.

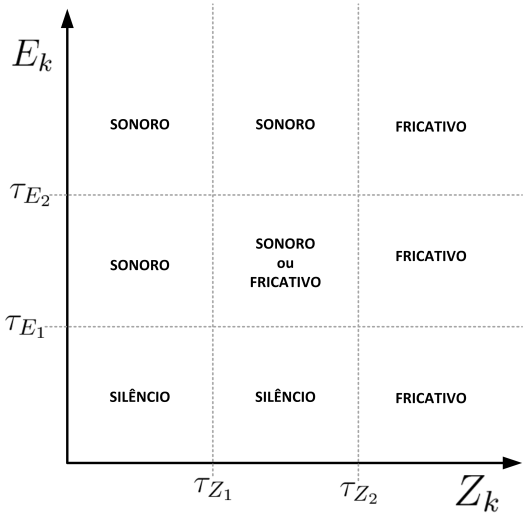


Figura 6.1: Classificação comum dos trechos de voz entre sonoro, fricativo e silêncio baseada na energia e na taxa de cruzamento por zero do quadro de interesse.

Muitos algoritmos buscam determinar os melhores valores limites τ_E e τ_Z para aumentar a precisão do classificador. No entanto, em casos que ainda haja incertezas, por exemplo, quando $\tau_{E_1} < E_k < \tau_{E_2}$ e $\tau_{Z_1} < Z_k < \tau_{Z_2}$, informações de sonoridade adicionais são importantes para tomar a decisão. Em muitos casos a estimação do pitch é um recurso bastante utilizado. Uma revisão do assunto pode ser encontrada em [31].

Como os trechos de silêncio têm baixa energia, uma pequena degradação no sinal reconstruído pode ser facilmente percebida pelo sistema auditivo humano. Isso contribui para reduzir a qualidade final e nos conduz ao próximo experimento. Avaliamos o comportamento do sinal decodificado pelo MMP, com melhor qualidade objetiva ($\hat{\mathbf{s}}_\mathbf{f}$), especificamente em trechos não-ativos, cuja classificação foi realizada partir do algoritmo definido na recomendação ITU-T P.56. Calculamos as notas PESQ-MOS quadro-a-quadro para compará-las as mesmas notas obtidas a partir do sinais decodificados CELP. A motivação desta análise é conhecer a eficiência do MMP relacionada à sonoridade da voz. A partir dela, podemos sugerir novas propostas de codificação, seja na etapa de casamento de padrões ou mesmo na inclusão

de ruído de fundo durante a compressão de quadros não-ativos, que podem ser classificados previamente. Nesse contexto, temos um novo campo a ser explorado, o que introduz características perceptuais ao MMP.

Processamos então, todo o banco de frases para avaliar duas condições: a proporção de quadros de silêncio e a nota PESQ-MOS quadro-a-quadro. Identificamos que, na média, 40,48% das amostras pertencem a trechos inativos e, portanto, representam uma parcela considerável de informação do nosso conteúdo. As Figuras 6.2 e 6.3 refletem a afirmativa anterior para todo o banco e para a frase us39.wav, selecionada como exemplo.

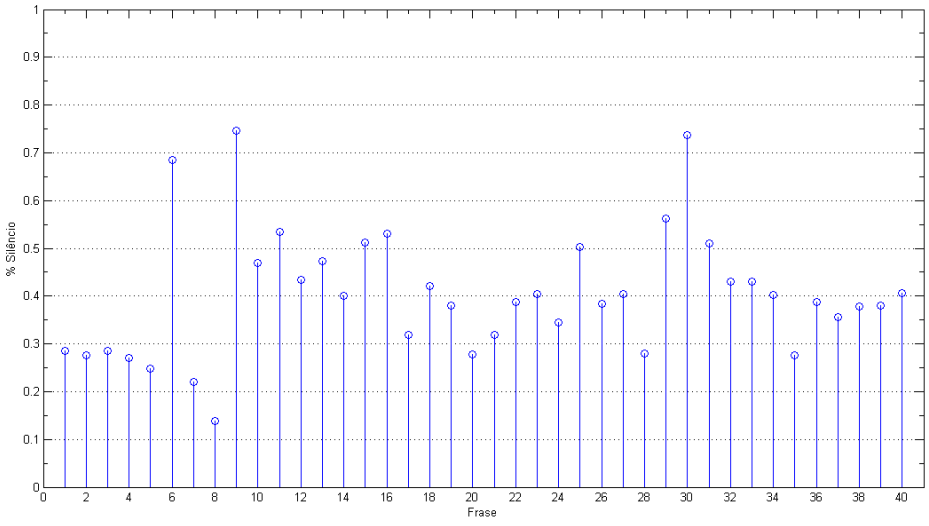


Figura 6.2: Proporção de silêncio por frase.

A Figura 6.4 apresenta as notas PESQ-MOS quadro-a-quadro, calculadas a partir dos sinais codificados pelos algoritmos G.729B e MMP Voz com valores de λ definidos para taxas abaixo e acima de 8 kbps. Observamos que a codificação MMP gera notas consistentemente abaixo do padrão CELP nos trechos de silêncio, já identificados no início e no fim da frase[1].

Diante deste cenário, substituímos tais trechos nas frases decodificadas MMP pelos blocos dos sinais CELP na expectativa de superar as notas PESQ-MOS intermediárias. Mais ainda, esperamos que tais amostras contribuam significativamente na melhora da nota global, uma vez que elas representam quase a metade de toda a sentença. O transplante desses blocos foi feito com os devidos alinhamentos temporal e em amplitude produzindo novas frases que submetemos a uma nova etapa de avaliação objetiva. Os resultados para todo o banco estão apresentados na Tabela

[1]Esse comportamento de qualidade é equivalente para todos os arquivos de voz do banco. As notas PESQ-MOS intermediárias dos sinais decodificados pelo MMP são menores do que as notas dos sinais CELP.

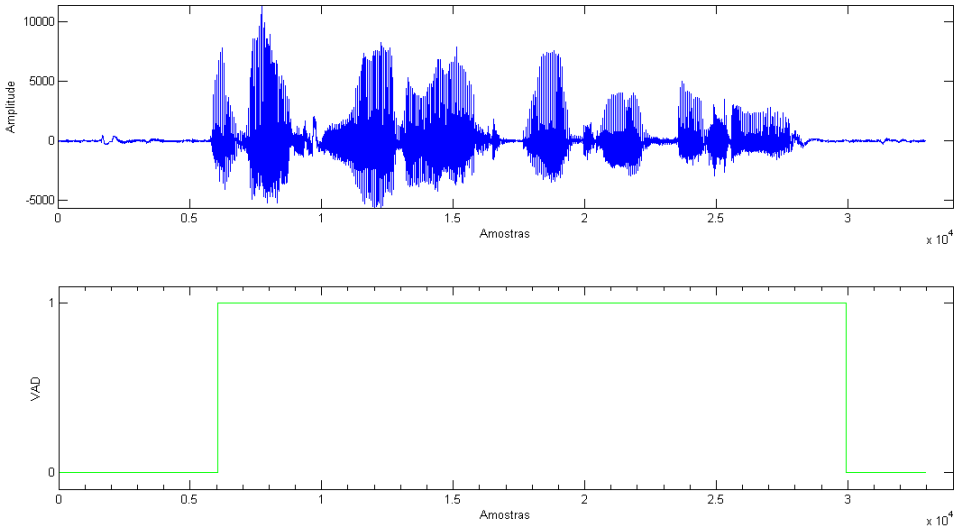


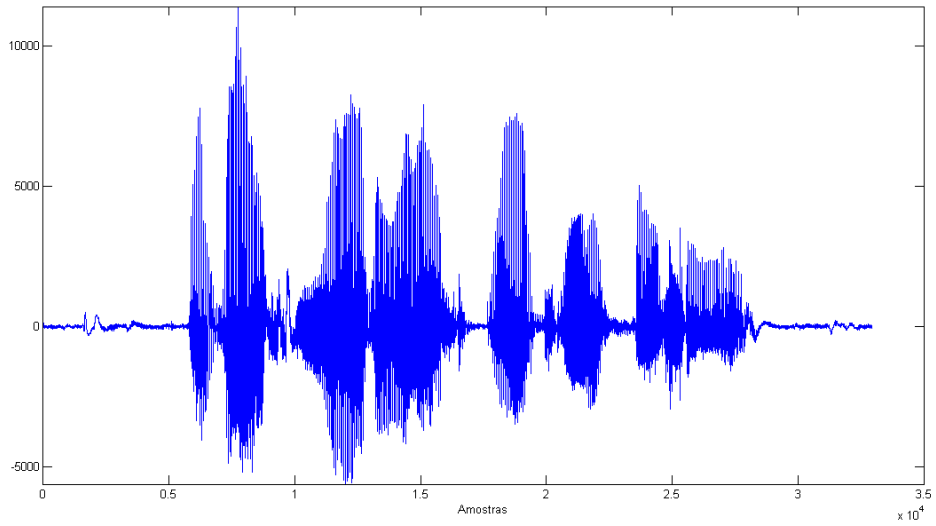
Figura 6.3: Resultado do algoritmo de detecção de atividade baseada na recomendação ITU-T P.56 para a frase us39.wav. O valor 1 significa trecho com atividade, o que deixa para o início e fim da frase a evidência de blocos de silêncio.

6.1 e, como podemos perceber, a qualidade total sofreu uma variação marginal embora o gráfico da Figura 6.5 apresente maiores notas nos trechos de silêncio da frase usada em nosso exemplo. Particularmente na frase tomada como exemplo, as notas globais PESQ-MOS sofreram modificações residuais.

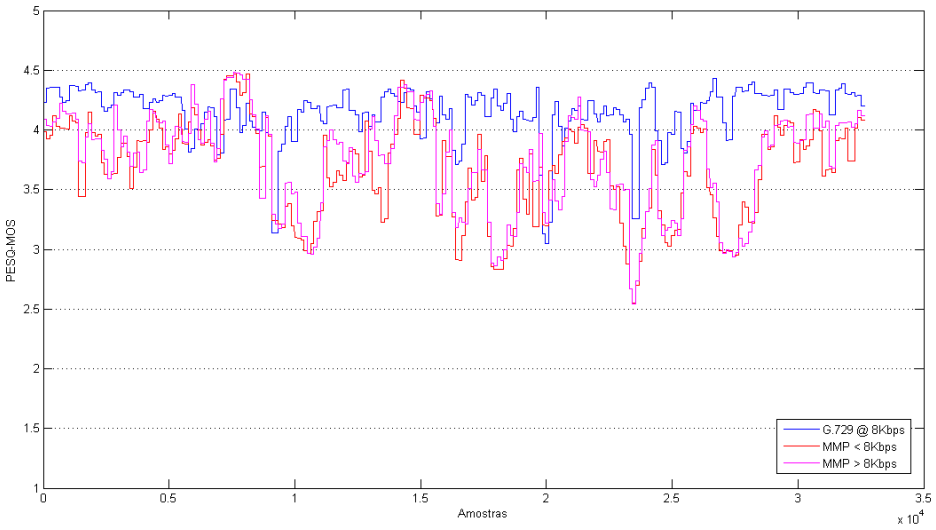
Tabela 6.1: Resultados do MMP Voz VIII com os trechos de silêncio substituídos

Frase	Sinal	PESQ-MOS	Descrição
us39.wav	$\hat{\mathbf{s}}_\mathbf{f}$ @ $<$ 8 kbps	2,91	Sem transplante
	$\hat{\mathbf{s}}_\mathbf{f}$ @ $>$ 8 kbps	2,97	Sem transplante
	$\hat{\mathbf{s}}_\mathbf{f}$ @ $<$ 8 kbps	2,91	Com transplante
	$\hat{\mathbf{s}}_\mathbf{f}$ @ $>$ 8 kbps	2,96	Com transplante
Média para todo o banco	$\hat{\mathbf{s}}_\mathbf{f}$ @ 8 kbps	3,14	Com transplante
G.729B		3,62	G.729 com detecção de atividade

A pequena variação na nota PESQ-MOS nos dá a indicação que fatalmente as degradações geradas pelo MMP nos blocos de silêncio não devam ser tão relevantes. Porém, mesmo com a redução mínima de qualidade implementamos funções no codificador e decodificador que forçam o algoritmo MMP a processar os trechos ativos, permitindo que apenas vetores originados de resíduos de quadros sonoros e surdos sejam incluídos no dicionário. Assim, obrigamos o algoritmo a se adaptar apenas aos padrões recorrentes de blocos com atividade. Novamente, novos valores de λ foram encontrados para cada frase do banco de forma a atingir a taxa média

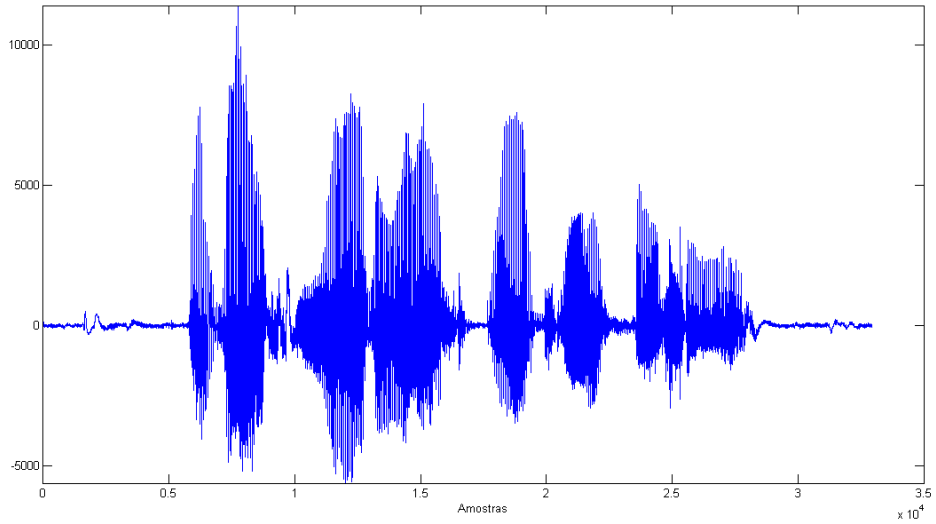


(a) Sinal referência.



(b) PESQ-MOS quadro-a-quadro com sinais originais.

Figura 6.4: Notas PESQ-MOS quadro-a-quadro para a frase exemplo us39.wav codificada à taxa de 1 bit/amostra.



(a) Sinal referência.

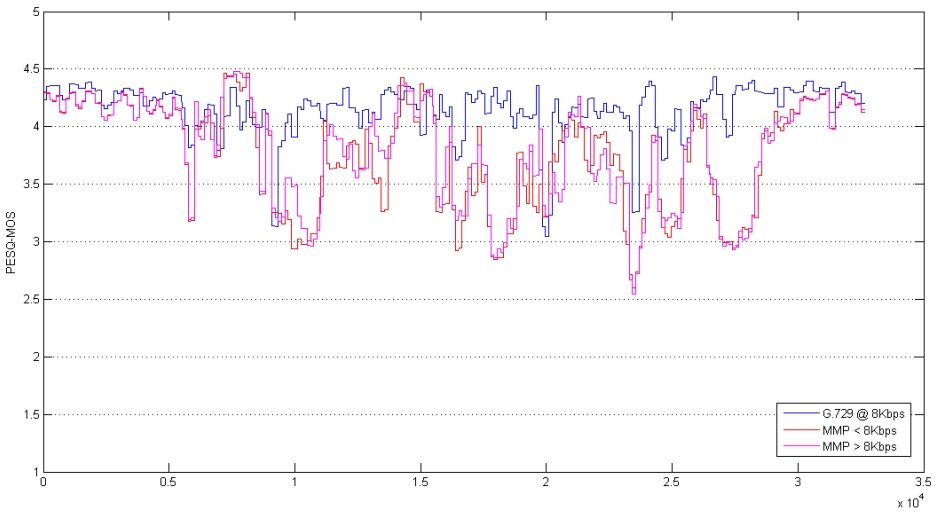


Figura 6.5: Notas PESQ-MOS quadro-a-quadro para a frase exemplo us39.wav codificada à taxa de 1 bit/amostra, com os blocos de silêncio substituídos pelas amostras codificados através do CELP.

de 8 kbps calculada como

$$T_{bps} = \left(\frac{[\text{tamanho do arquivo MMP}]_{Bytes} - [\text{tamanho do cabeçalho}]_{Bytes}}{[\text{duração ativa}]_{seg}} \right) \times 8 \tag{6.4}$$

e os resultados podem ser encontrados na Tabela 6.2. Como temos amostras decodificadas apenas dos blocos ativos, recompomos as parcelas de trechos de silêncio com a mesma técnica do transplante. As etapas de filtragem aplicadas no decodificador, como o filtro redutor de blocagem e o pós-filtro, não exercem influência nos blocos de silêncio. Os filtros são aplicados apenas nos trechos codificados pelo MMP.

Tabela 6.2: Resultados do MMP Voz - IX (versão VIII adaptada para codificar apenas trechos ativos à taxa de 8 kbps)

Sinal	PESQ-MOS
$\hat{\mathbf{s}}$	2,90
$\hat{\mathbf{s}}_\mathbf{b}$	2,93
$\hat{\mathbf{s}}_\mathbf{f}$	2,80
G.729B	3,62

De alguma forma, esse resultado indica apenas que o MMP é eficiente na codificação do silêncio. Mesmo assim, ainda não podemos descrever como o MMP codifica os blocos ativos sem analisar o comportamento do algoritmo ao longo do sinal. Sabemos apenas que a cardinalidade do dicionário cresce em trechos ativos e que se mantém estável em blocos de silêncio, como vimos no capítulo anterior. Isto significa que o MMP tende a gastar mais bits em blocos ativos do que em trechos sem locução, mas para comprovar tal hipótese, precisamos avaliar o custo J.

Produzimos novas análises do comportamento MMP durante a codificação do sinal de voz. Geramos os gráficos da Figura 6.6 que ilustram as variações do custo, bem como a dimensão dos vetores escolhidos e o tipo de dicionário utilizado. Claramente, podemos ver que os resíduos de blocos sonoros e surdos são codificados com custos maiores do que os resíduos de blocos de silêncio. Isto porque mais bits são empregados para representar palavras-código com alta entropia e também para sinalizar mais níveis da árvore binária. Certamente, na maioria do casos de blocos ativos, a árvore binária possui mais níveis segmentados, ou seja, um número maior de nós filhos. Podemos comprovar tal fato se observarmos que o tamanho das palavras de trechos em silêncio corresponde, em sua grande parte, à maior dimensão: 16.

De fato, o MMP é um codificador de taxa variável e esta análise suporta a seguinte afirmação a respeito dos sinais com transplantes: quando forçamos o MMP

a operar na média à 8 kbps apenas processando blocos com locução, na verdade, reduzimos a taxa de bits para essas amostras. Na configuração anterior, o MMP codificava eficientemente os trechos de silêncio e podia gastar mais bits nos outros blocos para alcançar a média de 8 kbps. Nessa nova configuração isso não ocorre e, portanto, geramos sinais com qualidade inferior, o que justifica a queda nas notas PESQ-MOS.

6.2 Perceptualidade no Domínio do Tempo

Para contornar o problema de usar informações perceptuais no domínio do tempo, a partir de vetores com dimensões pequenas, propomos incluir a quantização escalar não-uniforme definida pela lei μ na etapa de casamento de padrões. Desta forma, esperamos que os erros de quantização inseridos durante a aproximação dos vetores estejam distribuídos pelas amplitudes do sinal conforme sugere o padrão ITU-T G.711. Com esta tentativa, esperamos reduzir as distorções no domínio perceptual. Para isso, precisamos que a etapa de quantização escalar $\mathcal{Q}_\mu$ seja aplicada às amostras dos vetores recuperados $\mathbf{s_r}$ definidos como

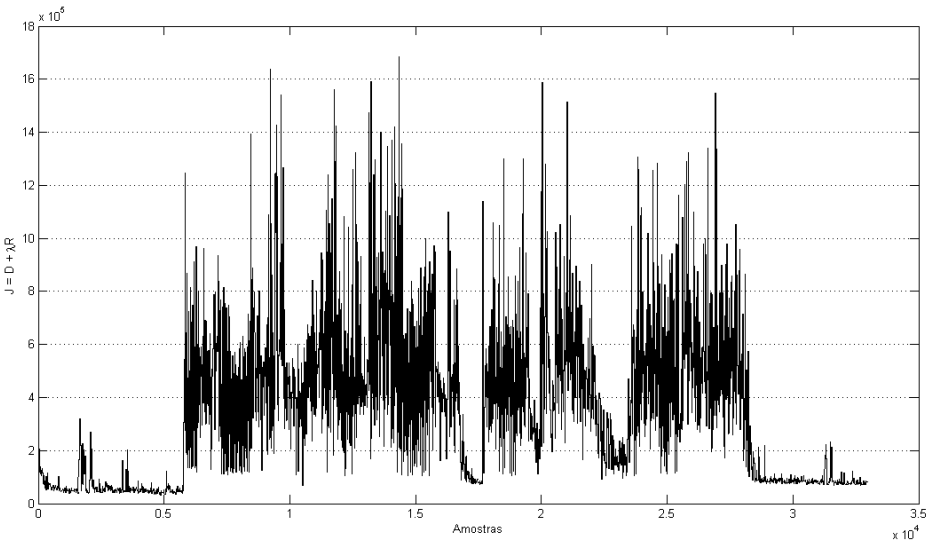
$$\mathbf{s_r} = \mathbf{X^{p_k}} + \tilde{\mathbf{s}}. \tag{6.5}$$

Incorporar a lei μ no casamento de padrões do MMP significa que estamos levando as amostras dos resíduos e dos vetores no dicionário a níveis cuja distorção no domínio perceptual é mínima. O diagrama em blocos do novo codificador pode ser encontrado na Figura 6.7.

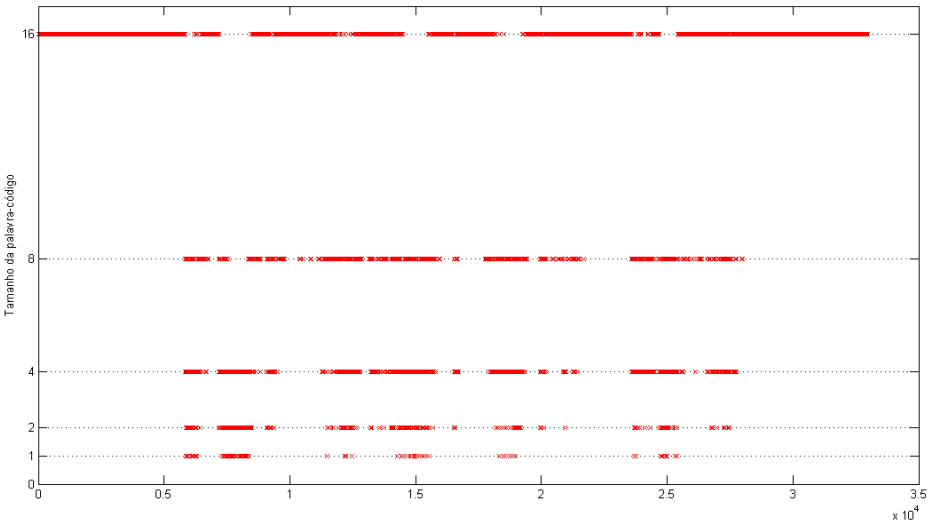
Ao quantizarmos o nosso banco de frases pela lei μ, antes do processamento MMP, garantimos a qualidade quase máxima pelo critério PESQ além de uma relação sinal-ruído acima de $37,0$ dB, como podemos ver na Tabela 6.3, ou seja, não comprometemos a qualidade final do sinal fonte. Nossa proposta, embora inovadora, não superou os resultados anteriores. A qualidade média do sinais decodificados se manteve igual à última configuração. O mesmo comportamento foi observado na relação sinal-ruído.

Tabela 6.3: Resultado da quantização pela lei μ em todo o banco de frases sem o processamento MMP

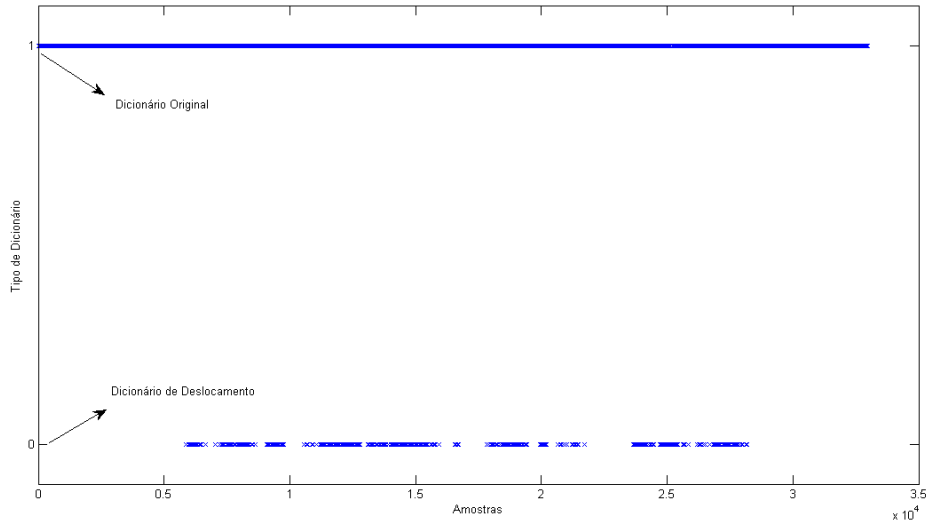
Sinal	SNR (dB)	PESQ-MOS
$\mathbf{s}$	37,08	4,37



(a) Custo $J = D + \lambda R$.



(b) Dimensão da palavra-código.



(c) Tipo de dicionário

Figura 6.6: Análise do algoritmo MMP ao longo da codificação do sinal de voz.

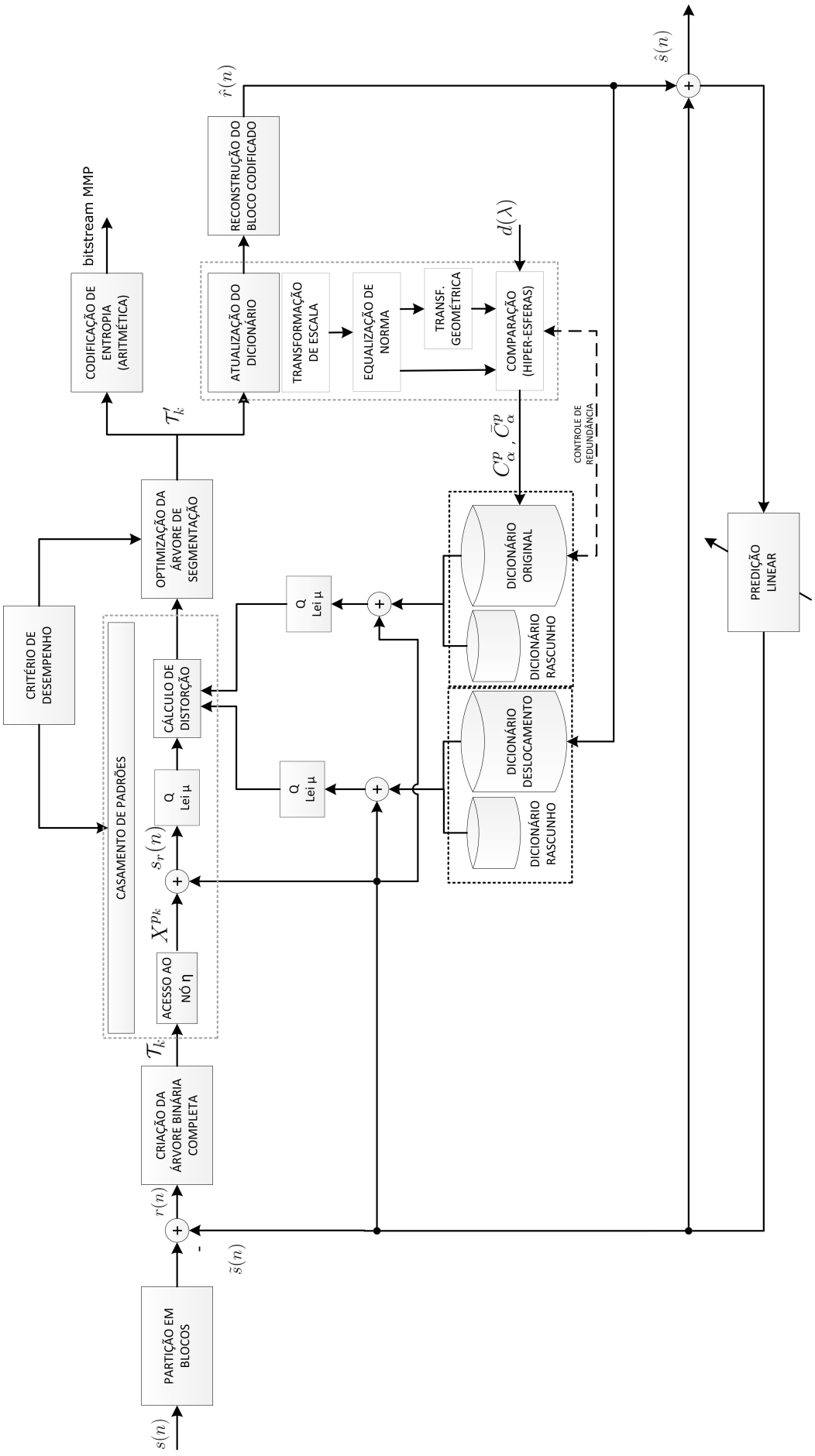


Figura 6.7: Diagrama em blocos do codificador com a lei *mu* incorporada ao processo de casamento de padrões.

6.3 Outras Contribuições

6.3.1 PESQ no *loop* MMP

Os sons que são percebidos quando escutamos palavras com /x/, /ch/ ou /sh/ são quase imperceptíveis para a média da população. Da mesma forma, pouco diferem sons nasalados de /n/ e /m/ em nosso sistema auditivo. Conseguimos distinguir tais sons, porque levamos em conta, intuitivamente, outros fatores linguísticos como a gramática, o contexto da frase, a intonação da fala, etc. Mas a representação no tempo desses fonemas podem variar bastante. Como o MMP codifica a forma de onda do sinal, o processo de casamento de padrões é muito vulnerável à qualquer mudança no domínio do tempo, uma vez que utilizamos o erro quadrático no cálculo da distorção. Vejamos, se o MMP busca codificar um padrão senoidal,

$$s(n) = A\ \mathrm{sin}(\omega_0) \tag{6.6}$$

e encontra um vetor no dicionário de mesma amplitude, mas com diferença de fase φ,

$$\hat{s}(n) = A\ \mathrm{sin}(\omega_0 + \varphi) \tag{6.7}$$

o erro quadrático varia à medida que φ varia. Por exemplo, no pior caso, quando $\varphi = 180°$, obtemos o maior erro quadrático, conforme exemplificado na figura 6.8.

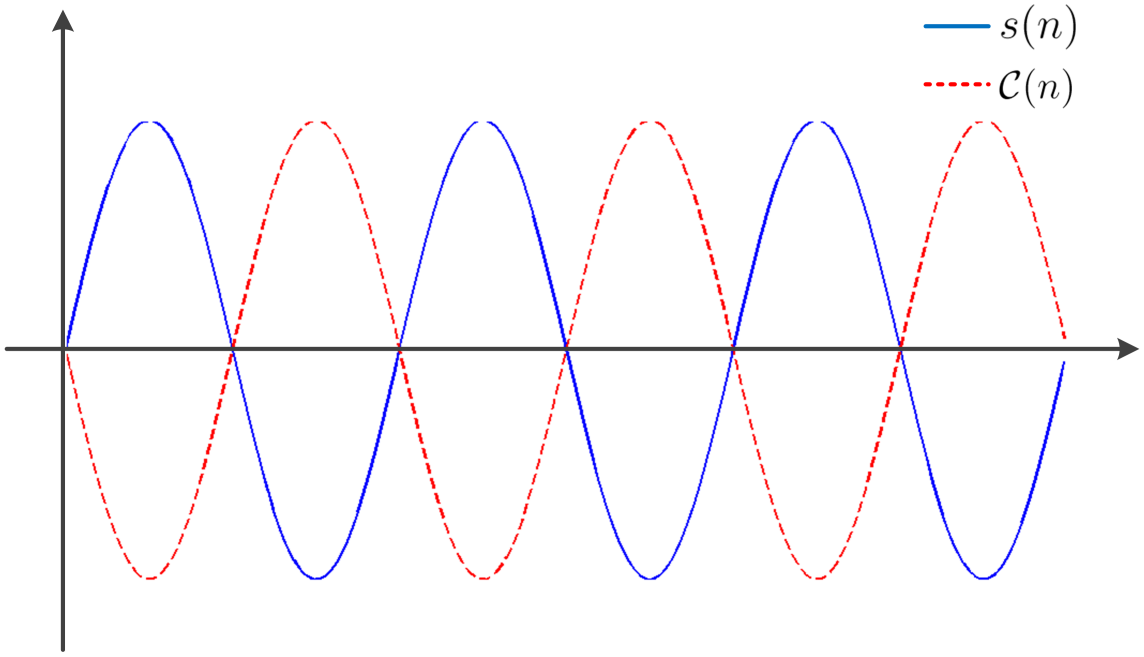


Figura 6.8: A linha pontilhada indica um vetor candidato à aproximação do padrão da linha contínua, segundo um critério perceptual. No entanto, a distorção é máxima se considerarmos o cálculo do erro quadrático.

No entanto, para nosso sistema auditivo, variações de fase são menos perceptíveis

do que as de amplitude, o que poderia fazer do padrão existente no dicionário $\hat{s}(n)$ um ótimo candidato sem necessidade de segmentar a árvore binária. Isto significa que se mudássemos o critério de distorção poderíamos melhorar a condição perceptual e a eficiência do MMP.

Com esse o objetivo alteramos o critério de distorção para que D passe a ser calculado a partir do algoritmo PESQ, mesmo ao preço de um custo computacional muito maior, pois as etapas de cálculo são bem mais complexas do que o erro quadrático. Fizemos as seguintes adaptações no algoritmo:

- O distorção é calculada a partir de dois sinais com 256 amostras cada: um de referência e outro degradado.
- O sinal de referência é composto pelas amostras do sinal original $s(n)$, centralizado no bloco de interesse.
- O vetor que está sob avaliação X^{p_k} é composto pelas amostras recuperadas, ou seja, amostras estimadas $\tilde{s}(n)$ somadas aos padrões de resíduo dos dicionários (original e deslocamento).
- Para criarmos o sinal degradado substituímos o vetor sob avaliação numa cópia do sinal original.
- Depois, submetemos os dois sinais (referência e degradado) à análise PESQ que retorna com uma distorção D_i^{PESQ}, para cada índice i do dicionário. Repetimos o processo de forma recursiva para todos os nós da árvore binária.
- Durante a otimização da árvore, realizamos um processo semelhante ao casamento de padrões que descrevemos abaixo:

    1. Concatenamos os vetores dos nós filhos que foram previamente selecionados na etapa de casamento de padrões.
    2. Substituímos esse novo vetor numa cópia do sinal original e somamos à ele as amostras estimadas.
    3. Em seguida, submetemos à análise PESQ este novo sinal degradado, que retorna com a distorção agregada dos nós filhos $D^{PESQ}_{(\eta_{2j+1}:\eta_{2j+2})}$. O novo valor compõe o custo J_{filhos}.

Os passos seguintes são os mesmos do MMP original: caso o custo de representação dos nós filhos seja maior que o custo do nó pai, o algoritmo poda os ramos do nó pai, e o processo se repete para toda a árvore.

Como o seu custo computacional é muito superior, rodamos este novo algoritmo para apenas 5 frases. Novos valores de λ foram encontrados, mas a qualidade piorou.

O novo algoritmo, da maneira que descrevemos com o PESQ no loop do MMP, não converge para a melhores palavras mesmo com $\lambda = 0$. Embora os primeiros resultados não sejam tão animadores, parece que temos um campo ainda a ser mais explorado no futuro.

6.3.2 Filtros de Pré-Ênfase e Dê-Enfase

Alguns codificadores de voz baseados em transformadas utilizam em sua grande maioria filtros de pré ênfase e de ênfase. Nesses casos, a codificação é feita em outro domínio, cujos coeficientes são calculados através de uma transformada unitária e inversível T

$$\mathbf{S} = \mathbf{Ts} \tag{6.8}$$

Essa condição garante que as amostras sejam inteiramente recuperadas

$$\mathbf{s} = \mathbf{T^{-1}S}, \tag{6.9}$$

onde $T^{-1} = T^H$. O símbolo ^{H} denota a forma hermitiana da matriz T (complexo-conjugado transposto). As transformadas mais comuns são a DCT (*Discrete Cosine Transform*), DFT (*Discrete Fourier Transform*), WHT (*Walsh-Hadamard Transform*) e KLT (*Karhunen-Loéve Transform*) que buscam descorrelacionar as amostras no domínio do tempo, produzindo coeficientes que representam, em sua maioria, a forma espectral do sinal. Como em geral esses algoritmos priorizam os componentes de baixa frequência na estratégia de alocação de bits, à taxas baixas, é comum que componentes de alta frequência sejam descartados. O filtro de pre-ênfase $H(z)$ que responde como um passa-altas pode ser empregado no codificador para aumentar a energia dessa região de alta frequência do espectro. No decodificador, o filtro de-ênfase $G(z)$ é empregado para compensar a pré-distorção, ou seja, $G(z) = H(z)^{-1}$. É comum que esses filtros sejam de primeira ordem da forma

$$H(z) = 1 - \rho z^{-1}, \tag{6.10}$$

com $0,9 < \rho < 1$. Apesar do algoritmo MMP não usar decomposições de frequência, incluímos essa filtragem no processamento MMP de acordo com o diagrama em blocos da figura 6.9, variando o valor de ρ, com o objetivo de melhorar a resolução espectral do sinal reconstruído, mas, neste caso, não produzimos resultados satisfatórios.

Durante a codificação do MMP à nossa taxa alvo, percebemos que muitas palavras de dimensões grandes, provenientes da expansão dos níveis do dicionário inicial são utilizadas. Esses vetores, como possuem valor constante, contribuem apenas para as componentes de baixa frequência, ou no nível DC para mais exato (Figura

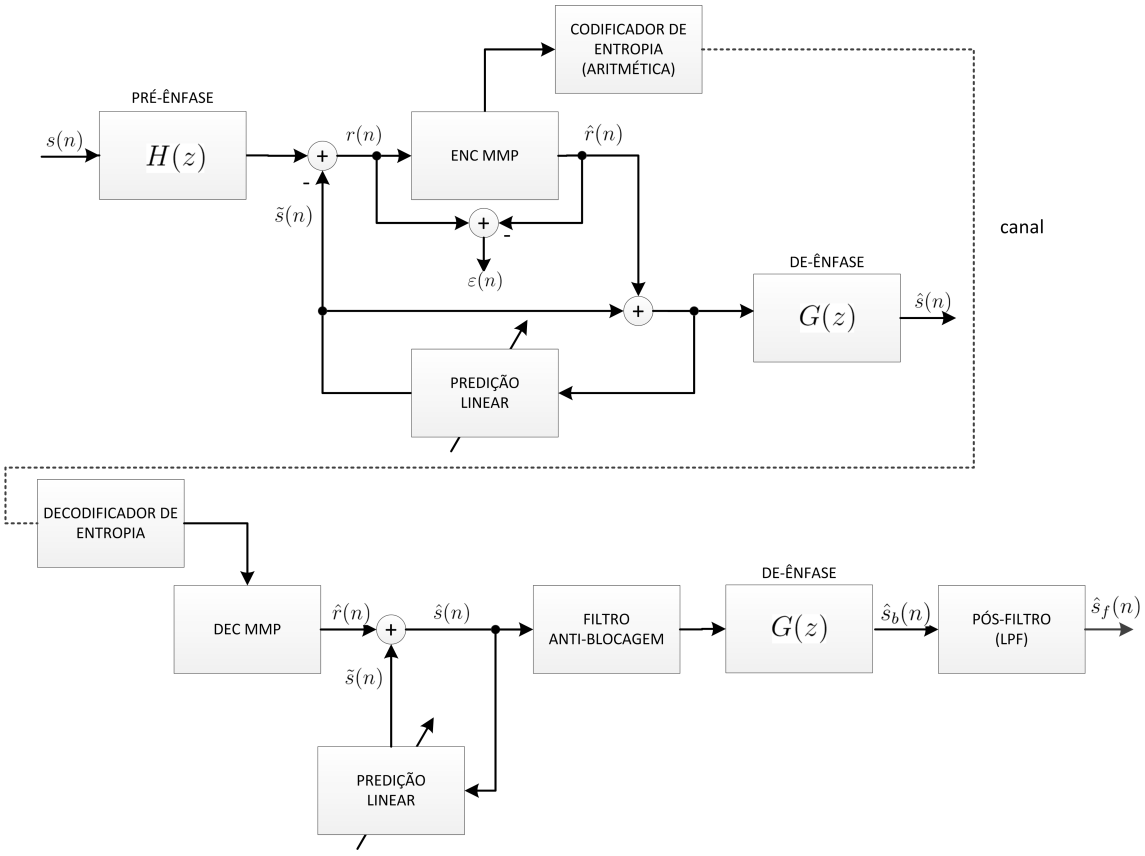


Figura 6.9: Diagrama em blocos do algoritmo MMP Voz que inclui o filtro pré-ênfase $H(z)$ e de-ênfase $G(z)$.

6.10). Ao aumentarmos a energia das componentes de alta frequência, forçamos que a árvore binária seja mais segmentada, ou seja, gaste mais bits para representá-la e, consequentemente, maiores valores de λ são necessários. Em outras palavras, isso nos conduz novamente a escolha pelas mesmas palavras com níveis constantes.

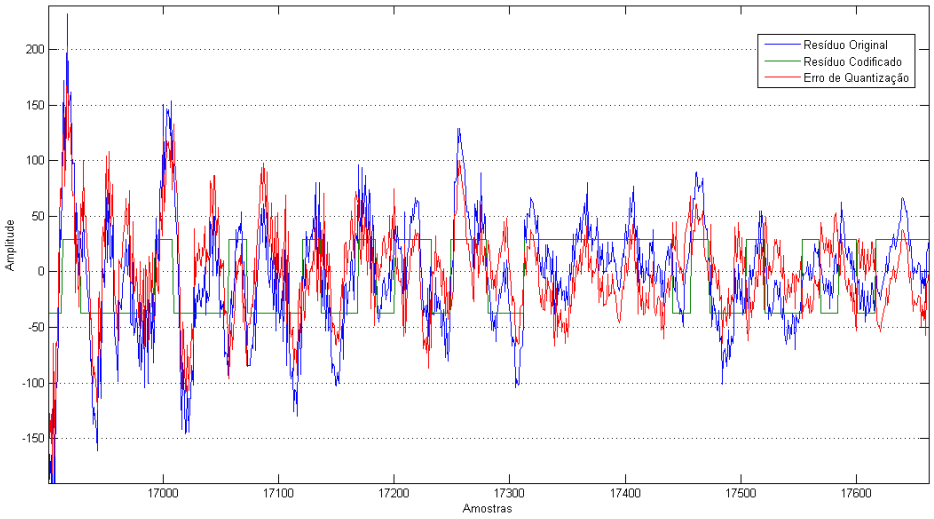


Figura 6.10: Comparação entre os resíduos original e codificados pelo MMP à taxa de 1 bit/amostra numa parcela do sinal us39.wav. Como os vetores usados no casamento de padrões pertencem à maior dimensão do dicionário, eles aproximam grosseiramente as variações do resíduo original. A consequência é a forma espectral do sinal codificado sem os componentes de alta frequência originais.

6.3.3 Escalamento

Como a faixa dinâmica do nosso sinal de voz é grande, $-2^{14} + 1 < s(n) < 2^{14}$ introduzimos uma nova etapa de pré-quantização no sinal fonte. Dividimos cada amostra do sinal original pelo fator 2 visando reduzir a faixa dinâmica a valores $-2^{13} + 1 < s < 2^{13}$ e ao mesmo tempo a quantidade de palavras no dicionário para diminuir a entropia dos vetores. No entanto, nossos resultados médios de SNR e nota PESQ-MOS também não sofreram alteração. Essa técnica apenas diminuiu os valores absolutos de distorção D. Em outras palavras, menores valores de λ foram encontrados para ajustar a codificação à taxa alvo e, portanto, não contribuiu no processo de casamento de padrões. O G.729 utiliza o mesmo padrão de escalamento com fator 2, mas com o objetivo de se precaver contra problemas de *overflows*.

6.3.4 Pré e Pós-Processamento

A ideia por trás do pré-processamento é adaptar o sinal em favor do codificador. Em nosso caso incluímos o filtro de pré processamento $H_{h1}(z)$ de ordem 2 definido

no G.729, cuja resposta em frequência é um passa-altas (HPF). Esse passa-altas tem frequência de corte em 140 Hz e previne contra componentes indesejáveis de baixa frequência. $H_{h1}(z)$ é definido como

$$H_{h1}(z) = 2\left(\frac{0,46363718 - 0,92724705z^{-1} + 0,46363718z^{-2}}{1 - 1,9059465z^{-1} + 0,9114024z^{-2}}\right) \tag{6.11}$$

O fator 2 corrige o escalamento que existe no G.729, já previsto nos coeficientes do filtro. Nesta seção vamos avaliar apenas o efeito do pré-processamento sem reduzir a escala do sinal. E, como está definido na equação (6.11), $H_{h1}(z)$ possui a resposta em frequência ilustrada na figura 6.11

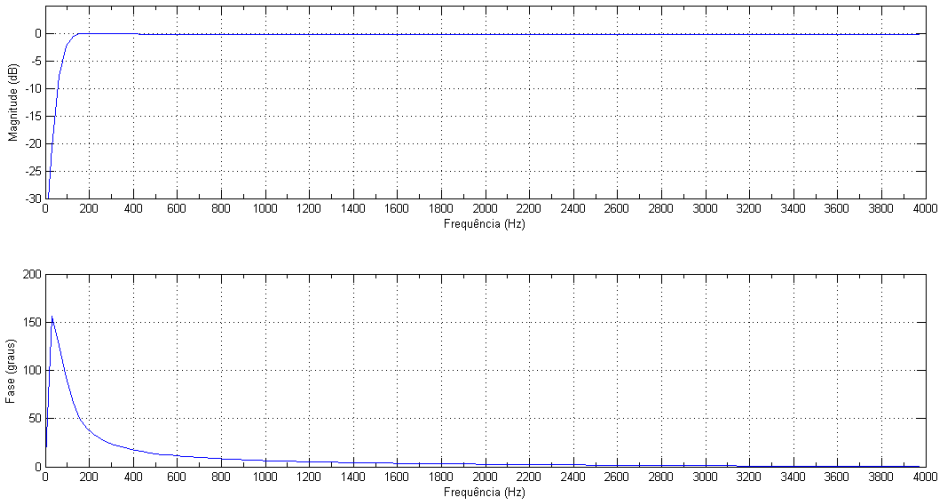


Figura 6.11: Resposta em frequência da magnitude (dB) e fase do filtro $H_{h1}(z)$.

Na etapa de pós-processamento, implementamos apenas o último estágio do pós-processamento do G.729, que inclui o filtro de ordem 2

$$H_{h2}(z) = \frac{0,93980581 - 1,8795834z^{-1} + 0,93980581z^{-2}}{1 - 1,9330735z^{-1} + 0,93589199z^{-2}} \tag{6.12}$$

A resposta em frequência de $H_{h2}(z)$ também é um passa-altas (HPF), com frequência de corte em 100Hz como demonstrado na figura 6.12

Com a inclusão dessas etapas de pré e pós processamento, produzimos melhores resultados do MMP Voz para a taxa alvo de 8 kbps, apresentados na tabela 6.4. A maior nota média PESQ-MOS até o momento de **3,25** foi alcançada ao preço de uma queda brusca na relação sinal-ruído proveniente das etapas de pré e pós processamento.

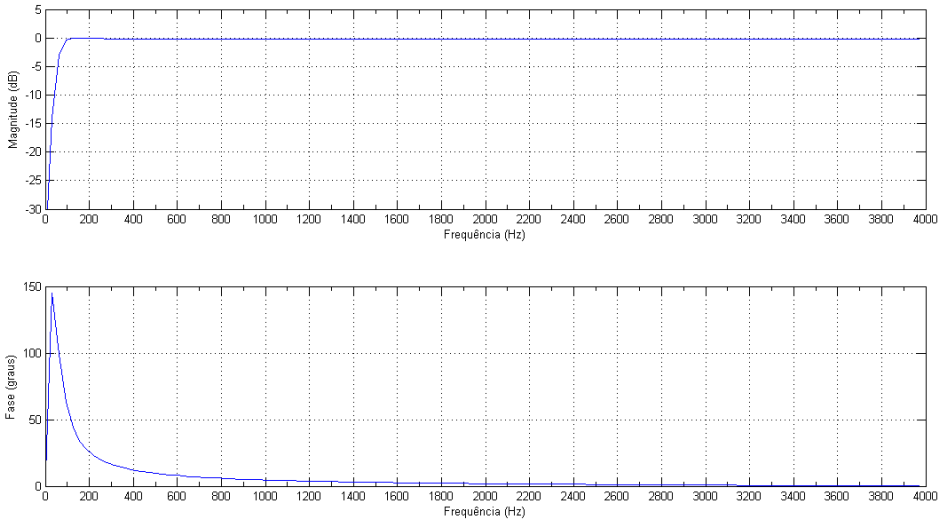


Figura 6.12: Resposta em frequência da magnitude (dB) e fase do filtro $H_{h2}(z)$.

Tabela 6.4: Resultados do MMP Voz X à taxa de 8 kbps (1 bit/amostra)

Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	3,37	3,17
$\hat{\mathbf{s}}_\mathbf{b}$	3,37	3,20
$\hat{\mathbf{s}}_\mathbf{f}$	3,77	**3,25**
G.729 (main)	5,58	3,74

6.3.5 Novo Pós-Filtro

Temos empregado diferentes técnicas na busca para superar o desempenho perceptual no MMP Voz, seja quando introduzimos novas análises que nos permitem compreender detalhes do comportamento do MMP Voz, seja quando atuamos em etapas de filtragem. Nesse sentido, procuramos encontrar um filtro passa baixas que reduza ainda mais a influência de distorções em regiões de alta frequência, com o compromisso de manter a inteligibilidade das sentenças de voz e evitar fenômenos indesejáveis.

Para isso, recalculamos novos coeficientes de acordo com o *script* descrito no Apêndice D.2. A Figura 6.13 ilustra a resposta em frequência das duas versões de filtro passa baixas e, nesta comparação, notamos que a atenuação média com o novo pós-filtro (linha contínua) é maior do que a atenuação com o filtro original (linha pontilhada) descrito na Seção 5.2, para toda a faixa de interesse.

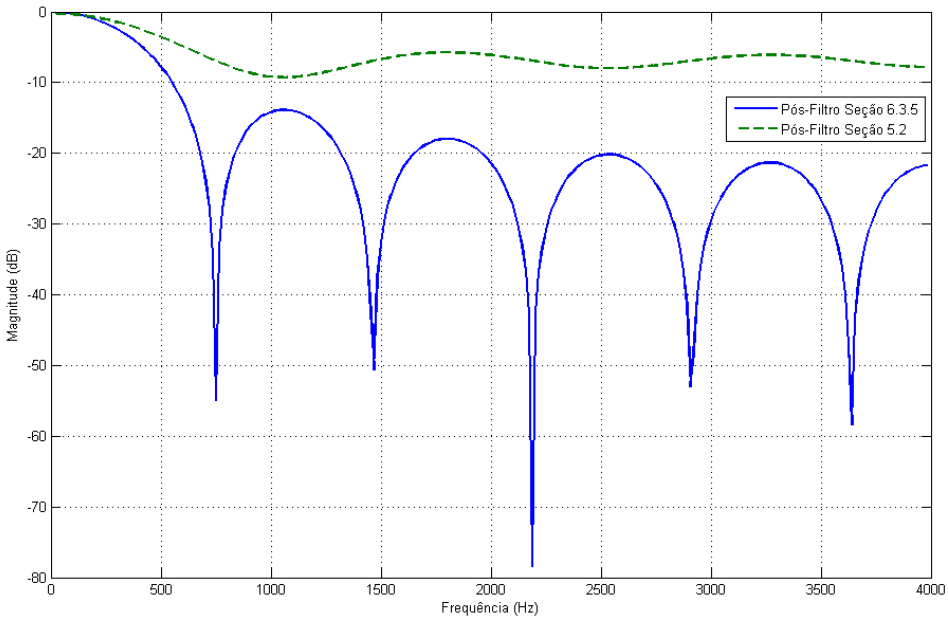


Figura 6.13: Resposta em frequência da magnitude (dB) do novo pós-filtro.

Logo após termos encontrado o novo pós-filtro, realizamos duas simulações para avaliar seu desempenho. Na primeira, substituímos inteiramente os coeficientes apresentados na Seção 5.2 pelos coeficientes recém calculados que podem ser vistos na Tabela 6.5.

Em seguida, aplicamos a etapa de filtragem no sinal $\hat{\mathbf{s}}_\mathbf{b}$ gerado pela versão MMP Voz X para produzir $\hat{\mathbf{s}}_\mathbf{f}$ em todos os sinais do banco de frases. Obtivemos uma melhora bastante significativa e tal avanço nos permitiu alcançar o valor médio de **3,45** na nota PESQ-MOS, a maior encontrada neste trabalho. A Tabela 6.6 apresenta os resultados finais com a versão MMP Voz X.

Tabela 6.5: Tabela com os novos coeficientes do filtro FIR passa-baixas usado na etapa de pós-filtragem.

Coeficiente FIR	Valor
h(0)	0,08451480743058200
h(1)	0,08828105720097000
h(2)	0,09128806440000900
h(3)	0,09347830733808200
h(4)	0,09480962377009000
h(5)	0,09525627972053299
h(6)	0,09480962377009000
h(7)	0,09347830733808200
h(8)	0,09128806440000900
h(9)	0,08828105720097000
h(10)	0,08451480743058200

Tabela 6.6: Melhores resultados do MMP Voz à taxa de 8 kbps (1 bit/amostra)

Sinal	SNR (dB)	PESQ-MOS
$\hat{\mathbf{s}}$	3,37	3,17
$\hat{\mathbf{s}}_\mathbf{b}$	3,37	3,20
$\hat{\mathbf{s}}_\mathbf{f}$	4,09	**3,45**
G.729 (main)	5,58	3,74

A segunda simulação teve o objetivo de avaliar os avanços reais deste trabalho e, para isso, fizemos a mesma substituição na etapa de pós filtragem. No entanto, usamos o sinal $\hat{\mathbf{s}}_\mathbf{b}$ gerado pela primeira versão do MMP Voz (MMP Voz I). Os resultados que encontramos estão na Tabela 6.7.

Tabela 6.7: Comparação de desempenho entre os filtros aplicados na etapa de pós processamento na versão MMP Voz I

Versão MMP Voz	Sinal	SNR (dB)	PESQ-MOS	Pós-filtro
MMP Voz I	$\hat{\mathbf{s}}_\mathbf{b}$	19,41	2,98	Nenhum
	$\hat{\mathbf{s}}_\mathbf{f}$	12,58	3,03	Com pós-filtro original (Seção 5.2)
	$\hat{\mathbf{s}}_\mathbf{f}$	6,83	3,25	Com novo pós-filtro (Seção 6.3.5)

Vemos que, ao preço da redução na relação sinal-ruído de aproximadamente 6 dB, superamos o valor na nota subjetiva em 0,22 na escala PESQ-MOS. Esse ganho é comparável ao ganho que registramos com a nova filtragem dos sinais gerados pelo MMP Voz X, onde saímos de 3,25 e alcançamos a nota média 3,45, o que indica que

as contribuições deste trabalho conferem ao MMP Voz uma melhora real da ordem de 0,2 na escala PESQ-MOS.

6.4 Conclusões

Neste capítulo apresentamos análises perceptuais importantes para entendermos o comportamento de codificação MMP ao longo do sinal, o que motivou a inclusão de novas estruturas ao MMP. Introduzimos critérios perceptuais no processo de casamento de padrões e nas etapas de pré e pós processamento. Com isso, atingimos a maior qualidade objetiva do MMP Voz na escala PESQ-MOS, **3,45** a partir do novo pós-filtro. Esta condição foi alcançada com os seguintes parâmetros:

- Sinal de entrada para o MMP: Resíduo original (sem quantização)
- Dimensão do bloco: 16
- Algoritmo LP: LS_C
- Ordem do preditor: 40
- Tamanho do conjunto de treinamento: 256
- Função da janela: Retangular
- Cardinalidade do Dicionário inicial: 256
- Casamento de Padrões do Dicionário Original: Com análise de dependência (Dicionário Rascunho)
- Casamento de Padrões do Dicionário de Deslocamento: Com análise de dependência (Dicionário de Deslocamento Rascunho).
- Métrica de Distorção (D): Erro quadrático (SE)
- Critério de Desempenho: $J = D + \lambda R$
- Otimização da Árvore Binária: Algoritmo RDI (RD Intermediário)
- Atualização do Dicionário
  - Controle de Redundância: raio da hiper-esfera $d = 256$
- Filtro *Anti-blocking*: aplicado às amostras reconstruídas ($\hat{s}(n)$).
- Pós-Filtro: Novos coeficientes do filtro FIR definido na Seção 6.3.5.

A Tabela 6.6 apresenta os resultados finais encontrados nesse trabalho. Contudo, observamos que há uma variação na qualidade subjetiva ainda que medida com um número reduzido de observadores e em poucas sentenças. Quando escutamos as frases filtradas com o novo pós-filtro percebemos sons mais abafados, um fenômeno que é consequência da atenuação mais rígida, cuja resposta em frequência do novo pós-filtro atenua mais fortemente as regiões de altas frequências se comparado ao pós-filtro da Seção 5.2. Isso faz com que a qualidade percebida seja diretamente afetada e, aparentemente, gere a sensação de perda de informação espectral. O desempenho, no entanto, para o algoritmo PESQ é melhor. Ainda neste capítulo comparamos os dois pós-filtros aplicados aos sinais $\hat{\mathbf{s}}_\mathbf{b}$ gerados nas versões MMP Voz I e X e concluímos que as contribuições deste trabalho representam, de fato, uma evolução no MMP Voz, com ganho real da ordem de 0,2 na escala PESQ-MOS.

Capítulo 7

Conclusões

7.1 Principais Resultados

Este trabalho investigou o processo de codificação de sinais de voz a partir de técnicas de casamento de padrões recorrentes. O algoritmo MMP foi utilizado como base nessa dissertação por proporcionar uma codificação em blocos muito eficiente. Elaboramos uma forma didática para descrever as ferramentas do MMP tendo como base os diagramas em blocos apresentados ao longo da dissertação. Essa nova organização permitiu que contribuições específicas fossem feitas de forma mais simples e objetiva à estrutura básica do algoritmo MMP.

Todo o trabalho foi orientado a buscar a melhor qualidade perceptual do sinal codificado à taxa de 8 kbps. A medida de qualidade foi calculada através do algoritmo PESQ, uma recomendação da UIT que reconhecidamente é a métrica mais avançada até o presente momento para avaliar padrões de codificação de voz.

Inicialmente, decidimos implementar inteiramente o codificador e o decodificador de voz, o que nos permitiu adaptar as ferramentas e adicionar novas funcionalidades ao algoritmo MMP. De fato, isso foi essencial para que as inovações fossem testadas e validadas com segurança. Outro fato relevante que contribuiu para melhorar o algoritmo foi a abordagem de avaliar as métricas de desempenho em diferentes pontos: na saída do codificador, na saída do filtro redutor de blocagem e na saída do pós-filtro. Além disso, com o desenvolvimento, fomos capazes de gerar ferramentas de análises e dados intermediários para avaliar o comportamento do MMP ao longo do sinal.

Os parâmetros que definiram nosso ponto de partida se basearam em [6], onde partimos de uma qualidade média de 3,03 na escala PESQ-MOS para nosso banco de frases. Novos algoritmos de predição linear foram propostos e analisados. Incorporamos à estrutura do MMP o algoritmo que chamamos de *Least Squares C* por produzir o melhor desempenho. Outras contribuições que merecem destaque são a

inclusão do dicionário de deslocamento rascunho, o novo controle de redundância baseado no critério de distância das hiper-esferas, bem como a inclusão de palavras simétricas, a partir da transformada geométrica. Ao fim do capítulo 5, conseguimos melhorar a performance do MMP Voz em 0,17, alcançando a nota média PESQ-MOS de 3,20.

As contribuições no domínio perceptual foram apresentadas no capítulo 6. A partir de análises quadro-a-quadro, identificamos que as notas PESQ-MOS dos blocos de silêncio das frases codificadas pelo MMP estão sempre abaixo das notas CELP. Porém, se observarmos o comportamento do custo J, concluímos que, na verdade, o MMP é eficiente na codificação destes trechos o que permite usar mais bits na codificação de trechos ativos. Além disso, fizemos propostas para incluir critérios perceptuais no *loop* do casamento de padrões, como é o caso da quantização pela lei μ e das distorções calculadas pelo PESQ. Embora não tenham produzido resultados satisfatórios, alcançamos um melhor entendimento do funcionamento geral do MMP Voz, mostrando que essa linha de trabalho trata-se de um campo ainda a ser explorado. Outras contribuições perceptuais foram propostas nas etapas de pré e pós processamento. Com um novo pós-filtro, atingimos a maior qualidade objetiva do MMP Voz: 3,45 na escala PESQ-MOS, ao preço de uma queda de 9,09 dB[1] na relação sinal-ruído.

7.2 Trabalhos Futuros

Essa dissertação continuou e expandiu os primeiros resultados de [6] no contexto de codificação de voz usando MMP. Nossos testes forneceram mais compreensão do mecanismo de aprendizado do algoritmo. Concluímos ainda, que muito a ser explorado neste campo, em particular, podemos citar as seguintes linhas de continuação do presente trabalho:

- O dicionário inicial poderia ser treinado com objetivo de incluir inicialmente palavras da maior dimensão e suas versões contraídas seriam adicionadas às escalas menores. Isso ajudaria o casamento nos blocos de resíduo que tendem a ter comportamento de ruído branco. Principalmente em blocos de baixa energia, à taxa de 8 kbps, observamos o uso de palavras código com grandes dimensões, mas com valores constantes, ou seja, são palavras provenientes da expansão do dicionário inicial original. Isso afeta a eficiência do algoritmo, já que o aprendizado é baseado nas palavras previamente codificadas.

[1]Esse valor é a diferença entre a relação sinal-ruído média do sinal $\hat{\mathbf{s}}_\mathbf{f}$ obtido na versão MMP Voz VIII com o pós-filtro original (Seção 5.2) e o sinal $\hat{\mathbf{s}}_\mathbf{f}$ gerado pelo MMP Voz X com novo pós-filtro (Seção 6.3.5).

- Os sub-contextos (partições) nas escalas do dicionário poderiam ser criados a partir do critério de sonoridade dos quadros de silêncio, sonoros e surdos, ao invés de se basearem apenas na escala de origem.

- Para superar a dificuldade de obter características perceptuais no domínio do tempo a partir de vetores pequenos, poderíamos compor o valor da distorção D com novos fatores além do erro quadrático, como a diferença de cruzamentos do valor médio. Ou seja, $D = \alpha D_{SE} + (1 - \alpha) D_{MC}$, onde D_{SE} representa o erro quadrático, D_{MC} representa a nova distorção e $0 < \alpha < 1$.

- Incluir as etapas do pós-filtro adaptativo do G.729.

- Avaliar técnicas de *dithering* para distribuir a energia do ruído ao longo do sinal de voz [32]. Distorções muito concentradas no domínio do tempo são prejudiciais para nossa audição.

Referências Bibliográficas

[1] HAYES, D. *Historical Atlas of the United States, with Original Maps*. 2006.

[2] DA SILVA MAIA, R. *Codificação CELP e Análise Espectral de Voz*. Tese de Mestrado, COPPE/UFRJ, Mar 2000.

[3] RODRIGUES, N. M. M. *Multiscale Recurrent Pattern Matching Algorithms for Image and Video Coding*. Tese de Doutorado, Departamento de Engenharia Eletrônica e de Computadores - Universidade de Coimbra, Jul 2008.

[4] SCHAFER, R. W., RABINER, L. R. "Digital Representations of Speech Signals". In: *Proceedings of IEEE*, v. 63, pp. 662–677, Abr 1975.

[5] ITU. *ITU-T REC. G.711: Pulse Code Modulation (PCM) of Voice Frequencies*. Relatório técnico, International Telecommunications Union, 1972.

[6] DA SILVA PINAGÉ, F. *Codificaç ão de Voz usando Recorrência de Padrões Multiescalas*. Tese de Doutorado, COPPE/UFRJ, Set 2011.

[7] DE CARVALHO, M. B. *Compressão de Sinais Multidimensionais usando Recorrência de Padrões Multiescala*. Tese de Doutorado, COPPE/UFRJ, Rio de Janeiro, RJ, Brasil, 2001.

[8] DE LIMA FILHO, E. B. *Compressão de Imagens utilizando Recorrência de Padrões Multiescalas com Critério de Continuidade Inter-Blocos*. Tese de Mestrado, COPPE/UFRJ, Abr 2004.

[9] GRAZIOZI, D. B. *Contribuições à Compressão de Imagens com e sem perdas utilizando Recorrência de Padrões Multiescalas*. Tese de Doutorado, COPPE/UFRJ, Abr 2011.

[10] DA SILVA JÚNIOR, W. S. *Compressão de Imagens utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível*. Tese de Mestrado, COPPE/UFRJ, Dez 2004.

[11] DE LIMA FILHO, E. B. *Aplicações em Codificação de Sinais: O Casamento aproximado de Padrões Multiescalas e a Codificação Distribuída de Eletrocardiograma*. Tese de Doutorado, COPPE/UFRJ, Nov 2008.

[12] DE OLIVEIRA, A. V. C. *Codificação de Textura e Profundidade de Imagens 3-D utilizando Recorrência de Padrões Multiescalas*. Tese de Mestrado, COPPE/UFRJ, Mar 2011.

[13] ITU. *ITU-T REC. P.862: Perceptual Evaluation of Speech Quality*. Relatório técnico, International Telecommunications Union, 2002.

[14] SHANNON, C. E. "A mathematical theory of communications", *Bell Syst. Tech. Journal*, 1948.

[15] COVER, T. M., THOMAS, J. A. *Elements of Information Theory*. 2nd ed. , Wiley, 2006.

[16] SAYOOD, K. *Introduction to Data Compression*. 3th ed. , Morgan Kaufmann, 2005.

[17] WANG, Z., BOVIK, A. C. "Mean Squared Error: Love it or Leave it". In: *IEEE Signal Processing Magazine*, IEEE, Jan 2009.

[18] ITU. *ITU-T REC. P.861: Perceptual Speech Quality Measure*. Relatório técnico, International Telecommunications Union, 1996.

[19] ZWICKER, E., FASTL, H. *Psychoacustics: Facts and Models*. Springer, 2007.

[20] ANDREAS SPANIAS, T. P., ATTI, V. *Audio Signal Processing and Coding*. Wiley-Interscience, 2007.

[21] CHU, W. C. *Speech Coding Algorithms: Foundation and Evolution of Standardized Coders*. Wiley-Interscience, 2003.

[22] DUDLEY, H. "Remaking Speech", *Journal of the Acoustical Society of America*, v. 11, pp. 169–177, 1939.

[23] SPANIAS, A. S. "Speech Coding: a tutorial review". In: *Proceedings of IEEE*, v. 82, pp. 1541–1582, Out 1994.

[24] GERSHO, A. "Advances in Speech and Audio Compression". In: *Proceedings of IEEE*, v. 82, pp. 900–918, Jun 1994.

[25] JOHN R. DELLER JR., J. H. L. H., PROAKIS, J. G. *Discrete-Time Processing of Speech Signals*. Institute of Electrical and Electronics Engineers, 2000.

[26] PHILIP A. CHOU, T. L., GRAY, R. M. "Entropy-Contrained Vector Quantization". In: *IEEE Transactions on Acoustics, Speech, and Audio Processing*, v. 37, pp. 31–42, Jan 1989.

[27] TIMOTHY C. BELL, JOHN G. CLEARY, I. H. W. *Text Compression*. Prentice-Hall, 1990.

[28] DE CARVALHO, M. B. "Multidimensional signal compression using multiscale recurrent patterns", *Elservier Signal Processing*, v. 82, pp. 1559–1580, 2002.

[29] LEVINSON, N. "The Weiner RMS Error Criterion in Filter Design and Prediction", *Journal of Mathematical Physics*, , n. 25, pp. 261–278, 1947.

[30] DURBIN, J. "The Fitting of Time Series Models", *Review of Institute Inter. Statist.*, , n. 28, pp. 233–243, 1960.

[31] RABINER, L. R., SAMBUR, M. R. "An Algorithm for Determining the Endpoints of Isolated Utterances", *Bell System Tech.*, v. 54, n. 2, pp. 297–315, Fev 1975.

[32] JAYANT, N. S., RABINER, L. R. "The Application of Dither to the Quantization of Speech Signals", *Bell System Tech.*, v. 51, n. 6, pp. 1293–1304, Ago 1972.

[33] "Open Speech Repository". Disponível em: `http://www.voiptroubleshooter.com/open_speech/index.html`. Acesso em: 01 de Julho de 2012 - 12:00:00.

Apêndice A

Sinais de teste

No total, 40 sentenças de 5 idiomas foram utilizadas para validar os resultados apresentados durante a pesquisa. As características de cada frase estão listadas nas tabelas abaixo, separadas por idioma, e foram obtidas de [33]. Cada sinal tem taxa de codificação

$$T_{bps} = \underbrace{8000}_{\text{Amostras/segundo}} \times \underbrace{16}_{\text{bits/amostra}} = 128000bps \tag{A.1}$$

A.1 Chinês

Tabela A.1: Características das 8 frases no idioma chinês.

Nome	**Gênero** M/F	**Formato**	**Resolução da amostra** (bits)	**Taxa de amostragem** kHz	**Total de amostras**	**Duração** seg
ch1.wav	M	PCM	16	8 kHz	30598	3,82 s
ch10.wav	M	PCM	16	8 kHz	25689	3,21 s
ch11.wav	M	PCM	16	8 kHz	36894	4,61 s
ch17.wav	M	PCM	16	8 kHz	28124	3,52 s
ch2.wav	M	PCM	16	8 kHz	28300	3,54 s
ch20.wav	M	PCM	16	8 kHz	80432	10,05 s
ch5.wav	M	PCM	16	8 kHz	39504	4,94 s
ch6.wav	M	PCM	16	8 kHz	34987	4,37 s

A.2 Francês

Tabela A.2: Características das 8 frases no idioma francês.

Nome	**Gênero** M/F	**Formato**	**Resolução da amostra** (bits)	**Taxa de amostragem** kHz	**Total de amostras**	**Duração** seg
fr104.wav	M	PCM	16	8 kHz	75423	9,43 s
fr13.wav	M	PCM	16	8 kHz	36440	4,56 s
fr14.wav	M	PCM	16	8 kHz	49226	6,15 s
fr53.wav	M	PCM	16	8 kHz	39560	4,95 s
fr61.wav	M	PCM	16	8 kHz	33254	4,16 s
fr70.wav	M	PCM	16	8 kHz	31729	3,97 s
fr80.wav	M	PCM	16	8 kHz	41919	5,24 s
fr96.wav	M	PCM	16	8 kHz	34569	4,32 s

A.3 Hindu

Tabela A.3: Características das 8 frases no idioma hindu.

Nome	**Gênero** M/F	**Formato**	**Resolução da amostra** (bits)	**Taxa de amostragem** kHz	**Total de amostras**	**Duração** seg
in17.wav	M	PCM	16	8 kHz	51864	6,48 s
in29.wav	M	PCM	16	8 kHz	32953	4,12 s
in37.wav	M	PCM	16	8 kHz	39347	4,92 s
in4.wav	M	PCM	16	8 kHz	37073	4,63 s
in62.wav	M	PCM	16	8 kHz	41820	5,23 s
in73.wav	M	PCM	16	8 kHz	47640	5,96 s
in74.wav	M	PCM	16	8 kHz	44236	5,53 s
in80.wav	M	PCM	16	8 kHz	62223	7,78 s

A.4 Inglês

Tabela A.4: Características das 8 frases no idioma inglês.

Nome	**Gênero** M/F	**Formato**	**Resolução da amostra** (bits)	**Taxa de amostragem** kHz	**Total de amostras**	**Duração** seg
uk100.wav	M	PCM	16	8 kHz	35353	4,42 s
uk103.wav	M	PCM	16	8 kHz	31042	3,88 s
uk14.wav	M	PCM	16	8 kHz	30388	3,80 s
uk28.wav	M	PCM	16	8 kHz	29460	3,68 s
uk37.wav	M	PCM	16	8 kHz	47890	5,99 s
uk40.wav	M	PCM	16	8 kHz	70892	8,86 s
uk55.wav	M	PCM	16	8 kHz	38463	4,81 s
uk91.wav	M	PCM	16	8 kHz	31313	3,91 s

A.5 Inglês Americano

Tabela A.5: Características das 8 frases no idioma inglês americano.

Nome	**Gênero** M/F	**Formato**	**Resolução da amostra** (bits)	**Taxa de amostragem** kHz	**Total de amostras**	**Duração** seg
us21.wav	F	PCM	16	8 kHz	28305	3,54 s
us217.wav	M	PCM	16	8 kHz	41571	5,20 s
us39.wav	M	PCM	16	8 kHz	32947	4,12 s
us4.wav	F	PCM	16	8 kHz	26420	3,30 s
us52.wav	F	PCM	16	8 kHz	28060	3,51 s
us68.wav	F	PCM	16	8 kHz	24293	3,04 s
us78.wav	F	PCM	16	8 kHz	26088	3,26 s
us89.wav	F	PCM	16	8 kHz	26455	3,31 s

Apêndice B

Ordenação na Árvore Binária

A forma como uma árvore binária é ordenada depende de como ela é referenciada. O procedimento de acesso a cada nó na árvore é chamado de transversalização. Como exemplo, considere a árvore da figura B.1:

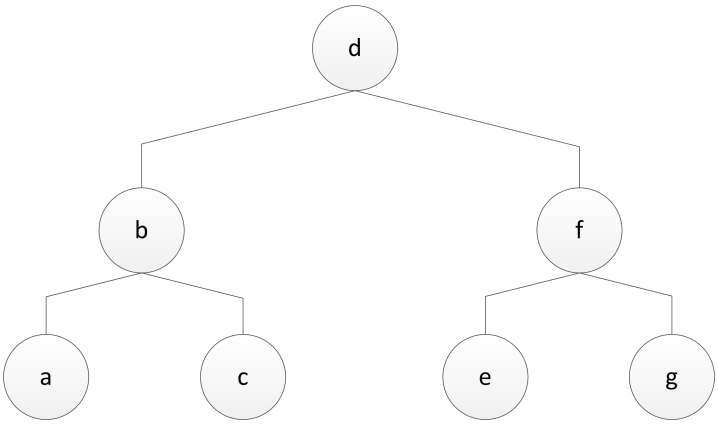


Figura B.1: Árvore binária genérica.

Existem três maneiras de percorrer uma árvore: de forma ordenada, preordenada e pós-ordenada. Na forma ordenada, primeiramente a sub-árvore da esquerda é visitada, em seguida a raiz e, por fim, a sub-árvore da direita. Na maneira preordenada, a raiz é visitada primeiro, depois a sub-árvore da esquerda e, por fim, a sub-árvore da direita. Na forma pós-ordenada, a ordem de acesso começa pela sub-árvore da esquerda, depois a sub-árvore da direita e depois a raiz.

Pelo exemplo, cada método de acesso é definido como:

- ordenada: a, b, c, d, e, f, g
- preordenada: d, b, a, c, f, e, g
- pós-ordenada: a, c, b, e, g, f, d

Apêndice C

Detalhes de Implementação

Este apêndice descreve as implementações do codificador e decodificador desenvolvidos neste trabalho, cujos cálculos são baseados em aritmética de ponto flutuante. A primeira versão dos programas foram concluídas em Junho de 2013 e teve como base a biblioteca *libmmp* desenvolvida a partir dos arquivos comuns entre os programas principais. Utilizamos a linguagem `C ANSI` no desenvolvimento da biblioteca, do codificador e decodificador. Os arquivos de cabeçalho (*.h) e os códigos fonte (*.c) estão descritos nas tabelas C.1 e C.2.

Tabela C.1: Lista dos arquivos *.h comuns ao codificar e decodificador.

Nome	Descrição
arithmetic.h	Contém os protótipos das funções utilizadas pelo codificador e decodificar aritmético
definitions.h	Definições de todas as constantes dos programas relacionadas aos parâmetros do MMP, codificador de entropia, dimensionamento dos dicionários e E/S de arquivos.
dictionary.h	Contém os protótipos das funções relacionadas à estrutura de dicionários.
dsp.h	Contém os protótipos das funções de processamento de sinais.
file_io.h	Contém os protótipos das funções de leitura e escrita (E/S) de arquivos.
memory.h	Contém os protótipos das funções de alocação e liberação de memória.
mmp.h	Contém os protótipos de todas as funções utilizadas pela estrutura básica do algoritmo MMP.
types.h	Define os tipos de variáveis e estruturas usados no programas.

Tabela C.2: Lista dos arquivos *.c.

Nome	Descrição
arithmetic.c	Funções do codificador e decodificador de entropia, atualização das estatísticas dos símbolos usados no sistema e estimação das taxas em bits por segundo dos símbolos de cada contexto.
dictionary.c	Funções de inicialização do dicionário, controle de redundância e atualização das palavras-código.
dsp.c	Funções de processamento digital de sinais.
file_io.c	Funções de leitura e escrita de arquivos, do formato WAV e dos arquivos de análise de dados intermediários.
memory.c	Funções de alocação e liberação de memória.
mmp.c	Principais funções utilizadas pelo algoritmo MMP com a criação da árvore binária, casamento de padrões, otimização RD, entre outras.
speech_mmp_enc.c	Programa principal do codificador MMP.
speech_mmp_dec.c	Programa principal do decodificador MMP.

C.1 Dicionário de Funções

O dicionário de funções da biblioteca *libmmp* está detalhado no código-fonte dos próprios arquivos *.h e *.c. Cada constante tem a descrição comentada no arquivo, bem como cada função tem comentários descrevendo os argumentos e o valor de retorno. Há comentários ao longo do código, conforme exemplo abaixo, facilitando o entendimento do algoritmo. O código abaixo é parte da função

```
1  int mmp_enc_RDI(...);
```

```
1
2   /*****************************/
3   /*    Main coding loop       */
4   /*****************************/
5   /*The following process includes prediction mode analyzes.*/
6   for (sample_index = 0;sample_index<p_wavein->numSamples;sample_index+=
        BLOCK_LENGTH)
7   {
8     if (sample_index>=TRAINING_WINDOW_LENGTH)
9     {
10      if(PREDICTION_MODE==1)
11        ls_prediction_FRED(&(p_data_out[sample_index-TRAINING_WINDOW_LENGTH]),&(
              x_est[0]),BLOCK_LENGTH);
12      else if(PREDICTION_MODE == 3)
13        ls_prediction(&(p_data_out[sample_index-TRAINING_WINDOW_LENGTH]),&(x_est
              [0]),BLOCK_LENGTH);
14
15      /*Pitch prediction*/
16      if(PITCH_PREDICTION == 1)
17        pitch_prediction(&(p_data_out[sample_index-TRAINING_WINDOW_LENGTH]),&(x_est
              [0]),BLOCK_LENGTH);
18
```

```
19        /*Store the estimated samples*/
20        for(data_index=0;data_index<BLOCK_LENGTH;data_index++)
21          x_est2[sample_index+data_index] = x_est[data_index];
22      }
23
24      p_data = &(p_wavein->data[0]);
25      p_data = p_data + sample_index;
26
27      /*Computes the residue data according to predicted samples.*/
28      for(data_index=0;data_index<BLOCK_LENGTH;data_index++)
29      {
30        x_residue[data_index] = p_data[data_index] - x_est[data_index];
31        #if (SAVE_MMP_CODING_ANALYSIS == 1)
32        x_residue_2[sample_index+data_index] = x_residue[data_index];
33        #endif
34      }
35
36      /*QuantResd feature: quantizes residues to initial dictionary entries levels.*/
37      if(SCALAR_QUANTIZATION==2)
38        scalar_quantization(&x_residue[0],BLOCK_LENGTH,&(dic_init[0]),
              INITIAL_DICTIONARY_LENGTH);
39
40      /*Builds the complete tree.*/
41      build_tree(predicted_tree,&x_residue[0],BLOCK_LENGTH,PREDICTION_MODE);
42
43      /*Since we are not sorting the encoded symbols neither flags nor indexs due to
            special version of update_model_RDI*/
44      /*called in RDI_Optimization procedure, we must copy the last state of
            statistics and dictionary to draft ones*/
45      /*before the block processing.*/
46      memcpy(&draft_stats[0],&stats[0],NUMBER_SUBDICS*sizeof(MMPStatistics));
47      memcpy(&p_draft_dic[0],&p_dic[0],NUMBER_SUBDICS*sizeof(Codebook));
48
49      /*Rate Distortion optimization.*/
50      if(MMP_DISTORTION_METRIC==0)
51        RDI_Optimization(&predicted_tree[0],p_dic,stats,p_draft_dic,draft_stats,
              lambda,p_data_residue,sample_index,1);
52      else if (MMP_DISTORTION_METRIC==1)
53        RDI_Optimization_muLaw(&predicted_tree[0],p_dic,stats,p_draft_dic,draft_stats
              ,lambda,p_data_residue,&x_est2[0],sample_index,q_muLaw,q_muLaw_levels);
54      else
55        RDI_Optimization_PESQ(&predicted_tree[0],p_dic,stats,p_draft_dic,draft_stats,
              lambda,p_data_residue,(int)sample_index,p_data_source,&x_est2[0],
              global_scale);
56
57      /*********************************/
58      /*     MMP OFFLINE ANALYZES      */
59      /*********************************/
60      /*Save the costs from optimized tree of leaves nodes, their type of dictionary
            and codeword size.*/
61      #if (SAVE_MMP_CODING_ANALYSIS == 1)
62      save_coding_analysis(predicted_tree,mmp_enc_analysis);
63      save_dictionary_growth(mmp_dic,p_dic,stats);
64      #endif
65
66      /*Encodes the optimized tree.*/
67      encode_tree_RDI(&predicted_tree[0],p_dic,stats,mmpFile,p_data_residue,
            sample_index,lambda);
68
```

```
69      int next_index=0;
70      build_coded_data_vector(p_dic, p_data_residue, sample_index, predicted_tree,&(
            p_coded_data[0]),&next_index);
71
72      /*Updates the dictionary struct based on encoded data.*/
73      update_dictionary_RDI(p_dic,stats,p_data_residue,sample_index,predicted_tree,&
            p_coded_data[0],BLOCK_LENGTH);
74
75      for(data_index=0;data_index<BLOCK_LENGTH;data_index++)
76        data_residue[sample_index+data_index] = p_coded_data[data_index];
77      for(data_index = 0;data_index<BLOCK_LENGTH;data_index++)
78        p_coded_data[data_index] += x_est[data_index];
79
80      /*Copies the encoded samples to output file.*/
81      memcpy(&(p_waveout->data[sample_index]),&(p_coded_data[0]),sizeof(p_coded_data)
            );
82
83      printf("\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b\b%8i␣of␣%8lu",(unsigned long)
            sample_index,(unsigned long) p_wavein->numSamples);
84              fflush(stdout);
85    }
86  /*continue ...*/
```

Apêndice D

Scripts Auxiliares

Inicialização do Dicionário

```
close all;
clear;
clc;

tstart = tic;

i_min = -2^14 + 1;
i_max = 2^14;
codebook_size = 256;
delta = (i_max - i_min + 1)/codebook_size
N = 1000000;

t = i_min:i_max;

alpha = 0.43;
beta = 1.1031*(10^3);
ni = (1/beta)*sqrt(gamma(3/alpha)/gamma(1/alpha));
cte = ((alpha*ni)/(2*gamma(1/alpha)));
pdf = cte*exp(-(ni*abs(t)).^alpha);

cdf = zeros(length(pdf),1);
cdf(1) = pdf(1);
for n=2:length(pdf)
        cdf(n) = cdf(n-1) + pdf(n);
end
s = [];
for i = 1:length(pdf)
        p_Xi = pdf(i);
        qtd_samples = floor(N*p_Xi);
        if(qtd_samples>0)
                s = cat(2,s,t(i)*ones(1,qtd_samples));
        end
end

disp('EOF loop');
disp('length(s):');disp(length(s));
figure; plot(s);
figure; hist(s,256);
figure; stem(t,pdf);
figure; plot(t,cdf);
```

```
41
42 init_codebook = (-(codebook_size/2)+1):codebook_size/2;
43
44 [partition,codebook, distor] = lloyds(s,init_codebook);
45
46 codebook = double(int16(codebook));
47 stem(codebook,ones(length(codebook)));
48 figure; hist(codebook,32);
49 disp('EOF');
50 telapsed = toc(tstart);
51 disp(telapsed);
```

Cálculo do Pós-Filtro

```
1 close all;
2 clear;
3 clc;
4
5 h = firrcos(10,200,0.5,8000,'rolloff');
6 h_norm = h./sum(h);
7
8 freqz(h_norm,1,256,8000);grid;
```