

4

Supressão de Ruído Baseada em Wavelets

Com o crescente número de aparelhos celulares digitais e computadores portáteis, as técnicas de supressão de ruído em aplicações de processamento de voz tiveram um considerável aumento de importância. Além da expectativa dos consumidores de um alto desempenho mesmo em ambientes ruidosos, a supressão de ruído é importante devido à necessidade de se mover progressivamente em direção a algoritmos de codificação de voz a taxas mais baixas a fim de acomodar um número crescente de usuários de aparelhos celulares e de transmissão de voz em redes IP. Nota-se que enquanto os algoritmos de codificação de voz a taxas mais altas conseguem manter bons desempenhos mesmo em condições com alto nível de ruído, isto não ocorre com algoritmos a taxas mais baixas, que têm seu desempenho bastante afetado em ambientes muito ruidosos.

Nas últimas décadas, as transformadas *wavelets* têm sido aplicada a um número cada vez maior de problemas ligados direta ou indiretamente à área de processamento de sinais. Das várias aplicações pode-se citar a supressão de ruído (*denoising*) em sinais, bem como compressão, detecção e reconhecimento de padrões [29]. Vale destacar que à medida que a teoria vem sendo difundida e compreendida, novas aplicações vêm surgindo nas mais diversas áreas.

Neste capítulo, será apresentada uma breve introdução à teoria de transformadas *wavelets* e sua aplicação em supressão de ruído. No Capítulo 3, vimos que o QVPC4 obteve melhor desempenho em todos os critérios de avaliação adotados. Aqui, será analisado o desempenho deste esquema de quantização utilizando vozes ruidosas e vozes que utilizaram *Wavelet Denoising* antes da codificação. O objetivo é verificar a eficiência da aplicação da técnica de *wavelet* na supressão de ruído antes da codificação sobre a qualidade da voz sintetizada pelo codec. A análise da qualidade da voz codificada será feita utilizando a avaliação PESQ e teste de comparação A/B.

4.1 Uma Introdução à Teoria de Wavelets

O conceito de *wavelets* na sua forma teórica atual foi proposto pela primeira vez por Jean Morlet e pela equipe de pesquisadores de Alex Grossman trabalhando no Centro de Física Teórica de Marseille na França [30].

A capacidade de analisar sinais com espectro variante no tempo é um dos grandes atrativos da teoria de wavelets. Geralmente, os sinais são estudados como função do tempo, ou como função da frequência. Mas, a maioria dos sinais encontrados na prática apresentam espectros variantes no tempo. Encontramos poucos sinais na natureza cujo conteúdo frequencial não mude ao longo do tempo. Em muitas aplicações práticas, é de grande utilidade caracterizar o sinal tanto no domínio temporal, quanto no frequencial, como no caso de processamento de sinais de voz [31].

Uma *wavelet* pode ser definida como uma forma de onda de duração limitada e com um valor médio igual a zero. Podemos comparar uma *wavelet* a uma senóide, que é a base da análise de Fourier, embora as senóides sejam suaves e previsíveis e as *wavelets* tendem a ser irregulares e assimétricas. Podemos visualizar essas diferenças na Figura 4.1.

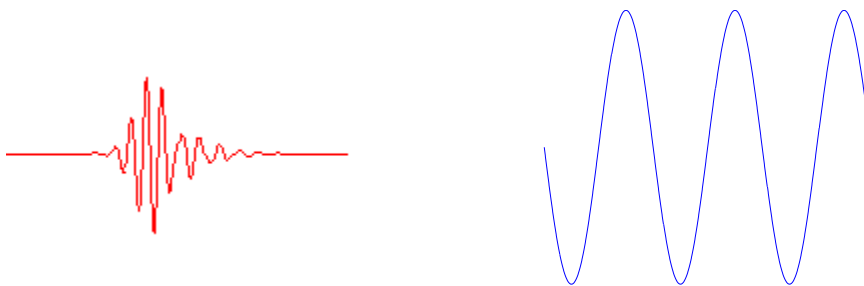


Figura 4.1: Comparação entre uma wavelet (db10) e uma senóide

Pode-se fazer uma analogia da análise por *wavelets* com a análise de Fourier. Na análise de Fourier, decompõe-se um sinal em ondas senoidais de várias frequências. Já no caso da análise por *wavelets* o sinal é decomposto em versões "deslocadas" e "escaloadas" da *wavelet* original.

O tratamento analítico para análise por *wavelets* inclui a transformada contínua de *wavelets*, a transformada discreta e suas respectivas transformadas inversas. Como a transformada contínua traz grande redundância de informação, geralmente é utilizada a transformada discreta [30].

A transformada contínua de *wavelets* (CWT - *Continuous Wavelet Transform*) é definida como a soma sobre todo domínio temporal do sinal multiplicado por versões escalonadas e deslocadas de uma função *wavelets* ψ :

$$C(a, b) = \int_{-\infty}^{\infty} f(t)\psi(a, b, t)dt \quad (4-1)$$

onde a e b são os fatores de escala e deslocamento, respectivamente, e C é o coeficiente *wavelet* em função da escala e da posição.

Simplificadamente, escalonar uma *wavelet* significa alongá-la ou comprimí-la e deslocar uma *wavelet* significa acrescentar um retardo. Sendo assim, a CWT é a soma sobre todo o domínio temporal da multiplicação do sinal por versões escalonadas e deslocadas de uma *wavelet* apropriadamente escolhida.

O termo "transformada discreta de *wavelet*" (DWT - *Discrete Wavelet Transform*) se refere ao algoritmo de Mallat em 1988 [32]. Esse esquema opera com escalas e posições baseadas em potência de 2 (as chamadas escalas e posições "diádicas"). Pode-se fazer uma analogia entre o algoritmo de decomposição diádico e o uso de um banco de filtros, conceito vindo da teoria geral de processamento de sinais.

A DWT divide o sinal analisado em componentes de alta escala e baixa escala. As componentes de alta escala são chamadas de "aproximações", que correspondem ao conteúdo de baixa frequência do sinal. Já as variações rápidas do sinal são chamadas de "detalhes". Com isso, o resultado de uma DWT apresentaria duas vezes mais dados que o sinal original, o que seria muito ruim em termos de eficiência computacional. Para compensar esse problema, a decomposição do sinal é realizada reduzindo-se a taxa de amostragem na mesma proporção que o número de decomposições.

O processo de decomposição da DWT pode ser repetido, atuando sempre sobre as componentes de aproximações. Sucessivas decomposições geram a chamada "árvore de decomposição *wavelet*", como podemos ver na Figura 4.2, onde A_1, A_2, A_3 são as componentes de aproximação e D_1, D_2, D_3 são as componentes de detalhe para 3 níveis. O limite máximo de decomposições corresponde ao nível em que a componente de detalhes consiste de apenas uma amostra. Mas o limite prático costuma ser selecionado com base na natureza do sinal, ou em um critério mais específico como o da entropia.

Na forma de um banco de filtros, a DWT implementa a "análise de multi-resolução" (MRA - *Multi-resolution Analysis*). A MRA permite

analisar um sinal em vários níveis de resolução.

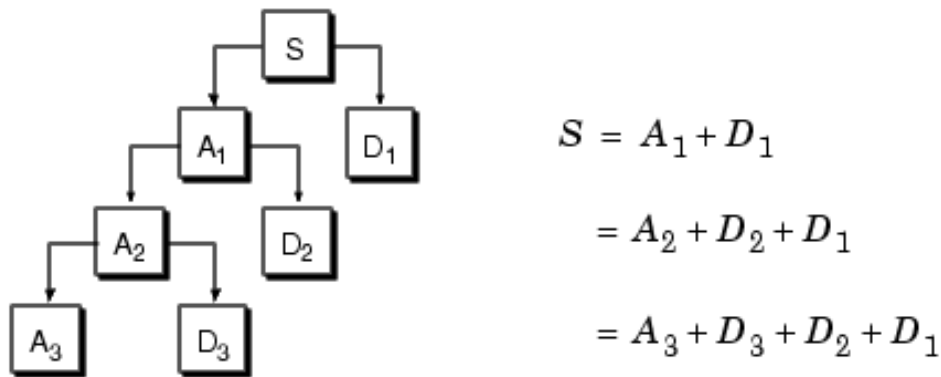


Figura 4.2: Exemplo de árvore de decomposição *wavelet* (DWT de 3 níveis de um sinal).

Uma generalização da DWT que oferece uma gama mais rica de possibilidades é o método por pacotes *wavelet*. Nesse caso os detalhes também são decompostos e com isso tem-se a chamada "árvore de decomposição por pacotes *wavelet*", como ilustrado na Figura 4.3. A análise por pacotes *wavelet* permite, por exemplo, que o sinal S seja representado como as somas das componentes $A_1 + AAD_3 + DAD_3 + DD_2$. Essa representação não seria possível com a DWT.

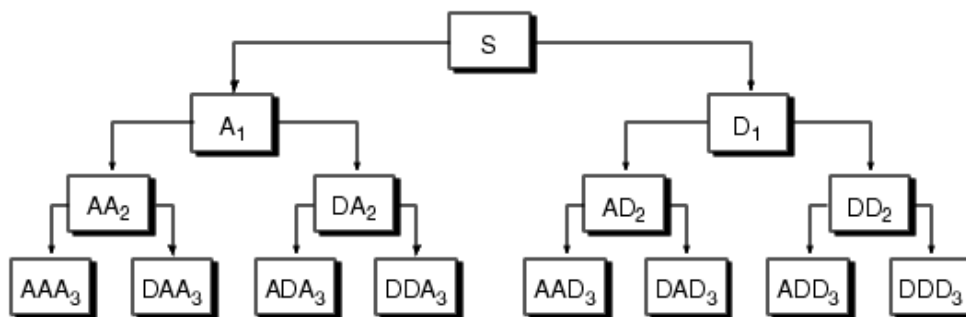


Figura 4.3: Árvore de decomposição por pacotes *wavelet*.

4.2

Supressão de Ruído Usando Wavelet

A idéia básica utilizada na supressão de ruído utilizando transformada *wavelet* é escolher quais coeficientes serão mantidos para preservar

a informação do sinal, removendo assim os coeficientes associados à contribuição do ruído.

Existem duas propriedades das transformadas *wavelet* que torna possível a supressão de ruído nos sinais. A primeira é que apenas alguns poucos coeficientes de decomposição serão não-nulos se as funções-base forem selecionadas adequadamente de acordo com as características do sinal analisado; assim se consegue uma alta concentração de energia nesses poucos coeficientes. A outra propriedade é que, se o sinal apresenta distribuição gaussiana, os coeficientes *wavelet* também apresentarão tal distribuição. Neste sentido, a transformada *wavelet* de ruído gaussiano branco é ruído gaussiano branco e a energia total está espalhada por todos os coeficientes. Dessas propriedades, observa-se que os coeficientes da transformada *wavelet* de um sinal terão amplitude comparativamente superior aos coeficientes da transformada do ruído, esta diferença de amplitude torna possível uma operação de filtragem onde as componentes espectrais do sinal e do ruído podem estar superpostas em tempo e frequência, o que não é possível com métodos baseados na transformada de Fourier [33].

Uma abordagem genérica para resolver o problema da supressão de ruído foi proposta originalmente por Donoho e Johnstone [33][34]. Ela consiste em eliminar os coeficientes menores que um certo limiar (*thresholding*), estabelecido de acordo com algum critério. O algoritmo genérico proposto por Donoho para realizar supressão de ruído, pode ser resumido nos seguintes passos: aplicar algum algoritmo *wavelet* rápido aos dados de entrada; aplicar uma função de *thresholding* (limiar) aos coeficientes de detalhe da transformada, usando um limiar especialmente estimado; e, por fim, inverter o algoritmo de *wavelet* para compor o sinal no domínio do tempo.

O procedimento genérico de supressão de ruído depende principalmente da definição do limiar. Algumas técnicas de definição do limiar bastante difundidas na literatura são descritas a seguir, assim como as principais maneiras de se efetuar a eliminação dos coeficientes.

4.2.1 Funções de Limiar

Na supressão de ruído, normalmente utilizam-se dois tipos de funções de limiar. O mais simples é o chamado limiar "abrupto" (*Hard-Thresholding*), que simplesmente substitui os coeficientes menores que o

limiar por zero. No *Hard-Thresholding* os coeficientes de detalhe da transformada do sinal limpo, β_{jk} , são estimados por

$$\hat{\beta}_{jk} = \begin{cases} Z_{jk}, & |Z_{jk}| \leq t \\ 0, & \text{caso contrário.} \end{cases} \quad (4-2)$$

onde Z_{jk} é o coeficiente de detalhe da transformada *wavelet* na banda k e nível j . O outro tipo utilizado é o limiar "suave" (*Soft-Thresholding*) que apresenta propriedades matemáticas mais interessantes pois "encolhe" os coeficientes para evitar descontinuidades. Já o *Hard-Thresholding*, devido às descontinuidades, gera estimativas de maior variância que podem causar instabilidade [35]. Por essa razão, nesta dissertação é usado o *Soft-Thresholding*. No *Soft-Thresholding* os coeficientes de detalhe da transformada do sinal limpo, são estimados por

$$\hat{\beta}_{jk} = \begin{cases} \text{sign}(Z_{jk})(|Z_{jk}| - t), & |Z_{jk}| \leq t \\ 0, & \text{caso contrário.} \end{cases} \quad (4-3)$$

A Figura 4.4 mostra as duas funções de limiar citadas.

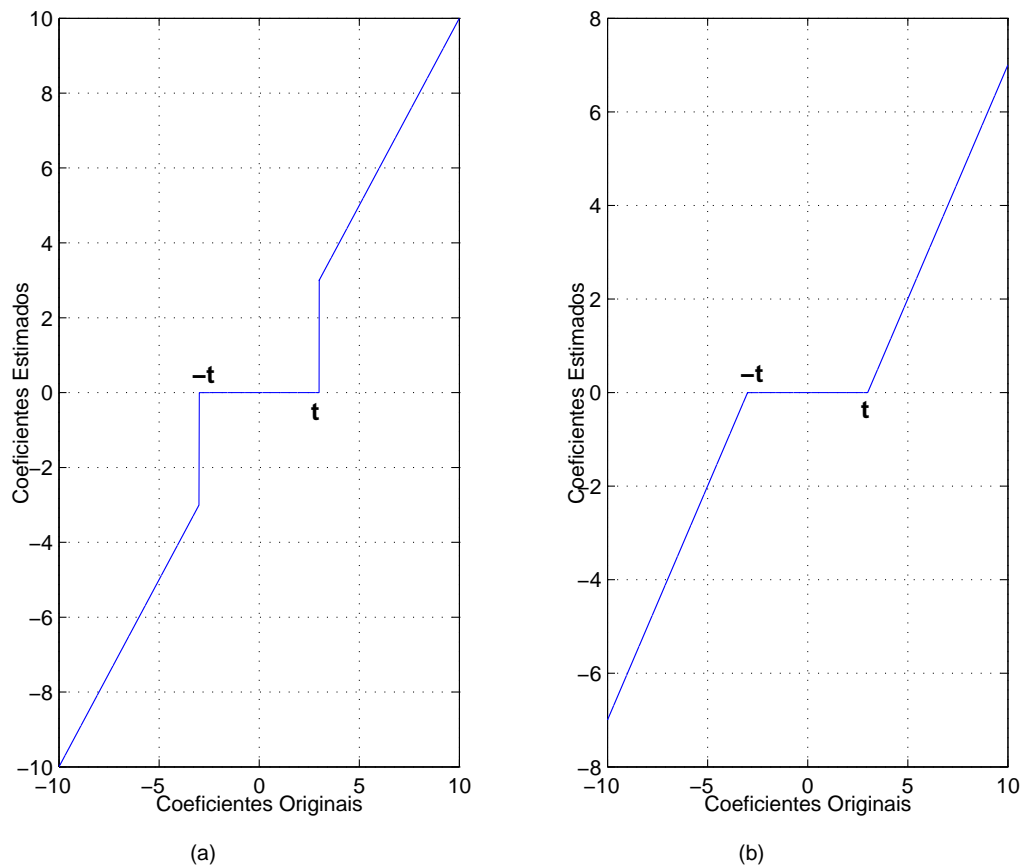


Figura 4.4: Funções de Limiar: (a) *Hard-Thresholding* e (b) *Soft-Thresholding*.

4.2.2

O Cálculo do Limiar

Antes de apresentar os principais métodos para cálculo do limiar, é interessante salientar que dependendo do tipo de ruído que queremos suprimir, podemos usar um limiar dependente do nível ou um limiar independente do nível.

O limiar independente do nível é calculado da observação dos coeficientes no nível de maior resolução da transformada e é geralmente usado para remover ruído branco, pois sua energia encontra-se espalhada de maneira uniforme nos coeficientes de todos os níveis da transformada. Se a base da transformada for apropriadamente escolhida para um dado sinal, sua energia está distribuída, sendo que a sua maior concentração está nos coeficientes dos primeiros níveis da transformada. Sendo assim, pode-se esperar que no último nível, o de maior resolução, tenha-se principalmente coeficientes relacionados ao ruído e não ao sinal; por esse motivo o limiar é calculado no nível de maior resolução e mantido constante para os outros.

Quando temos presença de ruído colorido, é comum o uso do limiar dependente do nível. Como para esse tipo de ruído a energia no domínio da transformada é espalhada de maneira não-uniforme, torna-se necessário o cálculo do limiar para cada nível. A Figura 4.5 apresenta a diferença entre a transformada *wavelet* de 5 níveis para ruído branco e ruído colorido. Os limites entre os níveis da transformada são indicados pelas linhas verticais. Pode-se observar dessa figura, a diferença entre a forma de espalhamento da energia para os dois tipos de ruído.

Donoho apresenta em seu trabalho [36] algumas regras para seleção do limiar, entre elas a mais popular é chamada "regra universal" ou *VisuShrink*. O termo *shrink* se refere ao efeito de encolhimento provocado pelo uso de limiar suave em seu trabalho.

O método *VisuShrink* foi desenvolvido como a solução de um problema de otimização, no qual um critério *minimax* submetido a restrições foi adotado. No caso de supressão de ruído, a função de regressão desconhecida a ser estimada é o sinal sem ruído. O estimador *minimax* prevê um resultado que corresponde ao menor erro médio quadrático máximo de um dado conjunto de funções. Este conjunto de funções é submetido à restrição de, com alta probabilidade, o sinal estimado ter pelo menos a mesma suavidade do sinal original [36]. A motivação para esta restrição foi a necessidade de uma melhor relação de compromisso entre variância e polarização da estimativa. Em subtração espectral, só minimizamos a função custo, normalmente o erro médio quadrático, que resulta em um

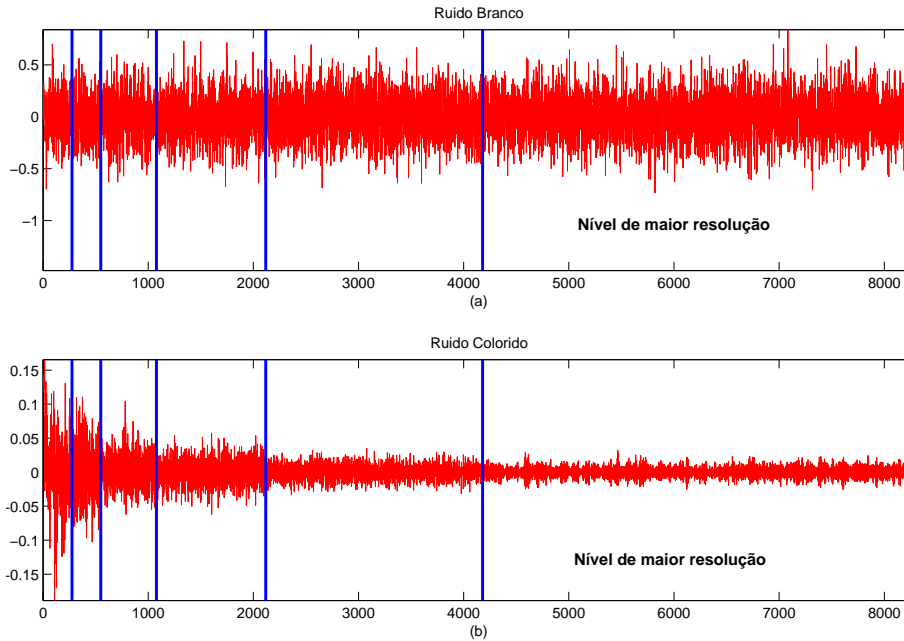


Figura 4.5: Transformada wavelet de 5 níveis para (a) Ruído Branco e (b) Ruído Colorido.

compromisso entre polarização e a variância na mesma ordem de amplitude. O resultado deste processo de minimização é ótimo do ponto de vista do erro médio quadrático mas gera o aparecimento de ruído residual como o ruído musical, ondulações (*ripples*), manchas (*blips*) e oscilações [36]. O *VisuShrink* ao incorporar na solução esta restrição, livra-se dos ruídos espúrios.

A regra de seleção *VisuShrink* é função da largura amostral n do sinal original, que é também o número de coeficientes computado para uma transformada *wavelet* ortogonal. Para um sinal $x(t)$ corrompido por ruído, o limiar *VisuShrink* é dado por [33]:

$$\hat{t} = \hat{\sigma} \sqrt{2 \log_{10} n} \quad (4-4)$$

onde n é o número de coeficientes, $\hat{\sigma}$ é uma estimativa grosseira ($\hat{\sigma} = m/0,6745$) do nível de ruído presente e m é chamado de desvio da mediana do valor absoluto, calculado como a mediana do valor absoluto dos coeficientes da transformada. No caso do limiar ser independente do nível, m é calculado a partir dos coeficientes do nível de maior resolução.

A regra de seleção *VisuShrink*, usualmente suaviza de forma muito acentuada as estimativas [37]. Com a finalidade de evitar essa possibilidade, pode ser usado no lugar da *VisuShrink*, uma outra classe de regras que

computa o limiar de uma forma adaptativa aos dados . Uma dessas regras utiliza o estimador de risco não-polarizado chamado *Stein's Unbiased Risk Estimator*, que minimiza uma função de risco para um valor de limiar particular; ou seja, para um dado conjunto de coeficientes, o limiar é aquele que apresenta menor risco [38]. Essa regra de seleção é conhecida como *SureShrink*.

No *SureShrink*, define-se a função risco do estimador de β_{jk} como o erro médio quadrático,

$$\mathcal{R}_j(\hat{s}, s) = \sum_{k=1}^N E[(\hat{\beta}_{jk} - \beta_{jk})^2] \quad (4-5)$$

onde $\hat{\beta}_{jk} = Z_{jk} + H_t(Z_{jk})$ é o estimador, $H_t[\cdot]$ é uma função real diferenciável para qualquer valor de t_j e Z_{jk} é o k -ésimo coeficiente de detalhe da transformada wavelet de um sinal com ruído, no nível j . O problema aqui, é escolher um t_j ideal que minimiza a função $\mathcal{R}_j(\hat{s}, s)$. Como na prática não conhecemos β_{jk} , só poderemos escolher um \hat{t}_j próximo de t_j que minimiza um estimador não polarizado, $\hat{\mathcal{R}}_j(\hat{s}, s)$, do risco $\mathcal{R}_j(\hat{s}, s)$.

A equação 4-5 pode ser expressa como $\mathcal{R}_j(\hat{s}, s) = E[\hat{\mathcal{R}}_j(\hat{s}, s)]$ [38], onde,

$$\hat{\mathcal{R}}_j(\hat{s}, s) = \sum_{k=1}^N R_j(\sigma_j, Z_{jk}, t_j) \quad (4-6)$$

é o estimador de risco (*Stein's Unbiased Risk Estimator*). O princípio de Stein é minimizar $\hat{\mathcal{R}}_j(\hat{s}, s)$ com respeito a t_j e usar aquele minimizador como um estimador do limiar. Portanto, o estimador do limiar é dado por:

$$\hat{t}_j = \min_{t \geq 0} \sum_{k=1}^N R_j(\sigma_j, Z_{jk}, t) \quad (4-7)$$

Este estimador depende da função de limiar. Para *soft-thresholding* (função de limiar usada nas simulações dessa Dissertação), a equação 4-7, pode ser expressa como [38]:

$$\hat{t}_j = \min_{t \geq 0} \sum_{k=1}^N (2\sigma_j^2 + t^2 - Z_{jk}^2) I\{|Z_{jk}| \geq t\}. \quad (4-8)$$

onde σ_j é estimado por $\hat{\sigma}_j = m_j/0,6745$, sendo que m é o desvio da mediana do valor absoluto calculado no nível j da transformada wavelet, e $I\{\cdot\}$ é uma

função indicadora e definida como:

$$I\{x\} = \begin{cases} 1, & x \text{ verdadeiro;} \\ 0, & x \text{ falso.} \end{cases} \quad (4-9)$$

4.3

Qualidade da Voz Utilizando Wavelet Denoising Antes da Codificação

O objetivo dessa seção é verificar se a utilização de *Wavelet Denoising* nas vozes ruidosas, antes destas serem processadas pelo codec, melhora a qualidade da voz codificada. O diagrama ilustrativo do sistema considerado nessa seção pode ser visto na Figura 4.6. Para testar a qualidade da voz, utilizaremos dois métodos: a medida PESQ e testes subjetivos de comparação A/B. Não utilizou-se a avaliação em termos de DE e percentual de *outliers*, pois optou-se por uma análise subjetiva de qualidade, a fim de compará-la com os resultados encontrados com o PESQ. As simulações foram feitas utilizando o codec e as mesmas condições de rede IP descritas no Capítulo 3. Para quantização das LSFs utilizou-se o QVPC4, por ter apresentado o melhor desempenho em relação aos outros dois esquemas testados.

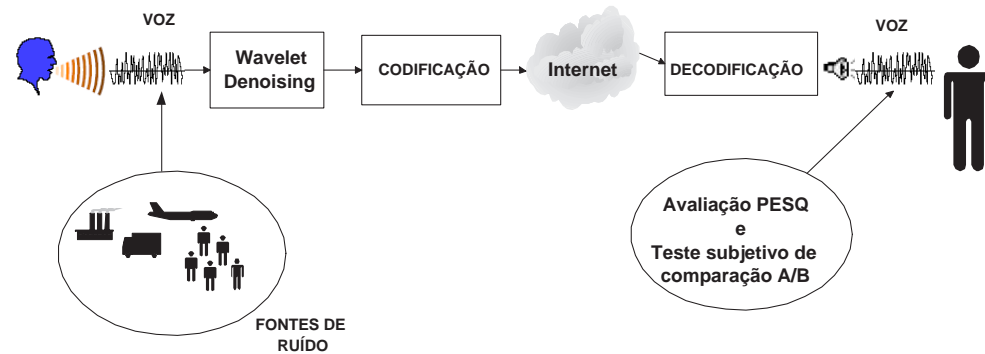


Figura 4.6: Diagrama ilustrativo do sistema com *wavelet denoising*.

Na supressão de ruído com *wavelet*, optou-se pela função de limiar *Soft-Thresholding*, com o objetivo de evitar descontinuidade e instabilidade que pode ocorrer com o uso do *Hard-Thresholding*. Para o cálculo do limiar, foi empregado o método *SureShrink*, pois, em geral, este método provoca menor distorção devido ao fato de computar o limiar de uma forma adaptativa aos dados. Como o *VisuShrink* realiza uma estimativa super-suavizada do sinal, isso poderia ocasionar a perda de características da voz úteis no processo de codificação [38].

É importante ressaltar, ainda, que não é qualquer base de funções que pode ser usada como *wavelets* para representar um sinal. Esta base deve atender a duas condições: a primeira é que deve ser ortonormal, para assim poder realizar a reconstrução do sinal estimado; a segunda condição se refere à propriedade da decomposição *wavelet* do sinal possuir poucos coeficientes não nulos [32]. Uma transformada *wavelet* adequada deve concentrar mais de 90% da energia do sinal nos primeiros $N/2$ coeficientes [39]. Existem algumas famílias de funções, comumente apresentadas pela abreviação do pesquisador que as desenvolveram ("coif" para Coifman ou "db" para Daubechies") que podem ser usadas como funções *wavelet*. Como para sinais de voz a transformada Daubechies 10 (*db10*) satisfaz as condições, esta será usada nas simulações [39]. Além disso, testes de escuta informais, mostraram que a utilização de 5 níveis na transformada *wavelet db10* era a melhor opção a ser utilizada.

4.3.1

Resultado PESQ

Antes de avaliar o impacto da utilização de *wavelet denoising* na qualidade da voz codificada, decidiu-se analisar o emprego dessa técnica em vozes corrompidas por ruído, mas que ainda não foram codificadas. Com essa análise, pode-se avaliar o desempenho do método de supressão de ruído, sem a interferência da codificação. Os resultados dessas simulações, podem ser vistos na Tabela 4.1, onde são mostrados os resultados PESQ em frases que utilizaram *wavelet denoising* e que não utilizaram. Como nas simulações anteriores, os ruídos utilizados são: ruído de falatório, ruído de fábrica e ruído branco. Para melhor compreensão e visualização dos resultados apresentados na Tabela 4.1, as Figuras 4.7, 4.8 e 4.9, mostram o resultado PESQ comparando vozes que passaram por supressão de ruído utilizando *wavelet denoising* e vozes que não passaram por esse método, em ambientes com ruído de falatório, fábrica e branco, respectivamente.

Tabela 4.1: Tabela comparativa do resultado PESQ, para vozes não-codificadas, em ambientes ruidosos, com e sem a utilização de wavelet denoising.

| $RSR(dB)$ | Wavelet Denoising | Falatório | Fábrica | Branco |
|-----------|-------------------|-----------|---------|--------|
| -5 | Não | 2,086 | 2,057 | 2,036 |
| | Sim | 2,048 | 1,952 | 1,355 |
| 0 | Não | 2,274 | 2,296 | 2,159 |
| | Sim | 2,248 | 2,265 | 1,921 |
| 5 | Não | 2,516 | 2,568 | 2,410 |
| | Sim | 2,550 | 2,578 | 2,574 |
| 10 | Não | 2,780 | 2,861 | 2,707 |
| | Sim | 2,843 | 2,885 | 2,930 |
| 15 | Não | 3,066 | 3,169 | 3,021 |
| | Sim | 3,144 | 3,208 | 3,285 |
| 20 | Não | 3,382 | 3,505 | 3,364 |
| | Sim | 3,462 | 3,500 | 3,629 |
| 25 | Não | 3,719 | 3,844 | 3,749 |
| | Sim | 3,700 | 3,726 | 3,887 |
| 30 | Não | 4,011 | 4,108 | 4,094 |
| | Sim | 3,860 | 3,880 | 4,009 |

Na Figura 4.7 pode-se observar, que a utilização de *wavelet denoising* provocou uma ligeira melhora na faixa de RSR entre $5dB$ e $25dB$. Observa-se, ainda, que em casos extremos de RSR muito baixas (abaixo de $0dB$) e muito altas (acima de $25dB$), a supressão de ruído provocou uma pequena deterioração da voz. Testes de escuta informais, sugerem que mesmo utilizando o método *SureShrink* é difícil estimar um limiar para RSR muito baixas, que não cause uma super-suavização da voz. A partir desses testes, verifica-se, também, que para RSR muito altas, a aplicação de denoising não é necessária, pois o ruído é muito baixo e ao passar a voz pelo método de supressão de ruído, esta perde um pouco da qualidade. A Figura 4.8 mostra o resultado PESQ para vozes com ruído de fábrica. Os resultados obtidos são similares aos fornecidos com o ruído de falatório. As simulações feitas com ruído branco, Figura 4.9, mostram, também, que o algoritmo de supressão de ruído tem um bom desempenho na faixa entre $5dB$ e $25dB$. E observa-se desempenhos inferiores nas extremidades da faixa de RSR analisada, principalmente abaixo de $0dB$.

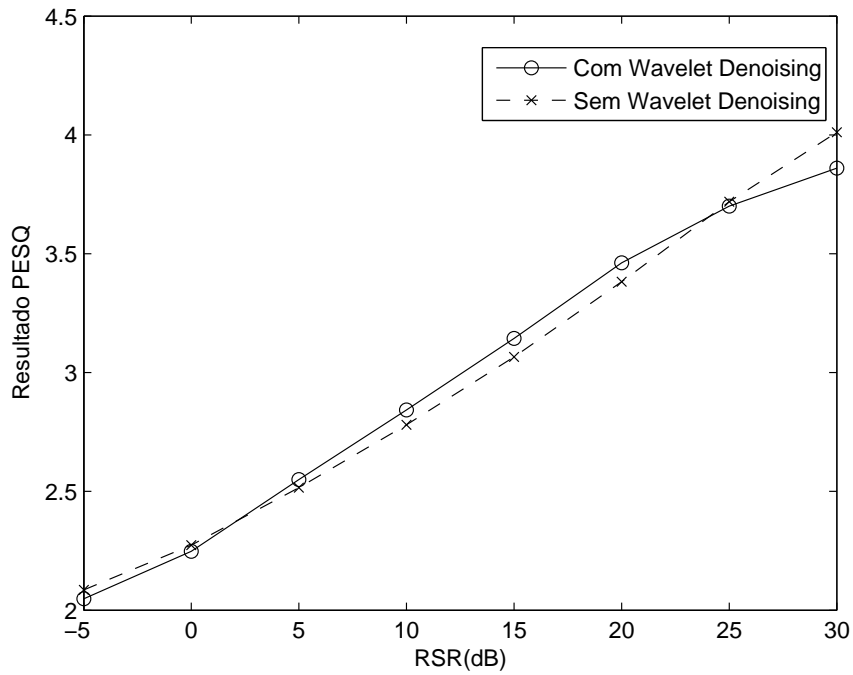


Figura 4.7: Comparação da voz com ruído de falatório utilizando *wavelet denoising* e não utilizando.

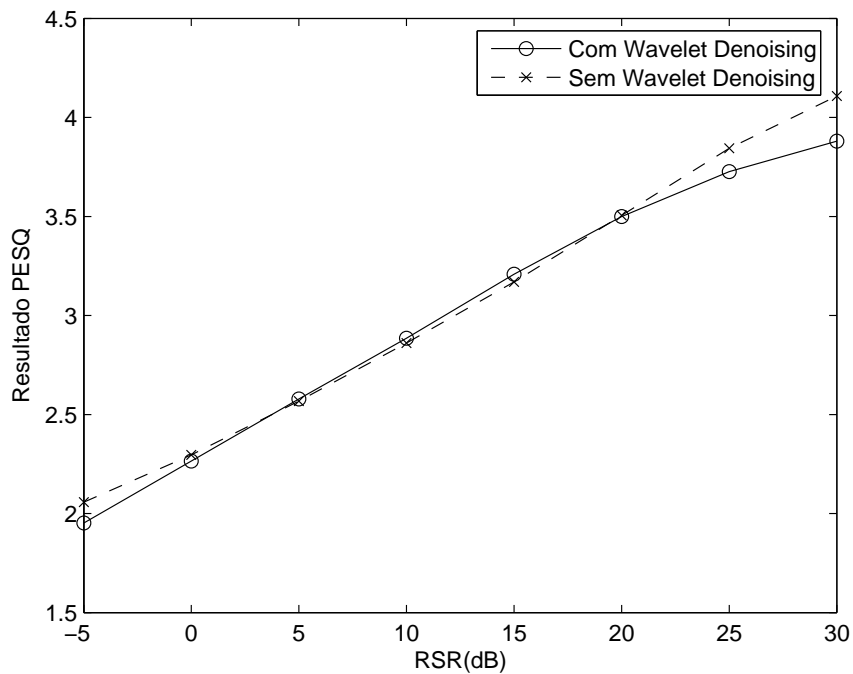


Figura 4.8: Comparação da voz com ruído de fábrica utilizando *wavelet denoising* e não utilizando.

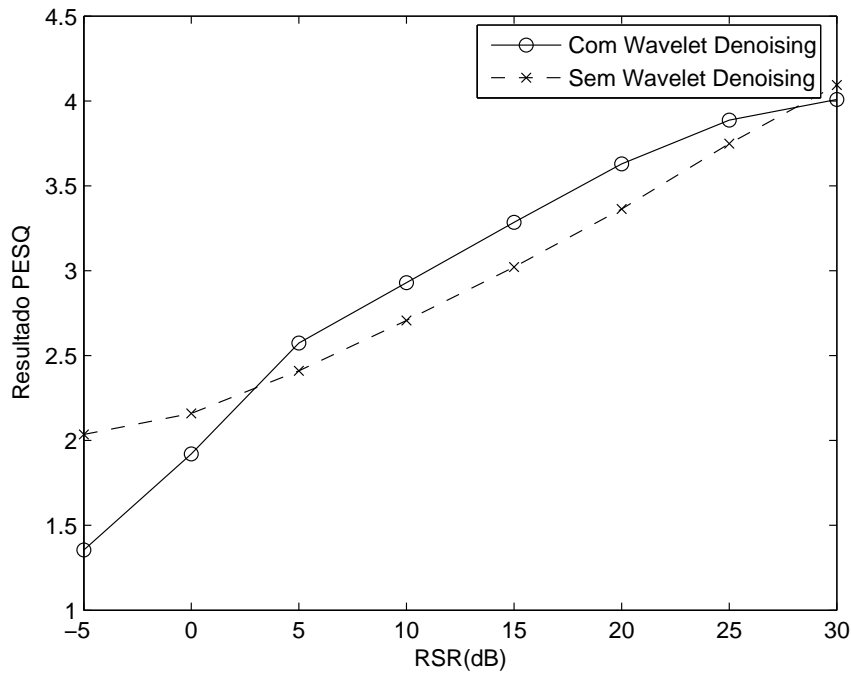


Figura 4.9: Comparação da voz com ruído branco utilizando *wavelet denoising* e não utilizando.

A Tabela 4.2 apresenta os resultados PESQ da voz codificada e sem perda de quadros. Pode-se verificar que o desempenho do codificador, de acordo com a medida PESQ, não melhorou com a utilização do algoritmo de *wavelet denoising*. Observa-se que para quase toda a faixa de RSR testada nos diferentes tipos de ruído, o codificador fornece melhores resultados sem a utilização do método de supressão de ruído. Para melhor visualização, os gráficos das Figuras 4.10, 4.11 e 4.12, extraídos da Tabela 4.2, mostram o desempenho do codificador com frases que utilizaram *wavelet denoising* e frases que não utilizaram, para os ruídos de falatório, fábrica e branco respectivamente.

Tabela 4.2: Tabela comparativa do resultado PESQ, para vozes codificadas com $TPQ = 0\%$, em ambientes ruidosos, com e sem a utilização de wavelet denoising.

| $RSR(dB)$ | Wavelet Denoising | Falatório | Fábrica | Branco |
|-----------|-------------------|-----------|---------|--------|
| -5 | Não | 1,833 | 1,965 | 2,032 |
| | Sim | 1,899 | 1,880 | 1,376 |
| 0 | Não | 1,966 | 2,112 | 2,155 |
| | Sim | 2,049 | 2,034 | 1,825 |
| 5 | Não | 2,132 | 2,271 | 2,327 |
| | Sim | 2,145 | 2,204 | 2,266 |
| 10 | Não | 2,275 | 2,415 | 2,475 |
| | Sim | 2,276 | 2,347 | 2,420 |
| 15 | Não | 2,389 | 2,536 | 2,597 |
| | Sim | 2,345 | 2,439 | 2,559 |
| 20 | Não | 2,502 | 2,598 | 2,659 |
| | Sim | 2,500 | 2,570 | 2,641 |
| 25 | Não | 2,590 | 2,649 | 2,698 |
| | Sim | 2,570 | 2,637 | 2,694 |
| 30 | Não | 2,670 | 2,696 | 2,733 |
| | Sim | 2,643 | 2,670 | 2,709 |

A Figura 4.10 apresenta o desempenho do codificador para vozes com ruído de falatório que utilizaram *denoising* e que não usaram. Observa-se que o codec obteve uma ligeira melhora de desempenho na faixa de RSR entre $-5dB$ e $10dB$ quando utilizado o algoritmo de supressão de ruído. Já para as vozes com ruído de fábrica e ruído branco, Figuras 4.11 e 4.12 respectivamente, o desempenho do codec foi inferior quando utilizado *denoising*. É interessante notar que esse resultado se opõe aos testes preliminares, quando comparou-se os resultados PESQ em vozes não-codificadas, onde obteve-se um valor de PESQ mais alto para vozes que utilizaram *wavelet denoising*, principalmente na faixa entre $5db$ e $25dB$.

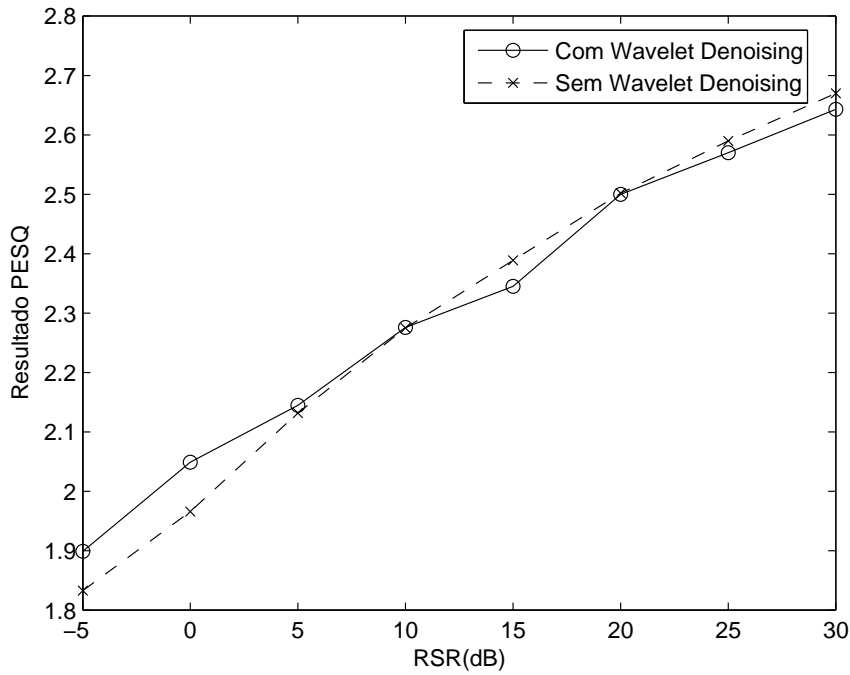


Figura 4.10: Resultado PESQ para voz codificada com ruído de falatório utilizando *wavelet denoising* e não utilizando.

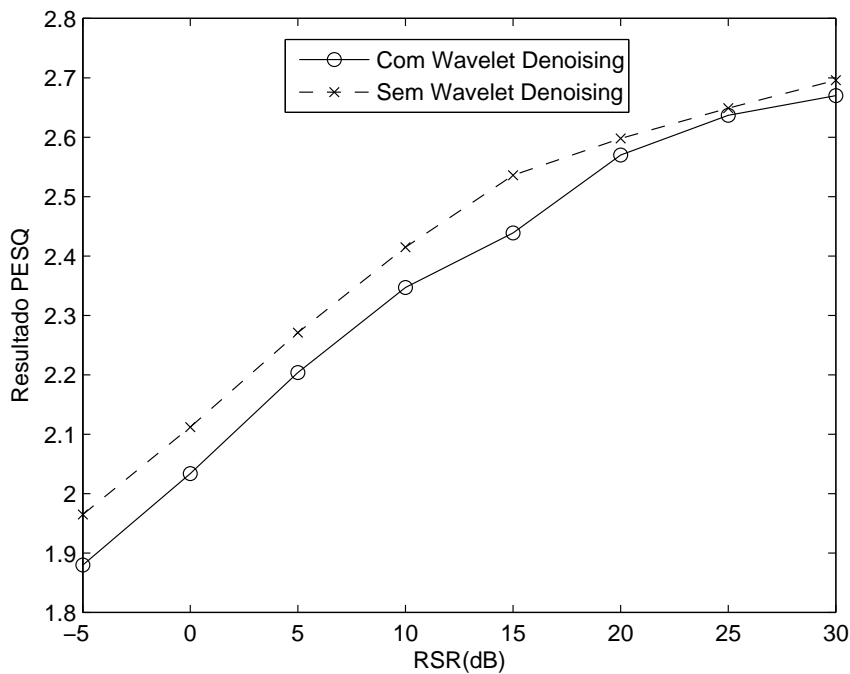


Figura 4.11: Resultado PESQ para voz codificada com ruído de fábrica utilizando *wavelet denoising* e não utilizando.

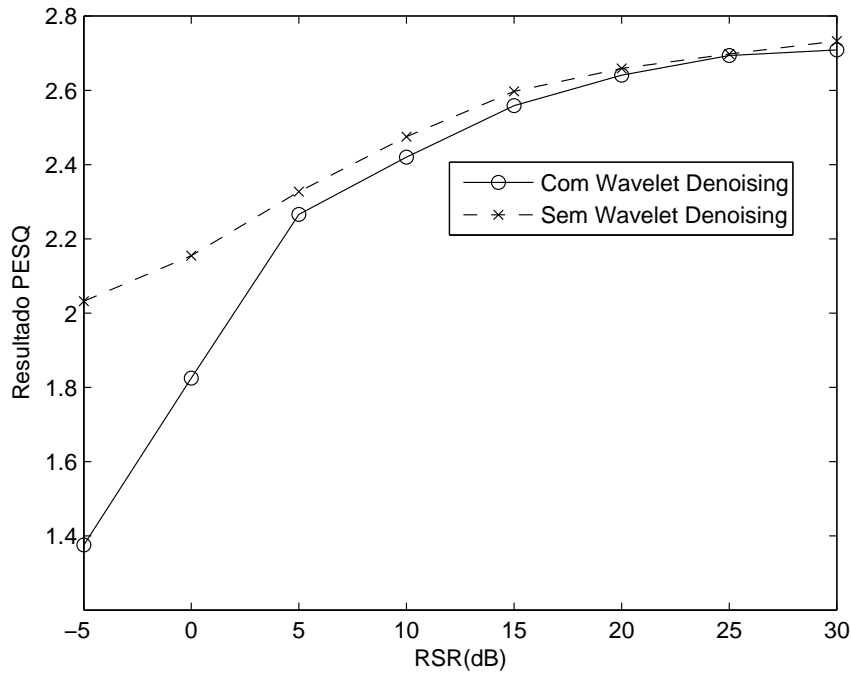


Figura 4.12: Resultado PESQ para voz codificada com ruído branco utilizando *wavelet denoising* e não utilizando.

Através da Tabela 4.3, pode-se comparar os efeitos da supressão de ruído com o codec operando em uma rede com $TPQ = 9\%$. A essa taxa, observa-se, também, que a utilização de *denoising* não melhorou o desempenho do codec. Somente no caso de ruído de falatório e entre $-5dB$ e $5dB$ de RSR, é que pode-se perceber uma pequena melhora, mas não muito significativa, no desempenho do codec com a utilização de *wavelet denoising*.

As Tabelas 4.4 e 4.5 também comparam o desempenho do codec com vozes utilizando *wavelet denoising* e não usando, à taxas de perda de quadros de 28,6% e 38,5% respectivamente. Aqui também não verificamos uma melhora de desempenho quando se emprega *wavelet denoising*.

Tabela 4.3: Tabela comparativa do resultado PESQ, para vozes codificadas com TPQ = 9%, em ambientes ruidosos, com e sem a utilização de wavelet denoising.

| $RSR(dB)$ | Wavelet Denoising | Falatório | Fábrica | Branco |
|-----------|-------------------|-----------|---------|--------|
| -5 | Não | 1,816 | 1,942 | 2,026 |
| | Sim | 1,862 | 1,851 | 1,304 |
| 0 | Não | 1,924 | 2,096 | 2,145 |
| | Sim | 2,014 | 2,013 | 1,405 |
| 5 | Não | 2,093 | 2,249 | 2,303 |
| | Sim | 2,140 | 2,177 | 2,047 |
| 10 | Não | 2,243 | 2,380 | 2,459 |
| | Sim | 2,232 | 2,325 | 2,358 |
| 15 | Não | 2,365 | 2,503 | 2,561 |
| | Sim | 2,309 | 2,438 | 2,526 |
| 20 | Não | 2,481 | 2,567 | 2,635 |
| | Sim | 2,460 | 2,557 | 2,596 |
| 25 | Não | 2,569 | 2,617 | 2,670 |
| | Sim | 2,562 | 2,615 | 2,653 |
| 30 | Não | 2,635 | 2,662 | 2,687 |
| | Sim | 2,616 | 2,635 | 2,689 |

Tabela 4.4: Tabela comparativa do resultado PESQ, para vozes codificadas com TPQ = 28,6%, em ambientes ruidosos, com e sem a utilização de wavelet denoising.

| $RSR(dB)$ | Wavelet Denoising | Falatório | Fábrica | Branco |
|-----------|-------------------|-----------|---------|--------|
| -5 | Não | 1,764 | 1,911 | 2,007 |
| | Sim | 1,847 | 1,850 | 1,190 |
| 0 | Não | 1,877 | 2,009 | 2,091 |
| | Sim | 1,976 | 1,978 | 1,341 |
| 5 | Não | 2,010 | 2,180 | 2,236 |
| | Sim | 2,087 | 2,110 | 1,997 |
| 10 | Não | 2,153 | 2,308 | 2,367 |
| | Sim | 2,154 | 2,247 | 2,295 |
| 15 | Não | 2,311 | 2,411 | 2,474 |
| | Sim | 2,245 | 2,354 | 2,440 |
| 20 | Não | 2,392 | 2,480 | 2,536 |
| | Sim | 2,387 | 2,450 | 2,522 |
| 25 | Não | 2,482 | 2,529 | 2,577 |
| | Sim | 2,451 | 2,508 | 2,551 |
| 30 | Não | 2,532 | 2,557 | 2,587 |
| | Sim | 2,505 | 2,531 | 2,573 |

Tabela 4.5: Tabela comparativa do resultado PESQ, para vozes codificadas com $TPQ = 38,5\%$, em ambientes ruidosos, com e sem a utilização de *wavelet denoising*.

| $RSR(dB)$ | Wavelet Denoising | Falatório | Fábrica | Branco |
|-----------|-------------------|-----------|---------|--------|
| -5 | Não | 1,706 | 1,866 | 1,972 |
| | Sim | 1,787 | 1,812 | 1,101 |
| 0 | Não | 1,807 | 1,970 | 2,055 |
| | Sim | 1,904 | 1,925 | 1,337 |
| 5 | Não | 1,958 | 2,105 | 2,176 |
| | Sim | 2,011 | 2,042 | 1,938 |
| 10 | Não | 2,085 | 2,205 | 2,299 |
| | Sim | 2,091 | 2,155 | 2,219 |
| 15 | Não | 2,209 | 2,224 | 2,301 |
| | Sim | 2,207 | 2,214 | 2,300 |
| 20 | Não | 2,295 | 2,287 | 2,340 |
| | Sim | 2,303 | 2,280 | 2,323 |
| 25 | Não | 2,303 | 2,32 | 2,36 |
| | Sim | 2,364 | 2,329 | 2,380 |
| 30 | Não | 2,340 | 2,361 | 2,363 |
| | Sim | 2,399 | 2,360 | 2,391 |

4.3.2

Resultado do Teste A/B

Na seção anterior, a análise da qualidade da voz utilizando a medida PESQ, mostrou que para vozes não-codificadas e com RSR variando entre $5dB$ à $25dB$, a utilização de *wavelet denoising* aumenta o valor da pontuação PESQ. Mas para testes realizados com o codec operando em redes IP, a utilização desse método de supressão de ruído nas frases a serem codificadas não ocasionou a melhora no desempenho do codec, contrariando as expectativas. Entretanto, testes informais de escuta sinalizam uma melhora de qualidade na voz codificada, pois a utilização do *denoising* ocasiona uma diminuição significativa do ruído de fundo. Por isso, nessa seção, utilizaremos uma medida subjetiva de qualidade - testes de comparação A/B - a fim de comparar os resultados obtidos com o PESQ.

Para realização dos testes subjetivos de comparação A/B, utilizou-se dois conjuntos de frases. Um com frases corrompidas por ruído de falatório e outro formado por frases com ruído branco. Cada conjunto foi formado com RSR variando entre $5dB$ e $15dB$. As frases foram codificadas com o codec operando a uma $TPQ = 0\%$. Essa análise com teste A/B, poderia ter sido feita em todas as RSR e todos os tipos de ruído, mas como essa análise seria muito exaustiva, optou-se por esses dois conjuntos.

O teste A/B foi realizado com 9 pares de sentença para cada con-

junto. O material de teste incluiu voz codificada corrompida por ruído de falatório, com e sem a utilização de *wavelet denoising*, em um conjunto, e voz codificada corrompida por ruído branco, com e sem a utilização de *wavelet denoising* em outro conjunto. As sentenças foram apresentadas a 30 ouvintes, que escolhiam ou a primeira sentença (correspondente a um dos casos avaliados) como de melhor qualidade, ou a segunda sentença, ou consideravam as duas como de qualidade comparável. Como cada par de sentenças foi também apresentado aos avaliadores com a ordem invertida, o teste inclui um total de 540 opiniões para cada conjunto considerado. Os resultados mostrados na Tabela 4.6, referentes ao conjunto de vozes com ruído de falatório, revelam que 42% dos ouvintes não mostraram uma preferência clara, 45,9% deles preferiram a voz codificada com a utilização de *wavelet denoising* e 12% preferiram a codificação sem a utilização de *wavelet denoising*. Na Tabela 4.7, referente ao conjunto de vozes com ruído branco, mostra que 25,4% dos ouvintes não têm uma preferência clara, 52,4% deles preferiram a voz codificada com a utilização de *wavelet denoising* e 22,2% preferiram a codificação sem a utilização de *wavelet denoising*.

Tabela 4.6: Comparação A/B para vozes com ruído de falatório.

| Resultados para vozes codificadas | % |
|-----------------------------------|------|
| Usando Wavelet Denoising | 45,9 |
| Qualidade Comparável | 42,0 |
| Sem Wavelet Denoising | 12,0 |

Tabela 4.7: Comparação A/B para vozes com ruído branco.

| Resultados para vozes codificadas | % |
|-----------------------------------|------|
| Usando Wavelet Denoising | 52,4 |
| Qualidade Comparável | 25,4 |
| Sem Wavelet Denoising | 22,2 |

4.4 Conclusões

Este capítulo apresentou inicialmente uma breve introdução às transformadas *wavelets* e seus principais conceitos. Foi avaliada a qualidade da voz codificada, com e sem a utilização de *wavelet denoising* e em diferentes taxas de perda de quadros, através da medida PESQ. Nessa medida o desempenho do codec não melhorou com a utilização da técnica de supressão de ruído apresentada. No entanto, testes subjetivos de comparação A/B, mostraram que a maioria dos ouvintes preferem a voz codificada com a utilização de *wavelet denoising*. Nota-se que embora a medida PESQ seja uma boa medida de qualidade objetiva de voz, fornecendo bons resultados na maioria dos casos, em determinadas situações são necessários testes de qualidade subjetivos para melhor avaliar a qualidade de voz percebida pelos ouvintes.