COPPE
Instituto Alberto Luiz Coimbra de
Pós-Graduação e Pesquisa de Engenharia
UFRJ

CODIFICAÇÃO DE VOZ USANDO RECORRÊNCIA DE PADRÕES MULTIESCALAS

Frederico da Silva Pinagé

Tese de Doutorado 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 Doutor em Engenharia Elétrica.

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Rio de Janeiro
Setembro de 2011

CODIFICAÇÃO DE VOZ USANDO RECORRÊNCIA DE PADRÕES MULTIESCALAS

Frederico da Silva Pinagé

TESE 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 DOUTOR EM CIÊNCIAS EM ENGENHARIA ELÉTRICA.

Examinada por:

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

Prof. Sergio Lima Netto, Ph.D.

Prof. Luiz Wagner Pereira Biscainho, D.Sc.

Prof. Murilo Bresciani de Carvalho, D.Sc.

Prof. José Antonio Apolinário Junior, D.Sc.

Prof. Miguel Arjona Ramírez, D.Sc.

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2011

Pinagé, Frederico da Silva

Codificação de Voz usando recorrência de padrões multiescalas/Frederico da Silva Pinagé. – Rio de Janeiro: UFRJ/COPPE, 2011.

XV, 96 p.: il.; 29, 7cm.

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Tese (doutorado) – UFRJ/COPPE/Programa de Engenharia Elétrica, 2011.

Referências Bibliográficas: p. 83 – 89.

1. Recorrência de Padrões. 2. Codificação de Voz. 3. Predição. 4. Gaussiana Generalizada. 5. Gaveta. I. Silva, Eduardo Antônio Barros da *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Elétrica. III. Título.

Aos meus pais Frederico e Neide,
com quem tudo começou, e às
minhas irmãs Gizele e Kellen,
pelo constante incentivo à
conclusão deste doutorado.

Agradecimentos

Agradeço especialmente aos meus orientadores Eduardo Antônio Barros da Silva e Sergio Lima Netto, pela paciência e por terem conduzido com equilíbrio necessário este projeto e pelos trabalhos que desenvolvemos em conjunto. Além disso, gostaria de agradecer pelo apoio de ambos nos momentos de problemas de saúde durante meu período de cirurgia de urgência quando estive no Rio de Janeiro. A vocês o meu muito obrigado.

Em seguida gostaria de agradecer ao Prof. Murilo B. de Carvalho, que me ajudou quanto ao código do MMP, quando dúvidas apareceram em momentos de desenvolvimento do programa. As suas contribuições foram fundamentais para esta tese, principalmente, a sugestão dada por ele no exame de qualificação, que trouxe melhorias ao trabalho.

Agradeço também a Fundação Centro de Análise, Pesquisa e Inovação Tecnológica (FUCAPI), pelo apoio incondicional por permitir a realização deste trabalho, e em especial a Niomar Pimenta e Antonio Luiz Maués, que me apoioram e incentivaram no doutorado. Sou grato também aos colegas de trabalho e aos alunos da FUCAPI que também são meu grande incentivo na vida acadêmica, e à legião de amigos e colegas.

Agradeço às pessoas que muito me ajudaram a escrever este trabalho, por suas reminiscências, paciência e orientação, sou grato ao Hélcio Maia Junior, Eddie Lima Filho, Waldir Sabino e Eduardo Lima. Ao Thiago Prego que me encaminhou arquivos necessários para o desenvolvimento deste trabalho e aos amigos Nuno e Danillo pela paciência de explicar trechos de suas teses que me foram de suma importância. À Lara Feio que me ajudou a escrever o código do algoritmo usado neste trabalho. Obrigado também aos colegas do LPS, que sempre me ajudaram quando pelas dicas no LATEX, no código do programa do algoritmo, e nos momentos de pôr o programa em processamento no cluster do laboratório.

Há muitos outros a quem agradecer. Todas as pessoas, embora não mencionadas aqui, me ajudaram de uma forma ou de outra, com apoio em distintos momentos e por suas presenças em minha vida durante este meu caminho penoso, mas muito gratificante. A todos, os meu sinceros agradecimentos.

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

CODIFICAÇÃO DE VOZ USANDO RECORRÊNCIA DE PADRÕES MULTIESCALAS

Frederico da Silva Pinagé

Setembro/2011

Orientadores: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Programa: Engenharia Elétrica

Este trabalho de pesquisa investiga o desempenho de um algoritmo de codificação de forma de onda para sinais de voz, usando recorrências de padrões multiescalas. O chamado algoritmo MMP (*Multiscale Multidimensional Parser*) usa um dicionário que é atualizado constantemente com expansões, contrações e outras transformações de segmentos previamente codificados. Isto proporciona uma capacidade de aprendizagem para o algoritmo MMP, particularmente adequada para o codificação de segmentos vozeados da voz. Características adicionais, como a quantização não-uniforme do dicionário MMP inicial e o uso de um dicionário auxiliar contendo amostras recentes, são consideradas, a fim de ajustar o mecanismo de aprendizagem do algoritmo MMP para o problema de codificação de voz. Um esquema baseado em predição linear é investigado, onde o algoritmo MMP opera associado ao erro de predição em vez de ao sinal de voz original. Outras características são consideradas, tais como: a quantização/normalização durante o processo de atualização do dicionário, a partição das escalas do dicionário, bem como alterações no tamanho do bloco de predição e a quantização do sinal de resíduo, e o uso de um filtro de conformação de espectro na saída do algoritmo MMP. É verificado que o algoritmo MMP resultante, combinando todas as características citadas, pode atingir uma pontuação objetiva perceptual comparável ao codificador ITU-T G.729 operando na taxa de 8 kbps.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

SPEECH CODING USING MULTISCALE RECURRENT PATTERNS

Frederico da Silva Pinagé

September/2011

Advisors: Eduardo Antônio Barros da Silva
Sergio Lima Netto

Department: Electrical Engineering

This thesis considers a waveform coding algorithm, based on a multiscale multidimensional parsing (MMP) strategy, for speech signals. The resulting MMP algorithm uses a signal codebook, which is constantly updated with expanded/contracted (and several other transformed) versions of previous signal samples. Such codebook update provides a learning ability to the MMP scheme, particularly suitable for voiced segments of the speech signal. Additional features, such as a nonuniform initial codebook and an additional auxiliary codebook with recent coded samples, are considered in an attempt to improve the MMP learning process. A linear prediction stage is also incorporated to the MMP, which operates on the residue signal as opposed to the original speech signal. Other features also considered include, for instance: quantization/normalization of codebook updating procedure, codebook scale partitioning, changing the prediction block size and model order, residue quantization, and an output passband filter. It is verified that the resulting MMP algorithm incorporating all these features can achieve a coding quality, as assessed by the PESQ algorithm, when operating at the 8 kbps rate, comparable to the ITU-T G.729 codec.

Sumário

Lista de Figuras **xi**

Lista de Tabelas **xv**

1 Introdução **1**

1.1 Objetivo 1

1.2 Organização do trabalho 2

2 Compressão de dados **3**

2.1 Técnicas de compressão de sinais 3

2.1.1 Compressão sem perdas 4

2.1.2 Compressão com perdas 5

2.2 Teoria da informação para compressão sem perdas 6

2.3 Teoria da informação para compressão com perdas 7

3 Codificação de voz **8**

3.1 Sinal de voz 8

3.2 Características dos codificadores de voz 9

3.2.1 Taxa de codificação 9

3.2.2 Qualidade do sinal reconstruído 10

3.2.3 Atraso de codificação 11

3.2.4 Complexidade computacional e memória necessária 12

3.2.5 Sensibilidade a erros de canal 12

3.3 Algoritmos padronizados de codificação de voz 12

4 MMP - Multidimensional Multiscale Parser **16**

4.1 Descrição do MMP 16

4.2 Evoluções do MMP 21

4.2.1 MMP com Codificador Aritmético 21

4.2.2 MMP com Otimização Taxa-distorção (MMPRD) 22

5 Adaptação do MMP à codificação de voz **25**
5.1 Leitura do sinal de entrada 26
5.2 Inicialização do dicionário 27
5.3 Casamento dos blocos 28
5.4 Atualização do dicionário 28
5.5 Codificação por entropia 29
5.6 Filtro para redução do efeito de blocagem 30
5.7 MMP com blocos de deslocamento fracionado 33

6 Codificação de forma de onda de voz usando MMP **35**
6.1 MMP com dicionário uniforme - MMP-UNI 35
6.2 MMP com dicionários de deslocamento - MMP-UD 39
6.3 MMP com dicionário não-uniforme - MMP-MU 42
6.4 MMP-MU com dicionários de deslocamento - MMP-MUD 45

7 Codificação de sinais de voz usando MMP baseado em Predição - MMP-P **50**
7.1 Predição Linear 51
7.2 MMP com predição linear - MMP-P 52
7.2.1 Inicialização do dicionário 53
7.2.2 Predição linear 54
7.2.3 Casamento de padrões 54
7.2.4 Atualização do dicionário 54
7.3 Ordem do modelo de predição linear 54
7.4 MMP-P com dicionário de deslocamento MMP-PD 55
7.5 MMP-P com dicionário inicial usando distribuição gaussiana generalizada MMP-GG 57
7.6 Equalização de norma - MMP-EN 61
7.7 Partição das escalas do dicionário MMP-GAV 66
7.8 Validação dos resultados 68
7.9 Alterando o tamanho do bloco de predição 69
7.10 Frases misturadas 72
7.11 Frases concatenadas 73
7.12 Quantizando o bloco de resíduos, saída do preditor LPC e o sinal de entrada 74
7.13 Pós-processamento com filtro passa-baixas 75
7.14 Razão Sinal-ruído (SNR) para o algoritmo MMP 77

8 Conclusão **79**
8.1 Propostas de continuação 81

Referências Bibliográficas **83**

A Pseudo-Código **90**

A.1 Inicialização do Dicionário 91
A.2 Predição 91
A.3 Procedimento de Otimização 91
A.4 Procedimento de Codificação 93
A.5 Procedimento de Atualização do dicionário 94
A.6 Procedimento de Decodificação 95

Lista de Figuras

2.1 Esquema do processo de compressão de dados. 4
2.2 Esquema de Técnica de Compressão sem perdas ($Y = X$). 4
2.3 Esquema de Técnica de Compressão com perdas. 5

3.1 Diagrama em blocos representando a produção da voz. 8
3.2 Diagrama em blocos do algoritmo PESQ (ITU-P.862). 11

4.1 Esquema do dicionário do MMP. 17
4.2 Exemplo de uma parte do sinal de voz para codificação. 18
4.3 Projeção do bloco de entrada X^0 (parte do sinal de voz) no dicionário. 19
4.4 Exemplo do Bloco X^0 segmentado. 19
4.5 Atualização do dicionário. 19
4.6 Árvore de segmentação do Bloco de entrada X^0. 20
4.7 Árvore de segmentação aproximada do Bloco de Entrada X^0. 20
4.8 Estrutura do dicionário do MMP para sinal de voz. 22
4.9 Árvore de segmentação representada com nós folhas. 23

5.1 Digrama de blocos do Codificador MMP para sinais de voz. 25
5.2 Digrama de blocos do Decodificador MMP para sinais de voz com o filtro de redução do efeito blocagem. 26
5.3 Exemplo de um bloco retirado do sinal de voz para processamento ($f_s = 8$ kHz). 27
5.4 Variação do filtro adaptativo usado no efeito blocagem. 31
5.5 Resposta ao impulso dos filtros adaptativos usados para eliminar o efeito blocagem. 32
5.6 Exemplo da redução do efeito de blocagem. 32
5.7 Exemplo de redução do efeito de blocagem. 33
5.8 Exemplo do deslocamento de janelas com comprimentos determinados para atualização do dicionário MMP. 34

6.1 Quantização do dicionário do MMP: dicionário uniforme. 36
6.2 Quantização do dicionário do MMP: dicionário não-uniforme. 36

6.3 Resultados PESQ-MOS para o algoritmo MMP-UNI, com taxa de codificação em torno de 8 kbps, com dicionário uniforme com diferentes tamanhos T em comparação ao resultado G.729. 37
6.4 Resultados PESQ-MOS para o algoritmo MMP-UNI ampliado na região em torno de 8 kbps de taxa de codificação. 38
6.5 PESQ-MOS × taxa de codificação para o algoritmo MMP-UNI. . . . 38
6.6 Gráfico da escala 1×2 do dicionário original do MMP-UD para taxa de 8 kbps. Cada ponto representa um elemento bi-dimensional do dicionário 39
6.7 Exemplos de trecho de voz: (a) sonoro; (b) surdo. 40
6.8 Procedimento do dicionário de deslocamento. 42
6.9 Resultados PESQ-MOS para o algoritmo MMP-UD com dicionário de deslocamento com taxa de codificação em torno de 8 kbps. 43
6.10 Número final de elementos do dicionário do algoritmo MMP-UD para diversas taxa de codificação. 43
6.11 Uso do dicionário de deslocamento para o algoritmo MMP-UD. 44
6.12 Resultados PESQ-MOS para o algoritmo MMP-UD em comparação com os demais codificadores de voz. 44
6.13 Resultados PESQ-MOS para o algoritmo MMP-MU, com taxa de codificação em torno de 8 kbps, com dicionários uniforme (linha tracejada) e não-uniforme (linha sólida) em comparação ao resultado G.729. 45
6.14 Número final de elementos do dicionário do algoritmo MMP-MU para taxa de codificação de 8 kbps. 46
6.15 Resultados PESQ-MOS para o algoritmo MMP-MU em comparação aos resultados G.729, G.726 e G.711. 46
6.16 Resultados PESQ-MOS para o algoritmo MMP-MUD, com taxa de codificação em torno de 8 kbps, para diferentes comprimentos do dicionário de deslocamento, em comparação aos resultados para o G.729. 47
6.17 Uso do dicionário de deslocamento ao longo do tempo no codificador MMP-MUD para um dado sinal de voz. 47
6.18 Estatística do tamanho dos segmentos no codificador MMP-MUD. . . 48
6.19 PESQ-MOS × taxa de codificação para o algoritmo MMP-MUD (linha sólida) comparando com o MMP-UD (linha tracejada). 49
6.20 Gráfico da escala 1×2 do dicionário original do MMP-MUD para taxa de 8 kbps. Cada ponto corresponde às coordenadas de um elemento do dicionário. 49

7.1 Esquema do processo de predição linear no algoritmo MMP-P. 53
7.2 Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-P com diferentes ordens do modelo LP. 55
7.3 Resultados PESQ-MOS para o algoritmo MMP-P ampliado na região em torno de 8 kbps para diferentes ordens N do modelo LP. 56
7.4 PESQ-MOS × taxa de codificação para o algoritmo MMP-P. 56
7.5 Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-PD com diferentes comprimentos L do dicionário de deslocamento. . . 57
7.6 PESQ-MOS × taxa de codificação para o algoritmo MMP-PD. 58
7.7 Histograma do erro de predição para o banco de frases DB2. 59
7.8 Modelando a envoltória do histograma do erro de predição (linha tracejada) usando uma DGG (linha sólida). 59
7.9 DGG para valores do parâmetro $\alpha = 0.5, 1, 2$ e 5. A distribuição é normalizada para variância unitária. 60
7.10 Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-GG usando diferentes estratégias para o dicionário inicial. "qtz" indica a quantização durante a atualização do dicionário, e "nqtz" indica que não há quantização durante a atualização do dicionário. . . 61
7.11 Resultado PESQ-MOS para diferentes taxas do algoritmo MMP-GG comparando com o algoritmo MMP-PD. 62
7.12 PESQ-MOS × taxa de codificação para o algoritmo MMP-GG. 62
7.13 Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-GL para diferentes ordens do modelo LP. 63
7.14 Resultado PESQ-MOS para diferentes taxas do algoritmo MMP-GD com diferentes comprimentos L do dicionário de deslocamento. 63
7.15 PESQ-MOS × taxa de codificação para o algoritmo MMP-GD. 64
7.16 Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-EN usando diferentes normas para o estágio de atualização do dicionário. 65
7.17 Resultado PESQ-MOS para diferentes taxas de codificação para versões MMP-PD (linha tracejada) e MMP-EN (linha sólida). Para as taxas em torno de 32 e 64 kbps, o MMP-PD usa o dicionário inicial com 1024 e 4096 elementos para cada escala do dicionário, respectivamente. 66
7.18 Processo de partição do dicionário, de acordo com a escala de origem. A partição i de uma escala k corresponde aos elementos da escala k originados da escala i, através de uma transformação de escala T_k^i. . . 67
7.19 Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-GAV. 68
7.20 Resultado PESQ-MOS para diferentes taxas do algoritmo MMP-GAV. 68

7.21 Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-GAV para a base de dados DB2. 69
7.22 Resultado PESQ-MOS para diferentes taxas do algoritmo MMP-GAV para a base de dados DB2, comparando com os outros codificadores de voz. 70
7.23 Trecho de um sinal de voz sendo comparado com o bloco de predição. 71
7.24 Trecho de um sinal de voz sendo comparado com o bloco predito, para uma tamanho de bloco de predição igual a 16 amostras. 71
7.25 Resultado PESQ-MOS para frases misturadas (contendo trechos de silêncio) em torno de 8 kbps usando o algoritmo MMP-GAV, baseado em partição das escalas do dicionário. 72
7.26 Resultado PESQ-MOS para frases misturadas (sem trechos de silêncio) em torno de 8 kbps usando o algoritmo MMP-GAV, baseado em partição das escalas do dicionário. 73
7.27 Resultado PESQ-MOS para frases concatenadas (sem trechos de silêncio) em torno de 8 kbps usando o algoritmo MMP-GAV baseado em partição das escalas do dicionário. 74
7.28 Resultado PESQ-MOS em torno de 8 kbps usando o algoritmo MMP-GAV baseado em partição das escalas do dicionário quantizando o bloco de resíduos, a saída do LPC e o sinal de entrada. 75
7.29 Resultado PESQ-MOS em torno de 8 kbps para o MMP com diferentes tipos de filtros passa-baixas. 76
7.30 Resposta em magnitude do filtro passa-baixas do tipo *raised cosine* usado no estágio de pós-filtragem. 76
7.31 Resultado do banco de dados DB2 utilizando pós-processamento com filtro passa-baixas, comparando com os outros codificadores de voz. . 77
7.32 Razão Sinal-ruído em torno de 8 kbps para o algoritmo MMP. 78

Lista de Tabelas

3.1 Tabela MOS de qualidade de voz. 10

5.1 Parâmetro de entrada do codificador 26

7.1 Experimentos alterando o tamanho do dicionário inicial e o tamanho do bloco de predição. Os valores dados são do PESQ-MOS obtido apenas com a predição. 70

7.2 Coeficientes do filtro *raised cosine*. 77

A.1 Parâmetros do algoritmo MMP. 90

A.2 Contadores para totalizar a frequência de utilização dos índices. . . . 90

A.3 Contadores para totalizar a frequência dos *flags* de segmentação. . . . 90

Capítulo 1

Introdução

A voz é o meio de comunicação mais utilizado pelo homem e é uma das características que difere o ser humano dos outros seres vivos, permitindo-lhe expressar suas ideias, opiniões e pensamentos. Com a voz o homem consegue comunicar-se mais facilmente e até mesmo remotamente através da comunicação telefônica para transmissão de uma informação à distância. A importância da comunicação telefônica tem crescido muito na sociedade, tornando o mundo menor e aproximando mais as pessoas. O crescimento da necessidade do ser humano de se comunicar à distância levou ao surgimento da comunicação digital, procurando representar o sinal de forma mais eficiente através dos codificadores digitais de voz.

1.1 Objetivo

Esta tese é sobre codificadores, cujo objetivo é representar a informação digital de voz na forma mais compacta possível. Os codificadores de voz podem ser divididos em dois grandes grupos: os codificadores de forma de onda e os baseados em modelos de produção de voz humana. Apesar de os codificadores de forma de onda terem sido muito usados no passado, os codificadores de voz de maior sucesso hoje são os que usam explicitamente modelos de produção da voz humana. Nesta tese, estudaremos codificadores de forma de onda baseados em um novo paradigma, a recorrência de padrões multiescalas. Esta classe de algoritmos é denominada genericamente por MMP (*multidimensional multiscale parser*).

De fato, após os ótimos resultados do algoritmo MMP em codificação de imagens e outros sinais [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14], surgiu a ideia de implementá-lo também no contexto de codificação de voz. Considerando-se a capacidade de adaptação deste algoritmo, e a periodicidade e regularidade existentes num sinal de voz, pode-se esperar um bom desempenho do MMP neste problema. Em termos gerais, o MMP se adapta às características do sinal de entrada à medida que este está sendo codificado, usando o conhecimento que vai sendo adquirido ao longo do

processo para codificar amostras futuras. O algoritmo MMP é bastante flexível, permitindo incluir no processo de atualização do dicionário características perceptuais do sinal de voz. Uma outra característica importante do algoritmo MMP é a de que sua relação taxa-qualidade é facilmente ajustável, o que o torna útil em diferentes aplicações com necessidades de diferentes níveis de qualidade ou de taxa de codificação.

Neste contexto, esta tese pretende investigar a viabilidade de se obterem, usando codificadores de forma de onda baseados na recorrência de padrões multiescalas, desempenhos compatíveis com os dos codificadores que usam explicitamente modelos de produção de voz, que representam o estado da arte atual.

1.2 Organização do trabalho

Este trabalho encontra-se dividido em 9 capítulos, cada um contendo informações importantes para seu desenvolvimento.

O Capítulo 2 apresenta uma pequena introdução à compressão de dados e introduz conceitos de teoria da informação.

No Capítulo 3 é feita uma introdução dos sinais de voz e apresentam-se ainda alguns tipos de codificadores de voz e suas principais características.

No Capítulo 4 é feita uma apresentação genérica do algoritmo MMP.

No Capítulo 5, é descrito o algoritmo MMP aplicado à codificação de um sinal de voz, bem como são vistas algumas modificações em sua implementação que possam redundar em melhorias no seu desempenho, obtendo maior qualidade perceptiva para uma dada taxa de compressão.

No Capítulo 6, é apresentado o algoritmo MMP como um codificador de forma de onda. Procedimentos de quantização uniforme e não-uniforme do dicionário inicial são incorporados ao algoritmo, bem como o uso de um dicionário auxiliar contendo amostras previamente codificadas, chamado de dicionário de deslocamento.

Um modelo de predição linear incorporado ao algoritmo MMP é apresentado no Capítulo 7 para avaliar seu desempenho quando operando sobre o sinal de resíduo. Recursos adicionais usando uma atualização do dicionário com segmentos quantizados e/ou normalizados e a partição das escalas do dicionário principal são também apresentados.

O Capítulo 8 investiga o desempenho do algoritmo MMP com alterações em sua estrutura, procurando melhorar seu desempenho (em termos da relação qualidade versus taxa) e comparando com o resultado do codec G.729.

Por fim, no Capítulo 9, tiramos conclusões quanto aos resultados obtidos e propomos caminhos para seguir no desenvolvimento deste trabalho.

Capítulo 2

Compressão de dados

A compressão de dados consiste em reduzir o número de bits necessários para representar uma informação (por exemplo, uma imagem, uma sequência de vídeo ou um sinal de voz). Podemos representar a informação de forma compacta identificando e usando estruturas que existem no sinal. A informação pode estar representada na forma de caracteres num arquivo texto, no número de amostras de um sinal de voz, nas formas de onda de uma imagem, entre outros. E pelo fato de sempre estarmos gerando e usando informações na forma digital, torna-se cada vez mais necessário o uso da compressão de dados para podermos representar a informação com um número reduzido de bits. Existem vários tipos de estruturas que são usados na compressão de dados. Por exemplo, num arquivo de texto, caracteres que ocorrem com mais frequência podem ser representados com um menor número de bits. Soluções similares ou equivalentes podem ser aplicadas a outros tipos de sinais, como visto na próxima seção.

2.1 Técnicas de compressão de sinais

Quando falamos em técnicas de compressão (ou algoritmos de compressão), estamos nos referindo a um algoritmo de codificação (codificador) que processará uma entrada X e produzirá uma saída X_c, e a um algoritmo de decodificação (decodificador) que reconstruirá a informação Y a partir da saída X_c do codificador. Este processo é representado na Figura 2.1.

Com base no esquema da Figura 2.1, podemos obter dois tipos de compressão de sinais: sem perdas (Figura 2.2), onde a saída Y do decodificador é idêntica à entrada X do codificador, e a com perdas (Figura 2.3), na qual a saída Y do decodificador é diferente da entrada X do codificador. Nas subseções seguintes iremos comentar sobre estes dois tipos de compressão de dados.

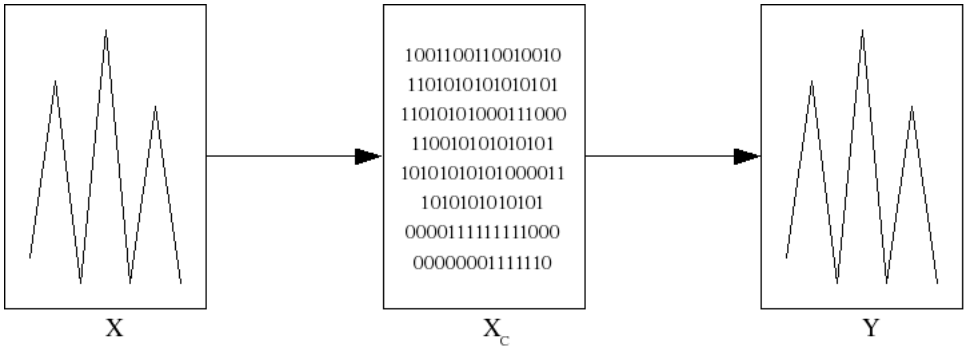


Figura 2.1: Esquema do processo de compressão de dados.

2.1.1 Compressão sem perdas

Como mencionamos anteriormente, nas técnicas de compressão sem perdas podemos recuperar completamente a informação da entrada quando efetuamos a decodificação. Uma característica destas técnicas (que pode ser encarada como uma limitação) é o fato de que em geral não se consegue uma alta taxa de compressão, quando comparada àquela que pode ser obtida por meio de técnicas de compressão com perdas.

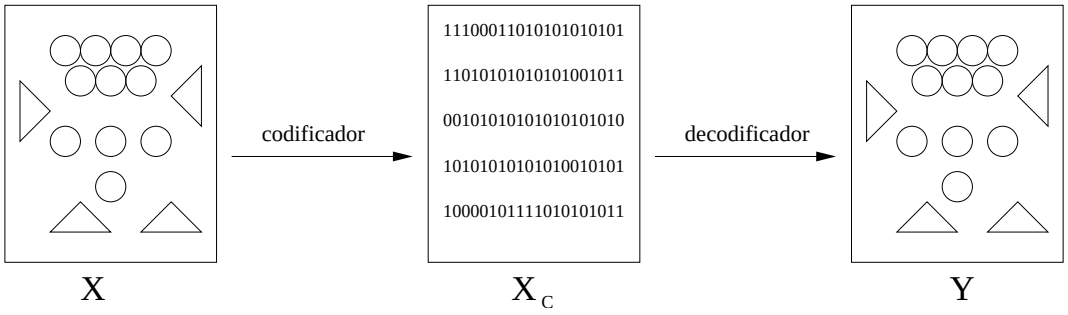


Figura 2.2: Esquema de Técnica de Compressão sem perdas ($Y = X$).

Usamos compressão sem perdas em aplicações que exigem que não haja diferença entre a informação reconstruída e a original. Por exemplo, em imagens de radiografia, uma pequena diferença entre a imagem reconstruída e a original pode levar a um diagnóstico errado. As técnicas de compressão sem perdas também têm aplicação em compressão de textos, de imagens de satélite e de sinais biomédicos.

Existem várias técnicas que utilizam este tipo de compressão sem perdas, dentre elas podemos citar a técnica desenvolvida por David Huffman, chamada código de Huffman [15, 16], a codificação aritmética de Peter Elias [17] e as técnicas baseadas em dicionários, que foram desenvolvidas por Abraham Lempel e Jacob Ziv [18, 19].

2.1.2 Compressão com perdas

Nas técnicas de compressão com perdas são admitidas diferenças entre os dados recuperados e os presentes à entrada. A vantagem delas é que, em geral, podemos obter uma maior taxa de compressão em relação às técnicas de compressão sem perdas.

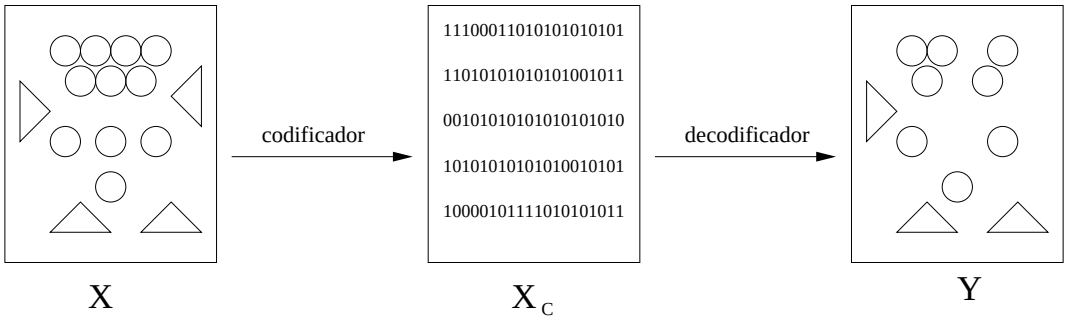


Figura 2.3: Esquema de Técnica de Compressão com perdas.

No processo de compressão com perdas (representada na Figura 2.3), definimos a distorção D como sendo alguma medida da diferença entre o sinal original X e o sinal recuperado Y. O estudo da relação entre a taxa R de compressão e a distorção D é chamado de teoria da taxa-distorção.

Entre as técnicas de compressão com perdas, podemos citar a *codificação por transformadas*. Estes codificadores efetuam uma transformação da fonte de entrada em uma outra sequência, onde a maior parte da informação está concentrada em poucos elementos, ou seja, o resultado da transformação resulta numa compactação de energia. Estes tipos de codificadores foram inicialmente utilizados por Huang e Schultheiss [20]. Este método consiste em três etapas: *transformação*, na qual se transforma uma sequência de entrada $\{x_n\}$ em outra sequência $\{\theta_n\}$; *quantização*, que consiste na quantização da sequência transformada; e por último a *codificação*.

A transformação, como mencionado anteriormente, consiste em mapear uma sequência de entrada $\{x_n\}$ em outra sequência $\{\theta_m\}$. Essa transformação pode ser linear de acordo com a equação abaixo:

$$\theta_n = \sum_{i=0}^{N-1} x_i a_{n,i}. \tag{2.1}$$

A sequência de entrada $\{x_n\}$ pode ser recuperada da sequência transformada pela transformada inversa:

$$x_n = \sum_{i=0}^{N-1} \theta_i b_{n,i}. \tag{2.2}$$

Todo este processo de transformação, direta e inversa, pode ser representado na

forma matricial por

$$\theta = \mathbf{A}\mathbf{x}, \mathbf{x} = \mathbf{B}\theta. \tag{2.3}$$

Exemplos de transformadas comumente usadas em compressão são a transformada discreta de Walsh-Hadamard (DWHT) [21], a transformada do cosseno [22, 23, 24] e a transformada wavelet [25, 26, 27].

A quantização consiste em representar a grande quantidade de elementos gerados pela transformação por uma quantidade (significativamente) menor de símbolos. Os quantizadores são de dois tipos: *quantizador escalar* e *quantizador vetorial*. Chamamos a quantização de escalar quando o conjunto de entrada tem valores escalares, e de vetorial quando o conjunto de entrada são vetores.

Na etapa de codificação os símbolos gerados na etapa de quantização são representados por símbolos de um alfabeto pré-definido, usualmente binário.

2.2 Teoria da informação para compressão sem perdas

Shannon [28] desenvolveu um dos primeiros estudos para verificar de forma quantitativa a informação, definindo-a da seguinte maneira: Tendo-se um evento aleatório A e sua probabilidade associada $P(A)$, então podemos definir a auto-informação associada a A, por:

$$i(A) = \log_b \frac{1}{P(A)} \tag{2.4}$$

onde a base b do logaritmo define a unidade de informação; especificamente, para $b = 2$ tem-se a unidade bits/símbolo. Pode-se concluir que se a probabilidade do evento é baixa, a quantidade de auto-informação é alta, e se a probabilidade de um evento é alta, então a auto-informação é baixa.

Shannon também contribuiu neste estudo desenvolvendo a chamada auto-informação média, ou entropia [16, 29, 30, 31], caracterizada por: Seja o alfabeto $A = \{a_1, a_2, \ldots, a_n\}$ da fonte S, com probabilidades associadas $P(a_j)$. A entropia da fonte S é definida por

$$H(S) = \sum_{j=1}^{n} P(a_j) i(a_j) = -\sum_{j=1}^{n} P(a_j) \log P(a_j). \tag{2.5}$$

Pode-se demonstrar que nenhum código de compressão sem perdas pode ter taxa inferior à entropia, que é por isso o limite de desempenho de tais códigos.

2.3 Teoria da informação para compressão com perdas

Na compressão com perdas, o sinal X, depois de comprimido, é representado por Y, que é diferente de X. Neste caso, definimos a distorção de um símbolo x_i que é recuperado como y_j por $d(x_i, y_j)$. Uma medida de distorção bastante usada é o erro quadrático, onde $d(x_i, y_j) = (x_i - y_j)^2$. Entretanto, em casos práticos, outras medidas de distorção podem ser usadas. Por exemplo, em codificação de voz, o objeto desta tese, a medida $d(x_i, y_j)$ pode incorporar aspectos da percepção auditiva humana.

Dada uma taxa média usada para representar Y atingindo uma distorção média D, a menor taxa de codificação R que pode gerar uma representação com distorção D é dada pela função taxa-distorção $R(D)$. Para melhor entendermos esta função precisamos de dois conceitos: *entropia condicional* e *informação mútua média*.

Dadas duas varáveis aleatórias [32] X e Y, dois alfabetos $X' = \{x_1, x_2, \ldots, x_n\}$ e $Y' = \{y_1, y_2, \ldots, y_m\}$ e distribuição conjunta $P(x, y)$, a entropia condicional $H(X|Y)$ e a informação mútua $I(X;Y)$ são definidas por [30]

$$H(X|Y) = -\sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log P(x_i|y_j), \tag{2.6}$$

$$I(X;Y) = \sum_{i=1}^{n}\sum_{j=1}^{m} P(x_i, y_j) \log \left[\frac{P(x_i|y_j)}{P(x_i)}\right], \tag{2.7}$$

de modo que

$$I(X;Y) = H(X) - H(X|Y). \tag{2.8}$$

A função taxa-distorção especifica a menor taxa R na qual uma fonte X pode ser codificada pela variável aleatória Y com uma distorção menor ou igual a D. Esta função é dada por [30]

$$R(D) = \min_{P(y_j|x_i)|\sum_i\sum_j P(x_i)P(y_j|x_i)d(x_i,y_j)\leq D^*} I(X;Y). \tag{2.9}$$

Esta tese trata justamente de métodos capazes de representar um sinal de voz previamente digitalizado com desempenho o mais próximo possível da função $R(D)$. Para tal usaremos a codificação de forma de onda usando a recorrência de padrões multiescalas. No capítulo seguinte será dada uma visão geral das técnicas mais usadas em codificação de sinais de voz hoje em dia.

Capítulo 3

Codificação de voz

Existe uma série de características que têm que ser levadas em consideração quando escolhemos um codificador de voz para uma determinada aplicação. Podemos citar algumas como: taxa de compressão, qualidade do sinal reconstruído, atraso da saída para a entrada, entre outras (que serão comentadas na Seção 3.2). Geralmente, tem que haver um compromisso entre estas características para que o codificador seja adequado para uma aplicação.

Este capítulo fala sobre a codificação de voz, abordando inicialmente os conceitos de como a voz humana é produzida, chegando nas características dos codificadores de voz e sistemas usados nos padrões de codificação existentes.

3.1 Sinal de voz

O diagrama em bloco da Figura 3.1 representa um modelo simplificado do processo de geração do sinal de voz.

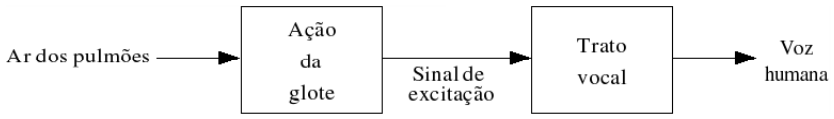


Figura 3.1: Diagrama em blocos representando a produção da voz.

Neste processo, o fluxo de ar gerado dos pulmões passa pela glote formando o sinal de excitação. Este sinal, ao passar pelo trato vocal, constituído pela faringe, cavidades bucal e nasal e os elementos articuladores (língua, lábios, dentes etc.), gera o sinal de voz (Figura 3.1). Podemos classificar um trecho de voz em diferentes tipos, de acordo com as características gerais do sinal de excitação [33]. Na prática, porém, a caracterização mais importante trata da classificação em sons sonoros e surdos. Os sons sonoros são aqueles produzidos pela vibração das cordas vocais, fazendo variar a frequência de abertura/fechamento e o volume de ar proveniente dos pulmões

através da glote. É esta variação quase periódica e impulsiva no volume de ar que excita o trato vocal, produzindo sons com harmônicos da frequência de vibração das cordas vocais, comumente designada por frequência de *pitch*. No diagrama da Figura 3.1, o trato vocal atua como um filtro linear, amplificando algumas zonas do espectro e atenuando outras. As zonas amplificadas correspondem às zonas de ressonância, definidas por uma frequência central, por uma largura de banda e por uma energia. A frequência central da ressonância é denominada de frequência do formante ou simplesmente formante. A configuração do trato vocal determina os formantes e as características de timbre da voz.

3.2 Características dos codificadores de voz

Quando se pretende projetar ou utilizar um codificador de voz deve-se levar em consideração uma série de características na escolha de um codificador para uma aplicação específica: a taxa de codificação do sinal original, neste caso temos que ter uma transmissão e armazenamento da informação mais eficientes; a qualidade do sinal reconstruído; o atraso correspondente ao processo de codificação; a complexidade computacional e memória necessárias para efetuar o processamento, etc. Em geral, um codificador escolhido para uma dada aplicação representa um compromisso entre todas essas características. A seguir fazemos uma breve descrição de cada uma das características.

3.2.1 Taxa de codificação

A codificação é uma forma de representar, através de bits, um sinal digital. Para cada tipo de codificação existe uma quantidade de bits necessárias para representar um sinal em um determinado intervalo de tempo; esta quantidade de bits é chamada de taxa de codificação. Assim, pode-se perceber que há tipos de codificadores mais eficazes do que outros. O codificador que tiver a menor taxa de codificação para uma mesma qualidade de codificação será mais eficiente na relação taxa-distorção. Em geral, a redução da quantidade de bits vem acompanhada da redução da qualidade do sinal codificado/decodificado ou reconstruído. Porém, dentre os codificadores de sinal de voz há um codificador que possui baixa taxa de codificação com uma qualidade boa para o sinal reconstruído, e por este motivo é largamente utilizado em vários padrões de telefonia digital. Este codificador é chamado CELP (*Code Excited Linear Prediction*) e será apresentado na Seção 3.3.

3.2.2 Qualidade do sinal reconstruído

Realizando-se o processo de codificação com perdas, o sinal de voz reconstruído não é igual ao sinal original. Nos codificadores em que uma redução do ruído de quantização leva a uma aproximação entre os sinais de entrada e de saída é comum encontrar como medida de qualidade a razão sinal-ruído (*Signal-to-Noise Ratio* - SNR), sendo normalmente expressa em decibéis (dB). Esta medida, porém, tem o inconveniente de ser dominada pelas zonas de maior energia. Uma vez que as zonas de baixa energia são também perceptualmente relevantes, é conveniente que se utilize uma forma alternativa de avaliar o desempenho de codificadores de voz. Uma possibilidade é calcular a SNR em blocos de dimensão de 10 a 20ms e, no final, determinar a média ao longo de todo o sinal de voz, ponderada pelo espectro de trato vocal de cada bloco. Esta medida é referida como SNR perceptual segmental [33]. Uma outra possibilidade é o uso de avaliações subjetivas padronizadas baseadas diretamente na avaliação humana. Em particular, o MOS (*Mean Opinion Score*) é uma das metodologias mais importantes e mais utilizadas para aferição da qualidade perceptual de codificadores de voz. Descrito na recomendação da P.800 do ITU-T [34], o MOS é calculado reunindo-se um grupo de pessoas (ouvintes) que são confrontados com frases originais e codificadas, sendo-lhes pedido que lhe atribuam notas uma escala de 1 a 5 como indicado na Tabela 3.1. O resultado final é obtido pela média dos valores das respostas.

Tabela 3.1: Tabela MOS de qualidade de voz.

Nota	**Qualidade da Voz**
5	Excelente
4	Bom
3	Razoável
2	Ruim
1	Péssimo

Por serem custosas operacionalmente, as avaliações objetivas são comumente complementadas, ou mesmo substituídas, pelos chamados avaliadores objetivos. O algoritmo PESQ - *Perceptual Evaluation of Speech Quality*, descrito na norma ITU-T P.862 [35], é um método objetivo de avaliação de qualidade com alta correlação com os tradicionais testes MOS, é próprio para sinais de voz telefônicos com taxa de amostragem de 8000 amostras/s. A variante W-PESQ (*Wide-Perceptual Evaluation of Speech Quality*), descrito na norma ITU-T P.862.2 [36], é própria para sinais com taxa de amostragem de 16000 amostras/s. O algoritmo PESQ quantifica a diferença entre os sinais original e reconstruído a partir de um modelo para o codificador que inclui módulos de mascaramento perceptual, que procuram estabelecer as distorções percebidas pelo nosso aparelho auditivo. O PESQ é composto de diversos blocos,

ilustrados na Figura 3.2 e descritos a seguir:

- Normalização (ajuste de nível): é responsável por normalizar a potência do sinal de entrada para um dado valor de referência, geralmente o mesmo valor dos testes subjetivos.

• o sinal de entrada é filtrado de forma a compensar distorções no espectro, que são notoriamente introduzidas pela rede telefônica ou pelos *handsets* dos aparelhos telefônicos.

- Alinhamento temporal: o sistema testado pode incluir um atraso, muitas vezes variável durante o teste. Para compensar este efeito, a amostra é dividida em blocos e o algoritmo tenta identificar similaridades entre eles e compensar atrasos devidos à codificação/decodificação do sinal.

- Mascaramento perceptual: os sinais original e degradado passam por uma transformação para mapear as propriedades da audição humana, e assim os níveis no domínio transformado que estão abaixo dos limiares da audição humana são eliminados, já que não contribuiriam para o resultado final de uma avaliação subjetiva.

- Modelagem cognitiva: estima a distância entre os dois sinais (original e reconstruído) devidamente processados. Esta distância PESQ é então mapeada na escala MOS [34] usando-se a relação

$$\text{PESQ–MOS} = 0{,}999 + \frac{4}{1 + e^{-1{,}4945\text{PESQ}+4{,}6607}}. \tag{3.1}$$

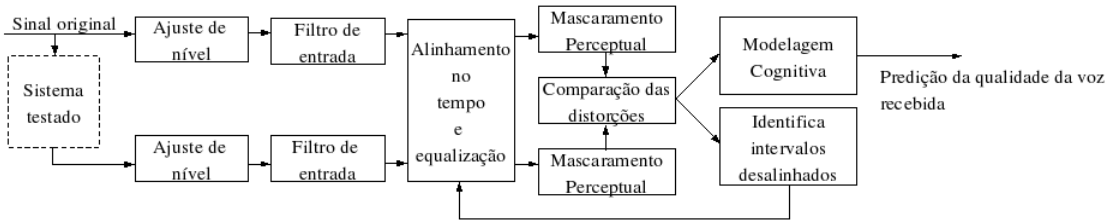


Figura 3.2: Diagrama em blocos do algoritmo PESQ (ITU-P.862).

3.2.3 Atraso de codificação

O atraso em codificação de voz é o tempo máximo entre o instante de uma amostra na entrada do emissor e a amostra correspondente que é gerada no receptor. Atrasos muito longos podem atrapalhar uma conversa (telefônica) em tempo real. O atraso de codificação é medido com o receptor ligado ao emissor, sem levar em consideração

o tempo de propagação do sinal e a contribuição dos equipamentos de emissão e recepção.

3.2.4 Complexidade computacional e memória necessária

Quando o algoritmo de codificação requer altas complexidade e quantidade de memória, os sistemas tendem a ser custosos e ter maior consumo de energia. A complexidade é medida através do número de MIPS (milhões de instruções por segundo), MFLOPS (milhões de operações em ponto flutuante por segundo) ou WMOPS (uma medida similar ao MFLOPS, mas com uma ponderação para cada tipo de operação) necessários para processar os algoritmos de codificação. A memória é medida em número de bytes.

3.2.5 Sensibilidade a erros de canal

O sinal codificado transmitido está sujeito a erros introduzidos pelo canal. Estes erros podem ser de dois tipos: erros aleatórios independentes, causados pelo ruído estacionário, e erros em rajada, causados, por exemplo, por interferências eletromagnéticas no canal. Para evitar que a qualidade do sinal de saída seja afetada por estes erros, os codificadores devem ter metodologias para recuperar o sinal original, na medida do possível.

3.3 Algoritmos padronizados de codificação de voz

As características descritas nas seções anteriores são até um certo ponto contraditórias e na prática procura-se um equilíbrio delas todas. Nesta seção mostramos os algoritmos de codificação de voz que procuram equilibrar estas características.

Existem três grupos principais de codificadores de voz: por forma de onda, paramétricos (geralmente baseados em modelos de produção de voz) ou híbridos [33].

Os codificadores forma de onda utilizam essencialmente as características temporais dos sinais de voz. Como exemplos de algoritmos desta família podemos citar o que está na norma G.711 (PCM - *Pulse Code Modulation*) [37] e o que está na norma G.721 (ADPCM - *Adaptive Differencial Pulse Code Modulation*) [38].

Os codificadores paramétricos conseguem os mais altos graus de compressão com média complexidade. Isto é feito através da extração dos parâmetros do sinal original que modelam o sistema humano de voz utilizando o esquema descrito na Figura 3.1. A grande desvantagem deste grupo de codificadores é a qualidade sintética do sinal

reconstruído [39]. O codificador paramétrico descrito é o LPC10 (*Linear Predictive Coding*).

Os codificadores híbridos realizam a extração de parâmetros do trato vocal tal como os codificadores paramétricos, e ao mesmo tempo utilizam características temporais do sinal de excitação, como os codificadores de forma de onda. Dessa forma consegue-se obter boa qualidade do sinal reconstruído a taxas razoavelmente baixas (entre 2 e 16 kbps), ao custo de um aumento na complexidade do processo de codificação [39]. Algumas das principais técnicas de codificação híbrida fazem parte da família CELP (*Code Excited Linear Prediction*) de codificadores apresentada em [40]. Com algumas melhorias no algoritmo como em [41, 42, 43, 44, 45], tornou-se possível sua aplicação em tempo real [46]. Exemplos de padrões desta família incluem o LD-CELP(*Low Delay CELP*) [47] e CS-ACELP (*Conjugate-Structure Algebraic - CELP*) [48].

A seguir comentamos brevemente acerca de cada um dos tipos de algoritmos de codificação mencionados anteriormente:

1. PCM (*Pulse Code Modulation*) - ITU-T G.711 [37]: A recomendação G.711 é o padrão da telefonia fixa digital usada no Brasil. O G.711 representa uma modulação por pulsos, que consiste em representar digitalmente um sinal de voz a uma taxa de 8000 amostras/s com 8 bits por amostra. Um sinal de voz reconstruído pela codificação PCM G.711 (codificação por forma de onda) possui qualidade quase indistinguível da original. As principais características deste codificador são:

   - Taxa de amostragem de 8 kHz;
   - 64 kbps de taxa de transmissão;
   - Atraso típico de 0,125 ms;
   - Complexidade muito baixa;
   - Excelente qualidade, em torno de 4,3 MOS.

2. ADPCM (*Adaptive Differencial Pulse Code Modulation*) - ITU-T G.726 [38]: Este codificador é uma variante do PCM que procura estimar o valor da amostra atual utilizando um preditor linear adaptativo. A codificação é feita sobre a diferença entre o valor real e o estimado. As principais características deste codificador são:

   - Frequência de amostragem de 8 kHz;
   - Taxas de 16, 24, 32 e 40 kbps;
   - Atraso típico de 0,125 ms;

- Complexidade relativamente baixa;
- Alta qualidade, em torno de 4,1 MOS, para a taxa de 32 kbps.

3. LPC10 (*Linear Predictive Code*) - FS-1015 [49]: Faz uso do modelo simplificado de produção da voz humana, e consegue obter uma voz inteligível a taxas de 1 e 2,4 kbps [50, 51]. Esta algoritmo foi padronizado em 1976 pelo governo dos Estados Unidos para comunicações seguras [39, 50, 51]. Suas principais características são:

   - Frequência de amostragem de 8 kHz;
   - Taxa de 2,4 kbps;
   - Complexidade média;
   - Qualidade baixa, em torno de 2,4 MOS.

4. LD-CELP (*Low - Delay Code Excited Linear Prediction*) - ITU-T G.728 [47, 52]: Este codificador pertence à família CELP e é usado principalmente nas comunicações móveis. Ele surgiu quando foi lançada a proposta de um codificador a uma taxa de 16 kpbs com baixo atraso. Os principais requisitos eram que a qualidade não fosse pior que a do ADPCM a 32 kbps e que o atraso de codificação máximo fosse de 5 ms. As principais características deste codificador são:

   - Frequência de amostragem de 8 kHz;
   - Taxa de 16 kbps;
   - Atraso típico de 0,625 ms;
   - Alta complexidade;
   - Boa qualidade, em torno de 4,1 MOS.

5. CS-ACELP (*Conjugate-Structure Algebraic Code Excited Linear Prediction*) - G.729 ITU-U [48]: Este algoritmo pertence também à família CELP e é o mais usado em aplicações VoIP. Ele oferece grande qualidade e robustez ao preço de uma grande complexidade. As principais características deste codificador são:

   - Frequência de amostragem de 8 kHz;
   - Taxa de 8 kbps;
   - Atraso típico de 15 ms;
   - Alta complexidade;
   - Boa qualidade, em torno de 4,0 MOS.

Este capítulo contribuiu para termos noção das características dos codificadores de voz padrão que serão usados como referência para avaliação do desempenho do algoritmo MMP. Desta forma podemos identificar no algoritmo as suas características que devem ser investigadas e melhoradas.

Nesta tese, para obtermos uma melhor relação qualidade×taxa outros aspectos de desempenho de um codificador de voz (como complexidade computacional e memória, por exemplo) são a princípio desconsiderados.

Capítulo 4

MMP - Multidimensional Multiscale Parser

Este capítulo descreve um algoritmo de compressão com perdas baseado no casamento aproximado de recorrências de padrões multiescalas, que é chamado de MMP (*Multi-dimensional Multiscale Parser*), que foi proposto em [53], originalmente desenvolvido para compressão de imagens e tendo aplicações em uma grande variedade de sinais [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]. O MMP tenta representar o sinal a ser codificado através de aproximações de partes do sinal que foram previamente codificadas, aprendendo, desta forma, sobre o sinal durante o processo de codificação, por isso o termo "recorrência". Já o termo "padrões multiescalas" é devido ao fato de o casamento entre o bloco que está sendo codificado e os elementos (blocos ou palavras) já conhecidos pelo algoritmo poder ser feito com dimensões diferentes (escalas). O MMP contém um dicionário que é construído e atualizado utilizando concatenações de versões contraídas ou dilatadas de blocos de imagem previamente codificados.

4.1 Descrição do MMP

Antes de inicializar a codificação, é necessário definir os parâmetros da estrutura do dicionário (inicializar o dicionário, vide Seção 5.2). O dicionário é composto de várias escalas k. O número de escalas é limitado e definido pelo tamanho do bloco usado para a codificação. O dicionário de dimensão 1 é inicializado com N_{blocos} com valores espaçados de um valor definido e pertencentes ao conjunto definido pela faixa dinâmica do sinal de voz codificados a 16 bits. As escalas k seguintes são atualizadas a partir de versões expandidas da dimensão 1. Este dicionário é adaptativo, e tem a capacidade de aprender as características do sinal de entrada através do estágio de atualização do dicionário. O MMP utiliza este dicionário adaptativo constituído de

elementos (blocos ou palavras) com dimensões (escalas) diferentes - como mostra a Figura 4.1 - para representar o n-ésimo bloco de entrada, X^n, ou seja, verifica entre os elementos do seu dicionário qual é o que melhor representa o bloco de entrada X^n obedecendo algum critério de desempenho. Podemos, inclusive, ajustar este critério de medida de modo que a compressão seja sem perdas. O dicionário do MMP é dito adaptativo porque ele é atualizado à medida que o sinal de entrada está sendo processado.

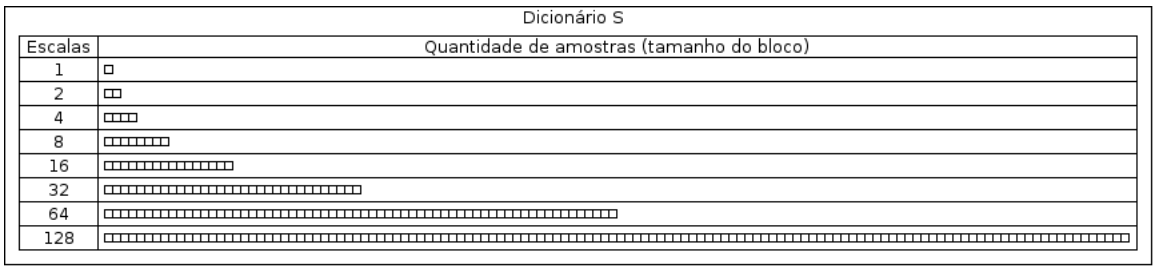


Figura 4.1: Esquema do dicionário do MMP.

Quando o MMP está sendo usado para compressão de um sinal de voz (sinal unidimensional), ele divide o sinal de voz (sinal de entrada X) em blocos, (Figura 4.2), cada um composto de N amostras (na figura, $N = 128$ como exemplo). Cada um dos blocos do sinal de voz será codificado individualmente. Inicialmente, o MMP retira do sinal de voz um bloco X^0 (bloco de entrada ou informação de entrada, Figura 4.2) com tamanho $1 \times N$. Com este bloco, inicia-se o processo de busca na escala do dicionário com a mesma dimensão equivalente ao bloco, para encontrar o vetor que mais se aproxima dele (Figura 4.3), obedecendo ao critério de desempenho. Um exemplo deste tipo de critério é a distorção (D), que pode ser dada por exemplo, pelo erro médio quadrático. Outro exemplo pode ser $D + \lambda R$ onde λ é uma constante e R a taxa. Caso algum dos blocos do dicionário satisfaça ao critério, o bloco é codificado. Caso nenhum dos blocos do dicionário satisfaça ao critério, o bloco de entrada é segmentado em blocos X^1 e X^2 (ver Figura 4.4) e o processo é repetido recursivamente para cada uma das partes. Terminado o processo, cada uma das partes selecionadas cumpre o critério de desempenho sendo codificada em seguida pelo algoritmo. Em seguida, um novo vetor é gerado com a concatenação da aproximação de cada uma das partes, e usado para o estágio de atualização do dicionário. Cada escala do dicionário é atualizada com versões contraídas e dilatadas deste vetor para se adequar à escala do dicionário. Este processo está ilustrado na Figura 4.5. A expansão ou contração das palavras que serão inseridas no dicionário é feita através de uma transformação de escala $T_M^N : \mathbb{R}^M \longrightarrow \mathbb{R}^N$ [53, 54], denotada por $S^s = T_M^N[S]$.

Para a dilatação ($M < N$), o vetor escalonado é dado por:

$$S_n^s = \left\lfloor \frac{\alpha_n \left(S_{m_n^1} - S_{m_n^0} \right)}{N} \right\rfloor + S_{m_n^0} \tag{4.1}$$

com

$$\begin{aligned} m_n^0 &= \left\lfloor \frac{n\,(M-1)}{N} \right\rfloor \\ m_n^1 &= \begin{cases} m_n^0 + 1 & , \quad m_n^0 < M-1 \\ m_n^0 & , \quad m_n^0 = M-1 \end{cases} \\ \alpha_n &= n\,(M-1) - N m_n^0, \end{aligned}$$

para $n = 0, 1, \ldots, N-1$.

Para a contração ($M > N$) o vetor escalonado é dado por:

$$S_n^s = S_{m_{n,k=0}^0} + \frac{1}{M+1} \sum_{k=0}^{M} \left\lfloor \frac{\alpha_{n,k} \left(S_{m_{n,k}^1} - S_{m_{n,k}^0} \right)}{N} \right\rfloor \tag{4.2}$$

com

$$\begin{aligned} m_{n,k}^0 &= \left\lfloor \frac{n\,(M-1)+k}{N} \right\rfloor \\ m_{n,k}^1 &= \begin{cases} m_{n,k}^0 + 1 & , \quad m_{n,k}^0 < M-1 \\ m_{n,k}^0 & , \quad m_{n,k}^0 = M-1 \end{cases} \\ \alpha_{n,k} &= n\,(M-1) + k - N m_{n,k}^0, \end{aligned}$$

para $n = 0, 1, \ldots, N-1$.

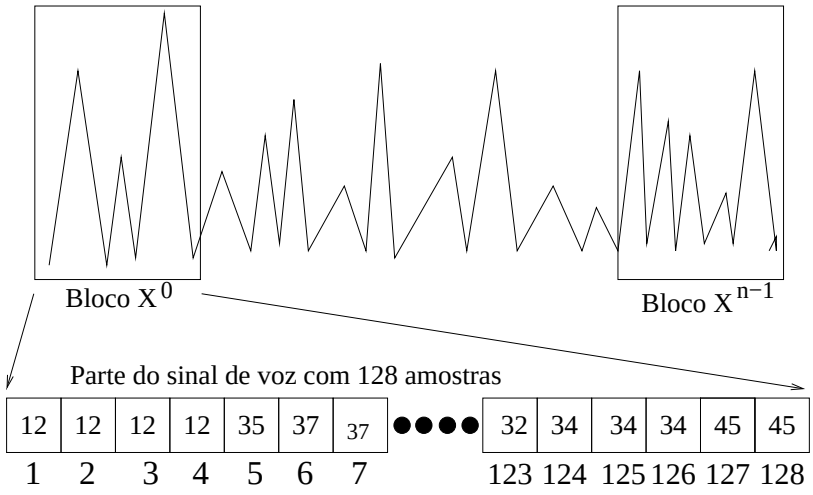


Figura 4.2: Exemplo de uma parte do sinal de voz para codificação.

Assim, cada segmento X^j cuja representação por um elemento do dicionário não

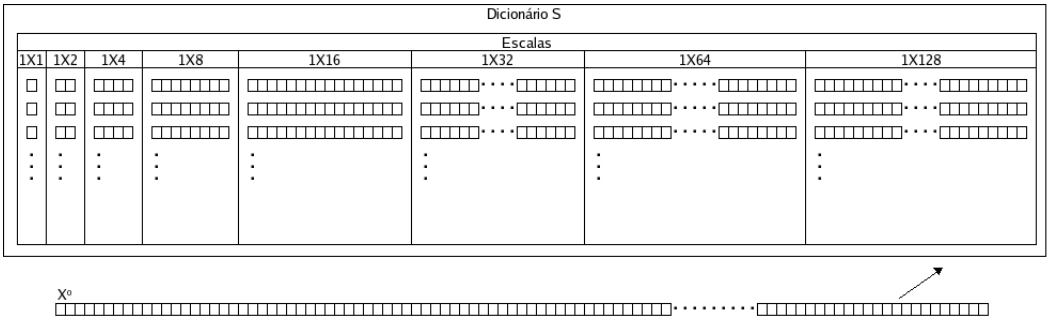


Figura 4.3: Projeção do bloco de entrada X^0 (parte do sinal de voz) no dicionário.

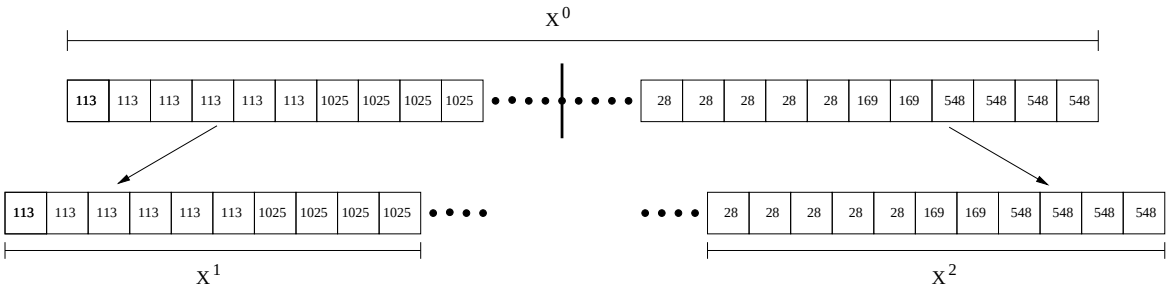


Figura 4.4: Exemplo do Bloco X^0 segmentado.

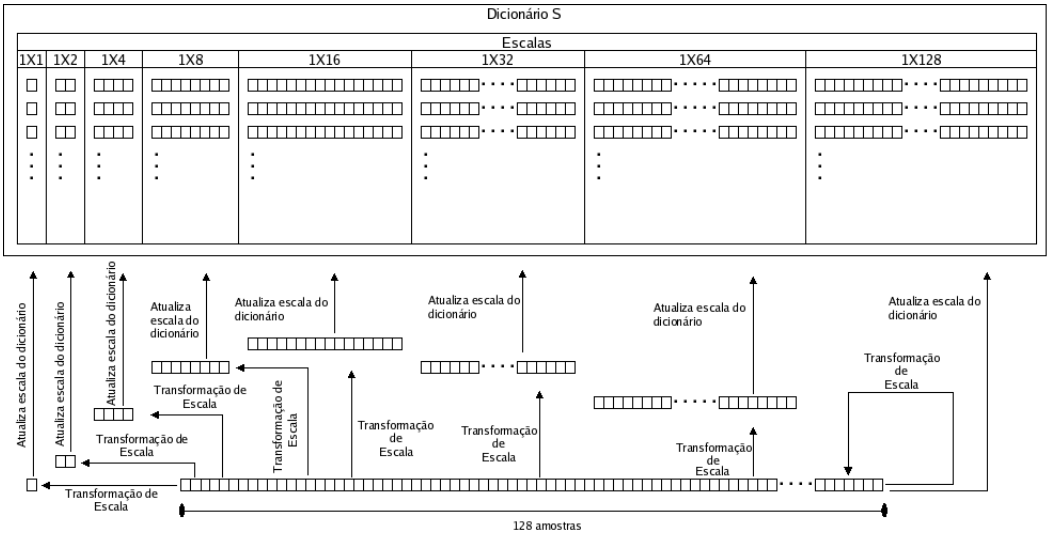


Figura 4.5: Atualização do dicionário.

satisfaça ao critério de desempenho pode ser segmentado em dois segmentos X^{2j+1} e X^{2j+2}; cada um destes segmentos pode ser segmentado recursivamente se eles não puderem ser representados convenientimente com palavras do dicionário. Isto pode criar uma segmentação do bloco de entrada em blocos de comprimentos diferentes, e pode ser representado por uma árvore binária como mostra a Figura 4.6.

O dicionário é atualizado sempre que as aproximações $\widehat{X}^{2j+1}$ e $\widehat{X}^{2j+2}$ estejam disponíveis. No caso da Figura 4.7, o segmento $\widehat{X}^1$ só será codificado e o dicionário atualizado quando as aproximações $\widehat{X}^3$ e $\widehat{X}^4$ estiverem disponíveis. A aproximação $\widehat{X}^1$ será a concatenação de $\widehat{X}^3$ e $\widehat{X}^4$. O dicionário atualizará todas as suas escalas

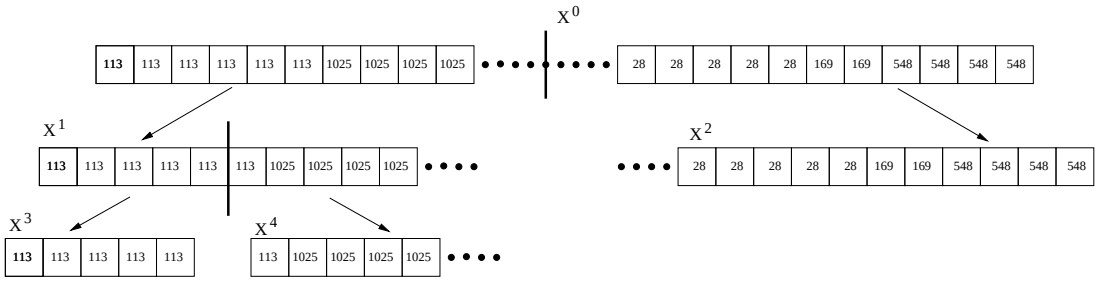


Figura 4.6: Árvore de segmentação do Bloco de entrada X^0.

através da transformação de escalas, que contrairá ou dilatará os blocos de acordo com a escala do dicionário a ser atualizada.

Por exemplo, um bloco X^0 será comparado com as palavras do dicionário de escala correspondente para verificar se existe alguma aproximação no dicionário que satisfaça ao critério de desempenho (Figura 4.3). Caso nenhum bloco do dicionário satisfaça ao critério de desempenho, o bloco X^0 será segmentado em X^1 e X^2, como mostra a Figura 4.4. Cada um desses segmentos será novamente comparado com as palavras do dicionário de escala correspondente verificando se ele satisfaz ao critério de desempenho. Caso o segmento X^1 não satisfaça ao critério de desempenho, o mesmo será dividido em X^3 e X^4 (Figura 4.6). Os segmentos X^3 e X^4 passam pelo mesmo processo que o segmento X^1. Se os segmentos X^3 e X^4 satisfazem ao critério de desempenho, obtêm-se as aproximações $\widehat{X}^3$ e $\widehat{X}^4$. Uma vez satisfeito o critério os segmentos $\widehat{X}^3$ e $\widehat{X}^4$ serão codificados cada um e a aproximação de X^1 será a concatenação de $\widehat{X}^3$ e $\widehat{X}^4$. Após a codificação das aproximações $\widehat{X}^3$ e $\widehat{X}^4$, é a vez de o segmento X^2 passar pelo mesmo processo. Com a aproximação do segmento X^2 satisfazendo ao critério de desempenho, obtemos a aproximação $\widehat{X}^2$. Portanto um bloco X^0 terá sua aproximação como mostra a Figura 4.7, e através da concatenação desses segmentos obtém-se $\widehat{X^0} = (\widehat{X}^3\widehat{X}^4\widehat{X}^2)$.

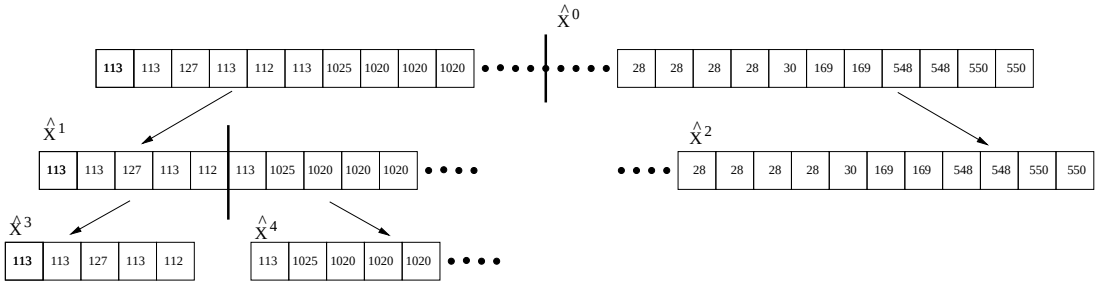


Figura 4.7: Árvore de segmentação aproximada do Bloco de Entrada X^0.

O MMP codifica os blocos somente quando as aproximações atendem ao critério de desempenho. Ele produz um arquivo contendo uma sequência de bits que representam os índices dos elementos do dicionário para cada segmento extraído do bloco de entrada, e os *flags* usados para informar sobre a árvore de segmentação (informa

se um nó da árvore corresponde a uma segmentação de bloco ou não). Por exemplo, se o *flag* é "*1*" o nó possui dois filhos e corresponde a uma segmentação de bloco; caso contrário o *flag* é "*0*". Para indicar qual elemento do dicionário corresponde a um bloco codificado, será usado um índice, ao qual será atribuído um código. Nas seções a seguir serão descritas várias melhorias ao algoritmo básico descrito aqui.

4.2 Evoluções do MMP

Nesta seção descrevemos dois melhoramentos ao algoritmo MMP descrito na seção anterior. Isto será feito através da descrição sequencial de várias versões do MMP, do básico (já explicado nas seções anteriores) até à versão que usa otimização taxa-distorção. Cada fase deste processo é comentada a seguir. Ao fim desta seção teremos descrita a versão do MMP que foi usada como base para este trabalho.

4.2.1 MMP com Codificador Aritmético

Como visto na Seção 4.1, o MMP codifica um sinal de voz através de uma sequência de *flags* que indica as segmentações dos blocos e índices para elementos dos dicionários; com o propósito de eliminar as redundâncias nesta representação, diminuindo a taxa de bits, esta sequência de *flags* e índices será codificada usando um codificador aritmético adaptativo [55]. Diferentemente das outras técnicas, a codificação aritmética representa a informação por um subintervalo do intervalo [0,1] da reta real, isto é, ele a produz como saída um intervalo ótimo para um dado conjunto de símbolos e probabilidades.

O codificador aritmético atribui intervalos aos símbolos do alfabeto de entrada. Cada intervalo terá tamanho proporcional à probabilidade de ocorrência de cada símbolo ao qual está associado. O codificador transmitirá somente a quantidade de dígitos suficientes para especificar a fração que pertence ao intervalo, de tal forma que todas as frações que comecem com esses dígitos pertençam ao mesmo intervalo representado, alocando números dentro dos intervalos correspondentes aos símbolos que recebe. Os símbolos com maior probabilidade de ocorrência geram intervalos que necessitam de poucos dígitos para serem especificados, e os símbolos com probabilidade baixa geram intervalos estreitos, que precisam de muitos dígitos para serem especificados. As probabilidades usadas para a codificação dos símbolos são estimadas com base na frequência de ocorrência de cada símbolo codificado antes do símbolo corrente. Neste trabalho, usamos um codificador aritmético adaptativo em vários contextos (são usados diferentes histogramas para estimar a codificação dos símbolos); um exemplo é a codificação dos *flags* e índices, que é condicionada à escala do bloco.

4.2.2 MMP com Otimização Taxa-distorção (MMPRD)

Nesta implementação do MMP há uma otimização na árvore de segmentação. Esta otimização é chamada de Otimização Taxa-Distorção [53], que otimiza a árvore de segmentação preocupando-se tanto com a taxa de bits quanto com a distorção. Neste caso, dentre todas as árvores possíveis escolhe-se a árvore que possui o menor custo, $J = D + \lambda R$, onde D é a distorção na representação do bloco, R é a sua taxa e λ é um multiplicador Lagrangeano que dá a importância relativa entre a taxa e a distorção. Esta análise depende dos nós pais e filhos. A próxima subseção explicará com mais detalhes esta otimização.

O MMP é um algoritmo de compressão de dados com perdas que utiliza um dicionário S (Figura 4.8) e segmentação de um bloco de entrada X. O bloco é dividido em segmentos de comprimento L. Cada bloco é segmentado sendo os segmentos codificados usando um elemento do dicionário S na escala apropriada e que atenda ao critério de desempenho, como descrito neste capítulo. Assim obtemos o bloco aproximação de X^0 ($\widehat{X^0}$). A segmentação resultante pode ser representada por uma árvore de segmentação, conforme a Figura 4.7; uma forma mais compacta para a sua representação é dada pela Figura 4.9.

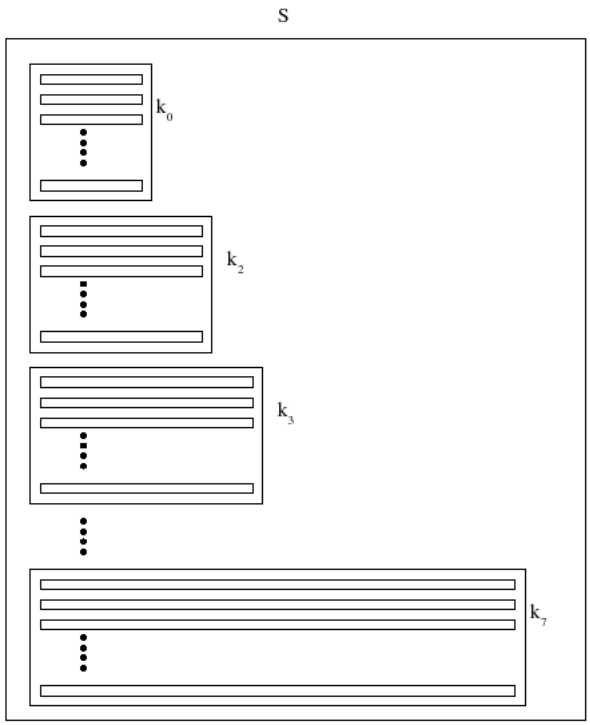


Figura 4.8: Estrutura do dicionário do MMP para sinal de voz.

Cada segmento utilizado para aproximar um bloco X^0 pode ser considerado como nó folha, sem descendentes, da árvore . Podemos observar pela Figura 4.9 que o bloco de entrada X^0 foi segmentado em 3 segmentos, ou seja, $X^0 = (X_3X_4X_2)$. Cada um dos segmentos foi aproximado por um bloco do dicionário S, com alguma

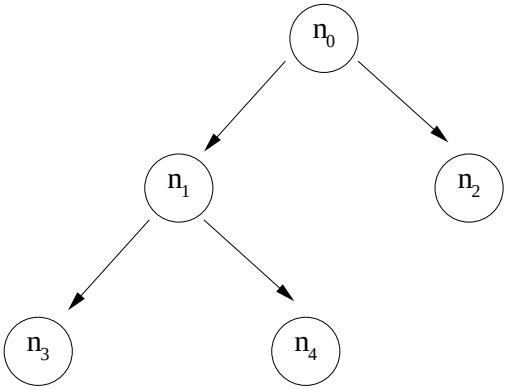


Figura 4.9: Árvore de segmentação representada com nós folhas.

distorção, e podemos associar cada nó folha da árvore a um valor de distorção, e cada nó está associado a algum segmento de X^0. Por exemplo, o nó n_1 está associado a um segmento X^1 que não foi aproximado por uma única palavra do dicionário, porque este foi segmentado em $X^1 = (X^3X^4)$. Portanto podemos associar a cada nó uma distorção $D(n_l)$. No caso da Figura 4.9, esta distribuição será dada pela soma das distorções associadas aos nós n_3 e n_4. Já no caso do nó n_2, como ele não corresponde a uma segmentação, a distorção corresponderá a sua aproximação por um elemento do dicionário. E também definimos $R(n_l)$ como a taxa necessária para especificar o índice i_l do dicionário S, ou seja,

$$R(n_l) = -log_2(Pr(i_l)) \tag{4.3}$$

O critério de desempenho, neste caso, será expresso para cada nó através do custo lagrangeano [53, 54, 56], que define a relação entre a distorção e a taxa para um nó em particular:

$$J(n_l) = D(n_l) + \lambda R(n_l), \tag{4.4}$$

onde:

- $J(n_l)$ é o custo do nó n_l.
- $D(n_l)$ é a distorção entre o bloco de entrada X associado ao nó n_l e a sua aproximação do dicionário S com a mesma dimensão do bloco de entrada X.
- $R(n_l)$ é a taxa para representar o índice i_l do dicionário S.
- λ é o fator ponderador entre a taxa e distorção.

A otimização pelo método de Lagrange foi primeiramente introduzida em [57], podemos encontrar aplicações deste método em [58, 59, 60]

Agora em vez de verificarmos a distorção num determinado nó, o que será verificado é o custo deste nó (n_l).

Um multiplicador de Lagrange é escolhido, e a partir deste valor calcula-se o custo do nó. Vale lembrar que neste momento calculamos tanto o custo do nó folha, pai, (n_l) quanto dos seus nós descendentes, filhos, (n_{2l+1} e n_{2l+2}). Verifica-se o custo do nó pai (n_l), através do multiplicador de Lagrange, distorção e taxa do índice, e dos nós filhos. Caso o custo do nó pai (n_l), $J(n_l) = D(n_l) + \lambda R(n_l)$ seja menor ou igual ao custo dos nós filhos + a taxa do *flag* de segmentação, como mostra a equação abaixo,

$$J(n_l) < J(n_{2l+1}) + J(n_{2l+2}) + \lambda R(flag(n_l)), \tag{4.5}$$

então o nó pai mantém-se e os nós filhos da árvore são descartados; assim, o nó pai passa a ser um nó folha ou um dos nós filhos de um nó folha ascendente. Caso contrário, os nós filhos mantêm-se (Figura 4.9).

Neste capítulo descrevemos o MMP básico, que será usado como ponto de partida para desenvolver os codificadores de voz estudados neste trabalho. No capítulo a seguir, serão descritas algumas modificações iniciais neste algoritmo com o intuito de torná-lo mais adequado à codificação de voz.

Capítulo 5

Adaptação do MMP à codificação de voz

Atualmente os codificadores que apresentam uma boa taxa de compressão com nível de qualidade de voz aceitável são os codificadores híbridos, da família CELP (detalhada, por exemplo, em [39, 40, 61]) ou similares, que combinam as vantagens dos codificadores paramétricos com as dos codificadores de forma de onda.

Neste trabalho objetivamos investigar a viabilidade de aplicar o princípio da recorrência de padrões multiescalas (MMP) na codificação de sinal de voz. Algoritmos baseados neste princípio [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 53] apresentam um certo processo de aprendizado que pode e deve ser adaptado aos sinais de voz. Inicialmente foi utilizado um algoritmo básico do MMP descrito em [4] e depois ele foi implementado com algumas modificações que pudessem redundar em melhorias no desempenho do algoritmo, e assim obter maiores taxas de compressão para um dado nível de qualidade de voz.

Nas próximas seções descrevemos a implementação do MMP modificado para codificação de voz, como representado nas Figuras 5.1 (codificador) e 5.2 (decodificador).

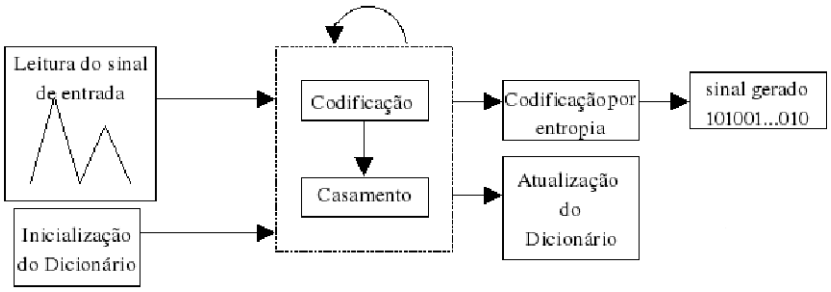


Figura 5.1: Digrama de blocos do Codificador MMP para sinais de voz.

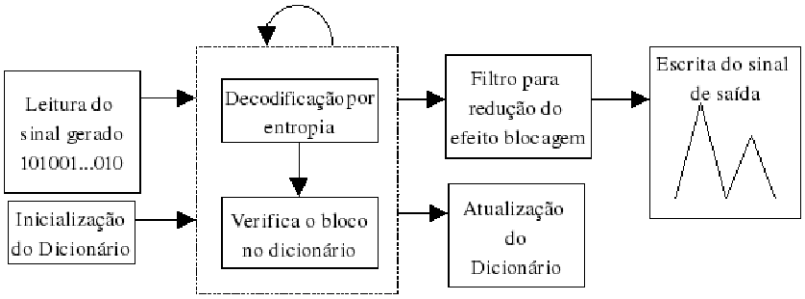


Figura 5.2: Digrama de blocos do Decodificador MMP para sinais de voz com o filtro de redução do efeito blocagem.

5.1 Leitura do sinal de entrada

O algoritmo MMP pode ser adaptado facilmente para ler arquivos de qualquer formato, bastando apenas alterar este módulo de leitura. No caso desta tese, este módulo foi alterado para capacitar o algoritmo MMP a ler arquivos com formato *.wav*.

Nesta fase é escolhido o tamanho do bloco a ser codificado (Tabela 5.1), o que influencia na taxa de compressão do algoritmo e na complexidade computacional requerida. No caso da taxa de compressão, ela se reduz drasticamente se forem utilizados blocos de tamanho muito grande e se existirem casamentos desses blocos com os blocos que se encontram nas escalas mais altas do dicionário, pois assim serão necessários poucos índices para codificar o sinal. Porém, se não houver muitos casamentos com os blocos das escalas mais altas do dicionário, serão necessários mais *flags* na saída para informar que o bloco foi dividido, levando a um *overhead* na taxa de compressão. A complexidade computacional cresce diretamente com o aumento do tamanho do bloco, já que a árvore binária correspondente passa a ter mais níveis.

Tabela 5.1: Parâmetro de entrada do codificador

$n_{amostras}$	Tamanho do bloco (n^o de amostras)
n_{128}	128
n_{64}	64
n_{32}	32
n_{16}	16
n_8	8
n_4	4

Um sinal de voz pode ser considerado estacionário em intervalos de 10 ms a 30 ms, que corresponde a um conjunto de 80 – 240 amostras numa taxa de amostragem de 8 kHz. No codificador CELP o sinal de voz é segmentado em intervalos de 20 ms que corresponde a 160 amostras. Procurando investigar o desempenho do MMP,

comparando com o do codificador CELP, o sinal de voz no MMP será segmentado em intervalos X_i de $n_{amostras} = 128$ amostras, como indicado na Figura 5.3, o que corresponde a 16 ms, que é o intervalo de operação mais próximo do codificador CELP, uma vez que o MMP opera com blocos com potência de 2.

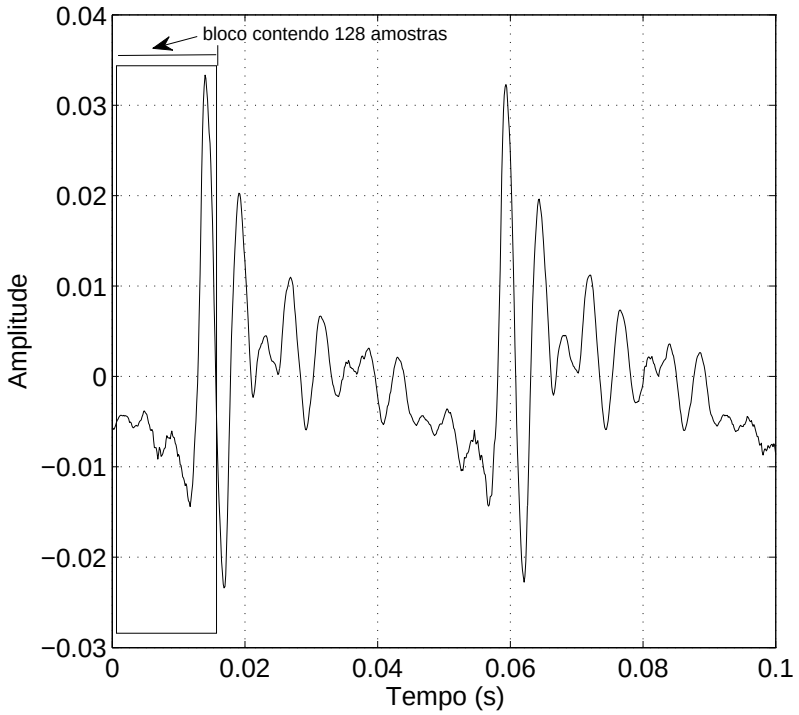


Figura 5.3: Exemplo de um bloco retirado do sinal de voz para processamento ($f_s =$ 8 kHz).

5.2 Inicialização do dicionário

Após escolhido o tamanho do bloco a ser codificado, é necessário definir a estrutura do dicionário S. Este dicionário contém k escalas, cada uma composta por elementos de uma determinada dimensão, onde

$$k = \log_2(n_{amostras}) + 1, \tag{5.1}$$

de modo que a última escala seja composta de elementos de dimensão 1 (escalares). No caso desta tese, tem-se então um total de $k = 8$ escalas. A amplitude dos vetores que constituem o dicionário inicial deve conter toda a faixa dinâmica do sinal de entrada, de forma a uma melhor representação. Então, para que a estrutura do dicionário inicial seja completamente definida, devemos especificar as amplitudes máxima, $max(X)$, e mínima, $min(X)$, do sinal de entrada e o passo de quantização,

q, utilizado. O número de blocos, N_{blocos}, de cada escala de tamanho 2^i, para $i = 0, .., (k-1)$ encontra-se relacionado com estes parâmetros através da relação

$$N_{blocos} = \frac{max(X) - min(X)}{q} + 1. \tag{5.2}$$

Desta forma, o dicionário tem sua escala de dimensão 1 inicializada com N_{blocos}. Tais blocos serão igualmente espaçados de um valor definido pela variável q, e terão valores pertecentes ao conjunto definido pela faixa dinâmica do sinal de voz codificados a 16 bits. As demais escalas do dicionário são determinadas a partir de versões dilatadas da dimensão 1 de acordo com o comprimento do bloco na escala correspondente.

Uma característica importante do algoritmo MMP é que o mesmo dicionário construído pelo codificador pode ser construído também pelo decodificador, não havendo assim necessidade de transmiti-lo.

5.3 Casamento dos blocos

Após a inicialização da estrutura do dicionário, inicia-se o processo de codificação. Primeiramente, temos que encontrar dentro do dicionário o bloco que mais se aproxima do (ou seja, o que possui melhor casamento) com o que está sendo codificado. Esta decisão se baseia na otimização da segmentação da árvore que minimiza o custo lagrangeano [53, 54, 56] combinando a taxa de codificação e a distorção correspondente. O custo langrangeano diz que se o custo do nó pai for menor que o custo dos nós filhos, então os nós filhos são descartados e o nó pai é usado na codificação. Caso contrário, o bloco é dividido e um novo cálculo do custo é feito para os novos nós. O processo é efetuado até que o bloco de entrada fique completamente codificado e assim passa-se para um outro bloco lido e inicia-se sua codificação.

5.4 Atualização do dicionário

Quando determinamos a árvore ótima que minimiza o custo lagrangeano para um dado bloco de entrada, o dicionário MMP deve ser atualizado. Esta atualização provê um certo aprendizado ao algoritmo MMP em relação às características intrínsecas ao sinal de entrada. Nesse processo, o dicionário MMP começa com seus elementos iniciais, como visto na Subseção 5.2, e depois vai incorporando outros blocos que foram obtidos durante o processo da codificação, fazendo com que ele "aprenda" os padrões que ocorreram na parte do sinal de entrada já processado. Quando na árvore que otimiza o custo lagrangeano há dois nós filhos de um nó n_l, então uma palavra do dicionário é gerada concatenando as aproximações dos dois

nós-filhos e suas versões dilatadas e contraídas são inseridas no dicionário (Figura 4.5). A inclusão destas versões no dicionário é feita através de uma transformação de escala $T_N^M : \mathbb{R}^M \longrightarrow \mathbb{R}^N$ [53]. Porém, antes destas palavras serem inseridas no dicionário é realizado um processo de busca no dicionário para verificar se já existe um bloco igual ou similar, de forma a evitar blocos redundantes. As versões dilatadas e contraídas do bloco são essenciais para o aprendizado mais rápido do dicionário. A tendência é de que as escalas mais altas do mesmo contenham blocos cada vez mais similares aos blocos de entrada, e assim teremos menos divisões da árvore ótima, acompanhadas de uma diminuição da taxa de codificação.

No estágio de atualização do dicionário, novos elementos são incluídos se não existirem no dicionário. Desta forma, o dicionário cresce e dependendo da escala em questão, o crescimento pode ser mais lento ou mais rápido, pois a existência de elementos repetidos acaba sendo mais provável. Pode-se observar este fato usando a escala de dimensão 2 do dicionário como exemplo. Se um elemento com dimensão 32 for incluído nesta escala com a devida contração, o bloco transformado terá grande possibilidade de já existir, o mesmo acontece para escala com dimensão 1. Portanto, escalas pequenas tendem a saturar mais rapidamente, pois o número de elementos possíveis é menor. O dicionário não pode crescer indefinidamente devido às restrições de memória, por isso o tamanho máximo do dicionário é fixado em $L = 450.000$ elementos.

5.5 Codificação por entropia

O uso de um codificador por entropia tem como objetivo eliminar toda a redundância na informação. A sua utilização conduz a uma redução muito significativa da taxa sem que seja introduzido qualquer tipo de distorção ao sinal gerado.

Uma das opções para um codificador de entropia é, como referido na Subseção 4.2.1, o uso de um codificador aritmético. A codificação aritmética é um nétodo de codificação por entropia que, diferentemente das outras técnicas que separam a mensagem de entrada em símbolos e as substituem por palavras, funciona codificando toda a mensagem em um número n contido no intervalo de 0 a 1.

A grande vantagem da codificação aritmética é a facilidade com que se podem tornar adaptativos os seus modelos, podendo-se assim descobrir a distribuição probabilística de símbolos do sinal sem ter maiores informações a priori.

O codificador funciona alocando números dentro dos intervalos correspondentes aos símbolos que recebe, subdividindo esses intervalos de acordo com o modelo de dados à medida que cada codificação acontece.

5.6 Filtro para redução do efeito de blocagem

Como o MMP é um algoritmo baseado em blocos, é comum acontecerem efeitos de blocagem [14, 62] que correspondem a descontinuidades entre blocos consecutivos que são codificados de forma independente uns dos outros. Estas descontinuidades, na codificação de sinal de voz, correspondem a ruídos de alta frequência nas fronteiras entre os blocos codificados. Os erros provocados pela blocagem são mais perceptíveis para taxas menores de codificação, quando as diferenças entre os sinais original e reconstruído são maiores, o mesmo se dando para as descontinuidades entre os blocos.

A princípio o MMP não impõe qualquer restrição à escolha dos elementos do dicionário usados para aproximar um dado bloco X^n que leve em consideração a continuidade entre segmentos. Uma forma de controlar a blocagem seria alterar o cálculo do custo utilizado no procedimento de busca de modo a considerar explicitamente a descontinuidade no ponto de segmentação. Uma outra possibilidade seria a utilização de blocos com sobreposição. Nesta abordagem, a soma dos comprimentos dos N blocos é maior que o comprimento total do sinal de entrada X, uma vez que dois vetores adjacentes não são apenas concatenados, mas adicionados com sobreposição de alguns elementos. Computacionalmente, a utilização de sobreposição no codificador torna o processo de codificação bastante complexo, pois fica difícil encontrar a melhor árvore de segmentação na codificação, já que a distorção associada a cada nó passa também a depender dos blocos usados para aproximação dos blocos adjacentes.

Em [53] foi proposta uma solução alternativa às anteriores, onde apenas são utilizados vetores com sobreposição para realizar a reconstrução do sinal pelo decodificador. Esta solução usa um filtro de média móvel com comprimento ajustado a cada amostra de maneira a coincidir com o tamanho do segmento que está sendo processado (filtro gaussiano). A Figura 5.4 mostra um bloco de saída composto por três segmentos, onde a seta indica o elemento que está sendo filtrado. O comprimento do filtro FIR é proporcional ao tamanho original do bloco de reconstrução. Como cada palavra do dicionário é em última instância composta por concatenações de palavras do dicionário inicial, que possui apenas blocos constantes, cada segmento na figura ou é constante ou é gerado por uma interpolação linear aplicada a um sinal constante por partes. Desta forma, o filtro pode ter tamanho igual ao do segmento sem que haja perda de informação significativa. A vantagem é que há a atenuação do efeito de blocos, pois nas bordas dos blocos é feita uma média ponderada entre os valores dos segmentos. Note que neste caso o decodificador deve manter uma lista contendo a composição de cada vetor do dicionário em função dos elementos do dicionário inicial que o compõem. O filtro gaussiano possui comprimento igual a

$(N+1)$ amostras, onde N é o tamanho do segmento onde o filtro está centrado.

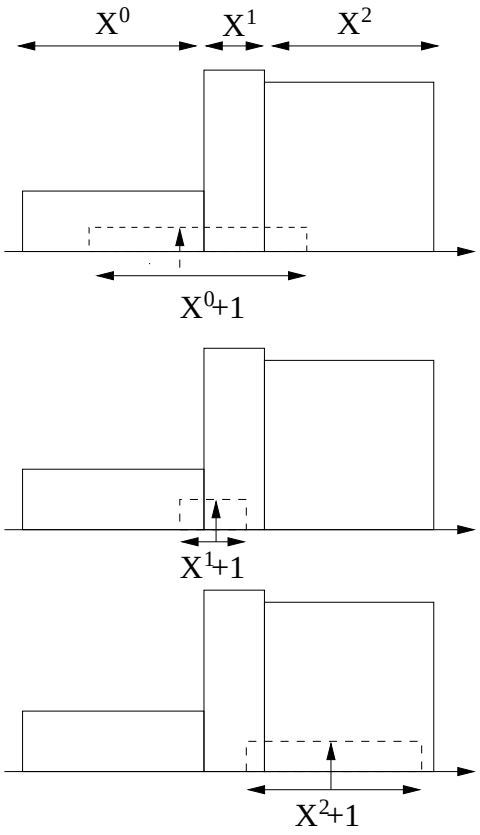


Figura 5.4: Variação do filtro adaptativo usado no efeito blocagem.

Neste trabalho foi utilizado um filtro gaussiano proposto em [54], com comprimento L variando de acordo com a variância σ^2 do sinal e resposta ao impulso

$$g^{k_i}(n) = e^{-\frac{\left(n-\frac{L_i-1}{2}\right)^2}{2(\alpha L_i)^2}}; \quad n = 0, 1, \ldots, (L_i - 1), \tag{5.3}$$

onde k_i é a escala do dicionário MMP, $L_i = 2^{k_i}$, e o parâmetro α controla a variância da resposta. A Figura 5.5 mostra um exemplo de resposta ao impulso dos filtros variantes no tempo (diferente para cada bloco) para diferentes valores do parâmetro α.

A vantagem do controle do parâmetro α é que podemos evitar que o filtro de efeito blocagem sofra influência das amostras que não sejam do bloco atual e nem de seus vizinhos imediatos. Ou seja, quando o filtro de blocagem processa o bloco X^2 (parte da concatenação de $X^0X^1X^2$ para gerar um bloco aproximação X^n) com uma ampla região, ele é influenciado pela blocagem nas bordas do bloco X^0 concatenado com X^1, como representado pela linha tracejada na Figura 5.6. Para evitar que isto aconteça, o parâmetro α é ajustado de forma que esta região diminua evitando este efeito na resposta do filtro, como indicado pela linha sólida na Figura 5.6.

O surgimento de descontinuidade entre os blocos codificadores é um problema à

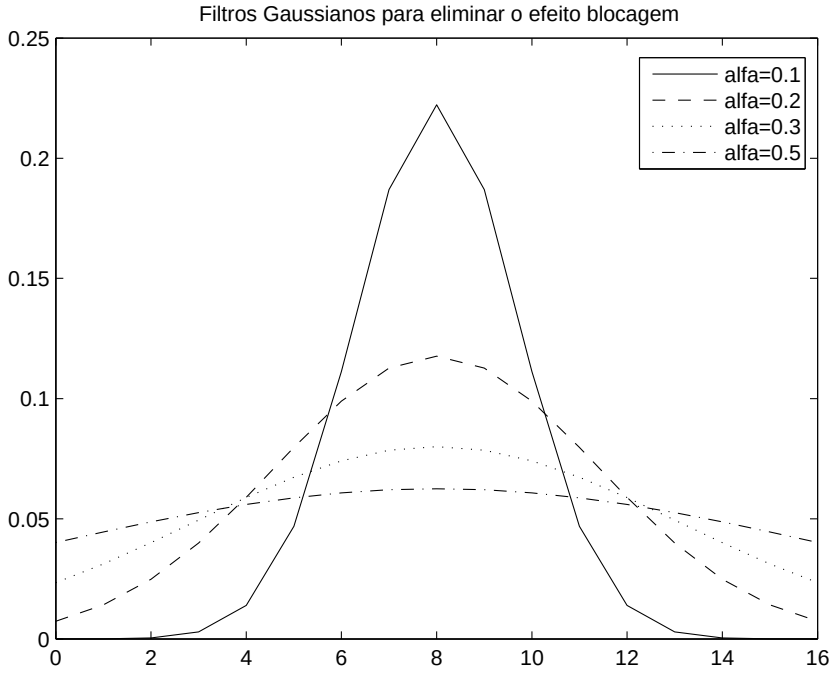


Figura 5.5: Resposta ao impulso dos filtros adaptativos usados para eliminar o efeito blocagem.

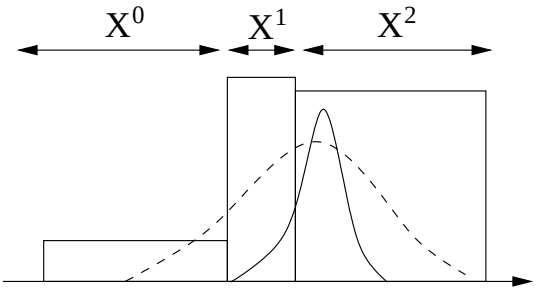


Figura 5.6: Exemplo da redução do efeito de blocagem.

estrutura usada no algoritmo MMP. Para tentar minimizar o aparecimento destes artefatos no sinal reconstruído, um modelo de pós-filtragem é usado. A Figura 5.7 mostra a comparação entre o sinal original, sinal de decodificado e o sinal reconstruído pós-filtragem. Neste exemplo, foi utilizado um pequeno trecho do sinal de voz para melhor visualização da redução do efeito de blocagem. Observa-se na figura que trechos que apresentam descontinuidade nas fronteiras entre os blocos codificados (trechos circulados) foram atenuadas com o uso da pós-filtragem (uso de um filtro gaussiano), mostrando a eficácia deste filtro, permitindo que o decodificador reduza efeitos destrutivos no sinal de voz e incrementando a qualidade perceptual do sinal codificado.

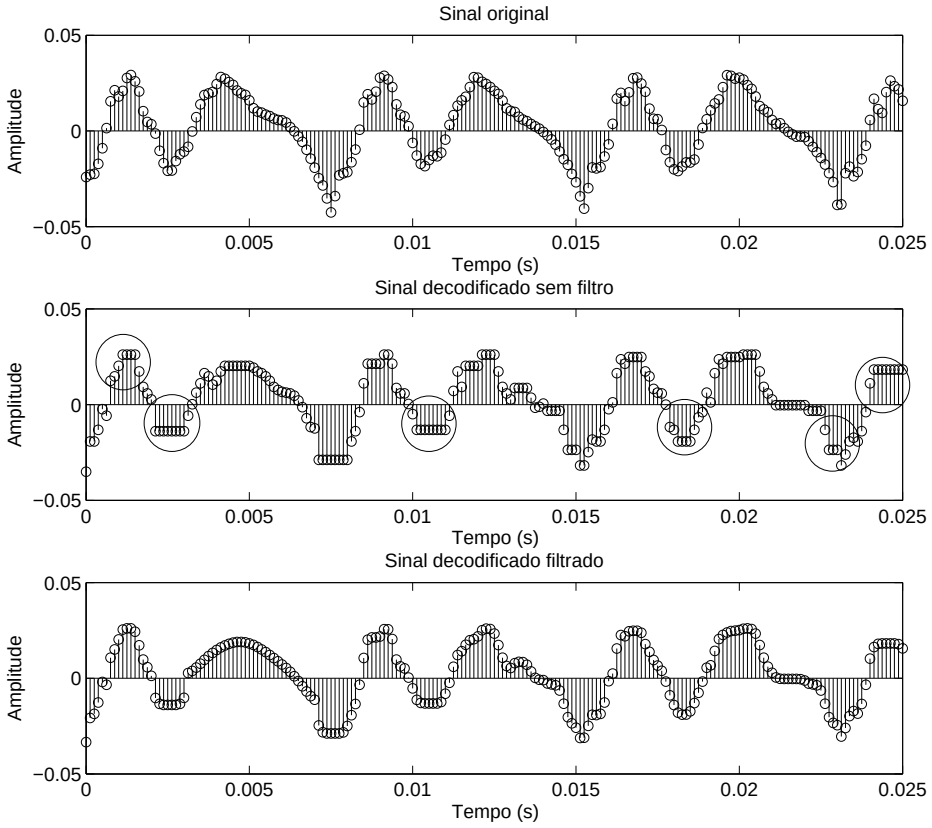


Figura 5.7: Exemplo de redução do efeito de blocagem.

5.7 MMP com blocos de deslocamento fracionado

Nesta tese, o dicionário MMP também é atualizado com blocos obtidos a partir do deslocamento de uma janela pela parte do sinal que já foi codificada. Seu comprimento pode ser múltiplo de 1/2, 1/4 e 1/8 do tamanho do bloco, e seu deslocamento pode se dar com passos múltiplos de 1, 1/2 e 1/4 de amostra. Este processo é representado na Figura 5.8, que mostra uma janela de tamanho 1×128 sendo deslocada pela parte do sinal de voz que já foi codificada. Em cada posição desta janela temos um bloco que será incorporado ao dicionário, com a devida transformação para atualizar cada escala do dicionário.

Este incremento no processo de atualização do dicionário MMP provê uma espécie de aceleração ao processo de aprendizado inerente ao algoritmo original, com o fornecimento de uma quantidade muito maior de padrões. Se o sinal possui estruturas que tendem a se repetir, como no caso de segmentos quase periódicos, a inclusão destes padrões pode levar a um aumento significativo do desempenho do codificador.

Uma outra forma de uso destes padrões é o dicionário auxiliar, contendo os resultados das codificações de blocos recentes. Este dicionário é diferente para cada

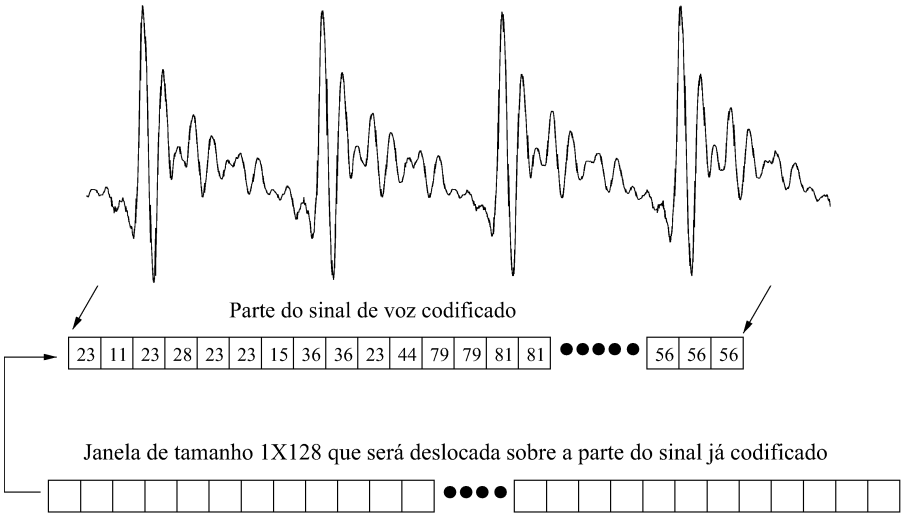


Figura 5.8: Exemplo do deslocamento de janelas com comprimentos determinados para atualização do dicionário MMP.

bloco, e seus vetores são indexados pelo deslocamento associado. Mais detalhes desta implementação serão dados no Capítulo 6, onde apresentamos um codificador de voz completo baseado no MMP, juntamente com a análise do seu desempenho taxa-distorção. Este codificador irá incorporar as características descritas nesta seção, além de outras que serão adicionadas com a finalidade de melhorar o seu desempenho taxa-distorção.

Capítulo 6

Codificação de forma de onda de voz usando MMP

No campo da codificação de voz, o MMP pode ser classificado como um codificador por forma de onda, uma vez que atua diretamente no domínio temporal [33]. A eficiência do MMP baseia-se fortemente na sua capacidade de aprendizagem ao longo do processo de atualização do dicionário. Na prática, porém, este processo de aprendizagem pode facilmente incorporar características temporais, espectrais e até mesmo perceptuais de determinado sinal, que no nosso caso particular é a voz. Ao fazer isso, o algoritmo MMP acaba modelando o sinal comprimido de uma forma não paramétrica através do conteúdo de seu dicionário [63].

Uma simples análise de sinais de voz no domínio do tempo indica que erros de codificação em amostras de grande amplitude são menos perceptíveis que em intervalos de pequena amplitude. Por esta razão, um quantizador não-uniforme, seguindo a lei-μ [33] é empregado por codificadores de forma de onda como o ITU-T G.711 [37], forçando uma mesma razão sinal-ruído (SNR) em toda a faixa dinâmica do sinal de voz. Este procedimento pode ser incorporado ao algoritmo MMP compondo o dicionário com segmentos de amplitudes distribuídas de acordo com a lei-μ, ao invés da quantização uniforme, como representado nas Figuras 6.1 e 6.2. Esta operação deve ser aplicada tanto ao dicionário inicial como durante seu estágio de atualização, incorporando uma característica perceptual no domínio do tempo ao processo de aprendizagem do MMP.

6.1 MMP com dicionário uniforme - MMP-UNI

Nesta seção investigamos o comportamento do MMP com seu dicionário composto de segmentos de diferentes amplitudes distribuídas de forma uniforme. Nos resultados mostrados aqui, consideramos um banco de dados DB1 composto de 10 frases fone-

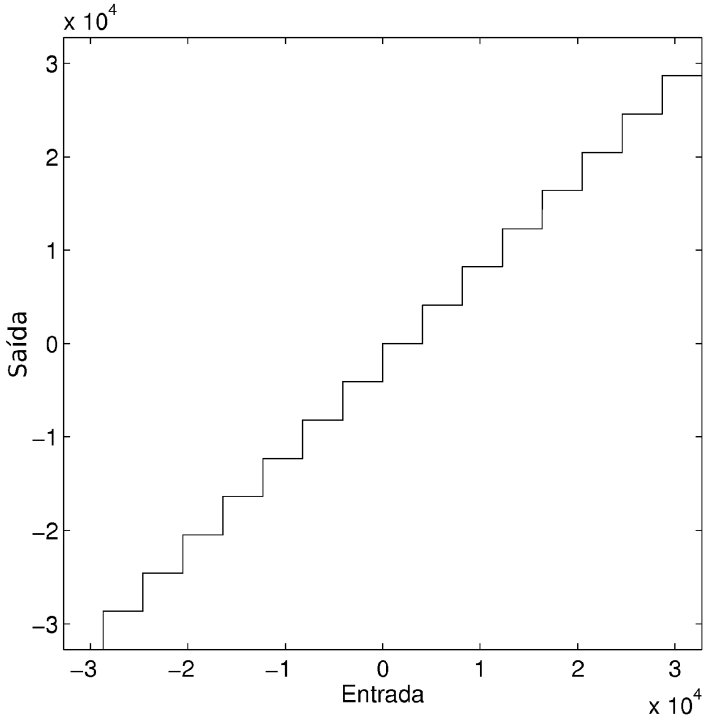


Figura 6.1: Quantização do dicionário do MMP: dicionário uniforme.

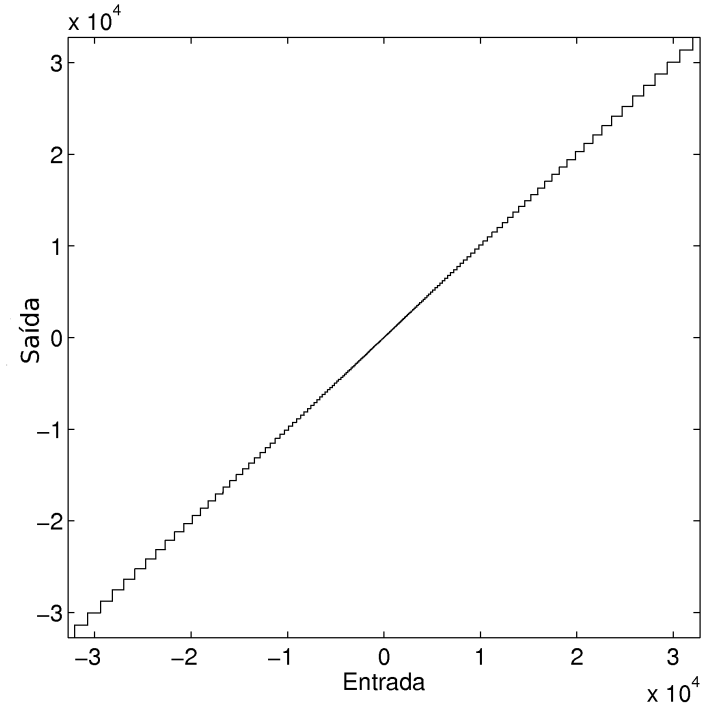


Figura 6.2: Quantização do dicionário do MMP: dicionário não-uniforme.

ticamente balanceadas para o Português Brasileiro [64], amostradas a 8 kHz e com precisão de 16 bits. A qualidade resultante do processo de codificação/decodificação do MMP-UNI é avaliada de forma objetiva com a medida *Perceptual Evaluation of*

Speech Quality (PESQ) [35], devidamente mapeada na escala MOS 1–5. Os valores PESQ-MOS para os codificadores padrões de voz, ITU-T G.711 (PCM) [37], ITU-T G.726 (ADPCM) [38] e ITU-T G.729 (CS-ACELP), apresentados nestes e nos próximos experimentos são resultados médios obtidos para as bases de dados usadas neste trabalho. Os resultados que foram mostrados na Seção 3.3, são valores médios estimados do desempenho dos codificadores padrões de voz.

Foram realizados experimentos com o algoritmo MMP alterando o tamanho do dicionário inicial para diferentes valores: $128, 256, 512, 1024, 2048$ e 4096 elementos, para cada escala do dicionário. Os resultados taxa-distorção para cada tamanho do dicionário inicial podem ser vistos na Figura 6.3, de onde se observa que, para esta base de dados, o melhor tamanho do dicionário inicial compreende 256 elementos para cada escala do dicionário. Uma melhor visualização dos resultados é obtida na Figura 6.4.

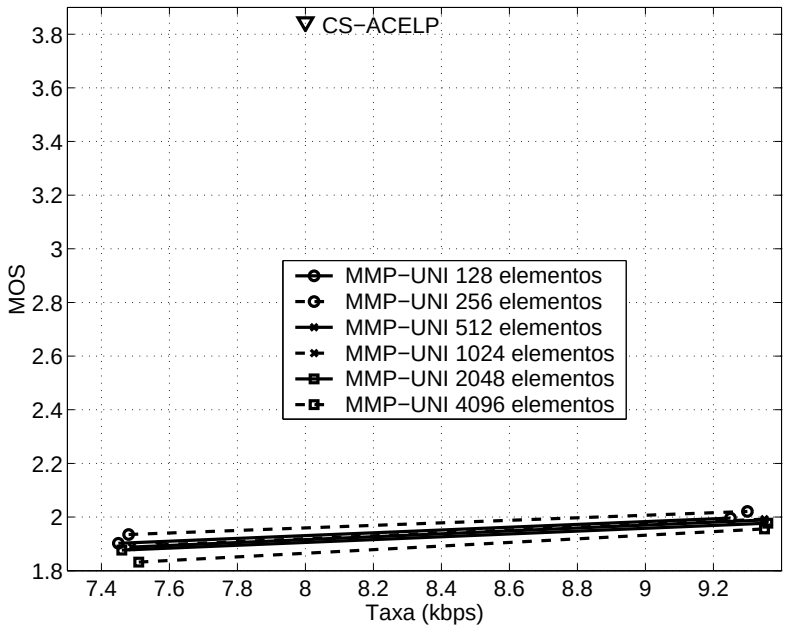


Figura 6.3: Resultados PESQ-MOS para o algoritmo MMP-UNI, com taxa de codificação em torno de 8 kbps, com dicionário uniforme com diferentes tamanhos T em comparação ao resultado G.729.

O desempenho total do algoritmo MMP-UNI para diferentes taxas de codificação do banco de dados DB1 é retratado na Figura 6.5. Para efeito de comparação, são mostrados resultados para os padrões ITU-T G.711 (PCM) [37], ITU-T G.726 (ADPCM) [38] e ITU-T G.729 (CS-ACELP) [48].

Pelos resultados acima, podemos concluir que a versão do MMP com dicionário inicial uniforme apresenta um desempenho taxa $\times$ qualidade muito ruim, atingindo um valor PESQ-MOS de 1,96 para uma taxa em torno de 8 kbps. Isto demonstra que o MMP-UNI não consegue aprender a tempo as características do sinal de voz, mesmo possuindo um dicionário com muitos padrões.

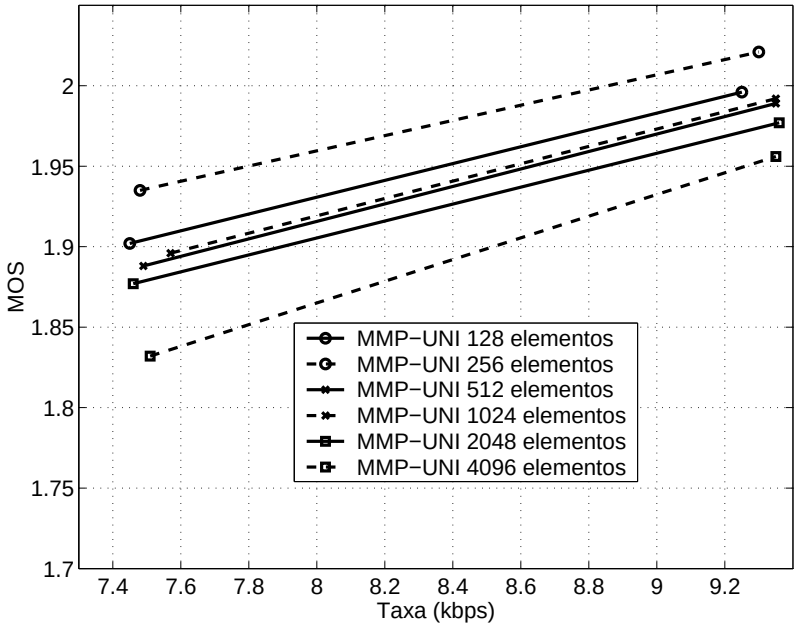


Figura 6.4: Resultados PESQ-MOS para o algoritmo MMP-UNI ampliado na região em torno de 8 kbps de taxa de codificação.

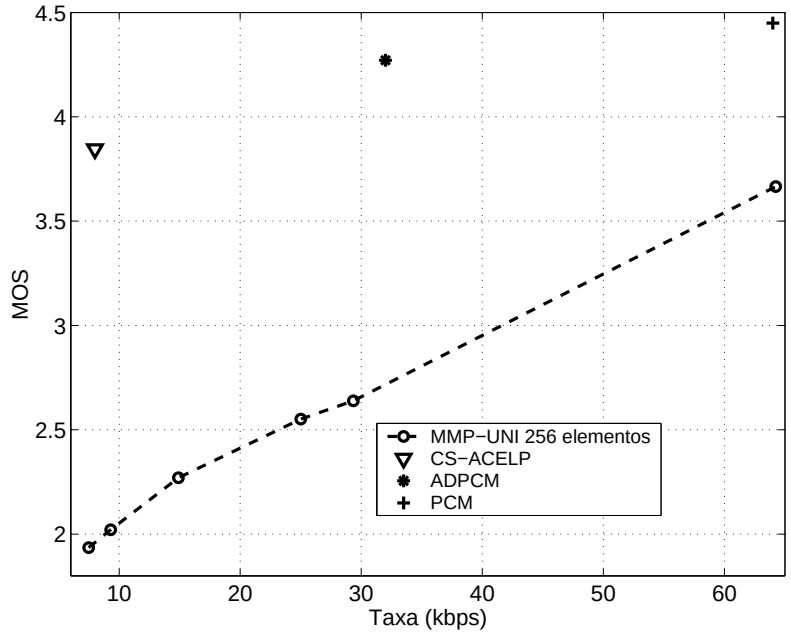


Figura 6.5: PESQ-MOS × taxa de codificação para o algoritmo MMP-UNI.

Nota-se pela Figura 6.6 que ao usarmos um dicionário uniforme as várias escalas do dicionário acabam sendo populadas por muitos elementos redundantes, que prejudicam a eficiência de codificação do MMP-UNI. De fato, um dicionário com muitos vetores implica um alto custo em termos de taxa de bit para as escalas menores. Por outro lado, para escalas maiores um dicionário bastante populoso pode aumentar as chances de sucesso do casamento de padrões. A redução dos elementos

redundantes permite uma codificação mais eficiente para as escalas menores e pode melhorar também o processo de codificação das escalas maiores. Espera-se que a utilização do dicionário não-uniforme contribua para a redução da quantidade de elementos redundantes no dicionário.

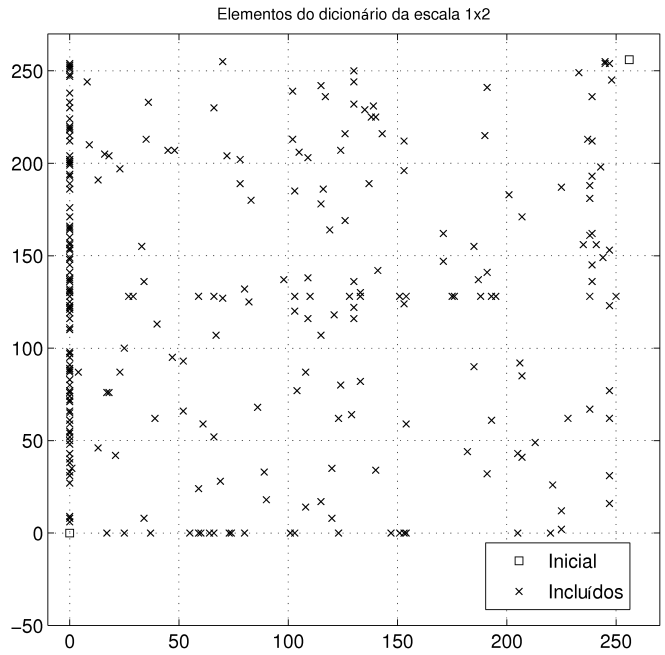
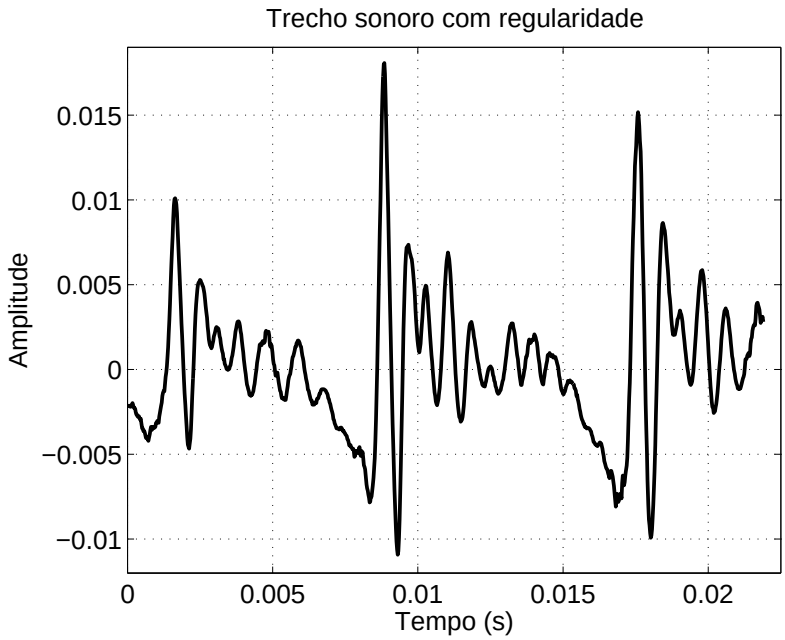


Figura 6.6: Gráfico da escala 1×2 do dicionário original do MMP-UD para taxa de 8 kbps. Cada ponto representa um elemento bi-dimensional do dicionário

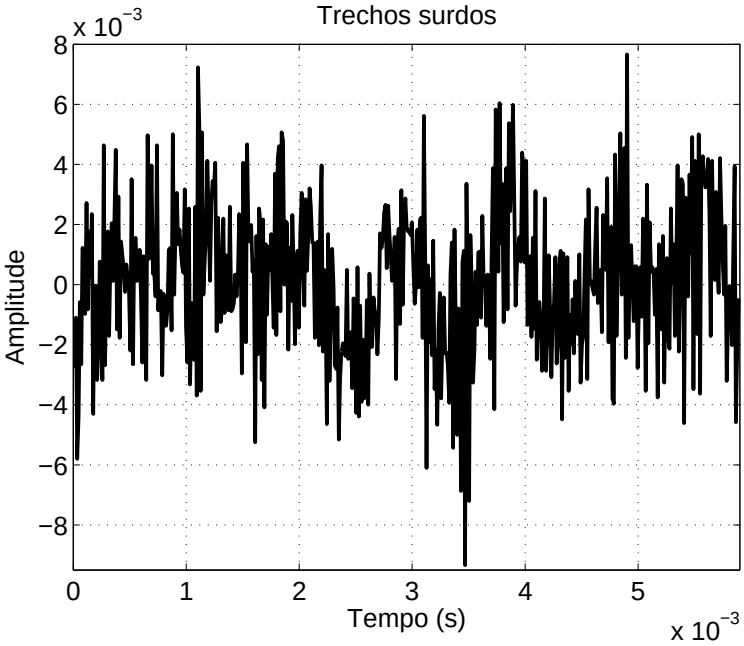
6.2 MMP com dicionários de deslocamento - MMP-UD

O sinal de voz costuma apresentar uma forte regularidade temporal, particularmente em trechos sonoros, como ilustrado na Figura 6.7a, ou de silêncio [33]. Já para trechos surdos (típicos de sons fricativos), a regularidade se dá num domínio estatístico e não temporal, como ilustrado na Figura 6.7b. Desta forma, vê-se que o MMP na sua forma básica tem potencial para gerar bons resultados para trechos sonoros, apesar de não ser particularmente adequado para trechos surdos ou mistos (que incluem mais de uma característica temporal).

Nos segmentos sonoros, surge um comportamento quase periódico, que pode ser facilmente codificado com o algoritmo MMP com auxílio de dicionário de deslocamento [4, 5] que contém apenas as amostras do sinal de voz recentemente codificadas (vide Seção 5.7 e Figura 6.8).



(a)



(b)

Figura 6.7: Exemplos de trecho de voz: (a) sonoro; (b) surdo.

Neste dicionário auxiliar, o MMP busca um segmento de tamanho N deslocado de δ amostras a partir do trecho do sinal de voz recentemente codificado (vide Figura 6.8) a fim de encontrar a melhor aproximação da parte do sinal de entrada. O tamanho N do segmento sendo codificado pode ter o comprimento de $1, 2, 4, 8, 16, 32, 64, 128$ amostras, dependendo de como o dicionário original foi inicializado e o deslocamento δ pode ter uma passo de variação no intervalo de $1, 2, 4, \ldots$ ou mesmo $1/2, 1/4, 1/8, \ldots$ amostras. A melhor aproximação encontrada no dicionário

de deslocamento é comparada com o resultado obtido com o dicionário original, e um *flag* '1' é usado para identificar o dicionário de origem, junto com índice de deslocamento δ, indicando a melhor posição deste segmento do dicionário de deslocamento. Caso contrário, o processo de codificação é baseado no uso do segmento do dicionário original, e um *flag* '0' é enviado junto com o índice do dicionário original. O processo de busca, com o dicionário de deslocamento, é encerrado quando o índice de deslocamento atinge um janelamento de comprimento L ($\delta = L$). O tamanho da janela é definido no início do processamento do algoritmo MMP e seu comprimento pode ser: $\ldots, 128, 256, 512, 1024, \ldots$. Com esta estratégia, o procedimento de busca é bastante simplificado, uma vez que um conjunto limitado de segmentos é comparado com o atual sinal de voz, e a taxa de codificação é reduzida em conformidade. Tipicamente, um dicionário com deslocamento com $L = 512$ e passo de variação δ igual a 1 leva a 512/1 índices e um dicionário original pode chegar a aproximadamente 17000 elementos a uma taxa de 8 kbps.

Nesta seção consideramos o algoritmo MMP-UNI incorporando o uso do dicionário de deslocamento (MMP-UD) quando na codificação do mesmo banco de dados DB1 usado na Seção 6.1 (com cada escala do dicionário inicial compostas de 256 elementos). O resultado deste processo para diferentes valores do comprimento L do janelamento, e passo de variação de δ igual a 1 amostra, é dado na Figura 6.10, onde se observa uma melhora substancial provocada pelo dicionário auxiliar, em especial para $L = 512$, atingindo um resultado PESQ-MOS de 2,71.

Observando a Figura 6.10, percebe-se que o uso do dicionário de deslocamento diminui ligeiramente a quantidade de elementos no dicionário original quando comparada com a quantidade requerida pelo MMP-UNI (linha sólida). Esta diferença não só tende a melhorar o desempenho do algoritmo em termos de taxa $\times$ qualidade, mas também reduz a complexidade computacional do algoritmo, pois quanto menor o número de elementos no dicionário original, mais rápida é a busca por padrões para o casamento.

A Figura 6.11 mostra que, neste experimento, 56% do sinal de voz codificado foi constituído de elementos do dicionário de deslocamento e 44% por elementos do dicionário original. Como o codificador aritmético necessita somente da informação de 2 (dois) *flags* (um indicando o dicionário de deslocamento e um outro o índice de deslocamento) e a faixa de variação do índice de deslocamento apresenta um valor fixo (512 neste caso), a chance de usarmos determinado índice de deslocamento tende a ser maior que a de utilizar o índice do dicionário original, o qual tende a ter uma faixa de variação muito maior, consumindo mais bits. Além disso, a frequência maior do uso do dicionário de deslocamento simplifica a modelagem realizada pelo codificador aritmético.

A Figura 6.12 mostra o desempenho do MMP-UD para diferentes taxas de codi-

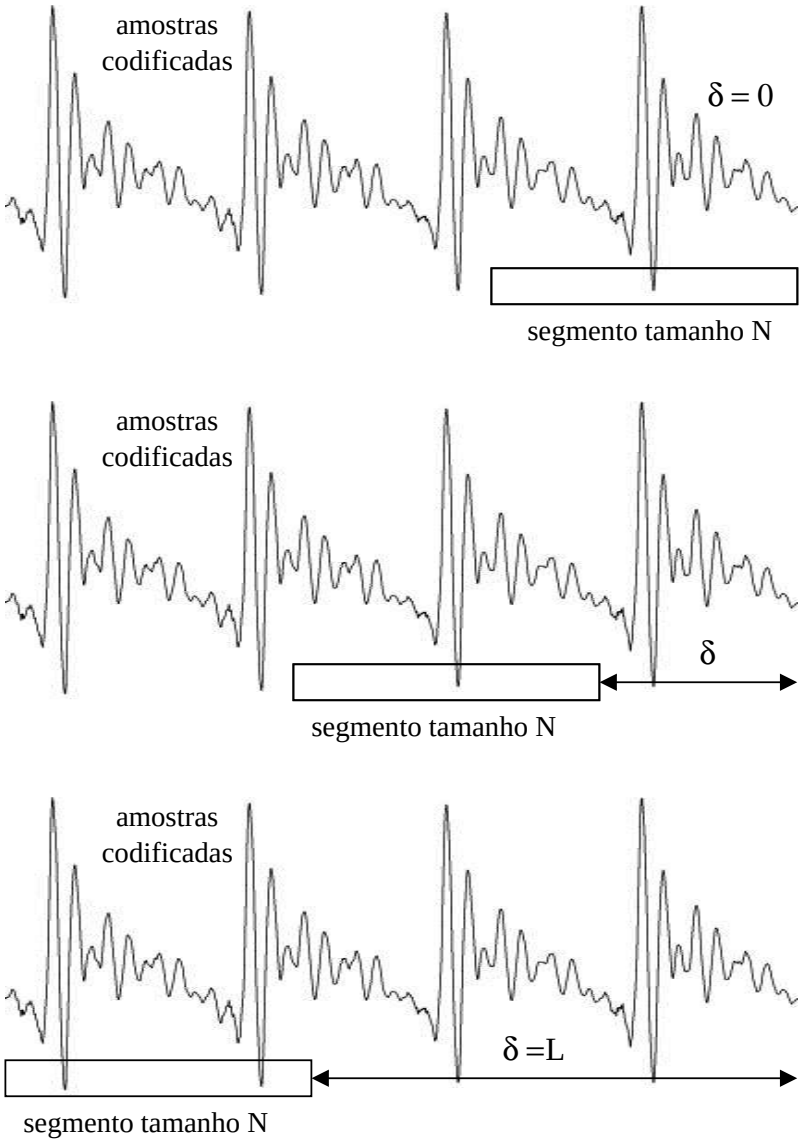


Figura 6.8: Procedimento do dicionário de deslocamento.

ficação em comparação com os codecs G.711, G.726 e G.729. Nota-se que para taxa de 64 kbps o MMP-UD já atinge um desempenho ligeiramente melhor (PESQ-MOS de 4,47) que o G.711 (PCM). Para a taxa de 32 kbps, o MMP-UD melhorou muito se comparado com os resultados do MMP-UNI (Figura 6.5), chegando próximo do desempenho do G.726 (ADPCM), que proporciona um valor PESQ-MOS de 4,13.

6.3 MMP com dicionário não-uniforme - MMP-MU

É sabido que, no domínio do tempo, as distorções do sinal de entrada ocorridas em baixas amplitudes são mais perceptivas que as geradas em altas amplitudes. Isto motivou o uso da quantização não-uniforme do sinal, como a definida pela lei-μ

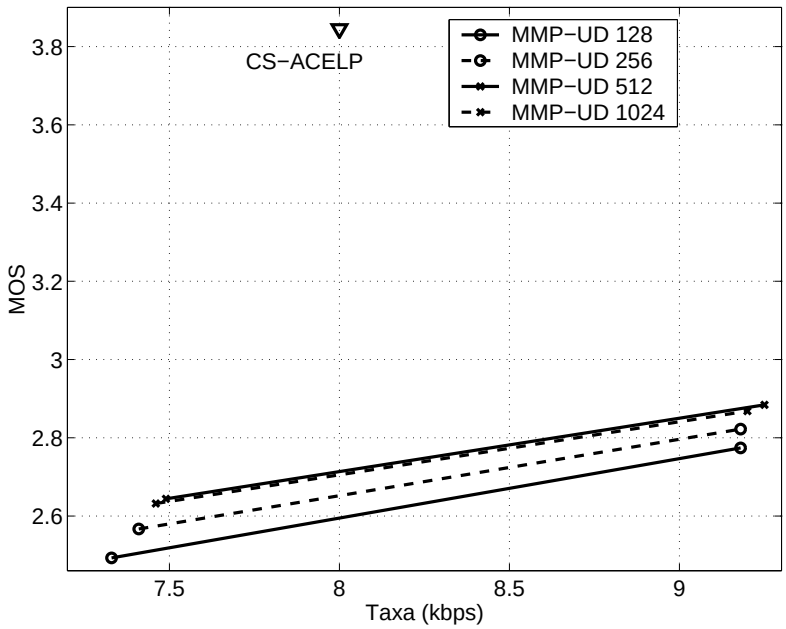


Figura 6.9: Resultados PESQ-MOS para o algoritmo MMP-UD com dicionário de deslocamento com taxa de codificação em torno de 8 kbps.

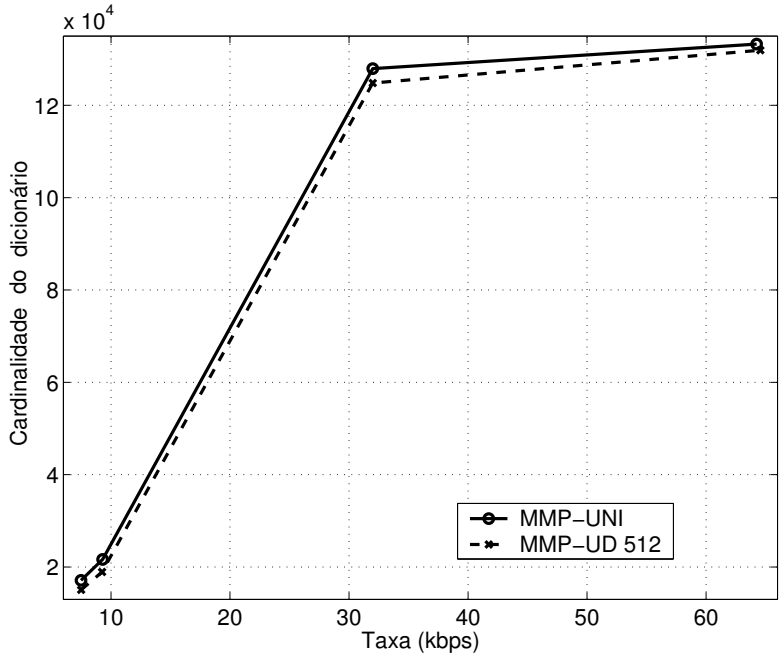


Figura 6.10: Número final de elementos do dicionário do algoritmo MMP-UD para diversas taxa de codificação.

utilizada no codificador para telefonia [37]. Este procedimento pode ser facilmente incorporado ao codificador MMP, compondo o dicionário com vetores distribuídos de acordo com esta lei (MMP-MU). Esta operação será aplicada ao dicionário inicial

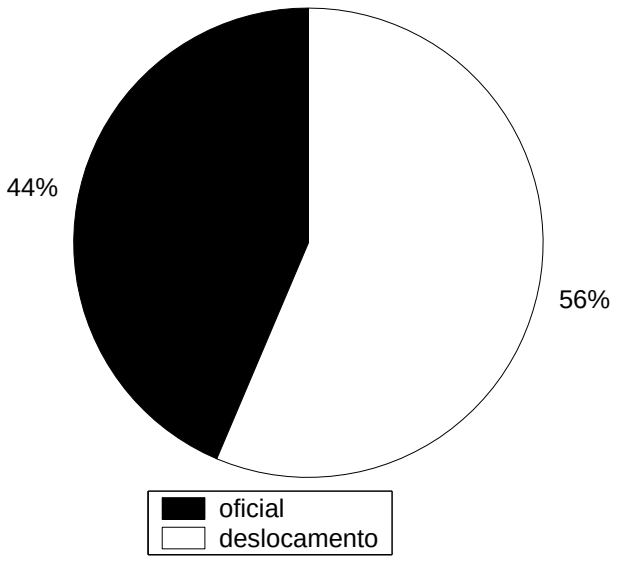


Figura 6.11: Uso do dicionário de deslocamento para o algoritmo MMP-UD.

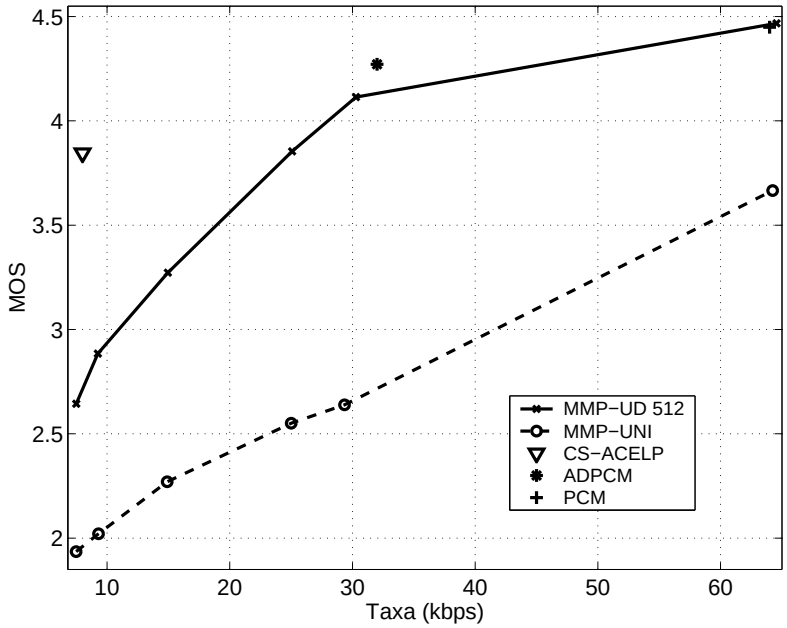


Figura 6.12: Resultados PESQ-MOS para o algoritmo MMP-UD em comparação com os demais codificadores de voz.

bem como no seu processo de atualização registrando os valores de suas amostras a

$$x = A\frac{2^{\frac{n}{16}} - 1}{255}. \tag{6.1}$$

onde A é a amplitude máxima considerada, $n = -128, -127, \ldots, 127, 128$ e x representa a saída, que no nosso caso é a amplitude do vetor do dicionário. O uso da lei-μ na quantização do dicionário MMP proporciona uma característica perceptual para o processo de aprendizagem do MMP-MU. Como efeito colateral positivo, este procedimento também limita o crescimento do dicionário e simplifica as sucessivas

buscas de elementos no dicionário, que nos leva à redução da complexidade computacional do processo de codificação, além de uma redução da taxa de codificação das palavras do dicionário, devido à redução da sua cardinalidade.

A Figura 6.13 mostra os resultados PESQ-MOS quando usamos o algoritmo MMP-MU sem o dicionário auxiliar de deslocamento introduzido na seção anterior. Desta figura, observa-se que o dicionário não-uniforme melhora a relação taxa×qualidade para um valor em torno de 2,75 na taxa de 8 kbps.

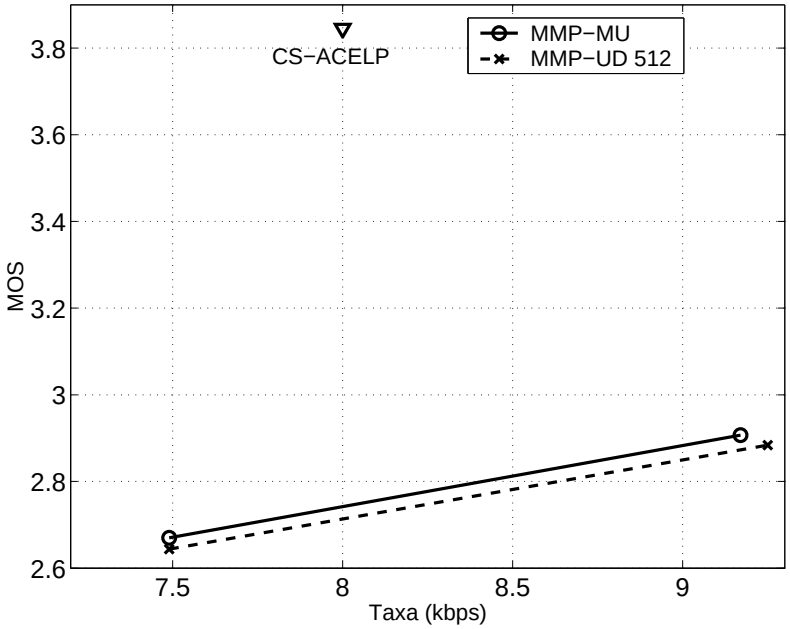


Figura 6.13: Resultados PESQ-MOS para o algoritmo MMP-MU, com taxa de codificação em torno de 8 kbps, com dicionários uniforme (linha tracejada) e não-uniforme (linha sólida) em comparação ao resultado G.729.

Nesta mesma taxa, a complexidade computacional é reduzida, já que o número final de elementos do dicionário principal é reduzido em cerca de 30% (de 18411 para 12520 elementos), como observado na Figura 6.14.

A Figura 6.15 apresenta o desempenho do MMP-MU para diferentes taxas, em comparação aos resultados dos padrões G.729, G.726 e G.711. Observa-se que o algoritmo MMP-UD aproxima-se ainda mais do resultado G.726, atingindo um valor PESQ-MOS de 4,20.

6.4 MMP-MU com dicionários de deslocamento - MMP-MUD

Nesta seção, o dicionário de deslocamento contendo L amostras previamente codificadas, com passo de variação de deslocamento de $\delta = 1$, é incorporado no algoritmo MMP-MU (MMP-MUD). Este dicionário adicional funciona como uma memória a

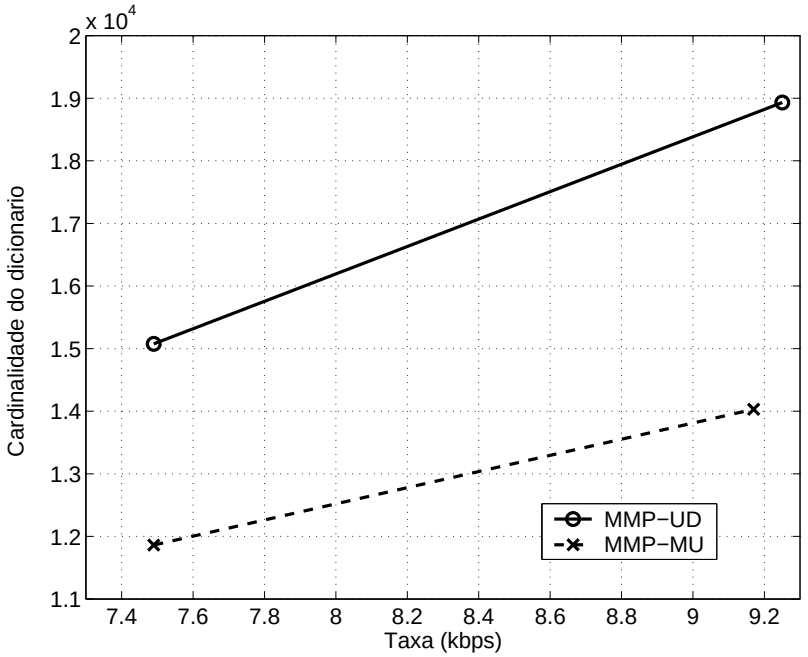


Figura 6.14: Número final de elementos do dicionário do algoritmo MMP-MU para taxa de codificação de 8 kbps.

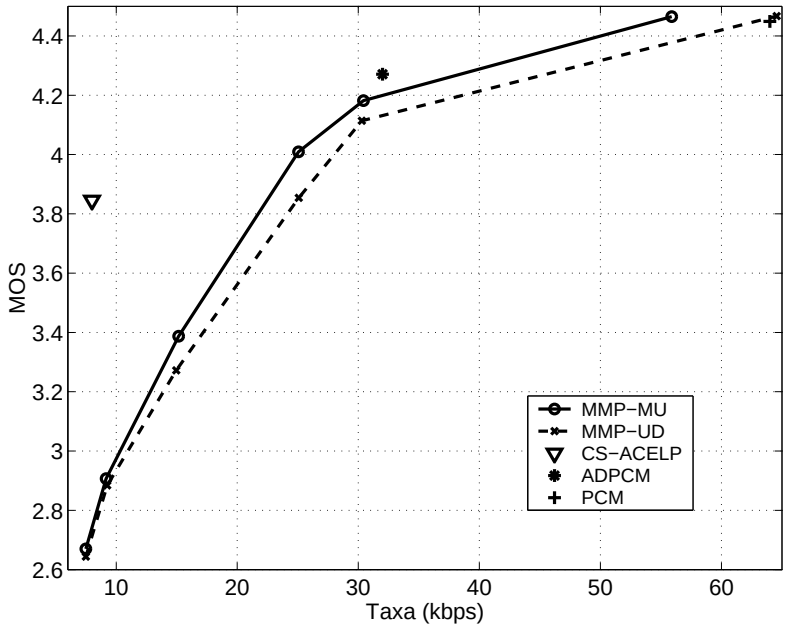


Figura 6.15: Resultados PESQ-MOS para o algoritmo MMP-MU em comparação aos resultados G.729, G.726 e G.711.

curto tempo que serve bem para o procedimento de casamento de padrões para sinais quase periódicos, tais como segmentos sonoros. Os resultados para o algoritmo resultante são vistos na Figura 6.16, que mostra que o comprimento $L = 512$ apresentou um PESQ-MOS de 2,88 na taxa de 8 kbps, uma melhoria em relação aos resultados sem o dicionário de deslocamento da seção anterior.

O uso do dicionário de deslocamento no procedimento de codificação do MMP-

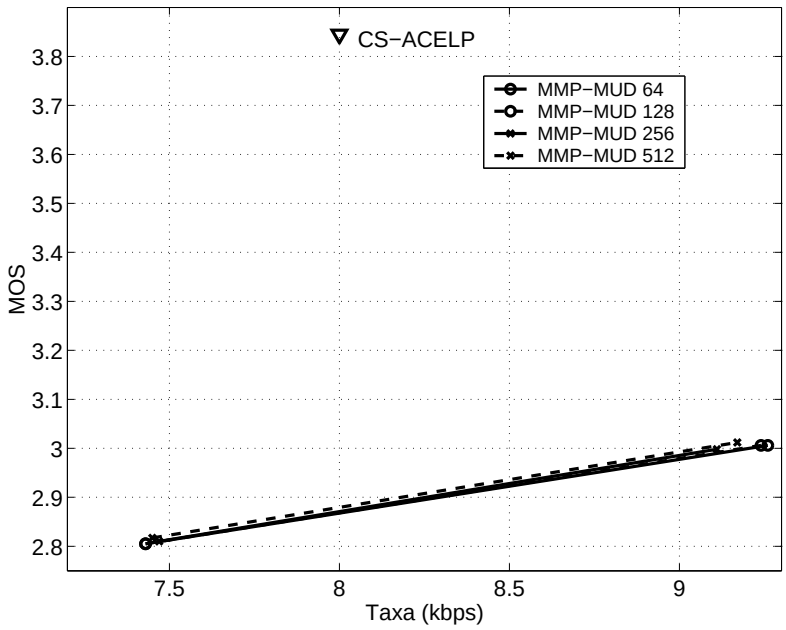


Figura 6.16: Resultados PESQ-MOS para o algoritmo MMP-MUD, com taxa de codificação em torno de 8 kbps, para diferentes comprimentos do dicionário de deslocamento, em comparação aos resultados para o G.729.

MUD é identificado pelas partes claras no gráfico inferior da Figura 6.17. Desta forma, um sinal altamente regular resulta em segmentos grandes que vão provavelmente ser casados com o dicionário de deslocamentos, portanto tendendo a reduzir a taxa de codificação resultante.

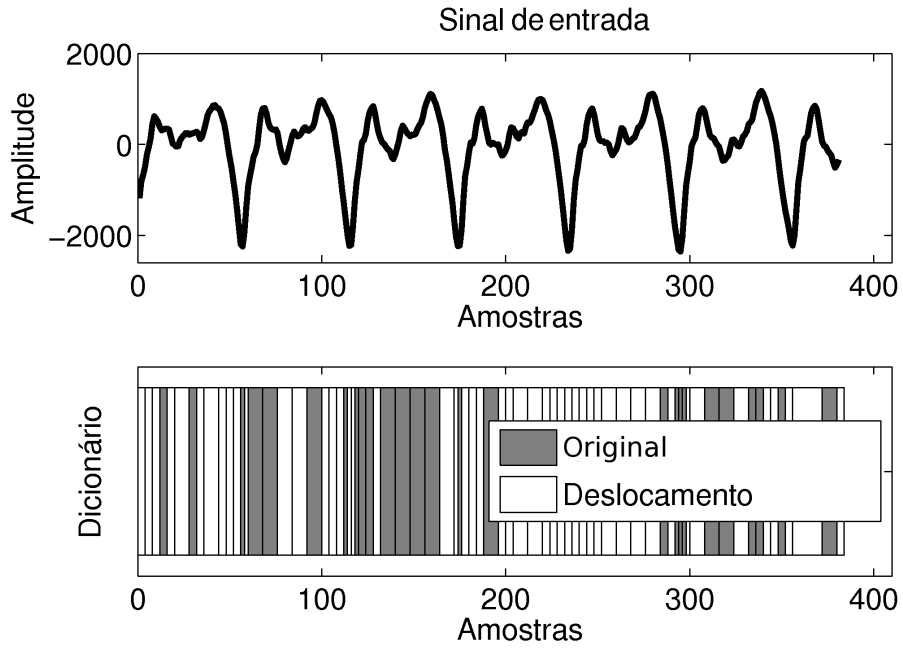


Figura 6.17: Uso do dicionário de deslocamento ao longo do tempo no codificador MMP-MUD para um dado sinal de voz.

Para o banco de dados DB1, o dicionário multiescala original foi usado em 70% de todos os segmentos, enquanto o dicionário de deslocamento foi usado em 30%, reduzindo a taxa de codificação. A Figura 6.18 apresenta a distribuição estatística dos comprimentos dos segmentos codificados para o banco de dados DB1. Desta figura, conclui-se que todos os níveis de segmentação são usados com frequência equivalente, deste modo, reforçando a utilidade da natureza multiescalar dos dicionários na codificação da voz.

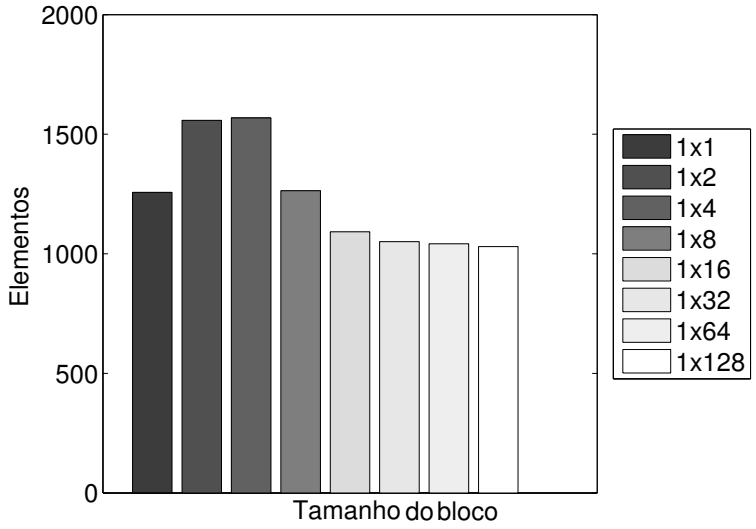


Figura 6.18: Estatística do tamanho dos segmentos no codificador MMP-MUD.

O comportamento do algoritmo MMP-MUD para diferentes taxas de codificação do banco de dados DB1 é retratado na Figura 6.19. Para efeito de comparação, os resultados também são mostrados para os padrões ITU-T G.711 (PCM) [37], ITU-T G.726 (ADPCM) [38] e ITU-T G.729 (CS-ACELP) [48].

Outra observação acerca do algoritmo MMP-MUD é quanto à redundância de elementos no dicionário. Neste sentido, a Figura 6.20 mostra que o uso dos dicionários não-uniforme e de deslocamento reduz a quantidade de elementos redundantes no dicionário original (na escala 1×2, por exemplo, o número de elementos caiu de 2049 para 1475). Desta forma, o processo de busca é reduzido e a codificação do índice do dicionário torna-se mais eficiente.

Entretanto, nota-se na Figura 6.19 que o desempenho de MMP-MUD na taxa de 8 kbps ainda é bastante inferior ao do desempenho do CS-ACELP. Como, quando o MMP é usado em compressão de imagens, é bastante vantajoso usar predição (o resíduo de uma predição é codificado com o MMP), atingindo-se resultados compatíveis com o estado-da-arte [10], no próximo capítulo se investiga o uso do MMP em conjunto com técnicas de predição.

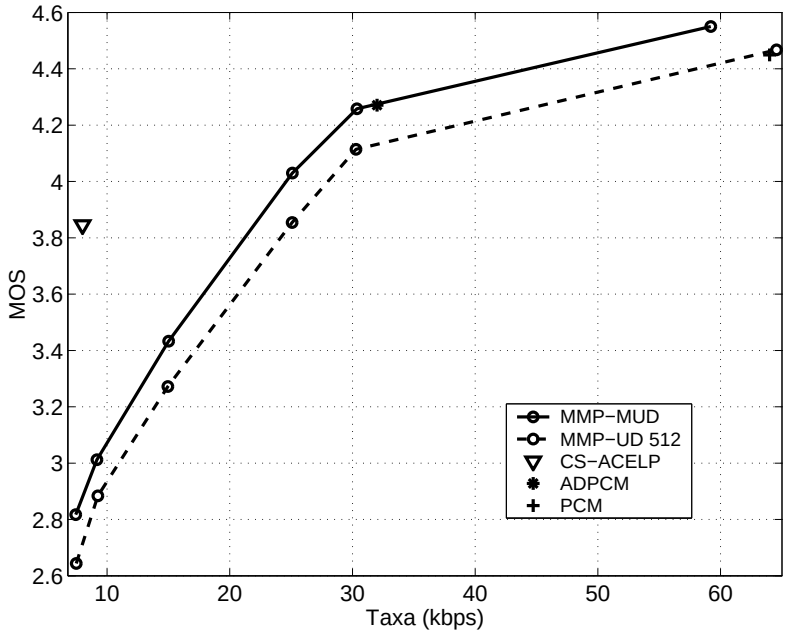


Figura 6.19: PESQ-MOS × taxa de codificação para o algoritmo MMP-MUD (linha sólida) comparando com o MMP-UD (linha tracejada).

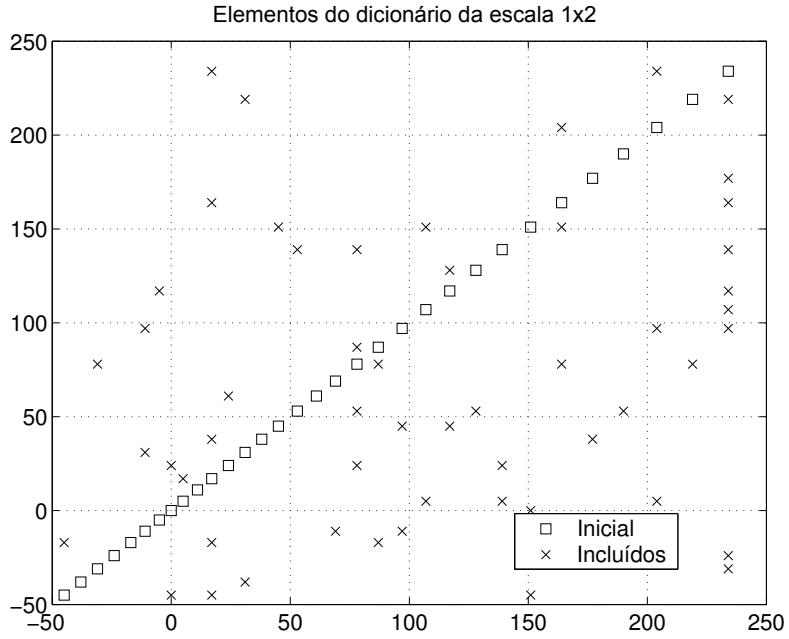


Figura 6.20: Gráfico da escala 1×2 do dicionário original do MMP-MUD para taxa de 8 kbps. Cada ponto corresponde às coordenadas de um elemento do dicionário.

Capítulo 7

Codificação de sinais de voz usando MMP baseado em Predição - MMP-P

Atualmente os codificadores de voz que atingem o melhor compromisso entre qualidade de voz e taxa de compressão são baseados em predição linear com excitação por código (*code-excited prediction linear*, CELP) [40]. Posteriormente à proposta CELP original, foram sugeridas diversas modificações visando a aprimorar a sua qualidade [43] ou diminuir sua complexidade [44, 65], uma vez que o sistema era de difícil implementação em tempo real. Esses codificadores utilizam o procedimento de análise por síntese (*analysis-by-synthesis*, AbS) para determinar o sinal de entrada para um modelo de predição linear do trato vocal humano. Pensando justamente neste modelo, resolvemos investigar o desempenho do algoritmo MMP quando operando sobre o sinal de resíduo gerado pela predição linear do sinal de voz em análise. A predição descorrelaciona os blocos e gera um sinal de resíduo com distribuição altamente concentrada em torno de zero. Além disto, segmentos do sinal de predição tendem a ter norma constante, o que pode ser explorado para facilitar o processo de aprendizado do algoritmo MMP.

Para analisar o desempenho do algoritmo MMP na codificação de voz usando o erro de predição, este capítulo apresentará a técnica de predição linear, bem como o MMP baseado neste conceito. Serão incorporados ao esquema de predição linear as técnicas adicionais consideradas no Capítulo 6 deste trabalho, a saber: dicionário não-uniforme, dicionário de deslocamento (vide Seção 6.4) e a atualização do dicionário usando segmentos normalizados, todos estes incorporados à codificação do erro de predição.

7.1 Predição Linear

Um exemplo de codificador de voz que usa a predição é o DPCM (*Differential Pulse Code Modulation*). Este codificador quantiza a diferença entre o sinal de voz e uma estimativa sua. Se a estimativa é eficaz, o erro entre as amostras do sinal de voz e as previstas terá uma faixa dinâmica reduzida, resultando em menos bits para a codificação. Um outro exemplo de codificador de voz que usa a predição é o ADPCM (*Adaptive Differential Pulse Code Modulation*) que usa o DPCM com técnicas de predição adaptativa. No ADPCM, o preditor é ajustado de acordo com as variações estatísticas do sinal de voz. O uso da predição adaptativa na codificação de um sinal de voz é muito apropriado, pois torna os codificadores capazes de se adaptarem ao sinal sendo processado. Entre os codificadores que usam a predição, merecem destaque também os que se baseiam na técnica CELP.

A predição linear (LP, do inglês *linear prediction*) é um procedimento que estima a(s) próxima(s) amostra(s) de um sinal a partir da combinação linear das amostras atual e passadas, isto é,

$$\hat{s}(n) = \sum_{i=1}^{N} a_i s(n-i), \tag{7.1}$$

onde N é a ordem do preditor e a_i, para $i = 1, 2, ..., N$ são chamados de coeficientes LP. Se o modelo de predição estiver bem dimensionado, o erro entre o sinal original e a predição tenderá a ter uma distribuição com uma pequena variância em torno do valor zero, o que tende a tornar o codificador por entropia bastante eficiente. Usando a estimativa $\hat{s}(n)$, podemos determinar o erro de predição entre os valores verdadeiros e os estimados como

$$e(n) = s(n) - \hat{s}(n). \tag{7.2}$$

Aplicando a transformada $\mathcal{Z}$, as Equações (7.1) e (7.2) correspondem a

$$\mathcal{Z}\{s(n)\} = H(z)\mathcal{Z}\{e(n)\}, \tag{7.3}$$

com

$$H(z) = \frac{1}{1 - a_1 z^{-1} - a_2 z^{-2} - \cdots - a_N z^{-n}}. \tag{7.4}$$

Esta equação indica que o procedimento LP modela o processo $\{s\}$ como saída de um sistema com função de transferência $H(z)$ para a estimação do erro $\{e\}$. Se o processo é estacionário, podemos mostrar que os coeficientes $a_i^\star$, que minimizam o erro médio quadrático $E[e^2(n)]$, são dados pela solução do sistema de equações

lineares [33]:

$$\begin{bmatrix} R_s(0) & R_s(1) & \cdots R_s(N-1) \\ R_s(1) & R_s(0) & \cdots R_s(N-2) \\ \vdots & \vdots & \ddots \quad \vdots \\ R_s(N-1) R_s(N-2) \ldots & & R_s(0) \end{bmatrix} \begin{bmatrix} a_1^\star \\ a_2^\star \\ \vdots \\ a_N^\star \end{bmatrix} = \begin{bmatrix} R_s(1) \\ R_s(2) \\ \vdots \\ R_s(N) \end{bmatrix}, \tag{7.5}$$

onde $R_s(\nu) = E[s(n)s(n-\nu)]$, para $\nu = 0, 1, \ldots, N$. Quando uma quantidade limitada de dados está disponível e a complexidade computacional é um fator importante, há vários algoritmos para estimar $R_s(\nu)$ e resolver a Equação (7.5) [33].

7.2 MMP com predição linear - MMP-P

Na prática, um sinal de voz pode ser considerado estacionário em intervalos de 10–30 ms [33], o que corresponde a um conjunto de 80–240 amostras numa taxa de amostragem de 8 kHz. No algoritmo MMP baseado em predição (MMP-P), o sinal de voz é segmentado, a princípio, em intervalos de 128 amostras, e cada segmento do sinal codificado será usado para estimar as próximas 128 amostras usando a predição linear, e assim, encontrar o erro de predição (Equação (7.2)) ou sinal de resíduo.

Suponha que um intervalo precedente $[s_{k-1}(n)]_{\mathrm{Q}}$ já foi codificado. Após uma estimativa adequada da função de auto-correlação $R_s(\nu)$, o modelo LP $H_{k-1}(z)$ correspondente, dado na Equação (7.4), é determinado através da Equação (7.5). Usando uma abordagem simples mas um tanto eficaz, o segmento seguinte de 128 amostras estimado $[\hat{s}_k(n)]$ pode ser obtido pela Equação (7.1) usando os coeficientes LP de $H_{k-1}(z)$. Assim da Equação (7.2) podemos calcular as 128 amostras correspondentes do erro residual $[e_k(n)]$, que são posteriormente codificadas pelo algoritmo MMP-P, como exemplificado abaixo, produzindo $[e_k(n)]_{\mathrm{Q}}$. O resultado para o intervalo atual de voz é então dado por

$$[s_k(n)]_{\mathrm{Q}} = [\hat{s}_k(n)] + [e_k(n)]_{\mathrm{Q}}, \tag{7.6}$$

que, seguindo o mesmo procedimento, permite codificar o próximo intervalo de voz e assim por diante (vide Figura 7.1). Ao utilizar este procedimento, o MMP-P se beneficia do bom comportamento das características estatísticas do sinal de erro quando comparado com o sinal de voz original [66]. Nota-se que com este procedimento não há necessidade de transmitir os coeficientes de predição para o decodificador, uma vez que eles podem ser deduzidos a partir do bloco previamente codificado, que já é conhecido pelo decodificador.

O algoritmo MMP-P representa segmentos do sinal a ser codificado, neste caso o erro de predição, usando um procedimento de casamento de segmentos realizado

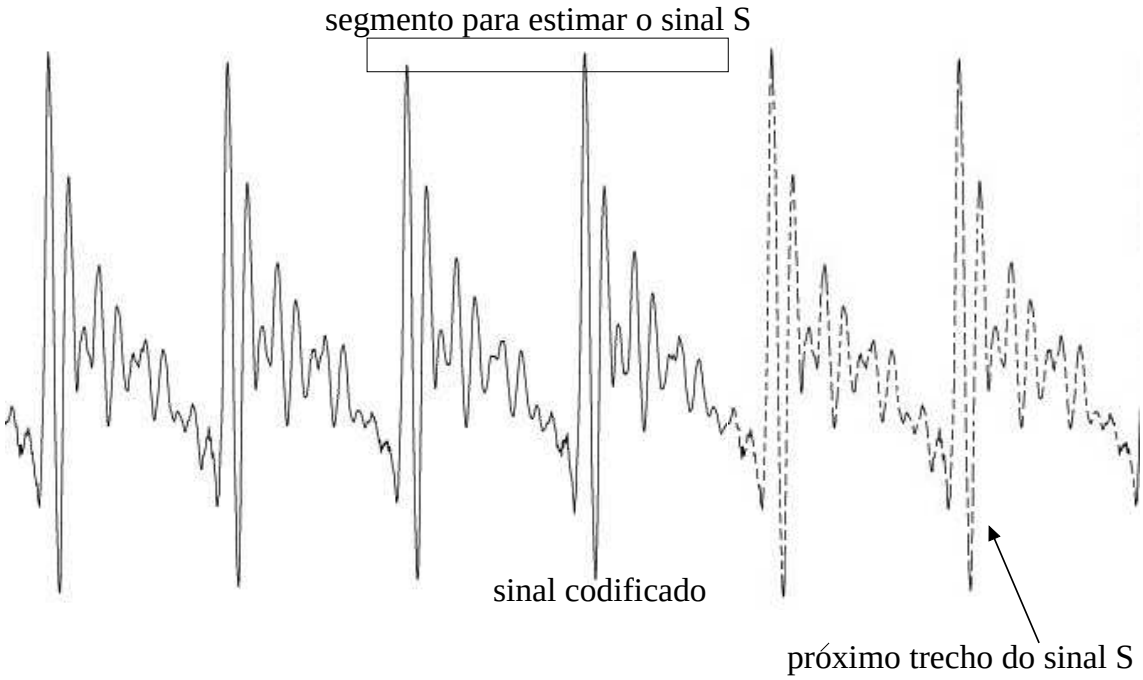


Figura 7.1: Esquema do processo de predição linear no algoritmo MMP-P.

em diferentes escalas e baseado em segmentos previamente codificados. Assim, o algoritmo MMP-P é capaz de aprender os padrões presentes no sinal de entrada (no caso, o sinal de resíduo). Ele consegue isso usando o dicionário multiescalas do MMP, que será composto de padrões de resíduos. A operação inteira do MMP-P pode ser dividida em quatro etapas discutidas a seguir.

7.2.1 Inicialização do dicionário

O dicionário inicial determina a habilidade do algoritmo de casar padrões como sinal de entrada não somente durante o estágio inicial de codificação, mas também durante todo o processo de codificação. Este dicionário é caracterizado pelo seu tamanho L, em cada escala, e pela distribuição uniforme ou não-uniforme de seus elementos. Um dicionário eficiente deve ser suficientemente grande para incluir padrões interessantes para o procedimento de casamento, aumentando assim a qualidade do sinal codificado, e pequeno o suficiente para evitar padrões desnecessários, reduzindo o tamanho do *bitstream* resultante.

Nesta tese, o dicionário inicial do MMP consiste de vetores com 8 diferentes escalas: 1×1, 1×2, 1×4, 1×8, 1×16, 1×32, 1×64 e 1×128. Neste capítulo cada escala do dicionário inicial será composta por 256 vetores.

7.2.2 Predição linear

Neste estágio o MMP-P estima o próximo trecho do sinal de voz com intervalo de 128 amostras que será usado para encontrar o erro de predição (sinal de resíduo), conforme mencionado anteriormente, para em seguida iniciar o processo de codificação.

7.2.3 Casamento de padrões

No algoritmo MMP-P, da mesma forma como em suas versões anteriores, cada bloco do sinal de entrada é segmentado de acordo com uma árvore, descrita por uma sequência de '0's e '1's, associados ou não à partição, respectivamente, de um segmento. Cada bit '1', que denota que o segmento não é particionado, é associado ao índice i_k no dicionário do vetor que melhor casa com o segmento sendo codificado. O código MMP para o sinal inteiro é obtido codificando o *stream* de símbolos gerados (incluindo a descrição da árvore de segmentação e os índices do dicionário para todo o bloco de 128 amostras) usando um codificador aritmético adaptativo [55].

7.2.4 Atualização do dicionário

Após a codificação, as escalas do dicionário são atualizadas. Os segmentos que formam o bloco inteiro de 128 amostras são concatenados e incluídos em cada escala do dicionário. Se o tamanho do segmento é maior ou menor que a dimensão da escala correspondente no dicionário, ele será devidamente dimensionado (expandido ou contraído) seguindo um padrão de mudança de escala. Neste processo, verifica-se se há no dicionário alguma palavra similar à que está sendo introduzida; se este é o caso, o dicionário não é atualizado [10]. Nesta verificação, o segmento é quantizado, como é feito na inicialização do dicionário, e normalizado, segundo um procedimento que será visto na Seção 7.6. Desse modo, reduzimos o número de elementos redundantes no dicionário, e tornamos o processo de codificação mais eficiente em termos de taxa e complexidade computacional.

7.3 Ordem do modelo de predição linear

Nesta seção, investigamos o desempenho de taxa×qualidade para o algoritmo MMP-P, considerando o mesmo banco de dados DB1 utilizado no capítulo anterior. Neste experimento, o dicionário inicial do MMP-P foi montado utilizando a quantização não-uniforme ditada pela lei-μ e procuramos verificar o desempenho do MMP-P para diversos valores de ordem do modelo LP. A ideia é encontrar a ordem do modelo de

predição que resulta no melhor compromisso taxa×qualidade do algoritmo MMP-P, e assim usá-la nas análises subsequentes.

Dos resultados mostrados na Figura 7.2, o MMP-P com a ordem do preditor sendo $N = 20$ apresentou o melhor desempenho, correspondendo a um valor PESQ-MOS de 2,93, para taxas em torno de 8 kbps.

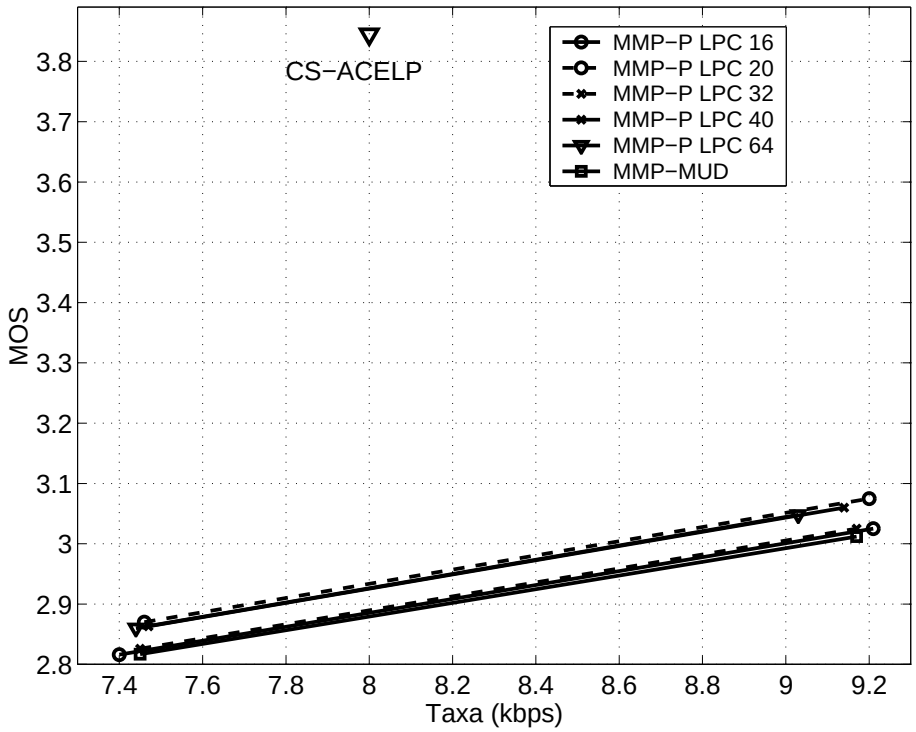


Figura 7.2: Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-P com diferentes ordens do modelo LP.

Para uma melhor visualização, a Figura 7.2 foi ampliada em torno da taxa de 8 kpbs, e o resultado é visto na Figura 7.3.

A Figura 7.4 mostra o comportamento geral do MMP-P em comparação com os outros codificadores de voz. Desta figura, nota-se que o MMP-P também obtém um desempenho melhor que o codificador G.726, alcançando um resultado PESQ-MOS igual a 4,31, para a taxa de 32 kbits/s.

7.4 MMP-P com dicionário de deslocamento MMP-PD

Nesta seção, um dicionário de deslocamento contendo L amostras previamente codificadas é incorporado ao algoritmo MMP-P resultando na versão MMP-PD. A ideia do dicionário de deslocamento, que funciona como uma memória de curto tempo, foi primeiramente vista na Seção 6.4. A Figura 7.5 mostra o desempenho do MMP-PD para deslocamentos múltiplos de 1 amostra e diferentes comprimentos L, onde se

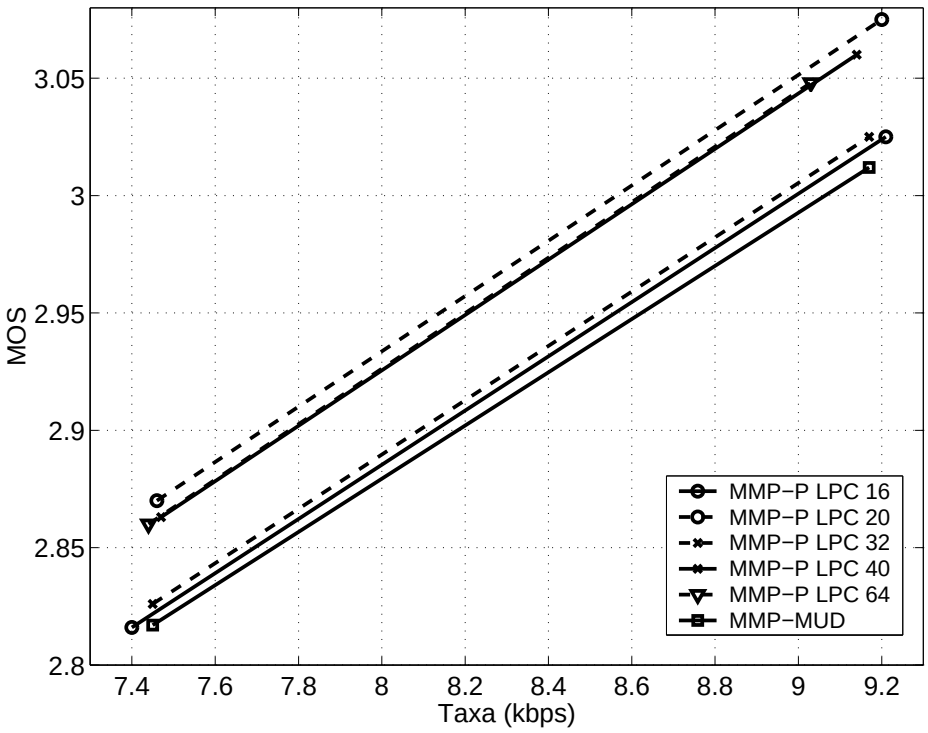


Figura 7.3: Resultados PESQ-MOS para o algoritmo MMP-P ampliado na região em torno de 8 kbps para diferentes ordens N do modelo LP.

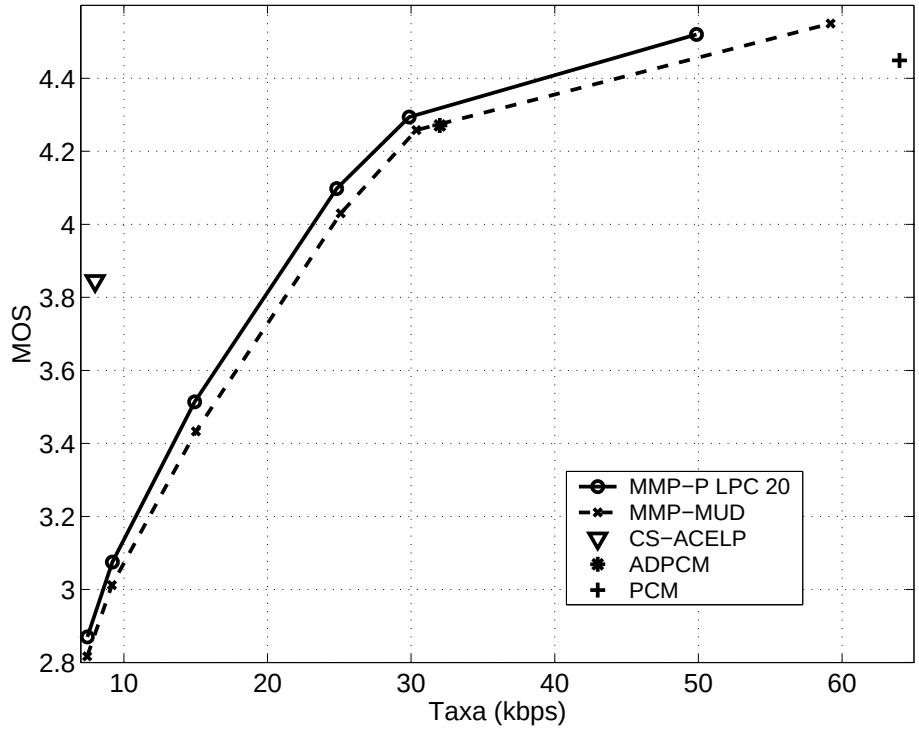


Figura 7.4: PESQ-MOS × taxa de codificação para o algoritmo MMP-P.

observa que $L = 512$ (o mesmo valor obtido anteriormente) apresentou um valor de PESQ-MOS de 3,00 na taxa de codificação de interesse, o melhor resultado obtido pelo MMP entre os vistos até agora neste trabalho.

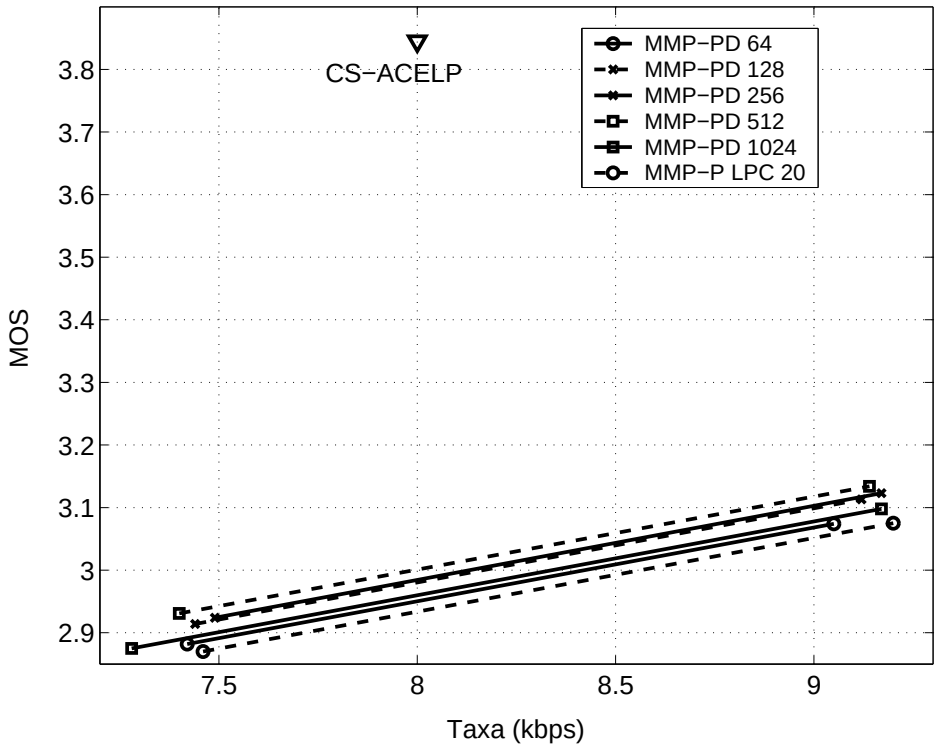


Figura 7.5: Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-PD com diferentes comprimentos L do dicionário de deslocamento.

O desempenho do MMP-PD para diferentes taxas é mostrado na Figura 7.6, em comparação ao desempenho do MMP-P, indicando uma melhora aproximadamente constante para as diferentes taxas de codificação.

7.5 MMP-P com dicionário inicial usando distribuição gaussiana generalizada MMP-GG

Para uma codificação eficiente do algoritmo MMP é necessário que o processo de adaptação do dicionário seja melhorado. Portanto, a estratégia de atualização do dicionário do MMP-PD deve explorar as características do sinal de voz inserindo novos segmentos no dicionário, usando-os para aproximar o sinal de entrada. O ganho obtido por usar os novos segmentos depende da existência de trechos do sinal de voz que podem ser aproximados por estes padrões. Assim, a eficiência da atualização do dicionário depende da regularidade do trecho do sinal. Sinais de voz com forte regularidade temporal, particularmente em trechos sonoros, tendem a favorecer a eficiência do processo de adaptação do dicionário, devido ao seu comportamento

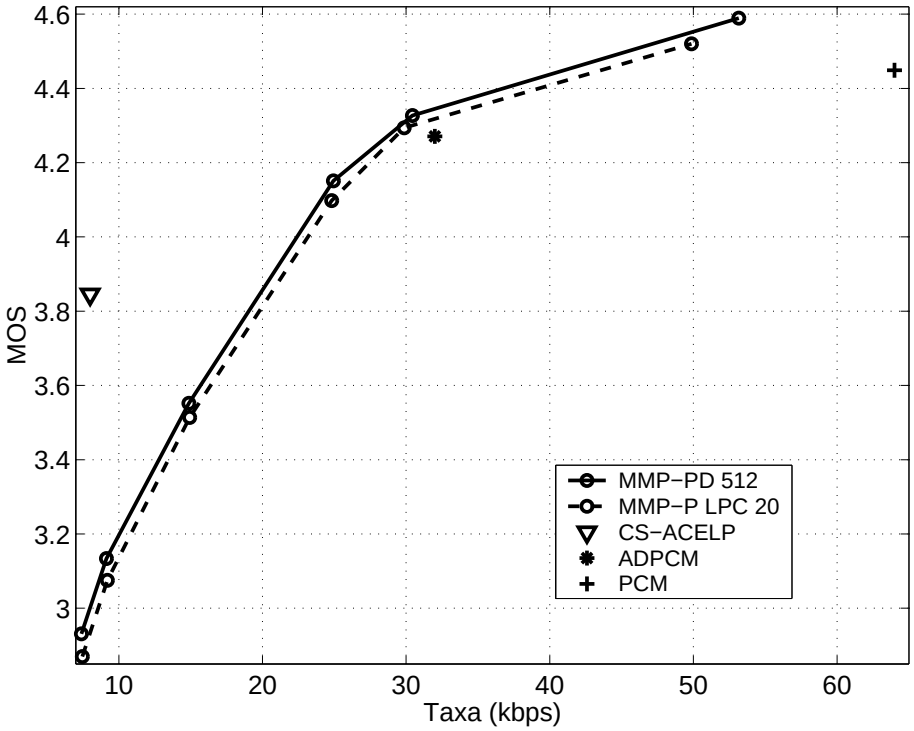


Figura 7.6: PESQ-MOS × taxa de codificação para o algoritmo MMP-PD.

quase periódico. Por outro lado, sinais com trechos surdos tendem a gerar um conjunto de padrões menos regular que são difíceis de serem aprendidos pelo processo de atualização.

A predição gera um sinal de resíduo com uma certa distribuição estatística, que pode ser modelada por uma distribuição gaussiana generalizada (DGG) [66]. Com base nesta ideia, propomos uma diferente quantização alternativa, diferente da $lei-\mu$ ou uniforme, para o dicionário inicial do MMP-PD e o seu processo de atualização. Para esta proposta, um histograma do erro de predição foi determinado para um banco de dados compreendendo 37 sentenças (incluindo 10 sentenças em Português Brasileiro, 7 em Chinês, 7 em Francês, 6 em Indiano e 7 em Inglês Britânico) com uma duração média de 5 s, amostrados a 8 kHz e com precisão de 16 bits, como pode ser visto na Figura 7.7. Nos referiremos a este banco de dados como DB1b.

A envoltória do histograma do erro de predição, mostrado com a linha tracejada na Figura 7.8, pode ser modelada por uma DGG [10, 67, 68] caracterizada por

$$p(x) = \left[\frac{\alpha\eta(\alpha,\beta)}{2\Gamma(1/\alpha)}\right] e^{-(\eta(\alpha,\beta)|x|)^{\alpha}}, \tag{7.7}$$

onde

$$\eta(\alpha,\beta) = \beta^{-1}\left[\frac{\Gamma\left(\frac{3}{\alpha}\right)}{\Gamma\left(\frac{1}{\alpha}\right)}\right]^{1/2} \tag{7.8}$$

e $\Gamma(.)$ é a função gama. Neste modelo, α define a taxa de decaimento da distribuição

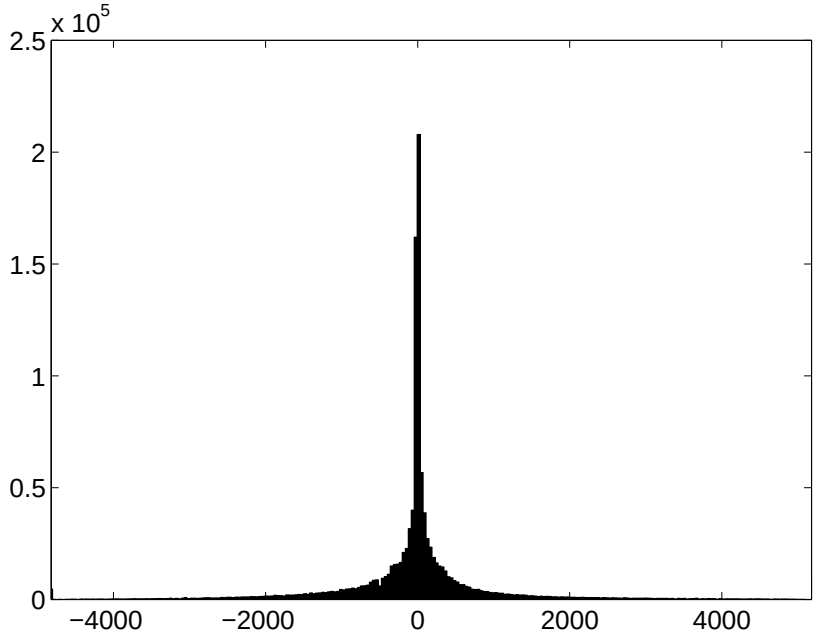


Figura 7.7: Histograma do erro de predição para o banco de frases DB2.

e β o desvio padrão correspondente. O comportamento da DGG com variância unitária é ilustrado na Figura 7.9 para vários valores de α. Para o erro de predição deste trabalho, os parâmetros encontrados foram $\alpha = 0{,}43$ e $\beta = 1{,}1031 \times 10^3$, produzindo a linha sólida na Figura 7.8.

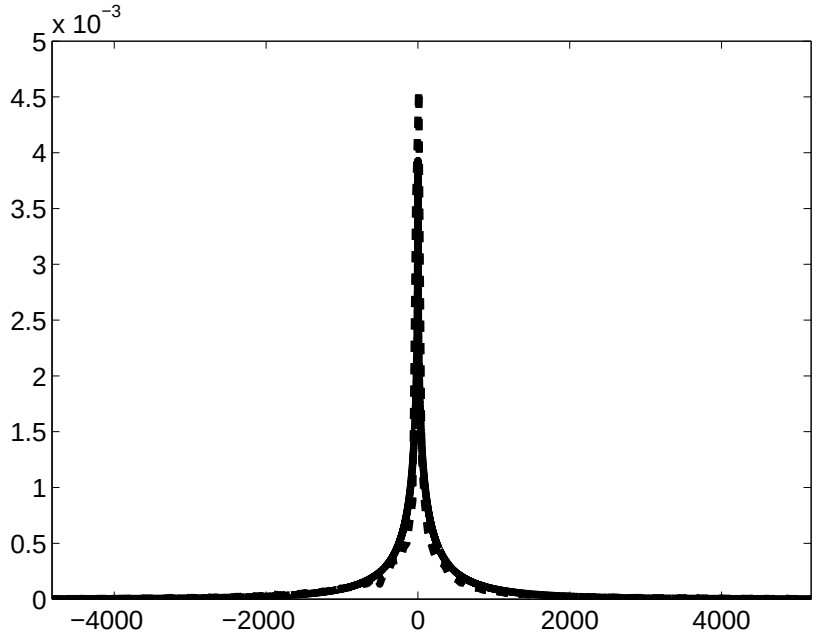


Figura 7.8: Modelando a envoltória do histograma do erro de predição (linha tracejada) usando uma DGG (linha sólida).

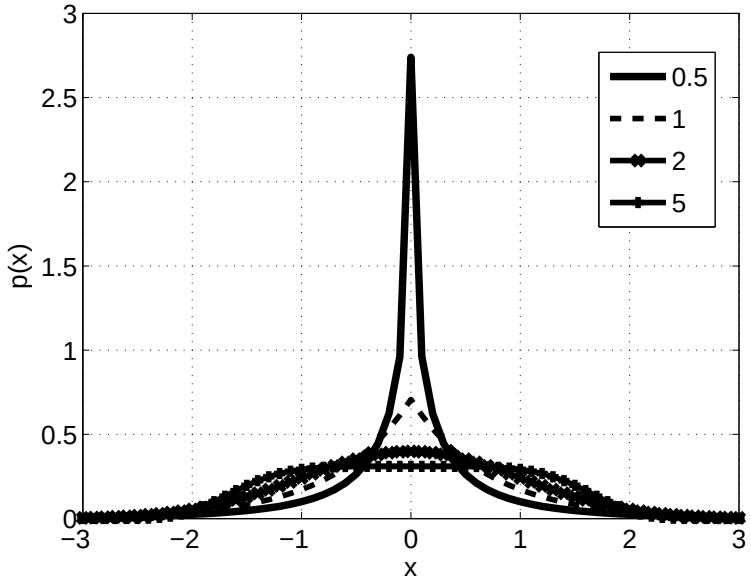


Figura 7.9: DGG para valores do parâmetro $\alpha = 0.5, 1, 2$ e 5. A distribuição é normalizada para variância unitária.

Com o modelo da DGG projetamos um dicionário não-uniforme usando o comando MATLAB `lloyd2`. Diferentes dicionários iniciais contendo 64, 128, 256 e 512 elementos em cada escala foram projetados. Além disso, também consideramos o uso ou não da quantização do sinal durante o processo de atualização do dicionário. O resultado do experimento é visto Figura 7.10, que indica um melhor desempenho para o dicionário GG com 256 elementos incorporando o estágio de quantização também na sua atualização. Todo este processo reduz o número de elementos no dicionário MMP, tornando o processo de codificação mais eficiente em termos de complexidade computacional.

As Figuras 7.11 e 7.12 comparam o desempenho do algoritmo MMP-GG com o da versão MMP-PD em torno de 8 kbps e para diferentes taxas, respectivamente, ambos os algoritmos com um modelo LP de ordem $N = 20$ e o dicionário de deslocamento com comprimento $L = 512$ e deslocamentos múltiplos de $\delta = 1$ amostra. Os resultados destas figuras indicam uma pequena melhora do MMP-GG para taxas altas (acima de 10 kbps) e um desempenho equivalente em torno de 8 kbps.

Com a nova quantização do dicionário inicial, novos experimentos foram feitos procurando uma configuração ótima do algoritmo MMP-GG em termos da ordem N do modelo LP (MMP-GL) e do comprimento L do dicionário de deslocamento (MMP-GD). Os resultados destes experimentos adicionais são apresentados nas Figuras 7.13 e 7.14, e indicam uma pequena melhora para um PESQ-MOS igual a 3,05 quando $N = 40$ e $L = 256$. Isto parece indicar uma melhor predição com a ordem mais alta e um menor espaçamento entre os trechos regulares do sinal de

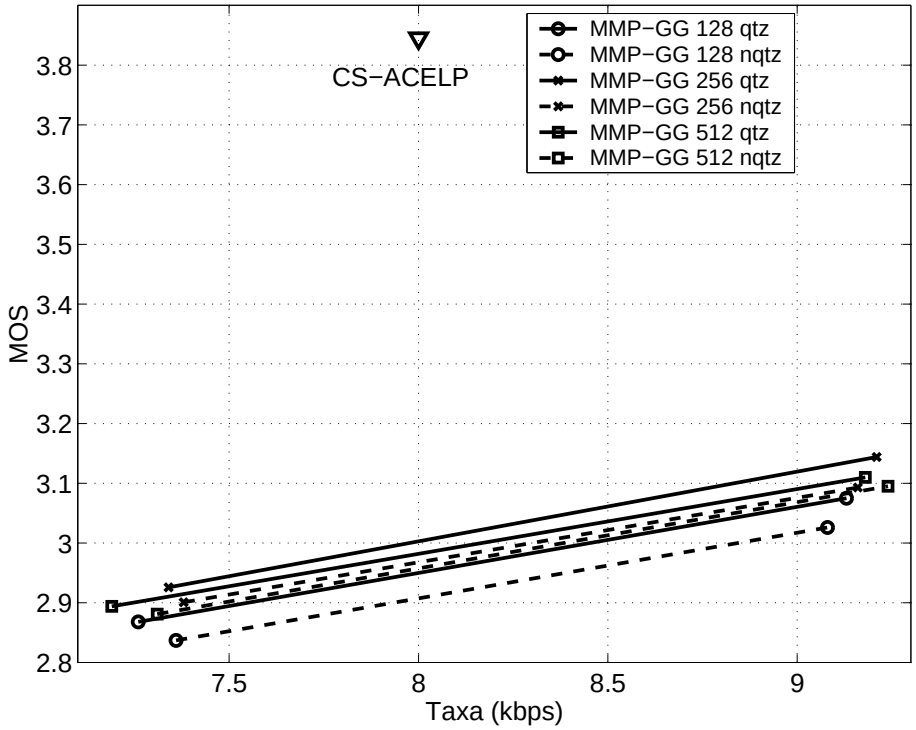


Figura 7.10: Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-GG usando diferentes estratégias para o dicionário inicial. "qtz" indica a quantização durante a atualização do dicionário, e "nqtz" indica que não há quantização durante a atualização do dicionário.

voz, tornando eficiente o uso do dicionário de deslocamentos e, consequentemente, a codificação resultante.

Analisando a Figura 7.15, onde comparamos os algoritmos MMP-GD e MMP-PD, observamos uma melhora de desempenho efetiva na faixa de 10 a 30 kbps.

7.6 Equalização de norma - MMP-EN

Estudos sobre as propriedades da DGG são descritos em [69, 70]. Com base em algumas teorias descritas nestas referências, introduzimos o conceito de equalização de norma no estágio de atualização do dicionário do MMP-GG, resultando na versão MMP-EN do nosso algoritmo. Resultados teóricos encontrados nestas referências mostram que a DGG descreve a estabilidade da norma L^α de um vetor $\mathbf{x}$ de amostras independentes, definida por

$$|x|_\alpha = \left(\sum_i |x_i|^\alpha \right)^{1/\alpha}, \tag{7.9}$$

quando sua dimensão tende ao infinito. Análises conduzidas em [69, 70] indicam que para altas taxas e distorções, os blocos do dicionário de resíduo se concentram numa

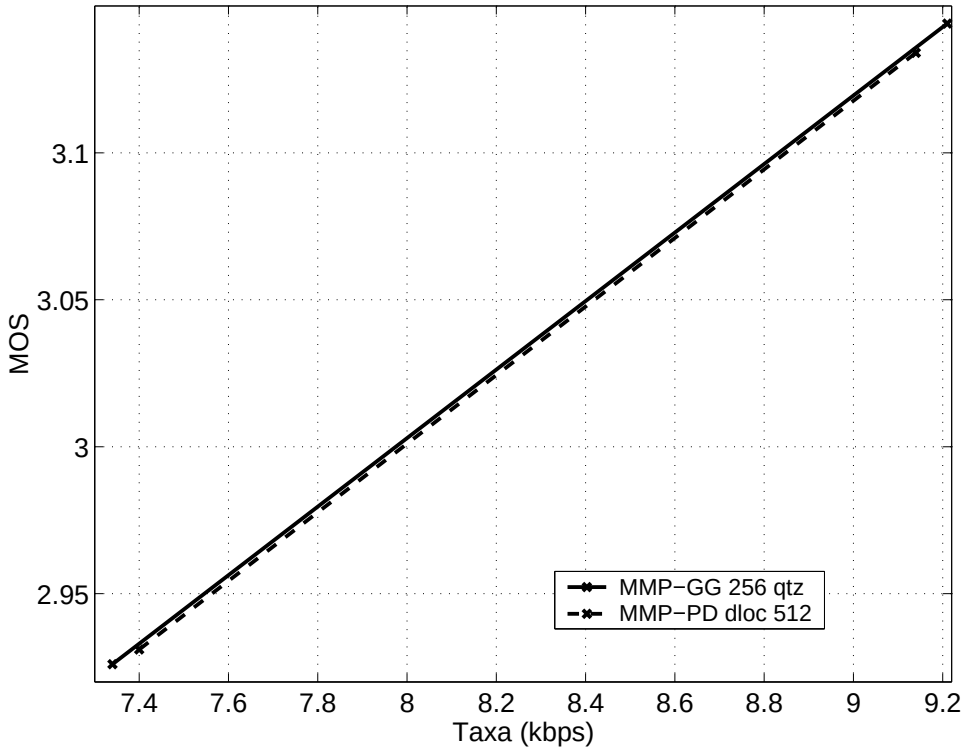


Figura 7.11: Resultado PESQ-MOS para diferentes taxas do algoritmo MMP-GG comparando com o algoritmo MMP-PD.

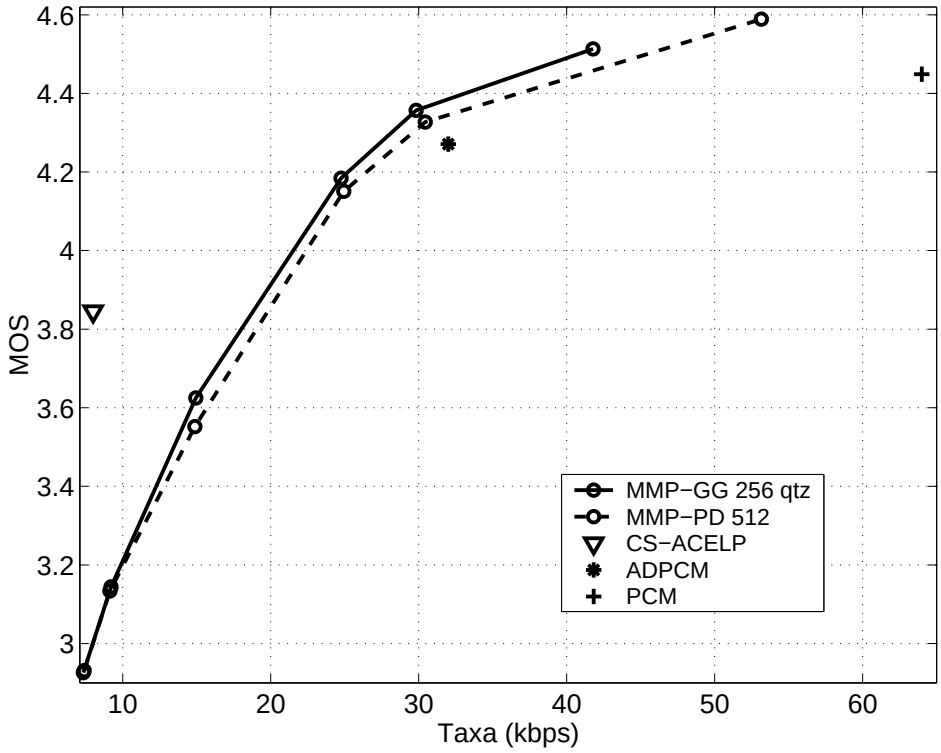


Figura 7.12: PESQ-MOS × taxa de codificação para o algoritmo MMP-GG.

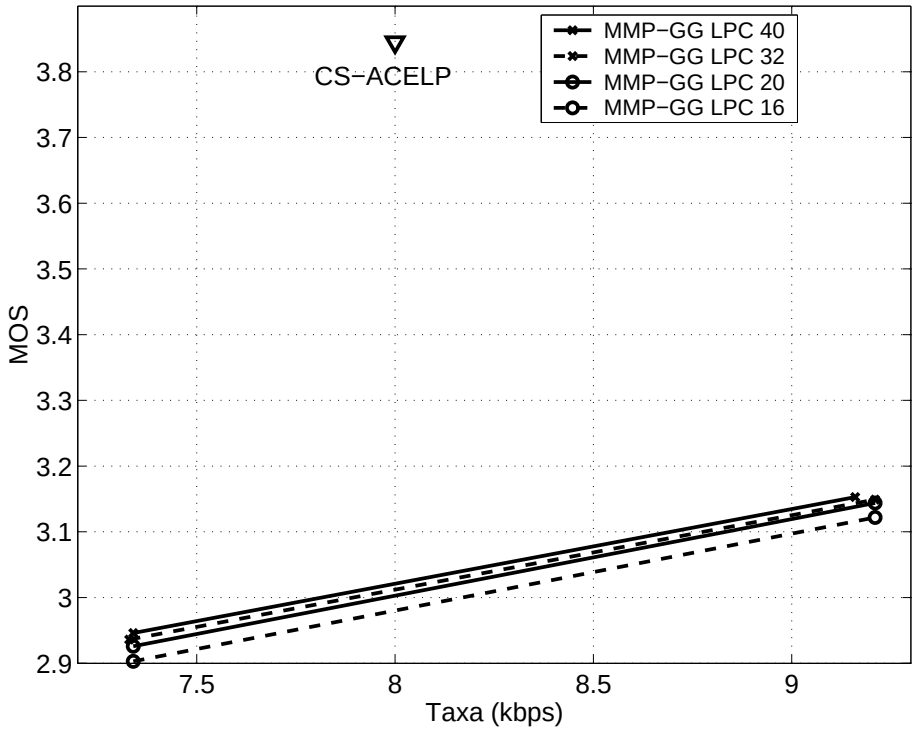


Figura 7.13: Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-GL para diferentes ordens do modelo LP.

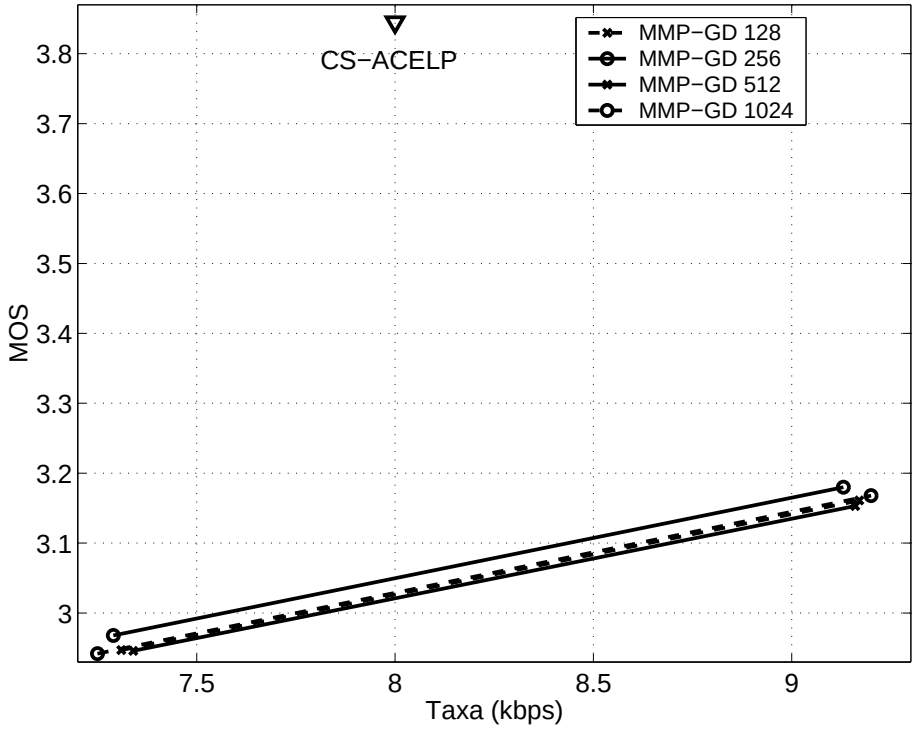


Figura 7.14: Resultado PESQ-MOS para diferentes taxas do algoritmo MMP-GD com diferentes comprimentos L do dicionário de deslocamento.

casca de probabilidade constante [71], cujos elementos seguem uma DGG. Como a predição tende a descorrelacionar os blocos, isto significa que os blocos tendem a

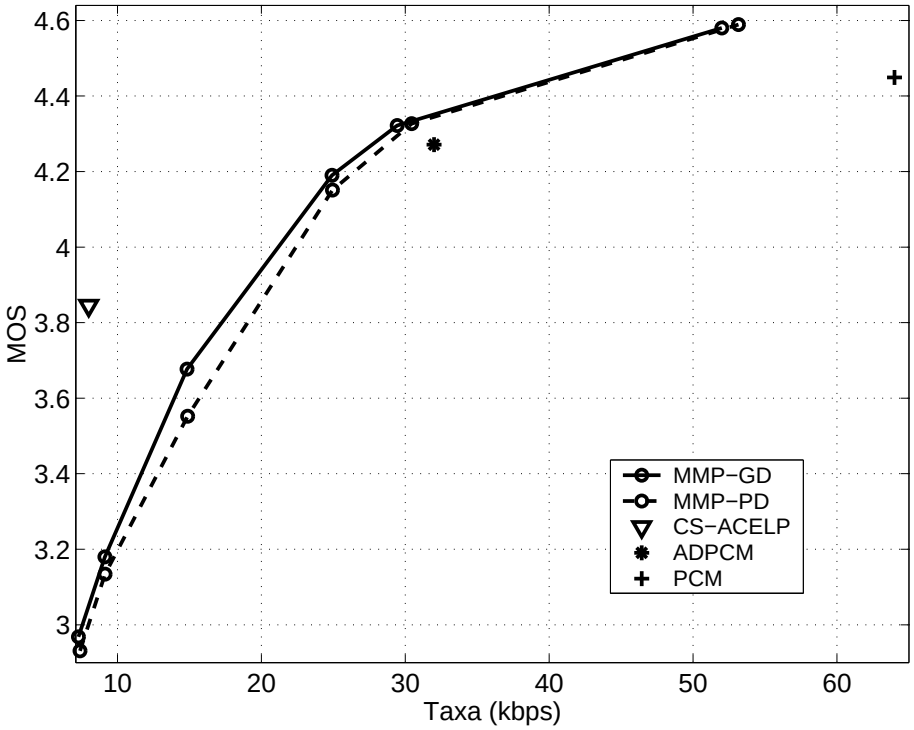


Figura 7.15: PESQ-MOS × taxa de codificação para o algoritmo MMP-GD.

formar uma casca multidimensional de pontos com norma L^α constante, para um valor particular de α. Observações foram feitas para validar esta suposição e os resultados mostraram que o desvio padrão da norma L^α cresce com a dimensão do vetor, mas a espessura da casca em relação à sua distância da origem tende a zero para vetores grandes, o que é de interesse prático do ponto de vista da distorção [70].

Resultados apresentados em [66] comprovam a eficiência do algoritmo MMP quando usamos a equalização de norma durante o processo de atualização do dicionário. É demonstrado que, apesar das dimensões reduzidas dos blocos usados, existe uma coerência na norma dos blocos de resíduos das diferentes escalas, significando que para evitar a inserção de blocos poucos úteis no dicionário, o novo bloco usado para atualizar o dicionário em escalas diferentes deverá ter uma norma aproximadamente constante. Porém, a transformação de escala do MMP modifica a norma L^α proporcionalmente ao fator de escalamento. Assim, é conveniente que se inclua, na operação de transformação de escala, uma normalização de modo a manter a norma L^α constante. Em outras palavras, ela usa o procedimento de escalonamento do bloco original,

$$\mathbf{R}^l = T^l_{l_o}(\mathbf{R}^{l_o}), \tag{7.10}$$

introduzindo uma etapa de equalização da norma, para assegurar a equivalência da

norma L^{α} do bloco original ($\mathbf{R}^{l_o}$) e do bloco escalonado ($\mathbf{R}^{l}$)

$$\mathbf{R}^{l} = s_{\alpha}^{l_o,l} T_{l_o}^{l}(\mathbf{R}^{l_o}), \tag{7.11}$$

onde o fator de equalização da norma é dado por

$$s_{\alpha}^{l_o,l} = \frac{|\mathbf{R}^{l_o}|_{\alpha}}{|\mathbf{R}^{l}|_{\alpha}}. \tag{7.12}$$

Seguindo esta ideia, a Figura 7.16 mostra o experimento considerando diferentes valores de α no processo de normalização para o algoritmo MMP-EN. Observa-se que o valor de α influencia no resultado final, e que a norma L^1 representa o melhor compromisso taxa×qualidade para a base de dados DB1, obtendo um resultado PESQ-MOS de 3,12. Além disso, computacionalmente a norma L^1 é de implementação mais eficiente que as demais, tornando a codificação mais eficiente também quanto à complexidade computacional.

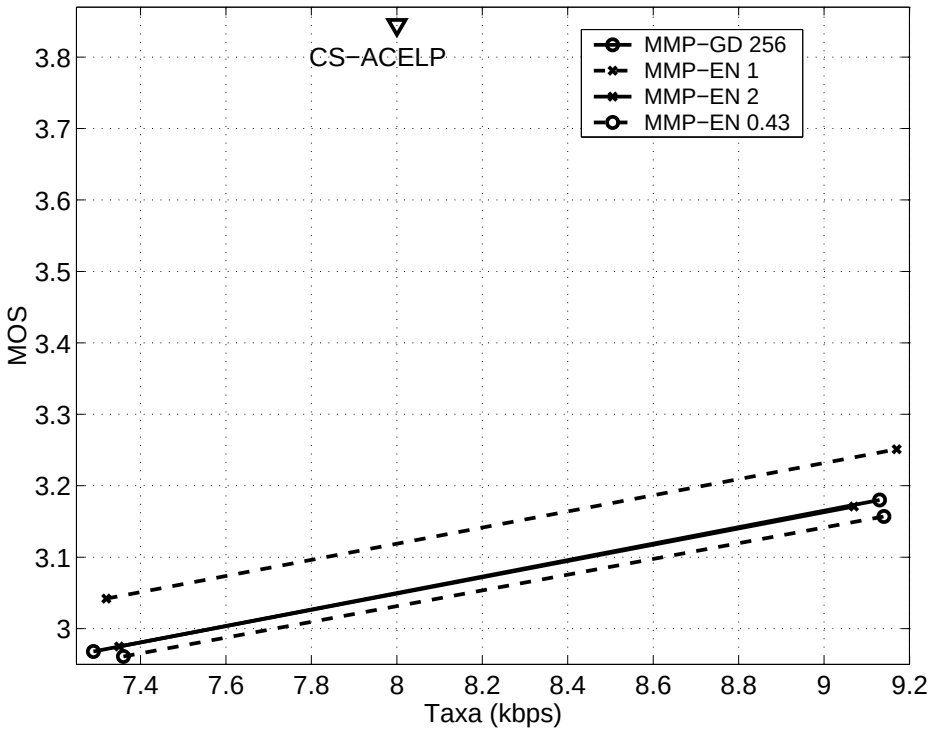


Figura 7.16: Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-EN usando diferentes normas para o estágio de atualização do dicionário.

A Figura 7.17 mostra o desempenho de ambas as versões do MMP (MMP-PD e MMP-EN) para uma faixa maior de taxas de codificação, juntamente com os resultados de codificadores G.711 (PCM) , G.726 (ADPCM), e G.729 (CS-ACELP). Nota-se que o MMP-EN recentemente proposto (linha sólida) consistentemente supera sua versão MMP-PD para quase todas as taxas, superando com folgas os desempenhos

dos codificadores G.711 e G.726. Para verificar o desempenho do MMP-EN para as taxas de 32 e 64 kbps o algoritmo teve o tamanho do seu dicionário inicial alterado para 1024 e 4096 elementos em cada uma de suas escalas. Com estas alterações o MMP-EN consegue apresentar o seu desempenho para as taxas de 32 e 64 kbps, como mostra a Figura 7.17.

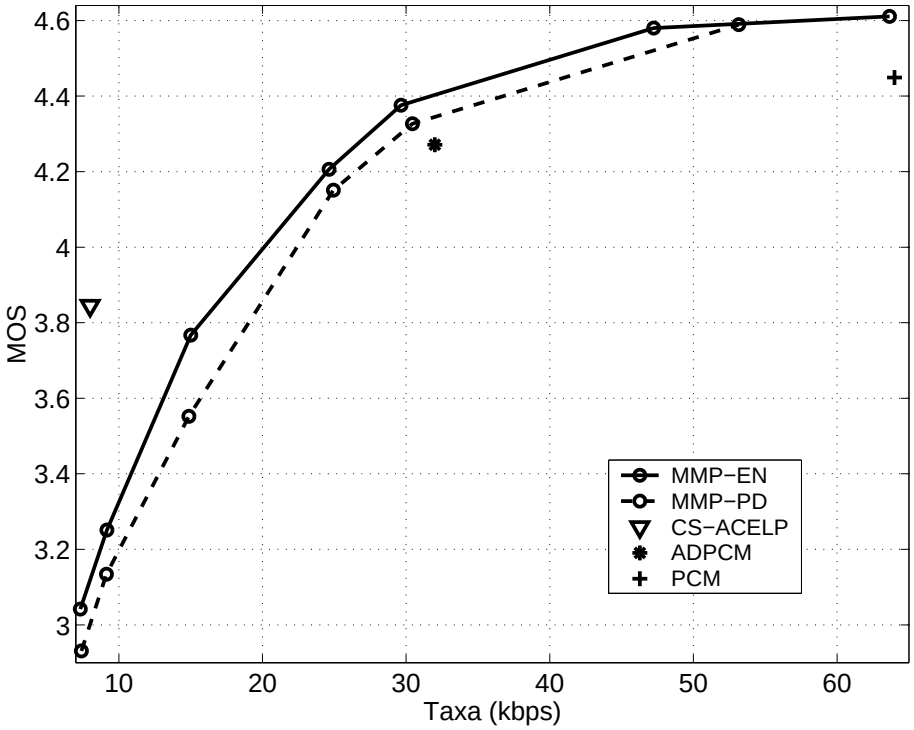


Figura 7.17: Resultado PESQ-MOS para diferentes taxas de codificação para versões MMP-PD (linha tracejada) e MMP-EN (linha sólida). Para as taxas em torno de 32 e 64 kbps, o MMP-PD usa o dicionário inicial com 1024 e 4096 elementos para cada escala do dicionário, respectivamente.

7.7 Partição das escalas do dicionário MMP-GAV

Estudos propostos em [4] apresentaram o conceito de partição do dicionário S de acordo com sua escala de origem. Este conceito explora, para uma dada escala, a distribuição estatística do uso dos elementos vindos de uma escala de origem (por contração ou expansão), permitindo a determinação das escalas de origem correspondentes aos elementos mais utilizados ao longo do processo de codificação. Neste sentido, o algoritmo MMP-GAV associa cada bloco do dicionário S a uma partição da escala do dicionário a que pertence, c_{k_l} (de acordo com a escala de origem), e um índice que aponta para o elemento dentro desta partição, $i_j^{c_{k_l}}$, como indicado na Figura 7.18. O termo "GAV" é devido a partição do dicionário de acordo com sua escala de origem apresentar a ideia de uma gaveta dentro de cada escala do

dicionário.

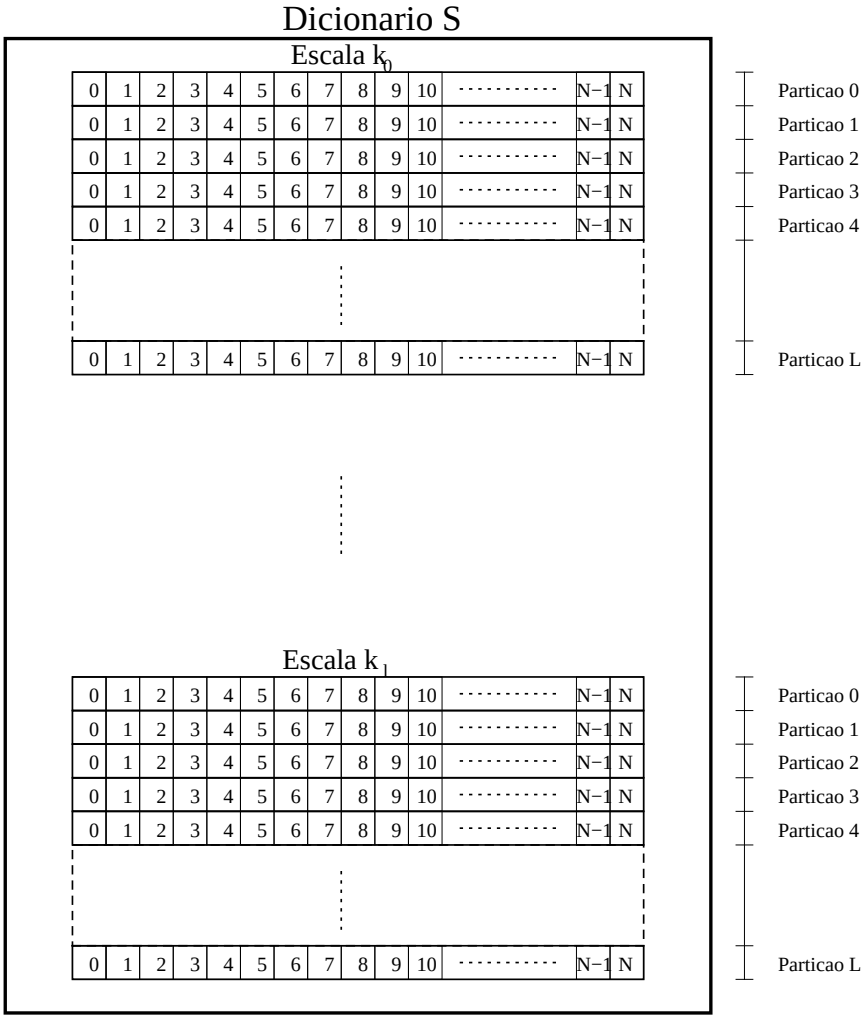


Figura 7.18: Processo de partição do dicionário, de acordo com a escala de origem. A partição i de uma escala k corresponde aos elementos da escala k originados da escala i, através de uma transformação de escala T_k^i.

Apesar de usar dois índices por bloco, ao invés de apenas um como anteriormente, a informação da escala facilita o trabalho do codificador aritmético, pois a estatística das escalas de origem é mais bem comportada do que a estatística dos vetores, melhorando o desempenho do processo de codificação como pode ser observado nas Figuras 7.19 e 7.20 para a base DB1.

Os experimentos apresentados nas Figuras 7.19 e 7.20 apresentam o dicionário inicial e o seu estágio de atualização quantizados de acordo com a DGG anteriormente obtida; o dicionário inicial é composto de 256 elementos, para cada uma de suas escalas, e particionado de acordo com a escala de origem; a ordem $N = 40$ para o modelo LP; uma janela de deslocamento de comprimento $L = 256$ para o dicionário auxiliar de deslocamento; e o processo de atualização do dicionário inclui equalização de norma para norma L^1.

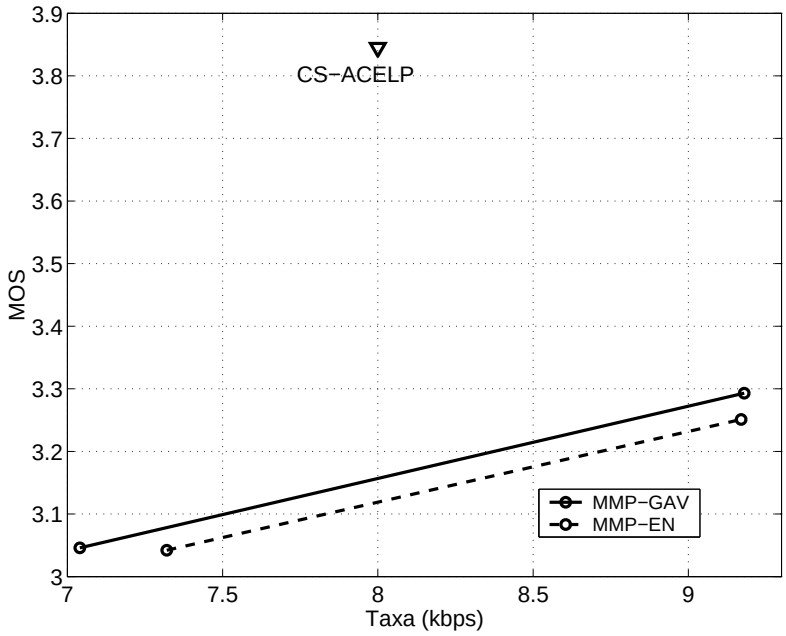


Figura 7.19: Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-GAV.

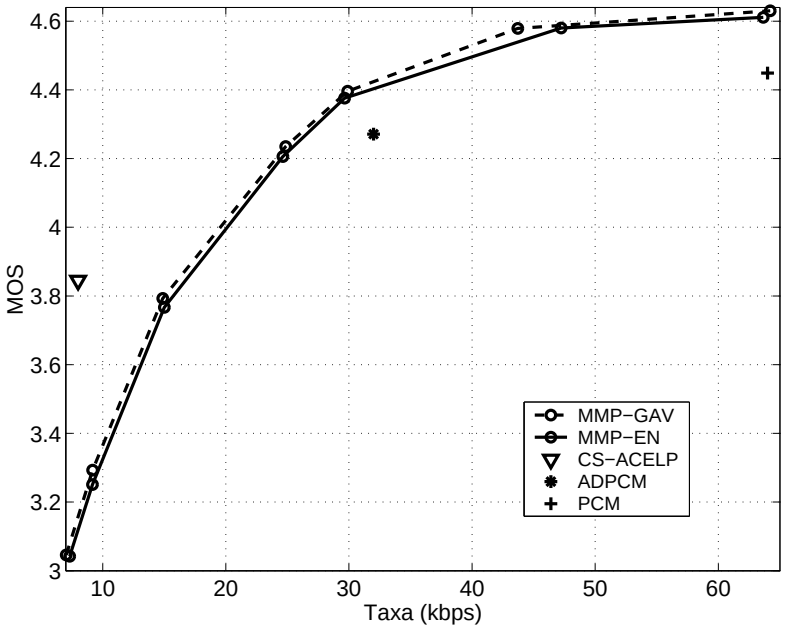


Figura 7.20: Resultado PESQ-MOS para diferentes taxas do algoritmo MMP-GAV.

7.8 Validação dos resultados

Até aqui, os experimentos de codificação foram realizados utilizando a base DB1, composta por sinais de voz de um mesmo orador e de uma mesma língua. Para validar nossos resultados de forma mais geral, avaliamos o desempenho do algoritmo MMP-GAV com uma nova base de dados (DB2) composta de 40 sinais (8 em Chinês, 8 em Francês, 8 em Indiano, 8 em Inglês Britânico e 8 em Inglês dos

EUA) com duração média de 8 s e codificados em PCM de 16 bits/amostra com 8000 amostras/s, extraídos do OSR (*Open Speech Repository*) [72].

As Figuras 7.19 e 7.20 comparam os resultados do MMP-GAV com os resultados alcançados na Seção 7.6: a primeira apresenta os resultados em torno da taxa de 8 kbps e a segunda mostra a comparação com os codificadores G.711 e G.726 em torno das taxas de 64 e 32 kbps, respectivamente. Por estas figuras, nota-se que o desempenho do algoritmo MMP-GAV mostrou-se equivalente ao dos codificadores G.711 e G.726, mas ainda significativamente inferior ao do G.729.

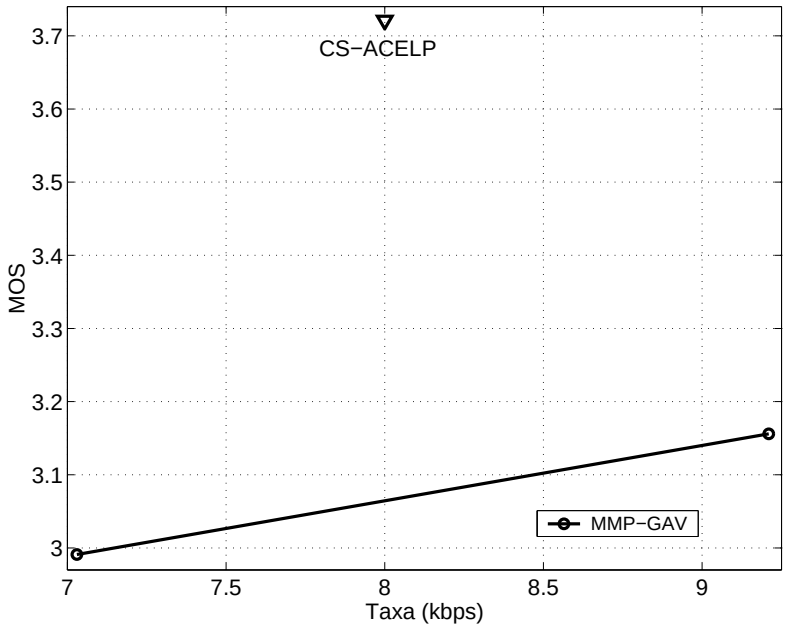


Figura 7.21: Resultado PESQ-MOS em torno de 8 kbps para o algoritmo MMP-GAV para a base de dados DB2.

7.9 Alterando o tamanho do bloco de predição

O desempenho do algoritmo MMP-P depende fundamentalmente de quão boa é a predição do próximo bloco a ser codificado: Se esta predição for muito boa o resíduo tenderá a apresentar um valor bem próximo de zero, facilitando todas as etapas subsequentes da codificação MMP. A Figura 7.23 compara, para um dado sinal codificado, o resultado apenas da predição com um trecho do sinal original. Neste caso a predição é feita para 128 amostras. As diferenças entre estes sinais são devidas a diferentes fatores tais como a quantização do sinal, ao modelo LP não ideal, e principalmente à codificação prévia das amostras passadas usadas na definição do modelo LP. Para amenizar alguns destes problemas, e melhorar a predição das amostras, vários testes foram realizados alterando o tamanho do bloco de predição e o tamanho do dicionário inicial. Nestes experimentos o algoritmo MMP usado

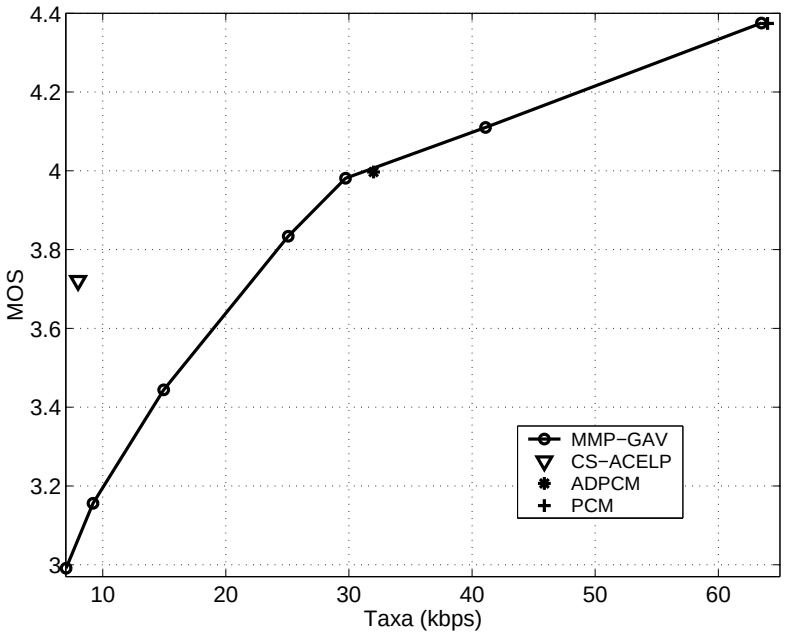


Figura 7.22: Resultado PESQ-MOS para diferentes taxas do algoritmo MMP-GAV para a base de dados DB2, comparando com os outros codificadores de voz.

foi o mesmo da Seção 7.7, MMP-GAV, porém o tamanho do bloco que será predito sofrerá alterações conforme a Tabela 7.1. Os resultados obtidos nestes testes estão apresentados na Tabela 7.1, de onde se infere que o desempenho do MMP-GAV pode ser melhorado usando-se um bloco de predição de tamanho 16 e um dicionário inicial com 256 elementos para cada escala do dicionário. Este resultado é usado nas próximas seções para analisar o desempenho do MMP-GAV.

Tabela 7.1: Experimentos alterando o tamanho do dicionário inicial e o tamanho do bloco de predição. Os valores dados são do PESQ-MOS obtido apenas com a predição.

Tamanho do bloco de predição	Diferentes tamanhos do dicionário			
	64	128	256	512
4	3,142	3,111	3,131	3,094
8	3,218	3,305	3,251	3,287
16	2,960	3,321	3,335	3,292
32	3,006	3,235	3,259	3,242

A Figura 7.24 apresenta o sinal predito usando o resultado ótimo da Tabela 7.1 comparado com o sinal original, indicando uma melhora em relação ao resultado da Figura 7.23, obtida com um bloco de predição de tamanho 256. Levando-se em consideração estes novos valores, o MMP-GAV melhorou seu desempenho elevando seu resultado PESQ-MOS de 3,06 para 3,34 para o banco DB2. Isto mostra claramente que é mais conveniente fazer a predição para blocos de tamanho 16, para ordem do modelo LP $N = 40$, muito menores que as de tamanho 128 originalmente utilizados. Note que, como os coeficientes LPC não são transmitidos, mas calculados apenas

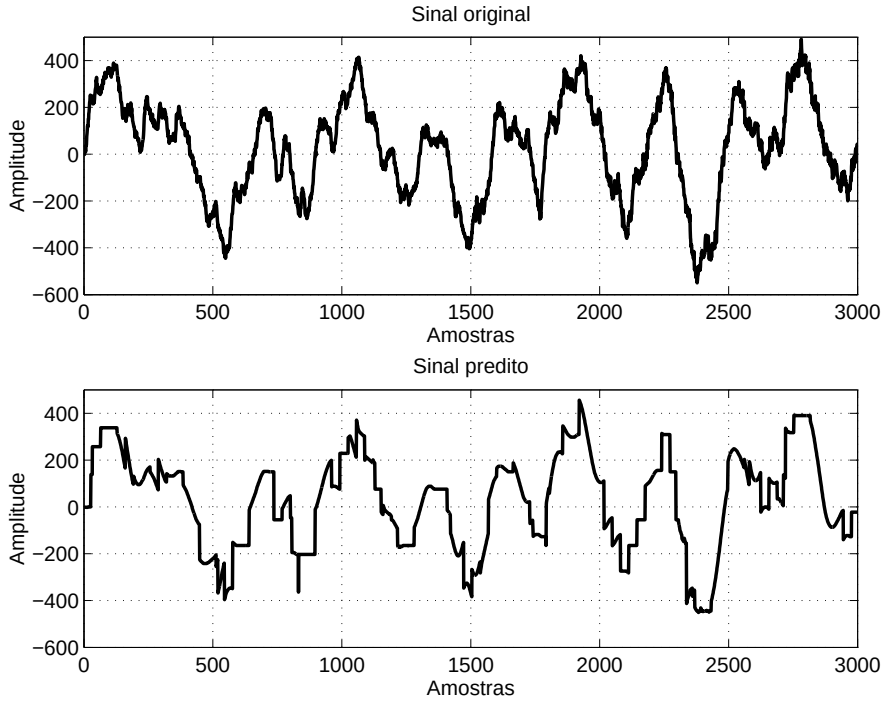


Figura 7.23: Trecho de um sinal de voz sendo comparado com o bloco de predição.

com base no sinal já decodificado, não há *overhead* de taxa nenhuma quando se usa um tamanho de bloco de predição pequeno e uma ordem alta.

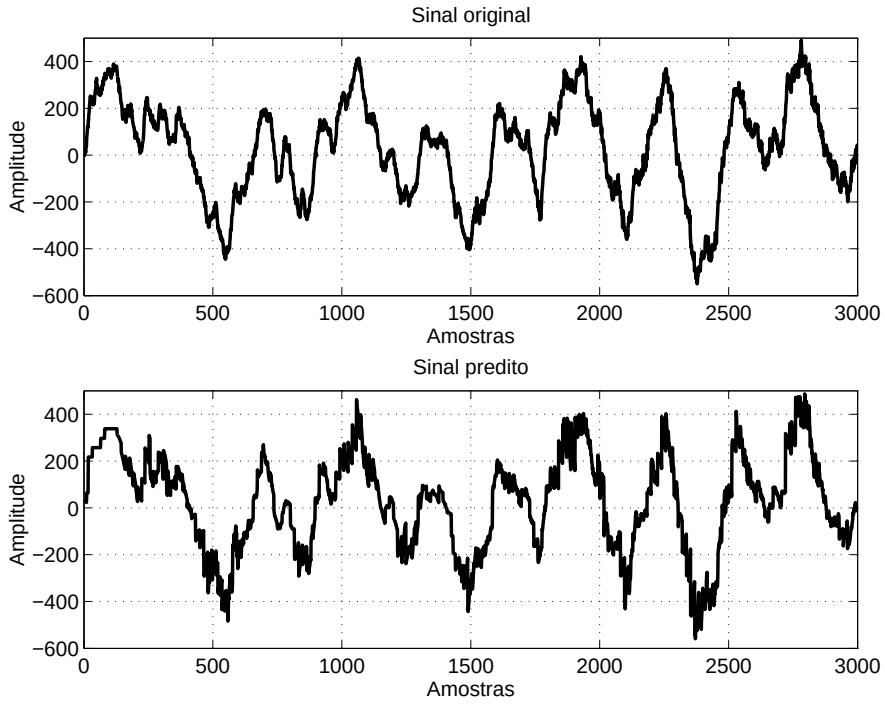


Figura 7.24: Trecho de um sinal de voz sendo comparado com o bloco predito, para uma tamanho de bloco de predição igual a 16 amostras.

7.10 Frases misturadas

Quando estamos em um ambiente cheio de pessoas (em uma videoconferência, por exemplo), os sinais de diferentes locutores chegam misturados num único sinal. Nesta seção, procuramos mostrar a eficiência do algoritmo MMP neste contexto de vozes misturadas, formando alguns sinais pela adição pura e simples de diferentes sinais da base DB2.

A Figura 7.25 mostra o desempenho do algoritmo MMP quando misturamos 2, 3, 4 e 5 frases contendo trechos de silêncio, comparando com o desempenho do G.729. Nota-se que quando misturamos 3 ou mais frases o algoritmo MMP consegue se aproximar do codificador G.729, confirmando o fato esperado de que o processo de mistura de vozes é menos crítico para a técnica de compressão baseada na forma de onda do que para o modelo paramétrico do G.729. De fato, uma análise detalhada dos resultados mostrou que o G.729 ainda obteve um resultado razoavelmente bom na mistura de dois sinais devido à existência de trechos de silêncio nos sinais individuais, que evitavam a sobreposição efetiva da voz no sinal soma. Assim, um trecho ativo de um determinado sinal se encaixava no trecho contendo silêncio do outro sinal, não afetando o desempenho do codificador G.729 de forma significativa.

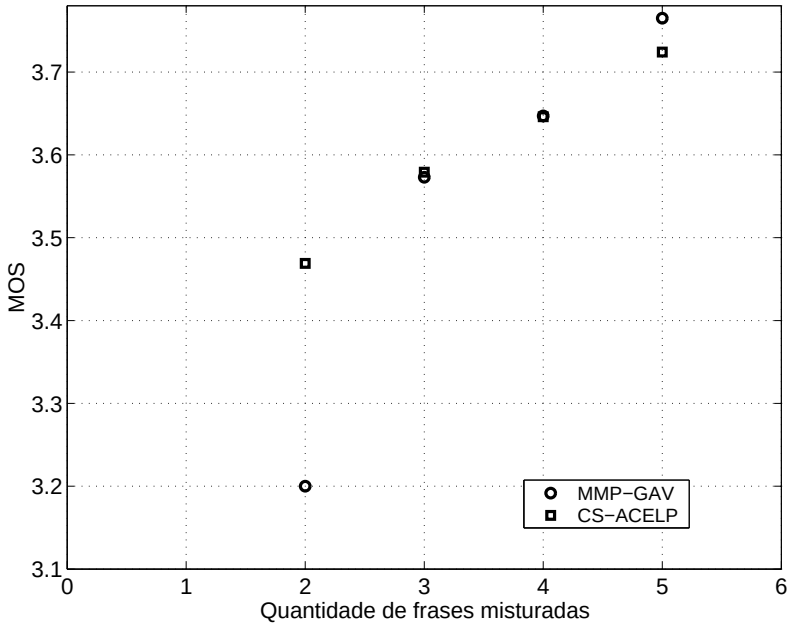


Figura 7.25: Resultado PESQ-MOS para frases misturadas (contendo trechos de silêncio) em torno de 8 kbps usando o algoritmo MMP-GAV, baseado em partição das escalas do dicionário.

Ao retirarmos os trechos de silêncio dos sinais individuais, antes de realizarmos a adição dos mesmos, o desempenho do G.729 passa a decair com o número de sinais sobrepostos, como era de se esperar, enquanto que o desempenho do MMP até melhora, como pode ser observado na Figura 7.26. De fato, desta figura podemos

concluir que ao aumentarmos a quantidade de frases misturadas fica difícil para o G.729 identificar o período de *pitch* corretamente, conforme mencionado anteriormente, dificultando seu processo de codificação, o que não acontece com o MMP.

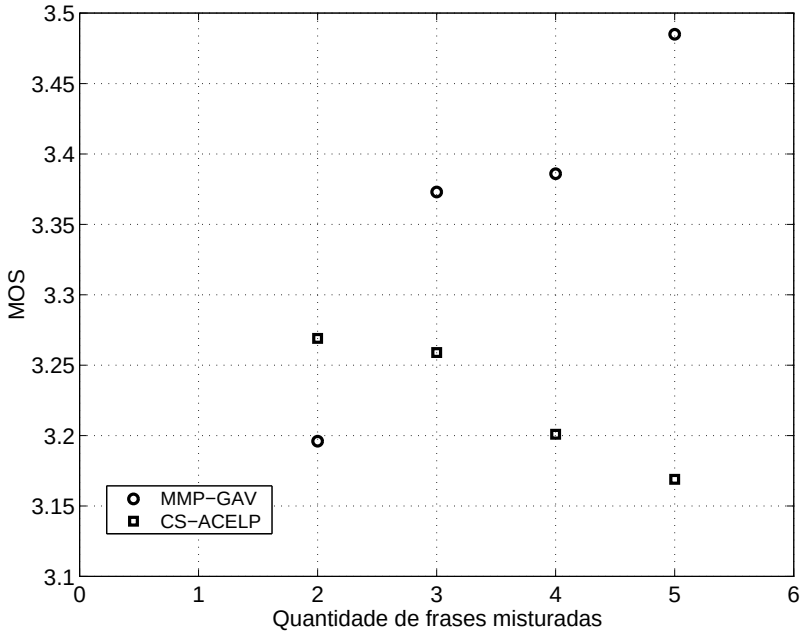


Figura 7.26: Resultado PESQ-MOS para frases misturadas (sem trechos de silêncio) em torno de 8 kbps usando o algoritmo MMP-GAV, baseado em partição das escalas do dicionário.

7.11 Frases concatenadas

Nesta seção procuramos apresentar o comportamento do MMP-GAV ótimo da Tabela 7.1 para experimentos envolvendo frases concatenadas, a fim de identificar o tempo que o algoritmo MMP-GAV leva para aprender as características do sinal de voz. Para isto, concatenamos diversas frases, após remover os trechos de silêncio, e realizamos a codificação, obtendo os resultados vistos na Figura 7.27 para 10 e 12 frases: o primeiro sinal era composto pelas 10 frases do banco DB1 (sem os trechos de silêncio), dando um total de 28 s; o segundo sinal era igual ao primeiro acrescido de 2 frases do banco DB2 (frases us68 e us78, também sem trechos de silêncio), totalizando 35 s. Os resultados indicam que o MMP-GAV é capaz de aprender o comportamento temporal do sinal de voz, melhorando seu desempenho com o tempo, mas de uma forma relativamente lenta. Isto é percebido pelo tempo de duração das frases concatenadas, quando possui 35 s de duração o algoritmo MMP apresenta uma pequena melhoria, mostrando que o processo de aprendizagem necessita de mais tempo para aprender de forma efetiva as características do sinal de voz.

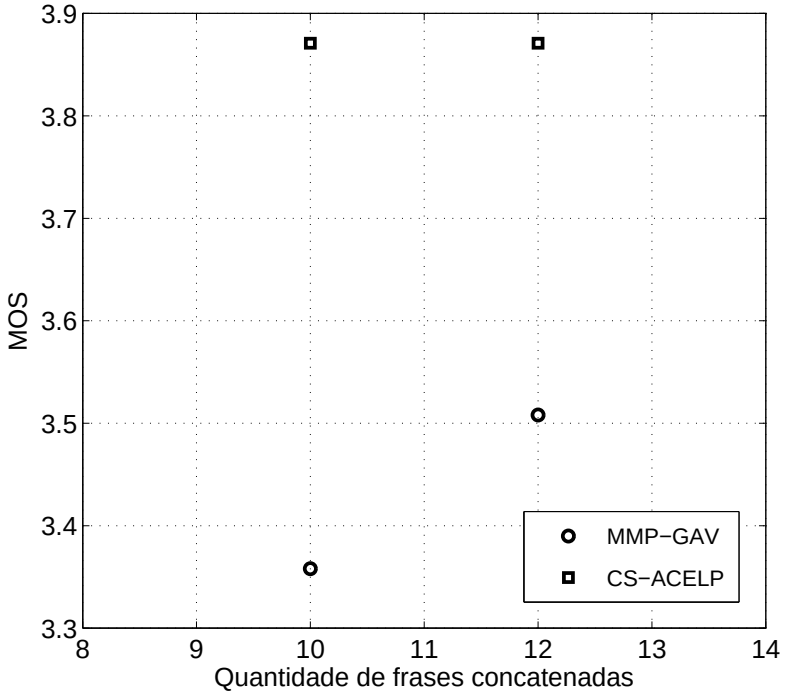


Figura 7.27: Resultado PESQ-MOS para frases concatenadas (sem trechos de silêncio) em torno de 8 kbps usando o algoritmo MMP-GAV baseado em partição das escalas do dicionário.

7.12 Quantizando o bloco de resíduos, saída do preditor LPC e o sinal de entrada

No processo de atualização do dicionário no MMP-GAV, o bloco a ser incluído no dicionário passa por um processo de quantização, colocando as amostras do bloco a ser incluído no mesmo nível de quantização do dicionário inicial. Seguindo esta mesma ideia realizamos experimentos quantizando a saída da predição linear, o sinal de entrada e o bloco de resíduo. A Figura 7.28 apresenta o resultado destes experimentos, considerando diferentes versões do algoritmo MMP-GAV: com quantização do bloco de resíduos (MMP-GAV QuantResd); com quantização da saída da predição linear (MMP-GAV QuantLPC); com quantizações da saída da predição linear e do bloco de resíduo (MMP-GAV QuantLPCREsd); e com quantizações da saída da predição linear e do sinal de entrada (MMP-GAV QuantLPCSinal). Observa-se pela Figura 7.28 que o melhor resultado foi para a versão MMP-GAV QuantResd, indicando que ao quantizarmos o bloco de resíduos para o mesmo nível de quantização do dicionário inicial, as amostras do bloco de resíduos ficam mais próximas dos vetores do dicionário, permitindo um melhor casamento de padrões.

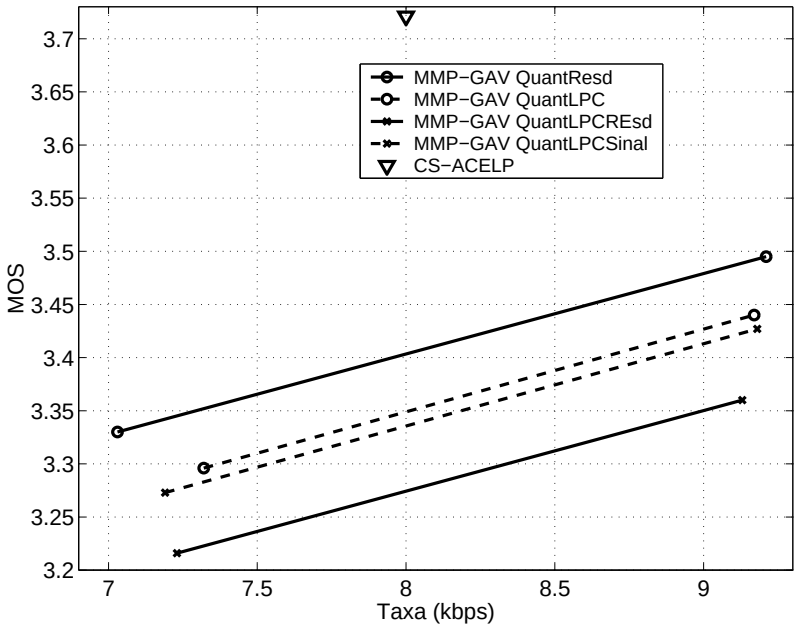


Figura 7.28: Resultado PESQ-MOS em torno de 8 kbps usando o algoritmo MMP-GAV baseado em partição das escalas do dicionário quantizando o bloco de resíduos, a saída do LPC e o sinal de entrada.

7.13 Pós-processamento com filtro passa-baixas

Na saída do decodificador, o sinal de voz pode apresentar componentes de alta frequência, correspondentes a artefatos da codificação. Para minimizar o efeito destas componentes espectrais, podemos inserir um filtro passa-baixas na saída do decodificador do algoritmo MMP-GAV, resultando numa melhora de seu desempenho, como é atestado pela Figura 7.29 que considera diferentes tipos de filtro. O resultado apresentado nesta figura mostra que o processo de pós-filtragem levou o resultado PESQ-MOS do algoritmo MMP operando em torno de 8 kbps para 3,58, aproximando-o do valor de 3,72 para o G.729. No caso, o melhor desempenho do MMP foi obtido com o filtro FIR de ordem 10 e frequencia de corte de 200 Hz, gerado com a janela *raised cosine* com fator *roll-off* 0,5, cuja resposta em magnitude é apresentada na Figura 7.30 e os coeficientes são dados na Tabela 7.2.

A Figura 7.31 apresenta o desempenho taxa×qualidade do algoritmo MMP-GAV com quantização do bloco de resíduos quando usamos pós-processamento com filtro passa-baixas na saída do seu decodificador, comparando-o ainda com os resultados obtidos na Figura 7.28. Os resultados desta figura indica uma pequena melhora do algoritmo MMP para taxa em torno de 8 kbps e um desempenho equivalente para as taxas altas (acima de 30 kbps).

Os resultados obtidos nesta seção mostram que o desempenho, em termos da nota PESQ-MOS, de nossa versão final do algoritmo MMP se aproxima bastante do desempenho correspondente do codificador G.729. No capítulo seguinte, de con-

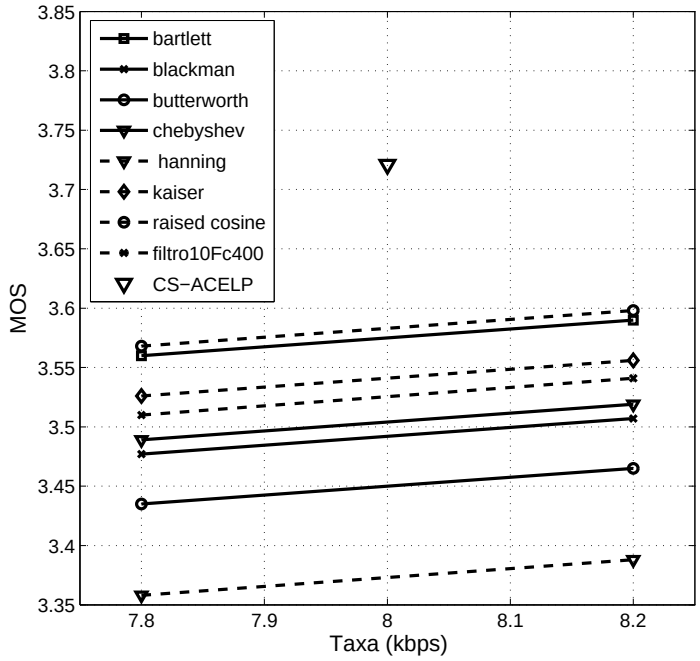


Figura 7.29: Resultado PESQ-MOS em torno de 8 kbps para o MMP com diferentes tipos de filtros passa-baixas.

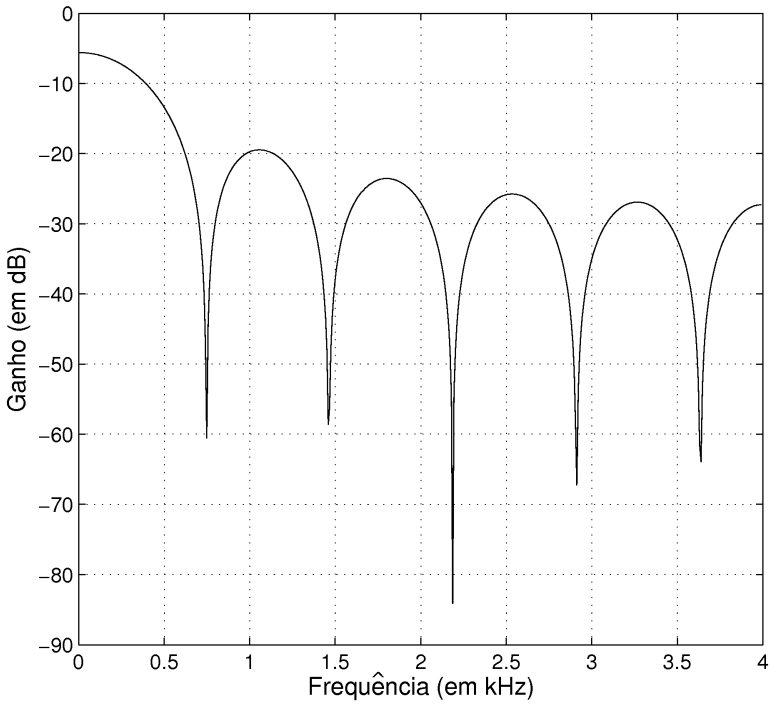


Figura 7.30: Resposta em magnitude do filtro passa-baixas do tipo *raised cosine* usado no estágio de pós-filtragem.

Tabela 7.2: Coeficientes do filtro *raised cosine*.

$h(0)$	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

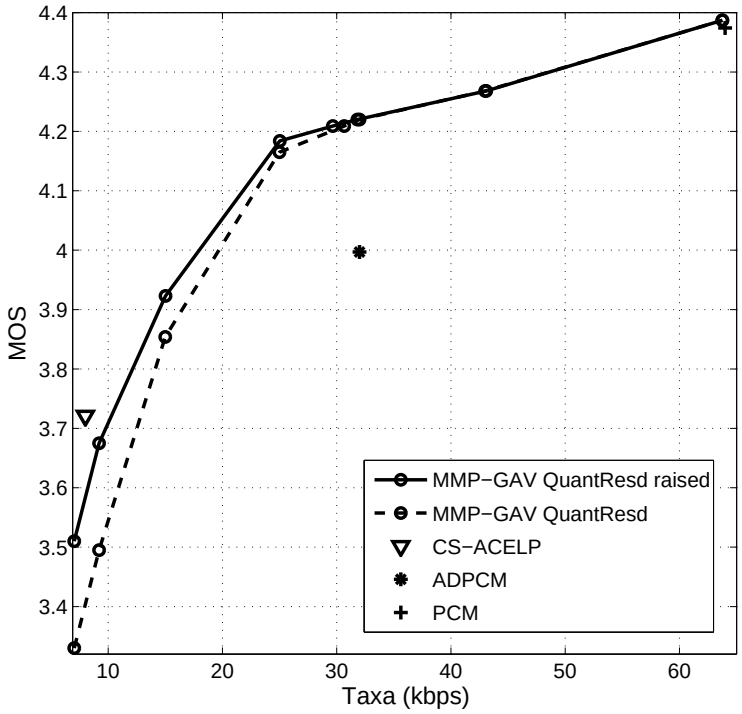


Figura 7.31: Resultado do banco de dados DB2 utilizando pós-processamento com filtro passa-baixas, comparando com os outros codificadores de voz.

clusões, serão comentadas as possibilidades de futuras melhorias no desempenho do algoritmo MMP aplicado à codificação de sinais de voz.

7.14 Razão Sinal-ruído (SNR) para o algoritmo MMP

Nesta seção analisamos o desempenho do algoritmo MMP verificando seu comportamento em relação à medida de razão sinal-ruído (*Signal-to-Noise Ratio* - SNR). O resultado para as taxas em torno de 8 kbps são mostrados na Figura 7.32, onde:

a versão MMP-GAV tem bloco de predição de 16 amostras e dicionário inicial com 256 elementos em cada escala (vide Seção 7.9); a versão MMP-GAV ResdQuant é o algoritmo MMP da Seção 7.9 com quantização do sinal de resíduo (vide Seção 7.12); e a versõ filt10Fc400 denota o algoritmo MMP da Seção 7.12 com um estágio de pós-filtragem usando um filtro FIR de ordem 10 com seus coeficientes apresentados na Tabela 7.2.

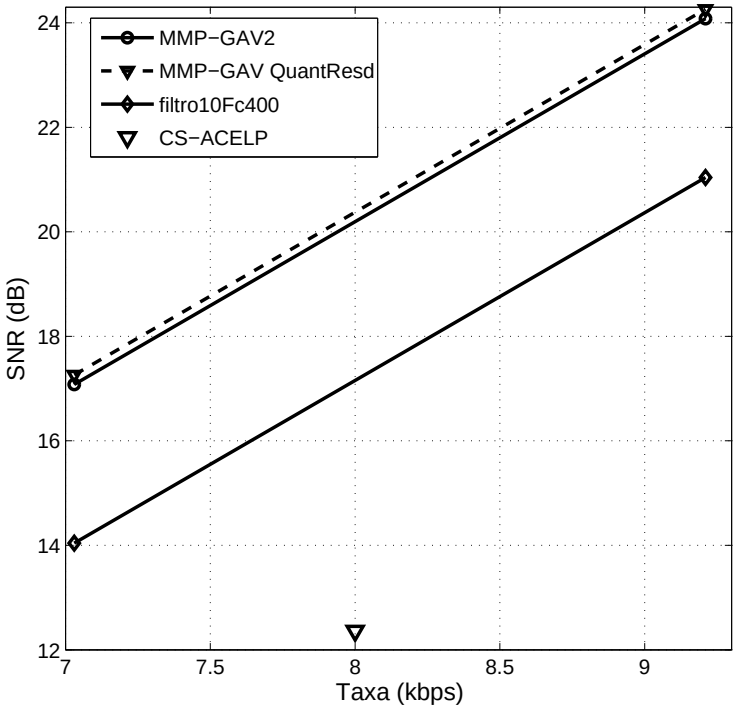


Figura 7.32: Razão Sinal-ruído em torno de 8 kbps para o algoritmo MMP.

Desta figura, podemos concluir que o algoritmo MMP atinge uma SNR bem superior ao G.729, o que é facilmente justificado por ser esta a medida de qualidade utilizada na busca do melhor casamento de padrões no algoritmo proposto. No caso, a versão com filtro passa-baixas reduz a SNR ao eliminar artefatos de alta frequencia que alteram de forma significativa a qualidade percebida do sinal resultante.

Capítulo 8

Conclusão

Nesta tese, investigamos a aplicação de um algoritmo de recorrência de padrões multiescalas para codificar um sinal de voz. Foi utilizado como ponto de partida o algoritmo MMP (*Multidimensional Multiscale Parser*), que foi originalmente desenvolvido para codificação de imagens, apresentando ótimos resultados. A ideia é fazer adaptações e alterações no algoritmo de forma a torná-lo mais adequado à codificação de um sinal de voz, investigando formas de explorar as características específicas deste sinal no contexto da recorrênica de padrões multiescalas. Propusemos para este fim o estudo dos comportamentos do seu estágio de codificação e do seu dicionário, bem como a sua forma de segmentação.

Inicialmente, avaliamos o algoritmo MMP proposto no Capítulo 6, estudando seu comportamento no domínio temporal. Experimentos envolvendo o tipo de quantização do dicionário inicial, bem como identificação do seu tamanho ótimo (quantidade de elementos no dicionário) e o uso de um dicionário auxiliar de deslocamento no processo de codificação estimularam a busca para alcançar os resultados obtidos pelos codificadores padrões de voz. No primeiro momento, tivemos um desempenho bem aquém dos codificadores padrões, conforme descrito na Seção 6.1. Analisando o comportamento do sinal de voz no domínio temporal verificou-se a possibilidade do uso de um dicionário auxiliar contendo informações previamentes codificadas, chamado dicionário de deslocamento (Seções 6.2 e 6.4). O resultado obtido continuou muito aquém dos obtidos com os padrões G.729 e G.726, no entanto superou o valor obtido pelo G.711. Isto apresentou uma indicação de que o dicionário pode afetar bastante o desempenho do algoritmo MMP, pois é necessário que se adapte às características do sinal de voz rapidamente para aprender e conhecer os padrões do sinal a ser codificado, e assim, efetuar casamentos melhores. Ainda no domínio do tempo, verificou-se que o uso de um dicionário não-uniforme melhora o desempenho do algoritmo MMP. Isto motivou investigações acerca do dicionário inicial, e os experimentos apresentados na Seção 6.4 foram compatíveis com o esperado. Entretanto, ainda ficamos com desempenho taxa×qualidade bastante inferior ao do

G.729, mas conseguimos ultrapassar o desempenho do codificador de voz G.726.

Procurando melhorar o procedimento de aprendizagem do algoritmo MMP, um modelo de predição linear foi inserido ao seu estágio de codificação. Este modelo gera um sinal de resíduos a partir do sinal de voz em análise, que apresenta maior regularidade do que o próprio sinal de voz, mostrando ser capaz de se adequar melhor ao procedimento de aprendizagem do algoritmo MMP. Nesta fase da tese, procuramos identificar a ordem ótima do modelo de predição linear e incorporamos o dicionário de deslocamento. Os experimentos mostram uma melhoria em relação às versões anteriores do MMP, com resultado superior ao dos codificadores G.726 e G.711, porém ainda bem inferior ao valor do G.729. O uso da predição linear gera um conjunto des resíduos regular com uma distribuição centralizada em zero, devido à sua propriedade de alterar a distribuição do sinal. A partir daí, a distribuição do resíduo foi modelada como uma gaussiana generalizada, com a qual foi projetado o quantizador lloyd-max ótimo, tanto para o dicionário inicial quanto para os vetores a serem inseridos no dicionário. Uma pequena melhoria no desempenho taxa×qualidade do algoritmo MMP foi observada. Continuando os estudos no estágio de atualização do dicionário, introduzimos o processo de equalização de norma. Variando o valor de α da norma L^α, observa-se uma variação no resultado final, mas a norma L^1 representou o melhor compromisso taxa×qualidade obtendo um valor PESQ-MOS de 3,118, aproximando-se mais da pontuação de 3,84 alcançada pelo G.729.

Numa outra parte da tese preocupamo-nos com a distribuição estatística do uso das escalas do dicionário. Observações em [4] levaram a concluir que, dado um dicionário em uma escala, a escala de origem dos vetores influencia na sua probabilidade de utilização, e sua exploração poderia melhorar o compromisso taxa×qualidade do algoritmo MMP. Com este conceito, a partição das escalas do dicionário foi implementada, apresentando melhorias em termos de PESQ-MOS de 3,118 para 3,156 para o banco de dados DB1 a uma taxa de 8kbits/s.

Também procuramos investigar alterações no algoritmo MMP e estudar seu comportamento em situações envolvendo mais de uma pessoa falando ao mesmo tempo e mostramos que o MMP tende a ter melhor desempenho que o CS-ACELP no caso de mistura de vozes.

Percebeu-se pelos capítulos anteriores que o problema do processo de aprendizagem do dicionário ainda é a causa do insucesso do algoritmo MMP em atingir o valor esperado. Esta observação levou a incluir no algoritmo MMP um processo de quantização no sinal de resíduos, que levou ao aumento do PESQ-MOS de aproximadamente 0,07. Outro problema identificado consiste na presença de artefatos de codificação que geram ruídos de alta frequência no sinal codificado. Este problema foi resolvido inserindo-se na saída do algoritmo MMP-GAV um filtro passa-baixas, eliminando os ruídos e obtendo um resultado PESQ-MOS de 3,525, muito próximo

do valor do G.729 para a mesma taxa de operação de 8 kbps.

Importantes conclusões podem ser tiradas deste trabalho. Uma delas é que há indícios de que ainda há espaço para melhorias no desempenho do MMP em codificação de voz. Nota-se que as propriedades perceptuais foram apenas preliminarmente exploradas neste trabalho. O simples uso de um filtro passa-baixas na saída do MMP proporcionou um ganho de 0,18 em termos de PESQ-MOS. Temos então uma indicação de que se incorporarmos mais características perceptuais no algoritmo de codificação do MMP, maiores ganhos podem ser obtidos.

8.1 Propostas de continuação

Como visto no Capítulo 7, o resultado do desempenho do algoritmo MMP para codificação de voz apresentou um resultado próximo ao do codificador padrão de voz G.729, operando na mesma taxa de 8 kbps. Procurando melhorar este resultado apresentamos a seguir algumas propostas de investigação para dar continuidade a este trabalho:

- Controle do crescimento do dicionário: A inclusão de blocos deslocados aumenta consideravelmente a cardinalidade de cada escala do dicionário desfavorecendo o crescimento da entropia média dos símbolos indexados; isso compromete o desempenho do codificador, pois teremos elementos que acabam sendo inúteis para aproximação dos padrões do sinal de voz. Uma investigação no controle do crescimento do dicionário resolveria o problema. O controle consiste em evitar a inclusão de blocos redundantes, usando uma esfera de raio d em um espaço l-dimensão, com a qual introduzimos uma distorção mínima entre os vetores existentes em cada escala do dicionário.

- Explorar o contexto de sonoridade na codificação para treinamento do algoritmo MMP: Podem ser identificados trechos sonoros do sinal de voz para que sejam utilizados como elementos do dicionário para que o algoritmo possa aprender mais rapidamente os padrões existentes no sinal de voz, e assim, adaptar-se às características deste sinal. Além disso, esta pré-classificação em trechos sonoros e surdos podem permitir que se usem partições do dicionário ou regras de atualização diferentes para os diferentes casos.

- Critério de similaridade para voz: Aqui faremos com o que o algoritmo MMP busque por similaridade entre o sinal a ser codificado e o seu dicionário, não necessariamente no domínio do tempo, e assim codificar pequenos trechos do sinal de voz, evitando segmentação excessiva da árvore do algoritmo. Alé disso, pode ser o usado o PESQ-MOS ou outra métrica perceptual em vez do

erro médio quadrático no cálculo do custo de codificação $D + \lambda R$. Como outro exemplo, o filtro passa-baixas utilizado na saída pode ser levado em conta quando da escolha dos vetores do dicionário para aproximar um determinado trecho do sinal.

- Incorporar o algoritmo MMP ao padrão G.729 (CS-ACELP): Aqui propomos implementar o algoritmo MMP numa estrutura CELP, para que o dicionário do algoritmo busque o aprendizado das características do sinal de voz da mesma forma que os codificadores da família CELP.

Assim sendo, consideramos que os resultados obtidos até agora, apesar de ainda estarem abaixo dos que podem ser obtidos com o CS-ACELP, indicam que o uso do MMP para a codificação de voz é uma linha de pesquisa que vale a pena ser continuada.

Referências Bibliográficas

[1] DUARTE, M. H. V. *Codificação de Imagens Estéreo Usando Recorrência de Padrões*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, Agosto de 2002.

[2] DE LIMA FILHO, E. B. *Compressão de Imagens Utilizando Recorrência de Padrões Multiescala com Critério de Continuidade Inter-blocos*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, Março de 2004.

[3] DA SILVA JUNIOR, W. S. *Compressão de Imagens Utilizando Recorrência de Padrões Multiescalas com Segmentação Flexível*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, Dezembro de 2004.

[4] PINAGÉ, F. S. *Avaliação de Desempenho de Algoritmos de Compressão de Imagens Usando Recorrência de Padrões Multiescalas*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, Julho de 2005.

[5] DE LIMA FILHO, E. B., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Universal Image Compression Using Multiscale Recurrent Patterns With Adaptive Probability Model", *IEEE Transactions on Image Processing*, v. 17, n. 4, pp. 512–527, 2008. ISSN: 1057-7149. doi: 10.1109/TIP.2008.918042.

[6] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Efficient Dictionary Design for Multiscale Recurrent Pattern Image Coding". In: *IEEE International Symposium on Circuits and Systems*, pp. 4939–4942, Kos, Greece, September 2006.

[7] DE LIMA FILHO, E. B. ., DA SILVA, E. A. B., DE CARVALHO, M. B. "On EMG Signal Compression with Recurrent Patterns", *IEEE Transactions on Biomedical Engineering*, v. 55, n. 7, pp. 1920–1923, 2008. ISSN: 0018-9294. doi: 10.1109/TBME.2008.919729.

[8] DE LIMA FILHO, E. B., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Eletrocardiographic Signal Compression Using Multiscale Re-

current Patterns", *IEEE Transactions on Circuits and Systems I: Regular Papers*, v. 52, n. 12, pp. 2739–2753, 2005. ISSN: 1549-8328. doi: 10.1109/TCSI.2005.857873.

[9] DE LIMA FILHO, E. B., DE CARVALHO, M. B., DA SILVA, E. A. B. "Muldimensional Signal Compression Using Multi-scale Recurrent Patterns With Smooth Side-match Criterion". In: *International Conference on Image Processing*, pp. 3201–3204, Singapore, October 2004.

[10] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "On Dictionary Adaptation for Recurrent Pattern Image Coding", *IEEE Transactions on Image Processing*, v. 17, n. 9, pp. 1640–1653, 2008. ISSN: 1057-7149. doi: 10.1109/TIP.2008.2001392.

[11] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Universal Image Coding Using Multiscale Recurrent Patterns and Prediction". In: *IEEE International Conference on Image Processing*, pp. II–245–8, Genoa, Italy, September 2005.

[12] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Improving H.264/AVC Inter Compression with Recurrent Patterns". In: *IEEE International Conference on Image Processing*, pp. 1353–1356, Atlanta, USA, October 2006.

[13] RODRIGUES, N. M. M., DA SILVA, E. A. B., DE CARVALHO, M. B., et al. "Improving Multiscale Recurrent Pattern Image Coding with Enhanced Dictionary Updating Strategies". In: *International Telecommunications Symposium*, pp. 257–262, Fortaleza, Brazil, September 2006.

[14] DE CARVALHO, M. B., DA SILVA, E. A. B., FINAMORE, W. A., et al. "Universal Multi-scale Matching Pursuits Algoritm with Reduced Blocking Effect". In: *International Conference on Image Processing*, pp. 853–856, Vancouver, Canada, September 2000.

[15] HUFFMAN, D. A. "A Method for the Construction of Minimum Redundancy Codes", *Proceedings of the Institute of Electrical and Radio Engineers*, v. 40, n. 9, pp. 1098–1101, September 1952.

[16] SAYOOD, K. *Introduction to Data Compression*. 2 ed. San Francisco, USA, Morgan Kaufmann Publishers, 2000.

[17] ABRAMSON, N. *Information Theory and Coding*. New York, USA, McGraw-Hill, 1963.

[18] ZIV, J., LEMPEL, A. "A Universal Algorithm for Data Compression", *IEEE Transactions on Information Theory*, v. 43, n. 1, pp. 9–21, 1977. ISSN: 0018-9448. doi: 10.1109/18.567642.

[19] ZIV, J., LEMPEL, A. "A Compression of Individual Sequences Via Variable-Rate Coding", *IEEE Transactions on Information Theory*, v. 24, n. 5, pp. 530–536, 1978. doi: 10.1109/TIT.1978.1055934.

[20] HUANG, J. Y., SCHULTHEISS, P. M. "Block Quantization of Correlated Gaussian Random Variables", *IEEE Transactions on Communication Systems*, v. 11, n. 3, pp. 289–296, 1963. ISSN: 0096-1965. doi: 10.1109/TCOM.1963.1088759.

[21] MACWILLIAMS, F. J., SLOANE, N. *The Theory of Error Correcting Codes*. Amsterdam, Netherlands, North-Holland, 1977.

[22] AHMED, N., NATARAJAN, T., RAO, K. "Discrete Cosine Transfom", *IEEE Transactions on Computers*, v. 23, pp. 90–93, 1974. doi: 10.1109/T-C.1974.223784.

[23] CLARKE, R. J. *Digital Compression of Still Images and Video*. San Diego, USA, Academic Press, 1995.

[24] KOVACEVIC, J., PUSCHEL, M. "Sampling Theorem Associated with the Discrete Cosine Transform". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 357–360, Pittsburgh, USA, May 2006.

[25] TOPIWALA, P. N. *Wavelet Image and Video Compression*. Norwell, USA, Kluwer Academic Publishers, 1998.

[26] VETTERLI, M., KOVACEVIC, J. *Wavelets and Subband Coding*. Englewood Cliffs, USA, Prentice-Hall, 1995.

[27] STRANG, G., NGUYEN, T. *Wavelets and Filter Banks*. Wellesley, USA, Wellesley-Cambrige Press, 1996.

[28] SHANNON, C. E. "A Mathematical Theory of Communication", *Bell Syst. Tech. Journal*, v. 27, 1948.

[29] JAIN, A. K. *Fundamentals of Digital Image Processing*. NJ, USA, Prentice-Hall, 1989.

[30] COVER, T. A., THOMAS, J. A. *Elements of Information Theory*. New York, USA, John Wiley and Sons, 1991.

[31] PAPOULIS, A. *Probability Rondom Variables, and Stochastic Processes*. 2 ed. USA, McGraw-Hill, 1984.

[32] PEEBLES, P. Z. *Probability Random Variables, and Random Signal Principles*. 4 ed. New York, USA, McGraw-Hill, 2001.

[33] DELLER, J. R. J., HANSEN, J. H. L., G.PROAKIS, J. *Discrete-time Processing of Speech Signals*. 2 ed. Piscataway, USA, IEEE Press, 2000.

[34] INTERNATIONAL TELECOMMUNICATIONS UNION. "Methods for Subjective Determination of Transmission Quality. ITU-T Recommendation P.800". In: ITU-T Recommendation P.800, Genebra, Suíça, 1996.

[35] INTERNATIONAL TELECOMMUNICATIONS UNION. "Perceptual Evaluation of Speech Quality (PESQ): An Objective Method for End-to-End Speech Quality Assessment of Narrow-Band Telephone Networks and Speech Codecs. ITU-T Recommendation P.862". In: ITU-T Recommendation P.862, Genebra, Suíça, 2001.

[36] INTERNATIONAL TELECOMMUNICATIONS UNION. "Wideband Extension to Recommendation P.862 for the Assessment of Wideband Telephone Networks and Speech Codecs". In: International Telecommunication Union – Telecommunication Standardization Sector, 2005.

[37] INTERNATIONAL TELECOMMUNICATIONS UNION. "Pulse Code Modulation (PCM) of Voice Frequencies". In: ITU-T Recomendation G.711, Genebra, Suíça, 1983.

[38] INTERNATIONAL TELECOMMUNICATIONS UNION. "Adaptive Differencial Pulse Code Modulation (ADPCM)". In: ITU-T Recomendation G.726, Genebra, Suíça, 1990.

[39] DA SILVA MAIA, R. *Codificação Celp e Análise Espectral de Voz*. Dissertação de M.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, Março de 2000.

[40] SCHROEDER, M., ATAL, B. "Code-Excited Linear Prediction(CELP): High-Quality Speech at Very Low Bit Rates". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, v. 10, pp. 937 – 940, April 1985. doi: 10.1109/ICASSP.1985.1168147.

[41] KROON, P., DEPRETTERE, E. F. "A Class of Analysis-by-Synthesis Predictive Coders for High Quality Speech Coding at Rates Between 4.8 and 16 kbits/s", *IEEE Journal on Selected Areas in Communications*, v. 6, n. 2, pp. 353–363, February 1988.

[42] KROON, P., SWAMINATHAN, K. "A High-Quality Multirate Real-Time CELP coder", *IEEE Journal on Selected Areas in Communications*, v. 10, n. 5, pp. 850–857, June 1992.

[43] KLEIJN, W., KRASINSKI, D., KETCHUM, R. "Improved speech quality and efficient vector quantization in SELP". In: *International Conference on Acoustics, Speech, and Signal Processing*, v. 1, pp. 155 –158, April 1988. doi: 10.1109/ICASSP.1988.196536.

[44] KIM, H. K. "Adaptive Enconding of Fixed Codebook in CELP Coders". In: *IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 149–152, Seattle, USA, May 1998.

[45] KIM, H. K., LEE, M. S., LEE, H. S. "A 4 kbps Adaptive Fixed Code-excited Linear Prediction Speech Coder". In: *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 2303–2306, Phoenix, USA, March 1999.

[46] GERSHO, A. "Advances in Speech and Audio Compression". In: *Proceedings of the IEEE*, pp. 900–918, June 1994.

[47] INTERNATIONAL TELECOMMUNICATIONS UNION. "Low-Delay Code Excited Linear Prediction (LD-CELP)". In: ITU-T Recomendation G.728, Genebra, Suíça, 1992.

[48] INTERNATIONAL TELECOMMUNICATIONS UNION. "Conjugate Structure Algebric Code Excited Linear Prediction". In: ITU-T Recomendation G.729, Genebra, Suíça, 1996.

[49] UNITED STATES DEPARTMENT OF DEFENSE. "Linear Predictive Coding (LPC)". In: US Department of Defense, 1984.

[50] SPANIAS, A. S. "Speech Coding: A Tutorial Review", *Proceedings of the IEEE*, v. 82, n. 10, pp. 1541–1582, October 1994.

[51] RABINER, L. R., SCHAFER, R. W. *Digital Processing of Speech Signals*. Englewood Cliffs, USA, Prentice-Hall, 1978.

[52] CHEN, J. H., COX, R. V., LIN, Y. C., et al. "A Low Delay CELP Coder for the CCITT 16 kb/s Speech Coding Standard", *IEEE Journal on Selected Areas in Communications*, v. 10, n. 5, pp. 830–849, June 1992.

[53] DE CARVALHO, M. B. *Compressão de Sinais Multi-Dimensionais Usando Recorrência de Padrões Multiescalas*. Tese de D.Sc., COPPE/UFRJ, Rio de Janeiro, Brasil, Março de 2001.

[54] DE CARVALHO, M. B., DA SILVA, E. A. B., FINAMORE, W. A. "Multidimensional Signal Compression Using Multiscale Recurrent Patterns", *Elsevier Special Edition in Image Coding Beyond Standards*, v. 82, n. 11, pp. 1559–1580, November 2002.

[55] WITTEN, I., NEAL, R., CLEARY, J. G. "Arithmetic Coding for Data Compression", *Communications of the Association for Computing Machinery*, v. 30, n. 6, pp. 520–540, June 1987.

[56] DENN, M. M. *Optimization by Variational Methods*. New York, USA, McGraw-Hill, 1969.

[57] EVERETT, H. "Generalized Lagrange Multiple Method for Solving Problems of Optimum Allocation Resources", *Operation Research*, v. 11, n. 3, pp. 399–417, May 1963.

[58] SIU-WAI WU, GERSHO, A. "Rate-Constrained Optimal Block-Adaptive Coding for Digital Tape Recording of HDTV", *IEEE Transactions on Circuits and Systems for Video Technology*, v. 1, n. 1, pp. 100–112, 1991. ISSN: 1051-8215. doi: 10.1109/TCSVT.1991.4519809.

[59] SULLIVAN, G. J., BAKER, B. L. "Efficient Quadtree Coding of Images and Video", *IEEE Transactions on Image Processing*, v. 3, n. 3, pp. 327–331, 1994. ISSN: 1057-7149. doi: 10.1109/83.287030.

[60] KIANG, S. Z., BAKER, R. L., SULLIVAN, G. J., et al. "Recursive Optimal Pruning with Applications to Tree Structured Vector Quantizers", *IEEE Transactions on Image Processing*, v. 1, n. 2, pp. 162–169, 1992. ISSN: 1057-7149. doi: 10.1109/83.136593.

[61] RAMIREZ, M. A. *Busca de Inovações e Tratamento de Transitórios em Codificadores de Voz CELP*. Tese de D.Sc., Escola Politécnica, São Paulo, Dezembro de 1997.

[62] TAUBMAN, D. S., MARCELIN, M. W. *JPEG 2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic Publishers, 2001.

[63] PINAGÉ, F. S., FEIO, L. C. R. L., DA SILVA, E. A. B., et al. "Waveform Speech Coding Using Multiscale Recurrent Patterns". In: *IEEE International Symposium on Circuits and Systems*, pp. 3072–3075, 2010.

[64] A. ALCAIM, J. A. S., DE MORAES, J. A. "Phone Occurrence Rates and Lists of Phonetically Balanced Sentences for Brazilian Portuguese Spoken

in Rio de Janeiro", *Revista da Sociedade Brasileira de Telecomunicações*, v. 7, n. 1, pp. 23–41, December 1992.

[65] WOODARD, J., HANZO, L. "Improvements to the Analysis-By-Synthesis Loop in CELP Codecs". In: *Radio Receivers and Associated Systems, 1995., Sixth International Conference on*, pp. 114 –118, September 1995. doi: 10.1049/cp:19951129.

[66] RODRIGUES, N. M. M. *Multiscale Recurrent Pattern Matching Algorithms for Image and Video Coding*. Ph.D., Universidade de Coimbra, Portugal, Março de 2008.

[67] DEVROYE, L. *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.

[68] DOMÍNGUEZ-MOLINA, J. A., GONZÁLEZ-FARÍAS, G., RODRÍGUEZ-DAGNINO, R. M. "A Practical Procedure to Estimate the Shape Parameter in the Generalized Gaussian Distribution". Disponível em: <http://www.cimat.mx/reportes/enlinea/>. Acesso em: Agosto de 2011.

[69] FISCHER, T. "A Pyramid Vector Quantizer", *Information Theory, IEEE Transactions on*, v. 32, n. 4, pp. 568 – 583, July 1986. ISSN: 0018-9448. doi: 10.1109/TIT.1986.1057198.

[70] CHEN, F., GAO, Z., VILLASENOR, J. "Lattice Vector Quantization of Generalized Gaussian Sources", *Information Theory, IEEE Transactions on*, v. 43, n. 1, pp. 92 –103, January 1997. ISSN: 0018-9448. doi: 10.1109/18.567652.

[71] FISCHER, T. "Geometric Source Coding and Vector Quantization", *Information Theory, IEEE Transactions on*, v. 35, n. 1, pp. 137 –145, January 1989. ISSN: 0018-9448. doi: 10.1109/18.42184.

[72] "Open Speech Repository". Disponível em: <http://www.voiptroubleshooter.com/open_speech/>. Acesso em: Setembro de 2011.

Apêndice A

Pseudo-Código

Neste apêndice incluímos uam descrição geral da versão final do algoritmo MMP sumarizada na Tabela A.1.

Tabela A.1: Parâmetros do algoritmo MMP.

Parâmetros	Quantidade/tamanho
dicionário inicial	256 elementos
escalas do dicionário	5
ordem do modelo LP	40
amostras para predição	128
tamanho do bloco predito	16
janela de deslocamento	256
passo de deslocamento	1 amostra
equalização da nomra L^{α}	$\alpha = 1$

Durante o processo de otimização do algoritmo são criadas variáveis para armazenar a frequência de utilização dos índices e dos *flags* de segmentação, ajustando seus modelos de frequência no formato de árvore de segmentação. Estas variáveis são apresentadas nas Tabelas A.2 e A.3.

Tabela A.2: Contadores para totalizar a frequência de utilização dos índices.

Sigla	Comentário
fi	Armazena as frequências dos índices do dicionário.
fio	Frequência espelho dos índices do dicionário.
fir	Armazena as frequências dos índices do dicionário rascunho.
fg	Armazena as frequências dos índices da partição do dicionário.
fgr	Armazena as frequências dos índices da partição do dicionário rascunho.

Tabela A.3: Contadores para totalizar a frequência dos *flags* de segmentação.

Sigla	Comentário
$fflag$	Armazena as frequências para os *flags* de segmentação (0 e 1).
$fflago$	Complementa a frequência dos *flags* de segmentação.

A.1 Inicialização do Dicionário

O dicionário D inicial do algoritmo MMP contém $n = 5$ escalas (K^j) de diferentes dimensões 2^j, para $j = 0, \ldots, (n-1)$. Cada uma das escalas é dividida em 5 partições c_{K^j} conforme a escala de origem (vide Capítulo 7). O processo de inicialização do dicionário é realizado de acordo com o algoritmo abaixo.

Passo 1: Inicialize a escala K^0 (dimensão 1) do dicionário D usando o comando `lloyd2` do MATLAB com 256 elementos. Insira os elementos na partição correspondente à escala de origem.

Passo 2: Inicialize as demais escalas K^j do dicionário D, expandindo cada elemento da escala K^0 para a dimensão 2^j, através da transformação de escala vista na Seção 4.1, e insira na partição correspondente K^j.

A.2 Predição

Passo 1: Encontre os coeficientes do modelo LP de ordem $N = 40$, usando as 128 amostras previamentes codificadas do sinal.

Passo 2: Realize a predição $Pred^l$ das 2^{n-1} amostras do bloco X^l usando o modelo LP encontrado no **Passo 1**.

Passo 3: Para predição $Pred^l$ encontrada, obtenha o sinal de resíduo $R^l = X^l - Pred^l$.

Passo 4: Quantize o sinal de resíduo R^l para o mesmo nível de quantização do dicionário inicial.

A.3 Procedimento de Otimização

$$\{\hat{R}^l, A(n_0)\} = OtimizaRDlp(M, R^l, no, \hat{A}(n_0))$$

Passo 1: Faz $A(n_0) = \hat{A}(n_0)$, onde $\hat{A}(n_0)$ é a árvore de segmentação com todos possíveis nós folhas.

Passo 2: Encontre um índice i do elemento k_i dentro das escalas $K^j \in D$ com dimensão 2^j (mesma de R^l), que representa R^l com menor custo $J_{n_l} = D_{n_l} + \lambda RG_{n_l} + \lambda RI_{n_l}$, onde

- J_{n_l} é o custo para o nó n_l;

- D_{n_l} é a distorção entre o bloco R^l associado ao nó n_l e a sua aproximação k_i da escala K^j com a mesma dimensão de R^l. Esta distorção é dada por

$$D_{n_l} = \sum_{i=1}^{M} (R^l - k_i)^2; \tag{A.1}$$

- λ é o fator ponderador entra a taxa e a distorção;
- RG_{n_l} é a taxa para representar a partição do dicionário;
- RI_{n_l} é a taxa para representar o índice. Armazene i e faça $\hat{R}^l = k_i$ e o bloco reconstruído $\hat{X}^l = \hat{R}^l + Pred^l$. O custo é dado por:

$$J_{n_l} = (R^l - k_i)^2 + \log_2 \frac{\sum fg}{fg} + \log_2 \frac{\sum fi + \sum fio + \sum fir}{fi + fio} \tag{A.2}$$

Passo 3: Procurar no dicionário D_R (dicionário Rascunho) o elemento k_{i_R} que melhor representa R^l e verifica se o custo J_{n_l} é menor que o escolhido no **Passo 2**. Se isso acontecer, substituir k_i por k_{i_R}, fazendo $\hat{R}^l = k_{i_R}$ e o bloco reconstruído igual a $\hat{X}^l = \hat{R}^l + Pred^l$, armazenando i_R. O custo é calculado conforme abaixo:

$$J_{n_l} = (R^l - k_i)^2 + \log_2 \frac{\sum fgr}{fgr} + \log_2 \frac{\sum fi + \sum fio + \sum fir}{fir} \tag{A.3}$$

Passo 4: Se a dimensão do bloco R^l fôr 1, incremente fio se o elemento escolhido para a aproximação foi feita por D ou incremente fir se a aproximação foi feita por D_R. Caso a dimensão do bloco seja maior que 1, vá para o Passo 5.

Passo 5: Acrescente, ao custo J_{n_l} calculado, o valor $\lambda Rf1$, conforme abaixo. Este é o valor da taxa para o *flag* de segmentação igual a "1" e representa completamente o custo para um nó folha.

$$\lambda Rf1 = \log_2 \frac{\sum fflag(1) + \sum fflago(1)}{fflag(1) + fflago(1)}.$$

Passo 6: Calcule e armazene, o valor da taxa para o *flag* de segmentação igual a "0", $\lambda Rf0$, que é calculada conforme abaixo

$$\lambda Rf0 = \log_2 \frac{\sum fflag(0) + \sum fflago(0)}{fflag(0) + fflago(0)}.$$

Passo 7: Incremente o *flag* "0" com a função fflago(o).

Passo 8: Se fôr possível dividir o bloco R^l, divida-o em dois blocos, $\left(R^{2l+1} R^{2l+2}\right)$, mudando a dimensão para $M/2$.

Passo 9: Compute $\{\hat{R}^{2l+1}, A(n_0)_a\} = OtimizaRDlp(M, R^{2l+1}, 2no+1, A(n_0))$.

Passo 10: Compute $\{\hat{R}^{2l+2}, A(n_0)_b\} = OtimizaRDlp(M, R^{2l+2}, 2no+2, A(n_0))$.

Passo 11: Faz $A(n_0) = A(n_0)_a$ e $A(n_0)_b$.

Passo 12: Se o $J_{no} \leq J_{2no+1} + J_{2no+2} + \lambda Rf0(no)$, então vá para o **Passo 13**. Caso contrário, vá para o **Passo 17**.

Passo 13: Decremente:

- os contadores $fflago(0)$ (*flag* "0") e $fflago(1)$ (*flag* "1") para as dimensões relacionadas às árvores $A(n_{2no+1})$ e $A(n_{2no+2})$ e para a dimensão M atual;
- os índices usados pelos nós que serão podados: se eles são do dicionáo D então decremente fio, caso sejam do dicionário rascunho D_R decremente fir.

Passo 14: Incremente:

- o $fflago(1)$ (*flag* "1") para a dimensão atual;
- o índice usado pelos nó da dimensão atual: se ele pertence ao dicionáo D então incremente fio, caso pertença ao dicionário rascunho D_R incremente fir.

Passo 15: Elimine as atualizações do dicionário rascunho D_R que foram ocasionadas pelas árvores $A(n_{2no+1})$ e $A(n_{2no+2})$ a serem podadas.

Passo 16: Informe em $A(n_0)$ que as árvores $A(n_{2no+1})$ e $A(n_{2no+2})$ foram eliminadas e retorne $\left[R^l, A(n_0)\right]$.

Passo 17: Realize a concatenação dos blocos $\hat{R}^{2l+1}$ e $\hat{R}^{2l+2}$, $\hat{R}^l = \left(\hat{R}^{2l+1}\hat{R}^{2l+2}\right)$.

Passo 18: Atualize o dicionário D_R em todas as escalas com $\hat{R}^l$ conforme procedimento *atualizadic*.

Passo 19: Aualize o custo do nó $J_{no} \leq J_{2no+1} + J_{2no+2} + \lambda Rf0(no)$.

Passo 20: Retorne $\left[\hat{R}^l, A(n_0)\right]$.

A.4 Procedimento de Codificação

$$\{\hat{R}^l\} = Codificalp(M, R^l, no, A(n_0))$$

Passo 1: Se $A(n_{2no+1}) == 0$ e $A(n_{2no+2}) == 0$ ou $M == 1$, então vá para o **Passo 2**. Caso contrário, siga para o **Passo 7**.

Passo 2: Encontre um índice i do elemento k_i dentro das escalas $K^j \in D$ com dimensão 2^j (mesma de R^l), que representa R^l com menor custo $J_{n_l} = D_{n_l} + \lambda RG_{n_l} + \lambda RI_{n_l}$. Armazene i e faça $\hat{R}^l = k_i$.

Passo 3: Realize o deslocamento, até um comprimento L, de um elemento d_i (elemento do dicionário de deslocamento D_d) de dimensão 2^j (mesma de R^l), sobre o sinal recentemente codificado, deslocando de um passo de variação $\delta = 1$, até encontrar o elemento que represente R^l com menor custo, $J_{n_l} = (R^l - d_i)^2 + \lambda Rd_{n_l}$. Verifique se o custo é menor que o escolhido no **Passo 2**. Se isso ocorrer, substitua k_i por d_i. Faça $R^l = d_i$ e $\hat{X}^l = \hat{R}^l + Pred^l$, e armazene δ.

Passo 4: Se $M == 1$ e o menor custo foi para o dicionário D_d, codifique δ e retorne o bloco aproximação $\hat{R}^l$. Se M não fôr igual 1, siga para o **Passo 6**.

Passo 4: Se $M == 1$ e e o menor custo foi para o dicionário D, então codifique o índice i do elemento k_i da partição c_{k^j} e retorne o bloco aproximação $\hat{R}^l$. Se M não fôr igual 1, siga para o **Passo 6**.

Passo 6: Se o elemento fôr do dicionário D, codifique um *flag* igual a "1", a partição c_{k^j}, o índice i e retorne o bloco $\hat{R}^l$. Senão, se o elemento fôr do dicionário D_d, codifique δ e retorne o bloco $\hat{R}^l$.

Passo 7: Codifique o *flag* "1".

Passo 8: Divida o bloco R^l, em dois blocos, $\left(R^{2l+1}R^{2l+2}\right)$, mudando sua dimensão para $M/2$, e:

- Compute $\{\hat{R}^{2l+1}\} = Codificalp(M/2, R^{2l+1}, 2no+1, A(n_0))$;
- Compute $\{\hat{R}^{2l+2}\} = Codificalp(M/2, R^{2l+2}, 2no+2, A(n_0))$.

Passo 9: Realize a concatenação dos blocos $\hat{R}^{2l+1}$ e $\hat{R}^{2l+2}$, $\hat{R}^l = \left(\hat{R}^{2l+1}\hat{R}^{2l+2}\right)$.

Passo 10: Atualize o dicionário D_R em todas as escalas com $\hat{R}^l$ conforme procedimento *atualizadic*.

Passo 11: Retorne $\hat{R}^l$.

A.5 Procedimento de Atualização do dicionário

$$\{Y^j\} = AtualizaDic(\hat{R}^l)$$

Passo 1: Transforme o bloco $\hat{R}^l$ para um bloco Y^j para cada escala K^j na partição c_{k^j} do dicionário D, usando a transformação de escala apresentada na Seção 4.1.

Passo 2: Aplique a equalização de norma L^1 (vide Seção 7.6) no bloco Y^j.

Passo 3: Quantize o bloco Y^j para o mesmo nível do dicionário D.

Passo 3: Verifique se o bloco Y^j já existe no dicionário D. Se não existir, inclua o bloco Y^j K^j na partição c_{k^j} da escala K^j do dicionário D.

A.6 Procedimento de Decodificação

$$\{X^l\} = Decodifica(M, Pred^l)$$

Passo 1: Inicialize a escala K^0 (dimensão 1) do dicionário D usando o comando `lloyd2` do MATLAB com 256 elementos. Insira os elementos na partição correspondente à escala de origem.

Passo 2: Inicialize as demais escalas K^j do dicionário D, expandindo cada elemento da escala K^0 para a dimensão 2^j através da transformação de escala vista na Seção 4.1 e insira na partição correspondente a K^j.

Passo 3: Encontre os coeficientes do modelo LP de ordem $N = 40$, usando as 128 amostras previamentes codificadas.

Passo 4: Realize a predição $Pred^l$ das 2^{n-1} amostras do bloco $\hat{X}^l$ usando o modelo LP encontrado no **Passo 3**.

Passo 5: Se $M == 1$ leia uma *flag* e armazene em *xflag*. Se $xflag == 1$ leia uma partição c_{k^j} e o índice i. Acesse o dicionário D na partição c_{k^j} no índice i de dimensão 1, faça $\hat{X}^l = k_i + Pred^l$ e retorne $\hat{X}^l$. Se $xflag == 2$ leia um deslocamento δ. Desloque o segmento de dimensão 1 de δ sobre as amostras previamentes codificadas. Faça $\hat{X}^l = d_i + Pred^l$ e retorne $\hat{X}^l$.

Passo 6: Leia um *flag* e armazene em *xflag*.

Passo 7: Se $xflag == 1$ leia uma partição c_{k^j} e o índice i. Acesse o dicionário D na partição c_{k^j} no índice i de dimensão 1, faça $\hat{X}^l = k_i + Pred^l$ e retorne $\hat{X}^l$. Senão, se $xflag == 2$ leia um deslocamento δ. Desloque o segmento de dimensão 1 de δ sobre as amostras previamentes codificadas. Faça $\hat{X}^l = d_i + Pred^l$ e retorne $\hat{X}^l$. Se não siga para o **Passo 8**.

Passo 8: Faça

- $\{X^{2l+1}\} = Decodifica(M/2, Pred^l)$;
- $\{X^{2l+2}\} = Decodifica(M/2, Pred^l)$.

Passo 9: Realize a concatenação dos blocos $\hat{R}^{2l+1}$ e $\hat{R}^{2l+2}$, $\hat{R}^l = \left(\hat{R}^{2l+1}\hat{R}^{2l+2}\right)$.

Passo 10: Atualize o dicionário D_R em todas as escalas com $\hat{R}^l$ conforme procedimento *atualizadic*.

Passo 11: Aplique o filtro de redução de efeito blocagem (vide Seção 5.6).