



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de Ilha Solteira

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**“Um Método Não-Limiar para Redução de Ruído em
Sinais de Voz no Domínio Wavelet”**

WENDEL CLEBER SOARES

Orientador: Prof. Dr. Francisco Villarreal Alvarado

Co-orientador: Prof. Dr. Jozué Vieira Filho

Tese apresentada à Faculdade de
Engenharia - UNESP – Campus de
Ilha Solteira, para a obtenção do título
de Doutor em Engenharia Elétrica.

Área de Conhecimento: Automação.

Ilha Solteira – SP
Maio/2009

FICHA CATALOGRÁFICA

Elaborada pela Seção Técnica de Aquisição e Tratamento da Informação
Serviço Técnico de Biblioteca e Documentação da UNESP - Ilha Solteira.

S676m Soares, Wendel Cleber.
Um método não-limiar para redução de ruído em sinais de voz no domínio wavelet / Wendel Cleber Soares. -- Ilha Solteira : [s.n.], 2009.
205 f. : il.

Tese (doutorado) - Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira. Área de Conhecimento: Automação, 2009

Orientador: Francisco Villarreal Alvarado
Co-orientador: Jozué Vieira Filho
Bibliografia: p. 140-148

1. Wavelets (Matemática). 2. Ruído. 3. Processamento de sinais.

CERTIFICADO DE APROVAÇÃO

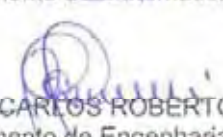
TÍTULO: Um Método Não-Limiar para Redução de Ruído em Sinais de Voz no Domínio Wavelet

AUTOR: WENDEL CLEBER SOARES


ORIENTADOR: Prof. Dr. FRANCISCO VILLARREAL ALVARADO

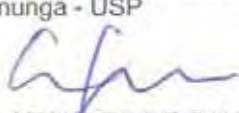
Aprovado como parte das exigências para obtenção do Título de DOUTOR em ENGENHARIA ELÉTRICA, Área: AUTOMACÃO, pela Comissão Examinadora:


Prof. Dr. FRANCISCO VILLARREAL ALVARADO
Departamento de Matemática / Faculdade de Engenharia de Ilha Solteira


Prof. Dr. CARLOS ROBERTO MINUSSI
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira


Prof. Dr. AILTON AKIRA SHINODA
Departamento de Engenharia Elétrica / Faculdade de Engenharia de Ilha Solteira


Prof. Dr. JORGE DIAZ CALLE
Departamento de Ciências Básicas / Faculdade de Zootecnia e Engenharia de Alimentos de Pirassununga - USP


Prof. Dr. LEANDRO DE CAMPOS TEIXEIRA GOMES
Diretoria de Tecnologia de Serviços / Fundação Centro de Pesquisa e Desenvolvimento em Telecomunicações - CPqD/ Campinas

Data da realização: 29 de maio de 2009.

Dedico esta tese ao meu pai Claudio Soares (*in memoriam*), a minha mãe Clarice Santana Martins Soares, a minha esposa Luciana Carrara Adas, ao meu amigo Marco Aparecido Queiroz Duarte e para toda a minha família.

AGRADECIMENTOS

- Primeiramente agradeço a Deus, por ter me dado força e a oportunidade de cumprir mais uma etapa na minha vida.
- À Universidade Estadual Paulista (UNESP – Campus de Ilha Solteira), através do Programa de Pós Graduação em Engenharia Elétrica, pela oportunidade de realizar esta pesquisa e pela acolhida.
- Ao meu orientador Professor Dr. Francisco Villarreal Alvarado, pela competência, estímulo e amizade que me foram fundamentais para o desenvolvimento desta pesquisa.
- O meu mais profundo agradecimento ao Professor Dr. Jozué Vieira Filho, meu co-orientador, pelos conselhos e orientações em todos os momentos da presente pesquisa, particularmente os decisivos, pelas correções e sugestões apresentadas, e principalmente, pela amizade da qual tenho o privilégio de compartilhar.
- Ao companheiro e grande amigo desde os tempos de graduação, o professor Dr. Marco Aparecido Queiroz Duarte, pela orientação, apoio, companheirismo e dedicação total a esta pesquisa. Uma pessoa que não mediu esforços para que esta pesquisa se concretizasse e que a todo instante esteve ao meu lado dando muita força para que continuasse firme nessa empreitada.
- Ao professor Dr. Ricardo Tokio Higuti, pelo apoio dado no início desta nova etapa, pois no momento de desespero por perder um ente querido, soube entender as dificuldades apresentadas por mim no decorrer de sua disciplina oferecida na época.

- Ao professor Dr. Carlos Roberto Minussi, que com algumas palavras de conforto, me deu força para que não desistisse e abrisse mão de tudo, em uma das horas mais difíceis da minha vida.
- A minha mãe Clarice Santana Martins Soares, que sempre me apoiou e esteve ao meu lado, sofrendo junto com as barreiras enfrentadas, sempre invocando o meu nome e a minha pesquisa em suas preces.
- A minha esposa Luciana Carrara Adas, pela solidariedade em todos os momentos difíceis e principalmente pela compreensão.
- Aos colegas do departamento de engenharia elétrica, por sua amizade e apoio.
- A todos os funcionários do departamento de engenharia elétrica e da secretaria de pós-graduação, pela amizade e competência com que sempre me atenderam.
- A todos os funcionários da biblioteca, em especial a senhora Marta Satiko Suzuki e ao senhor João Josué Barbosa, pelo auxílio bibliográfico na confecção da tese.
- Ao Conselho Nacional de Pesquisa (CNPq) pelo apoio financeiro na execução desta pesquisa.
- A todos aqueles que contribuíram para a finalização desta pesquisa.

SOARES, W. C. **Um método não-limiar para redução de ruído em sinais de voz no domínio wavelet**, 2009.205 f. Tese de Doutorado em Engenharia Elétrica, Faculdade de Engenharia, Universidade Estadual Paulista, Ilha Solteira, 2009.

RESUMO

Neste trabalho é feito um estudo dos métodos de redução de ruído aditivo em sinais de voz baseados em *wavelets* e, através deste estudo, propõe-se um novo método não-limiar para redução de ruído em sinais de voz no domínio *wavelet*. Em geral os sinais de voz podem estar contaminados com ruídos artificiais ou reais. O problema consiste que dado um sinal limpo adiciona-se o ruído branco ou colorido, obtendo assim o sinal ruidoso, ambos no domínio do tempo. O que se propõe neste trabalho, é aplicar a transformada *wavelet*, obtendo assim o sinal transformado no domínio *wavelet*, reduzindo ou atenuando o ruído sem o uso de limiar. Os métodos mais usados no domínio *wavelet* são os métodos de redução por limiar, pois permitem bons resultados para sinais contaminados por ruído branco, mas não são eficientes no processamento de sinais contaminados por ruído colorido, que é o tipo de ruído mais comum em situações reais. Nesses métodos, o limiar, geralmente, é calculado nos intervalos de silêncio e aplicado em todo o sinal. Os coeficientes no domínio *wavelet* são comparados com este limiar e aqueles que estão abaixo deste valor são eliminados ou reduzidos, fazendo assim uma aplicação linear deste limiar. Esta eliminação, na maioria das vezes, causa descontinuidades no tempo e na frequência no sinal processado. Além disso, a forma com que o limiar é calculado pode degradar os trechos de voz do sinal processado, principalmente nos casos em que o limiar depende fortemente da última janela do último trecho de silêncio. O método proposto nesta pesquisa consiste na execução de três processamentos, agindo de acordo com as suas características nas regiões de voz e silêncio, sem o uso de limiar. A execução dos três processamentos é sintetizada numa

única função, denominada de função de transferência, que atua como um filtro no processamento do sinal. Este método tem como principal objetivo superar as deficiências dos métodos que utilizam limiares e de realizar o processamento eficaz de sinais contaminados por ruídos reais. Isto faz com que a redução do ruído seja mais uniforme e introduza menos distorções no sinal processado. Para a redução do ruído uma curva de atenuação é gerada. Esta curva é dependente dos coeficientes do sinal, dispensando tratamentos diferenciados para trechos diferentes do sinal. Após a geração dessa curva, a redução é feita efetuando o produto entre os coeficientes do sinal e os coeficientes que geram a curva. O método proposto é testado para quatro tipos de sinais de voz, contaminados por ruído branco e por ruído colorido. Para verificar a qualidade dos sinais processados, são avaliados os níveis de redução de ruído e de distorção usando como medidas de qualidade a SNR e a PESQ. Os resultados mostram que o método é eficiente, pois faz a redução de ruído no sinal sem introduzir distorções e sem grandes variações em sua amplitude. A principal vantagem do método proposto neste trabalho, em relação aos demais métodos de redução de ruído no domínio *wavelet*, é o uso de uma função de transferência que não utiliza limiar para reduzir o ruído. Como essa função não é descontínua, ela não introduz distorções nos trechos de voz do sinal.

Palavras-chave – *Wavelets*, Transformada *Wavelet* Discreta, Redução de Ruído Não-Limiar em Sinais de Voz.

SOARES, W. C. **A non-thresholding method for speech noise reduction in the wavelet domain**, 2009. 205 pg. Thesis (Doctor in Engineering Electrical) – Faculdade de Engenharia, Universidade Estadual Paulista, Ilha Solteira, 2009.

ABSTRACT

In this work a study of the methods for speech noise reduction based on wavelets is done and, through this study, a new non-thresholding method for speech noise reduction in the wavelet domain is proposed. Generally, a speech signal may be corrupted by artificial or real noise. Let a clean signal be corrupted by white or colored noise, rising a noisy signal in time domain. This work proposes the wavelet application to which gives rise to in the wavelet domain. In this domain, noise is reduced or attenuated without a threshold use. After, the signal is recomposed using the inverse discrete wavelet transform.

The most used methods in the wavelet domain are the thresholding reduction methods, because they allow good results for signals corrupted by white noise, but they do not have the same efficiency when processing signals corrupted by colored noise, this is the most common noise in real situations. In those methods, the threshold is usually calculated in the silence intervals and applied to the whole signal. The coefficients in the wavelet domain are compared with this threshold and those that have absolute value below this value are eliminated or reduced, making a linear application of this threshold. This elimination causes discontinuities in time and in the frequency of the processed signal. Besides, the form with that the threshold is applied can degrade the voice segments of the processed signal, principally in cases that the threshold depends strongly on the last window of the last silence segment. The method proposed in this research consists in the execution of three processing, acting according to their characteristics in the voice and silence segments, without the threshold use. The

three processing execution is synthesized in an unique function, called transfer function, acting as a filter in the signal processing. This method has as main objective the overcoming of the deficiencies of the thresholding methods and the effective processing of real noises. When acting, this method is more uniform and introduces less distortion in the processed signal. For the reduction of the noise a reduction curve is generated. This curve is dependent of the signal coefficients, avoiding a differentiated treatment for different segments of the signal. After generating this curve, the reduction is made by the multiplication between the signal coefficients and the curve coefficients. The proposed method is tested in for speech signals, corrupted by white or colored noise. In order to verify the processed signals quality, noise reduction and distortion levels are evaluated using SNR and PESQ measures. The results show that the method is efficient, because it makes the noise reduction without introducing distortions and without changing width. The main advantage of the method proposed in this work, in relation to the other methods of noise reduction in the wavelet domain, is the use of a transfer function that does not use threshold to reduce the noise. This function is not discontinuous, so it does not introduce distortions in the voice segments of the signal.

Keywords – Wavelets, Discrete Wavelet Transform, Non-Thresholding Speech Noise Reduction.

LISTA DE ILUSTRAÇÕES

3.1 – <i>Wavelet</i> de <i>Morlet</i>	42
3.2 – (a) Função escala de <i>Haar</i> ; (b) Função <i>wavelet</i> de <i>Haar</i>	44
3.3 – (a) Função escala de <i>Meyer</i> ; (b) Função <i>wavelet</i> de <i>Meyer</i>	45
3.4 – Função <i>wavelet</i> de <i>Morlet</i>	46
3.5 – (a) Função escala de <i>Daubechies</i> ; (b) Função <i>wavelet</i> de <i>Daubechies</i>	48
3.6 – Decomposição em duas faixas de frequência.....	53
3.7 – Reconstrução em duas faixas de frequência.....	53
3.8 – Decomposição em três faixas de frequência.....	54
3.9 – Reconstrução em três faixas de frequência.....	54
4.1 – <i>Hard Thresholding</i> ($\lambda=2$).....	58
4.2 – <i>Soft Thresholding</i> ($\lambda=2$).....	58
4.3 – <i>Semi-soft Thresholding</i> ($\lambda_1=2, \lambda_2=2\sqrt{2}$).....	61
4.4 – <i>Hard Soft Thresholding</i> ($\lambda=2, \alpha=0,3$).....	62
4.5 – <i>Super Super Soft Thresholding</i> ($\lambda=2, \alpha=0,3$).....	63
4.6 – <i>Hard Super Soft Thresholding</i> ($\lambda=2, \alpha=0,3$).....	64
4.7 – <i>Super Soft Thresholding</i> ($\lambda=2, \alpha=0,3$).....	65
4.8 – <i>Sigmoidal Thresholding</i> ($\lambda=2, \alpha=1$).....	67
4.9 – Sinal limpo no domínio <i>wavelet</i>	69
4.10 – Sinal contaminado por ruído branco no domínio <i>wavelet</i>	69

4.11 – Sinais Limpo e Ruidoso no domínio <i>wavelet</i>	70
4.12 – Método proposto por <i>Akhaee e Marvasti (2006)</i>	74
4.13 – Escolha do segmento para cálculo do limiar	76
5.1 – Janela de trecho de voz de um sinal limpo e contaminado por ruído branco, no domínio <i>wavelet</i>	79
5.2 – Segunda metade da janela apresentada na Figura 5.1	80
5.3 – Diagrama do Método Proposto.....	81
5.4 – Sinal Limpo $x(n)$	86
5.5 – Sinal de voz limpo $x(n)$ e sinal processado $\hat{y}_s(n)$, usando o operador OMS e a transformada <i>wavelet</i> inversa.....	87
5.6 – Sinal de voz $s(n)$ contaminado por ruído branco, $r(n)$	87
5.7 – Amplificação do sinal processado $\hat{y}_s(n)$, usando o operador OMS e a transformada <i>wavelet</i> inversa	88
5.8 – Sinal processado $\hat{y}_x(n)$ a partir do sinal original limpo $x(n)$, usando o operador OMS e a transformada <i>wavelet</i> inversa.....	89
5.9 – Sinal processado $\hat{y}_s(n)$, usando o operador OMS e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído branco $r(n)$	90
5.10 – Sinal processado $\hat{y}_0(n)$, usando o operador OMS e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído colorido $r_0(n)$	91
5.11 – Sinal de voz limpo $x(n)$ e sinal processado $\hat{y}_s(n)$, usando o operador SNRP e a transformada <i>wavelet</i> inversa.....	93
5.12 – Processamento usando o operador SNRP (sinal processado $\hat{y}_s(n)$).....	93
5.13 – Sinal processado $\hat{y}_x(n)$ a partir do sinal original limpo $x(n)$, usando o operador SNRP e a transformada <i>wavelet</i> inversa.....	95
5.14 – Sinal processado $\hat{y}_s(n)$, usando o operador SNRP e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído branco $r(n)$	95
5.15 – Sinal processado $\hat{y}_0(n)$, usando o operador SNRP e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído colorido	

$r_0(n)$	96
5.16 – Sinal processado $\hat{y}_s(n)$ a partir do sinal original limpo $x(n)$, usando o operador SNRPR e a transformada <i>wavelet</i> inversa.....	98
5.17 – Sinal processado $\hat{y}_s(n)$ com o operador SNRPR	98
5.18 – Sinal processado $\hat{y}_x(n)$ a partir do sinal original limpo $x(n)$, usando o operador SNRPR e a transformada <i>wavelet</i> inversa.....	100
5.19 – Sinal processado $\hat{y}_s(n)$, usando o operador SNRPR e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído branco $r(n)$	100
5.20 – Sinal processado $\hat{y}_0(n)$, usando o operador SNRPR e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído colorido $r_0(n)$	101
5.21 – Sinal processado $\hat{y}_s(n)$ via Combinação Polinomial e a transformada <i>wavelet</i> inversa	104
5.22 – Curva de atenuação usada no processamento, aplicando a função de Transferência <i>fil</i>	106
5.23 – Gráfico de uma Função Sigmoide	107
5.24 – Gráfico do valor absoluto de uma Função Sigmoide	107
5.25 – Sinal de voz limpo $x(n)$ e sinal processado $\hat{y}_s(n)$, usando o método proposto e a transformada <i>wavelet</i> inversa	108
5.26 – Sinal processado $\hat{y}_s(n)$, pelo método proposto	109
5.27 – Sinais limpo e processado no domínio <i>wavelet</i> com ruído branco – erro 2,83 %	110
5.28 – Trecho “vozeado” do sinal processado com ruído branco no domínio <i>wavelet</i> – erro 1,81 %	111
5.29 – Trecho “não vozeado” do sinal processado com ruído branco no domínio <i>wavelet</i> – erro 4,16 %	111
5.30 – Sinais limpo e processado com ruído colorido no domínio <i>wavelet</i> – erro 3,51 %	112
5.31 – Trecho “vozeado” do sinal processado com ruído colorido no domínio <i>wavelet</i> – erro 0,4 %	113

5.32 – Trecho “não vozeado” do sinal processado com ruído colorido no domínio <i>wavelet</i> – erro 0,5 %	113
6.1 – (a) forma de onda do sinal A x_A (8); (b) Sinal A com ruído branco $s_{A,b}$ (8); (c) Sinal A com ruído colorido $s_{A,0}$ (8)	119
6.2 – (a) forma de onda do sinal B x_B (8); (b) Sinal B com ruído branco $s_{B,b}$ (8); (c) Sinal B com ruído colorido $s_{B,0}$ (8)	120
6.3 – (a) forma de onda do sinal C x_C (8); (b) Sinal C com ruído branco $s_{C,b}$ (8); (c) Sinal C com ruído colorido $s_{C,0}$ (8)	122
6.4 – (a) forma de onda do sinal D x_D (8); (b) Sinal D com ruído branco $s_{D,b}$ (8); (c) Sinal D com ruído colorido $s_{D,0}$ (8)	123
6.5 – Sinal processado $\hat{y}_{A,b}$ (8) (10 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_A (8) contaminado com ruído branco r_b (8)	127
6.6 – Sinal processado $\hat{y}_{A,b}$ (8) (5 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_A (8) contaminado com ruído branco r_b (8)	127
6.7 – Sinal processado $\hat{y}_{A,0}$ (8) (10 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_A (8) contaminado com ruído colorido r_0 (8)	128
6.8 – Sinal processado $\hat{y}_{A,0}$ (8) (5 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_A (8) contaminado com ruído colorido r_0 (8)	128
6.9 – Sinal processado $\hat{y}_{B,b}$ (8) (10 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_B (8) contaminado com ruído branco r_b (8)	129
6.10 – Sinal processado $\hat{y}_{B,b}$ (8) (5 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_B (8) contaminado com ruído branco r_b (8)	129
6.11 – Sinal processado $\hat{y}_{B,0}$ (8) (10 dB), usando o método proposto e a transformada	

<i>wavelet</i> inversa, a partir do sinal de voz limpo x_B (8) contaminado com ruído colorido r_0 (8)	130
6.12 – Sinal processado $\hat{y}_{B,0}$ (8) (5 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_B (8) contaminado com ruído colorido r_0 (8)	130
6.13 – Sinal processado $\hat{y}_{C,b}$ (8) (10 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_C (8) contaminado com ruído branco r_b (8)	131
6.14 – Sinal processado $\hat{y}_{C,b}$ (8) (5 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_C (8) contaminado com ruído branco r_b (8)	131
6.15 – Sinal processado $\hat{y}_{C,0}$ (8) (10 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_C (8) contaminado com ruído colorido r_0 (8)	132
6.16 – Sinal processado $\hat{y}_{C,0}$ (8) (5 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_C (8) contaminado com ruído colorido r_0 (8)	132
6.17 – Sinal processado $\hat{y}_{D,b}$ (8) (10 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_D (8) contaminado com ruído branco r_b (8)	133
6.18 – Sinal processado $\hat{y}_{D,b}$ (8) (5 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_D (8) contaminado com ruído branco r_b (8)	133
6.19 – Sinal processado $\hat{y}_{D,0}$ (8) (10 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_D (8) contaminado com ruído colorido r_0 (8)	134
6.20 – Sinal processado $\hat{y}_{D,0}$ (8) (5 dB), usando o método proposto e a transformada <i>wavelet</i> inversa, a partir do sinal de voz limpo x_D (8) contaminado com ruído colorido r_0 (8)	134
A1 – Representação dos coeficientes do sinal s (3)	151

A2 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $s^{(3-1)}$, resultante do primeiro processamento de decomposição	153
A3 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $s^{(3-2)}$, resultante do segundo processamento de decomposição.....	154
A4 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $S[3]$, resultante do terceiro processamento de decomposição.....	156
A5 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $s^{(3-2)}$, resultante do primeiro processamento da reconstrução do sinal	158
A6 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $s^{(3-1)}$, resultante do segundo processamento da reconstrução do sinal.....	160
A7 – Representação dos coeficientes do sinal reconstruído $s(3)$ usando a transformada <i>wavelet</i> inversa	162
A8 – Representação da compressão dos coeficientes do sinal original $s(3)$	163
B1 – Representação dos coeficientes do sinal ruidoso $S[3]$	165
B2 – Representação dos coeficientes de $F[3]$ após aplicação do Operador OMS ao sinal $S[3]$	167
B3 – Representação dos coeficientes de $F'[3]$ após aplicação do Operador OMS ao sinal $S[3]$, sem utilização do valor absoluto	168
B4 – Representação dos coeficientes de $S[3]$, $F[3]$ e $F'[3]$	169
B5 – Resultante da operação $y_i = s_i \cdot f_i$: representação dos coeficientes do sinal $Y[3]$	170
B6 – Comparação entre as curvas (sinal original $S[3]$ e sinal processado $Y[3]$).....	171
C1 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $\hat{y}_{(3-1)}$, resultante do primeiro processamento da reconstrução do sinal ruidoso	175
C2 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $\hat{y}_{(3-2)}$, resultante do segundo processamento da reconstrução do sinal ruidoso	177
C3 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $\hat{y}(3)$, resultante do terceiro processamento da reconstrução do sinal ruidoso	179
D1 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal	

$\hat{y}^3(3)$, resultante do terceiro processamento da reconstrução do sinal ruidoso	182
D2 – Representação dos coeficientes do sinal $s(3)$ (em azul); representação dos coeficientes do sinal $\hat{y}(3)$ (em verde) e representação dos coeficientes do sinal $\hat{y}^3(3)$ (em vermelho).....	183
D3 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $\hat{z}^2(3)$, resultante do terceiro processamento da reconstrução do sinal ruidoso	185
D4 – Representação dos coeficientes do sinal $s(3)$ (em azul); representação dos coeficientes do sinal $\hat{z}(3)$ (em verde) e representação dos coeficientes do sinal $\hat{z}^2(3)$ (em vermelho).....	186
D5 – Sinal $s(3)$ (em azul) e $\hat{y}_2^3(3)$, combinação de $\hat{y}^3(3)$ e de $\hat{z}^2(3)$	187
E1 – Sinal $s(3)$ (em azul) e $\hat{y}_2^3(3)$, combinação de $\hat{y}^3(3)$ e de $\hat{z}^2(3)$	189
E2 – Trecho de um sinal de voz (Sinal A) no domínio <i>wavelet</i> (domínio transformado)	189
E3 – Trecho do sinal de voz (Sinal A) processado no domínio <i>wavelet</i> utilizando o operador OMS	190
E4 – Reconstrução do trecho do sinal de voz (Sinal A) no domínio do tempo usando a TWI.....	191
E5 – Trecho do sinal de voz (Sinal A) original no domínio do tempo (em azul: Figura E1) e trecho do sinal de voz (Sinal A) processado pelo operador OMS reconstruído no domínio do tempo (em vermelho:Figura E4)	192
E6 – Trecho do sinal de voz (Sinal A) processado no domínio <i>wavelet</i> utilizando a combinação proposta	193
E7 – Reconstrução do trecho do sinal de voz (Sinal A) usando a TWI no domínio do tempo utilizando a combinação proposta	193
E8 – Trecho do sinal de voz (Sinal A) original no domínio do tempo (em azul: Figura E1) e trecho do sinal de voz (Sinal A) processado utilizando a combinação proposta no domínio do tempo (em vermelho: Figura E7)	194
F1 – Sinal limpo $X[n]$	196
F2 – Sinal contaminado com ruído branco $S[n]$	196
F3 – Sinal contaminado com ruído colorido $S_0[n]$	197

F4 – Sinal limpo processado pelo operador OMS, $Y_x [n]$	197
F5 – Sinal contaminado com ruído branco processado pelo operador OMS, $Y_s [n]$..	198
F6 – Sinal contaminado com ruído colorido processado pelo operador OMS, $Y_0 [n]$	198
F7 – Sinal limpo processado pelo operador SNRP, $Y_x [n]$	199
F8 – Sinal contaminado com ruído branco processado pelo operador SNRP, $Y_s [n]$.	199
F9 – Sinal contaminado com ruído colorido processado pelo operador SNRP, $Y_0 [n]$	200
F10 – Sinal limpo processado pelo operador SNRPR, $Y_x [n]$	200
F11 – Sinal contaminado com ruído branco processado pelo operador SNRPR, $Y_s [n]$	201
F12 – Sinal contaminado com ruído colorido processado pelo operador SNRPR, $Y_0 [n]$	201
F13 – Sinal limpo processado pela Combinação Proposta, $Y_x [n]$	202
F14 – Sinal contaminado com ruído branco processado pela Combinação Proposta, $Y_s [n]$	202
F15 – Sinal contaminado com ruído colorido processado pela Combinação Proposta, $Y_0 [n]$	203

LISTA DE TABELAS

6.1 – SNR dos sinais limpos x_k (8)	118
6.2 – SNR dos sinais processados (5 dB)	125
6.3 – SNR dos sinais processados (10 dB)	125
6.4 – PESQ dos sinais ruidosos $s_{k,l}$ (8)	125
6.5 – PESQ do sinais processados $\hat{y}_{k,l}$ (8)	126

LISTA DE SÍMBOLOS

σ_x^2 - Variância do sinal original

σ_e^2 - Variância do erro

dB – Decibel

log – Logaritmo

ms - milissegundo

\mathbf{N} – Conjunto dos números naturais

\mathbf{R} – Conjunto dos números reais

\mathbf{Z} – Conjunto dos números inteiros

$L^2(\mathbf{R})$ – Espaço das funções Lebesgue - mensuráveis de quadrado integráveis

$\psi(t)$ – Função *Wavelet*

$\phi(t)$ – Função de escala

a – Parâmetro de escala da função *wavelet*

b – Parâmetro de translação da função *wavelet*

$|\lambda|$ - Módulo de λ

$\|x\|$ - Norma de x

\hat{f} - Transformada de Fourier de f

f^* - Conjugado complexo de f

$W_f(a,b)$ – Transformada *Wavelet* Contínua de f

$\langle a, b \rangle$ - Produto interno a e b

$f(t)$ – Função no domínio do tempo

π - Número Pi

Im – Parte Imaginária

Re – Parte Real

$h[n]$ – Filtro passa baixa

$g[n]$ – Filtro passa banda

$c_i[n]$ – Coeficientes de aproximação do nível i da DWT

$d_i[n]$ – Coeficientes de detalhes do nível i da DWT

$(\downarrow 2)$ – Operador de dizimação de ordem 2

$(\uparrow 2)$ – Operador de inserção de zeros de ordem 2

k – Número de faixas de frequências

W – Matriz da transformada *wavelet* discreta

λ - Limar usado para filtragem dos sinais

γ - Parâmetro de inclinação da Sigmoides

fil – Função de transferência

km/h – Quilômetro por hora

$x(n)$ - Sinal de voz limpo no domínio do tempo

$r(n)$ - Ruído branco no domínio do tempo

$r_0(n)$ - Ruído colorido no domínio do tempo

$s(n)$ - Sinal de voz contaminado com ruído branco no domínio do tempo

$s_0(n)$ - Sinal de voz contaminado com ruído colorido no domínio do tempo

$X[n]$ - sinal de voz limpo no domínio transformado (domínio *wavelet*)

$R[n]$ - Ruído branco no domínio transformado (domínio *wavelet*)

$R_0[n]$ - Ruído colorido no domínio transformado (domínio *wavelet*)

$S[n]$ - Sinal de voz contaminado com ruído branco no domínio *wavelet*

$S_0[n]$ - Sinal de voz contaminado com ruído colorido no domínio *wavelet*

$Y_x[n]$ - Sinal de voz limpo processado a partir de $X[n]$, pelos operadores e pela combinação proposta no domínio *wavelet*

$Y_s[n]$ - Sinal de voz contaminado com ruído branco processado a partir de $S[n]$, pelos operadores e pela combinação proposta no domínio *wavelet*

$Y_0[n]$ - Sinal de voz contaminado com ruído colorido processado a partir de $S_0[n]$, pelos operadores e pela combinação proposta no domínio *wavelet*

$\hat{y}_x(n)$ - Sinal processado a partir do sinal original limpo $x(n)$, usando os operadores, a combinação proposta no domínio do tempo e a transformada *wavelet* inversa

$\hat{y}_s(n)$ - Sinal processado a partir do sinal original limpo $x(n)$, contaminado com ruído branco $s(n)$, usando os operadores, a combinação proposta no domínio do tempo e a transformada *wavelet* inversa

$\hat{y}_0(n)$ - Sinal processado a partir do sinal original limpo $x(n)$, contaminado com ruído colorido $s_0(n)$, usando os operadores, a combinação proposta no domínio do tempo e a transformada *wavelet* inversa

LISTA DE ABREVIATURAS

ACR – *Absolute Category Rating method*

AN – Amplitude Numérica

CTR – Coeficiente por limiar a *priori*

CWT – Transformada *Wavelet* Contínua

dB2 – Função *Wavelet* de *Daubechies* Nível 2

DR – Faixa Dinâmica

DRT – Teste de Rima Diagnóstico

DWT – Transformada *Wavelet* Discreta

EMF – Filtros de Ephraim e Malah

FBI – *Federal Bureau of Investigation*

fil – Função de Transferência

IAJ – Teste de Qualidade do Julgamento Absoluto Isométrico

IDWT – Transformada *Wavelet* Discreta Inversa

ITU – *International Telecommunications Union*

JPEG – *Joint Photographic Experts Group*

LMS – Algoritmo dos quadrados mínimos

LP – Predição Linear

MATLAB – *Matrix Laboratory*

MOS – *Mean Square Opinion*

MRT – Teste de Rima Modificado

NMT – *Noise Masking Threshold*
OMS – Operador Média Simples
PESQ – *Perceptual Evaluation of Speech Quality*
SNR – Relação Sinal/Ruído
SNRP – Relação Sinal/Ruído a *Priori*
SNRPR – Relação Sinal/Ruído a *Posteriori*
TEO – *Teager Energy Operator*
TRH – *Threshold* ou limiar
TW – Transformada *Wavelet*
TWI – Transformada *Wavelet* Inversa
WBSS – *Wavelet Based Spectral Subtraction*
WPT – Transformada *Wavelet Packet*

SUMÁRIO

CAPÍTULO 1 – INTRODUÇÃO	27
1.1 Considerações Preliminares e Motivação.....	27
1.2 Organização do Texto.....	29
1.3 Contribuições da Tese	30
CAPÍTULO 2 – REDUÇÃO DE RUÍDO EM SINAIS DE VOZ	
Introdução.....	31
2.1 Medidas de Qualidade	34
2.1.1 Medidas de Qualidade Subjetivas.....	34
2.1.2 Medidas de Qualidade Objetivas.....	36
2.1.2.1 A Medida Relação Sinal Ruído	36
2.1.2.2 Outras Medidas Objetivas Baseadas em Distância	37
CAPÍTULO 3 – ANÁLISE WAVELET	
Introdução.....	39
3.1 <i>Wavelets</i>	41
3.2 Funções <i>Wavelets</i>	42
3.2.1 <i>Wavelet</i> de <i>Haar</i>	43
3.2.2 <i>Wavelet</i> de <i>Meyer</i>	44
3.2.3 <i>Wavelet</i> de <i>Morlet</i>	45

3.2.4 <i>Wavelets</i> de <i>Daubechies</i>	46
3.3 Transformada <i>Wavelet</i>	48
3.3.1 Transformada <i>Wavelet</i> Contínua	49
3.3.2 As <i>Wavelets</i> Filhas (Famílias de <i>Wavelets</i>)	49
3.3.3 Transformada <i>Wavelet</i> Contínua Inversa	50
3.3.4 Transformada <i>Wavelet</i> Discreta (DWT)	51
3.3.5 Transformada <i>Wavelet</i> Discreta Inversa (IDWT)	52
3.3.6 Transformada <i>Wavelet</i> como um Banco de Filtros	52

CAPÍTULO 4 – WAVELETS E PROCESSAMENTO DE SINAIS DE VOZ

Introdução	55
4.1 Método <i>Wavelet</i> Básico	56
4.2 Métodos de Redução de Ruído que Utilizam Limiar	59
4.3 Desvantagens dos Métodos que Utilizam Limiar	67
4.4 Análise dos Métodos Híbridos para Redução de Ruído em Sinais de Voz (Usando <i>Wavelets</i>)	70
4.5 Análise de um Método Sem o Uso de Limiar	74
4.5.1 Processamento	75

CAPÍTULO 5 – METODOLOGIA PROPOSTA

5.1 Aspectos Gerais	78
5.1.1 Objetivos do Método Proposto	80
5.2 Processamento dos operadores propostos	82
5.2.1 Primeiro processamento – Operador OMS	84
5.2.2 Segundo processamento – Operador SNRP	91
5.2.3 Terceiro processamento – Operador SNRPR	96
5.3 Processamento via Combinação Polinomial	102
5.4 Análise dos sinais limpo e processado no domínio <i>wavelet</i>	109

CAPÍTULO 6 – ANÁLISE E DISCUSSÃO DOS RESULTADOS

Introdução	115
6.1 Sinais de Voz e Considerações	116
6.1.1 Avaliações Objetivas dos Sinais Processados pelo Método Proposto	117

6.2 Conclusões.....	135
CAPÍTULO 7 – CONCLUSÕES	137
REFERÊNCIAS	140
APÊNDICE A	149
A1 Processo de Decomposição usando a Transformada <i>Wavelet</i>	149
A2 Processo de Reconstrução usando a Transformada <i>Wavelet</i> Inversa	156
APÊNDICE B.....	164
B1 O Operador OMS e a Transformada <i>Wavelet</i>	164
APÊNDICE C	173
C1 Processo de Decomposição usando a Transformada <i>Wavelet</i> Inversa no Sinal Ruidoso.....	173
APÊNDICE D	180
D1 Método Proposto.....	180
APÊNDICE E.....	188
E1 Análise com um Sinal de Voz.....	188
APÊNDICE F.....	195
Sinais no Domínio <i>Wavelet</i>	195
F1 Sinais antes do Processamento	195
F2 Operador OMS	197
F3 Operador SNRP	199
F4 Operador SNRPR	200
F5 Combinação Proposta.....	202
APÊNDICE G	204
G1 Publicações	204

CAPÍTULO 1

INTRODUÇÃO

1.1 Considerações Preliminares e Motivação

O processamento de sinais é uma das áreas mais importantes da matemática aplicada e da engenharia, pois muitos fenômenos físicos necessitam ser equacionados e analisados para possibilitar o estudo de suas mais variadas características. Existem, porém, sinais que não podem ser representados por modelos matemáticos, devido ao alto grau de complexidade que envolve seus componentes, mas que da mesma forma necessitam ser analisados. Estes tipos de sinais podem ser analisados por alguma ferramenta matemática que seja capaz de evidenciar suas propriedades (LYNN; FUERST, 1994). Dentre as ferramentas usadas para a análise de sinais estão as transformadas integrais, dentre as quais se destacam a transformada de Fourier e a transformada *wavelet* (MARTINS, 2000).

Um dos sinais que não pode ser equacionados por um modelo matemático é o sinal de voz. Porém, o processamento de sinais de voz é uma área muito importante, pois existem muitas aplicações envolvendo este tipo de sinal, tais como codificação, reconhecimento automático de fala e redução de ruído (NORDSTRÖM, 2002).

Um sinal de voz pode ser contaminado por ruído ao ser gravado, transmitido por meios de telecomunicações ou devido ao meio de reprodução do mesmo. De

qualquer forma, a presença de ruído é algo incômodo para o ouvinte, além de comprometer a inteligibilidade do sinal. Por isso, muitos trabalhos científicos têm se dedicado à busca de métodos eficientes para redução ou eliminação do ruído presente em sinais de voz.

A transformada *wavelet* é uma ferramenta muito usada na redução de ruídos em sinais de voz, devido a forma em que a mesma atua num sinal, decompondo-o em várias faixas de frequência (DAE-SUNG et al., 2002). Além disso, é notório o fato de que a maioria dos coeficientes de um sinal no domínio *wavelet* tem valor absoluto muito próximo de zero, de forma que os mesmos podem ser eliminados sem comprometer a inteligibilidade do sinal. Assim, a eliminação destes coeficientes, num sinal ruidoso, já se configura como redução de ruído, embora, neste caso, a redução seja muito baixa (DUARTE, 2005).

A maioria dos métodos usados para redução de ruído em sinais de voz que usam a transformada *wavelet* tem como metodologia básica corte ou atenuação baseada em um limiar (DAE-SUNG et al., 2002; DONOHO; JOHNSTONE, 1994). Nestes métodos, o valor calculado no domínio *wavelet*, geralmente utilizando a potência do ruído, e os coeficientes do sinal no domínio *wavelet*, que têm valores absolutos menores que este limiar, são eliminados ou atenuados. Após esta operação, o sinal é reconstruído pela transformada *wavelet* inversa. Uma das deficiências dos métodos que usam limiar é a descontinuidade, geralmente causada pelas funções de transferência usadas nestes métodos, no sinal processado. Essa descontinuidade causa forte degradação no sinal reconstruído, comprometendo sua inteligibilidade. Além disso, a maioria desses métodos é muito eficaz quando aplicado a sinais contaminados por ruído branco. Porém, não apresenta a mesma eficiência quando aplicado a sinais contaminados por ruídos reais, como, por exemplo, o ruído presente no interior de uma fábrica (DUARTE, 2004).

O método proposto neste trabalho tem como objetivo superar as deficiências dos métodos baseados em limiar. A principal diferença deste método em relação aos métodos mencionados anteriormente está exatamente no fato de não usar limiar. Em vez de uma função de transferência que elimina ou atenua coeficientes que têm valores absolutos abaixo de um determinado valor, este método usa uma curva de atenuação baseada no ruído presente em cada coeficiente do sinal. Desta forma, o que se propõe é um método de redução e não de eliminação de ruído. A metodologia de redução de ruído proposta neste trabalho consiste na execução de três processamentos parciais

independentes, descritos no capítulo 5, cada um agindo de acordo com as suas características nas regiões de voz e silêncio sem o uso de limiar. Usando os três processamentos, aplica-se um processamento via combinação polinomial e, em seguida, utiliza-se uma função de transferência, associada à esta combinação, que realiza uma operação de filtragem no sinal de voz ruidoso, fazendo uma adequação de amplitude via ajuste sigmoidal. A principal motivação para o desenvolvimento deste trabalho usando esta metodologia é a ausência de descontinuidade no sinal processado, fato que não acontece quando se usam métodos que empregam limiar.

1.2 Organização do Texto

O texto é organizado inicialmente com um capítulo sobre um histórico da redução de ruído em sinais de voz e as medidas para avaliação do desempenho dos métodos de redução de ruído, apresentados no capítulo 2. No capítulo 3, é apresentado um estudo detalhado sobre a teoria *wavelet*, com seus principais fundamentos matemáticos, a ser usada no trabalho. No capítulo 4, é feito um levantamento bibliográfico, estudo e análises dos métodos de redução de ruído em sinais de voz baseado em *wavelets* nos últimos anos, verificando as vantagens e desvantagens desses métodos. No capítulo 5 é apresentado e discutido o novo método proposto para redução de ruído em sinais de voz usando *wavelets* sem o uso de limiar. No capítulo 6 são apresentados os resultados obtidos e a avaliação da aplicação do método proposto no capítulo 5 e, no capítulo 7, são apresentadas as conclusões e as considerações finais deste trabalho.

Ainda, ao final deste trabalho, apresentam-se seis apêndices de suma importância para a leitura e para a compreensão dos conceitos relacionados e abordados principalmente no capítulo 5. Estes apêndices vêm detalhar, via exemplos, de uma forma simples, as principais abrangências consideradas no trabalho, para um melhor entendimento do leitor, as quais estão distribuídas da seguinte forma:

Apêndice A – Mostra os procedimentos básicos de decomposição de um sinal usando a Transformada *Wavelet* (TW) e de reconstrução usando a Transformada *Wavelet* Inversa (TWI), usando como ilustração, um vetor com apenas 8 elementos, para um melhor entendimento do leitor.

Apêndice B – Tem como o intuito é facilitar a compreensão da execução dos processamentos denominados no trabalho como operadores. O objetivo é fazer com que o leitor entenda os procedimentos básicos desta operação.

Apêndice C – Aborda o processo de reconstrução de um sinal, usando a transformada *wavelet* inversa, a partir de um sinal ruidoso. O objetivo é entender as transformações que são feitas sobre um sinal ruidoso, que está no domínio *wavelet*, e que após sofrer a aplicação da transformada *wavelet* inversa, obtém-se o sinal original “limpo” no domínio do tempo.

Apêndice D – Detalha a metodologia proposta no trabalho, que consiste na aplicação de três operadores parciais e independentes. Propõe-se ainda, um processamento que faz uma combinação polinomial dos três operadores e, em seguida, faz uma adequação de amplitude via ajuste sigmoidal. Toda essa explanação é ilustrada com o mesmo vetor que contém 8 elementos apresentados no apêndice A.

Apêndice E – Faz uma análise do método proposto com um trecho de um sinal de voz que é apresentado no capítulo 6 (Sinal A) utilizando os conceitos que foram apresentados nos apêndices anteriores. O objetivo principal é que o leitor possa visualizar e compreender a metodologia proposta neste trabalho, utilizando-se de um sinal de voz real.

Apêndice F – Apresenta as figuras relativas aos três processamentos e a combinação proposta do sinal de voz usado no capítulo 5 no domínio *wavelet*. O intuito é que o leitor possa visualizar as formas de onda, pois elas mantêm a mesma característica no domínio *wavelet*.

E, por fim, o Apêndice G é apresentada uma lista de artigos resultantes dos trabalhos desenvolvidos em vínculos científicos com esta tese, até o momento.

1.3 Contribuições da Tese

As principais contribuições da tese são:

1. Uma nova abordagem na aplicação de *wavelets* na redução de ruído em sinais de voz, o que elimina o uso de limiares;
2. Um processamento mais uniforme do sinal ruidoso, o que evita descontinuidades ao longo do processo de redução de ruído.

CAPÍTULO 2

REDUÇÃO DE RUÍDO EM SINAIS DE VOZ

Introdução

Em muitos sistemas de comunicação, a presença ou interferência de ruído de fundo pode causar a degradação da qualidade ou inteligibilidade de um sinal de voz (DELLER; PROAKIS; HANSEN, 1993). Do ponto de vista de acústica, ruído é qualquer sinal que tem a capacidade de reduzir a inteligibilidade de uma informação de som (NOCETI FILHO, 2004). De um modo geral, pode ser definido também como sendo um som indesejável (FERNANDES, 2005).

Se não fossem os ruídos, um sinal desejado poderia ser amplificado por um conjunto de amplificadores e/ou filtros de alto ganho em cascata, de modo que informações de reduzidíssima energia poderiam ser detectadas sem problemas. Porém, quando um sinal é amplificado, o ruído também é amplificado.

Por outro lado, os ruídos também têm seu lado útil, pois devido a sua riqueza espectral, alguns tipos de ruídos servem de fonte para a síntese da fala, de inúmeros sons da natureza e de sons de instrumentos musicais. Além disso, por conter um grande número de frequências, alguns tipos de ruídos foram padronizados, sendo úteis na calibração de equipamentos eletrônicos, como sinais de teste, e nas medidas das

características de filtros, amplificadores, sistemas de áudio eletroacústicos e outros sistemas.

O nível de influência de um ruído nos sistemas eletrônicos é apresentado de várias formas. Uma das mais importantes é a razão entre a potência do sinal desejado e a potência do ruído ou, simplesmente, a relação sinal/ruído (SNR). Outra forma é a caracterização de um sistema e não de um sinal, chamada de faixa dinâmica (DR). A DR exprime a razão entre o máximo sinal desejado que o sistema admite e o mínimo sinal desejado detectável, que é o nível do ruído apresentado quando não existe sinal. Essas duas grandezas são geralmente representadas em decibéis (dB) (NOCETI FILHO, 2004).

Empiricamente, sabe-se que quando a comunicação é feita num ambiente silencioso, a troca de informação é feita de maneira fácil e precisa. Entretanto, em ambiente ruidoso, como por exemplo, no interior de uma fábrica, os ouvintes têm uma redução na sua habilidade auditiva, o que significa uma dificuldade no que é dito.

Além disso, na comunicação interpessoal, o sinal é transmitido por telefone, canais de rádios, alto-falantes ou fones de ouvido, sofrendo a influência da conversão dos dados (microfone), transmissão (ruído através dos canais) ou reprodução (alto-falantes ou fones de ouvido).

Pode-se dizer que o objetivo principal de um algoritmo de redução de ruído em sinais de voz é melhorar a qualidade do sinal, seja reduzindo o ruído de fundo, seja suprimindo o ruído de um canal ou interlocutor gerador de interferência (DELLER; PROAKIS; HANSEN, 1993). Conseqüentemente, a redução de ruído em sinais de qualquer dimensão é importante, devido ao fato de que, em aplicações práticas, a maioria dos sinais é obtida em ambientes ruidosos. Por isso, para análise destes sinais é necessário que, primeiro, faça-se a eliminação ou redução (atenuação) ao máximo do ruído presente.

A distinção entre sinais e ruídos depende do modelo real do sinal medido. Em outras palavras, isto depende da relação assumida entre o sinal e o fenômeno apresentado pelo sinal. Assim, dependendo da natureza do ruído e do sinal, existem vários métodos para eliminar ou atenuar ruídos (LOUIS; MAAB; RIEDER, 1998).

O procedimento básico de eliminação de ruído normalmente empregado para processamento em blocos, com análise via janelamento e sobreposição de 50 %, pode ser resumidamente descrito como a seguir:

- tome um quadro do sinal de entrada (ruidoso);

- calcule a sua transformada *wavelet*;
- aplique uma função de limiar (*threshold*) aos coeficientes de detalhe;
- calcule a transformada *wavelet* inversa destes coeficientes;
- finalmente, use a parte central do quadro para compor a saída;
- repita os procedimentos anteriores para todos os blocos.

Geralmente, o limiar é aplicado apenas aos coeficientes de detalhes, deixando inalterados os coeficientes de aproximação na escala mais grossa, pois se considera que estes representam a estrutura básica do sinal original. De maneira equivalente, pode-se fazer a decomposição *wavelet* completa (até que o último nível contenha um único coeficiente de detalhe) e aplicar a função de limiar aos coeficientes de detalhe abaixo de um dado nível da decomposição (IRINEU, 2006).

Particularmente, no caso de sinais de voz, o método deve preservar os coeficientes que representam os sons do tipo “não vozeado”. Para este fim, muitos métodos de redução de ruído dividem o trecho processado em segmentos “vozeado” e “não vozeado”. Em seguida, tratam os segmentos de maneira distinta, segundo diferentes abordagens concebidas de maneira *ad hoc* por meio de experimentação computacional. Pode-se verificar isto através de dois exemplos (IRINEU, 2006):

1) Se for encontrada uma região “não vozeado”, faz-se o *threshold* apenas da banda de frequências mais altas. Caso contrário, aplica-se o *threshold* a todos os coeficientes (SEOK; BAE, 1997).

2) Aplica-se *Soft threshold* aos segmentos “não vozeados” e *Hard threshold* aos demais, utilizando-se diferentes valores de limiar para cada tipo de segmento (KIM; YANG; KWON, 2001).

Após a aplicação de um método de redução de ruído, uma medida de qualidade deve ser usada para verificar a eficiência deste método. Uma medida de qualidade é um processo subjetivo ou objetivo aplicado ao sinal processado que faz comparações entre este sinal e o sinal de referência, geralmente o sinal original, sem a presença de ruído.

A seguir são apresentadas as principais medidas de qualidade usadas para avaliação de desempenho dos algoritmos de redução de ruído.

2.1 Medidas de Qualidade

Por décadas históricas, ao longo da análise do processamento de sinais, estabeleceram-se meios de comparação e avaliação de algoritmos de melhoria em sinais de voz. Assim, uma variedade de técnicas de avaliação de qualidade foi formulada.

De um modo geral, os testes podem agrupar-se em duas classes: medidas de qualidade subjetivas e medidas de qualidade objetivas.

Medidas subjetivas estão baseadas em comparações dos dados dos sinais de voz, original e processado, por um ouvinte ou um grupo de ouvintes, que subjetivamente classificam a qualidade da voz ao longo de uma escala predeterminada (DELLER; PROAKIS; HANSEN, 1993).

Medidas de qualidade objetivas estão baseadas em uma comparação matemática dos sinais de voz, original e processado. A maioria das medidas de qualidade objetivas quantificam a qualidade com uma medida de distância numérica ou um modelo de como o sistema auditivo interpreta a qualidade (DELLER; PROAKIS; HANSEN, 1993).

2.1.1 Medidas de Qualidade Subjetivas

Conforme definido anteriormente, medidas subjetivas estão baseadas na opinião de um ouvinte ou de um grupo de ouvintes na qualidade de uma expressão vocal. Segundo Hecker e Williams (1966), o meio de se classificar as medidas de qualidade subjetivas é agrupá-las como utilitárias ou analíticas.

As medidas utilitárias empregam procedimentos de teste que são eficientes e seguros e que produzem uma medida de qualidade de voz em uma escala unidimensional. Como principal vantagem tem-se o resultado em um único número que pode ser usado diretamente para comparar sistemas de processamento de voz (DELLER; PROAKIS; HANSEN, 1993).

Por sua vez, os métodos analíticos buscam identificar os componentes psicológicos subjacentes que determinam a qualidade perceptiva. Estes métodos estão orientados mais para caracterizar a percepção da voz do que para medir a qualidade de percepção e usa-se mais de uma dimensão para informar os seus resultados (DELLER; PROAKIS; HANSEN, 1993).

Além disso, as medidas subjetivas tinham como foco a inteligibilidade da voz, fato este considerado um aspecto importante da qualidade de voz global. Formularam-se vários testes como o *teste de rima modificado* (MRT) que consiste em listas de palavras com rima; o *teste de rima diagnóstico* (DRT) onde são apresentados aos ouvintes palavras de rima que só diferem nos seus fonemas consonantais principais. A qualidade do sistema de processamento de voz está baseada no número e, possivelmente, na facilidade dos ouvintes identificarem as respostas corretas (DELLER; PROAKIS; HANSEN, 1993).

Outro teste é o *teste de qualidade do julgamento absoluto isométrico* (IAJ) onde uma lista de palavras distorcidas é apresentada aos ouvintes e pede-se para julgar a qualidade geral sem qualquer sinal de referência. Aqui, constata-se uma enorme vantagem, pois os ouvintes são livres para nomear qualquer significado perceptivo em suas escalas, de excelente a insatisfatório. A desvantagem destes testes é que pode haver variações ou influências entre os ouvintes, o que significa que um ouvinte pode sempre classificar uma voz distorcida a algum nível abaixo que outro ouvinte (DELLER; PROAKIS; HANSEN, 1993).

Assim, conclui-se facilmente a dependência do grupo de ouvintes, ou seja, diferentes grupos de ouvintes podem classificar diferentemente os mesmos sinais. Além disso, estes procedimentos requerem muito tempo para sua execução considerando ainda que cheguem a resultados, por vezes, insatisfatórios.

Não se pode, no entanto, menosprezar a importância dos testes subjetivos; eles fazem parte da metodologia mais completa que existe na avaliação de sistemas de melhoramento de sinais de voz.

Assim, propostas de avaliações subjetivas são apresentadas abaixo, como sendo dois testes, simples e suficientes:

- Apresentam-se aos ouvintes os sinais originais e o processado, que devem opinar se os sinais são iguais, se o sinal processado é melhor ou pior que o original.
- Após a aplicação de vários métodos de melhoria em sinais de voz, as frases processadas são apresentadas aos ouvintes pelo menos duas vezes em ordem diferenciada; o ouvinte deverá apenas informar qual é a melhor frase.

Uma metodologia completa para avaliação subjetiva foi padronizada pela ITU (International Telecommunication Union) e é denominada de MOS (Mean Opinion Score) (ITU-T, P.800) (ITU-T, 1996).

2.1.2 Medidas de Qualidade Objetivas

O critério de desempenho para uma medida de qualidade objetiva de voz é sua correlação com estimativas de qualidade subjetivas. Para obter o coeficiente de correlação, ambas as medidas, subjetiva e objetiva, devem ser aplicadas para uma base de dados processados. Uma análise de correlação é aplicada para determinar a habilidade das medidas de qualidade objetivas para predizer a qualidade julgada pelos ouvintes na avaliação subjetiva.

Tais procedimentos de avaliação foram executados em laboratórios de pesquisa usando bancos de dados extensos, incorporando uma variedade de distorções através de decodificadores de voz (BARNWELL; VOIERS, 1979; BARNWELL et al., 1984; BARNWELL, 1985; QUACKENBUSH, 1985; VOIERS, 1977). A avaliação através de medidas de qualidade objetiva proporciona meios quantitativos, repetitivos e precisos de se comparar o desempenho de métodos de melhoria de sinais de voz. Todas as medidas objetivas fazem a comparação direta entre a versão da forma de onda original (ou referência) e a processada (re-sintetizada ou melhorada) (DELLER; PROAKIS; HANSEN, 1993). Por isso, é necessário que, no processo de cálculo do algoritmo, as formas de onda dos sinais de voz estejam sincronizadas.

O método mais comum de se implementar uma medida objetiva consiste em dividir o sinal de voz em janelas de curta duração e, então, calcular a medida de distância/distorção de cada uma das janelas. A maioria das medidas faz comparações entre as características espectrais dos sinais originais e processado. Uma medida final é formada por meio da combinação das medidas de distorção de cada janela ou através da observação dos valores obtidos ao longo do processamento, apresentados em um gráfico de distâncias/distorção.

Várias medidas objetivas de qualidade são propostas e a mais comum e simples delas é a relação Sinal-Ruído (SNR).

2.1.2.1 A Medida Relação Sinal Ruído

A relação Sinal-Ruído (SNR) é a medida mais amplamente utilizada em sistemas analógicos de codificação de forma de onda. Sequencialmente, ao longo dos anos, foram desenvolvidas diferentes variações, incluindo a *SNR clássica*, *SNR*

segmentada, a *SNR segmentada ponderada na frequência*, entre outras (DELLER; PROAKIS; HANSEN, 1993).

A principal vantagem da medida de qualidade SNR está na simplicidade matemática. A medida representa um erro médio sobre o tempo e a frequência para um sinal processado.

A SNR é dada por (DELLER; PROAKIS; HANSEN, 1993):

$$SNR = 10 \log_{10} \left(\frac{\sigma_x^2}{\sigma_e^2} \right) \quad (2.1)$$

sendo σ_x^2 a variância do sinal original e σ_e^2 a variância do erro, calculada como a diferença entre o sinal original e o sinal processado.

O sistema de audição humana é relativamente insensível a distorção de fase (WANG; LIM, 1982). Isto constitui uma importante particularidade em muitos sistemas de comunicação de voz (por exemplo, telefone ou rádio) e em muitos algoritmos de melhoria. Por isso, muitos sistemas de codificação e melhoria focam apenas na magnitude do espectro de voz. Assim, a codificação ou a melhoria da forma de onda pode ser bastante diferente do original, podendo ser percebida similarmente pelo ouvinte.

Medidas como a SNR, que obtém uma medida de distorção baseada em diferenças amostra-por-amostra nas formas de onda dos sinais, original e processado, não fornecem uma medida significativa de desempenho quando as duas formas de onda diferem na fase do espectro. Assim, fazem-se necessárias medidas de distância que são sensíveis às variações na magnitude do espectro de voz.

2.1.2.2 Outras Medidas Objetivas Baseadas em Distância

Uma das mais bem sucedidas medidas de distância é a de *Itakura-Saito* (DELLER; PROAKIS; HANSEN, 1993; GRAY Jr.; MARKEL, 1976). Esta medida é baseada na diferença entre modelos do sinal de referência e do sinal processado. A medida de distância é calculada entre conjuntos de parâmetros de Predição Linear (LP) estimados a partir de blocos sincronizados (a cada 16 ms a 32 ms) dos dois sinais.

Outra medida muito usada é a distância cepstral (DELLER; PROAKIS; HANSEN, 1993; GRAY Jr.; MARKEL, 1976) que representa uma distância euclidiana ponderada entre os sinais original e processado, usando como base seus cepstros.

Além dessas medidas, existem outras mais sensíveis, capazes de ultrapassar a capacidade do ouvido humano na detecção de ínfimas diferenças, chamadas medidas psicoacústicas, que fazem avaliações baseadas em características do ouvido humano (FLANAGAN, 1972; HOWARD; ANGUS, 1996; SCHROEDER; ATAL; HALL, 1979; TSOUKALAS; PARASKEVAS; MOURJOPOULOS, 1993).

Uma das medidas baseadas em critérios psicoacústicos, que tem sido amplamente usada para avaliação de sistemas de melhoria de voz, é a PESQ (*Perceptual Evaluation of Speech Quality*), proposta pela ITU (*International Telecommunications Union*) através da recomendação P.862 de fevereiro de 2001 (BEERENDS et al., 2002; ITU-T, 2001; RIX et al., 2000).

A avaliação feita através da PESQ é baseada em características do ouvido humano, onde os sinais originais e processado são comparados com relação a atraso, distúrbios simétricos e assimétricos (BEERENDS et al., 2002; RIX et al., 2000). A pontuação final de qualidade, emitida pelo algoritmo da PESQ, é obtida calculando a distância entre os sinais original e processado. A pontuação varia entre -0,5 e 4,5, embora, na maioria dos casos, a faixa de saída assuma valores semelhantes aos observados nos testes subjetivos do tipo MOS (*Mean Square Opinion*) (DELLER; PROAKIS; HANSEN, 1993), entre 1,0 e 5,0, sendo esta a faixa normal de valores alcançados em experimentos ACR (*Absolute Category Rating Method*) (RIX, et al., 2000). Assim, quanto maior for a semelhança entre os sinais comparados, maior será o valor da pontuação obtida.

Não existe nenhuma referência na recomendação P.862 ou em qualquer artigo especializado, mas os profissionais que trabalham com equipamentos de medição PESQ, atribuem ao valor 3 a menor pontuação para um sinal de boa qualidade. Valores PESQ menores do que 3 são considerados de baixa qualidade, não sendo adequados ao uso corporativo ou comercial, exceto quando não se tem opção por melhor qualidade ou se faça uma escolha consciente por solução de baixo custo (BEERENDS et al., 2002).

CAPÍTULO 3

ANÁLISE WAVELET

Introdução

Embora a primeira menção às *wavelets* tenha acontecido em 1910 em uma tese de *Alfred Haar* (HAAR, 1910), as *wavelets* de *Haar* ficaram no anonimato por muitos anos e, por um período muito longo, elas continuaram a ser a única base ortonormal de *wavelets* conhecida (SOARES, 2004). A forma teórica atual da análise *wavelet* foi apresentada em meados dos anos oitenta por *Jean Morlet*, *Yves Meyer* e *Alex Grossman* que faziam parte da equipe do Centro de Física Teórica de Marseille na França. No final dos anos oitenta, *Stephane Mallat* deu às *wavelets* um grande impulso ao fazer a ligação entre *wavelets* e processamento de sinais (MALLAT, 1989(A)). O algoritmo de *Mallat*, apresentado em *Mallat* (1989(A)), pode ser considerado como o início da aplicação das *wavelets* na área de processamento de sinais e, inspirado nos resultados de *Mallat*, *Y. Meyer* construiu a primeira *wavelet* não-trivial (suave) (MEYER, 1993). Ao contrário das *wavelets* de *Haar*, que não são continuamente diferenciáveis, o que de certo modo limita suas aplicações, as *wavelets* de *Meyer* são continuamente diferenciáveis; contudo, essas funções não têm suporte compacto. Poucos anos mais tarde, a atenção da comunidade de processamento de sinais foi atraída quando *Ingrid Daubechies* (DAUBECHIES, 1998; DAUBECHIES, 1990; DAUBECHIES, 1992) usou os trabalhos

de *Mallat* (MALLAT, 1989(A); MALLAT, 1989(B)) para construir uma base ortonormal de *wavelets suaves* (funções que têm todas as derivadas contínuas), com suportes compactos, que são, talvez, as mais elegantes e que se tornaram um marco nas aplicações de *wavelets*. Além de suas contribuições para a teoria *wavelet*, estabeleceram a conexão entre os dois assuntos e obtiveram resultados via processamento de sinal discreto. Os trabalhos de *Daubechies* são os alicerces das aplicações atuais de *wavelets*.

O princípio mais geral na construção das *wavelets* é o uso de dilatações e translações (DAUBECHIES, 1992). As *wavelets* mais usadas formam um sistema ortonormal de funções de suportes compactos (DAUBECHIES, 1992). Esta é a razão pela qual elas podem distinguir as características locais de um sinal em diferentes escalas e, por translações, elas cobrem toda a região na qual o sinal é estudado. Na análise de sinais não-estacionários, a propriedade de localidade das *wavelets* conduz às suas vantagens sobre a transformada de Fourier (DAUBECHIES, 1992).

Os algoritmos *wavelets* processam dados em diferentes escalas ou resoluções e, independentemente da função de interesse ser uma imagem, uma curva ou uma superfície, as *wavelets* oferecem uma técnica elegante na representação dos níveis de detalhes presentes. Elas constituem uma ferramenta matemática para decompor funções hierarquicamente, permitindo que uma função seja descrita em termos de uma forma grosseira, e uma outra forma que apresenta detalhes que vão desde os menos delicados aos mais finos. O objetivo na análise *wavelet* é “ver a floresta e as árvores” (LIMA, 2004), isto é, o sinal e seus detalhes.

Por causa de suas propriedades inerentes, as *wavelets* se desenvolveram nos campos da Matemática, da Engenharia e da Física Quântica. Hoje vêm sendo incluídas em uma larga gama de aplicações: geologia sísmica, processamento de imagens (compressão de vídeo), visão computacional e humana, radar e sonar, espectrometria, computação gráfica, predição de terremotos e maremotos, turbulência, análise de transitórios em linha de potência, fractais, bancos de filtros, distinção celular (células normais versus patológicas), caracterização de sinais acústicos, física quântica, modelos para trato auditivo, compressão de imagens (por exemplo, o padrão *JPEG2000 Still Image Compression* é baseado em *wavelets*, o padrão do FBI para armazenamento de impressões digitais (BRADLEY; BRISLAWN, 1994), (MEENEN; ADHAMI, 2001), (TICO; KUOSMANEN; SAARINEN, 2001), (FBI *Wavelet /Scalar Quantization Fingerprint Image Compression Standard* <http://www.c3.lanl.gov/~brislawn/FBI/FBI.html>), descontaminação de sinais

(*denoising*), detecção de rupturas e bordas, análise de tons musicais, neurofisiologia, detecção de curtos eventos patológicos (por exemplo, crises epiléticas) e análise de sinais biomédicos (eletrocardiogramas, mamografias digitais, eletroencefalogramas, sequências de DNA etc.), espalhamento em banda larga, modelagem de sistemas lineares, óptica e eletromagnetismo, modelagem geométrica, reconhecimento de alvos, hidrodinâmica, análise de transitório e falhas em linhas de potência, Metalurgia (rugosidade de superfícies), visualização volumétrica, Telecomunicações (incluindo espalhamento espectral), previsão de comportamento de mercados financeiros, estatística, solução de equações diferenciais ordinárias e parciais, não sendo esta lista de aplicações nem de longe exaustiva (OLIVEIRA, 2007).

3.1 Wavelets

Wavelets são ondas pequenas (*ondelettes*, em francês) com determinadas propriedades que as tornam adequadas a servirem de base para a decomposição de outras funções, assim como senos e co-senos servem de base para decomposições usando séries de Fourier (DAUBECHIES, 1992).

A novidade em relação à análise de Fourier é que a análise *wavelet* não é feita segundo a frequência, mas sim segundo a escala. Os algoritmos usados no domínio *wavelet* processam dados em diferentes escalas e resoluções, permitindo que sejam vistos tanto o global quanto os detalhes de um sinal (GRAPS, 1995).

É denotado por $L^2(R)$ o espaço das funções de quadrado integrável ou, ainda, o espaço das funções de energia finita, isto é, se $\psi(t) \in L^2(R)$, então:

$$\int_{-\infty}^{\infty} |\psi(t)|^2 dt < \infty \quad (3.1)$$

Definição: Uma função $\psi(t) \in L^2(R)$ é denominada *wavelet* se sua transformada de Fourier $\hat{\psi}(\omega)$ satisfaz a condição (MORETTIN, 1999):

$$C_{\psi} = \int_{-\infty}^{\infty} \frac{|\hat{\psi}(\omega)|^2}{|\omega|} d\omega < \infty \quad (3.2)$$

A condição anterior, equação (3.2), é chamada de condição de admissibilidade (DAUBECHIES, 1992). Segue da condição da admissibilidade que $\lim_{\omega \rightarrow 0} \hat{\psi}(\omega) = 0$.

Assim, se $\hat{\psi}(\omega)$ é contínua, então $\hat{\psi}(0) = 0$, ou seja:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (3.3)$$

Geometricamente, a equação (3.3) estabelece que o gráfico de $\psi(t)$ deve oscilar de modo a cancelar as áreas positivas e negativas a fim de anular a integral. Portanto, o gráfico de $\psi(t)$ tem a forma de uma onda, conforme ilustra a Figura 3.1, que é um exemplo de *wavelet*.

Como $\psi(t) \in L^2(\mathbb{R})$, significando que esta função tem energia finita, então se conclui que sua energia está localizada em certa região e que esta função tende a oscilar acima e abaixo do eixo t . Assim, ela terá a forma de uma onda de curta duração. Por essa razão a função $\psi(t)$ que satisfaz a equação (3.2) é chamada *wavelet* (do inglês: onda pequena) (MORETTIN, 1999).

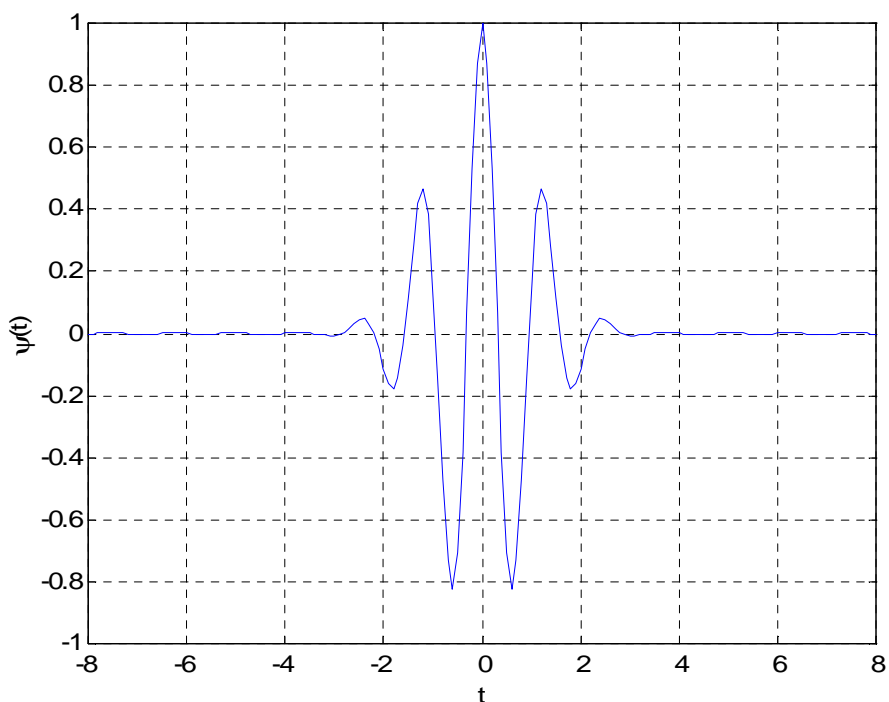


Figura 3.1 – *Wavelet* de Morlet.

3.2 Funções *Wavelets*

Geralmente, a função *wavelet* mãe recebe o nome do seu criador e, atualmente, existem inúmeras *wavelets* mãe citadas na literatura. O uso de uma ou outra está associado à aplicação. Regras de construção de *wavelets* são propostas por vários

pesquisadores, segundo as restrições e necessidades que cada aplicação específica impõe (SWELDENS; SCHÖDER, 2007). Conclui-se que pode-se gerar uma infinidade de *wavelets* diferentes e, particularmente, construir um conjunto de *wavelets* adequado ao processamento de um tipo de sinal ou aplicação específica, levando à obtenção de resultados melhores.

Geralmente uma função *wavelet* $\psi(t)$ está associada a uma função escala $\phi(t)$. A *wavelet* é gerada através de translações e dilatações dessa função escala. Um exemplo é função escala de *Haar* definida na equação (3.4). A *wavelet* de *Haar*, obtida através de $\phi(t)$ é dada por $\psi(t)=\phi(2t)-\phi(2t-1)$ e é expressa na equação (3.5). Apesar da importância da função escala, nem toda função *wavelet* está associada a uma função escala, podemos citar como exemplo, a *wavelet* de *Morlet* apresentada na equação (3.10). Pode-se dizer que as *wavelets* que possuem função escala são mais aplicáveis no domínio discreto, ao passo que, as *wavelets* que não possuem função escala, suas aplicações são mais frequentes no processamento em tempo contínuo.

A seguir, são apresentadas as *wavelets* mais conhecidas.

3.2.1 Wavelet de Haar

É a função *wavelet* mais simples. Suponha que $\phi(t)$ é a função definida por:

$$\phi(t)=\begin{cases} 1, & \text{se } 0 \leq t \leq 1 \\ 0, & \text{caso contrário} \end{cases} \quad (3.4)$$

assim, $\phi(t)$ é uma função escala. Uma função escala é a função inicial para se construir uma *wavelet*. A *wavelet* é construída através de operações de mudança de escala e translação aplicada na função $\phi(t)$.

Definindo a função $\psi(t)$ como $\psi(t)=\phi(2t)-\phi(2t-1)$, obtém-se a seguinte expressão para essa função:

$$\psi(t)=\begin{cases} 1, & \text{se } 0 \leq t \leq \frac{1}{2} \\ -1, & \text{se } \frac{1}{2} < t \leq 1 \\ 0, & \text{caso contrário} \end{cases} \quad (3.5)$$

A função $\phi(t)$ é a função escala de *Haar* e $\psi(t)$ é a *wavelet* de *Haar*, cujos gráficos são apresentados na Figura 3.2. Esta função é ortogonal às suas translações e dilatações, isto é, à família:

$$\psi_{m,n}(t) = 2^{-m/2} \psi(2^{-m}t - n), \quad m, n \in \mathbb{Z} \quad (3.6)$$

sendo \mathbb{Z} o conjunto dos números inteiros. Esta família constitui uma base ortonormal no espaço $L^2(\mathbb{R})$. Historicamente, as translações e dilatações de uma função de *Haar* formam a *wavelet* original.

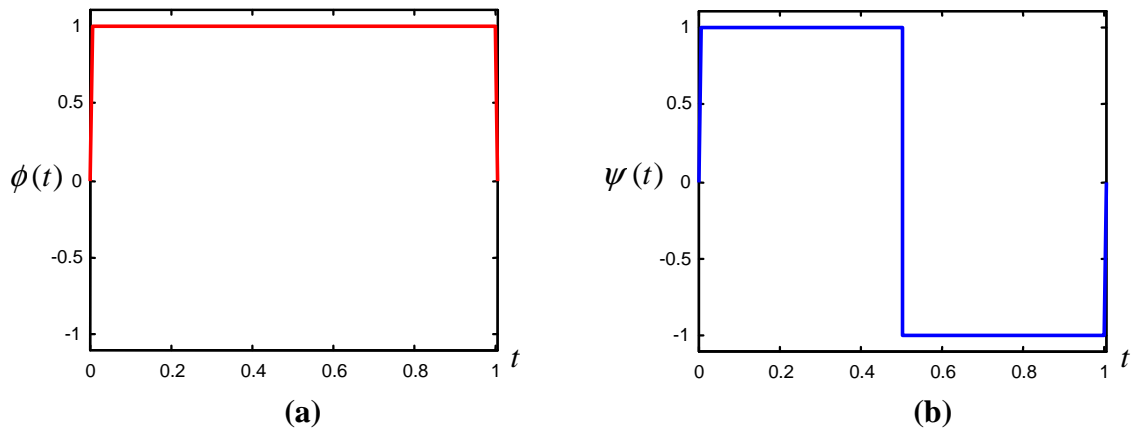


Figura 3.2 – (a) Função escala de *Haar*; (b) Função *wavelet* de *Haar*.

3.2.2 Wavelet de Meyer

As *wavelets* de *Haar* não são continuamente diferenciáveis, o que de certo modo limita suas aplicações. Após 1980 *Yves Meyer* construiu a primeira *wavelet* trivial diferente da *wavelet* de *Haar*, que é continuamente diferenciável (SILVA; EYNG, 2000). Desta forma, uma base *wavelet* suave ortonormal foi criada. Primeiro, definiu-se a Transformada de Fourier $\hat{\phi}(\omega)$ de uma função escala $\phi(t)$ satisfazendo:

$$\hat{\phi}(\omega) = \begin{cases} (2\pi)^{-1/2}, & \text{se } |\omega| \leq \frac{2\pi}{3} \\ (2\pi)^{-1/2} \cos\left[\frac{\pi}{2} \nu\left(\frac{3}{2\pi}|\omega| - 1\right)\right], & \text{se } \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3} \\ 0, & \text{se } |\omega| > \frac{4\pi}{3} \end{cases} \quad (3.7)$$

onde

$$v(a) = a^4(35 - 84a + 70a^2 - 20a^3), \quad a \in [0,1] \quad (\text{MISITI et al., 1996}) \quad (3.8)$$

Deste modo, a função *wavelet* $\psi(t)$ pode ser encontrada facilmente através de $\phi(t)$:

$$\hat{\psi}(\omega) = \begin{cases} (2\pi)^{-1/2} e^{j\omega} \operatorname{sen} \left[\frac{\pi}{2} v \left(\frac{3}{2\pi} |\omega| - 1 \right) \right], & \text{se } \frac{2\pi}{3} \leq |\omega| \leq \frac{4\pi}{3} \\ (2\pi)^{-1/2} e^{j\omega} \operatorname{cos} \left[\frac{\pi}{2} v \left(\frac{3}{4\pi} |\omega| - 1 \right) \right], & \text{se } \frac{4\pi}{3} \leq |\omega| \leq \frac{8\pi}{3} \\ 0, & \text{caso contrário} \end{cases} \quad (3.9)$$

A Figura 3.3 ilustra os gráficos das funções definidas por (3.7) e (3.9), respectivamente.

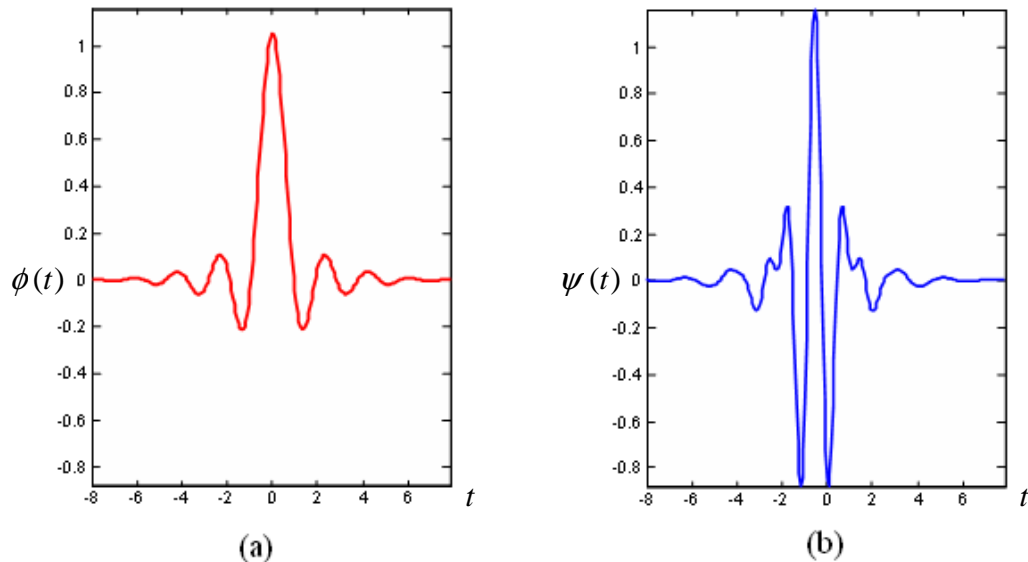


Figura 3.3 – (a) Função escala de *Meyer*; (b) Função *wavelet* de *Meyer*.

3.2.3 *Wavelet de Morlet*

Outra função é a *wavelet de Morlet*, introduzida por *Jean Morlet*, pertencente à família das *wavelets* não-ortogonais. A *wavelet de Morlet* não possui função escala e é explicitada por uma função Gaussiana modulada (*shifted*), levemente ajustada. De forma que $\psi(0) = C$, conforme a equação (3.10), cujo gráfico é apresentado na Figura 3.4:

$$\psi(t) = C e^{-t^2/2} \cos(5t) \quad (\text{MISITI et al., 1996}) \quad (3.10)$$

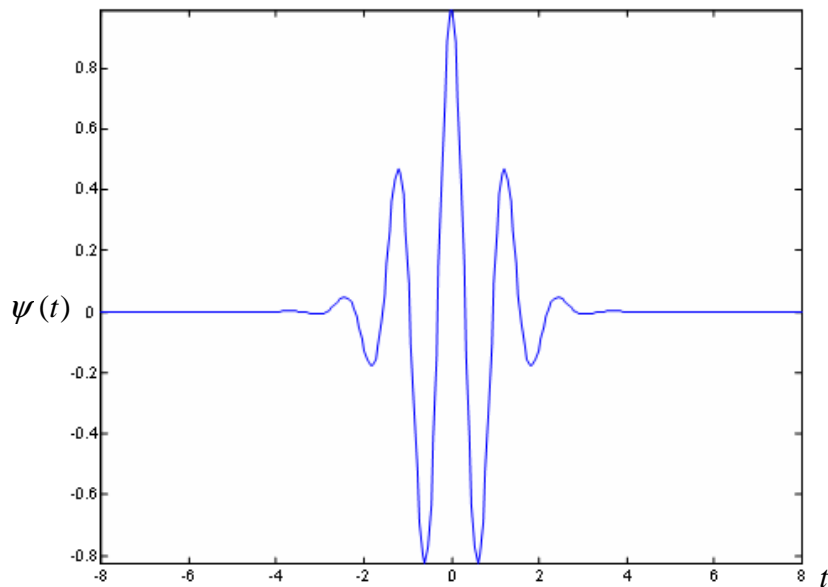


Figura 3.4 – Função *wavelet* de *Morlet*.

3.2.4 Wavelets de Daubechies

Com exceção da base de *Haar*, todos os exemplos de bases *wavelets* ortonormais consistem de funções de suporte infinito. *Ingrid Daubechies* construiu uma família de *wavelets* ortonormais, todas com suporte compacto. A forma de garantir que a *wavelet* $\psi(t)$ tenha suporte compacto é escolher a função escala $\phi(t)$ com suporte compacto. As *wavelets* de *Daubechies* são funções regulares (DAUBECHIES, 1992).

A geração de tais funções passa pela obtenção da respectiva função de escala (DAUBECHIES, 1992):

$$\phi(t) = \sum_{k \in \mathbb{Z}} \alpha_k \phi(2t - k) \quad (3.11)$$

Os parâmetros α_k são geralmente designados como coeficientes de filtro. É através da imposição de determinadas condições, sobre os valores que estes coeficientes podem assumir, que se conseguem obter determinadas propriedades para um conjunto de funções *wavelet*, tal como, por exemplo, a ortonormalidade.

Quando se pretende que as *wavelets* tenham suporte compacto, isto é, que sejam definidas sobre um intervalo limitado, é necessário que apenas um número finito de coeficientes de filtro seja diferente de zero. As *wavelets* de *Daubechies* são

organizadas em diferentes famílias, cada uma das quais caracterizadas por um número diferente de coeficientes α_k não-nulos. Cada família é identificada pelo seu número de família, denotado por N , e cujo valor é igual à metade do número de coeficientes de filtro utilizados na equação (3.11). Por isso alguns autores usam o termo dB2 para simbolizar a *wavelet* dBN.

A função escala descrita na equação (3.11) pode então ser escrita da seguinte forma:

$${}_N\phi(t) = \sum_{k=0}^{2N-1} \alpha_k {}_N\phi(2t-k) \quad (3.12)$$

As funções escalas assim definidas assumem valores diferentes de zero apenas no intervalo $[0, 2N - 1]$. A este intervalo costuma chamar-se suporte da função escala ${}_N\phi(t)$. Pode-se escrever ainda como sendo:

$$\text{supp } {}_N\phi(t) = [0, 2N - 1] \quad (3.13)$$

Daubechies (1988) definiu em seu artigo as famílias *wavelets* com valores de N compreendidos entre 2 e 10.

Sendo assim, dada uma família com um determinado N , qualquer polinômio de grau igual ou inferior a $N - 1$ pode ser representado de uma forma exata pela combinação linear da função escala ${}_N\phi(t)$ e de todas as suas translações inteiras.

A *wavelet* primária de cada uma das famílias associadas pode ser definida através de:

$${}_N\psi(t) = \sum_{k=0}^{2N-1} b_k {}_N\phi(2t-k) = \sum_{k=0}^{2N-1} (-1)^k \alpha_{2N-1-k} {}_N\phi(2t-k) \quad (3.14)$$

A escolha dos coeficientes $b_k = (-1)^k \alpha_{2N-1-k}$ é determinada pela necessidade de garantir a ortonormalidade entre a função escala e a *wavelet* primária.

Na Figura 3.5 são apresentadas as funções escala e *wavelet* de *Daubechies* de ordem 2.

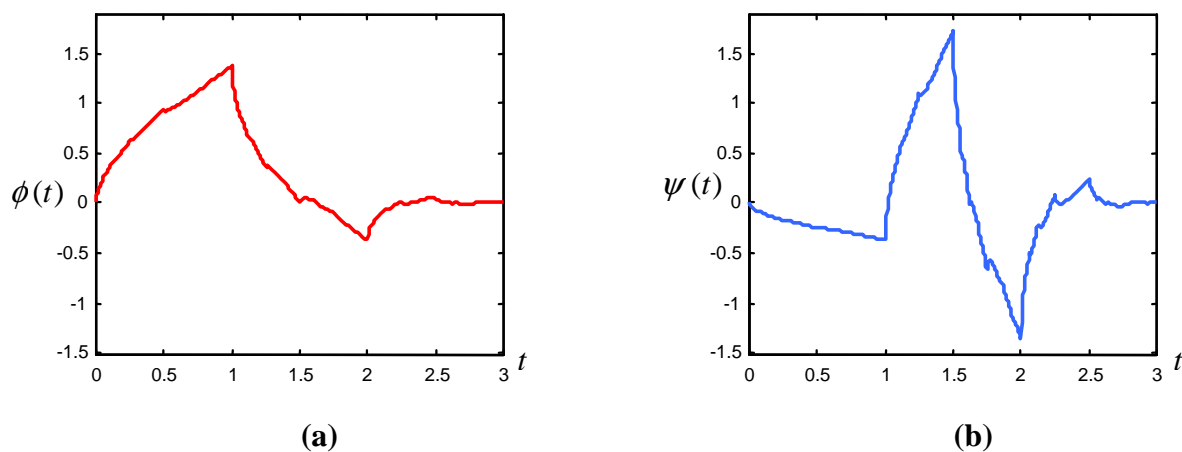


Figura 3.5 – (a) Função escala de *Daubechies*; (b) Função *wavelet* de *Daubechies*.

3.3 Transformada *Wavelet*

A transformada *wavelet* é uma ferramenta de processamento de sinais cujo grande trunfo sobre as técnicas clássicas de Fourier é o fato de que suas funções bases (*wavelets*) não pertencem a um espaço de funções que tem finitas soluções, isto é, existem, teoricamente, infinitas possibilidades de se projetar *wavelets* com propriedades especiais, voltadas para aplicações específicas. As funções moduladoras da transformada *wavelet* podem ser expandidas ou comprimidas com o objetivo de se obter uma correlação mais forte entre o sinal processado e essas funções, evidenciando assim mais detalhes no domínio transformado.

As transformadas *wavelets* podem ser vistas como mecanismos para decompor ou quebrar sinais nas suas partes constituintes, permitindo analisar os dados em diferentes domínios de frequências com a resolução de cada componente amarrada à sua escala.

A transformada *wavelet* pode ser vista como uma técnica de janelamento instantânea, o que corresponde a uma filtragem (DAUBECHIES, 1992; RIOUL; VETTERLI, 1991), muito útil no processamento de sinais não-estacionários.

3.3.1 Transformada *Wavelet* Contínua (CWT)

Supondo que $\psi(t)$ seja uma *wavelet* mãe, a transformada *wavelet* contínua de uma função $f(t) \in L^2(\mathbb{R})$, com relação à *wavelet* mãe $\psi(t)$, é dada por (DAUBECHIES, 1992):

$$W_f(a,b) = \langle f, \psi_{a,b} \rangle = \int_{-\infty}^{+\infty} f(t) |a|^{-p} \psi^* \left(\frac{t-b}{a} \right) dt \quad (3.15)$$

com a e $b \in \mathbb{R}$ e $a \neq 0$.

Na equação (3.15), $\psi^*(t)$ denota o conjugado complexo de $\psi(t)$; a é o parâmetro escala; b é o parâmetro translação e t denota o tempo.

Assim, fazendo uma analogia do fator de escala a com a escala utilizada em mapas, se a escala aumenta, $\psi_{a,b}(t)$ “aumenta” no tempo, capturando comportamentos de tempo longo. Escala maior, “significa” visão global, enquanto que escala menor “significa” visualização de detalhes.

No caso $p = \frac{1}{2}$, a equação (3.15) toma a forma:

$$W_f(a,b) = \langle f, \psi_{a,b} \rangle = \frac{1}{\sqrt{|a|}} \int_{-\infty}^{+\infty} f(t) \psi^* \left(\frac{t-b}{a} \right) dt \quad (3.16)$$

3.3.2 As *Wavelets* Filhas (Famílias de *Wavelets*)

A idéia fundamental na teoria *wavelet* é a operação de escalonamento realizada pelo parâmetro a . O escalonamento possibilita a compressão ($a < 1$) ou dilatação ($a > 1$) da função *wavelet* mãe $\psi(t)$ (SOARES et al., 2005).

A *wavelet* mãe, quando escalonada e deslocada no tempo (translação), dá origem às famílias *wavelets* ou *wavelets* “filhas”:

$$\psi_{a,b}(t) = \frac{1}{\sqrt{|a|}} \psi \left(\frac{t-b}{a} \right) \quad (3.17)$$

O coeficiente $\frac{1}{\sqrt{|a|}}$ torna a energia das *wavelets* filhas a mesma da *wavelet*

mã (DAUBECHIES, 1992).

Na teoria *wavelet*, a *wavelet* mãe deve sempre ser mencionada, pois a representação no domínio *wavelet* é feita em função da *wavelet* mãe escolhida. As considerações estabelecidas para uma transformada são totalmente inválidas para outra transformada, se as *wavelets* mãe utilizadas não forem as mesmas. Deste modo, deve sempre ser usada a expressão: “transformada *wavelet* de uma função $f(t)$ com relação à *wavelet* mãe $\psi(t)$ ” (SOARES et al., 2007 (A)).

A Transformada *Wavelet* Contínua ou simplesmente, “transformada *wavelet*”, pode ter várias interpretações. Pela própria definição, na transformada *wavelet* é usada uma operação de produto interno, o qual pode ser interpretado como uma medida de semelhança entre a função $f(t)$ e cada uma das *wavelets* filhas (DAUBECHIES, 1992). Do ponto de vista de processamento de sinais, a transformada *wavelet* é interpretada como uma operação de filtragem de um sinal $f(t)$ por um filtro cujos coeficientes são gerados pela função *wavelet* que está sendo usada na análise (STRANG; NGUYEN, 1996).

3.3.3 Transformada *Wavelet* Contínua Inversa

De acordo com Young (YOUNG, 1995), a transformada *wavelet* é uma operação de ruptura. Seguindo seu raciocínio, a transformada *wavelet* “quebra” uma função em muitos pedaços e estes pedaços são representados pelos coeficientes *wavelets* $W_f(a,b)$.

Os coeficientes *wavelets* representam a semelhança entre a função $f(t)$ e as *wavelets* filhas, e quanto maior a semelhança, maior será o valor do coeficiente *wavelet* (LIMA, 2004). O conjunto de todos os coeficientes *wavelets* constitui uma representação da função $f(t)$ no domínio *wavelet*.

Se o sinal $f(t)$ for decomposto usando uma *wavelet* $\psi(t)$ que satisfaz a condição de admissibilidade é possível reconstruir o sinal através da transformada *wavelet* inversa definida pela equação:

$$f(t) = \frac{1}{C_\psi} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W_f(a,b) \left\{ \frac{1}{\sqrt{|a|}} \psi \left(\frac{t-b}{a} \right) \right\} \frac{dad b}{a^2} \quad (3.18)$$

Observa-se da equação (3.18) que o mesmo núcleo: $\frac{1}{\sqrt{|a|}}\psi\left(\frac{t-b}{a}\right)$ é utilizado

na transformada *wavelet* contínua e em sua inversa.

De acordo com *Daubechies* (DAUBECHIES, 1992), a equação (3.18) pode ser vista de dois modos diferentes:

- um modo de reconstrução de $f(t)$, desde que sua transformada *wavelet* inversa seja conhecida;
- um modo de representação de $f(t)$, como uma superposição das *wavelets* filhas.

3.3.4 Transformada *Wavelet* Discreta – (DWT)

Embora a transformada *wavelet* contínua seja de grande interesse teórico, principalmente para a obtenção e a compreensão das propriedades matemáticas das funções *wavelets*, a sua discretização é necessária para aplicações práticas.

A necessidade de discretização é resultante da redundância presente na equação (3.17) já que os parâmetros a , b da transformada variam continuamente. Do processo de discretização origina a transformada *wavelet* discreta.

Na transformada *wavelet* discreta apenas os parâmetros da transformada contínua são discretizados, ou seja, o parâmetro de escala a e o parâmetro de translação b . Uma discretização comum é do tipo (HERNANDES; WEISS, 1996):

$$a = a_0^m \tag{3.19}$$

$$b = na_0^m b_0 \tag{3.20}$$

com m e $n \in \mathbb{Z}$, $a_0 > 1$ e $b_0 \neq 0$.

Substituindo a e b na equação (3.16), tem-se a transformada *wavelet* discreta:

$$W_f(m, n) = \frac{1}{\sqrt{a_0^m}} \int_{-\infty}^{+\infty} f(t) \psi\left(\frac{t - na_0^m b_0}{a_0^m}\right) dt \tag{3.21}$$

A partir destas equações têm-se as seguintes observações:

- a transformada *wavelet* discreta é definida apenas para valores de escalas positivos ($a_0 > 1$);

- o passo da translação é proporcional a escala ($b=na_0^m b_0$);
- a transformada *wavelet* discreta produz um conjunto finito de coeficientes *wavelet* $W_f(m, n)$;
- o processamento é realizado sobre tempo contínuo.

3.3.5 Transformada *Wavelet* Discreta Inversa – (IDWT)

No caso contínuo, dada a função *wavelet* mãe, uma função qualquer $f(t)$ pode sempre ser recuperada a partir do seu conjunto de coeficientes *wavelet* contínuos. No caso discreto, entretanto, o processo de reconstrução pode não convergir para a função $f(t)$. A reconstrução depende da escolha da *wavelet* mãe e do processo de discretização realizado.

De acordo com *Daubechies* (DAUBECHIES, 1992), a reconstrução ideal seria aquela que ocorresse com o máximo de eficiência e com um mínimo de perda de informação. Neste sentido, a função $f(t)$ pode ser reconstruída a partir dos seus coeficientes *wavelet* discretos com uma aproximação razoavelmente boa dada por:

$$f(t) \approx c \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} (W_f(m, n)) (\psi_{m,n}(t)) \quad (3.22)$$

sendo, c uma constante que depende do processo de discretização e da *wavelet* mãe utilizada.

3.3.6 Transformada *Wavelet* como um Banco de Filtros

Na maioria das aplicações, a Transformada *Wavelet* Discreta é usada como um banco de filtros em cascata. Essa estrutura será usada neste trabalho também.

A Transformada *Wavelet* é, basicamente, uma operação de filtragem (DAUBECHIES, 1992; NIEVERGELT, 1999), através de um filtro passa baixa $h[n]$ e um filtro passa banda $g[n]$, que decompõem o sinal $c_0[n]$ em $c_1[n]$ e $d_1[n]$, respectivamente. Como os sinais decompostos estão localizados em duas faixas diferentes, tem-se uma decomposição em duas faixas. O sinal $c_1[n]$ é obtido através da convolução de $c_0[n]$ com $h[n]$ mais uma operação de subamostragem de ordem 2; $d_1[n]$

também é obtido por convolução, mas de $c_0[n]$ com $g[n]$ e também uma operação de subamostragem de ordem 2. A Figura 3.6 ilustra este processo. Assim, $c_1[n]$ contém os componentes de baixa frequência e $d_1[n]$ contém os componentes de alta frequência de $c_0[n]$. $c_1[n]$ contém os chamados coeficientes de aproximação e $d_1[n]$, os coeficientes de detalhes. As equações (3.23) e (3.24), expressam as relações entre $c_0[n]$, $c_1[n]$ e $d_1[n]$.

$$c_1[n] = \sum_k h(n-2k)c_0[n] \quad (3.23)$$

$$d_1[n] = \sum_k g(n-2k)c_0[n] \quad (3.24)$$

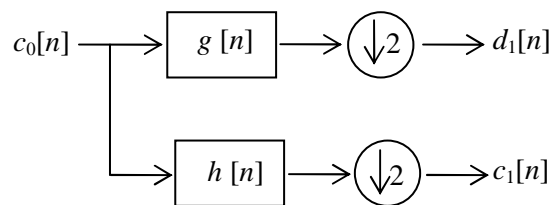


Figura 3.6 – Decomposição em duas faixas de frequência.

Depois de decomposto pela transformada *wavelet*, fazem-se as operações desejadas e o sinal original pode ser reconstruído pela transformada *wavelet* inversa (DAUBECHIES, 1992; NIEVERGELT, 1999). A Figura 3.7 mostra este processo de reconstrução. Primeiro insere-se um zero entre cada elemento dos sinais $c_1[n]$ e $d_1[n]$ (sobreamostragem de ordem 2); depois fazem-se as convoluções de $c_1[n]$ com $h[n]$ e $d_1[n]$ com $g[n]$. O sinal reconstruído é, então, a soma destes dois sinais convoluídos, conforme expresso pela seguinte equação:

$$c_0[n] = \sum_k h(n-2k)c_1[k] + \sum_k g(n-2k)d_1[k] \quad (3.25)$$

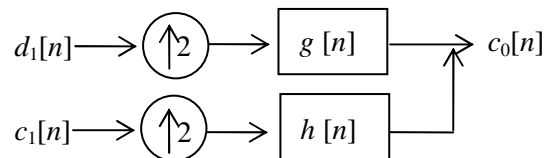


Figura 3.7 – Reconstrução em duas faixas de frequência.

A operação de filtragem em duas faixas pode ser estendida para uma decomposição em k faixas, sendo k o número de faixas de frequências nas quais os sinais decompostos estão localizados. As Figuras 3.8 e 3.9 ilustram, respectivamente, os

processos de decomposição e reconstrução de um sinal em três faixas. O sinal $c_0[n]$ é decomposto em componentes de baixas e altas frequências, $c_1[n]$ e $d_1[n]$; em seguida, $c_1[n]$ é novamente decomposto em componentes de baixas e altas frequências, $c_2[n]$ e $d_2[n]$. Este processo é repetido conforme o número de faixas em que se quer decompor o sinal. Para um sinal com m pontos amostrais, a decomposição pode ser feita em $k=\log_2 m$ faixas, que é o máximo de faixas de decomposições que podem ser usadas (NIEVERGELT, 1999).

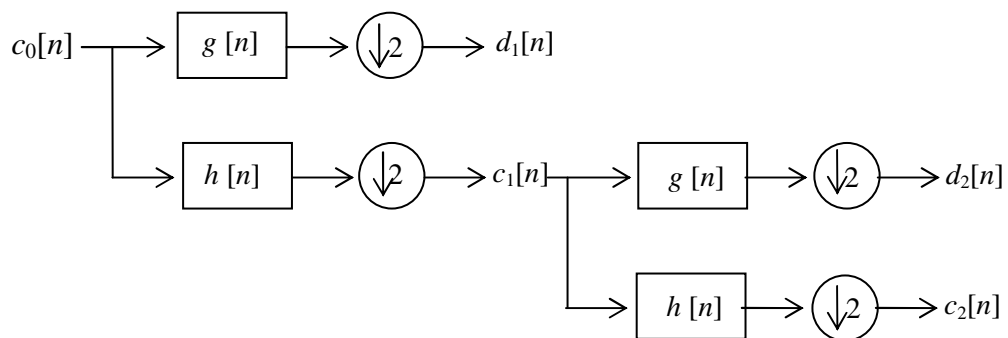


Figura 3.8 – Decomposição em três faixas de frequência.

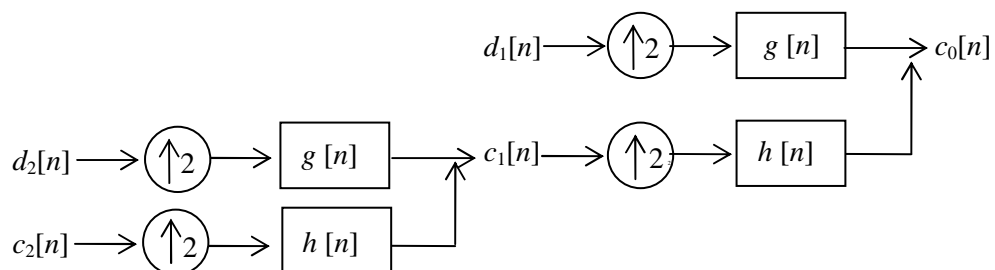


Figura 3.9 – Reconstrução em três faixas de frequência.

Se a função *wavelet* usada no processo for ortogonal, os filtros usados para decomposição e reconstrução são os mesmos, porém em ordem reversa para reconstrução.

CAPÍTULO 4

WAVELETS E O PROCESSAMENTO DE SINAIS DE VOZ

Introdução

A transformada *wavelet* tem sido usada intensamente em vários campos de processamento de sinais. Ela tem a vantagem de usar janelas variáveis no tempo para diferentes faixas de frequência. Isto resulta numa alta resolução nas baixas frequências (e baixa resolução no tempo) e baixa resolução nas frequências mais altas (SARKAR et al., 1998). Conseqüentemente, a transformada *wavelet* é uma ferramenta eficiente para modelar sinais não-estacionários como sinais de voz (STRANG; NGUYEN, 1996). Além disso, quando se está restrito a usar apenas um canal (como a melhoria de um sinal captado por microfone), geralmente o uso de processamento em subfaixas pode resultar num rendimento melhor (SHEIKHZADEH; ABUTALEBI, 2001). Assim, a transformada *wavelet* pode proporcionar um modelo apropriado para remoção (ou redução) de ruído em sinal de voz.

Os métodos que usam limiares apresentam, geralmente, resultados com base em ruídos aditivos do tipo Gaussiano branco (DONOHO, 1995). Porém, a maioria das aplicações envolve sinais com outros tipos de ruído, como os produzidos por carros, aviões, trens, entre outros. Além disso, muitos ruídos são não-estacionários, o que também causa sérias restrições nas aplicações desses métodos. Outro problema que

existe nas aplicações em sinais de voz é a introdução de distorções nos segmentos de voz do tipo “não vozeado”. Nesses segmentos, há conteúdo importante nos componentes de alta frequência, o que requer um processamento específico visando à minimização das distorções. Em alguns métodos (SHEIKHZADEH; ABUTALEBI, 2001; SEOK; BAE, 1997; DUARTE, 2005) o janelamento permite a obtenção de um pequeno segmento de sinal, cuja duração depende da taxa de amostragem. A detecção de pausa permite a implementação de técnicas para a separação entre trechos de voz e silêncio, que são usados na determinação das características básicas do ruído e na implementação de um sistema adequado de identificação de trechos de sinal do tipo “não vozeado” (DUARTE, 2005).

Na sequência são apresentados os assuntos:

- Método *Wavelet* Básico;
- Métodos de Redução de Ruído que Utilizam Limiar;
- Desvantagens dos Métodos que Utilizam Limiar e
- Análise dos Métodos Híbridos Para Redução de Ruído em Sinais de Voz (Usando *Wavelets*).

4.1 Método *Wavelet* Básico

O método *wavelet* básico foi proposto por *Donoho* e *Johnstone* (1994) e é apresentado a seguir.

Considere $y(n)$ uma sequência de observação de comprimento finito do sinal $x(n)$ que está contaminado por ruído aditivo $r(n)$, isto é:

$$y(n) = x(n) + r(n) \quad (4.1)$$

Os sinais $x(n)$ e $r(n)$ são considerados processos aleatórios independentes e o objetivo é recuperar o sinal $x(n)$ a partir de $y(n)$. Se W é a matriz da transformada *wavelet* discreta (DWT) e se W_τ , $\tau = y, x, r$, é a matriz no ambiente *wavelet*, a equação (4.1), que está no domínio do tempo, pode ser escrita no domínio *wavelet* como:

$$Y = X + R$$

sendo:

$$Y = W_y, \quad X = W_x, \quad R = W_r.$$

Seja X_{est} a melhor estimativa do sinal limpo X , baseada na estimação ruidosa Y . No domínio *wavelet*, o sinal limpo x_{est} pode ser estimado por:

$$x_{est} = W^{-1} X_{est} = W^{-1} Y_{thr}$$

sendo que Y_{thr} representa os coeficientes *wavelet* após a redução do ruído com base no uso do limiar (DONOHO; JOHNSTONE, 1994).

O valor do limiar pode ser determinado de várias formas (MORETTIN, 1999). *Donoho e Johnstone* (1994) sugerem um limiar λ de energia como:

$$\lambda = \sigma \sqrt{2 \log_{10}(N)}, \quad (4.2)$$

sendo N o número de amostras (comprimento) do sinal ruidoso. A variável σ representa o desvio padrão do ruído e é dada por (DONOHO; JOHNSTONE, 1994):

$$\sigma = \text{mediana}(|Y|) / 0,6745. \quad (4.3)$$

O processamento com o uso de um limiar pode ser realizado com um *limiar rígido (hard)* ou um *limiar suave (soft)* conforme definidos, respectivamente, nas equações (4.4) e (4.5) a seguir:

$$X_{est} = THR_H(Y, \lambda) = \begin{cases} Y & , se |Y| > \lambda \\ 0 & , se |Y| \leq \lambda \end{cases} \quad (4.4)$$

$$X_{est} = THR_S(Y, \lambda) = \begin{cases} \text{sgn}(Y)(|Y| - \lambda), & se |Y| > \lambda \\ 0 & , se |Y| \leq \lambda \end{cases} \quad (4.5)$$

Os gráficos das funções definidas acima são apresentados nas Figuras 4.1 e 4.2, respectivamente.

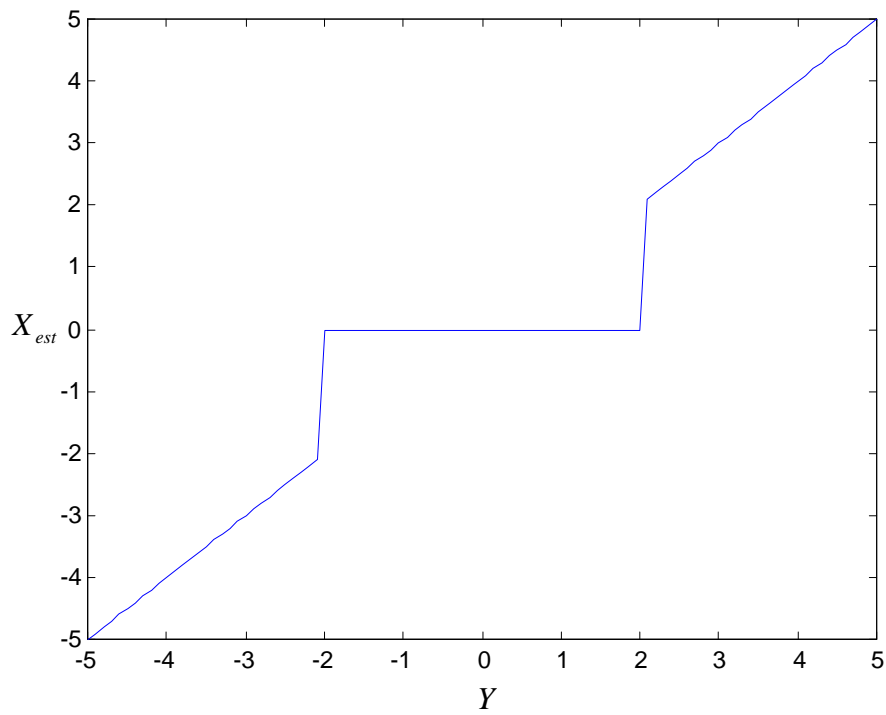


Figura 4.1 – *Hard Thresholding* ($\lambda=2$).

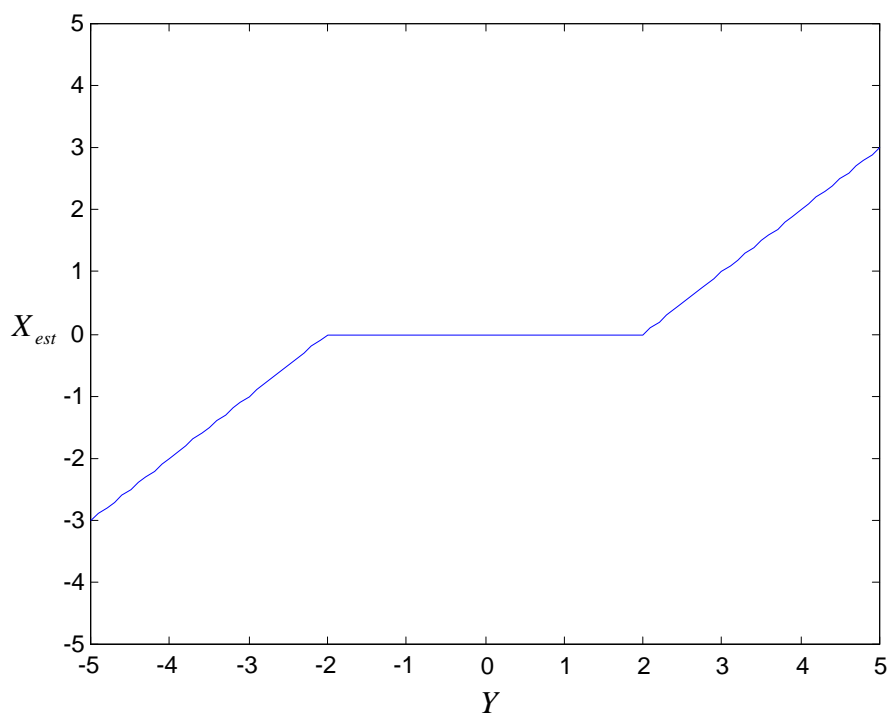


Figura 4.2 – *Soft Thresholding* ($\lambda=2$).

O limiar duro é do tipo “elimina” ou “preserva”, enquanto que o limiar suave é do tipo “elimina” ou “reduz” (reduz o tamanho na quantidade λ).

No caso do *limiar duro (hard)*, assume-se que os coeficientes *wavelets* menores que o limiar, são componentes ruidosos, ficando assim o sinal bem descrito pelos coeficientes *wavelets* maiores que o limiar. Deste modo, os coeficientes menores que o limiar são eliminados. Este método apresenta grande eficiência no caso de compressão de sinais (DUARTE et al., 2003). No caso do *limiar suave (soft)* assume-se que os componentes ruidosos são distribuídos igualmente em todos os coeficientes *wavelets*, assim todos os coeficientes *wavelets* são reduzidos pelo limiar. A proposta de *limiar suave*, dada pela equação (4.5), é uma das mais usadas. Entretanto, do ponto de vista de pesquisa, a determinação do *limiar suave* é, sem dúvida, um fator que pode alterar significativamente o desempenho do método de redução de ruído. Este método não produz uma qualidade satisfatória para muitos dos tipos de ruídos reais (SOARES et al., 2007 (B)).

4.2 Métodos de Redução de Ruído que Utilizam Limiar

Em processamento de sinais de voz, um fator que tem atraído bastante a comunidade científica é o de como reduzir o ruído presente nos sinais sem deteriorar a qualidade do sinal.

A redução de ruído é importante nas mais variadas aplicações que envolvem processamento de sinais. Existem diversos métodos para reduzir o ruído. Alguns usando a transformada de Fourier (VIEIRA FILHO, 1996) e outros que utilizam a transformada *wavelet* discreta (DWT) (DONOHO, 1995; DAE-SUNG et al., 2002; SHEIKHZADEH; ABUTALEBI, 2001). As aplicações que envolvem a transformada *wavelet* discreta (DWT) têm ganhado destaque nos últimos tempos devido a forma como esta transformada atua no sinal em processamento (RIOUL; VETTERLI, 1991). Os métodos mais aplicados no domínio *wavelet* são os métodos de redução por limiar, isto é, coeficientes do sinal transformado que estão abaixo de um determinado valor são eliminados ou reduzidos (DONOHO, 1995; DONOHO; JOHNSTONE, 1994). A vantagem da transformada *wavelet* está no fato de que no domínio transformado vários coeficientes são irrelevantes na reconstrução do sinal, de forma que a eliminação destes coeficientes não afeta a reconstrução perfeita do sinal (NIEVERGELT, 1999), este é o princípio básico da compressão de sinais no domínio *wavelet* (DUARTE et al., 2003). Quando se trata de redução de ruído, o limiar é multiplicado por um fator estimado na potência do ruído, causando, assim, a redução ou eliminação do mesmo (DUARTE,

2005). Nos dois casos, a eficiência dos métodos de limiar no domínio *wavelet* se justifica pela eliminação dos coeficientes redundantes do sinal transformado.

Além do método básico proposto por *Donoho e Johnstone* (1994) visto na seção 4.1, existem outros métodos que empregam limiares com algumas modificações. A seguir são descritos alguns deles (SOARES et. al, 2008 (A)).

O primeiro deles é o chamado *Semi-soft thresholding*, também conhecido como *Firm thresholding* (BRUCE; GAO, 1995). Neste método, dois limiares são usados, λ_1 e λ_2 , sendo que λ_2 é dependente de λ_1 , geralmente, $\lambda_2 = \sqrt{2} \lambda_1$. Este método é vantajoso sobre os métodos que usam as funções definidas pelas equações (4.4) e (4.5), devido a variância e ponderação do ruído estimado para os coeficientes de Y que têm valores absolutos entre λ_1 e λ_2 . Uma proposta de redução de ruído utilizando este método é apresentada em Seok e Bae (1997) e a detecção de segmentos “não vozeado” é feita avaliando a energia nas quatro primeiras faixas do sinal transformado. A equação (4.6) apresenta a função de transferência usada no método *Semi-soft thresholding* e a Figura 4.3 apresenta os valores de entrada e saída do respectivo método:

$$y_{thr} = \begin{cases} Y, & se \quad |Y| > \lambda_2 \\ \text{sgn}(Y) \frac{\lambda_2 (|Y| - \lambda_1)}{\lambda_2 - \lambda_1}, & se \quad \lambda_1 \leq |Y| \leq \lambda_2 \\ 0, & se \quad |Y| < \lambda_1 \end{cases} \quad (4.6)$$

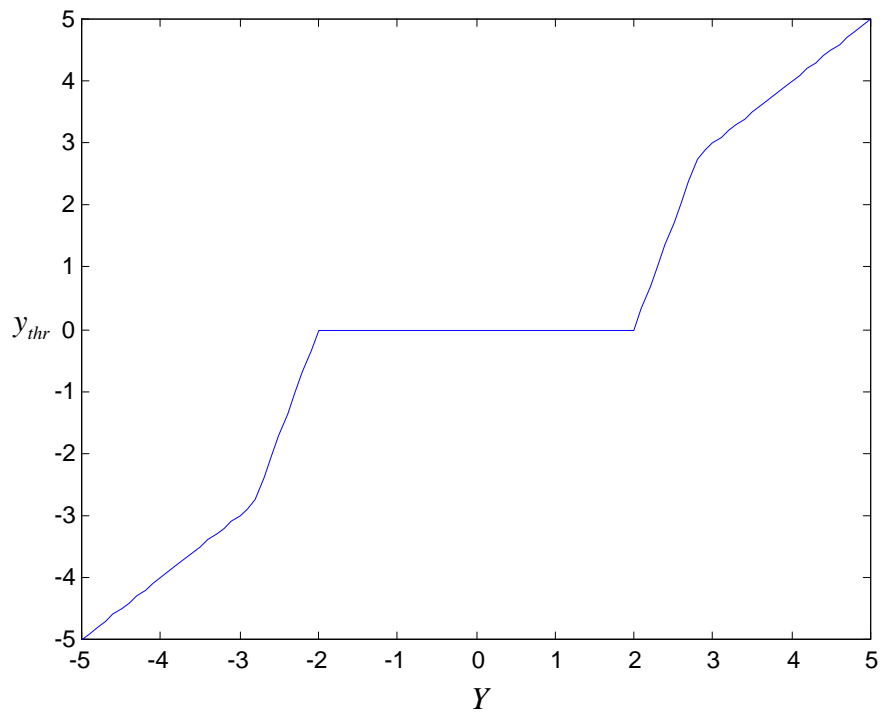


Figura 4.3 – *Semi-soft Thresholding* ($\lambda_1 = 2, \lambda_2 = 2\sqrt{2}$).

A seguir é apresentado o método chamado *Hard soft thresholding* (STORM, 1998), que usa a seguinte função de transferência:

$$y_{thr} = \begin{cases} Y, & se |Y| > \lambda \\ \alpha Y, & se |Y| \leq \lambda \end{cases} \quad (4.7)$$

Na Figura 4.4 apresenta-se a característica de entrada e saída do método proposto.

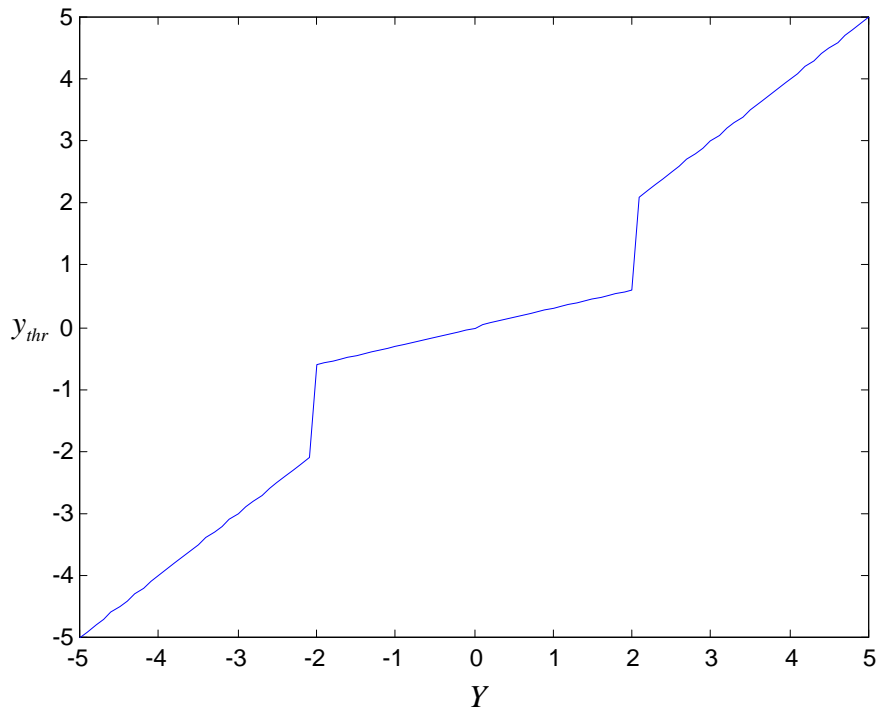


Figura 4.4 – *Hard Soft Thresholding* ($\lambda=2, \alpha=0,3$).

Na equação (4.7), α está no intervalo $[0,1]$. Observa-se que para $\alpha=0$ a equação (4.7) é semelhante à equação (4.4), concluindo que este limiar é uma variação do *Hard thresholding* evitando apenas a eliminação abrupta dos coeficientes que estão abaixo do limiar.

Além destes métodos, em Storm (1998) são apresentados ainda outros dois métodos que utilizam curvas de atenuação para Y . São eles: *Super super soft thresholding* e *Hard super soft thresholding*. No limiar *Super super soft thresholding*, dado $\lambda > 0$ e $\alpha \in [0,1]$, ocorre atenuação de todos os coeficientes, de acordo com a função de transferência associada $y_{thr} = y_{thr}(Y)$ definida por:

$$y_{thr} = \begin{cases} Y - \text{sgn}(Y)(1-\alpha)\lambda, & \text{se } |Y| > 1,5\lambda \\ \text{sgn}(Y)(c|Y|^2 + b|Y| + a), & \text{se } 0,5\lambda < |Y| \leq 1,5\lambda \\ \alpha Y, & \text{se } |Y| \leq 0,5\lambda \end{cases} \quad (4.8)$$

sendo $a = (1-\alpha)\lambda/8$, $b = 1,5\alpha - 0,5$ e $c = (1-\alpha)/2\lambda$. O gráfico da função polinomial em (4.8) une os pontos $(0,5\lambda, \alpha Y)$, com $Y = 0,5\lambda$, e $(1,5\lambda, Y - \text{sgn}(Y)(1-\alpha)\lambda)$, com $Y = 1,5\lambda$, notando que no segundo caso, o valor das funções $y_{thr} = Y - \text{sgn}(Y)(1-\alpha)\lambda$ e $y_{thr} = \text{sgn}(Y)(c|Y|^2 + b|Y| + a)$ em $Y = 1,5\lambda$ é $0,5\lambda + \alpha\lambda$. As inclinações das retas

tangentes ao gráfico da função polinomial em (4.8), nos pontos extremos considerados, são iguais a α e 1, respectivamente. Assim, as inclinações dessas retas tangentes nesses pontos coincidem com as inclinações das retas definidas na terceira e na primeira equação em (4.8), respectivamente. Na Figura 4.5 é apresentado o gráfico da função $y_{thr} = y_{thr}(Y)$ definida em (4.8).

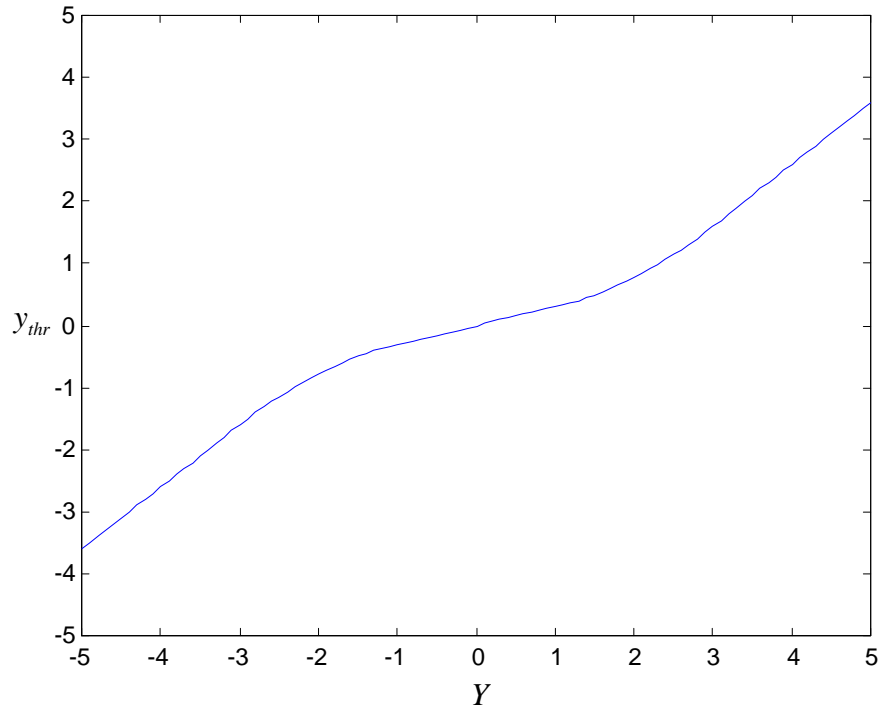


Figura 4.5 – *Super Super Soft Thresholding* ($\lambda=2, \alpha=0,3$).

No limiar *Hard Super Soft Thresholding*, dados $\lambda > 0$ e $\alpha \in [0,1]$, a atenuação é feita em todos os coeficientes, exceto quando $|Y| > 1,5\lambda$, de acordo com a função de transferência associada $y_{thr} = y_{thr}(Y)$ definida por:

$$y_{thr} = \begin{cases} Y, & \text{se } |Y| > 1,5\lambda \\ \text{sgn}(Y)(e|Y|^4 + d|Y|^3 + c|Y|^2 + b|Y| + a), & \text{se } 0,5\lambda < |Y| \leq 1,5\lambda \\ \alpha Y, & \text{se } |Y| \leq 0,5\lambda \end{cases} \quad (4.9)$$

sendo $a=0$, $b=1$, $c=4,5(\alpha-1)/\lambda$, $d=6(1-\alpha)/\lambda^2$ e $e=2(\alpha-1)/\lambda^3$. O gráfico da função polinomial considerada em (4.9) une os pontos $(0,5\lambda, \alpha Y)$, $(\lambda, 0,5Y(1+\alpha))$ e $(1,5\lambda, Y)$. As inclinações das retas tangentes ao gráfico da função polinomial em (4.9), nos pontos $(0,5\lambda, \alpha Y)$ e $(1,5\lambda, Y)$ são iguais a α e 1, respectivamente. Assim, as

inclinações dessas retas tangentes nesses pontos coincidem com as inclinações das retas definidas na primeira e na terceira equação em (4.9), respectivamente. Na Figura 4.6 é apresentado o gráfico da função $y_{thr} = y_{thr}(Y)$ definida em (4.9).

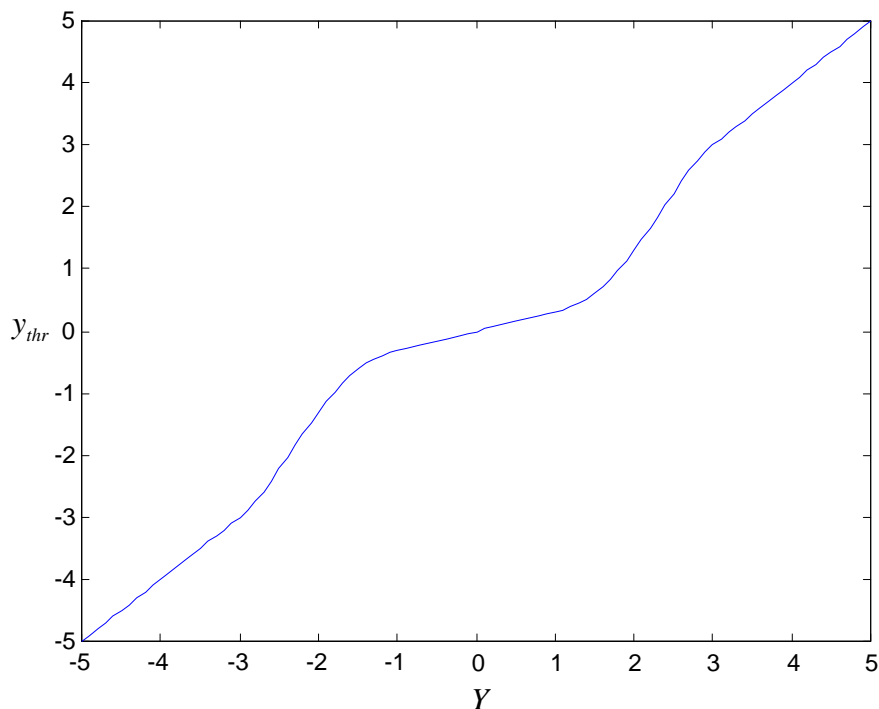


Figura 4.6 – *Hard Super Soft Thresholding* ($\lambda=2, \alpha=0,3$).

Em *Sheikhzadeh e Abutalebi* (2001) uma versão modificada do método *wavelet* básico (DONOHO; JOHNSTONE, 1994) é apresentada, com o objetivo de eliminar os problemas que surgem na sua aplicação. Para detectar os trechos do tipo “não vozeado” usa-se a distribuição da energia em cada segmento do sinal analisado e o perfil de ruído é definido como $\sigma = (1/0.6745) \cdot \text{mediana}(|c|)$, e é usado no cálculo do limiar, que é obtido de acordo com a equação (4.2). Este método, embora faça a redução do ruído por faixa de frequência nos trechos de voz, usa o limiar calculado no último trecho de silêncio. Este limiar é calculado apenas na última janela deste trecho de silêncio. Calcular o limiar apenas em uma janela pode significar pouca redução de ruído nos trechos de voz, se esta janela tem pouco ruído, ou forte redução se esta janela tem muito ruído, em ambos os casos a inteligibilidade do sinal pode ser comprometida. Consequentemente, uma das desvantagens deste método é o uso de um único limiar para todas as faixas de frequência.

Outro método, proposto em *Nordström* (2002), chamado *Super soft thresholding*, usa a função definida na equação (4.10), com α em $[0,1]$. Na Figura 4.7 apresentam-se as suas características de entrada e saída:

$$y_{thr} = \begin{cases} Y - \text{sgn}(Y)(1-\alpha)\lambda, & \text{se } |Y| > \lambda \\ \alpha Y & , \text{se } |Y| \leq \lambda \end{cases} \quad (4.10)$$

Quando $\alpha=0$ a equação (4.10) se torna semelhante à equação (4.5).

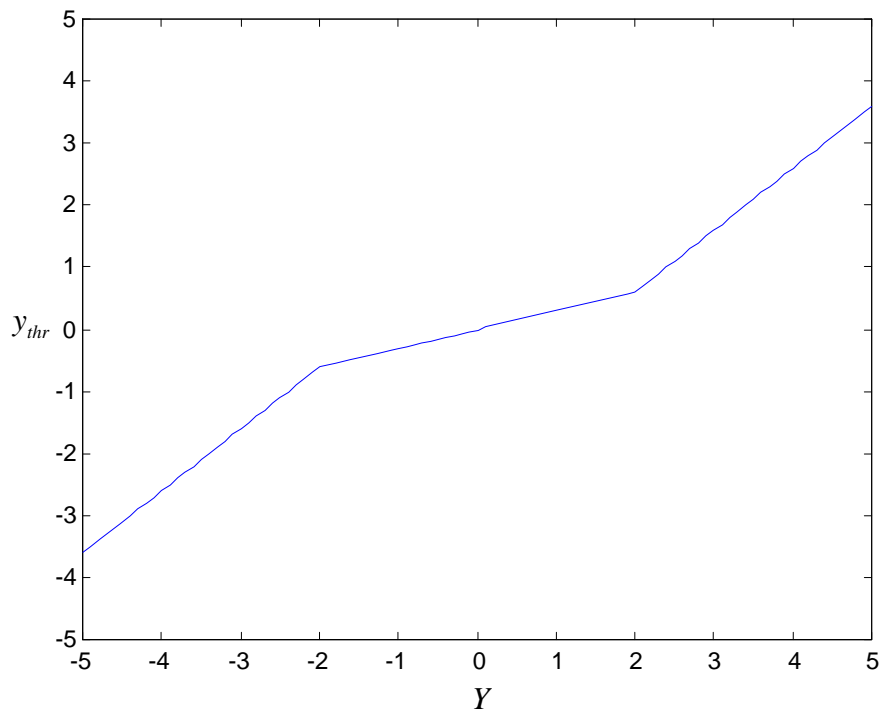


Figura 4.7 – *Super Soft Thresholding* ($\lambda=2, \alpha=0,3$).

O método proposto por *Lallouani, Gabrea e Gargour* (2004) consiste em combinar dois algoritmos μ -law thresholding, originado do μ -law logarithm thresholding (*CHANG et al.*, 2002) e soft thresholding. O sinal ruidoso a ser eliminado é decomposto em sete níveis da transformada *Wavelet Packet* (WPT) usando a função wavelet de *Daubechies* de ordem 11 (dB11). A WPT vem sendo sucessivamente utilizada em muitas aplicações, particularmente em melhoria de voz (*CHANG et al.*, 2002; *HU; LOIZOU*, 2004), pois esta faz uma análise mais rica, já que a cada nível de decomposição, além dos coeficientes de aproximação, os coeficientes de detalhes são decompostos, ao contrário da DWT que só decompõe os coeficientes de aproximação. O μ -law thresholding é aplicado em todos os coeficientes nos níveis de decomposição no final de cada subfaixa, exceto nas duas mais baixas subfaixas onde o soft thresholding é

aplicado. Para cada faixa de decomposição, o limiar é escolhido de acordo com *Johnstone e Silverman (1997)* ao invés de usar o método universal proposto por *Donoho e Johnstone (1994)*. A razão desta escolha é que no caso de ruído não-estacionário ou colorido, o nível de ruído é diferente de uma faixa para outra no mesmo nível de escala. O método proposto herda a eficiência de dois métodos: a técnica do *soft thresholding* e a técnica *μ -law thresholding* modificado.

Um outro método existente é o *Sigmoidal thresholding* (DUARTE, 2005), este método atenua os coeficientes que estão abaixo do limiar usando a função sigmoide apresentada na equação (4.11). Sua vantagem sobre os demais é que esses coeficientes são atenuados por um fator dependente de seus próprios valores conforme apresentado na equação (4.12)

$$\text{Sigmoide}(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (4.11)$$

$$y_{thr} = \begin{cases} Y, & \text{se } |Y| > \lambda \\ Y \cdot |\text{sigmoide}(Y)|, & \text{se } |Y| \leq \lambda \end{cases} \quad (4.12)$$

Na equação (4.11), γ controla a inclinação da função sigmoide.

A Figura 4.8 apresenta o gráfico da função de transferência do método sigmoidal.

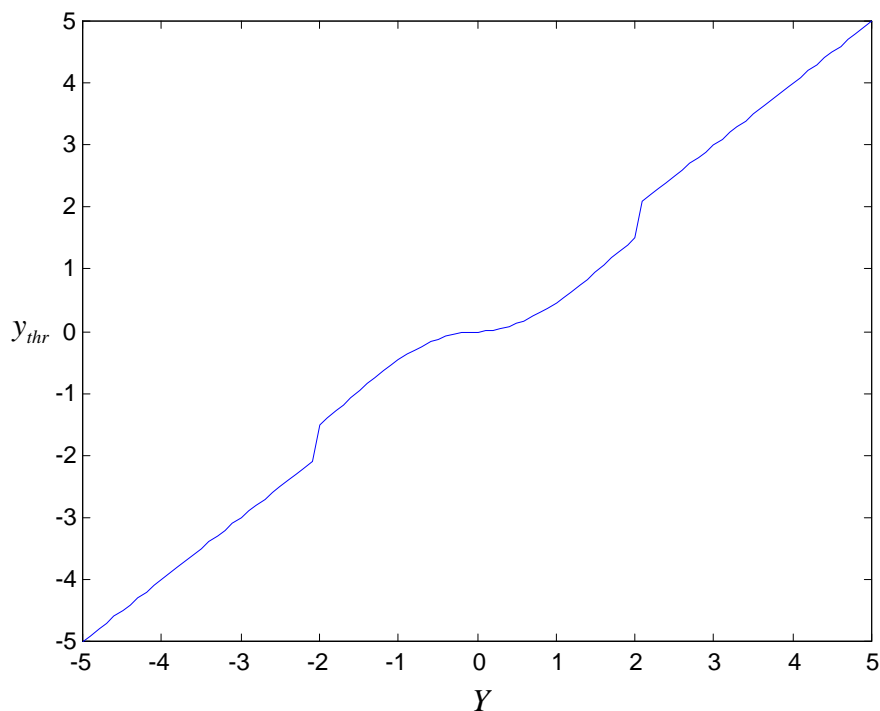


Figura 4.8 – *Sigmoidal Thresholding* ($\lambda=2, \alpha=1$).

4.3 Desvantagens dos Métodos que Utilizam Limiar

A maioria dos métodos empregados na redução de ruído propõe o uso de algoritmos que tratam das aplicações de limiares. Muitos deles trazem resultados bons, quando são aplicados em ruído gaussiano branco. Mas quando estes são aplicados em sinais não-estacionários ou coloridos, os resultados nem sempre são satisfatórios. Um dos problemas consiste em detectar as regiões de pausa que permite a implementação de técnicas para a separação entre trechos de voz e silêncio, regiões estas que são usados na determinação das características básicas do ruído e na implementação de um sistema adequado de identificação de trechos de sinal do tipo “não vozeado” (SEOK; BAE, 1997). Alguns métodos (como por exemplo, *Seok e Bae* (1997), *Sheikhzadeh e Abutalebi* (2001), *Duarte* (2005), entre outros), usam a distribuição da energia em cada segmento do sinal analisado e o perfil de ruído, na maioria das vezes, usando os métodos propostos por *Donoho* (1995); *Donoho e Johnstone* (1994). Outros métodos (como por exemplo, *Bahoura e Rouat* (A) e (B), (2001), etc.) fazem a detecção dos trechos “não vozeado” dividindo o sinal transformado em várias faixas e, ainda, aplicam a Transformada *Wavelet Packet* (WPT) nos diferentes níveis de escala, fazendo, assim,

uma decomposição nos coeficientes de aproximação e também nos coeficientes de detalhes.

A eficiência de um método que utiliza limiar está no fato de que o sinal transformado Y apresenta coeficientes considerados redundantes na reconstrução do sinal y . Estes coeficientes geralmente são menores que o limiar e por isso são eliminados ou atenuados dependendo do método aplicado. Porém, quando o ruído é adicionado ao sinal, muitos coeficientes redundantes assumem valores acima do limiar, além de que a introdução do ruído causa a inversão de fase de outros coeficientes. Assim, na aplicação do limiar, muitos coeficientes que deveriam ser eliminados acabam sendo preservados, devido ao ruído presente nos mesmos, causando excesso de ruído no sinal processado. Esta é a grande deficiência dos métodos que utilizam limiar quando aplicados com o objetivo de reduzir ruído.

Na Figura 4.9 é apresentada uma representação dos coeficientes *wavelet* de um sinal limpo de 256 amostras. Na Figura 4.10 o mesmo sinal, porém contaminado com ruído branco. Nas duas figuras, as linhas tracejadas representam o limiar, isto é, os coeficientes que estão na faixa limitada por estas linhas devem ser eliminados. Pode-se observar graficamente que o número de coeficientes redundantes diminui na Figura 4.10 em relação à Figura 4.9, isto é, coeficientes que seriam eliminados na Figura 4.9 não são eliminados na Figura 4.10. Como estes coeficientes não são eliminados, os seus conteúdos ruidosos permanecem no sinal.

Na Figura 4.11 é apresentado o trecho final dos dois sinais apresentados nas Figuras 4.10 e 4.9. As setas indicam alguns coeficientes onde há inversão de fase. Esta inversão de fase causa fortes distorções no sinal.

Os dois problemas citados anteriormente e ilustrados nas Figuras 4.9 a 4.11 mostram que os métodos que utilizam limiares nem sempre são eficientes na redução de ruído, pois eles ou eliminam coeficientes importantes na reconstrução do sinal, em caso de um limiar com valor alto, ou eles não eliminam coeficientes redundantes, no caso de um limiar com valor baixo. Além de não detectar quando ocorre a inversão de fase em alguns trechos do sinal. Tudo isso faz com que haja pouca eliminação de ruído, ou muita eliminação, inclusive de coeficientes importantes, introduzindo muita distorção no sinal final.

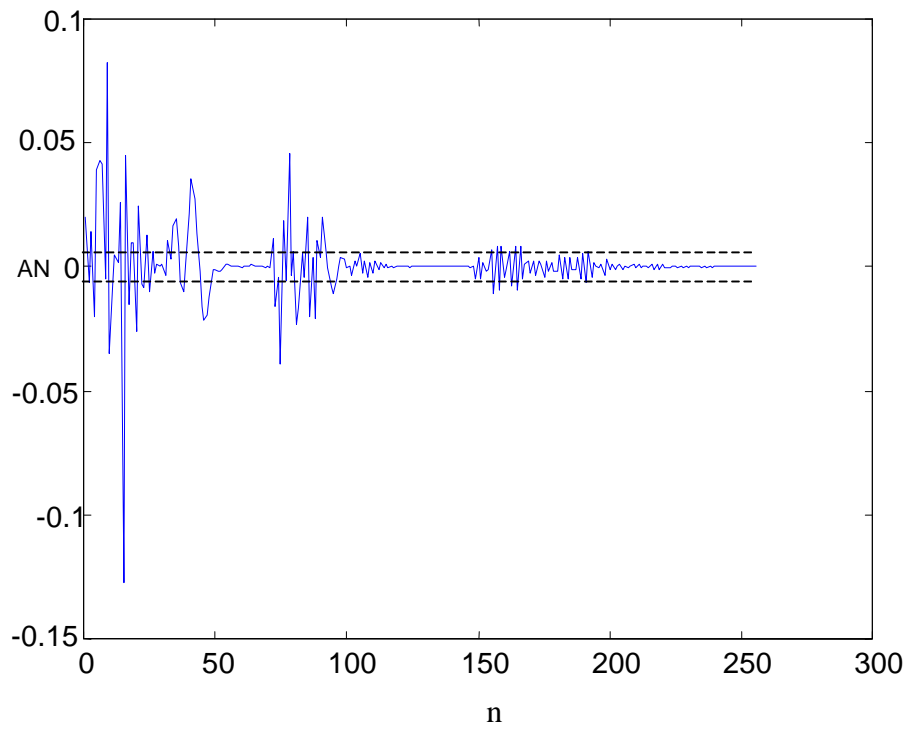


Figura 4.9 – Sinal limpo no domínio *wavelet*.

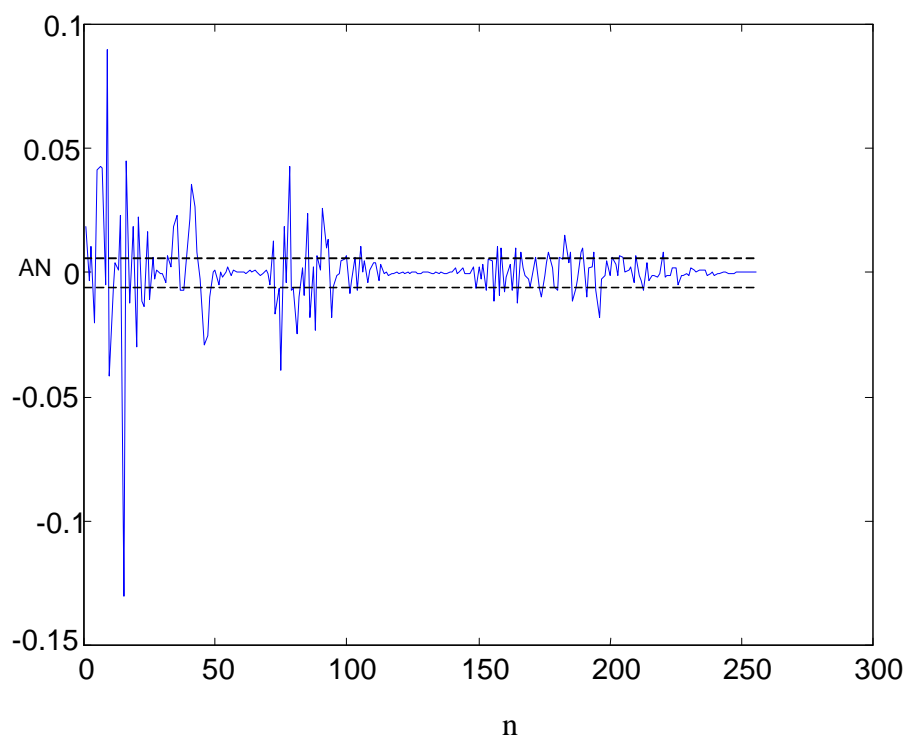


Figura 4.10 – Sinal contaminado por ruído branco no domínio *wavelet*.

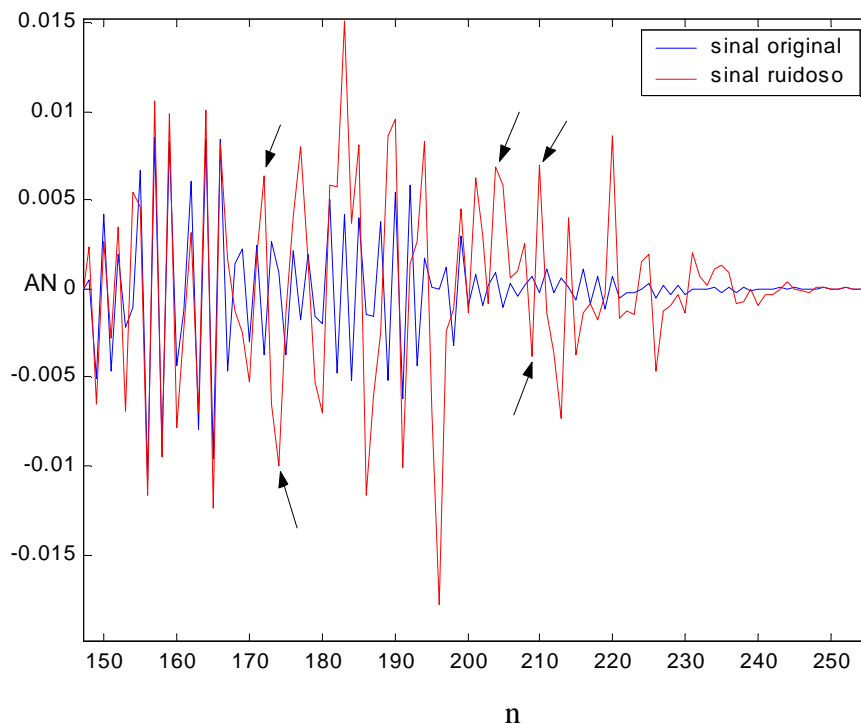


Figura 4.11 – Sinais Limpo e Ruidoso no domínio *wavelet*.

4.4 Análise dos Métodos Híbridos para Redução de Ruído em Sinais de Voz (Usando *Wavelets*)

O propósito desta seção é apresentar outros métodos de redução de ruído no domínio *wavelet* que não usam limiar ou que combinam limiar com outra metodologia. Essa apresentação será feita de forma resumida.

Dentre os diferentes métodos encontrados na literatura, os métodos que utilizam o cálculo de um limiar são os mais usados. Porém, há métodos híbridos que combinam subtração espectral e transformada *wavelet* (NISHIMURA et al., 1998; AKHAEI, AMERI; MARVASTI, 2005; AKHAEI; MARVASTI, 2006). Outros que empregam um fator robusto de ponderação para cada subfaixa *wavelet* (LU; WANG, 2004), além dos que usam filtros adaptativos na reconstrução do sinal (AKHAEI, AMERI; MARVASTI, 2005) empregando também ainda as WPT (BAHOURA; ROUAT (A) e (B), 2001).

No método proposto em *Nishimura et al.* (1998), cujos comentários são feitos a seguir, é aplicada a transformada *wavelet* na técnica de subtração espectral. Considerando a estrutura da voz humana, foram usadas as *wavelets* de *Gabor* e

Daubechies, e uma senóide com decaimento como *wavelet* fundamental. Nas avaliações feitas, os resultados mostraram que a relação sinal ruído (SNR) melhora em relação ao método tradicional da subtração espectral. Quando a *wavelet* de *Gabor* e a senóide foram usadas, foi obtido um ganho de aproximadamente 6 dB na melhoria da SNR, sob todas as condições examinadas. No caso de uma *wavelet* ortogonal, tal como a *wavelet* de *Daubechies*, a melhoria na SNR é menor que 3 dB. Uma possível razão para isto, no caso da *wavelet* ortogonal, é que a resolução no tempo e frequência é restrita pela ortogonalidade, resultando numa resolução insuficiente. No caso das *wavelets* não-ortogonais, por outro lado, é possível selecionar arbitrariamente a densidade das bases *wavelets* nos eixos do tempo e da frequência. Esta flexibilidade é uma das vantagens de usar as *wavelets* não-ortogonais. Pois, para ruído gaussiano branco, a SNR é melhorada pela subtração espectral convencional, tal como, pela WBSS (*wavelet-based spectral subtraction*) usando a senóide com decaimento ou a *wavelet* de *Gabor* como *wavelet* fundamental. Mas, no ruído colorido, a melhoria na SNR pela subtração espectral original é menor. A razão disto não está bem clara. Porém, a seguinte argumentação pode ser possível: a potência do ruído colorido está concentrada nas regiões de baixa frequência e os componentes com longos tamanhos de ondas são predominantes. Portanto, ao analisar este sinal usando uma janela com comprimento relativamente curta, a flutuação no espectro de curto tempo aumenta. Adequadamente, a diferença entre o espectro do atual ruído e a média do espectro do ruído aumenta, e o resíduo na subtração espectral, conseqüentemente, também. No caso da transformada *wavelet*, o comprimento da efetiva janela para análise é estendido nas regiões de baixa-frequência, que pode resultar em uma estimacão melhor do espectro do ruído.

Um método para melhorar o desempenho das técnicas de melhoria de voz, utilizando limiarização, *wavelet* foi proposto em *Bahoura e Rouat (B) (2001)*. Este método em primeiro lugar, faz uma adaptacão do limiar no tempo, obtida através da extensão do princípio do limiar dependente do nível na WPT para remover vários ruídos. A adaptacão no tempo é introduzida usando o *Teager Energy Operator (TEO)* (*BAHOURA; ROUAT (A), 2001*) nos coeficientes *wavelets*. Finalmente, para prevenir a deterioracão durante o processo de limiarização, um novo limiar é proposto e comparado com o algoritmo de filtros de *Ephraim e Malah (EMF)* (*EPHRAIM; MALAH (A), 1984; EPHRAIM; MALAH (B), 1985*). Observar que o algoritmo EMF requer uma estimacão explícita do nível de ruído ou um conhecimento *a priori* da SNR, não sendo necessário neste método citado. No entanto, a limiarização dependente do

tempo permite remover vários ruídos em ambientes diferentes (faixas estreitas ou largas de frequência) e no processo de limiar adaptado no tempo é evitada a degradação da qualidade da voz durante o processo de limiarização.

Dae-Sung, et al. (2002) propõem uma filtragem passa-baixa nos coeficientes *wavelet* localizados nas faixas de baixa frequência do sinal. Para sinais de voz, este método não é eficiente porque não reduz o ruído presente nas faixas de alta frequência e a aplicação do filtro passa-baixa deteriora muito a voz, mesmo nas faixas de baixa frequência.

Nos métodos propostos por *Visser*, *Lee* e *Otsuka* (2003), *Hu* e *Loizou* (2004), aplicam-se um limiar no espectro do sinal no domínio *wavelet* com o objetivo principal de reduzir o ruído musical e efeitos causados por multilocutores (CAPPÉ, 1994; VIEIRA FILHO, 1996).

Em *Lu* e *Wang* (2004) aplica-se uma máscara de um limiar no ruído NMT (*Noise Masking Threshold*) e emprega-se um fator robusto de ponderação para cada subfaixa *wavelet*. Duas restrições são utilizadas para otimizar o fator de ponderação para cada subfaixa. O primeiro consiste em manter a energia do ruído residual menor que NMT, ou seja, o fator de ponderação fica pequeno para suprimir o ruído. Se a energia do ruído residual é menor que NMT, o ruído pode não ser percebido pelo ouvido humano. Por reter a qualidade da voz, não há necessidade de trocar os coeficientes *wavelets*, isto é, o fator de ponderação é fixado em 1 e a segunda restrição garante que a distorção da voz é menor que o ruído residual.

Em *Akhaee*, *Ameri* e *Marvasti* (2005) usam-se filtros adaptativos em baixas escalas da transformada *wavelet* em sinais de voz, junto com os métodos convencionais (limiaries, subtração espectral e filtragem de *Wiener*) (VIEIRA FILHO, 1996) nos coeficientes das faixas de alta frequência. Neste método é empregado o algoritmo dos quadrados mínimos (LMS) para os principais coeficientes *wavelets* e para o restante dos coeficientes *wavelets*, foram empregados um dos três métodos:

- a) zeram-se todos os coeficientes;
- b) aplica-se o método de subtração espectral; ou
- c) aplica-se o filtro de *Wiener* usando uma das seguintes aproximações:
 - i) máxima probabilidade, ou
 - ii) método da decisão direta.

Os filtros adaptativos têm uma desvantagem para sinais de baixa SNR, pois apresentam pouca redução de ruído, e a principal vantagem, é que em escalas mais

baixas eles produzem um melhor desempenho. Para a decomposição do sinal, neste método, foi usada a função de *Daubechies* de ordem 1 (dB1) e esta decomposição foi feita em quatro níveis de resolução. A característica do ruído de referência foi extraída por subtração espectral ou pelo filtro de *Wiener*.

No método proposto por *Akhaee e Marvasti* (2006) para prevenir a degradação na melhoria da voz, trechos de silêncio são separados dos trechos de voz por uma nova função *thresholding*, que é aplicada, diferentemente para cada escala de detalhes, e o procedimento de limiarização que é escolhido por algumas hipóteses estatísticas é aplicado diferentemente em cada escala de detalhes. Para obter um melhor desempenho e robustez contra o ruído “murmúrio”, o método de subtração espectral é usado na aproximação de escalas. Assim, as desvantagens da função *thresholding* são reduzidas e a melhor qualidade na voz é atingida. O algoritmo tem resultados satisfatórios, mas tem suas desvantagens, como por exemplo, a ineficácia em ruídos não-estacionários ou coloridos, a degradação na qualidade por causa da introdução dos sons no silêncio e um som de voz “não-natural” devido ao emprego dos limiares *Hard* e *Soft Thresholding*. A estimação da potência do ruído é calculada de acordo com a equação (4.3) e o limiar proposto é aplicado conforme a equação (4.13):

$$y_{thr} = \begin{cases} 0, & se \quad |Y| < \lambda/4 \\ k \cdot (\sqrt{|Y|} - \sqrt{\lambda})^2 \cdot \text{sgn}(Y), & se \quad |Y| \geq \lambda/4 \end{cases} \quad (4.13)$$

sendo $k = 1.6$.

Na Figura 4.12 está representada a curva de limiar do método proposto em *Akhaee e Marvasti* (2006) de acordo com a equação (4.13).

Os métodos clássicos baseados em subtração espectral são efetivos para este propósito, porém eles introduzem ruído artificial, conhecido como ruído musical, e alteram o sinal original.

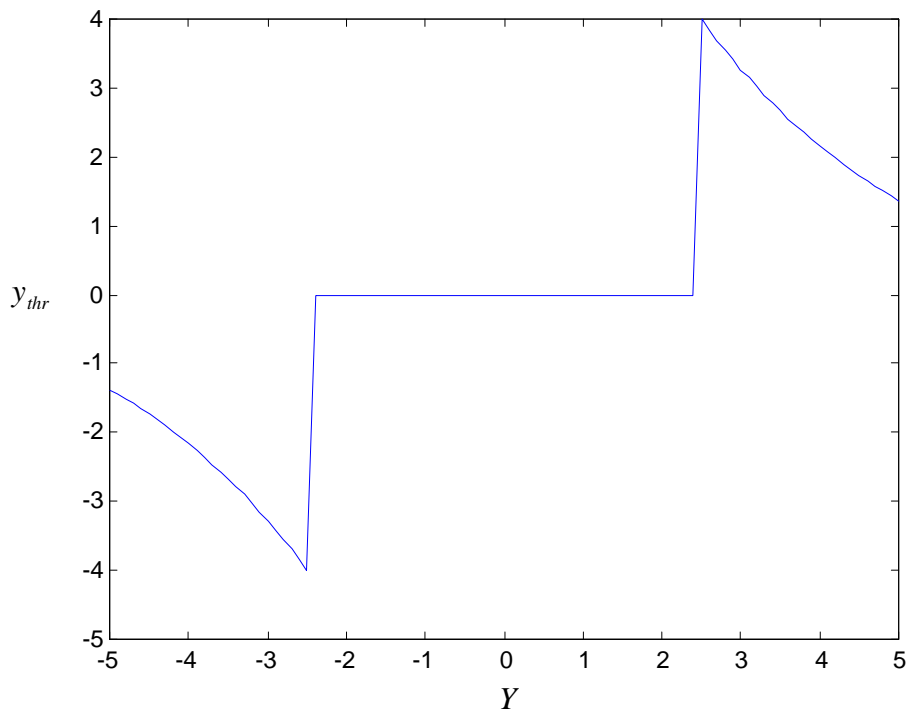


Figura 4.12 – Método proposto por *Akhaee e Marvasti* (2006).

Os métodos estudados e analisados nesta seção mostram que, além de utilizar o cálculo com limiar, existem também outras técnicas para reduzir o ruído em sinais de voz no domínio *wavelet*. Os estudos e as análises destes métodos levam a algumas analogias no que diz respeito à redução de ruído em sinais de voz. Os estudos de todos estes métodos com aplicação de limiar permitiram traçar a metodologia proposta para esta pesquisa que está descrita no capítulo 5.

4.5 Análise de Um Método Sem o Uso de Limiar.

A maioria dos métodos de redução de ruído baseado em *wavelets*, como se pode observar nas seções anteriores, faz uso de limiar para reduzir ou atenuar ruído. Porém, como foi descrito na seção 4.3, estes métodos tem várias desvantagens, a principal delas é a descontinuidade em suas funções de transferência, o que causa deterioração nos trechos de voz do sinal. Assim, a busca por métodos baseados em *wavelets* que não usam limiar é um dos objetivos da comunidade de processamento de sinais.

Um método que não faz uso de limiar e que apresenta bons resultados foi proposto por *Qiang e Wan* (2003). Este método faz redução de ruído combinando

subtração espectral e uma filtragem baseada em uma curva de atenuação no domínio *wavelet* usando a transformada *wavelet* perceptiva. Uma síntese deste método é apresentada a seguir.

4.5.1 Processamento

A subtração espectral é aplicada antes da decomposição do sinal pela transformada *wavelet* perceptiva.

Cada *i*-ésima banda dos coeficientes decompostos é segmentada em janelas de comprimento L_{frm}^i .

$\hat{\sigma}^{i,p}$ é o nível de ruído estimado da *p*-ésima janela na *i*-ésima subbanda. $\hat{\sigma}$ é estimado usando o segmento de dados anteriores.

$$\{c_m^{i,p}, m = 0, \dots, L_{seg}^i - 1\}, \text{ onde } L_{seg}^i > L_{frm}^i. \quad (4.14)$$

Os pontos nos quais as janelas devem ser segmentadas são ordenados da seguinte forma:

$$c_0^{i,p} < c_1^{i,p} < \dots < c_{L_{seg}^i - 1}^{i,p} \quad (4.15)$$

Dado um valor q ($0 < q < 1$), o ponto de separação para este segmento é $c_{\text{int}(q.L_{seg}^i)}^{i,p}$, onde a função $\text{int}(\cdot)$ representa o inteiro mais próximo.

O ruído estimado é dado por

$$\hat{\sigma}^{i,p} = \beta \cdot \sum_{j=0}^{\text{int}(q.L_{seg}^i)} c_j^{i,p} / L_{seg}^i \quad (4.16)$$

Sendo β um fator de escolha apropriado, os valores usados são $q = 0,2$ e $\beta = 0,38$.

Os correspondentes comprimentos de tempo para L_{seg} e L_{frm} são 512ms e 64ms, respectivamente, e a transição é feita a cada 32ms, conforme se pode observar na Figura 4.13.

O limiar para cada subbanda da *p*-ésima janela, $\lambda^{i,p}$, é estimado por

$$\lambda^{i,p} = \sqrt{2 \log [L_{seg}^i \log_2 (L_{seg}^i)]} \cdot \hat{\sigma}^{i,p} \quad (4.17)$$

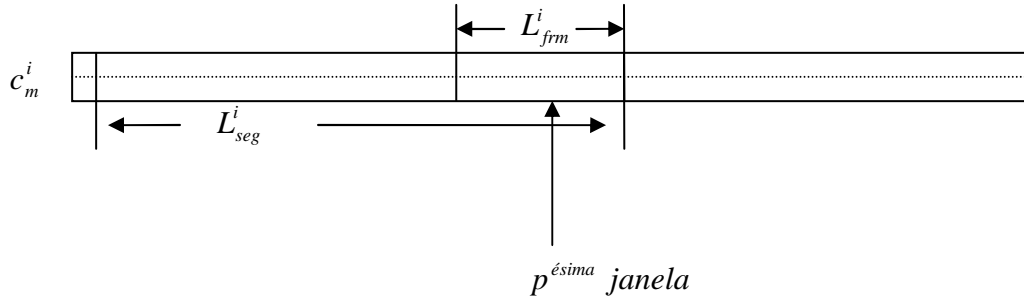


Figura 4.13 – Escolha do segmento para cálculo do limiar.

A curva de atenuação é calculada da seguinte forma:

Por simplicidade, seja $\lambda_m^i = \lambda^{i \cdot p}$ se m cai na p -ésima janela. Define-se uma razão coeficiente por limiar a *priori* (CTR) (isto é análogo a SNR usada na regra de *Ephraim* e *Malah*),

$$(R_m^i)^{priori} = \frac{|c_m^i|}{\lambda_m^i} \quad (4.18)$$

A correspondente CTR a *posteriori* é dada por

$$(R_m^i)^{post} = \alpha \frac{|c_m^i - L_{frm}|}{\lambda_m^i - L_{frm}} + (1 - \alpha) \cdot \max[0, (R_m^i)^{priori} - 1] \quad (4.19)$$

sendo que α ($0 \leq \alpha \leq 1$) é usado para controlar o grau de supressão (eliminação). Aqui é considerado $\alpha = 0,5$.

Com os CTRs a *priori* e a *posteriori*, um filtro de supressão pode ser escrito como:

$$H_m^i = \frac{(R_m^i)^{post}}{1 + (R_m^i)^{post}} \left(\frac{1}{(R_m^i)^{priori}} + \frac{(R_m^i)^{post}}{1 + (R_m^i)^{post}} \right) \quad (4.20)$$

Após seu cálculo, H é aplicado aos coeficientes ruidosos decompostos, c_m^i , isto é,

$$\hat{c}_m^i = H_m^i \cdot c_m^i \quad i = 0, 1, \dots, 17 \quad (4.21)$$

A última fase envolve a reconstrução do melhoramento de voz usando a transformada *wavelet* perceptiva inversa, dada por:

$$\hat{s}_k = PW^{-1} \{ \hat{c}_m^i \}, \quad i = 0, \dots, 17 \quad (4.22)$$

A seguir, faz-se um produto vetorial, ponto a ponto, entre o sinal no domínio *wavelet* e este vetor de limiares de forma que haja atenuação e não eliminação de coeficientes, numa tentativa de evitar descontinuidades no sinal reconstruído e o ruído residual. A grande deficiência deste método está na relação limiar/coeficiente, pois ao atenuar o ruído o mesmo faz com que o sinal apresente fortes distorções quando o produto vetorial é efetuado.

CAPÍTULO 5

METODOLOGIA PROPOSTA

5.1 – Aspectos Gerais

Em processamento de sinais, os métodos que utilizam limiar exploram o fato de que, no domínio *wavelet*, muitos coeficientes de um determinado sinal podem ser modificados ou anulados sem comprometer a reconstrução do sinal. O método para o cálculo de limiar proposto por *Donoho* e *Johnstone* (1994) é eficiente para este princípio, tornando os métodos que utilizam limiares muito eficientes para a compressão de sinais. Porém, quando um ruído é adicionado ao sinal, muitos coeficientes que são considerados redundantes assumem valores significativos e assim não são eliminados pelo limiar usado. Outro problema que pode ocorrer, quando é aplicado um método que usa limiar, é a inversão de fase de alguns coeficientes. Na Figura 5.1, é apresentada uma janela de 128 amostras de um trecho de sinal de voz limpo no domínio *wavelet* e contaminado por ruído branco. Pode-se observar que, após a adição do ruído, os coeficientes com valores menores estão alterados de forma significativa. Isto pode ser melhor observado na Figura 5.2, que apresenta um trecho mais detalhado dos coeficientes *wavelets*. No trecho apresentado, verifica-se que houve um aumento significativo no valor absoluto de vários coeficientes e a inversão no sinal em outros. Logo, considerando o processamento de um longo trecho de sinal, muitos

coeficientes são alterados de forma aleatória e geram distorções que podem reduzir significativamente a qualidade do sinal (SEOK; BAE, 1997). Além disso, qualquer método que utiliza limiar também elimina coeficientes de voz que não estão alterados. Embora, o valor de limiar calculado seja usado para eliminar apenas as redundâncias do sinal, para que a redução de ruído seja eficaz, o limiar deve ser multiplicado por um fator baseado na potência do ruído e, assim, os coeficientes do sinal de voz que são menores que este produto também são eliminados, causando distorções nos trechos de voz do sinal.

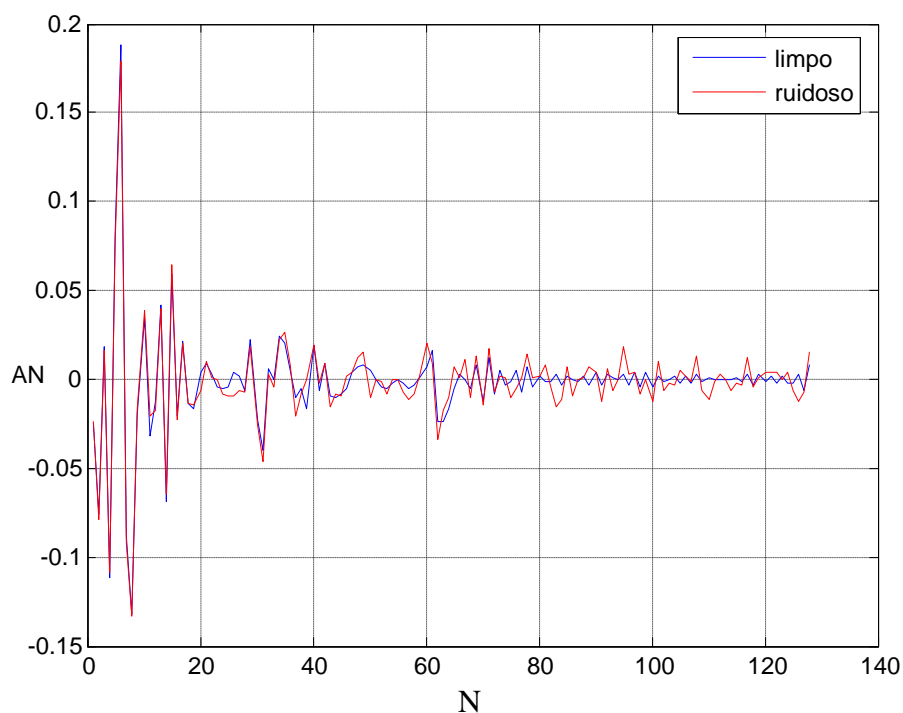


Figura 5.1 – Janela de trecho de voz de um sinal limpo e contaminado por ruído branco, no domínio *wavelet*.

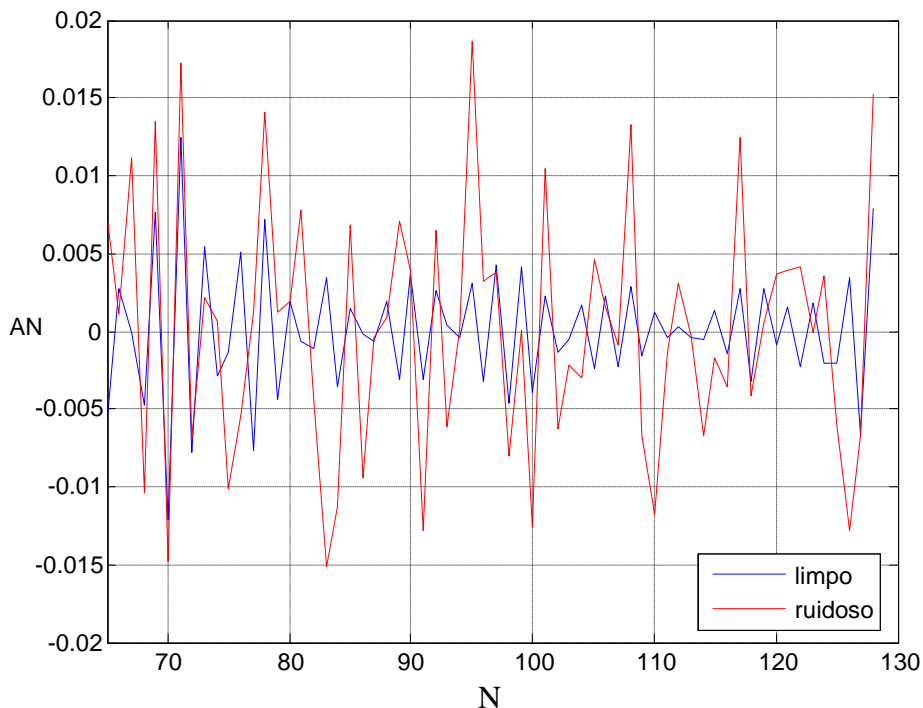


Figura 5.2 – Segunda metade da janela apresentada na Figura 5.1.

Levando em conta as observações feitas anteriormente, busca-se um método de redução de ruído, no domínio *wavelet*, que seja capaz de superar as dificuldades apresentadas. Com esse objetivo é proposta neste capítulo uma técnica de filtragem cujo processo não elimine ou atenuar coeficientes baseado na utilização de um limiar. Isto é, o objetivo é evitar que um único valor (valor do limiar) interfira nos valores de todos os coeficientes de uma faixa de frequência do sinal.

5.1.1 Objetivos do Método Proposto

Os algoritmos de redução de ruído, que utilizam *wavelets* apresentados na literatura especializada, em sua maioria, são baseados em corte por limiar. Estes métodos apresentam as deficiências citadas anteriormente, bem como descontinuidades em suas funções de transferência, além de não serem tão eficazes no processamento de sinais contaminados por ruído colorido (ruído real).

A metodologia de redução de ruído proposta neste trabalho consiste na aplicação de três operadores parciais e independentes. A aplicação tem como objetivo reduzir o ruído presente no sinal de forma a torná-lo inteligível. Os operadores são gerados de forma independente, porém a curva usada para redução de ruído é gerada por

uma combinação dos três, como se pode observar na Figura 5.3. Cada operador age de acordo com as suas características nas regiões de voz e silêncio, sem o uso de limiar. A execução dos três processamentos é condensada numa única função, denominada de função de transferência e denotada por *fil*, que atua como um filtro em processamento de sinais de voz.

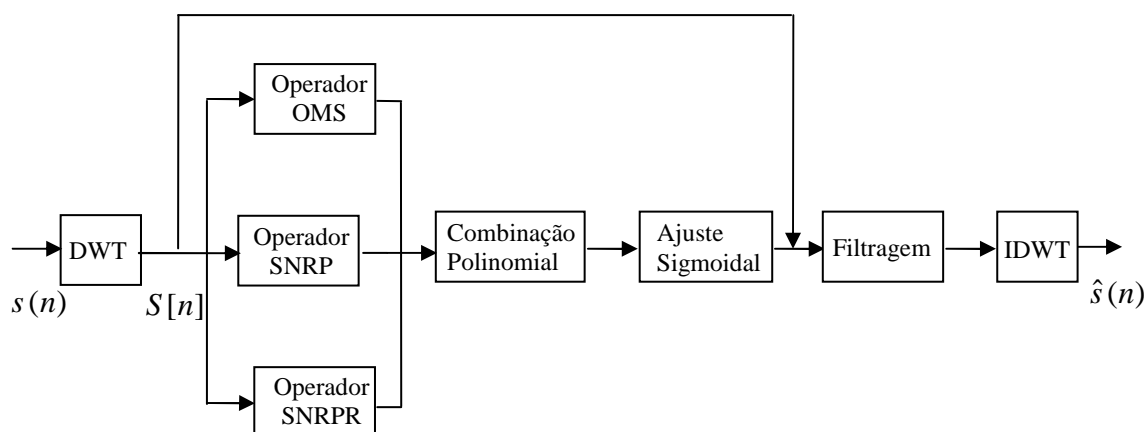


Figura 5.3 – Diagrama do Método Proposto.

De acordo com o diagrama da Figura 5.3, dado um sinal de voz ruidoso $s(n)$ no domínio do tempo, o método proposto, usando a função de transferência *fil*, é aplicado da seguinte forma:

- 1- Aplica-se a transformada *wavelet* a $s(n)$, obtendo o sinal transformado $S[n]$, sendo $s(n) = x(n) + r(n)$, onde $x(n)$ é o sinal de voz limpo e $r(n)$ é o ruído adicionado;
- 2- A partir do sinal transformado $S[n]$ obtêm-se as curvas determinadas pelos operadores propostos, OMS, SNRP e SNRPR. As características de cada um desses operadores e o porquê de se usar os três são explicadas nas próximas seções deste capítulo;
- 3- Após obtenção das curvas determinadas pelos operadores, é feita uma combinação polinomial dos três, para que as características de cada um, que não são comuns aos outros dois, possam ser usadas no processamento do sinal;
- 4- A curva obtida pela combinação polinomial sofre um ajuste sigmoidal para que seus coeficientes permaneçam no intervalo $[-1,1]$ e assim a amplitude do sinal original seja mantida;

- 5- Feito o ajuste sigmoidal, que consiste na introdução da função de transferência fil , a curva determinada pela filtragem está pronta: $Fil[n]$. A filtragem do ruído (que é a obtenção do sinal associado $Y_s[n]$) é feita através do produto dos coeficientes desta curva $Fil[n]$ pelos respectivos coeficientes do sinal $S[n]$, no domínio *wavelet*;
- 6- Finalmente, após a filtragem, o sinal é reconstruído usando a transformada *wavelet* inversa: obtendo o sinal $\hat{s}(n) = \hat{y}_s(n)$.

Na sequência são apresentados detalhadamente os processamentos dos operadores e em seguida o processamento da combinação proposta neste trabalho, que descreve o processamento da função de transferência fil . Detalhes do processamento de um dado sinal, pelos operadores, pela DWT, pela função de transferência fil e pela IDWT, são apresentados nos Apêndices A-E.

5.2 Processamento dos operadores propostos

Os processamentos propostos dependem do sinal ruidoso isoladamente ou de uma combinação deste com o ruído de fundo estimado. Cada um dos operadores é gerado de forma individualizada, extraindo características do sinal a ser processado. Posteriormente, a combinação dos mesmos, de acordo com os passos 4 e 5 citados na seção anterior, gera a curva usada para filtragem do ruído presente neste sinal. Na equação (5.1) é apresentada a função básica de processamento $f(s,r)$, onde s representa um elemento do sinal ruidoso e r um elemento que representa o ruído presente no sinal ou mesmo o próprio sinal. A variável $d = d(s,r)$ depende de cada tipo de processamento:

$$f(s,r) = \frac{|s| + |r|}{d(s,r)} \quad (5.1)$$

Para explicar a aplicação dos três processamentos, considere $S[n]$, no domínio *wavelet*, uma sequência de observação de comprimento finito (considerado aqui N uma potência n de base 2: $N=2^n$) de um sinal limpo $X[n]$ que está contaminado por ruído aditivo ou real $R[n]$, isto é:

$$S[n] = X[n] + R[n] \quad (5.2)$$

Os sinais limpo $X[n]$ e de ruído $R[n]$ são considerados processos aleatórios independentes. O objetivo é recuperar o sinal limpo $X[n]$ a partir de $S[n]$. Os três processamentos serão aplicados ao sinal ruidoso $S[n]$, no domínio *wavelet*. Serão indicados por s_i os coeficientes desse sinal. Assim, tem-se:

$$S[n]=\{s_0, s_1, s_2, \dots, s_{N-1}\} \quad (5.3)$$

Cada processamento aplica funções distintas, aqui denominadas de operadores. Para a aplicação do primeiro processamento, é considerado um operador denominado de *operador média simples* (OMS), assim como para a realização dos outros dois processamentos, os operadores são denominados como *relação sinal/ruído a priori* (SNRP) e *relação sinal/ruído a posteriori* (SNRPR), respectivamente. Usando a execução dos operadores, em cada caso, são obtidos os respectivos conjuntos associados, indicados por $F[n]$, conjuntos estes descritos nas seções 5.2.1, 5.2.2 e 5.2.3, respectivamente. Os elementos dos referidos conjuntos são coeficientes que foram calculados de acordo com cada processamento. Os elementos de cada conjunto $F[n]$ serão indicados por f_i , elementos esses calculados usando a fórmula que define cada operador. Desse modo tem-se:

$$F[n]=\{f_0, f_1, f_2, \dots, f_{N-1}\} \quad (5.4)$$

A redução de ruído em sinais de voz proposta no domínio *wavelet*, aplicando os três operadores, em cada caso, é feita através da simples multiplicação dos respectivos elementos do sinal ruidoso $S[n]$, expresso na equação (5.3), pelos correspondentes elementos do conjunto $F[n]$. Assim, tem-se:

$$Y_S[n]=\{s_0 \cdot f_0, s_1 \cdot f_1, s_2 \cdot f_2, \dots, s_{N-1} \cdot f_{N-1}\} \quad (5.5)$$

O conjunto considerado em (5.5) pode ser escrito ainda da seguinte forma:

$$Y_S[n]=\{y_0, y_1, y_2, \dots, y_{N-1}\}, \text{ sendo } y_i = s_i \cdot f_i \quad (5.6)$$

Os elementos do conjunto $Y_S[n]$ são coeficientes resultantes da caracterização da operação de redução de ruído. Concluído esse processo, aplica-se a transformada *wavelet* inversa nos coeficientes do conjunto $Y_S[n]$ obtendo-se, assim, o sinal reconstruído, no domínio do tempo, indicado por $\hat{y}_S(n)$. Todo esse processo pode ser visto com maiores detalhes nas ilustrações apresentadas nos apêndices A, B, C, D e E no final deste trabalho. Como resultante das operações acima descritas, é feita uma

combinação polinomial, definida na equação (5.10), usando os três operadores considerados e em seguida se faz uma adequação de amplitude via ajuste sigmoidal, propostos neste trabalho e descritos na seção 5.3.

Nas seções 5.2.1 – 5.2.3, serão descritas as execuções de cada operador acima citado, OMS, SNRP, SNRPR, e na seção 5.3, será descrita também a execução do operador “*combinação polinomial*”, que é uma combinação dos três operadores anteriormente considerados, bem como da execução da função de transferência “*fil*”, que é introduzido via ajuste sigmoidal, usando a combinação polinomial.

Para ilustrar as respectivas execuções, será considerado o sinal de voz limpo $x(n)$, no formato *wav*, que modela a seguinte frase:

$$x(n): \begin{cases} \text{" À medida que o tempo passa mais nos} \\ \text{convencemos da eficácia do convênio"} \end{cases} \quad (5.6')$$

A partir do sinal de voz limpo $x(n)$, após contaminação com um ruído branco $r(n)$, obtém-se o sinal ruidoso $s(n)=x(n)+r(n)$ (vide Figura 5.6). Para a descrição das execuções, serão considerados também:

- os sinais $X[n]$ e $R[n]$ obtidos usando a transformada *wavelet* discreta de *Daubechies* (dB10), a partir dos sinais $x(n)$ e $r(n)$, respectivamente;
- o sinal ruidoso $S[n]$ (vide Figura F2), obtido após contaminação com o ruído $R[n]$, usando a fórmula (5.2).

5.2.1 Primeiro processamento - Operador OMS

Dado um sinal ruidoso qualquer $S[n]=\{s_0, \dots, s_{N-1}\}$, sendo N uma potência de base 2 ($N=2^n$), no domínio *wavelet*, a redução de ruído desse sinal pode ser feita usando o operador média simples (OMS), cuja execução é definida por:

$$f_i = \frac{|s_i| + |s_{i+1}|}{2}, \text{ para } i=0, \dots, N-2, \text{ e } f_{N-1} = f_{N-2}. \quad (5.7)$$

Para a realização da redução, o sinal $S[n]$ é processado para gerar o sinal associado $Y_s[n]$, considerado também no domínio *wavelet*, usando o conjunto $F[n]=\{f_0, \dots, f_{N-1}\}$ (vide (5.4)). Esta redução de ruído é feita através da simples multiplicação dos respectivos elementos s_i de $S[n]$ pelos correspondentes elementos f_i de $F[n]$ (vide equação (5.5) e Apêndice B, para uma ilustração). Após a redução do

ruído, tendo o sinal $Y_S[n]$, esse sinal é reconstruído usando a transformada *wavelet* inversa, obtendo um sinal denotado por $\hat{y}_S(n)$, no domínio do tempo (nos Apêndices A e C é apresentada uma ilustração desse processo).

Para ilustrar a execução do operador OMS são considerados:

- o sinal de voz limpo $x(n)$ que representa a frase considerada em (5.6’);
- o sinal ruidoso $s(n)=x(n)+r(n)$, contaminado com ruído $r(n)$;
- os sinais $X[n]$ e $R[n]$, obtidos a partir dos sinais $x(n)$ e $r(n)$, respectivamente;
- o sinal ruidoso $S[n]$, obtido após contaminação com o ruído $R[n]$, usando (5.2).

As formas de onda do sinal $X[n]$ e do sinal processado $Y_X[n]$ (usando o operador OMS) podem ser visualizadas no apêndice F no final deste trabalho (Figuras F1 e F4).

Na Figura 5.5 apresenta-se, em azul, uma representação do sinal de voz limpo $x(n)$ e, em amarelo, uma representação do mesmo sinal processado $\hat{y}_S(n)$, obtido usando a transformada *wavelet* inversa após a aplicação do operador OMS. Como se pode observar na Figura 5.5, o processamento reduz significativamente a amplitude do sinal. Na Figura 5.7 é apresentada uma versão amplificada do sinal processado $\hat{y}_S(n)$, ou seja, a mesma curva amarela apresentada na Figura 5.5. Fazendo uma comparação visual entre a curva da Figura 5.4 e a curva apresentada na Figura 5.7, é possível verificar que o sinal processado usando o operador OMS causa fortes distorções nos trechos de voz do sinal. Porém, comparando as Figuras 5.6 e 5.9, pode-se verificar também que o operador OMS é eficiente na redução do ruído, pois embora reduza fortemente a amplitude do sinal, os coeficientes dos trechos de voz são destacados.

O gráfico do operador OMS ($F[n]$), dado pela equação (5.7), é uma curva poligonal (passando pelos pontos $(i + 1, f_i)$) no qual cada elemento f_i é obtido pela média dos respectivos valores absolutos de cada duas amostras consecutivas do sinal $S[n]$ (vide também Figura B2).

É indicado por $F'[n]$ o conjunto dos elementos do operador obtidos sem a consideração do valor absoluto na equação (5.7) (um exemplo pode ser visto na Figura B3). O uso do operador OMS dado pela equação (5.7) se justifica pelo fato de que, o

produto entre os coeficientes do sinal $S[n]$ pelos coeficientes de $F[n]$ apenas reduzirá o sinal processado, ao passo que quando se faz o produto dos coeficientes do sinal $S[n]$ pelos coeficientes $F'[n]$, além de reduzir o sinal processado, a fase de alguns coeficientes do sinal processado será invertida, e isto causará distorções no sinal processado (vide Apêndice B, Figura B4). Quando o valor absoluto é usado, nenhum coeficiente do sinal sofre inversão de fase. Assim, ao multiplicar s_i de $S[n]$ por f_i de $F[n]$, o sinal tem a sua forma de onda mantida, porém há uma forte redução de amplitude como se pode observar na Figura 5.5, mas ocorrendo também uma forte redução de ruído justificada por esta drástica redução de amplitude.

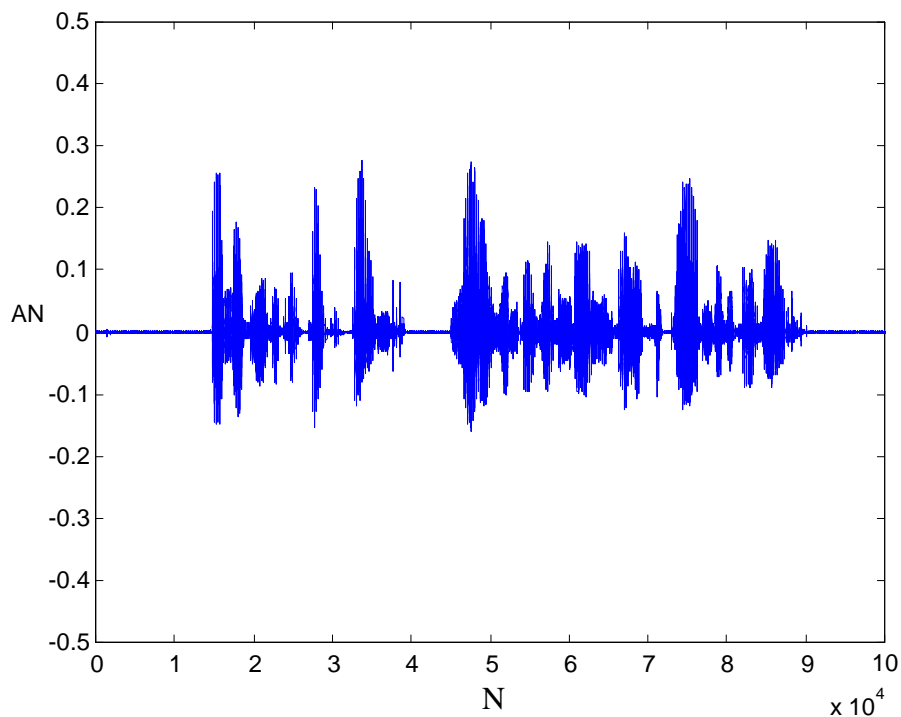


Figura 5.4 – Sinal Limpo $x(n)$.

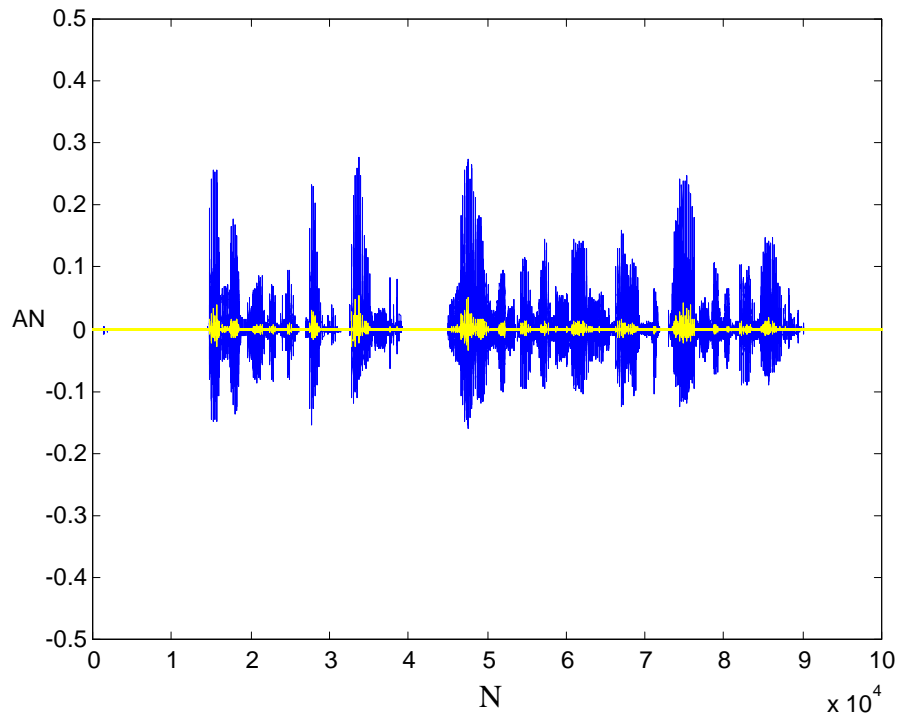


Figura 5.5 – Sinal de voz limpo $x(n)$ e sinal processado $\hat{y}_s(n)$, usando o operador OMS e a transformada *wavelet* inversa.

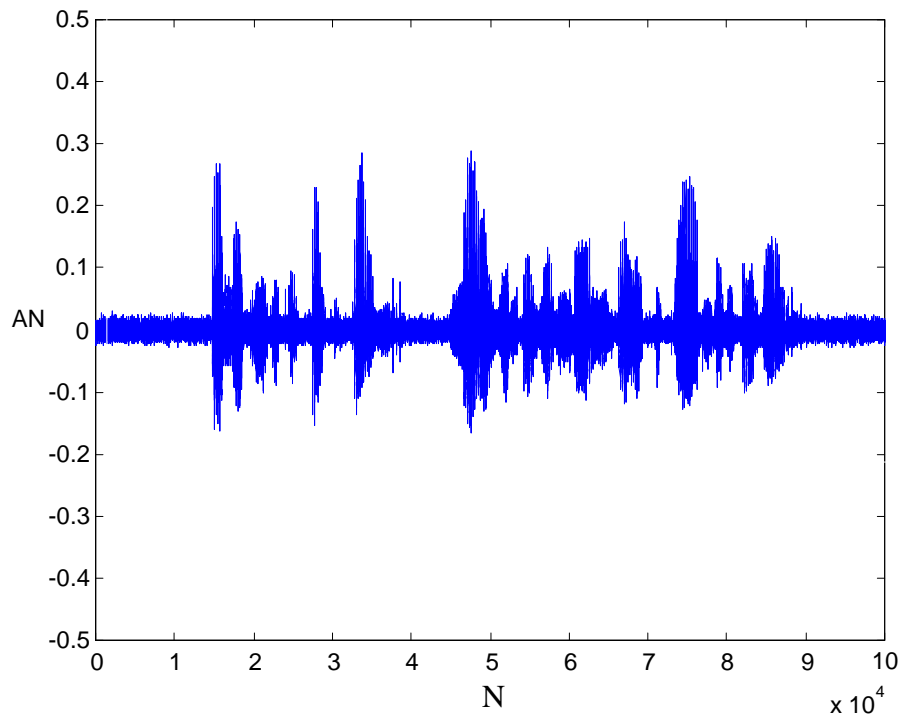


Figura 5.6 – Sinal de voz $s(n)$ contaminado por ruído branco, $r(n)$.

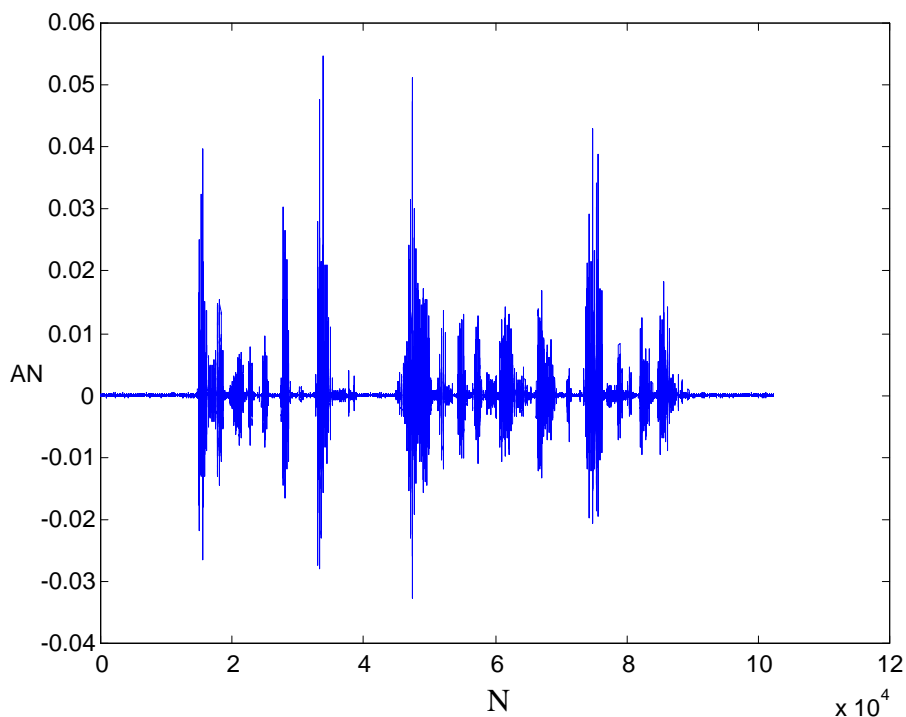


Figura 5.7 – Amplificação do sinal processado $\hat{y}_s(n)$, usando o operador OMS e a transformada *wavelet* inversa.

O processamento realizado pelo operador OMS reduz drasticamente a amplitude do sinal como se pode observar na Figura 5.5. Esta forte redução na amplitude do sinal indica a introdução de distorções em seus trechos de voz do sinal, devido ao fato de se eliminar muito ruído e, conseqüentemente, coeficientes de voz também são eliminados. Porém, quando se compara as duas curvas da Figura 5.5, é possível verificar que a forma de onda do sinal é mantida, isto significa que os coeficientes significativos no domínio *wavelet*, embora tenham suas amplitudes reduzidas, não são eliminados, mas sim atenuados. Isto é, as características principais do sinal são mantidas.

Dado um sinal de voz limpo $x(n)$, representado na Figura 5.4, o sinal transformado $X[n]$ é contaminado, com ruído branco $R[n]$ e com ruído colorido $R_0[n]$, para se obter os respectivos sinais ruidosos $S[n]$ e $S_0[n]$, de acordo com a equação (5.2). Após processamento desses sinais: limpo e ruidosos, usando o operador OMS e a transformada *wavelet* inversa, obtêm-se, respectivamente, os sinais $\hat{y}_x(n)$, $\hat{y}_s(n)$ e $\hat{y}_0(n)$. Nas Figuras 5.8, 5.9 e 5.10 são apresentadas, respectivamente, as curvas que representam esses sinais. As formas de onda dos sinais transformados

$X[n]$, $S[n]$, $S_0[n]$, $Y_x[n]$, $Y_s[n]$ e $Y_0[n]$, podem ser visualizadas no apêndice F (Figuras F1 – F6). Pode-se ainda observar que as formas de onda mantêm as mesmas características no domínio *wavelet*.

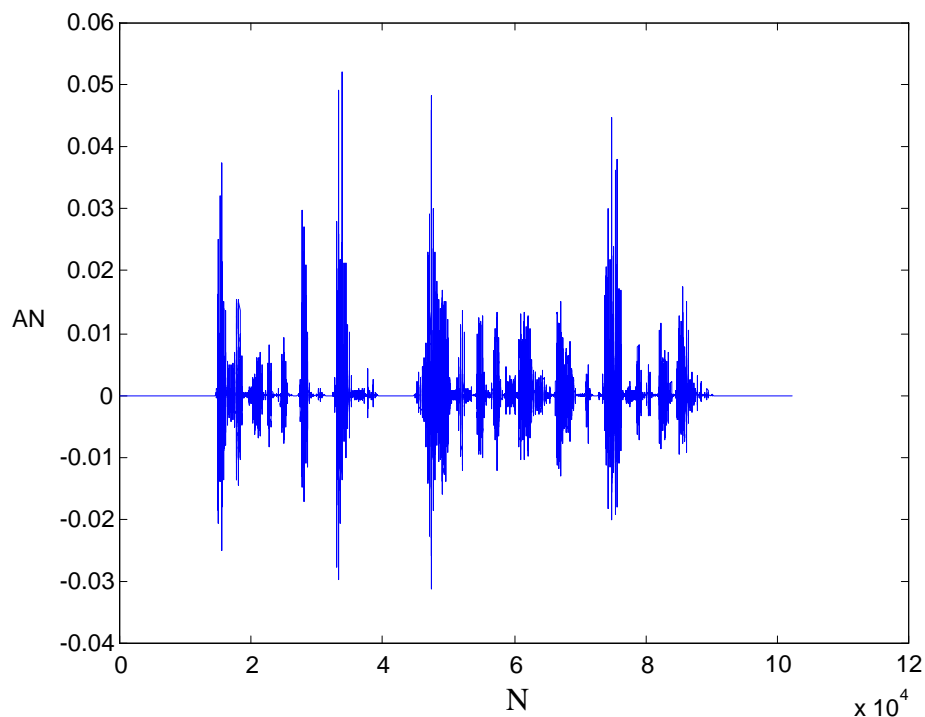


Figura 5.8 – Sinal processado $\hat{y}_x(n)$ a partir do sinal original limpo $x(n)$, usando o operador OMS e a transformada *wavelet* inversa.

Comparando as três figuras (5.4, 5.5 e 5.8) pode-se observar que, mesmo após a adição do ruído, o processamento do operador OMS mantém as principais características do sinal, pois as três figuras têm o mesmo comportamento nos trechos de voz do sinal.

A forte redução ocorrida na amplitude do sinal se deve ao fato que, ao se aplicar a transformada *wavelet* num sinal cuja amplitude está normalizada entre -1 e 1, os coeficientes *wavelets* assumem valores absolutos menores que 1, com a grande maioria próxima de zero. Como cada coeficiente da curva de filtragem gerada pelo operador OMS é sempre a média entre dois coeficientes consecutivos do sinal transformado, o produto dos coeficientes do operador OMS pelo sinal transformado faz com que a amplitude do mesmo diminua mais ainda.

Na equação (5.10), o termo y é elevado ao cubo. Como y representa o operador OMS, então essa parcela reduzirá mais ainda a amplitude do sinal processado, conforme se pode observar na seção 5.3. Porém, mesmo introduzindo distorções nos trechos de

voz, os coeficientes significativos ainda serão mantidos pelo operador OMS, de forma que essas distorções serão mínimas para este novo sinal e assim não comprometem a inteligibilidade do sinal de voz. Assim, o uso do operador OMS na combinação proposta na equação (5.10) se justifica pelo fato do mesmo manter os coeficientes significativos na reconstrução do sinal, independente do tipo de ruído pelo qual o sinal foi contaminado. Detalhes destas operações, via um exemplo, são apresentados no Apêndice D.

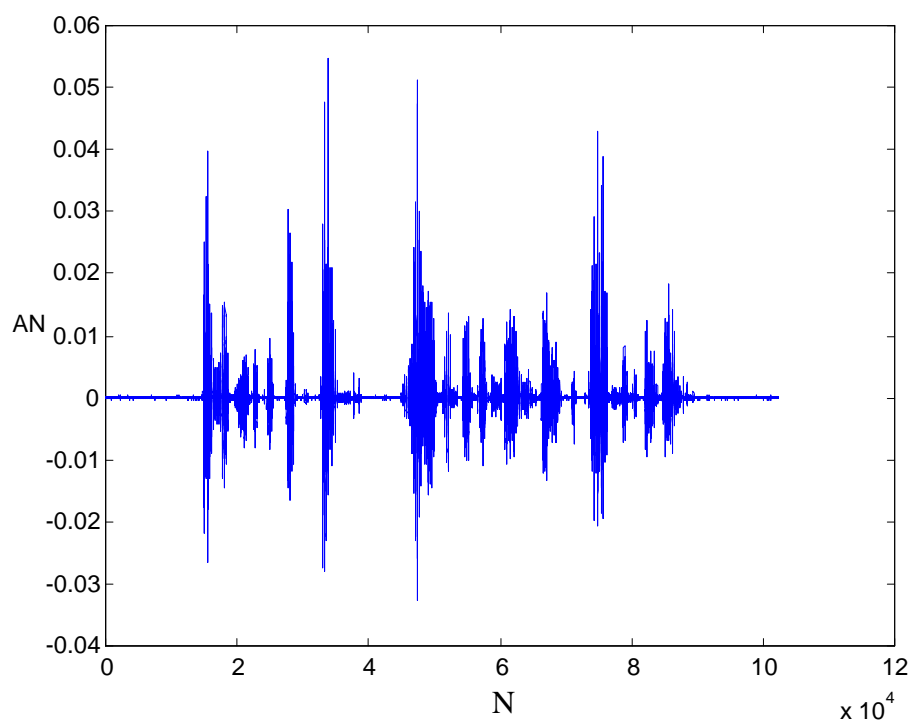


Figura 5.9 – Sinal processado $\hat{y}_s(n)$, usando o operador OMS e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x(n)$ contaminado com ruído branco $r(n)$.

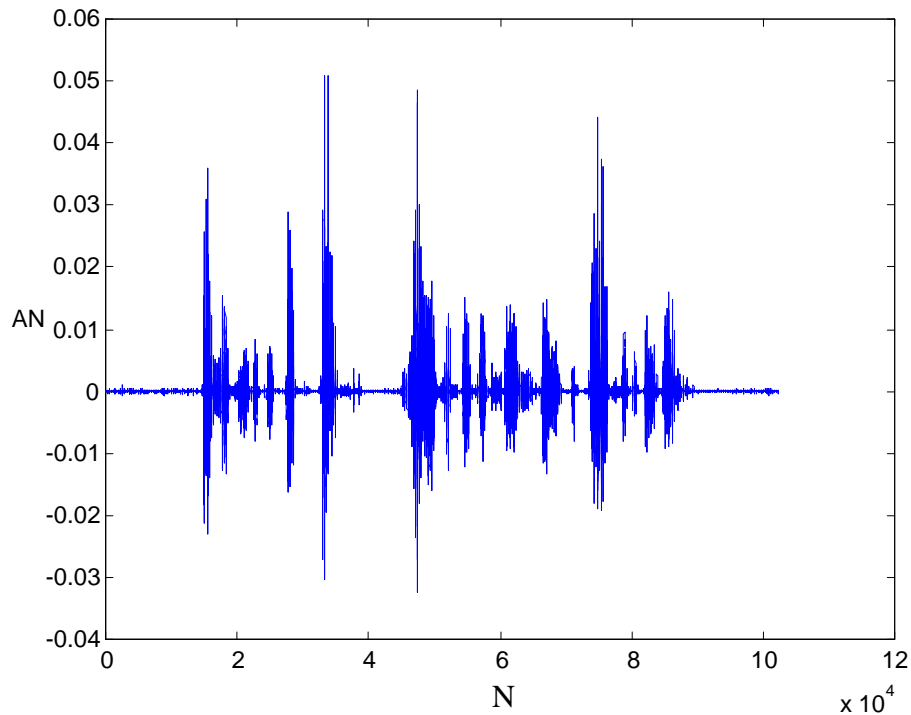


Figura 5.10 – Sinal processado $\hat{y}_0(n)$, usando o operador OMS e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x(n)$ contaminado com ruído colorido $r_0(n)$.

5.2.2 Segundo processamento – Operador SNRP

Dado um sinal ruidoso qualquer $S[n]=\{s_0, \dots, s_{N-1}\}$, sendo N uma potência n de base 2 ($N=2^n$), no domínio *wavelet*, a redução de ruído desse sinal pode ser feita usando o operador relação sinal/ruído a priori (SNRP) (SOARES et al., 2008 (B)), definido por:

$$f_i = \frac{|s_i|}{\alpha + |s_i|}, \text{ para } i=0, \dots, N-2 \quad (5.8)$$

sendo α um valor entre 0 e 1.

Para realizar a redução, o sinal $S[n]$ é processado para gerar o sinal $Y_S[n]$, considerado também no domínio *wavelet*, usando o conjunto $F[n]=\{f_0, \dots, f_{N-1}\}$ (vide equação (5.4)). Esta redução de ruído é feita através da simples multiplicação dos respectivos elementos s_i de $S[n]$ pelos correspondentes elementos f_i de $F[n]$ (vide equação (5.5) e Apêndice B: o processo neste caso é análogo ao que foi feito para o caso do operador OMS). Após a redução do ruído, tendo o sinal $Y_S[n]$, esse sinal é

reconstruído usando a transformada *wavelet* inversa, obtendo um sinal denotado por $\hat{y}_s(n)$, no domínio do tempo (para uma ilustração vide Apêndices A e C, o processo é análogo ao que foi feito para o caso do operador OMS).

Para ilustrar a execução do operador SNRP são considerados: $\alpha=0,01$,

- o sinal de voz limpo $x(n)$ que representa a frase considerada em (5.6');
- o sinal ruidoso $s(n)=x(n)+r(n)$, contaminado com ruído $r(n)$;
- os sinais $X[n]$ e $R[n]$, obtidos a partir dos sinais $x(n)$ e $r(n)$, respectivamente;
- o sinal ruidoso $S[n]$, obtido após contaminação com o ruído $R[n]$, usando (5.2).

As formas de onda do sinal $X[n]$ e do sinal processado $Y_x[n]$ (usando o operador SNRP) podem ser visualizadas no apêndice F no final deste trabalho (Figuras F1 e F7).

Na Figura 5.11 apresenta-se, em azul, uma representação do sinal de voz limpo $x(n)$ e, em vermelho, uma representação do mesmo sinal processado $\hat{y}_s(n)$, obtido usando a transformada *wavelet* inversa após a aplicação do operador SNRP. Pode se observar que este operador reduz a amplitude do sinal, sendo eficiente na redução de ruído. Observando a Figura 5.12, que apresenta apenas o sinal processado $\hat{y}_s(n)$ pelo operador SNRP, constata-se que não há fortes distorções nos trechos de voz. Embora haja redução na amplitude do sinal, esta redução não é tão forte quanto a causada pelo operador OMS.

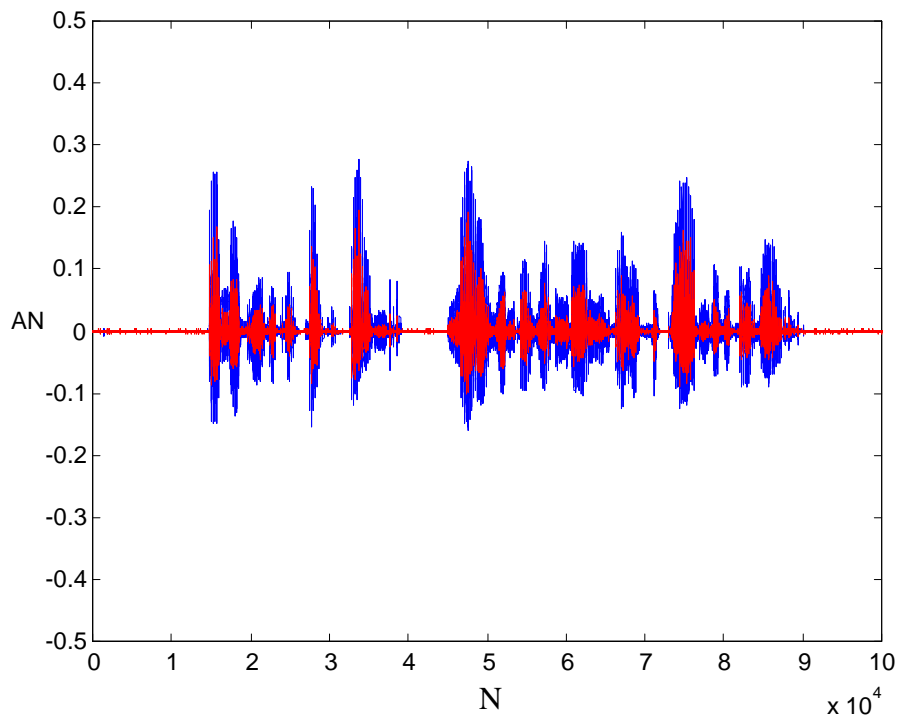


Figura 5.11 – Sinal de voz limpo $x(n)$ e sinal processado $\hat{y}_s(n)$, usando o operador SNRP e a transformada *wavelet* inversa.

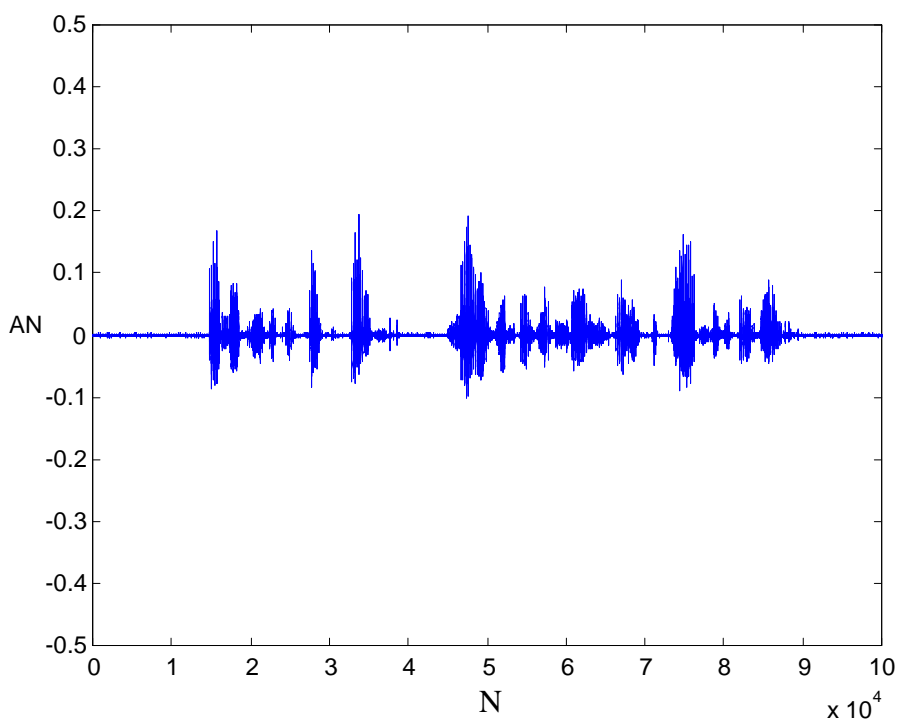


Figura 5.12 – Processamento usando o operador SNRP (sinal processado $\hat{y}_s(n)$).

O operador SNRP, da forma que é calculado, conforme a equação (5.8), faz a estimativa do ruído presente em cada coeficiente do sinal s_i , característicos dos métodos

baseados na SNR *a priori* (COHEN, 2004). Por isso, ao efetuar a multiplicação dos elementos de s_i pelos respectivos elementos de f_i , não há distorções nos trechos de voz.

O processamento do operador SNRP reduz a amplitude do sinal, porém não o faz de maneira tão drástica quanto o operador OMS. Isto pode ser observado na Figura 5.14. Na mesma figura também se pode observar que este operador é eficiente na redução do ruído presente no sinal. O operador SNRP também mantém a forma de onda do sinal, significando que ao ser aplicado no domínio *wavelet*, os coeficientes significativos são mantidos ou apenas atenuados, porém esta atenuação é bem menor que no operador anterior, de forma que as distorções também são menores. Dado um sinal de voz limpo $x(n)$, representado na Figura 5.4, esse sinal é contaminado, com ruído branco $R[n]$ e com ruído colorido $R_0[n]$, para se obter o respectivo sinal ruidoso $S[n]$ e $S_0[n]$, de acordo com a equação (5.2). Após processamento desses sinais: limpo e ruidosos, usando o operador SNRP e a transformada *wavelet* inversa, obtêm-se respectivamente os sinais $\hat{y}_x(n)$, $\hat{y}_s(n)$ e $\hat{y}_0(n)$. Nas Figuras 5.13, 5.14 e 5.15 são apresentadas, respectivamente, as curvas que representam esses sinais. As formas de onda dos sinais transformados $X[n]$, $S[n]$, $S_0[n]$, $Y_x[n]$, $Y_s[n]$ e $Y_0[n]$, podem ser visualizadas no apêndice F (Figuras F1 – F3 e F7 – F9). Pode-se ainda observar que as formas de onda mantêm a mesma característica no domínio *wavelet*.

A diferença de redução de amplitude entre os operadores OMS e SNRP está no fato de que no operador OMS, conforme a equação (5.7), o fator de redução para cada coeficiente é praticamente igual ao valor absoluto do coeficiente, enquanto que no SNRP, de acordo com a equação (5.8), esse tem valor absoluto próximo de 1, porém menor.

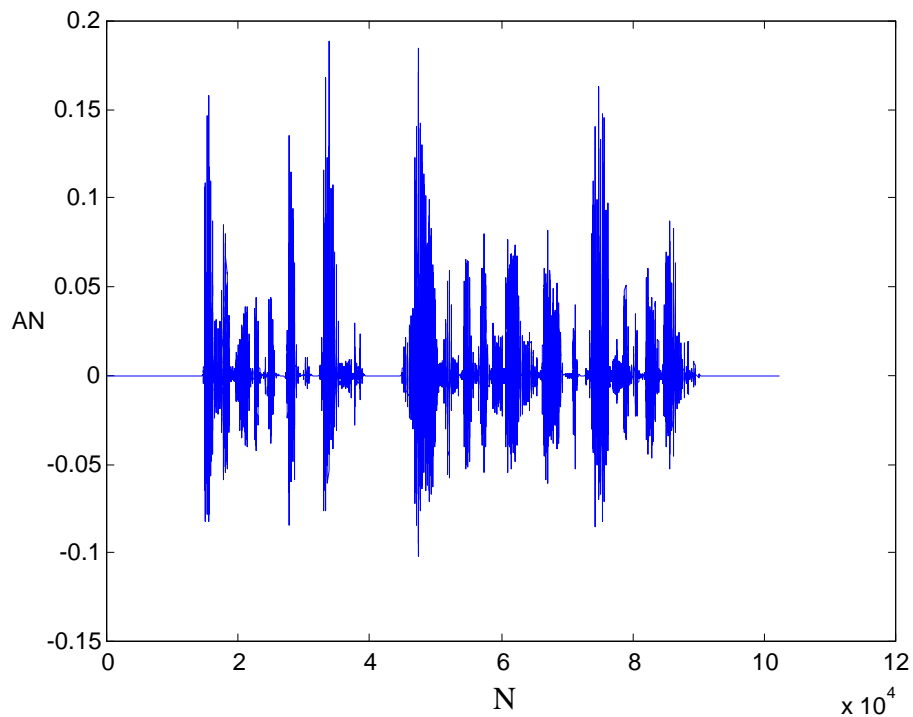


Figura 5.13 – Sinal processado $\hat{y}_x(n)$ a partir do sinal original limpo $x(n)$, usando o operador SNRP e a transformada *wavelet* inversa.

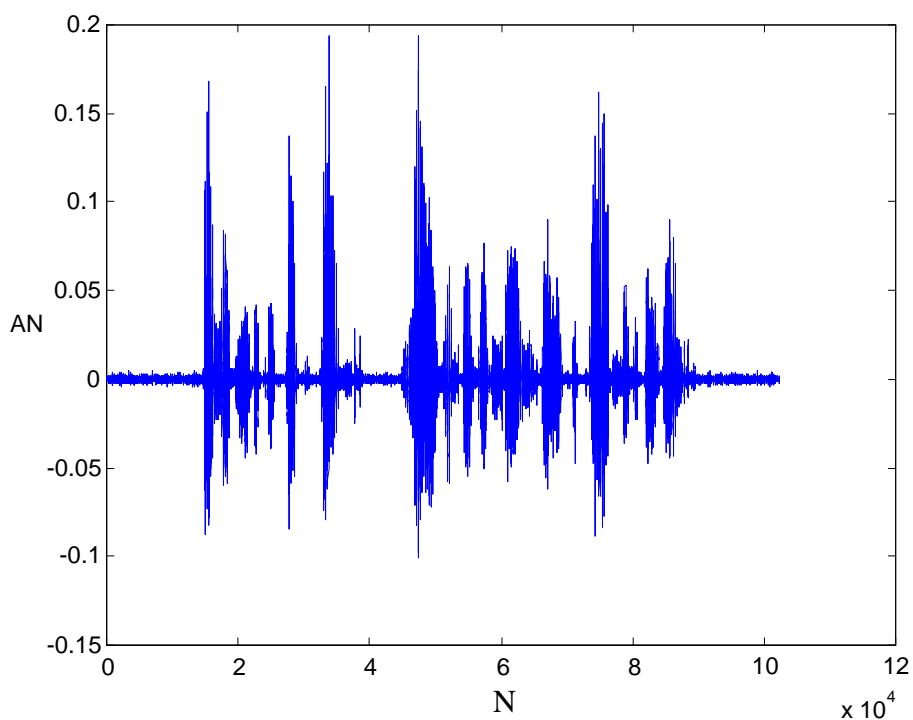


Figura 5.14 – Sinal processado $\hat{y}_s(n)$, usando o operador SNRP e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído branco $r(n)$.

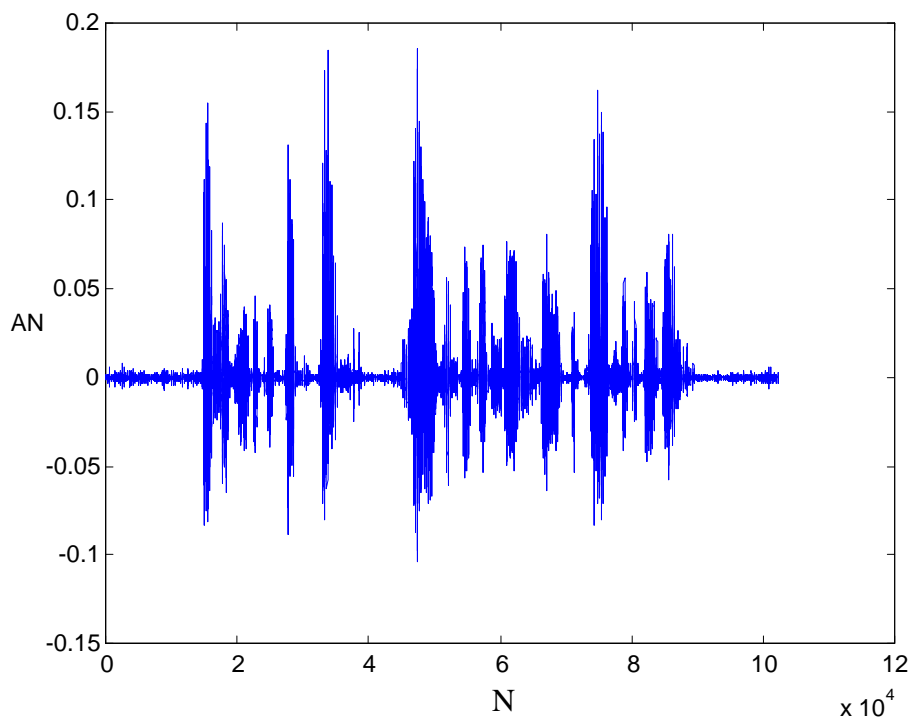


Figura 5.15 – Sinal processado $\hat{y}_0(n)$, usando o operador SNRP e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído colorido $r_0(n)$.

Na equação (5.10), o termo z , que representa o operador SNRP, é elevado ao quadrado. Como z é menor que 1, a redução do ruído e da amplitude do sinal será maior ainda. Porém, o objetivo dessa operação é manter os coeficientes significativos que foram eliminados pelo operador OMS.

5.2.3 Terceiro processamento – Operador SNRPR

Dado um sinal ruidoso qualquer $S[n]=\{s_0, \dots, s_{N-1}\}$, sendo N uma potência n de base 2 ($N=2^n$), no domínio *wavelet*, a redução de ruído desse sinal pode ser feita usando o operador relação sinal/ruído *a posteriori* (SNRPR), cuja execução é definida por:

$$f_i = \frac{|s_i|}{\alpha + |sr_i|}, \text{ para } i=0, \dots, N-2 \quad (5.9)$$

sendo α um valor entre 0 e 1, sr_i é um vetor de mesmo comprimento que s_i e cada componente representa a média do ruído presente nos correspondentes componentes das janelas do último trecho de silêncio do sinal de voz em processamento.

Para efetuar a redução, o sinal $S[n]$ é processado para gerar o sinal associado $Y_S[n]$, considerado também no domínio *wavelet*, usando o conjunto $F[n]=\{f_0, \dots, f_{N-1}\}$ (vide equação (5.4)). Esta redução de ruído é feita através da simples multiplicação dos respectivos elementos s_i de $S[n]$ pelos correspondentes elementos f_i de $F[n]$ (vide equação (5.5) e Apêndice B: o processo neste caso é análogo ao que foi feito para o caso do operador OMS). Após a redução do ruído, tendo o sinal $Y_S[n]$, esse sinal é reconstruído usando a transformada *wavelet* inversa, obtendo um sinal denotado por $\hat{y}_S(n)$, no domínio do tempo (vide Apêndices A e C: o processo neste caso é análogo ao que foi feito para o caso do operador OMS).

Porém, este processamento é usado apenas para as regiões de voz do sinal. Quando o operador SNRPR é usado, as regiões de silêncio do sinal são tratadas usando o operador OMS.

Para ilustrar a execução do operador SNRPR são considerados: $\alpha=0,01$, o vetor sr_i ,

- o sinal de voz limpo $x(n)$ que representa a frase considerada em (5.6’);
- o sinal ruidoso $s(n)=x(n)+r(n)$, contaminado com ruído $r(n)$;
- os sinais $X[n]$ e $R[n]$, obtidos a partir dos sinais $x(n)$ e $r(n)$, respectivamente;
- o sinal ruidoso $S[n]$, obtido após contaminação com o ruído $R[n]$, usando (5.2).

As formas de onda do sinal $X[n]$ e do sinal processado $Y_X[n]$ (usando o operador SNRPR) podem ser visualizadas no apêndice F no final deste trabalho (Figuras F1 e F10).

Na Figura 5.16 apresenta-se, em azul, uma representação do sinal de voz limpo $x(n)$ e, em vermelho, uma representação do mesmo sinal processado $\hat{y}_S(n)$, obtido usando a transformada *wavelet* inversa após a aplicação do operador SNRPR. Verifica-se que, embora haja forte redução de ruído, há também uma grande distorção nos trechos de voz do sinal.

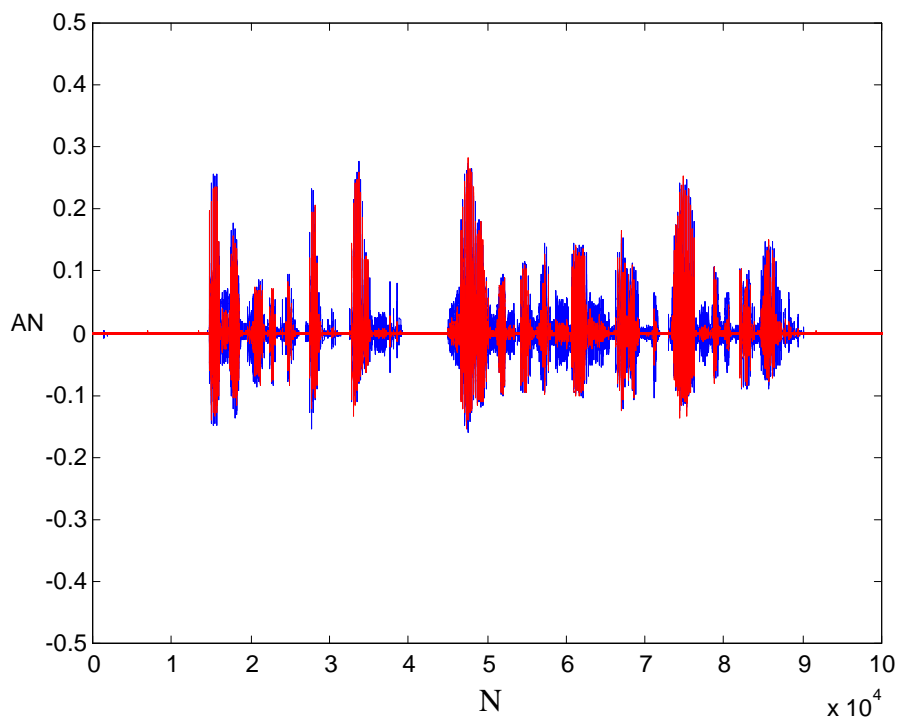


Figura 5.16 – Sinal processado $\hat{y}_s(n)$ a partir do sinal original limpo $x(n)$, usando o operador SNRPR e a transformada *wavelet* inversa.

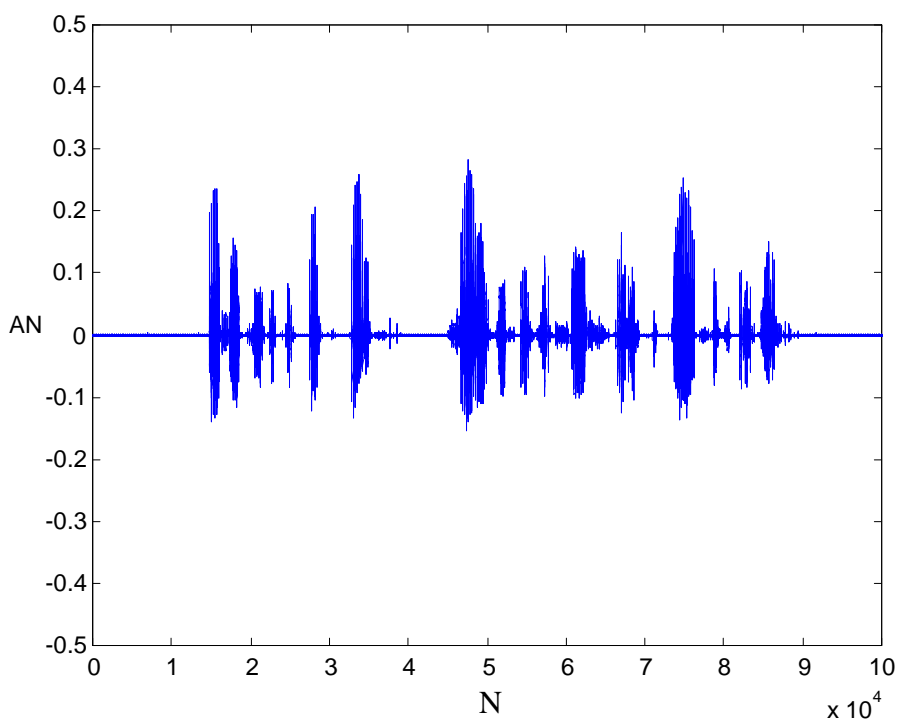


Figura 5.17 – Sinal processado $\hat{y}_s(n)$ com o operador SNRPR.

O operador SNRPR é calculado praticamente da mesma forma que o operador SNRP. Porém utilizando a média do ruído do último trecho de silêncio. Isto se justifica,

quando o sinal é contaminado por ruído estacionário, pois o ruído se espalha uniformemente pelo sinal.

Dentre as desvantagens deste processamento está o fato do tratamento diferenciado para silêncio e voz. Este tipo de processamento geralmente é desagradável para o ouvinte, pois causa “estampidos” na audição do sinal.

O processamento realizado pelo operador SNRPR é o que menos reduz a amplitude do sinal processado e, em alguns trechos, este sinal tem a mesma amplitude do sinal original. Isto significa que a curva de atenuação gerada através da equação (5.9) contém coeficientes cujos valores são menores ou iguais a 1, ao contrário dos outros dois operadores que apresentam apenas valores menores do que 1. Dado um sinal de voz limpo $x(n)$, representado na Figura 5.4, esse sinal é contaminado, com ruído branco $R[n]$ e com ruído colorido $R_0[n]$, para se obter o respectivo sinal ruidoso $S[n]$ e $S_0[n]$, de acordo com a equação (5.2). Após processamento desses sinais: limpo e ruidosos, usando o operador SNRPR, obtêm-se respectivamente os sinais $\hat{y}_x(n)$, $\hat{y}_s(n)$ e $\hat{y}_0(n)$. Nas Figuras 5.18, 5.19 e 5.20 são apresentadas, respectivamente, as curvas que representam esses sinais. As formas de onda dos sinais transformados $X[n]$, $S[n]$, $S_0[n]$, $Y_x[n]$, $Y_s[n]$ e $Y_0[n]$, podem ser visualizadas no apêndice F (Figuras F1 – F3 e F10 – F12). Pode-se ainda observar que as formas de onda mantêm as mesmas características no domínio *wavelet*. Observando as três Figuras (5.18, 5.19 e 5.20), verifica-se que os coeficientes mantidos na reconstrução do sinal são os mesmos para as três versões do sinal. Nos sinais contaminados por ruído, há forte eliminação de ruído e conseqüentemente introdução de distorções nos trechos de voz. Porém, essas distorções são menores que nos outros operadores, pois muitos coeficientes originais são mantidos.

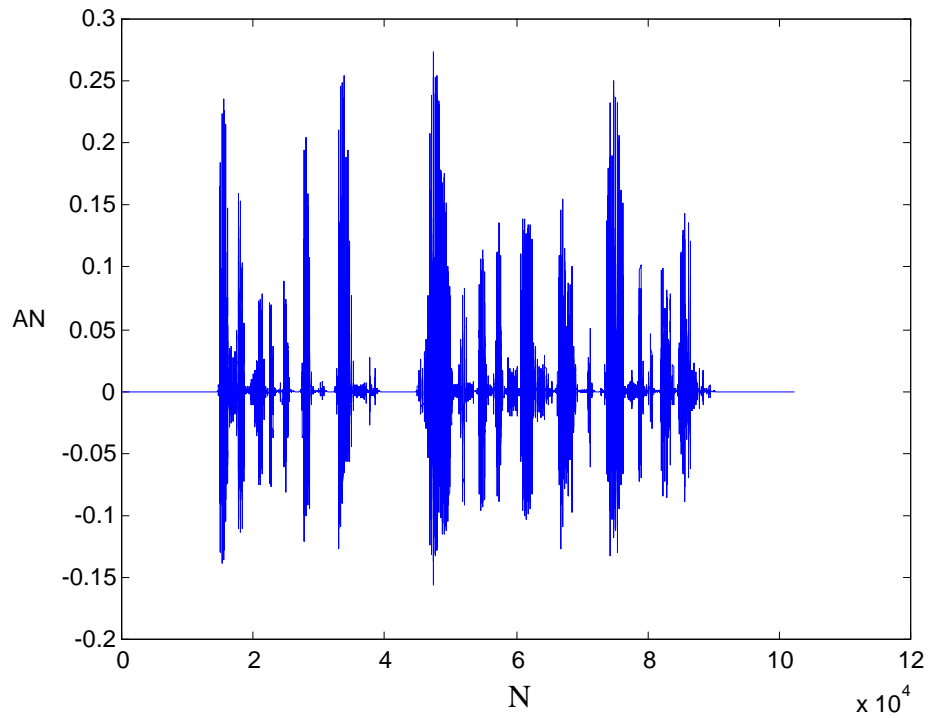


Figura 5.18 – Sinal processado $\hat{y}_x(n)$ a partir do sinal original limpo $x(n)$, usando o operador SNRPR e a transformada *wavelet* inversa.

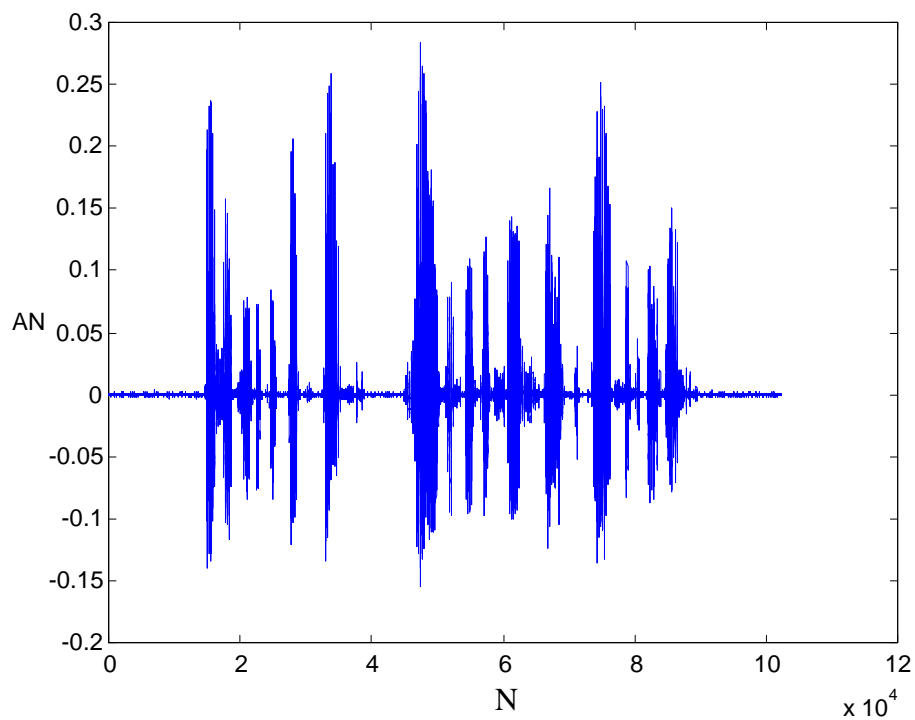


Figura 5.19 – Sinal processado $\hat{y}_s(n)$, usando o operador SNRPR e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído branco $r(n)$.

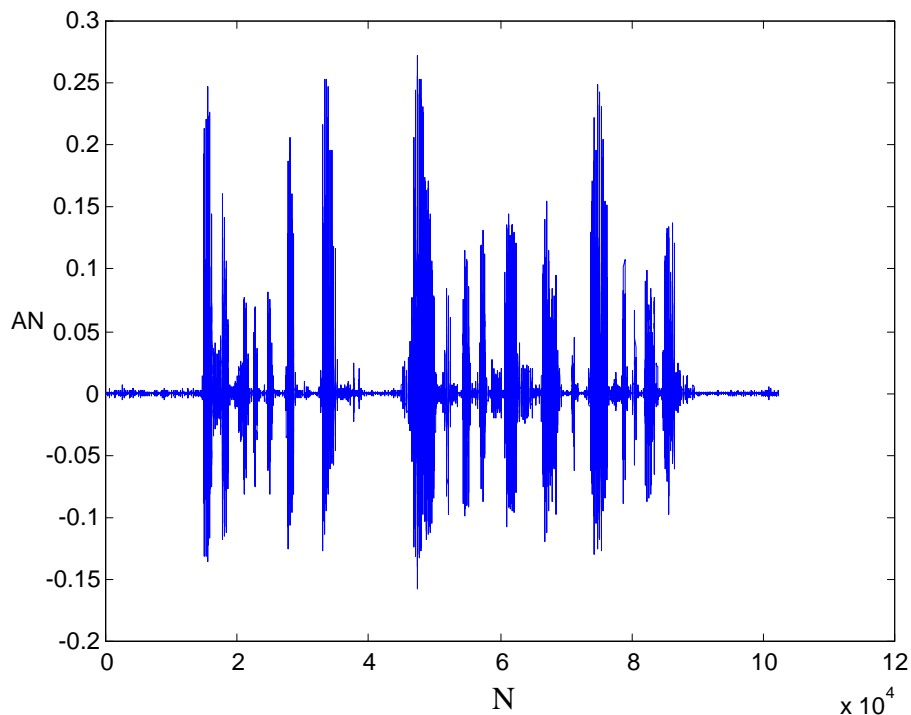


Figura 5.20 – Sinal processado $\hat{y}_0(n)$, usando o operador SNRPR e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x(n)$, contaminado com ruído colorido $r_0(n)$.

Na equação (5.10), o expoente do termo w , que representa o operador SNRPR, é 1, significando que os coeficientes obtidos com o processamento desse operador não sofrerão alterações. Pois, como os valores dos coeficientes da curva de atenuação do SNRPR são menores ou iguais a 1, esse operador será responsável por manter a maioria dos coeficientes originais do sinal, enquanto que os outros operadores, que apresentam amplitudes menores para os sinais processados, ajudarão a recompor os coeficientes que tiveram suas amplitudes reduzidas.

Em geral, objetivo da soma apresentada na equação (5.10) é que um operador possa manter os coeficientes significativos que foram eliminados pelos outros dois.

Na próxima seção, a combinação dos três operadores é apresentada, juntamente com uma adequação de amplitude via ajuste sigmoidal. O objetivo desta combinação, que é feita de forma polinomial, é aproveitar as vantagens de cada um dos três processamentos considerados anteriormente, a fim de se chegar à reconstrução mais próxima possível do sinal original.

5.3 Processamento via Combinação Polinomial

O processamento denominado Combinação Polinomial combina os operadores usados nos três processamentos apresentados anteriormente, com o objetivo de reduzir ruído e evitar distorções, como será descrito a seguir. O processamento usado é o mesmo tanto para trechos de silêncio quanto para trechos de voz.

Como dois dos processamentos apresentados reduzem significativamente o ruído, naturalmente eles introduzem fortes distorções nos trechos de voz, pois a redução também é forte nesses trechos. Assim, minimizando a redução de ruído, as distorções também serão minimizadas, pois o ruído se espalha de maneira uniforme pelo sinal.

Uma alternativa para a introdução de um operador visando a redução de ruído, num sinal contaminado, consiste em aumentar os valores dos operadores usados nos processamentos descritos anteriormente e considerar a soma dos mesmos.

Para esse objetivo, considerando um sinal $S[n]$ no domínio *wavelet*, tendo $N=2^n$ coeficientes, fazendo, respectivamente, nas equações (5.7), (5.8) e (5.9)

$$y_j = \frac{|s_j| + |s_{j+1}|}{2},$$

$$z_j = \frac{|s_j|}{\alpha + |s_j|}$$

e

$$w_j = \frac{|s_j|}{\alpha + |sr_j|}$$

a *combinação polinomial* proposta é a função $F_{cp}[n] = \{f_1, \dots, f_{N-1}\}$, cujos elementos f_j são obtidos de acordo com a seguinte equação:

$$f_j = y_j^3 + z_j^2 + w_j \quad (5.10)$$

Esta combinação polinomial atua no domínio *wavelet* da mesma forma que nos processamentos anteriormente considerados, ou seja, a redução do ruído, que consiste na obtenção do sinal processado $\hat{y}_s(n)$ (usando o conjunto $F_{cp}[n]$ acima) é feita através da simples multiplicação dos respectivos elementos s_j de $S[n]$ pelos correspondentes elementos f_j de $F_{cp}[n]$ definidos na equação (5.10) (vide Apêndice B: o processo realizado usando o conjunto $F_{cp}[n]$ acima considerado é semelhante ao

processo aplicado ao operador OMS). Este procedimento é feito tanto para janelas de silêncio quanto para janelas de voz.

Para ilustrar a execução do operador *combinação polinomial* e da função de transferência *fil*, que será introduzida a seguir, são considerados: $\alpha=0,01$, o vetor sr_j ,

- o sinal de voz limpo $x(n)$ que representa a frase considerada em (5.6');
- o sinal ruidoso $s(n)=x(n)+r(n)$, contaminado com ruído $r(n)$;
- os sinais $X[n]$ e $R[n]$, obtidos a partir dos sinais $x(n)$ e $r(n)$, respectivamente;
- o sinal ruidoso $S[n]$, obtido após contaminação com o ruído $R[n]$, usando (5.2).

A aplicação do processamento, usando o operador *combinação polinomial*, causa aumento demasiado na amplitude do sinal como se pode observar na Figura 5.21, pois o sinal $x(n)$ é o mesmo representado na Figura 5.4 e, conseqüentemente, introduz fortes distorções nos trechos de voz. Este problema é causado porque elementos, no domínio *wavelet*, após a soma apresentada na equação (5.10), passam a ter amplitude fora do intervalo compreendido entre -1 e 1, causando assim fortes distorções no sinal processado.

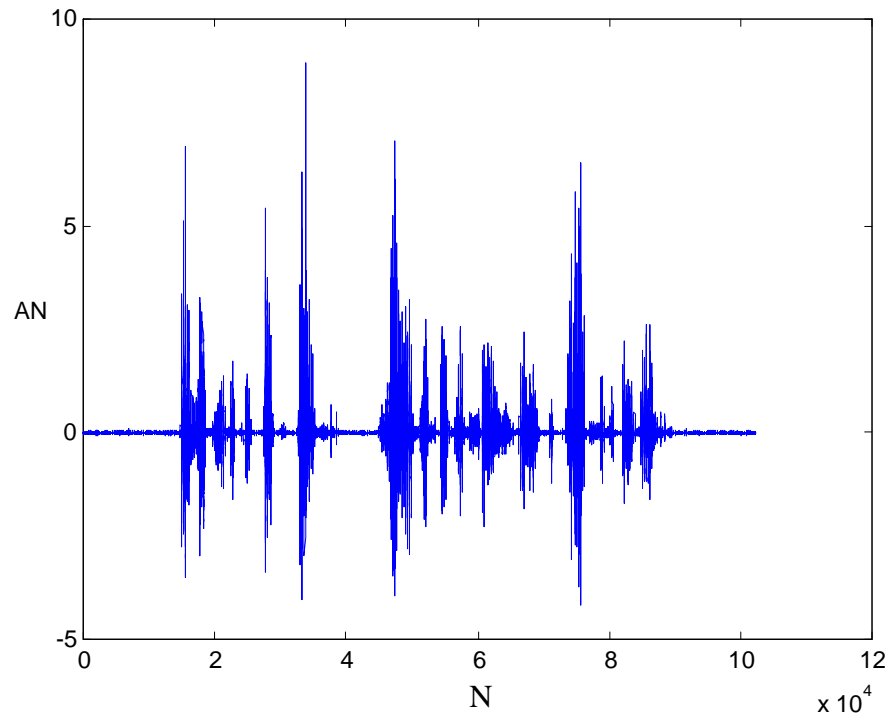


Figura 5.21 – Sinal processado $\hat{y}_s(n)$, via Combinação Polinomial e a transformada *wavelet* inversa.

Para evitar o aumento da amplitude do sinal, propõe-se *uma adequação de amplitude via ajuste sigmoidal*. Para tanto, o sinal $S[n]$ que está sendo considerado, será processado utilizando a função de transferência definida na equação (5.11), usando a função sigmoide, função esta que faz a atenuação da amplitude do sinal deixando-a no intervalo $[-1,1]$,

$$fil_j = \left| \frac{1 - e^{-\sigma_j}}{1 + e^{-\sigma_j}} \cdot \frac{1 - e^{\sigma_j}}{1 + e^{\sigma_j}} \right| \quad (5.11)$$

sendo f_j definida em (5.10). O objetivo deste ajuste é combinar os três processamentos apresentados na seção 5.2, mantendo a amplitude do sinal original.

Assim, o método que está sendo proposto utiliza uma função de transferência, chamada de *fil*, no processamento de sinal, atuando simultaneamente com a equação (5.10). A função *fil* é gerada através do produto de duas sigmoides inversas em relação ao eixo horizontal. A função sigmoide é usada para normalizar a amplitude do sinal processado no intervalo $[-1,1]$. Esta função de transferência *fil* atua como um filtro em processamento de sinais de voz. A diferença está no fato de que na filtragem em

processamento de sinais, o filtro separa o sinal em altas e baixas frequências através de uma convolução que pode ser efetuada em vários domínios. Já no método que está sendo proposto, a função de transferência, fil , atua no domínio *wavelet* da mesma forma que nos processos anteriores, no sentido de que, a redução do ruído do sinal $S[n]$, que consiste na obtenção do sinal processado $\hat{y}_s(n)$ (usando o conjunto $Fil[n]$ definido em (5.11')), é feita através da simples multiplicação dos respectivos elementos s_j de $S[n]$ pelos correspondentes elementos fil_j de

$$Fil[n]=\{ fil_0, fil_1, \dots, fil_{N-1} \} \quad (5.11')$$

sendo fil_j definido em (5.11) e sendo τ um valor, dependente do desvio padrão da janela que está sendo processada, que controla a inclinação da função sigmoide (vide Apêndice B: o processo realizado, usando a função de transferência Fil , é semelhante ao processo aplicado ao operador OMS). Este procedimento é feito tanto para janelas de silêncio quanto para janelas de voz.

O valor de $\tau=\tau_s$ em (5.11), para cada janela $S[n]$ que está sendo processada, é dado pela seguinte equação

$$\tau_s = \beta \cdot \tau_s \quad (5.12)$$

sendo τ_s o desvio padrão da janela que está sendo processada e β um valor baseado na potência do ruído.

O produto executado na equação (5.11), em conjunto com o valor absoluto, quando aplicado no intervalo $[-1,1]$ gera uma curva similar à apresentada na Figura 5.22. Esta curva se aproxima de 1 quando o valor do coeficiente aumenta, logo os coeficientes que são considerados redundantes corresponderão a valores pequenos na curva, enquanto que aqueles que são significativos na reconstrução do sinal terão valores mais próximos de 1. Como a redução de ruído é feita através da simples multiplicação dos respectivos elementos s_j de $S[n]$ pelos correspondentes elementos fil_j de $Fil[n]$ (Apêndice B), os elementos redundantes do sinal serão praticamente eliminados, enquanto que os coeficientes significativos serão preservados ou sofrerão uma pequena redução em suas amplitudes.

O produto usado na equação (5.11) se justifica pelo fato de que qualquer uma das funções sigmoides que são fatores desta equação geraria uma curva cujas multiplicações pelos coeficientes do sinal causariam deterioração total neste sinal. O

produto das duas funções sigmóides em conjunto com o valor absoluto causa uma transição suave na origem, como se pode observar na Figura 5.22. Se apenas uma das funções sigmóides fosse usada, sem o valor absoluto, haveria inversão de fase de elementos do sinal, como se pode observar na Figura 5.23, pois seu gráfico é simétrico em relação à origem. Se o valor absoluto for usado apenas com uma função sigmoide, a transição na origem não será suave, pois como se pode observar na Figura 5.24, a função que se obtém através do módulo da função sigmoide não é diferenciável na origem.

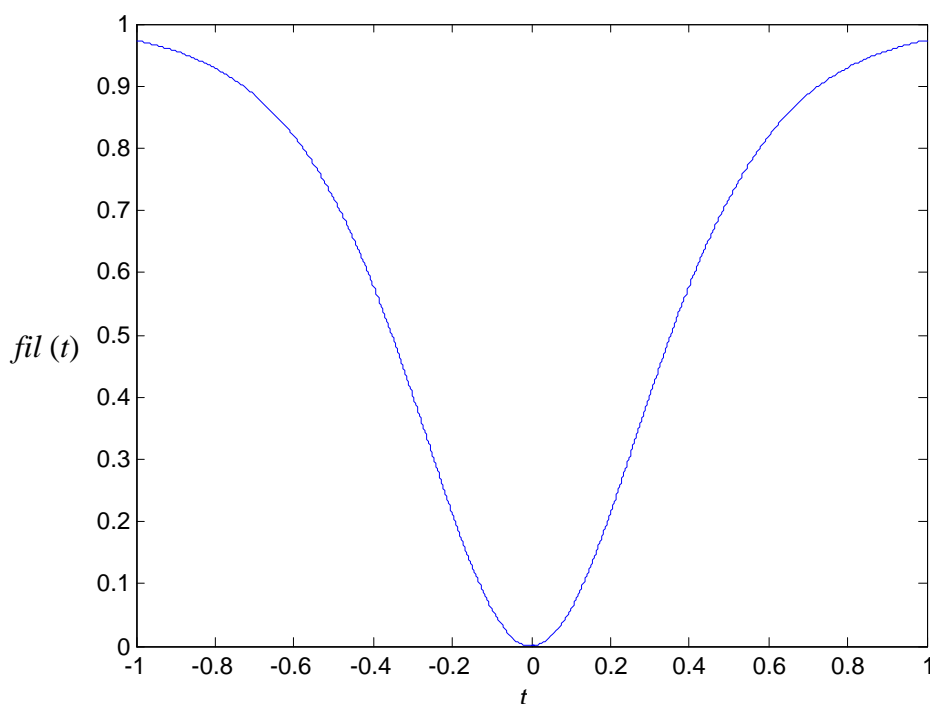


Figura 5.22 – Curva de atenuação usada no processamento, aplicando a função de transferência fil .

A curva da Figura 5.22 representa o gráfico da função dada na equação (5.11) com abscissa variando no intervalo $[-1,1]$ e $\tau=5$. Naturalmente, quando o operador definido pela equação (5.11) é aplicado a sinais de voz, a curva de atenuação de ruído não é tão suave quanto a apresentada na Figura 5.22, pois o cálculo é feito para cada coeficiente do sinal de voz processado no domínio *wavelet* e, naturalmente, um sinal de voz não tem características lineares, como o intervalo usado na Figura 5.22.

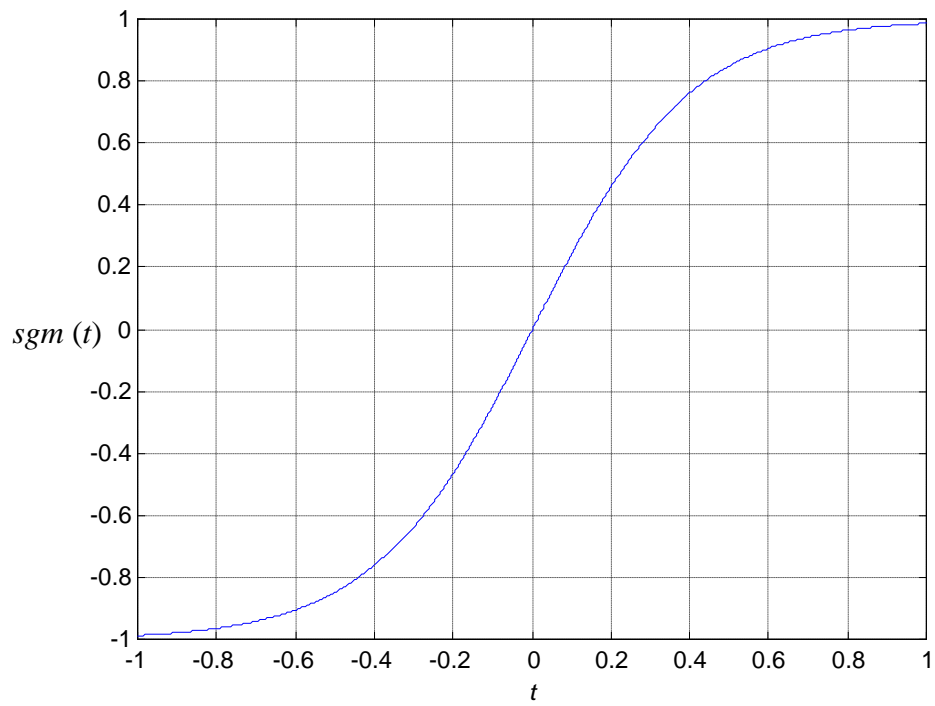


Figura 5.23 – Gráfico de uma Função Sigmoide.

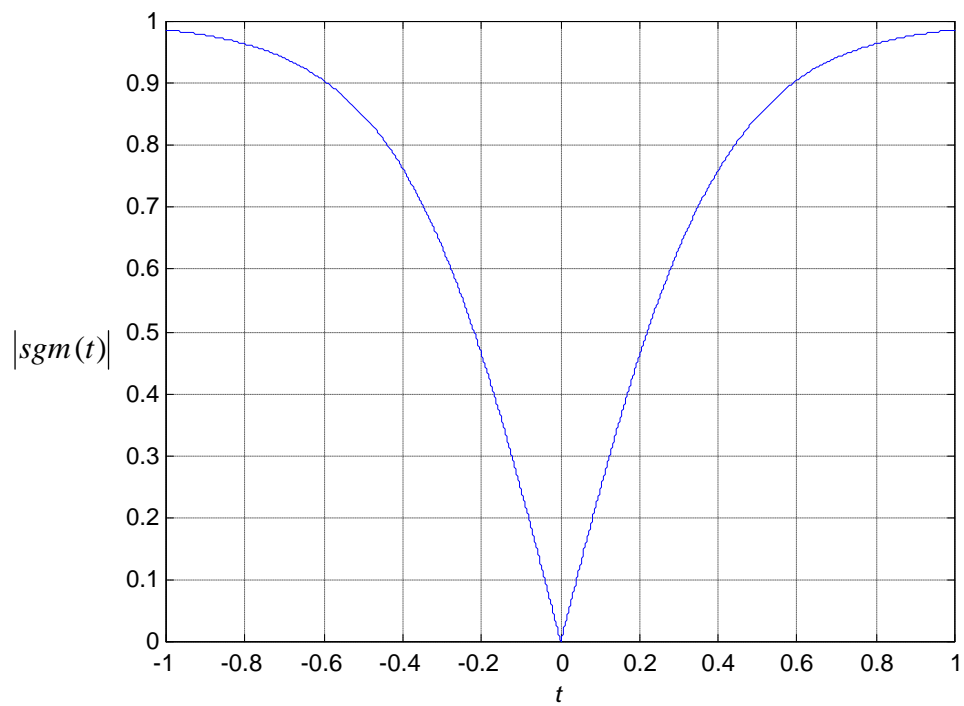


Figura 5.24 – Gráfico do valor absoluto de uma Função Sigmoide.

Na Figura 5.25 apresenta-se, em azul, uma representação do sinal de voz limpo $x(n)$ e, em vermelho, uma representação do mesmo sinal processado $\hat{y}_s(n)$, obtido usando a transformada *wavelet* inversa após a aplicação do método proposto (usando a função de transferência *fil*). Observando essa figura, é possível verificar que este método é eficiente na redução do ruído, além de efetuar uma redução uniforme ao longo de todo o sinal. A amplitude do sinal não aumenta nem diminui. E, embora haja diferenças entre os dois sinais nos trechos de voz, essas diferenças são mínimas e não representam fortes distorções para o sinal. A Figura 5.26 apresenta apenas o sinal processado $\hat{y}_s(n)$, usando o método proposto (usando a função de transferência *fil*), para que o mesmo possa ser comparado com o sinal limpo $x(n)$, apresentado na Figura 5.4.

Observando as Figuras 5.5, 5.11, 5.16 e 5.21 verifica-se que o sinal processado pelo método proposto é o que mais se assemelha ao sinal limpo.

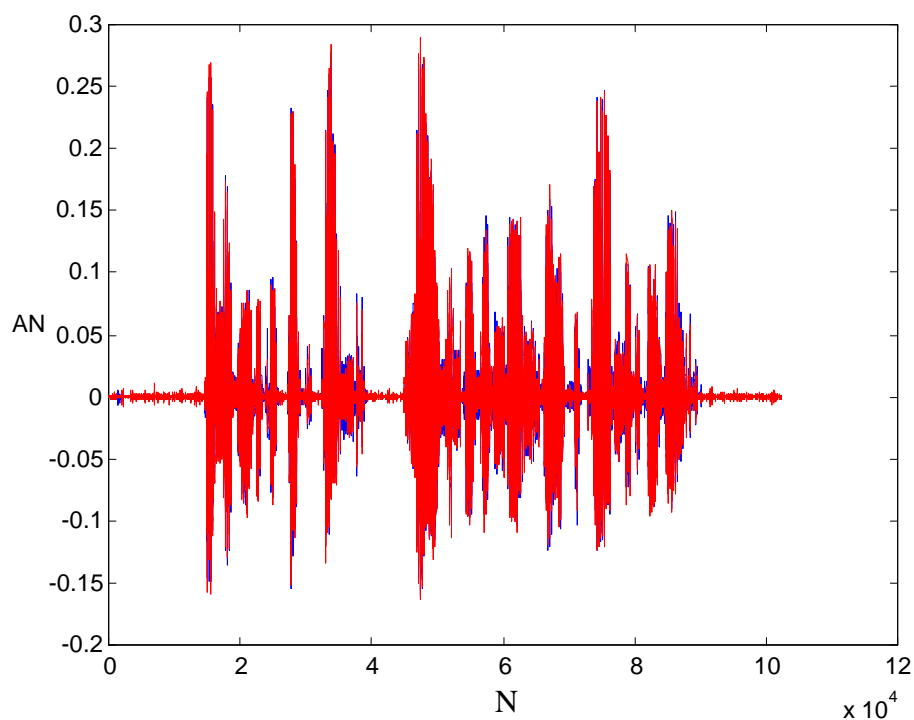


Figura 5.25 – Sinal de voz limpo $x(n)$ e sinal processado $\hat{y}_s(n)$, usando o método proposto e a transformada *wavelet* inversa.

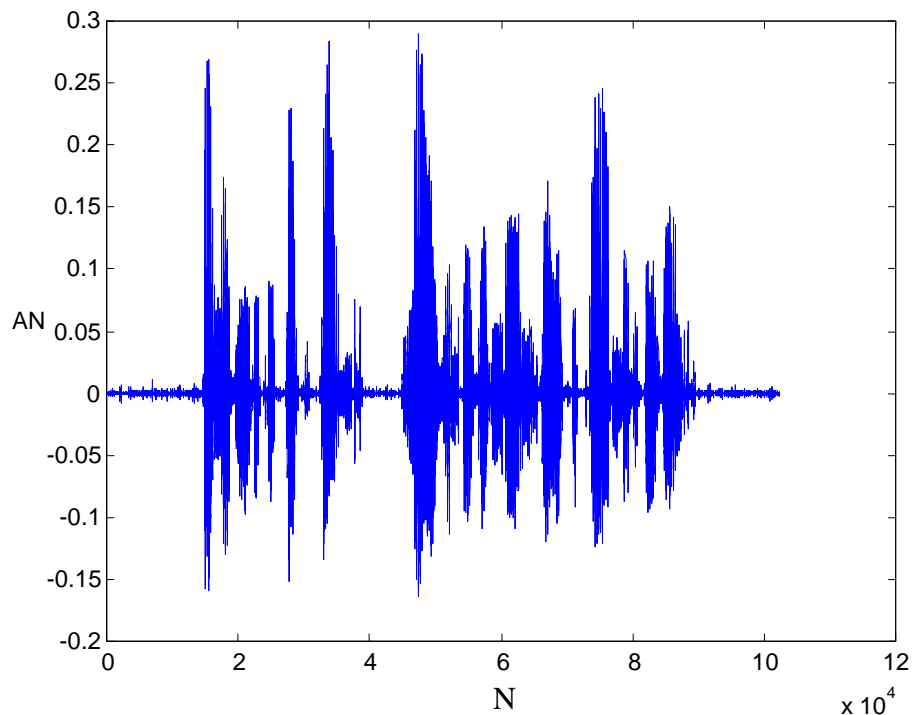


Figura 5.26 - Sinal processado $\hat{y}_s(n)$, pelo método proposto.

Na próxima seção é apresentada uma justificativa da aplicabilidade do método proposto através da análise do erro percentual entre as duas curvas que apresentam o sinal limpo e o sinal processado e, além disso, são feitas análises em trechos “vozeado” e “não vozeado” dos sinais limpo e processado no domínio *wavelet*.

5.4 Análise dos sinais limpo e processado no domínio *wavelet*

Nesta seção, é abordado o uso do método proposto nesta pesquisa, onde se faz uma análise quantitativa através do erro percentual entre os dois sinais, limpo $X[n]$ e processado $Y_x[n]$, no domínio *wavelet*. É feita também uma análise em determinadas janelas do sinal onde se expressa o erro percentual para os trechos “vozeado” e “não vozeado” dos sinais em questão. A análise é feita para sinais contaminados por ruído branco e por ruído colorido (ruído de carro). O sinal usado para a análise é o mesmo das seções anteriores deste capítulo.

O objetivo é mostrar as diferenças existentes nas formas de onda, quando se comparam os dois sinais, limpo $X[n]$ e processado $Y_x[n]$ no domínio *wavelet*, usando a metodologia proposta nesta pesquisa.

Uma maneira eficiente de se comparar dois sinais é através do erro percentual, ou seja, a diferença entre as duas formas de onda é dada em porcentagem. Para o cálculo do erro entre o sinal de voz limpo e o sinal processado, ambos com comprimento n , é usada a fórmula proposta por (SOARES et al., 2006), expressa na equação (5.13)

$$\text{Erro}(x, \hat{y}_s) = \frac{\sum_{i=0}^{N-1} |x_i^2 - \hat{y}_{s_i}^2|}{\sum_{i=0}^{N-1} x_i^2} \cdot 100\% \quad (5.13)$$

sendo $x(n) = \{x_i / i=0, \dots, N-1\}$ e $\hat{y}_s(n) = \{\hat{y}_{s_i} / i=0, \dots, N-1\}$.

Na Figura 5.27, é possível comparar os sinais limpo e processado com ruído branco. Verifica-se que as diferenças entre as duas curvas no domínio *wavelet* são muito pequenas, apresentando um erro percentual em torno de 2,83 %. Este resultado vem enfatizar ainda mais a aplicabilidade do método proposto nesta pesquisa.

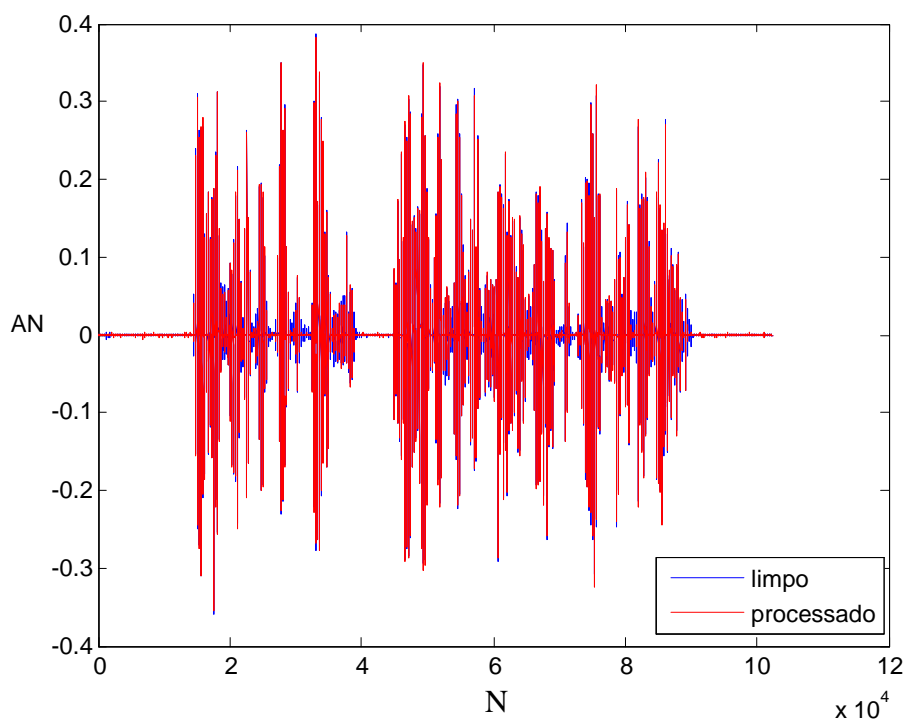


Figura 5.27 – Sinais limpo e processado no domínio *wavelet* com ruído branco – erro 2,83 %.

Nas Figuras 5.28 e 5.29, são apresentadas janelas com 512 amostras de um trecho de sinal de voz limpo no domínio *wavelet* e contaminado por ruído branco. A análise é feita em um trecho “vozeado” e num trecho “não vozeado”, respectivamente.

Para o trecho “vozeado”, o percentual de erro referente aos dois sinais (limpo e processado) é de 1,81 %, e para o trecho “não vozeado” o erro percentual é de 4,16 %.

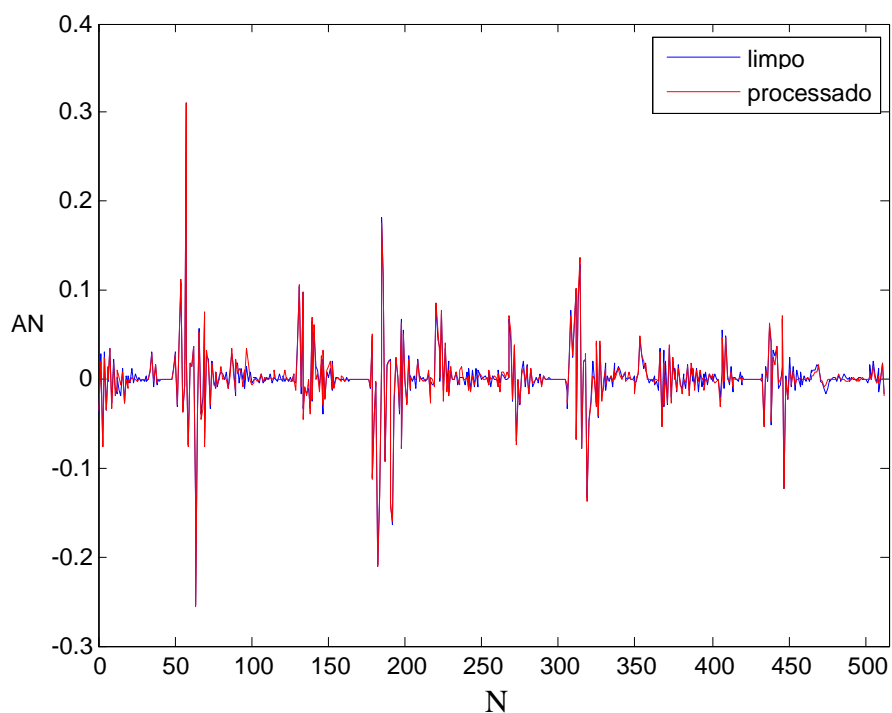


Figura 5.28 – Trecho “vozeado” do sinal processado com ruído branco no domínio *wavelet* - erro 1,81 %.

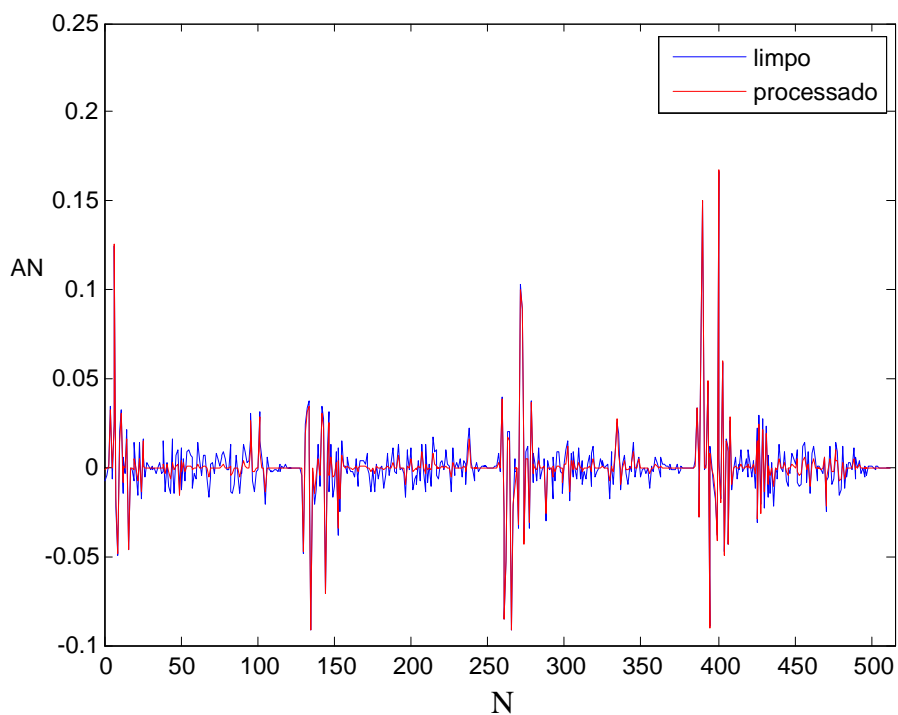


Figura 5.29 – Trecho “não vozeado” do sinal processado com ruído branco no domínio *wavelet* - erro 4,16 %.

Na Figura 5.30 é possível comparar os sinais limpo (azul) e processado (vermelho) com ruído colorido (ruído de carro). Verifica-se que as diferenças entre as duas curvas, que representam os sinais, no domínio *wavelet*, são pequenas, apresentando um erro percentual em torno de 3,51 %.

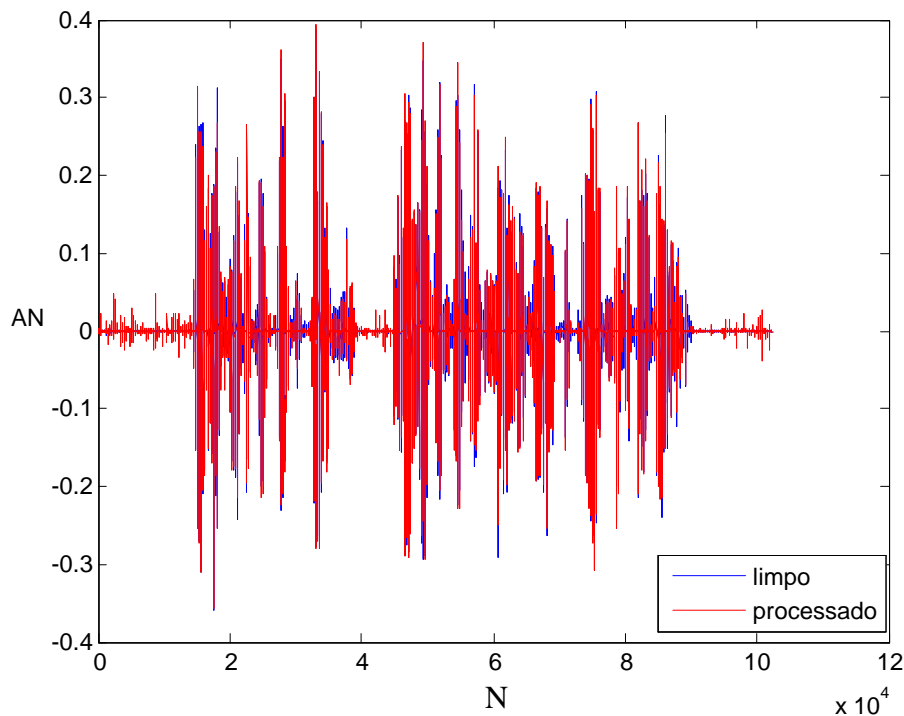


Figura 5.30 – Sinais limpo e processado com ruído colorido no domínio *wavelet* – erro 3,51 %.

Nas Figuras 5.31 e 5.32, apresenta-se uma janela com 512 amostras de um trecho de sinal de voz limpo no domínio *wavelet* e contaminado por ruído colorido (carro). Analogamente ao ruído branco, a análise é feita em um trecho “vozeado” e outro “não vozeado”, respectivamente. Para o trecho “vozeado”, o percentual de erro referente aos dois sinais (limpo e processado) é de aproximadamente de 0,4 %, e para o trecho “não vozeado” o erro percentual é de aproximadamente 0,5 %.

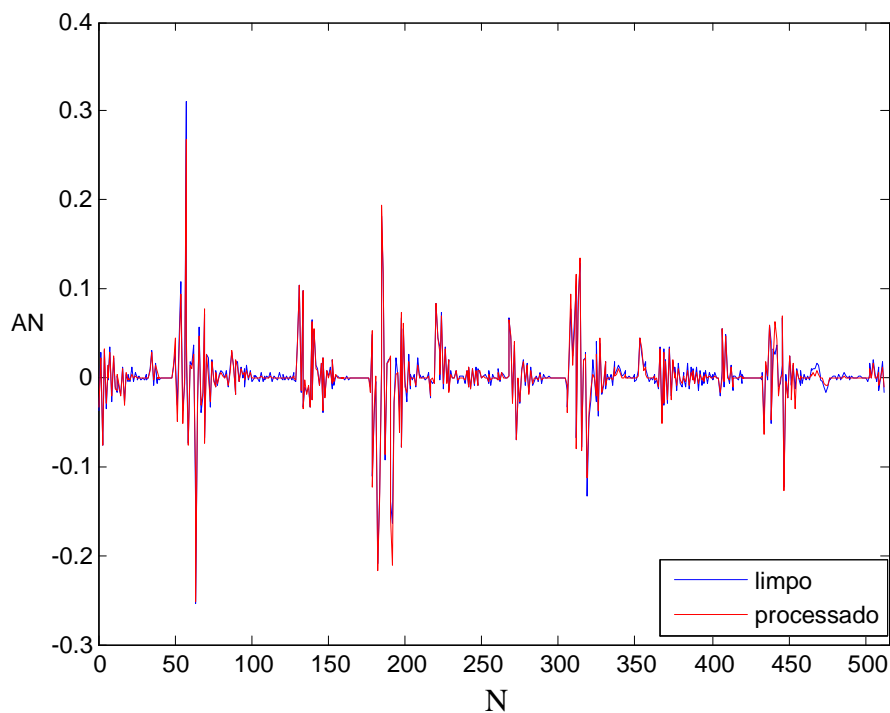


Figura 5.31 – Trecho “vozeado” do sinal processado com ruído colorido no domínio *wavelet* - erro 0,4 %.

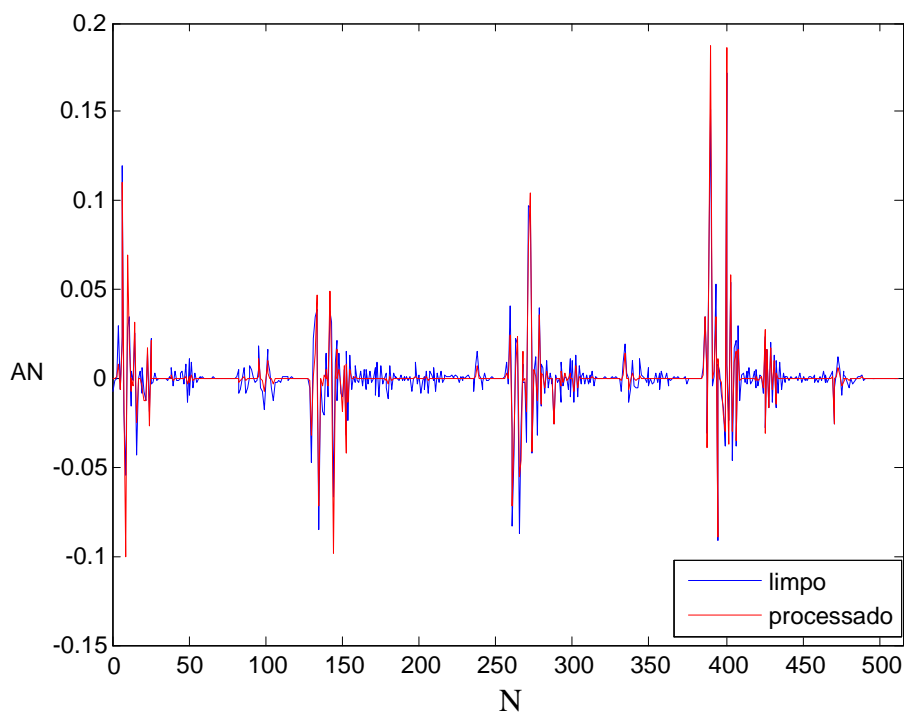


Figura 5.32 – Trecho “não vozeado” do sinal processado com ruído colorido no domínio *wavelet* - erro 0,5 %.

As análises feitas nesta seção são mais uma justificativa para aplicação do método proposto nesta pesquisa. Os resultados apresentados através do erro percentual

entre os sinais limpo e processado, assim como, as análises nos trechos “vozeado” e “não vozeado”, vem enfatizar mais ainda, o uso da metodologia proposta.

Embora haja diferenças significativas entre os dois sinais no domínio *wavelet*, essas diferenças são devidas a eliminação dos coeficientes redundantes, característicos da transformada *wavelet*. A eliminação desses coeficientes não compromete a reconstrução do sinal e nem sua inteligibilidade.

A combinação polinomial dos três operadores neste trabalho foi obtida por exaustivos testes, chegando a discussões e resultados satisfatórios. O que diferencia este método em relação aos demais métodos de redução de ruído no domínio *wavelet*, é a aplicação de uma função de transferência que não utiliza limiar para reduzir o ruído.

O capítulo seguinte é dedicado a análise e discussão dos resultados do método proposto aplicando a vários sinais de voz.

CAPÍTULO 6
ANÁLISE E DISCUSSÃO DOS RESULTADOS
Introdução

Neste capítulo são apresentados os resultados das implementações do método proposto e as avaliações dos sinais processados. Para verificar a qualidade dos sinais processados, foram avaliados os níveis de redução de ruído e de distorção usando como medidas a SNR e a PESQ.

A SNR é calculada de acordo com a equação (6.1), seu cálculo é feito tomando a razão entre um segmento de voz e um segmento de silêncio de cada sinal, conforme Deller; Proakis e Hansen (1993):

$$SNR = 10 \log_{10} \left(\frac{\sum_{i=0}^{N-1} x_i^2}{\sum_{i=0}^{N-1} r_i^2} \right) \quad (6.1)$$

sendo N o número de amostras do segmento escolhido, $x(n) = \{x_i / i=0, \dots, N-1\}$ as amostras no trecho de voz e $r(n) = \{r_i / i=0, \dots, N-1\}$ as amostras no trecho de silêncio. Neste trabalho, baseando-se nos sinais utilizados, adotou-se $N=5000$. O valor de SNR apresentado nas Tabelas 6.1, 6.2 e 6.3 é uma média dos valores de SNR obtidos pela

aplicação da equação (6.1) em diferentes segmentos do sinal de voz, neste caso foram usados três segmentos diferentes.

Quanto à SNR do sinal final devem-se levar em conta as seguintes observações. Se, após o processamento, a SNR do sinal for muito baixa em relação à SNR do sinal limpo, significa que houve pouca redução de ruído. Se, após o processamento a SNR do sinal processado for muito maior que a SNR do sinal limpo, significa que houve grande redução de ruído e, possivelmente, o sinal deve estar bastante distorcido. Por isso, após o processamento, sinais processados que têm SNR próxima da SNR do sinal original são mais atrativos, pois indicam que a redução de ruído não causou distorções nos trechos de voz do sinal.

6.1 Sinais de voz e Considerações

Para a verificação da eficiência dos métodos e a apresentação dos resultados foram usados quatro sinais de voz, aqui referenciados como A e C, gravados por voz masculina, sendo o primeiro em português e o segundo em inglês, e B e D, gravados por voz feminina, sendo o sinal B em português e o sinal D em inglês. O sinal de voz A foi obtido através do Centro de Pesquisa e Desenvolvimento em Telecomunicações (CPqD), o sinal de voz B foi gravado no laboratório de computação do programa de Pós-Graduação em Engenharia Elétrica da Universidade Estadual Paulista (UNESP – Campus de Ilha Solteira) e os sinais de voz C e D, são sinais recomendados pelo IEEE, obtidos através do site <http://www.utdallas.edu/~loizou/speech/noizeus/>. As frases dos sinais são as seguintes:

Sinal A – “FOI DETECTADO UM PROBLEMA EM SEU CARTÃO, ELE DEVE SER SUBSTITUÍDO”

Sinal B – “DIRIJA COM CUIDADO SEMPRE VERIFICANDO A SINALIZAÇÃO DO TRÂNSITO”

Sinal C – “WE FIND JOY IN THE SIMPLEST THINGS”

Sinal D – “SHE HAS A SMART WAY OF WEARING CLOTHES”

Os sinais **A** e **B** foram amostrados a 16 kHz e quantizados com 16 bits por amostra, enquanto que os sinais **C** e **D** foram amostrados a 8 kHz e também quantizados a 16 bits por amostra. Nos testes, foram usados arquivos no formato *wav*, que

apresentam a amplitude normalizada no intervalo (-1,1). Os segmentos de voz foram obtidos usando-se uma janela de *Hann* com sobreposição de 50 % entre os segmentos. O comprimento de cada janela é de 256 amostras o que, no domínio *wavelet*, permite a obtenção de até 8 níveis de resolução na decomposição do sinal. Para a implementação computacional foi usado o software MATLAB e a função *wavelet* usada na DWT foi a *wavelet* de *Daubechies* número 10 (Daub10).

6.1.1 Avaliações Objetivas dos Sinais Processados pelo Método Proposto

Para os testes do algoritmo, relativo à aplicação do método proposto, os sinais considerados foram contaminados por dois tipos de ruído: ruído Gaussiano branco e ruído colorido. No caso do ruído colorido, foi usado o ruído de carro, gravado numa rodovia asfaltada, a 100 km/h. Os sinais processados foram analisados com dois níveis de SNR, 5 dB e 10 dB.

Para ilustrar a aplicação da função de transferência fil , no método proposto na seção 5.3, os sinais de voz, no domínio do tempo, $\mathbf{K} = \mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$ acima considerados, serão indicados por $x_K = x_k$ (8) (observar que se tem $N=2^n$, com $N = 256$ e $n = 8$). O ruído Gaussiano branco será indicado por $r_b = r_b$ (8) e o ruído colorido por $r_0 = r_0$ (8). Agora, os sinais de voz limpo $x(n)$ contaminado pelos ruídos considerados, no domínio do tempo, são denotados por ($s_{K,l} = s_{K,l}$ (8))

$$s_{K,l} = x_K + r_l, \text{ para } K = A, B, C, D \text{ e para } l = b, 0.$$

Os sinais contaminados pelos ruídos citados, no domínio transformado, são indicados por $S_{K,l} = S_{K,l}$ [8], para $K = A, B, C, D$ e para $l = b, 0$. Assim, $S_{K,l}$ será um conjunto da seguinte forma

$$S_{K,l} = \{s_{K,l,0}, s_{K,l,1}, \dots, s_{K,l,255}\} = \{s_{K,l,j} / j = 0, \dots, 255\}$$

Para cada $K = A, B, C, D$ e cada $l = b, 0$, tem-se a função de transferência $Fil_{K,l} = Fil_{K,l}$ [8] tendo como elementos

$$Fil_{K,l} = \{fil_{K,l,0}, fil_{K,l,1}, \dots, fil_{K,l,255}\} = \{fil_{K,l,j} / j = 0, \dots, 255\}$$

sendo que a cada $fil_{K,l,j}$ é definido por (vide equação (5.11))

$$fil_{K,l,j} = \left| \frac{1 - e^{-\alpha f_{K,l,j}}}{1 + e^{-\alpha f_{K,l,j}}} \cdot \frac{1 - e^{\alpha f_{K,l,j}}}{1 + e^{\alpha f_{K,l,j}}} \right|$$

com $\alpha=0,01$, e cada função $f_{K,l,j}$ é definida pela expressão (vide equação (5.10))

$$f_{K,l,j} = \left[\frac{|s_{K,l,j}| + |s_{K,l,j+1}|}{2} \right]^3 + \left[\frac{|s_{K,l,j}|}{\alpha + |s_{K,l,j}|} \right]^2 + \frac{|s_{K,l,j}|}{\alpha + |sr_{K,l,j}|}$$

Para cada $K = A, B, C, D$, e cada $l=b,0$, multiplicando os elementos de $S_{K,l}$ pelos elementos de $Fil_{K,l}$, obtém-se o sinal $Y_{K,l}=Y_{K,l}$ [8] ainda no domínio transformado (vide (5.5)). A partir desse sinal, usando a transformada *wavelet* inversa, obtém-se o sinal processado $\hat{y}_{K,l}=\hat{y}_{K,l}$ (8), considerado no domínio do tempo (vide também os apêndices D e E). Os gráficos das representações, de cada sinal processado $\hat{y}_{K,l}$, estão apresentados nas Figuras 6.5 – 6.20.

Para verificar o nível de redução de ruído nos sinais processados pelo método proposto, a SNR foi calculada para o sinal limpo ($x_{K,l}, K = A, B, C, D$), antes da contaminação por ruído, para o sinal ruidoso, e no caso foi 5 dB ou 10 dB para o sinal processado ($\hat{y}_{K,l}, K = A, B, C, D$, e $l=b,0$).

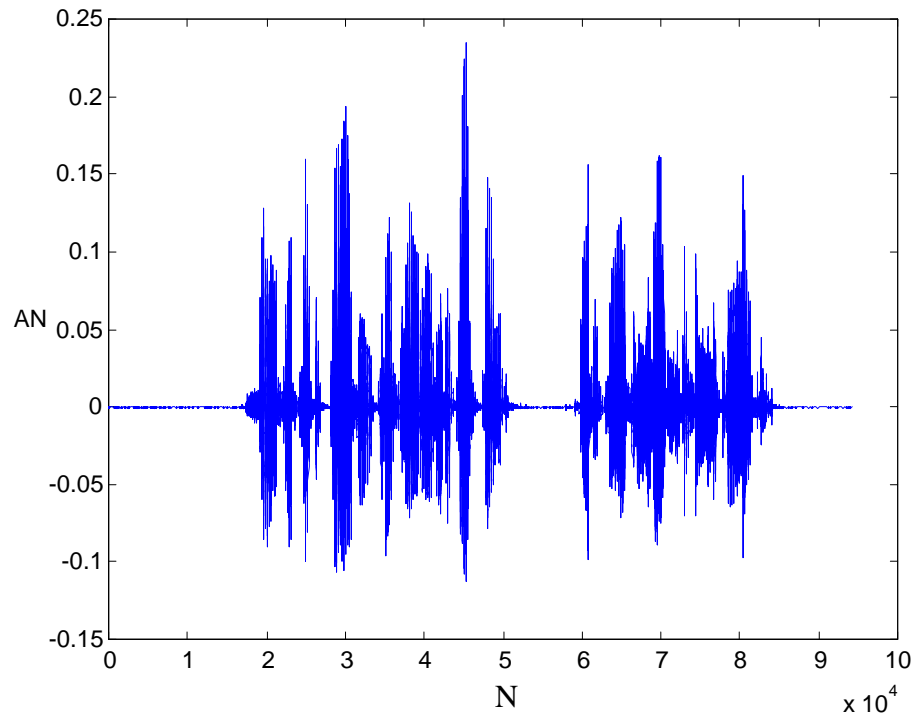
Além da SNR foi usada mais uma medida para avaliar os sinais processados. A outra medida usada é a PESQ, que faz uma avaliação objetiva a partir dos sinais original $x_{K,l}$ (8) e processado $\hat{y}_{K,l}$ (8), tendo como base um modelo psicoacústico da audição humana (RIX et al., 2000).

Na Tabela 6.1 são apresentadas as SNRs, em dB, dos sinais limpos x_K (8).

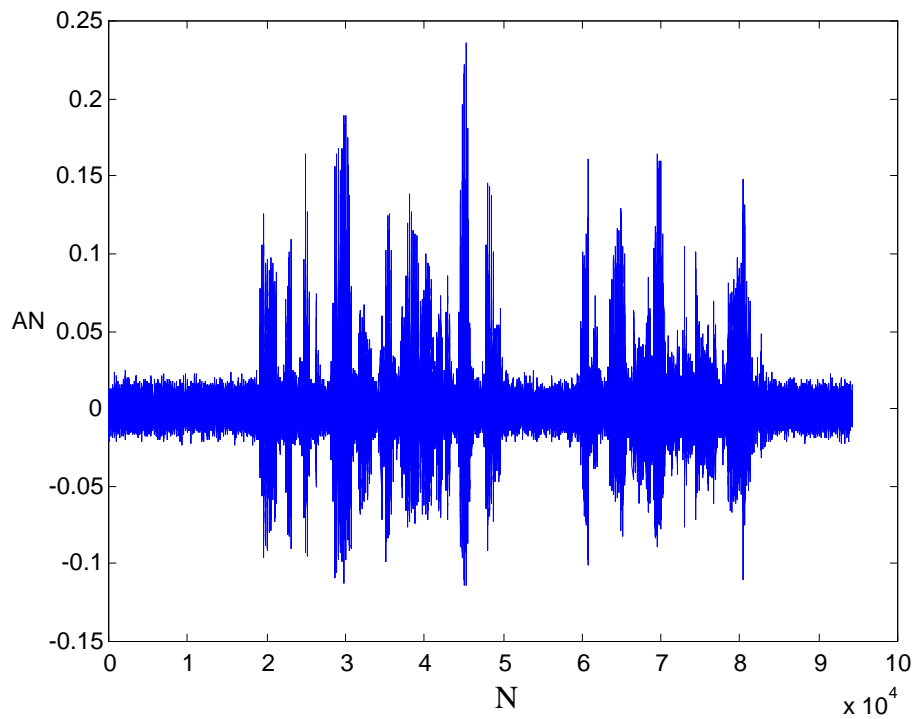
Nas Figuras 6.1, 6.2, 6.3 e 6.4 são apresentadas, respectivamente, as formas de onda dos sinais A, B, C e D, e os mesmos sinais contaminados por ruído branco $s_{K,b}$ (8) e por ruído colorido $s_{K,0}$ (8), sendo $N = 256$ o número de amostras e AN a amplitude numérica do sinal.

Tabela 6.1 – SNR dos sinais limpos x_K (8).

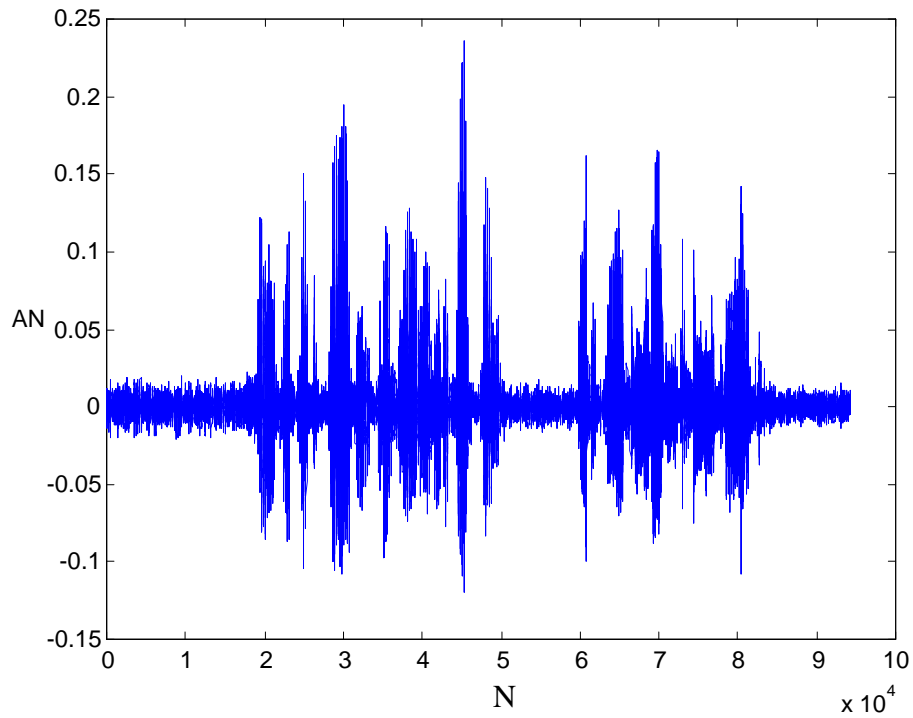
	Sinal limpo x_K (8)
Sinal A	42,192
Sinal B	49,922
Sinal C	45,922
Sinal D	41,754



(a)

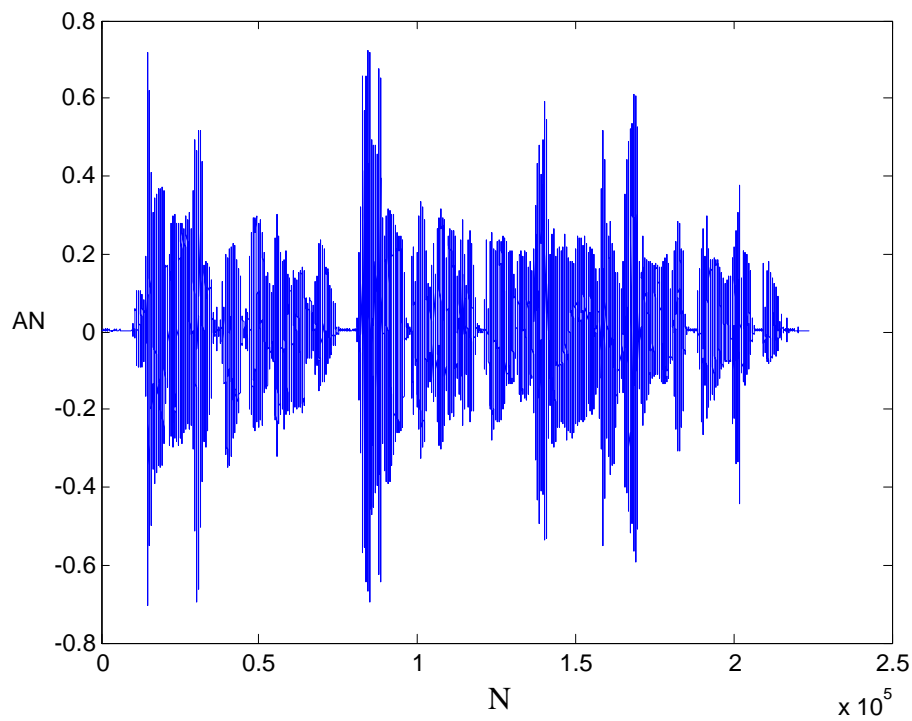


(b)

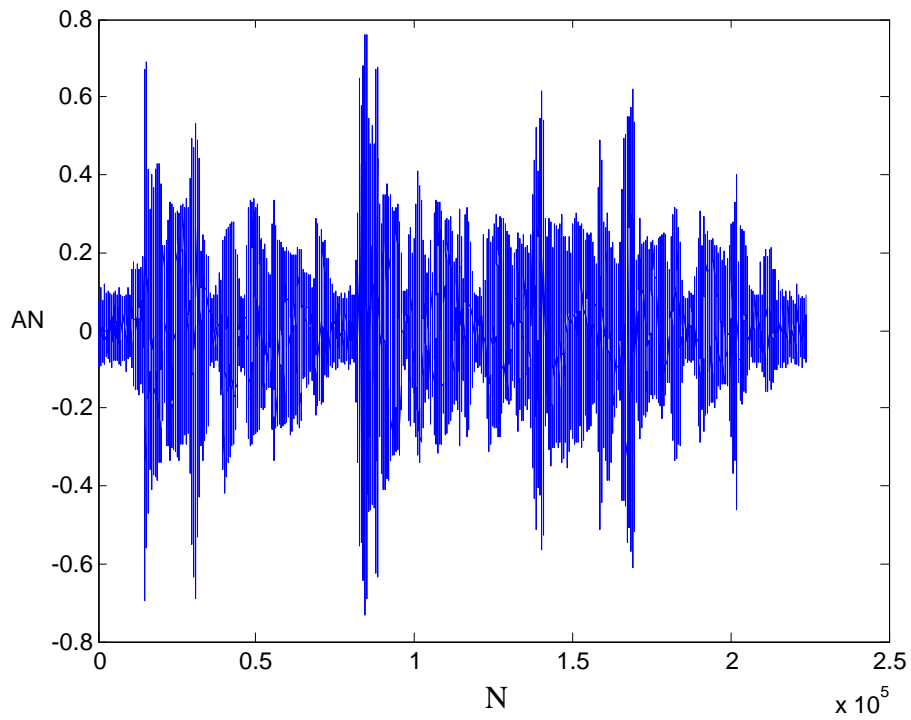


(c)

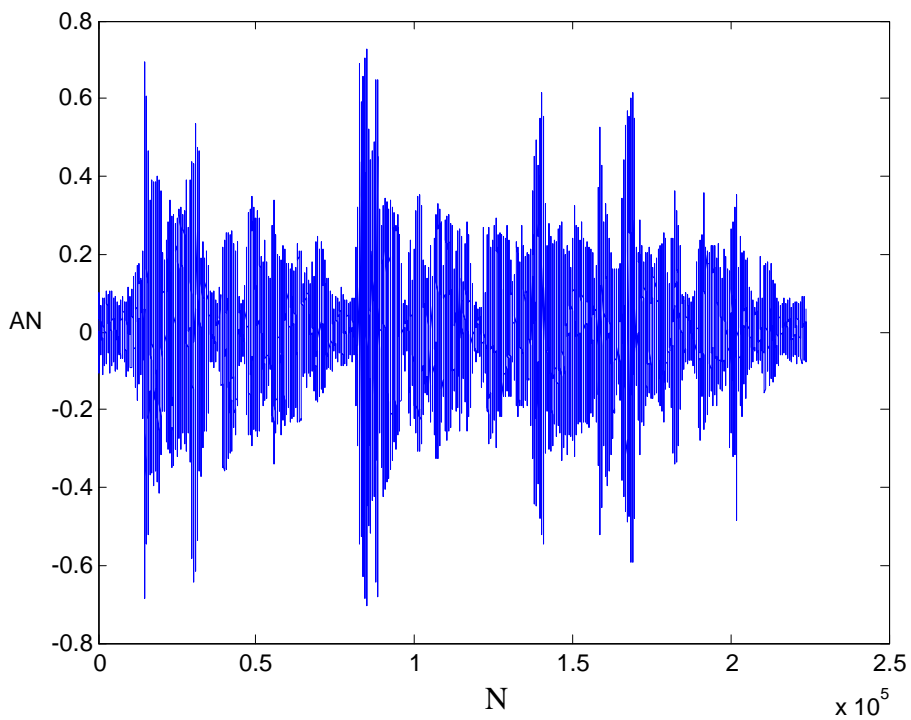
Figura 6.1 – (a) forma de onda do sinal A $x_A(8)$; (b) sinal A com ruído branco $s_{A,b}(8)$; (c) sinal A com ruído colorido $s_{A,0}(8)$.



(a)

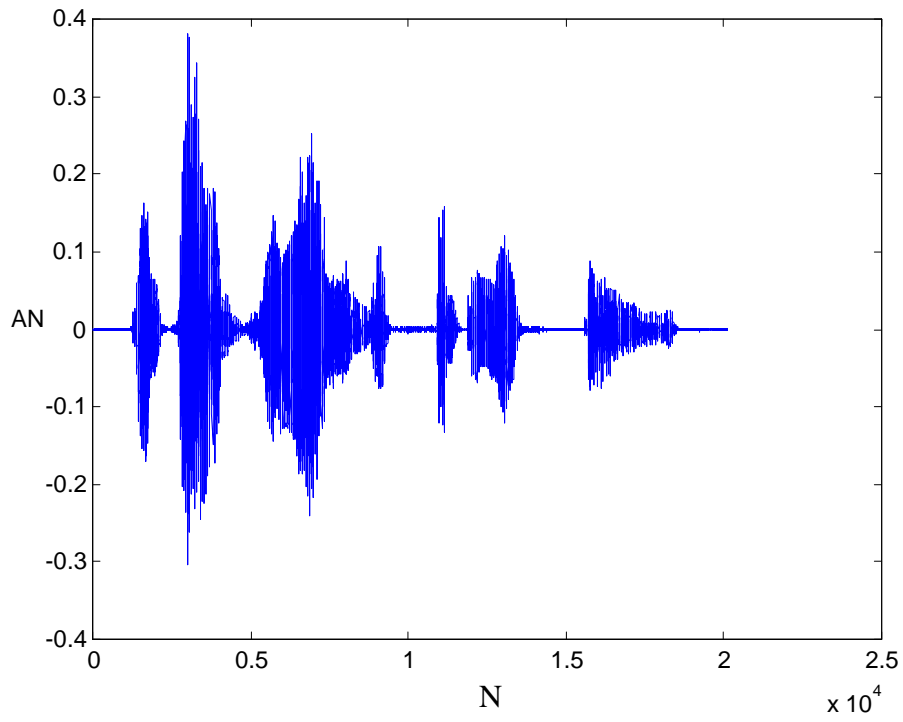


(b)

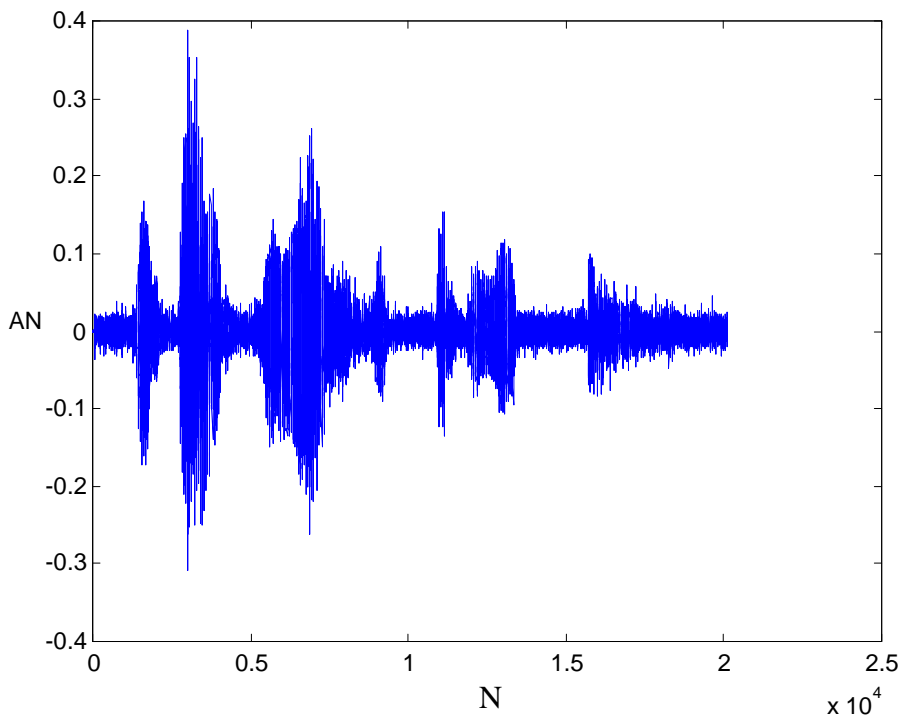


(c)

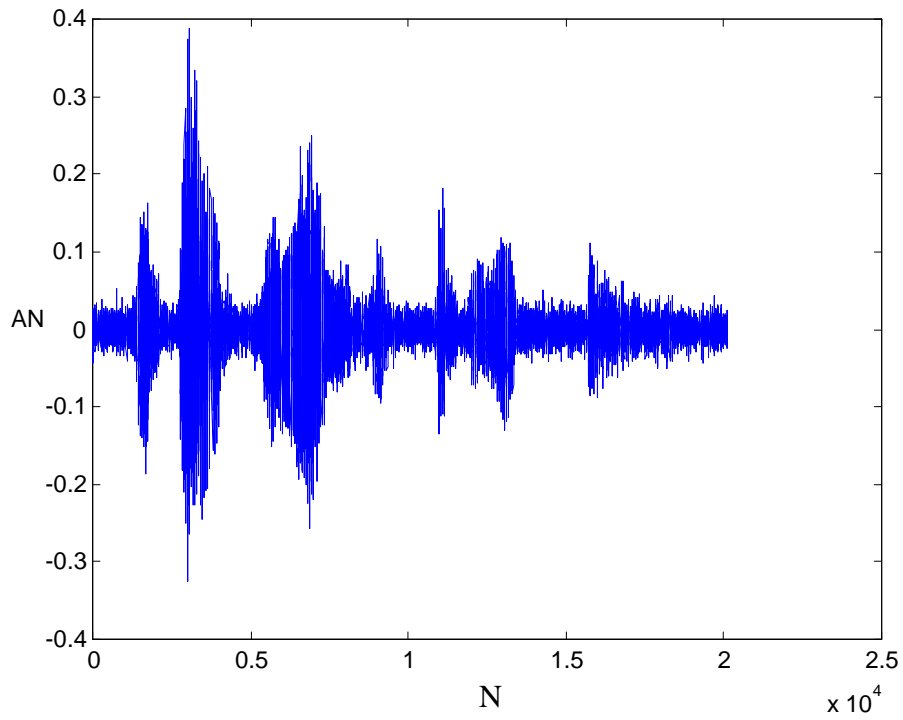
Figura 6.2 – (a) forma de onda do sinal B $x_B(8)$; (b) sinal B com ruído branco $s_{B,b}(8)$; (c) sinal B com ruído colorido $s_{B,0}(8)$.



(a)

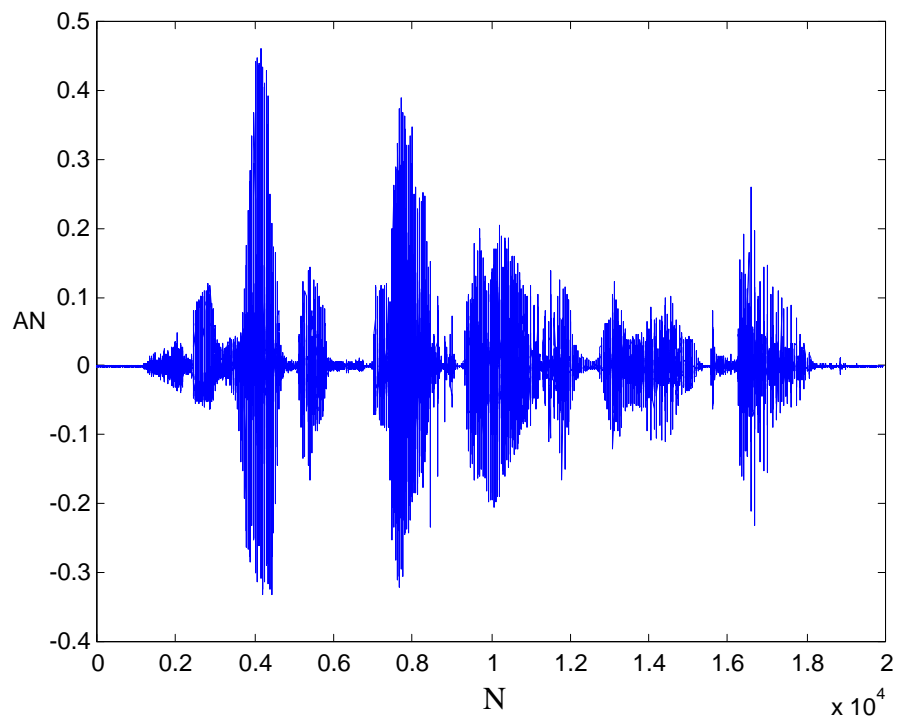


(b)

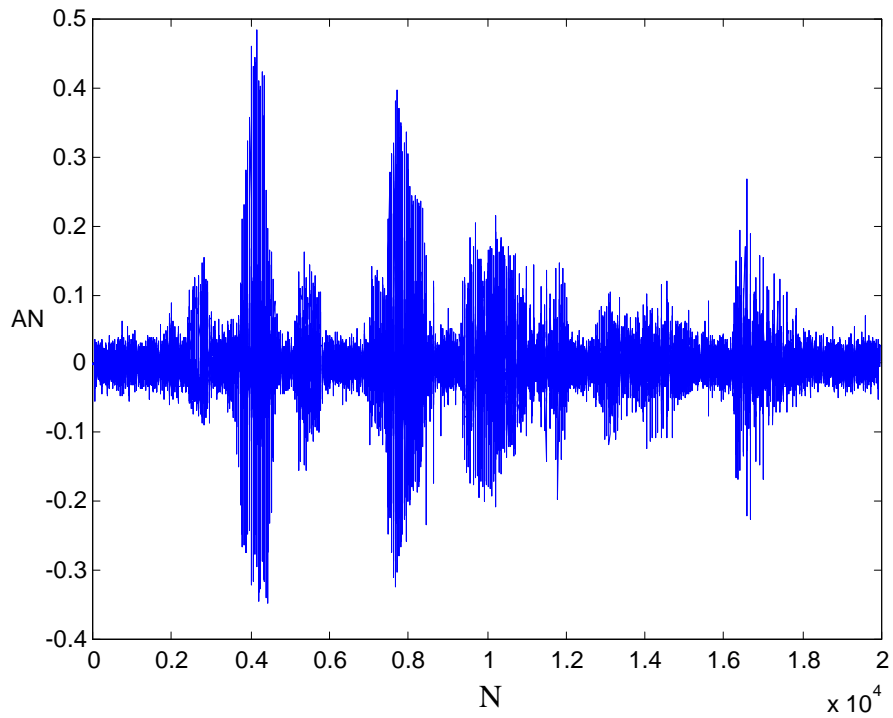


(c)

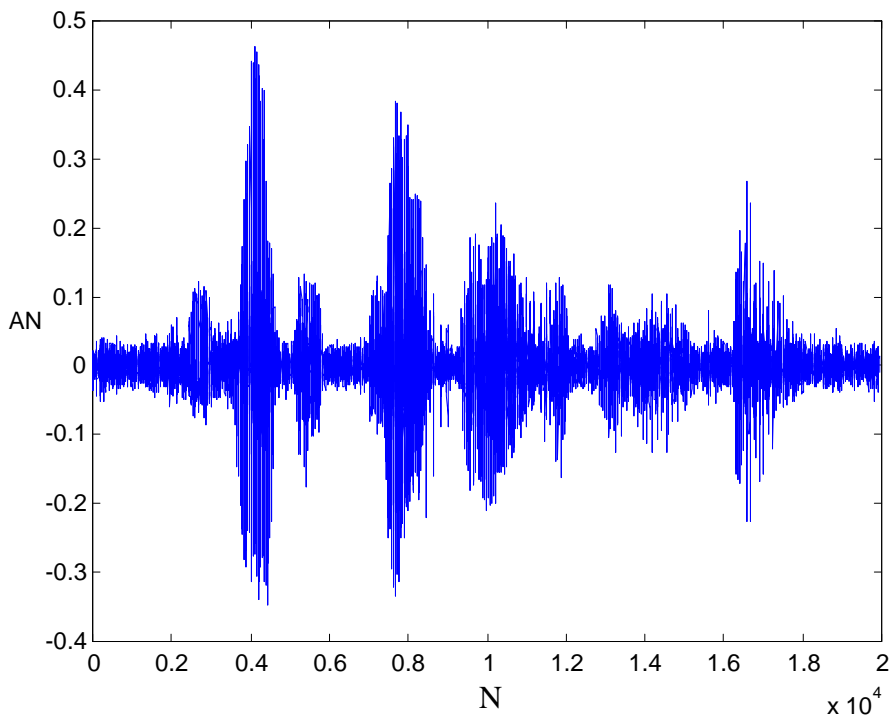
Figura 6.3 – (a) forma de onda do sinal C $x_C(8)$; (b) sinal C com ruído branco $s_{C,b}(8)$; (c) sinal C com ruído colorido $s_{C,0}(8)$.



(a)



(b)



(c)

Figura 6.4 – (a) forma de onda do sinal D $x_D(8)$; (b) sinal D com ruído branco $s_{D,b}(8)$; (c) sinal D com ruído colorido $s_{D,0}(8)$.

Nas Tabelas 6.2 e 6.3 são apresentadas as SNRs, em dB, dos sinais processados pelos ruídos mencionados anteriormente, utilizando o método proposto, com SNRs de 5 dB e 10 dB, respectivamente.

Tabela 6.2 – SNR dos sinais processados (5 dB).

	Sinal com ruído branco $\hat{y}_{K,b}$ (8)	Sinal com ruído colorido $\hat{y}_{K,0}$ (8)
Sinal A	19,418	11,121
Sinal B	18,332	11,378
Sinal C	19,851	10,880
Sinal D	15,765	23,209

Tabela 6.3 – SNR dos sinais processados (10 dB).

	Sinal com ruído branco $\hat{y}_{K,b}$ (8)	Sinal com ruído colorido $\hat{y}_{K,0}$ (8)
Sinal A	39,294	23,101
Sinal B	41,853	20,242
Sinal C	46,635	33,538
Sinal D	39,382	34,151

Na Tabela 6.5 são apresentados os resultados da PESQ para os sinais processados com ruído branco $\hat{y}_{K,b}$ (8) e para os sinais processados com ruído colorido $\hat{y}_{K,0}$ (8) com as SNRs de 5 dB e 10 dB, respectivamente. Antes, porém, para possibilitar uma melhor avaliação da metodologia proposta, os resultados da PESQ calculados para os sinais ruidosos são apresentados na Tabela 6.4. A maior nota que a PESQ atribui a um sinal é 4,5 e sinais que recebem nota a partir de 3 são considerados de boa qualidade.

Tabela 6.4 – PESQ dos sinais ruidosos $s_{K,l}$

	Ruído Branco $s_{K,b}$ (8)		Ruído Colorido $s_{K,0}$ (8)	
	5 dB	10 dB	5 dB	10 dB
Sinal A	2,187	2,653	2,361	2,536
Sinal B	2,076	2,355	2,030	2,407
Sinal C	1,805	2,273	1,950	2,220
Sinal D	1,617	2,021	1,848	2,310

Tabela 6.5 – PESQ dos sinais processados $\hat{y}_{k,l}$

	Ruído Branco $\hat{y}_{k,b}$ (8)		Ruído Colorido $\hat{y}_{k,c}$ (8)	
	5 dB	10 dB	5 dB	10 dB
Sinal A	3,183	3,656	3,280	3,423
Sinal B	3,048	3,333	2,957	3,137
Sinal C	2,842	3,315	2,715	3,054
Sinal D	2,448	3,065	2,708	2,992

Nas Figuras de 6.5 a 6.20 são apresentadas as formas de onda de todos os sinais processados, cujos valores de SNR e PESQ foram apresentados nas tabelas anteriores. Observando as formas de onda dos sinais processados, Figuras 6.5 a 6.20, pode-se notar que o método proposto é eficiente na redução de ruído, tanto branco quanto colorido, porém essa redução é mais eficaz para os sinais com SNR 10 dB e com ruído branco. Para os sinais com SNR 5 dB a redução é menor, principalmente para os sinais femininos contaminados com ruído colorido. Porém, observando todas as figuras, percebe-se que o método provoca uma redução de ruído uniforme ao longo do sinal e, comparando os sinais processados com os respectivos sinais limpos, não há introdução de distorções nos trechos de voz.

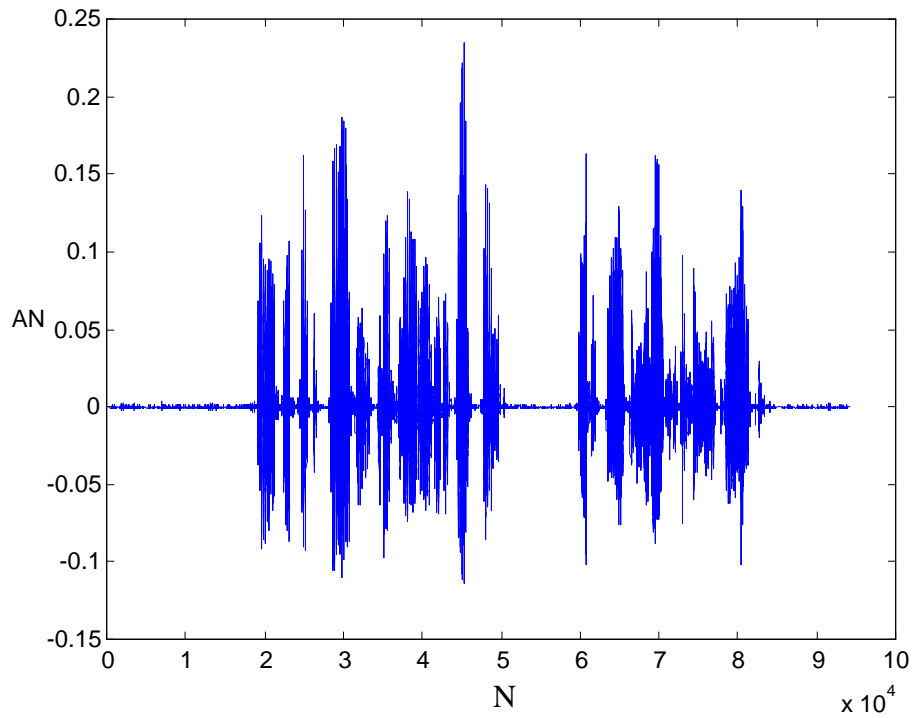


Figura 6.5 – Sinal processado $\hat{y}_{A,b}(8)$ (10 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_A(8)$ contaminado com ruído branco $r_b(8)$.

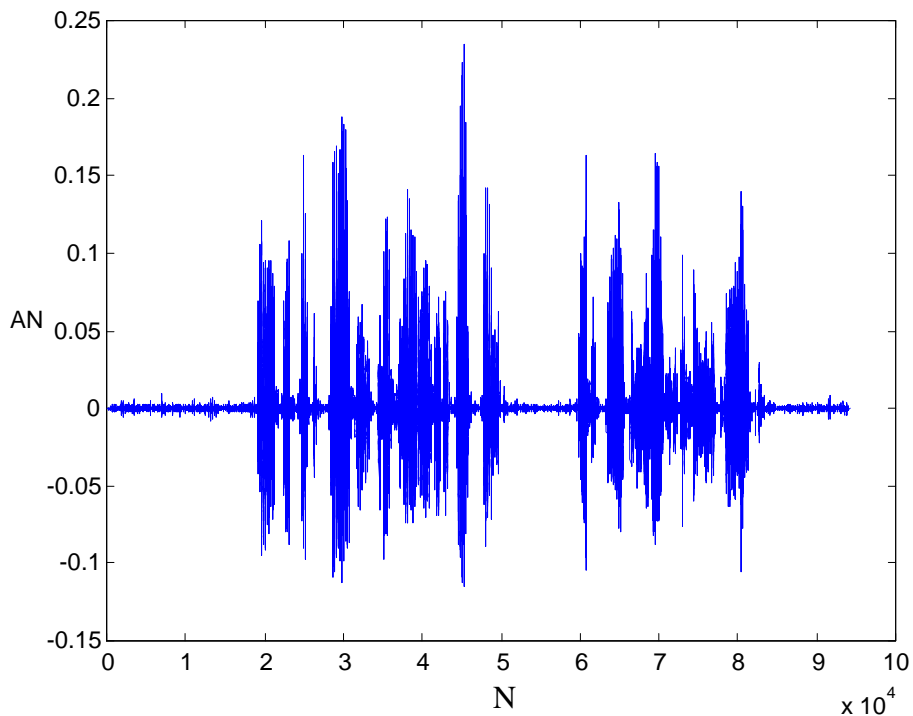


Figura 6.6 – Sinal processado $\hat{y}_{A,b}(8)$ (5 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_A(8)$ contaminado com ruído branco $r_b(8)$.

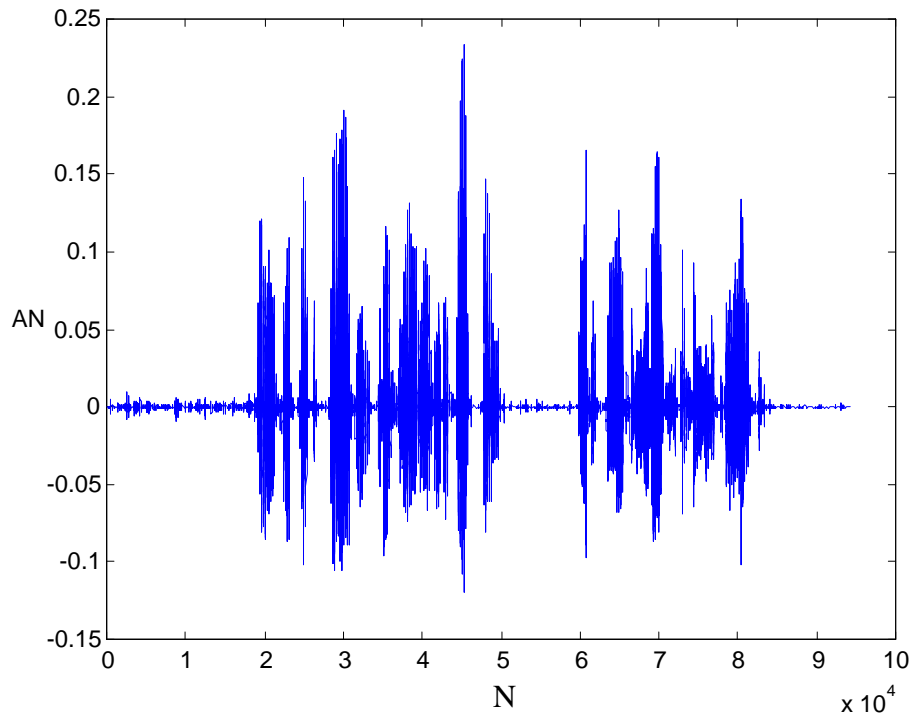


Figura 6.7 – Sinal processado $\hat{y}_{A,0}(8)$ (10 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_A(8)$ contaminado com ruído colorido $r_0(8)$.

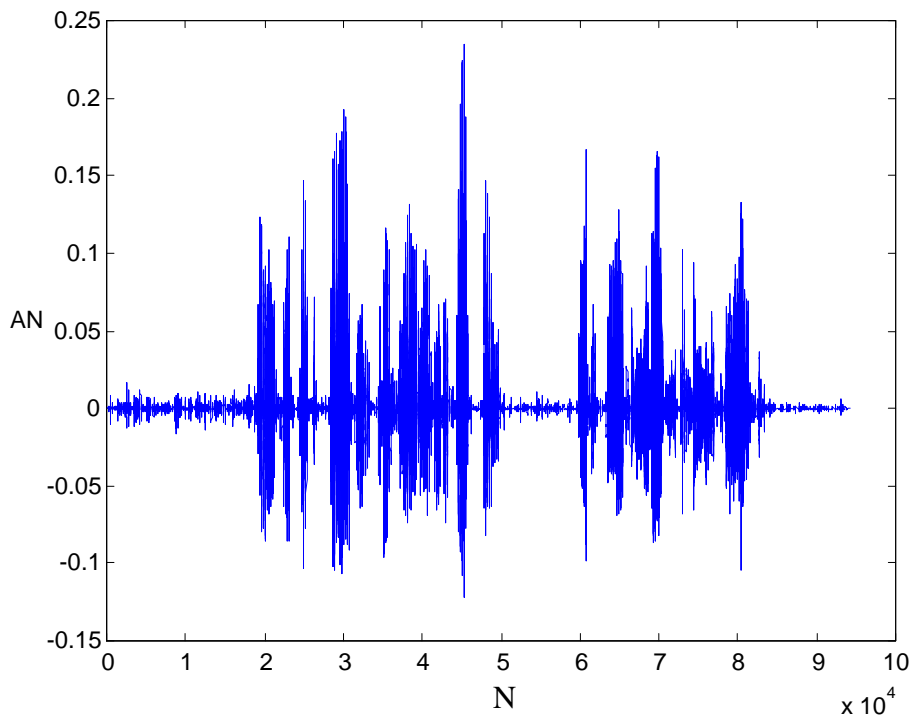


Figura 6.8 – Sinal processado $\hat{y}_{A,0}(8)$ (5 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_A(8)$ contaminado com ruído colorido $r_0(8)$.

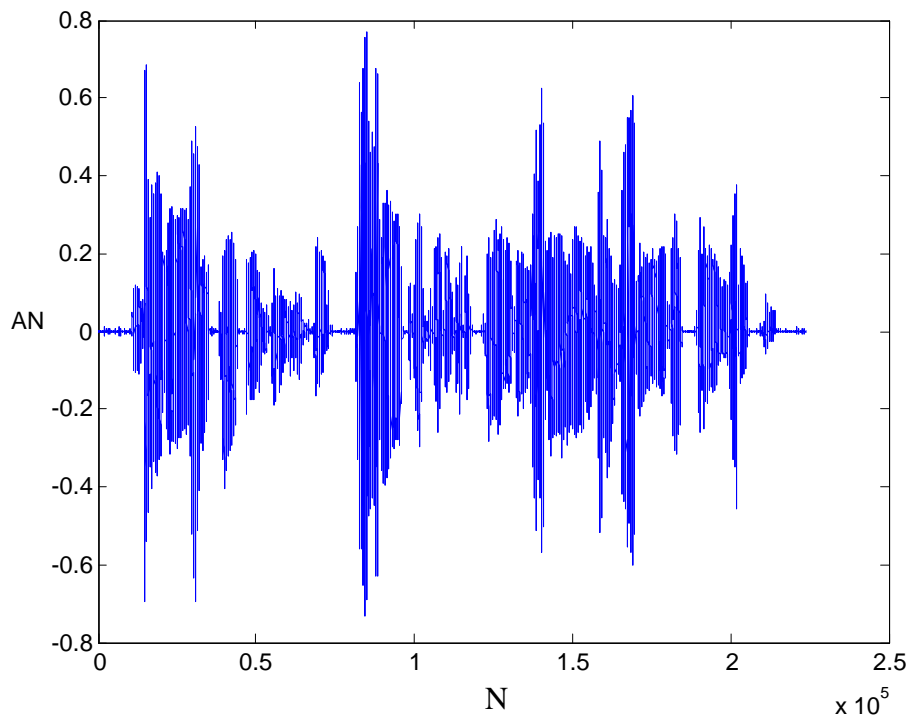


Figura 6.9 – Sinal processado $\hat{y}_{B,b}(8)$ (10 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_B(8)$ contaminado com ruído branco $r_b(8)$.

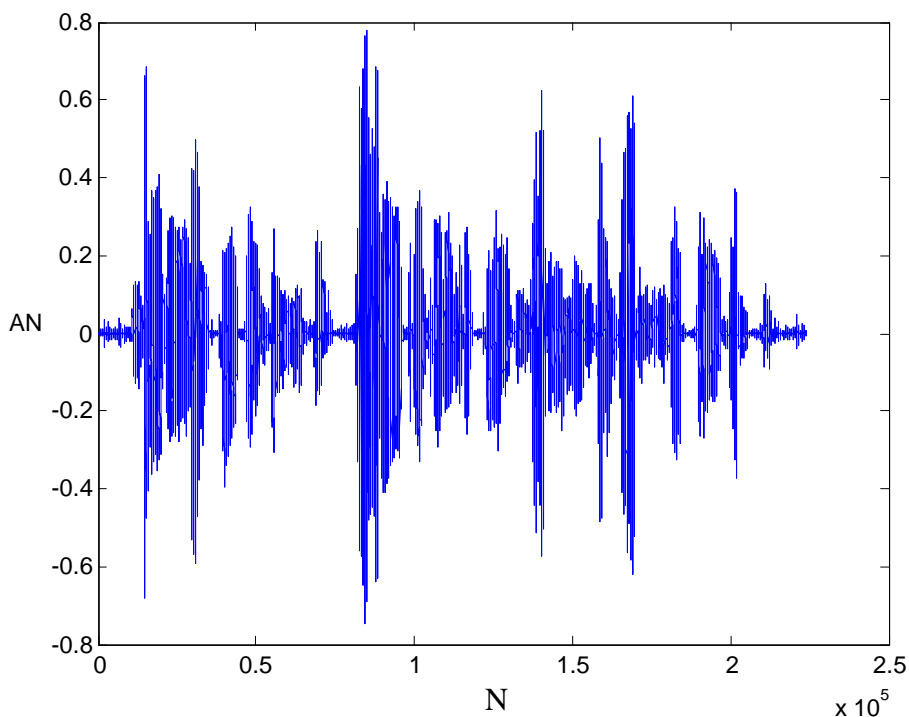


Figura 6.10 – Sinal processado $\hat{y}_{B,b}(8)$ (5 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_B(8)$ contaminado com ruído branco $r_b(8)$.

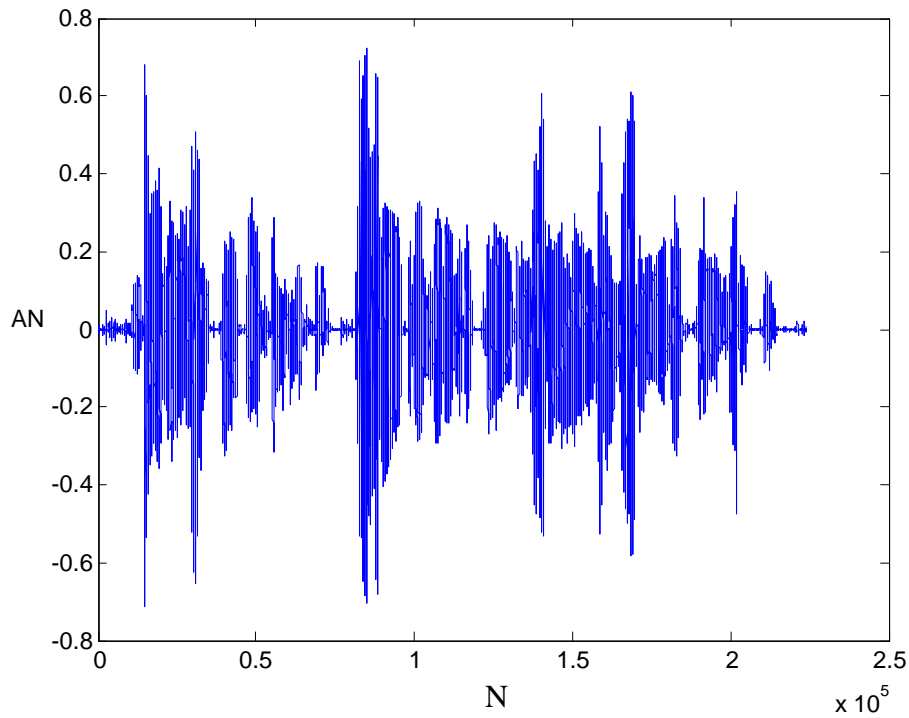


Figura 6.11 – Sinal processado $\hat{y}_{B,0}(8)$ (10 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_B(8)$ contaminado com ruído colorido $r_0(8)$.

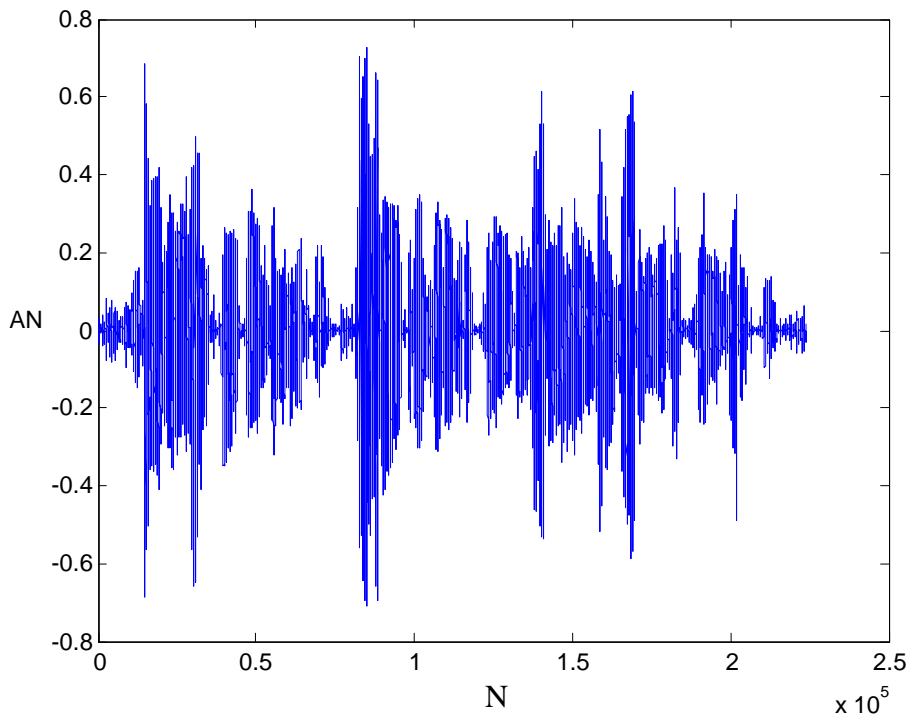


Figura 6.12 – Sinal processado $\hat{y}_{B,0}(8)$ (5 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_B(8)$ contaminado com ruído colorido $r_0(8)$.

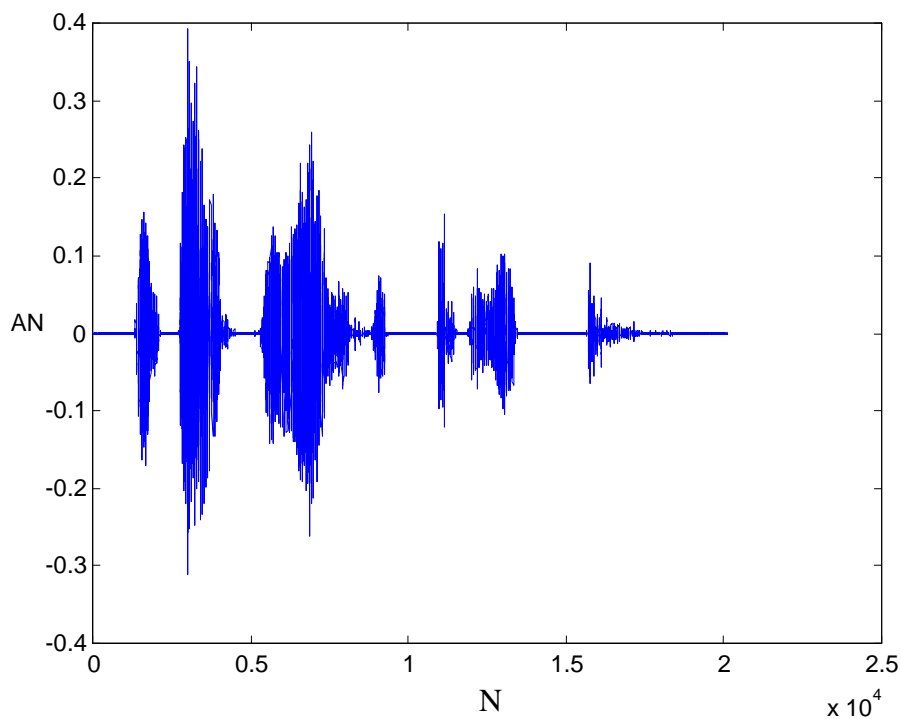


Figura 6.13 – Sinal processado $\hat{y}_{C,b}(8)$ (10 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_C(8)$ contaminado com ruído branco $r_b(8)$.

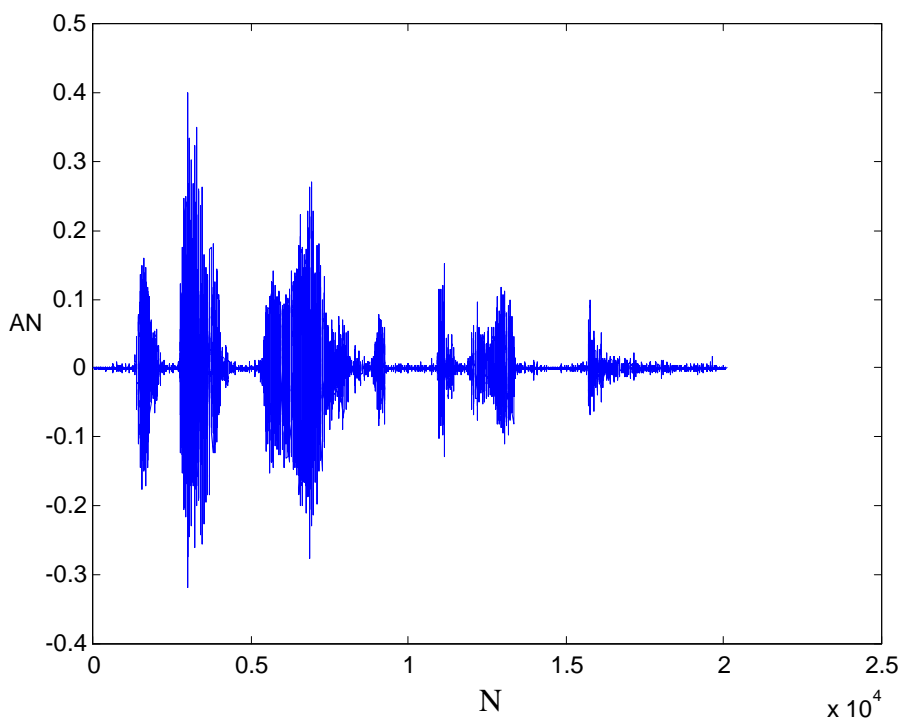


Figura 6.14 – Sinal processado $\hat{y}_{C,b}(8)$ (5 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_C(8)$ contaminado com ruído branco $r_b(8)$.

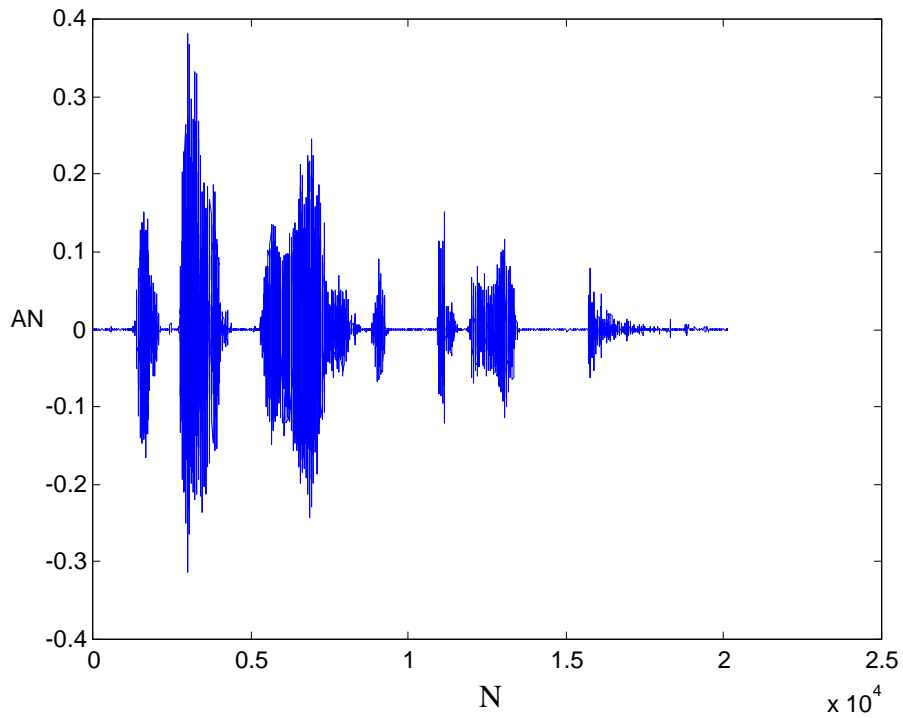


Figura 6.15 – Sinal processado $\hat{y}_{C,0}(8)$ (10 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_C(8)$ contaminado com ruído colorido $r_0(8)$.

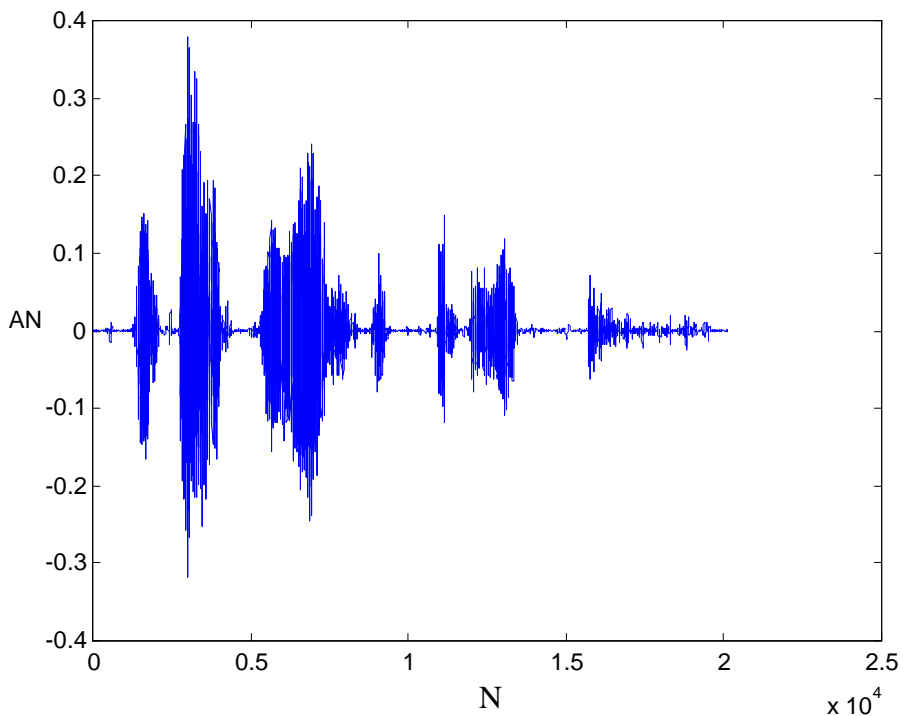


Figura 6.16 – Sinal processado $\hat{y}_{C,0}(8)$ (5 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_C(8)$ contaminado com ruído colorido $r_0(8)$.

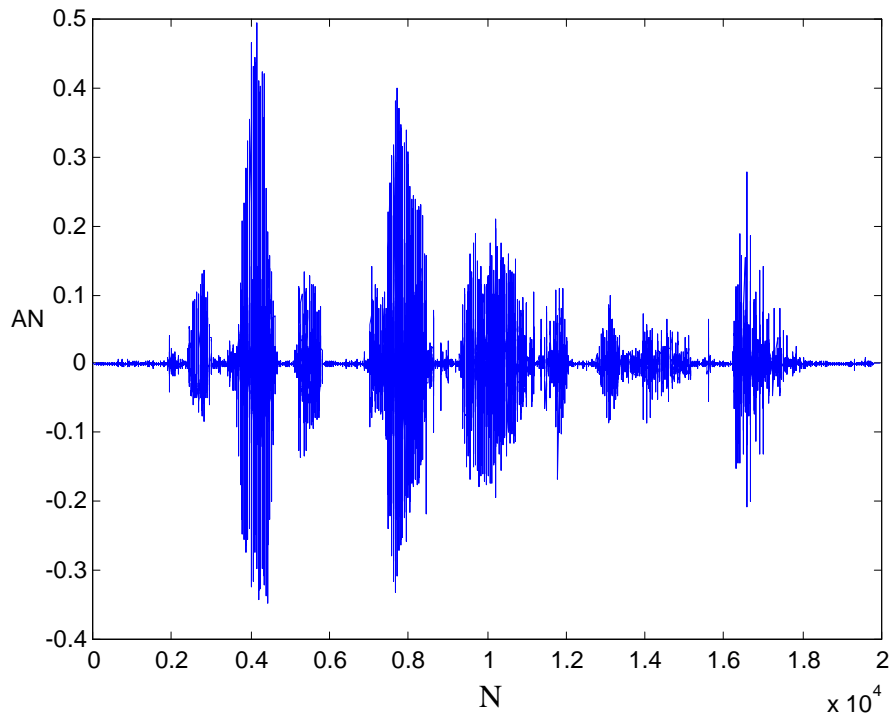


Figura 6.17 – Sinal processado $\hat{y}_{D,b}(8)$ (10 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_D(8)$ contaminado com ruído branco $r_b(8)$.

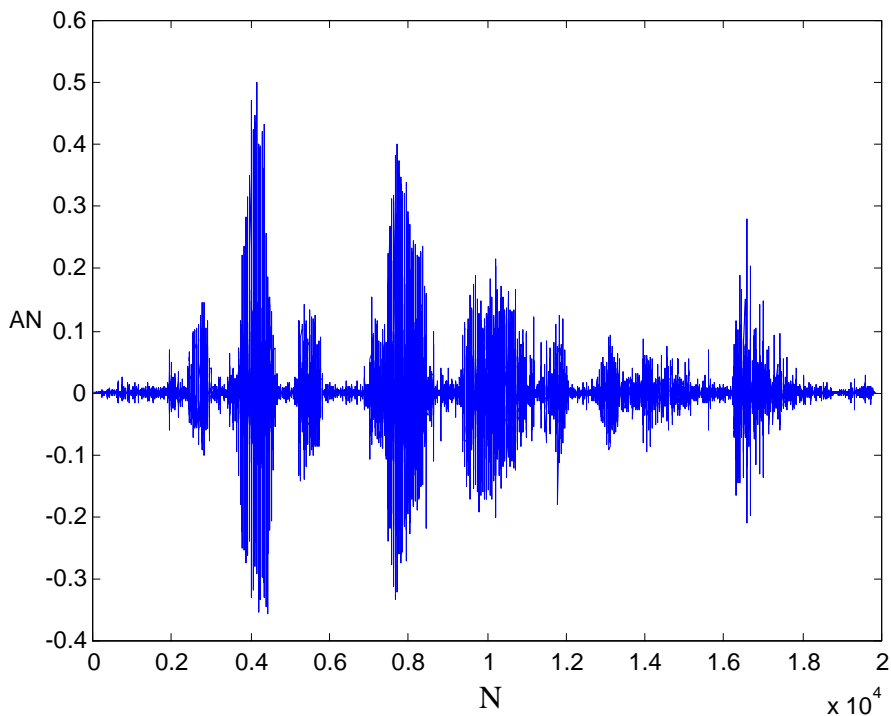


Figura 6.18 – Sinal processado $\hat{y}_{D,b}(8)$ (5 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_D(8)$ contaminado com ruído branco $r_b(8)$.

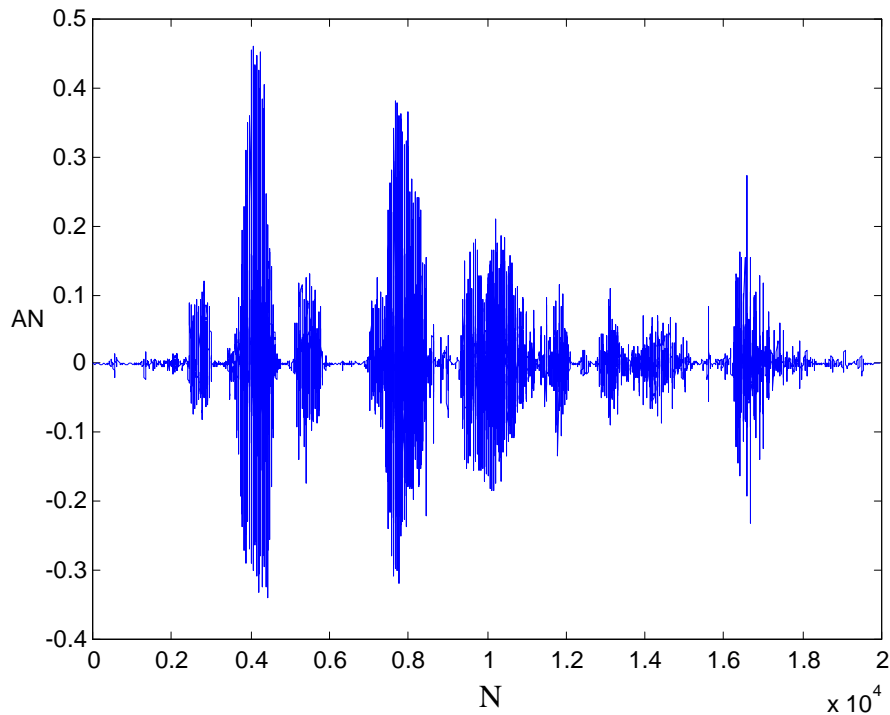


Figura 6.19 – Sinal processado $\hat{y}_{D,0}(8)$ (10 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_D(8)$ contaminado com ruído colorido $r_0(8)$.

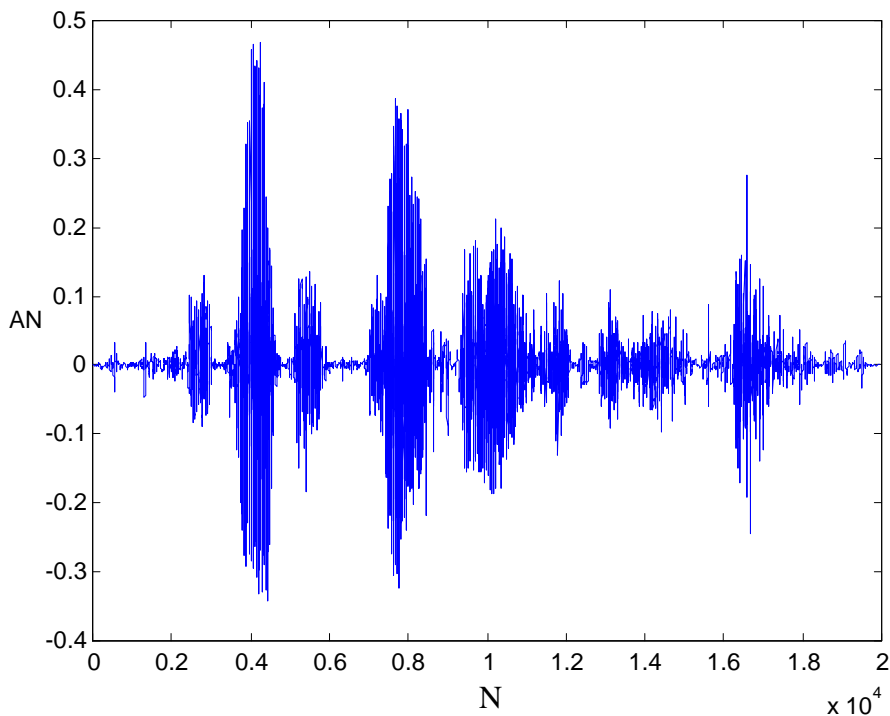


Figura 6.20 – Sinal processado $\hat{y}_{D,0}(8)$ (5 dB), usando o método proposto e a transformada *wavelet* inversa, a partir do sinal de voz limpo $x_D(8)$ contaminado com ruído colorido $r_0(8)$.

6.2 Conclusões

A SNR é uma medida objetiva que serve para avaliar o nível do ruído presente num sinal. Esta medida não determina a inteligibilidade de um sinal após o seu processamento, mas serve para indicar o nível de redução de ruído que ocorreu no sinal após o seu processamento. Assim, dados três sinais; referência (limpo), contaminado e processado, um método de redução de ruído eficaz, após o processamento, deve apresentar os valores de SNRs dos sinais limpos e processados próximos. Se os valores de SNRs dos sinais limpos e processados forem muito diferentes, significa que: ou houve muita redução de ruído e, conseqüentemente, introdução de distorções nos trechos de voz do sinal ou houve pouca redução e a inteligibilidade do sinal continua comprometida.

Os resultados apresentados na Tabela 6.3, principalmente para o sinal com ruído branco, mostram que todos os sinais tiveram uma redução significativa no nível de ruído, isto é, redução de ruído sem distorcer o sinal, pois todos os sinais apresentam valores de SNR muito próximos dos valores dos sinais limpos. Enquanto que para os sinais com ruído colorido as variações das SNR dos sinais processados ficaram em torno de 50 % a 70 % dos valores apresentados pelas SNR dos sinais limpos. Já os resultados apresentados na Tabela 6.2 mostram que todos os sinais processados apresentam SNR abaixo de 50 % quando comparados com as SNR dos sinais limpos.

Os valores da PESQ apresentados na Tabelas 6.5 indicam que os sinais processados pelo método proposto, em sua maioria, são sinais considerados bons para audição humana, pois os sinais processados que inicialmente tinham SNR igual a 10 dB, todos com exceção do sinal D colorido tiveram notas maiores que 3, para os dois tipos de ruídos. Já os sinais que inicialmente tinham SNR igual a 5 dB, quando avaliados pela PESQ, tiveram notas superiores a 3, com exceção dos sinais C, D e B colorido que tiveram notas inferiores a 3. Porém, comparando os resultados dos sinais processados, na Tabela 6.5, com os resultados dos sinais ruidosos, na Tabela 6.4, verifica-se que houve melhora em todos os sinais, independente do nível de ruído.

Em geral, os valores de SNR e PESQ apresentados nas Tabelas de 6.2 a 6.5, indicam que o método proposto reduz ruído sem introduzir fortes distorções nos sinais processados.

E ainda, pode-se comparar as formas de ondas dos sinais limpos x_k (8) (apresentados nas Figuras 6.1 a 6.4) e processados $\hat{y}_{k,l}$ (8) (apresentados nas Figuras 6.5 a 6.20).

Nas Figuras de 6.5 a 6.20 são apresentadas as formas de onda de todos os sinais processados, cujos valores de SNR e PESQ foram apresentados nas Tabelas 6.2 a 6.5. Observando as formas de onda dos sinais processados, Figuras 6.5 a 6.20, pode-se notar que o método proposto é eficiente na redução de ruído, tanto branco quanto colorido, porém essa redução é mais eficaz para os sinais com SNR 10 dB e com ruído branco (Figuras 6.5, 6.9, 6.13 e 6.17). Para os sinais com SNR 5 dB a redução é menor, principalmente para os sinais femininos contaminados com ruído colorido (Figuras 6.12 e 6.20). Porém, observando todas as figuras, percebe-se que o método provoca uma redução de ruído uniforme ao longo do sinal e, comparando os sinais processados com os respectivos sinais limpos, não há introdução de distorções nos trechos de voz.

CAPÍTULO 7

CONCLUSÕES

Neste trabalho foram estudadas técnicas de redução de ruído em sinais de voz com o objetivo de propor um método eficiente utilizando a transformada *wavelet* discreta, principalmente para o processamento de sinais contaminados por ruídos coloridos.

Consequentemente, foram estudadas as características básicas dos métodos de redução de ruído em sinais de voz, a teoria da análise *wavelet*, alguns métodos de redução de ruído em sinais de voz baseados em *wavelets* e o emprego das *wavelets* em processamentos de sinais de voz.

A transformada *wavelet* discreta é uma ferramenta potente para o processamento de sinais de voz, tanto para compressão e codificação, quanto para a melhoria do sinal. Quando se trata da melhoria do sinal de voz, o problema consiste em determinar o melhor processo de filtragem para reduzir o ruído de fundo presente no sinal, sem deteriorar os trechos de voz. Os métodos que usam o corte por limiar são os mais comuns nas técnicas baseadas em *wavelets*, tendo como principal ponto positivo o fato de que no domínio *wavelet* muitos coeficientes são redundantes e por isso podem ser eliminados sem comprometer a reconstrução do sinal. Quando esses métodos são usados, o maior problema consiste em determinar a melhor forma de se calcular e aplicar o limiar.

O objetivo deste trabalho foi realizar estudos e pesquisas visando à apresentação de uma proposta de um método não-limiar para redução de ruído em sinais de voz. A metodologia de redução de ruído proposta consiste na execução de três processamentos, cada um agindo de acordo com as suas características nas regiões de voz e silêncio sem o uso de limiar. Além dos três processamentos, foi aplicada uma função de transferência que atua fazendo o papel de um filtro em processamento de sinal de voz.

Este método tem como principal objetivo superar as deficiências dos métodos que utilizam limiares e realizar o processamento eficaz de sinais contaminados por ruídos reais.

O método proposto neste trabalho foi testado para quatro tipos de sinais de voz diferentes. Foram usados dois sinais de voz masculina e dois sinais de voz feminina, um em Inglês e o outro em Português para cada gênero. Para verificar a qualidade dos sinais processados, foram avaliados os níveis de redução de ruído e de distorção usando como medidas a SNR e a PESQ. A PESQ faz uma comparação entre o sinal original e processado baseado em parâmetros do ouvido humano enquanto a SNR compara o nível de ruído presente entre os dois sinais limpo e processado.

As SNRs dos sinais processados apresentados nas Tabelas 6.2 e 6.3 tiveram valores sempre próximos dos valores dos sinais limpos, indicando que houve redução de ruído sem introdução de distorções nos trechos de voz do sinal.

Observando as formas de onda dos sinais processados, Figuras 6.5 a 6.20, pode-se notar que o método proposto é eficiente na redução de ruído, tanto branco quanto colorido, porém essa redução é mais eficaz para os sinais com SNR 10 dB e com ruído branco (Figuras 6.5, 6.9, 6.13 e 6.17). Para os sinais com SNR 5 dB a redução é menor, principalmente para os sinais femininos contaminados com ruído colorido (Figuras 6.12 e 6.20). Porém, observando todas as figuras, percebe-se que o método provoca uma redução de ruído uniforme ao longo do sinal e, comparando os sinais processados com os respectivos sinais limpos, não há introdução de distorções nos trechos de voz.

Os valores da PESQ apresentados na Tabela 6.5, para a maioria dos sinais, foram superiores a 3, indicando que os sinais são bons para audição. Apenas o sinal D, quando processado com ruído branco e SNR 5 dB teve nota PESQ muito inferior a 3. Os outros valores de PESQ que foram inferiores a 3 estiveram muito próximos desse valor.

Em geral, os resultados mostraram que o método é eficiente, pois faz a redução de ruído no sinal sem introduzir distorções e sem mudar sua amplitude.

A principal vantagem do método proposto neste trabalho, em relação aos demais métodos de redução de ruído no domínio *wavelet*, é o uso de uma função de transferência que não utiliza limiar para reduzir o ruído. Como essa função não é descontínua, ela não introduz distorções nos trechos de voz do sinal. O uso da função sigmoide faz com que a imagem da máscara de redução de ruído não seja maior do que 1, evitando assim que a amplitude do sinal processado seja modificada em relação ao sinal de referência. Os resultados de PESQ e SNR, bem como as formas de onda dos sinais processados, comprovam a eficiência desse método, como se pode observar pelas figuras e tabelas apresentadas no capítulo 6. Dá-se assim uma nova opção para o processamento de sinais de voz.

Como contribuições da tese pode-se citar uma nova abordagem na aplicação de *wavelets* na redução de ruído em sinais de voz, eliminando o uso de limiares e, ainda, um processamento mais uniforme do sinal ruidoso, o que evita descontinuidades ao longo do processo de redução de ruído.

Como sugestões para trabalhos futuros propõe-se o uso de um único operador que contempla todas as características do sinal de voz, para facilitar implementações práticas. Outra proposta seria a implementação de um método não-limiar usando uma abordagem bayesiana.

REFERÊNCIAS

AKHAEI, M. A.; MARVASTI, I. M. A robust approach for speech enhancement using wavelet transform. In: INTERNATIONAL CONFERENCE ON TELECOMMUNICATION, ICT, 13, 2006, Funchal, Portugal. **Proceedings of the...** Funchal: IEEE, 2006. p. 671-675.

AKHAEI, M. A.; AMERI, A.; MARVASTI, F. A. Speech enhancement by adaptive noise cancellation in the wavelet domain. In: IEEE INTERNATIONAL CONFERENCE ON INFORMATION THEORY COMMUNICATIONS AND SIGNAL PROCESSING, ICICS, Bangkok, Thailand. **Proceedings of the...** Bangkok: IEEE, 2005. p. 1934-1938.

BAHOURA, M.; ROUAT, J. Wavelet speech enhancement based on the teager energy operator. **IEEE Signal Processing Letters**, New York, v.8, p. 10-12, 2001a.

BAHOURA, M.; ROUAT, J. A new approach for wavelet speech enhancement. In: EUROPEAN CONFERENCE ON SPEECH COMMUNICATION AND TECHNOLOGY, EUROSPEECH, 7, 2001, Aalborg. **Proceedings of the ...** Aalborg: ISCA, 2001b. p. 1937-1940.

BARNWELL, T. P.; VOIERS, W. D. (Ed.). **An analysis of objective measures for user acceptance of voice communication systems**. Englewood: Prentice-Hall, 1979. (DCA Final Technical Report, no. DCA 100-78-C-0003).

BARNWELL, T. P.; CLEMENTS, M. A.; QUACKENBUSH, S. R. *et al.* **Improved objective measures for speech quality testing**. Englewood: Prentice-Hall, 1984. (DCA Final Technical Report, no. DCA 100-83-C-0027).

BARNWELL, T. P. **Improved objective quality measures for low bit speech compression**. S.l.: National Science Foundation, 1985. (Final Technical Report ECS-8016712, 1985).

BEERENDS, J. G.; HEKSTRA, A. P.; RIX, A. W.; HOLLIER, M. P. Perceptual evaluation of speech quality (PESQ) the new ITU standard for end-to-end speech quality assessment part ii: psychoacoustic model. **AES Journal**, New York, v. 50, n.10, p. 765-778, 2002.

BRADLEY, J. N.; BRISLAWN, C. M. The Wavelet/scalar quantization compression standard for digital fingerprinting images. **IEEE International Symposium On Circuits and Systems**, New York, v.3, p. 205-208, 1994.

BRISLAWN, C. The FBI **fingerprint image compression standard**. Disponível em: <http://www.c3.lanl.gov/~brislawn/FBI/FBI.html>. Acesso em: 10 mar. 2007.

BRUCE, A.; GAO, H. Y. Waveshrink: shrinkage functions and thresholds. In: WAVELET APPLICATIONS IN SIGNAL AND IMAGE PROCESSING, 3, 1995, Pacific Time. **Proceedings of SPIE**. Pacific Time: SPIE, 1995. v. 2569, p. 270 – 281, 1995.

CAPPÉ, O. Elimination of the musical noise phenomenon with the ephraim and malah noise suppressor. **IEEE Transaction on ASSP**, New York, v. 2, ed. 2, p. 345-349, 1994.

CHANG, S.; KWON, Y.; YANG, S.; KIM, I. Speech enhancement for non-stationary noise environment by adaptive wavelet packet. In: INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING - ICASSP'02, 2002. **Proceedings of IEEE**. v.1, p. 561-564, 2002. Disponível em: <http://ieeexplore.ieee.org/iel5/7874/21701/01005801.pdf?arnumber=1005801>. Acesso em: 09 mar. 2008.

COHEN, I. Speech enhancement using a noncausal a priori SNR estimator. **IEEE Signal Processing Letters**, New York, v.11, n.9, p. 725-728, 2004.

DAE-SUNG, K.; JUNG-GO, C.; GEUN-TAEK, R; MUN-SEOB, B.; HYEON-DEOK, B. Noise reduction using coefficients smoothing wavelet transforms. INTERNATIONAL CONFERENCE ON SIGNAL PROCESSING APPLICATIONS AND TECHNOLOGY, ICSPAT, 2002, San Jose. **Proceedings of the...** San José: IEEE, 2002. p. 29-33.

DAUBECHIES, I. Othonormal Bases of Compactly Supported Wavelets, **Communications on Pure and Applied Mathematics**, New York, v.41, p. 909-996, 1998.

DAUBECHIES, I. **Ten lectures on wavelets**. Philadelphia: SIAM Books, 1992.

DAUBECHIES, I. The wavelet transform, time frequency localization and signal analysis. **IEEE Transactions on Information Theory**, New York, v.36, p. 961-1005, 1990.

DELLER, J. L.; PROAKIS, J. G.; HANSEN, J. H. L. **Discrete-time processing of speech signals**. New York: Macmillan, 1993.

DONOHO, D. L. De-noising by soft-thresholding. **IEEE Transactions on Information Theory**, New York, v.41, n.3, p. 613-627, 1995.

DONOHO, D. L.; JOHNSTONE, I. M. Ideal spatial adaptation via wavelet shrinkage. **Biometrika**, London, v. 81, n.3, p. 425-455, 1994.

DUARTE, M. A. Q. **Redução de ruído em sinais de voz no domínio wavelet**. 2005. Tese (Doutorado) - Faculdade de Engenharia de Ilha Solteira, Universidade Estadual Paulista, Ilha Solteira, 2005.

DUARTE, M. A. Q.; VIEIRA FILHO, J.; VILLARREAL, F. Implementação e avaliação de técnicas de redução de ruído em sinais de voz baseadas em wavelets. In: CONGRESSO BRASILEIRO DE AUTOMÁTICA, CBA, 9,2004, Gramado-RS. **Anais...** Gramado: SBA, 2004, 6p. (CD-ROM).

DUARTE, M. A. Q.; OLIVEIRA, L. C. O. ; VILLARREAL, F.; DÍAZ, L. A. Compressão de sinais elétricos usando a transformada de wavelet. **TEMA: Tendências em Matemática Aplicada e Computacional**, São Carlos, v.4, n.1, p. 21-30, 2003.

EPHRAIM, Y.; MALAH, D. Speech enhancement using a minimum mean square error short time spectral amplitude estimator. **IEEE Transactions Acoustics Speech Signal Processing**, New York, v.32, n.6, p. 1109-1121, 1984a.

EPHRAIM, Y.; MALAH, D. Speech enhancement using a minimum mean square error log spectral amplitude estimator. **IEEE Transactions Acoustics Speech Signal Processing**, New York, v.33, n.2, p. 443-445, 1985b.

FERNANDES, J. C. **Acústica e ruídos**. Faculdade de Engenharia, Departamento de Engenharia Mecânica, Laboratório de Acústica e vibrações – LAV, Universidade Estadual Paulista, Campus de Bauru. Bauru: FE/UNESP, 2005. (Apostila). Disponível em: <http://wwwp.feb.unesp.br/jcandido/acustica/Apostila/Capitulo%2002.pdf>. Acesso em: 03 mar. 2007.

FLANAGAN, J. L. **Speech analysis synthesis and perception**. Berlin:Springer-Verlag, 1972. 444p.

GRAPS, A. An Introduction to wavelets. **IEEE Computacional Science and Engineering**, Los Alamitos, v.2, n.2, p.1-18, 1995. Disponível em: <<http://www.cis.udel.edu/~amer/CISC651/IEEEwavelet.pdf>>. Acesso em: 15 dez. 2007.

GRAY Jr., A. H.; MARKEL, J. D. Distance measures for speech processing. **IEEE Transactions on Acoustics, Speech and Signal Processing**, New York, v. 24, n.5, p.380 – 391, 1976.

HAAR, A. Zur theorie der orthogonalen functionen-systeme. **Mathematische Annalen**, Berlin, v.69, p. 331-371, 1910.

HECKER, M. H. L.; WILLIAMS, C. E. Choice of reference conditions for speech preference tests. **Journal of the Acoustical Society of America**, New York, v.39, p. 946-952, 1966.

HERNANDES, E.; WEISS, G. **A first course on wavelet**. Boca Raton: CRC Press, 1996.

HOWARD, D. M.; ANGUS, J. **Acoustics and psychoacoustics**. Oxford: Focal Press, 1996.

HU, Y.; LOIZOU, P. C. Speech enhancement based on wavelet thresholding the multipaper spectrum. **IEEE Transactions on Speech and Audio Processing**, New York, v.12, n.1, p. 59-67, 2004.

IRINEU, A. J. **Redução de ruído em sinais de voz usando curvas especializadas de modificação dos coeficientes da transformada em co-seno**. 2006. 112 f. Tese (Doutorado) - Escola Politécnica, Universidade de São Paulo, São Paulo, 2006.

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.** Geneva: ITU, 2001. (ITU-T P.862).

INTERNATIONAL TELECOMMUNICATIONS UNION. **Methods for subjective determination of transmission quality.** Geneva: ITU, 1996. (ITU-T P.800).

JOHNSTONE, I. M.; SILVERMAN, B. W. Wavelet threshold estimators for data with correlated noise. **Journal of the Royal Statistical Society. Methodological . Serie B** , London, v.59, p.319-351, 1997.

KIM, I. J.; YANG, S. I.; KWON, Y. Speech enhancement using adaptive wavelet shrinkage. PROCEEDINGS OF IEEE INTERNATIONAL SYMPOSIUM ON INDUSTRIAL ELECTRONICS, ISIE, 2001, Pusan. **Proceedings of the...** Pusan: IEEE, 2001. v. 1, p. 501 – 504.

LALLOUANI, A.; GABREA, M.; GARGOUR, C. S. Wavelet based speech enhancement using two different threshold-based denoising algorithms. In: ELECTRICAL AND COMPUTER ENGINEERING, CANADIAN CONFERENCE, 2004, Canada. **Conference...** Canadá: IEEE, 2004. v.1, p. 315-318, 2004. Disponível em: <http://ieeexplore.ieee.org/xpl/tocresult.jsp?isnumber=29618&isYear=2004>. Acesso em: 09 mar. 2008.

LIMA, P. C. **Wavelets: teoria, algoritmos e aplicações.** Belo Horizonte: Departamento de Matemática-ICEX – UFMG, 2004.

LOUIS, A. K.; MAAB, P.; RIEDER, A. **Wavelets theory and applications.** Chichester: John Wiley & Sons, 1998.

LU, C. T.; WANG, H. C. Speech enhancement using robust weighting factors for critical-band-wavelet-packet transform. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH, AND SIGNAL PROCESSING, ICASSP, 29, 2004, Canadá. **Proceedings of the...** Canadá: IEEE, 2004. Disponível em: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1326087. Acesso em : 09 mar. 2008.

LYNN, P. A.; FUERST, W. **Introductory digital signal processing with computer applications.** New York: John Wiley & Sons, 1994.

- MALLAT, S. A Theory for multiresolution representation signal decomposition: the wavelet representation. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, New York, v.11, n.7, p. 674-693, 1989a.
- MALLAT, S. Multiresolution approximation and wavelets. **Transactions American Mathematical Society**, Providence, v.315, p. 69-88, 1989b.
- MARTINS, A. **Signal analysis: wavelets, filter banks, time-frequency transforms, and applications**. New York: John Wiley & Sons, 2000.
- MEENEN, P.; ADHAMI, R. Fingerprinting for security. **IEEE Potentials**, New York, v. 20, ed. 3, p. 33 – 38, 2001.
- MEYER, Y. **Wavelets: algorithms and applications**. Philadelphia: SIAM Books, 1993.
- MISITI, M., MISITI, Y.; OPPENHEIM, G. ;POGGI, J. **Matlab: wavelet toolbox user's guide**. Natick: Math Works, 1996.
- MORETTIN, P. A. **Ondas e ondaletas: da análise de fourier à análise de ondaletas**. São Paulo: Edusp, 1999.
- NIEVERGELT, Y. **Wavelets made easy**. Boston: Birkhäuser, 1999.
- NISHIMURA, R.; ASANO, F.; SUZUKI, Y.; SONE, T. Speech enhancement using spectral subtraction with wavelet transform. **Electronics and Communications in Japan**, Tokyo, v.81, n.1, pt. 3, p. 24-31, 1998.
- NOCETI FILHO, S. **Fundamentos sobre ruídos**. Parte II – definição, caracterização e tipos de ruídos. Parte III – ruído branco e ruído rosa. Universidade Federal de Santa Catarina. Disponível em: <http://www.selenium.com.br/site2004/downloads/trabalhos/ruídos.pdf>. Acesso em: 05 jul. 2007.
- NORDSTRÖM, F. **Time and frequency dependent noise reduction speech signals**. 2002. Master (Thesis) - Department of Mathematical Statistics, Lund Institute of Technology, Lund, Sweden, 2002.
- OLIVEIRA, H. M. **Análise de sinais para engenheiros: uma abordagem via wavelets**. Rio de Janeiro: Brasport, 2007.
- QIANG, F.; WAN, E. A. **A novel speech enhancement system based on wavelet denoising**. School of Science and Engineering at OHSU, Center of Spoken

Language Understanding. 2003. Disponível em: <http://cslu.cse.ogi.edu/publications/ps/fu03.pdf>. Acesso em: 09 mar. 2008.

QUACKENBUSH, S. R. **Objective measures of speech quality.** 1985. Thesis (Ph.D) - Georgia Institute of Technology, Georgia, 1985.

RIOUL, O.; VETTERLI, M. Wavelets and signal processing. **IEEE Signal Processing Magazine**, New York, v.8, n.4, p. 14-38, 1991.

RIX, A. W.; BEERENDS, J. G.; HOLLIER, M. P.; HEKSTRA, A. P. PESQ – the new ITU standard for end-to-end speech quality assessment. In: AES CONVENTION, 109, 2000, Los Angeles. **Aes Convention...** Los Angeles: S.n., 2000. 18p.

SARKAR, K.; SU, C.; ADVE, R.; PALMA, M. S.; CASTILHO, L. G.; BOIX, R. R. A tutorial on wavelets from an electrical engineering perspective, Part 1: discrete wavelet techniques. **IEEE Antennas and Propagation Magazine**, New York, v.40, n.5, p. 49-70, 1998.

SCHROEDER, M. R.; ATAL, B. S.; HALL, J. L. Optimizing digital speech coders by exploiting masking properties of human ear. **Journal of the Acoustical Society of America**, New York, v. 66, p. 1647-1652, 1979.

SEOK, J. W.; BAE, K. S. Speech enhancement with reduction of noise components in the wavelet domain. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, ICASSP'97, 1997, Munich. **Proceedings of the...** Munich: IEEE, 1997. v.2, p. 1323-1326.

SHEIKHZADEH, H.; ABUTALEBI, H. R. An improved wavelet-based speech enhancement system. In: EUROPEAN CONFERENCE ON SPEECH COMMUNICATION AND TECHNOLOGY, EUROSPEECH, 7, 2001, Aalborg. **Proceedings of the...** Aalborg: ISCA, p. 1855-1858, 2001.

SILVA, A.V. ;EYNG, J. **Wavelets e wavelet de packets.** 2000. Disponível em: <http://www.inf.ufsc.br/~visao/2000/Wavelets/index.html>. Acesso em: 18 out. 2007. (Seminário Visão Computacional, CPGCC).

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Análise de métodos de redução de ruído por limiar no domínio wavelet. **TEMA: Tendências em Matemática Aplicada e Computacional**, São Carlos, v.9, n.3, p.471-480, 2008a.

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Redução não-linear de ruído no domínio wavelet. In BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, DINCON'2008, 5, 2008, Presidente Prudente. **Proceedings of the...** Presidente Prudente: SBMAC, 2008b. p.961 – 967.

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Análise de métodos de redução de ruído em sinais de voz no domínio wavelet. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, DINCON'2007, 6, 2007, São José do Rio Preto. **Proceedings of the...** São José do Rio Preto: SBMAC, 2007a. p 918 – 923.

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Sobre a transformada wavelet discreta. In: SIMPÓSIO REGIONAL DE MATEMÁTICA E SUAS APLICAÇÕES, SRMAIS, 1, 2007, Ilha Solteira. **Simpósio...** Ilha Solteira: Unesp/Feis, 2007b. p.1-4, (CD-ROM).

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Analysis of thresholding methods in the wavelet domain. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, DINCON'2006, 6, 2006, Guaratinguetá. **Proceedings of the...** Guaratinguetá: SBMAC, 2006. (CD-ROM).

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Wavelets: teoria e aplicações. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, DINCON'2005, 6, 2005, Bauru. **Proceedings of the...** Bauru: SBMAC, 2005.(CD-ROM).

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Wavelets: Theory and Applications. In: ITALIAN-LATIN AMERICAN CONFERENCE ON APPLIED AND INDUSTRIAL MATHEMATICS, 5, 2004, Trujillo - Peru. **Resumenes do Italian-Latin American Conference on Applied and Industrial Mathematics.** Trujillo: Universidad Nacional de Trujillo – Peru. 2004. p 26.

STORM, H. **Noise reduction of speech signals with wavelets.** Gothenburg: Department of Mathematics, Chalmers University of Technology and Gothenburg University, 1998. (Tech. rep. No 1998-02/ISSN 0347-2809).

STRANG, G.; NGUYEN, T. **Wavelets and filter banks.** Wellesley:Wellesley Cambridge Press, 1996. 490 p.

SWELDENS, W; SCHRÖDER, P. Building your own wavelets at home. Disponível em: <http://cm.bell-labs.com/who/wim/papers/athome/athome.pdf>. Acesso em: 10 mar. 2007.

TICO, M.; KUOSMANEN, P.; SAARINEN, J. Wavelet domain features for fingerprint recognition. **Electronics Letters**, London, v.37, n.1, p. 21-22, 2001.

TSOUKALAS, D.; PARASKEVAS, M.; MOURJOPOULOS, J. Speech enhancement using psychoacoustics criteria. In: IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH AND SIGNAL PROCESSING, 1993, Minneapolis. **Proceedings of the...** Minneapolis: IEEE, 1993. p. II.359-II.362.

VIEIRA FILHO, J. **Redução de ruído em sinais de voz nos sistemas rádio móveis veiculares**. 1996. 113 f. Tese (Doutorado) - Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, 1996.

VISSER, E.; LEE, T.; OTSUKA, M. Speech enhancement in a noisy car environment. In: INTERNATIONAL WORKSHOP ON INDEPENDENT COMPONENT ANALYSIS, ICA'01, 2001, San Diego. **International...** San Diego: ICA, 2001. p. 272 – 277.

VOIERS, W. D. Diagnostic acceptability measure for speech communication systems. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS, SPEECH, AND SIGNAL PROCESSING, 1997, Hartford. **Proceedings of the...** Hartford: IEEE, 1977. p. 204-207.

WANG, D. L.; LIM, J. S. The unimportance of phase in speech enhancement. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, New York, v.30, p. 679-681, 1982.

YOUNG, R. K. **Wavelet Theory and its Applications**. Boston: Kluwer Academic Publisher, 1995.

APÊNDICE A

A1. PROCESSO DE DECOMPOSIÇÃO USANDO A TRANSFORMADA WAVELET

Nesta parte, serão realizados os procedimentos básicos de decomposição de um sinal usando a Transformada *Wavelet* (TW) e de reconstrução usando a Transformada *Wavelet* Inversa (TWI). Esses processos serão denominados de *procedimentos*.

Para realizar esses procedimentos, por um lado, suponha que é dado um sinal, indicado por $s(n)$, no domínio do tempo. O sinal $s(n)$ será decomposto, usando a TW, num sinal, considerado no domínio *wavelet*, domínio este denominado de *domínio transformado*. O sinal decomposto, considerado no domínio transformado, será indicado por $S[n]$. Por outro lado, dado um sinal $\hat{S}[n]$ no domínio transformado, será decomposto, usando a TWI, num sinal $\hat{s}(n)$, no domínio do tempo. Uma ilustração de decomposição usando a TWI é dada no apêndice C. Cabe observar que, no caso em que $\hat{S}[n]$ é o sinal $S[n]$, acima citado, o sinal $\hat{s}(n)$ é necessariamente o sinal $s(n)$, considerado para obter $S[n]$, ou seja, o sinal $s(n)$ é o sinal reconstruído a partir de $S[n]$, usando a TWI.

Para a análise dos procedimentos a serem apresentados a seguir, serão efetuadas as operações necessárias em cada nível de resolução. Também serão

apresentadas as curvas poligonais associadas em cada fase do processamento, para uma melhor ilustração dos procedimentos a serem efetuados.

Cabe observar que, quando for considerado um sinal de comprimento N (ou seja, um sinal com N coeficientes), o número máximo de níveis que a transformada *wavelet* discreta pode decompor este sinal é $n = \log_2 N$. Assim, de acordo com o tipo de processamento que se deseja fazer no sinal, para decompor um sinal, escolhe-se o número de níveis da transformada *wavelet* discreta.

Para a ilustração dos procedimentos, na decomposição de um sinal, será usada a transformada *wavelet* de *Haar*. Assim, os processos que serão seguidos para a decomposição são técnicas específicas dessa transformada (BEYLKIN; COIFMAN; ROKHLIN, 1990).

Suponha que um dado sinal $s(n)$ é de comprimento 2^n , ou seja:

$$s(n) = \left[s_{2k}, s_{2k+1} : 0 \leq k \leq \frac{2^n - 2}{2} \right] = [s_0, s_1, s_2, s_3, \dots, s_{2k}, s_{2k+1}, \dots, s_{2^n-2}, s_{2^n-1}].$$

Para ilustrar os procedimentos acima citados, considere o sinal $s(n)$ sendo um vetor com 8 elementos, ou seja;

$$s(3) = [s_{2k}, s_{2k+1} : k = 0, 1, 2, 3] = [s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7].$$

O vetor apresentado em seguida, com os valores numéricos, é apenas um exemplo, para explanar a aplicação dos procedimentos da transformada *wavelet* e da transformada *wavelet* inversa. No que se segue o vetor $s(3)$, que ilustrará as aplicações das transformadas, será:

$$s(3) = [0,5 \quad 0,4 \quad -0,2 \quad -0,7 \quad -0,8 \quad 0,5 \quad -0,1 \quad 0] \quad (\text{A.1})$$

conforme apresentado na Figura A1, que é uma representação dos seus coeficientes. Essa representação é a curva poligonal passando pelos pontos $(i + 1, s_i)$ para $i = 0, \dots, 7$.

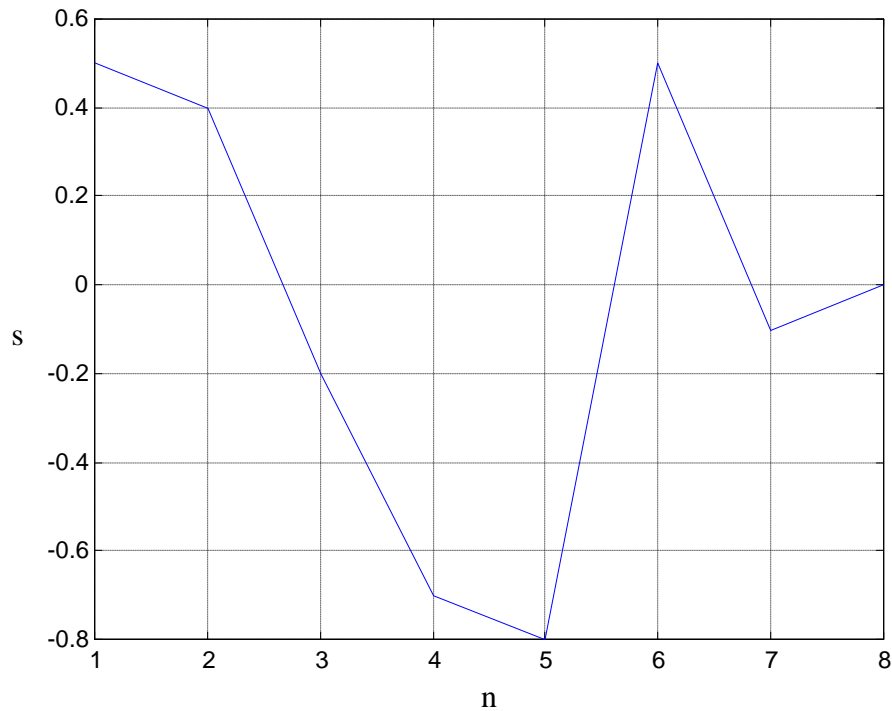


Figura A1 – Representação dos coeficientes do sinal $s(3)$.

Para realizar a decomposição do sinal $s(n)$, considerado no domínio do tempo, usando a transformada *wavelet* de *Haar*, é aplicado ao sinal $s(n)$ um processo de decomposição para calcular os vetores $s^{(n-k)}$ (decomposição do sinal $s(n)$ no passo k), $k=1, \dots, n$, associados a $s(n)$, seguindo os passos que são descritos a seguir. Esses passos seguem a seguinte sequência

$$s(n) \mapsto s^{(n-1)} \mapsto s^{(n-2)} \mapsto \dots \mapsto s^{(n-n)}.$$

O sinal $s^{(n-n)}$, obtido no último passo de decomposição, é indicado por $S[n]$, ou seja, o sinal $S[n]$, no domínio transformado, é o sinal decomposto a partir de $s(n)$.

Primeiro Passo: Cálculo de $s^{(n-1)}$

O cálculo dos coeficientes do sinal $s^{(n-1)}$, usando a TW de *Haar*, é realizado usando as fórmulas (BEYLKIN; COIFMAN; ROKHLIN, 1990)

$$d_k^{j+1} = \frac{S_{2k-1}^j - S_{2k}^j}{2}$$

e

$$s_k^{j+1} = \frac{s_{2k-1}^j + s_{2k}^j}{2}$$

De acordo com as fórmulas dadas acima, os coeficientes do sinal $s^{(n-1)}$ são:

$$s^{(n-1)} = \left[\frac{s_{2k} + s_{2k+1}}{2}, \frac{s_{2k} - s_{2k+1}}{2} : 0 \leq k \leq \frac{2^n - 2}{2} \right]$$

isto é,

$$s^{(n-1)} = \left[\frac{s_0 + s_1}{2}, \frac{s_0 - s_1}{2}, \frac{s_2 + s_3}{2}, \frac{s_2 - s_3}{2}, \dots, \frac{s_{2k} + s_{2k+1}}{2}, \frac{s_{2k} - s_{2k+1}}{2}, \dots, \frac{s_{2^n-2} + s_{2^n-1}}{2}, \frac{s_{2^n-2} - s_{2^n-1}}{2} \right].$$

Efetuada os cálculos descritos no vetor anterior, no caso particular $n = 3$, utilizando os coeficientes do sinal $s(3)$, tem-se a resultante da transformada processada na primeira iteração, o vetor $s^{(3-1)}$.

$$s^{(3-1)} = \left[\frac{s_0 + s_1}{2}, \frac{s_0 - s_1}{2}, \frac{s_2 + s_3}{2}, \frac{s_2 - s_3}{2}, \frac{s_4 + s_5}{2}, \frac{s_4 - s_5}{2}, \frac{s_6 + s_7}{2}, \frac{s_6 - s_7}{2} \right]$$

$$s^{(3-1)} = \left[\frac{0,5 + 0,4}{2}, \frac{0,5 - 0,4}{2}, \frac{-0,2 + (-0,7)}{2}, \frac{-0,2 - (-0,7)}{2}, \frac{-0,8 + 0,5}{2}, \frac{-0,8 - 0,5}{2}, \frac{-0,1 + 0}{2}, \frac{-0,1 - 0}{2} \right]$$

Os elementos desse vetor serão denotados da seguinte forma:

$$s^{(3-1)} = [s_0^{(3-1)}, s_1^{(3-1)}, s_2^{(3-1)}, s_3^{(3-1)}, s_4^{(3-1)}, s_5^{(3-1)}, s_6^{(3-1)}, s_7^{(3-1)}]$$

que, uma vez efetuados os cálculos, é o vetor

$$s^{(3-1)} = [0,45 \quad 0,05 \quad -0,45 \quad 0,25 \quad -0,15 \quad -0,65 \quad -0,05 \quad -0,05] \quad (\text{A.2})$$

Na Figura A2 apresentam-se duas curvas poligonais, sendo a azul uma representação dos coeficientes do sinal original proposto $s(3)$ e em vermelho uma representação dos coeficientes resultantes da transformada efetuada no primeiro passo, conforme o vetor $s^{(3-1)}$ resultante da operação apresentada acima.

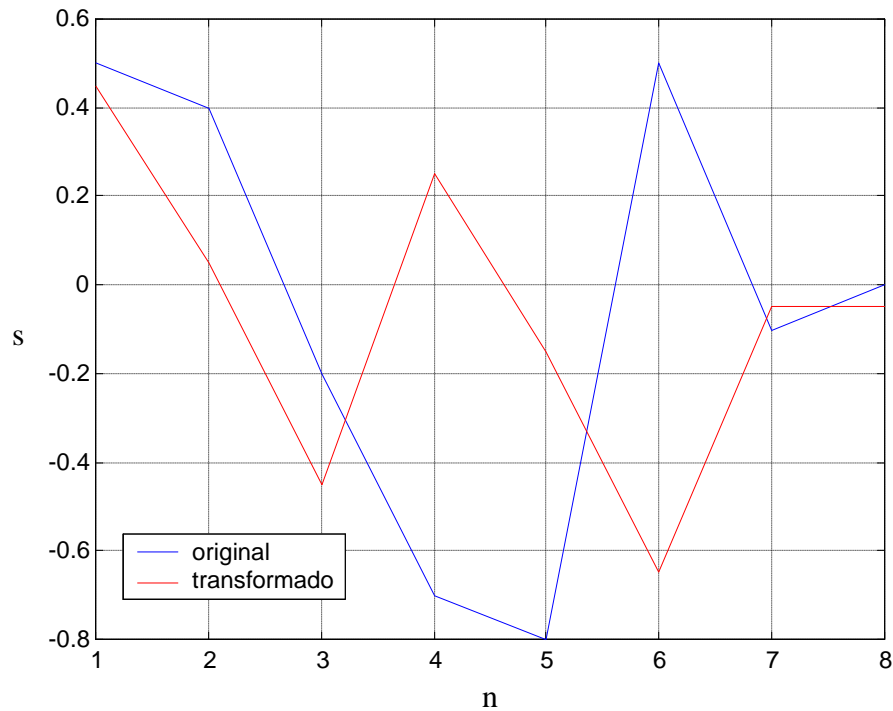


Figura A2 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $s^{(3-1)}$, resultante do primeiro processamento de decomposição.

Segundo Passo: Cálculo de $s^{(n-2)}$

Na segunda iteração, a operação da transformada *wavelet* para os coeficientes de aproximação e de detalhes, os coeficientes do vetor $s^{(n-2)}$ são calculados da seguinte forma:

$$s^{(n-2)} = \left[\begin{array}{l} \frac{s_{2k} + s_{2k+1} + s_{2k+2} + s_{2k+3}}{4}, \frac{s_{2k} - s_{2k+1}}{2}, \frac{s_{2k} + s_{2k+1} - s_{2k+2} - s_{2k+3}}{4}, \frac{s_{2k+2} - s_{2k+3}}{2} \\ k=0, \frac{2^n - (n+1)}{2} \end{array} \right].$$

No caso particular, os coeficientes do vetor $s^{(3-2)}$ são calculados assim:

$$s^{(3-2)} = \left[\begin{array}{l} \frac{s_0 + s_1 + s_2 + s_3}{4}, \frac{s_0 - s_1}{2}, \frac{s_0 + s_1 - s_2 - s_3}{4}, \frac{s_2 - s_3}{2}, \frac{s_4 + s_5 + s_6 + s_7}{4}, \frac{s_4 - s_5}{2}, \\ \frac{s_4 + s_5 - s_6 - s_7}{4}, \frac{s_6 - s_7}{2} \end{array} \right].$$

Substituindo os coeficientes do sinal original $s(3)$ nesse vetor, obtém-se:

$$s^{(3-2)} = \left[\begin{array}{l} \frac{0,5 + 0,4 + (-0,2) + (-0,7)}{4}, \frac{0,5 - 0,4}{2}, \frac{0,5 + 0,4 - (-0,2) - (-0,7)}{4}, \frac{-0,2 - (-0,7)}{2}, \\ \frac{-0,8 + 0,5 + (-0,1) + 0}{4}, \frac{-0,8 - (0,5)}{2}, \frac{-0,8 + 0,5 - (-0,1) - 0}{4}, \frac{-0,1 - 0}{2} \end{array} \right].$$

Denotando, os coeficientes desse vetor, por

$$s^{(3-2)} = [s_0^{(3-2)}, s_1^{(3-2)}, s_2^{(3-2)}, s_3^{(3-2)}, s_4^{(3-2)}, s_5^{(3-2)}, s_6^{(3-2)}, s_7^{(3-2)}].$$

tem-se

$$s^{(3-2)} = [0 \quad 0,05 \quad 0,45 \quad 0,25 \quad -0,1 \quad -0,65 \quad -0,05 \quad -0,05] \quad (\text{A.3})$$

Na Figura A3 apresentam-se duas curvas poligonais, sendo a azul a representação dos coeficientes do sinal original proposto $s(3)$ e em vermelho dos coeficientes resultantes da transformada efetuada na segunda iteração do vetor $s^{(3-2)}$.

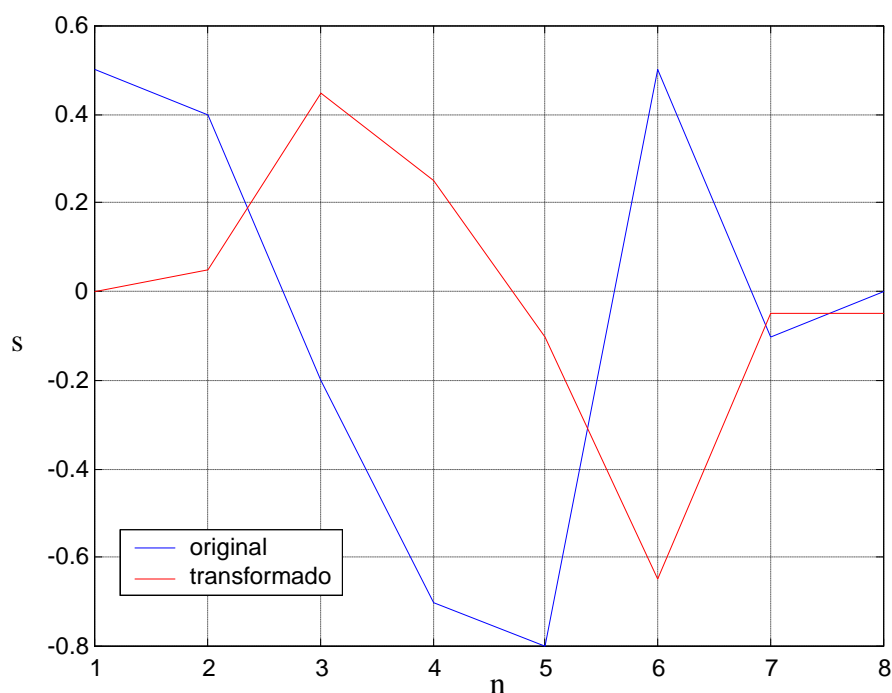


Figura A3 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $s^{(3-2)}$, resultante do segundo processamento de decomposição.

Terceiro Passo: Cálculo de $s^{(n-3)} = S[3]$

Como o sinal original $s(3)$ adotado tem apenas oito coeficientes, esta iteração configura-se como sendo a última. Assim, o vetor $s^{(3-3)}$ é a resultante da operação da transformada *wavelet*. Em geral, para a terceira iteração (a última, neste caso), a operação da transformada *wavelet* nos coeficientes de aproximação e de detalhes, se resume em efetuar o seguinte procedimento:

$$s^{(3-3)} = \left[\begin{array}{cccc} \frac{s_0 + s_1 + s_2 + s_3 + s_4 + s_5 + s_6 + s_7}{8}, & \frac{s_0 - s_1}{2}, & \frac{s_0 + s_1 - s_2 - s_3}{4}, & \frac{s_2 - s_3}{2}, \\ \frac{s_0 + s_1 + s_2 + s_3 - s_4 - s_5 - s_6 - s_7}{8}, & \frac{s_4 - s_5}{2}, & \frac{s_4 + s_5 - s_6 - s_7}{4}, & \frac{s_6 - s_7}{2} \end{array} \right].$$

Na sequência apresentam-se os coeficientes do sinal original $s(3)$ efetuando as operações propostas pelo vetor $s^{(3-3)}$, resultando em:

$$s^{(3-3)} = \left[\begin{array}{ccc} \frac{0,5+0,4+(-0,2)+(-0,7)+(-0,8)+0,5+(-0,1)+0}{8}, & \frac{0,5-0,4}{2}, & \frac{0,5+0,4-(-0,2)-(-0,7)}{4}, \\ \frac{-0,2-(-0,7)}{2}, & \frac{0,5+0,4+(-0,2)+(-0,7)-(-0,8)-0,5-(-0,1)-0}{8}, & \frac{-0,8-(0,5)}{2}, \\ \frac{-0,8+0,5-(-0,1)-0}{4}, & \frac{-0,1-0}{2} & \end{array} \right]$$

onde seus coeficientes são denotados como segue

$$s^{(3-3)} = [s_0^{(3-3)}, s_1^{(3-3)}, s_2^{(3-3)}, s_3^{(3-3)}, s_4^{(3-3)}, s_5^{(3-3)}, s_6^{(3-3)}, s_7^{(3-3)}].$$

Tem-se

$$s^{(3-3)} = [-0,05 \quad 0,05 \quad 0,45 \quad 0,25 \quad 0,05 \quad -0,65 \quad -0,05 \quad -0,05] \quad (\text{A.4})$$

O sinal $s^{(3-3)}$ representa a decomposição completa do sinal $s(3)$ pela transformada *wavelet*.

No domínio *wavelet*, dado um sinal no domínio do tempo $s(n)$, o sinal decomposto $s^{(n-n)}$, conforme já dito anteriormente, é denotado por $S[n]$.

Assim, no caso do exemplo apresentado, o sinal $S[3]$ denotará o sinal decomposto $s^{(3-3)}$. Tem-se então:

$$S[3] = [-0,05 \quad 0,05 \quad 0,45 \quad 0,25 \quad 0,05 \quad -0,65 \quad -0,05 \quad -0,05] \quad (\text{A.5})$$

Na Figura A4 apresentam-se as duas curvas poligonais representando coeficientes de sinais, sendo a azul os coeficientes do sinal original proposto $s(3)$ e em vermelho os coeficientes da decomposição completa do sinal $s(3)$ pela transformada *wavelet*, o vetor $S[3]$.

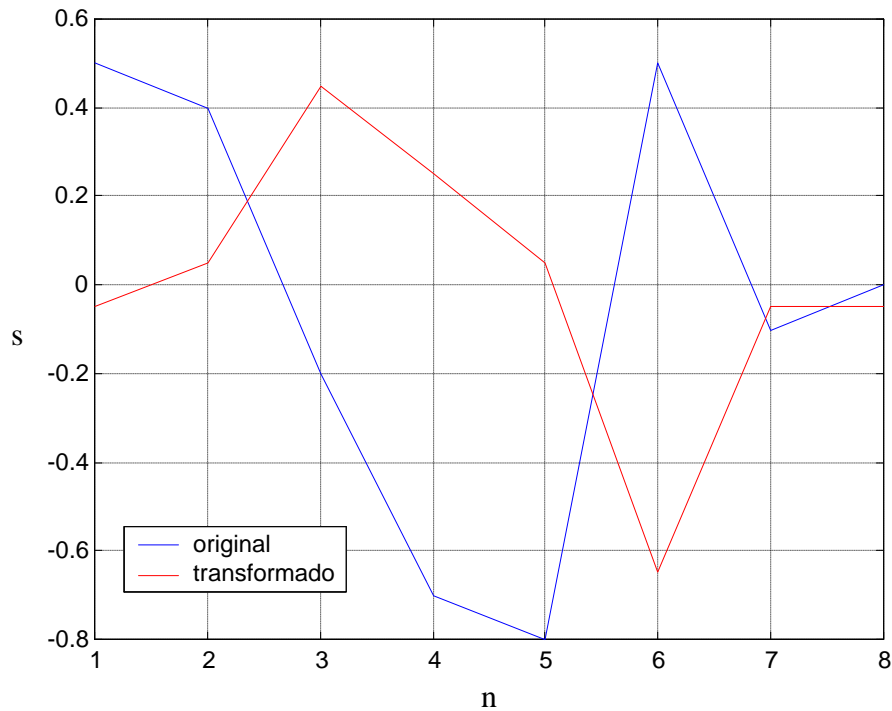


Figura A4 - Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $S[3]$, resultante do terceiro processamento de decomposição.

A2. PROCESSO DE RECONSTRUÇÃO USANDO A TRANSFORMADA WAVELET INVERSA

De um modo geral, para decompor um sinal $\hat{S}[n]$, considerado no domínio transformado, usando a transformada *wavelet* inversa de *Haar*, é aplicado ao sinal $\hat{S}[n]$ um processo de decomposição calculando os sinais, que são indicados por $\hat{s}_{(n-k)}$, para $k=1, \dots, n$, seguindo a seguinte sequência

$$\hat{S}[n] \mapsto \hat{s}_{(n-1)} \mapsto \hat{s}_{(n-2)} \mapsto \dots \mapsto \hat{s}_{(n-n)}.$$

O sinal $\hat{s}_{(n-n)}$ obtido no último passo de decomposição é indicado por $\hat{s}(n)$, sinal este considerado no domínio do tempo. No caso particular em que $\hat{S}[n]$ é o sinal $S[n]$, obtido conforme os passos citados no item anterior, o sinal $\hat{s}(n)$ é necessariamente o sinal $s(n)$ considerado no início deste apêndice. Assim, o sinal $s(n)$ será obtido seguindo a seguinte sequência de passos

$$S[n] = s^{(n-n)} \mapsto \hat{s}_{(n-n)} = s^{(n-(n-1))} \mapsto \hat{s}_{(n-(n-1))} = s^{(n-1)} \mapsto \dots \mapsto \hat{s}_{(n-n)} = s(n)$$

sendo $s^{(n-k)}$, $k=1, \dots, n$ os sinais considerados no item anterior.

No caso em que $n = 3$, para a reconstrução, usando a transformada *wavelet* inversa de *Haar*, do sinal $s(3)$, exemplo do item anterior, a partir do sinal $S[3]$, obtido pelo processo de decomposição no item anterior:

$$S[3] = [-0,05 \quad 0,05 \quad 0,45 \quad 0,25 \quad 0,05 \quad -0,65 \quad -0,05 \quad -0,05]$$

será seguida a seguinte sequência de passos

$$S[3] = s^{(3-3)} \mapsto \hat{s}_{(3-1)} = s^{(3-2)} \mapsto \hat{s}_{(3-2)} = s^{(3-1)} \mapsto \hat{s}_{(3-3)} = s(3).$$

Para a análise destes procedimentos serão efetuados os cálculos em cada nível de resolução, sendo ainda, apresentadas as curvas poligonais, que representam os coeficientes do sinal reconstruído, em cada fase do processamento. Segue então, os procedimentos para efetuar a operação de reconstrução do sinal, usando a transformada *wavelet* inversa de *Haar*.

Primeiro Passo: Reconstrução do sinal $s^{(3-2)}$

A operação de reconstrução do sinal transformado para o domínio do tempo na primeira iteração é feita executando as duas operações algébricas descritas abaixo. O processo consiste na obtenção de dois novos coeficientes de aproximação s_0 e s_4 , que serão utilizados na segunda iteração do processo. Para esse fim, faz-se o uso dos coeficientes de aproximação do sinal transformado $S[3]$, conforme os cálculos realizados a seguir.

$$\frac{s_0 + s_4}{2} = s_0^{(3-3)} \qquad \frac{s_0 - s_4}{2} = s_4^{(3-3)}$$

Substituindo os valores numéricos dos dois coeficientes de aproximação do sinal transformado $s_0^{(3-3)}$ e $s_4^{(3-3)}$, resolve-se o sistema linear nas variáveis s_0 e s_4 para se obter os novos coeficientes de aproximação, que serão utilizados na sequência dos procedimentos para a obtenção do sinal original.

$$\begin{aligned} \frac{s_0 + s_4}{2} &= -0,05 & \frac{s_0 - s_4}{2} &= 0,05 \\ \Rightarrow \begin{cases} s_0 + s_4 = -0,1 \\ s_0 - s_4 = 0,1 \end{cases} &\Rightarrow s_0 = 0; \quad s_4 = -0,1 \end{aligned}$$

Com os novos valores para s_0 e s_4 a primeira iteração tem como resultante o vetor $s^{(3-2)}$ obtido em (A.3):

$$s^{(3-2)} = [0 \quad 0,05 \quad 0,45 \quad 0,25 \quad -0,1 \quad -0,65 \quad -0,05 \quad -0,05] \quad (\text{A.6})$$

Na Figura A5 apresentam-se duas curvas poligonais, sendo em azul a curva que representa os coeficientes do sinal original proposto $s(3)$ no domínio do tempo e em vermelho a curva que representa os coeficientes do vetor $s^{(3-2)}$, obtidos usando a transformada *wavelet* inversa, calculados na primeira iteração.

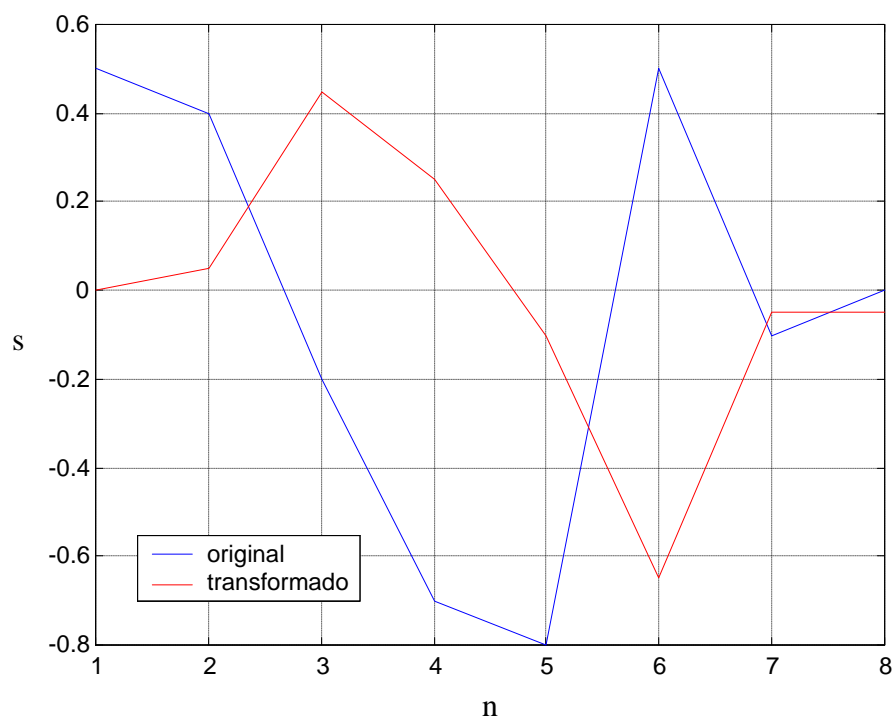


Figura A5 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $s^{(3-2)}$, resultante do primeiro processamento da reconstrução do sinal.

Segundo Passo: Reconstrução do sinal $s^{(3-1)}$

Na segunda iteração, o processo de reconstrução é feito executando as quatro operações algébricas apresentadas abaixo. Operações essas para obter quatro novos coeficientes de aproximação s_0 , s_2 , s_4 e s_6 que serão os novos coeficientes de aproximação a serem utilizados no terceiro passo da reconstrução usando a transformada *wavelet* inversa. Para esse fim, faz-se uso dos coeficientes de aproximação do sinal $s^{(3-2)}$, obtidos na iteração anterior, conforme as operações algébricas:

$$\frac{s_0 + s_2}{2} = s_0^{(3-2)} \quad \frac{s_0 - s_2}{2} = s_2^{(3-2)} \quad \frac{s_4 + s_6}{2} = s_4^{(3-2)} \quad \frac{s_4 - s_6}{2} = s_6^{(3-2)}$$

Substituindo os valores numéricos dos quatro coeficientes de aproximação do sinal transformado, resultante da primeira iteração, $s_0^{(3-2)}$, $s_2^{(3-2)}$, $s_4^{(3-2)}$ e $s_6^{(3-2)}$, resolve-se o sistema linear e obtêm-se os novos coeficientes de aproximação s_0 , s_2 , s_4 e s_6 que serão utilizados na próxima iteração para a obtenção do sinal original. Tem-se

$$\frac{s_0 + s_2}{2} = 0 \quad \frac{s_0 - s_2}{2} = 0,45 \quad \frac{s_4 + s_6}{2} = -0,1 \quad \frac{s_4 - s_6}{2} = -0,05$$

Das duas primeiras:

$$\begin{cases} s_0 + s_2 = 0 \\ s_0 - s_2 = 0,9 \end{cases} \Rightarrow s_0 = 0,45; \quad s_2 = -0,45$$

Das duas últimas:

$$\begin{cases} s_4 + s_6 = -0,2 \\ s_4 - s_6 = -0,1 \end{cases} \Rightarrow s_4 = -0,15; \quad s_6 = -0,05$$

Com os novos valores para s_0 , s_2 , s_4 e s_6 a segunda iteração tem como resultante, o vetor $s^{(3-1)}$ obtido em (A.2):

$$s^{(3-1)} = [0,45 \quad 0,05 \quad -0,45 \quad 0,25 \quad -0,15 \quad -0,65 \quad -0,05 \quad -0,05] \quad (\text{A.7})$$

Na Figura A6 apresentam-se duas curvas poligonais, sendo em azul a curva que representa os coeficientes do sinal original proposto $s(3)$ no domínio do tempo e em vermelho a curva que representa os coeficientes do vetor $s^{(3-1)}$ obtidos usando a transformada inversa efetuada na segunda iteração.

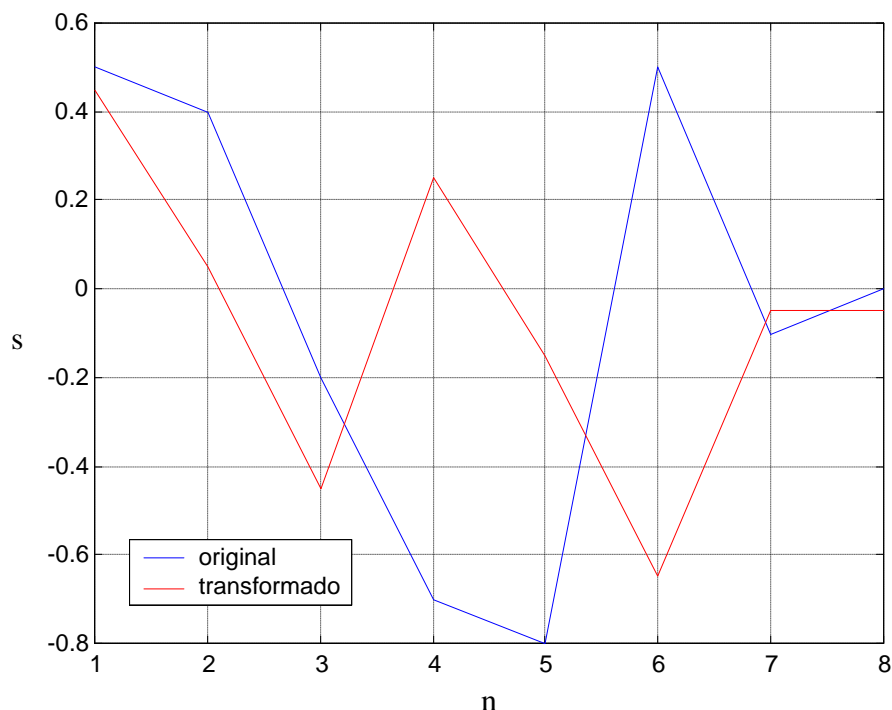


Figura A6 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $s^{(3-1)}$, resultante do segundo processamento da reconstrução do sinal.

Terceiro Passo: Reconstrução do sinal $s(3)$

Neste último procedimento, as operações são efetuadas nos coeficientes de aproximação, assim como, nos coeficientes de detalhes. Para obter esses novos coeficientes usam-se todos os coeficientes do sinal transformado $s^{(3-1)}$ obtidos no passo anterior, conforme expresso pelas equações algébricas descritas a seguir. Como o sinal original $s(3)$ adotado tem apenas oito coeficientes, esta iteração configura-se como sendo a última. Assim, o vetor $s(3)$ é a resultante da operação de reconstrução usando a transformada *wavelet* inversa e como consequência é a reconstrução do próprio sinal original no domínio do tempo apresentado no começo desta seção. A seguir são apresentadas as operações algébricas que são usadas para a obtenção dos coeficientes do sinal original $s(3)$.

$$\begin{array}{cccc} \frac{s_0 + s_1}{2} = s_0^{(3-1)} & \frac{s_0 - s_1}{2} = s_1^{(3-1)} & \frac{s_2 + s_3}{2} = s_2^{(3-1)} & \frac{s_2 - s_3}{2} = s_3^{(3-1)} \\ \frac{s_4 + s_5}{2} = s_4^{(3-1)} & \frac{s_4 - s_5}{2} = s_5^{(3-1)} & \frac{s_6 + s_7}{2} = s_6^{(3-1)} & \frac{s_6 - s_7}{2} = s_7^{(3-1)} \end{array}$$

Substituindo os valores numéricos dos coeficientes do sinal transformado resultante da segunda iteração, resolve-se o sistema linear e obtêm-se os novos coeficientes do sinal original $s(3)$.

$$\begin{array}{cccc} \frac{s_0 + s_1}{2} = 0,45 & \frac{s_0 - s_1}{2} = 0,05 & \frac{s_2 + s_3}{2} = -0,45 & \frac{s_2 - s_3}{2} = 0,25 \\ \frac{s_4 + s_5}{2} = -0,15 & \frac{s_4 - s_5}{2} = -0,65 & \frac{s_6 + s_7}{2} = -0,05 & \frac{s_6 - s_7}{2} = -0,05 \end{array}$$

$$\begin{cases} s_0 + s_1 = 0,9 \\ s_0 - s_1 = 0,1 \end{cases} \Rightarrow s_0 = 0,5; \quad s_1 = 0,4$$

$$\begin{cases} s_2 + s_3 = -0,9 \\ s_2 - s_3 = 0,5 \end{cases} \Rightarrow s_2 = -0,2; \quad s_3 = -0,7$$

$$\begin{cases} s_4 + s_5 = -0,3 \\ s_4 - s_5 = -1,3 \end{cases} \Rightarrow s_4 = -0,8; \quad s_5 = 0,5$$

$$\begin{cases} s_6 + s_7 = -0,1 \\ s_6 - s_7 = -0,1 \end{cases} \Rightarrow s_6 = -0,1; \quad s_7 = 0$$

Com os novos coeficientes calculados, o vetor resultante $s(3)$ é o sinal reconstruído usando a transformada *wavelet* inversa, ou seja, o processo descrito anteriormente, nos três passos, é o processo de reconstrução do sinal original $s(3)$ no domínio do tempo, proposto no começo deste apêndice.

$$s(3) = [0,5 \quad 0,4 \quad -0,2 \quad -0,7 \quad -0,8 \quad 0,5 \quad -0,1 \quad 0] \quad (\text{A.8})$$

Na sequência a Figura A7, apresenta a curva poligonal resultante da transformada *wavelet* inversa, ou seja, a mesma curva apresentada na Figura A1 que representa o sinal original $s(3)$ proposto no início desta seção.

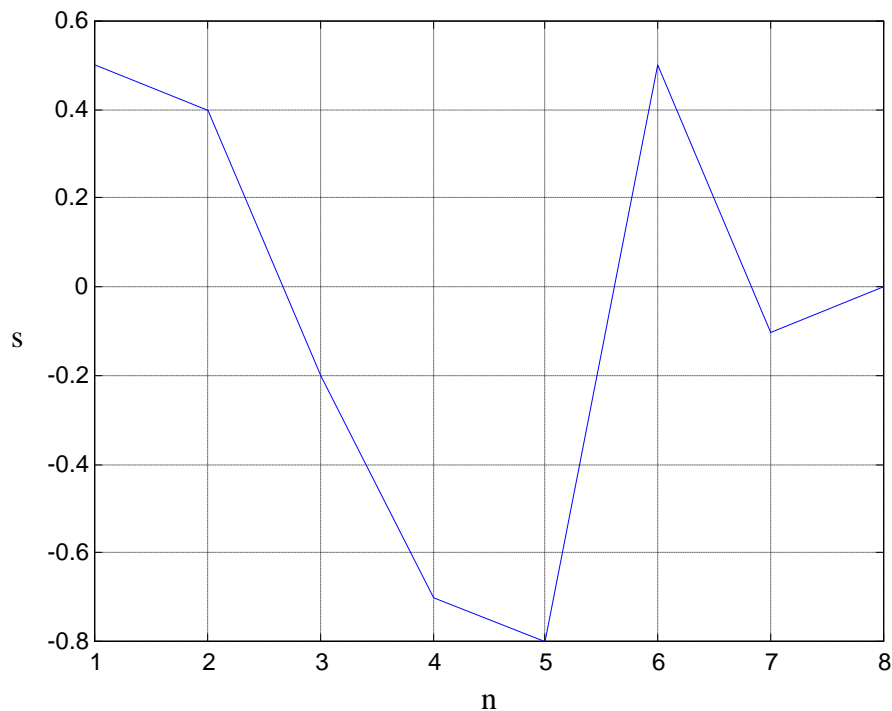


Figura A7 – Representação dos coeficientes do sinal reconstruído $s(3)$ usando a transformada *wavelet* inversa.

Entre o último passo da decomposição e o início da reconstrução pode ser feito o processamento do sinal de acordo com os objetivos desejados, por exemplo, compressão ou redução de ruído.

Se, por exemplo, o objetivo fosse a compressão, antes de aplicar a transformada *wavelet* inversa em $s^{(3-3)}$, poderíamos eliminar os coeficientes com menores valores absolutos e, neste caso teríamos o seguinte vetor:

$$s^{(3-3)} = [0 \ 0 \ 0,45 \ 0,25 \ 0 \ -0,65 \ 0 \ 0].$$

A reconstrução desse sinal, usando o processo da transformada *wavelet* inversa mostrado anteriormente, resultaria em:

$$\hat{s}[3] = [0,45 \ 0,45 \ -0,2 \ -0,7 \ -0,65 \ 0,65 \ 0 \ 0]$$

que é uma boa aproximação de $s(3)$, como se pode observar na Figura A8.

Assim, para uma reconstrução aproximada do sinal, foram necessários apenas três coeficientes no domínio transformado, ou seja, foram usados apenas aqueles que tinham maior valor absoluto.

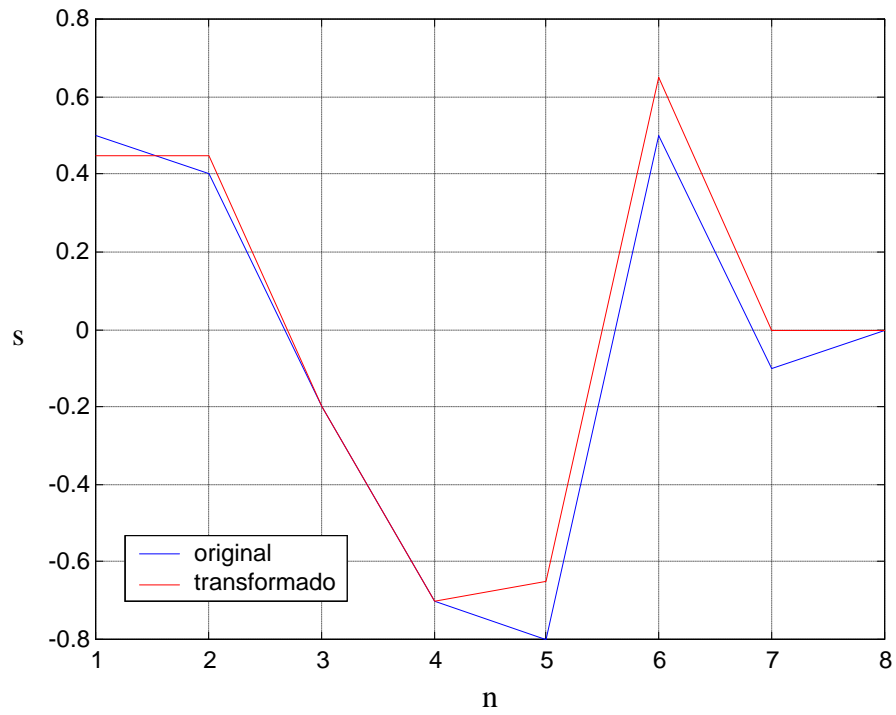


Figura A8 – Representação da compressão dos coeficientes do sinal original $s(3)$.

APÊNDICE B

B1. O OPERADOR OMS E A TRANSFORMADA WAVELET

Este apêndice tem o intuito de facilitar a compreensão da execução dos processamentos denominados aqui como operadores. O objetivo é fazer com que o leitor entenda os procedimentos básicos desta operação. Para ilustrar este procedimento, considere um sinal ruidoso $S[n]$, no domínio *wavelet*, tendo 2^n elementos

$$S[n] = \left\{ s_{2k}, s_{2k+1} : 0 \leq k \leq \frac{2^n - 2}{2} \right\} = \{ s_0, s_1, s_2, s_3, \dots, s_{2k}, s_{2k+1}, \dots, s_{2^n-2}, s_{2^n-1} \} \quad (\text{B.1})$$

e o primeiro processamento denominado como Operador Média Simples (OMS), apresentado na seção 5.2.1, descrito conforme a seguinte equação:

$$f_i = \frac{|s_i| + |s_{i+1}|}{2}, \text{ para } i=0, \dots, n-1 \text{ e } f_n = f_{n-1} \quad (\text{B.2})$$

No que se segue, o sinal ruidoso $S[n]$ será denotado por:

$$S[n] = \left[s_{2k}, s_{2k+1} : 0 \leq k \leq \frac{2^n - 2}{2} \right] \quad (\text{B.3})$$

considerando esse sinal como um vetor em R^N , em que $N=2^n$.

Para ilustrar o processo, considere o sinal ruidoso (com $n = 3$), sendo o vetor apresentado anteriormente em (A.5):

$$S[3] = [-0,05 \quad 0,05 \quad 0,45 \quad 0,25 \quad 0,05 \quad -0,65 \quad -0,05 \quad -0,05] \quad (\text{B.4})$$

Observar que o sinal $S[3]$ tem uma amplitude absoluta de 0,65 unidades de medida.

Este vetor é somente uma ilustração com apenas oito coeficientes, para explicar a aplicação do operador OMS de uma forma simples, pois os sinais que são analisados, em geral, contêm milhares de coeficientes. Na Figura B1 é apresentada uma representação dos coeficientes do sinal $S[3]$ considerado acima. Essa representação é a curva poligonal passando pelos pontos $(i + 1, s_i)$, para $i=0, \dots, 7$, que também será denominada de sinal $S[3]$.

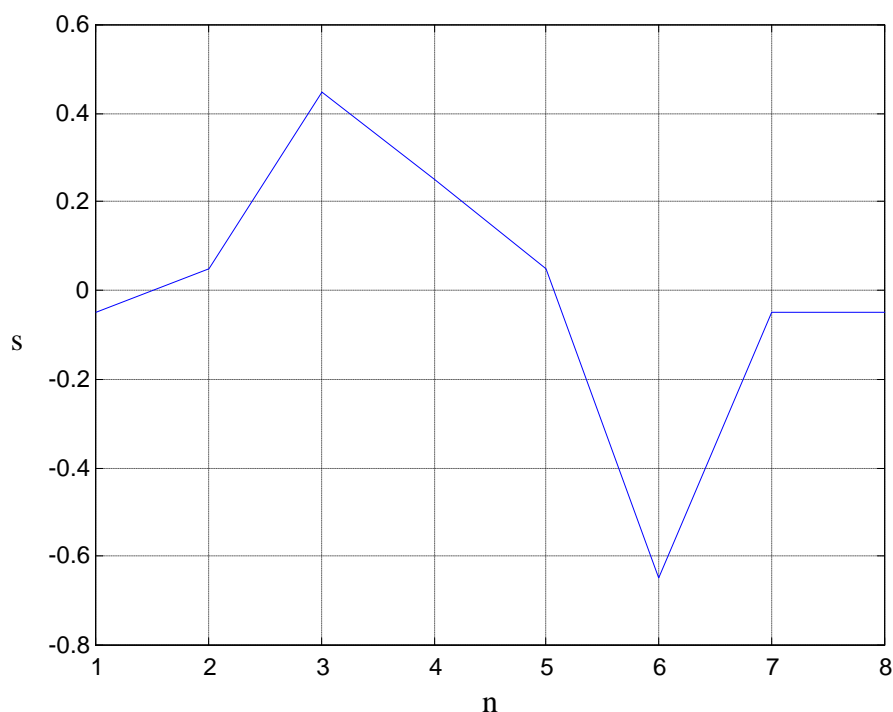


Figura B1 – Representação dos coeficientes do sinal ruidoso $S[3]$.

O gráfico do operador OMS, dado pela equação (B.2), é a curva poligonal (passando pelos pontos $(i + 1, f_i)$) no qual cada elemento f_i é obtido pela média dos respectivos valores absolutos de cada duas amostras consecutivas do sinal $S[n]$, conforme a equação (B.2). O conjunto dos elementos f_i obtidos usando a equação (B.2), no sinal $S[n]$, será indicado por $F[n]$; assim,

$$F[n] = \{f_0, f_1, f_2, \dots, f_n\} \quad (\text{B.5})$$

Os elementos do conjunto $F[3]$, associado ao sinal $S[3]$, são calculados a seguir. O gráfico do operador OMS, associado a $F[3]$, é apresentado na Figura B2.

$$\begin{aligned}
 f_0 &= \frac{|s_0| + |s_1|}{2} \Rightarrow f_0 = \frac{|-0,05| + |0,05|}{2} \Rightarrow f_0 = 0,05 \\
 f_1 &= \frac{|s_1| + |s_2|}{2} \Rightarrow f_1 = \frac{|0,05| + |0,45|}{2} \Rightarrow f_1 = 0,25 \\
 f_2 &= \frac{|s_2| + |s_3|}{2} \Rightarrow f_2 = \frac{|0,45| + |0,25|}{2} \Rightarrow f_2 = 0,35 \\
 f_3 &= \frac{|s_3| + |s_4|}{2} \Rightarrow f_3 = \frac{|0,25| + |0,05|}{2} \Rightarrow f_3 = 0,15 \\
 f_4 &= \frac{|s_4| + |s_5|}{2} \Rightarrow f_4 = \frac{|0,05| + |-0,65|}{2} \Rightarrow f_4 = 0,35 \\
 f_5 &= \frac{|s_5| + |s_6|}{2} \Rightarrow f_5 = \frac{|-0,65| + |-0,05|}{2} \Rightarrow f_5 = 0,35 \\
 f_6 &= \frac{|s_6| + |s_7|}{2} \Rightarrow f_6 = \frac{|-0,05| + |-0,05|}{2} \Rightarrow f_6 = 0,05 \\
 f_7 &= \frac{|s_7| + |s_6|}{2} \Rightarrow f_7 = \frac{|-0,05| + |-0,05|}{2} \Rightarrow f_7 = 0,05
 \end{aligned}$$

Com os cálculos efetuados, os elementos do conjunto $F[3]$, utilizando a equação (B.2), de acordo com a equação (B.5) é o conjunto dado por:

$$F[3] = \{0,05 \quad 0,25 \quad 0,35 \quad 0,15 \quad 0,35 \quad 0,35 \quad 0,05 \quad 0,05\} \quad (\text{B.5'})$$

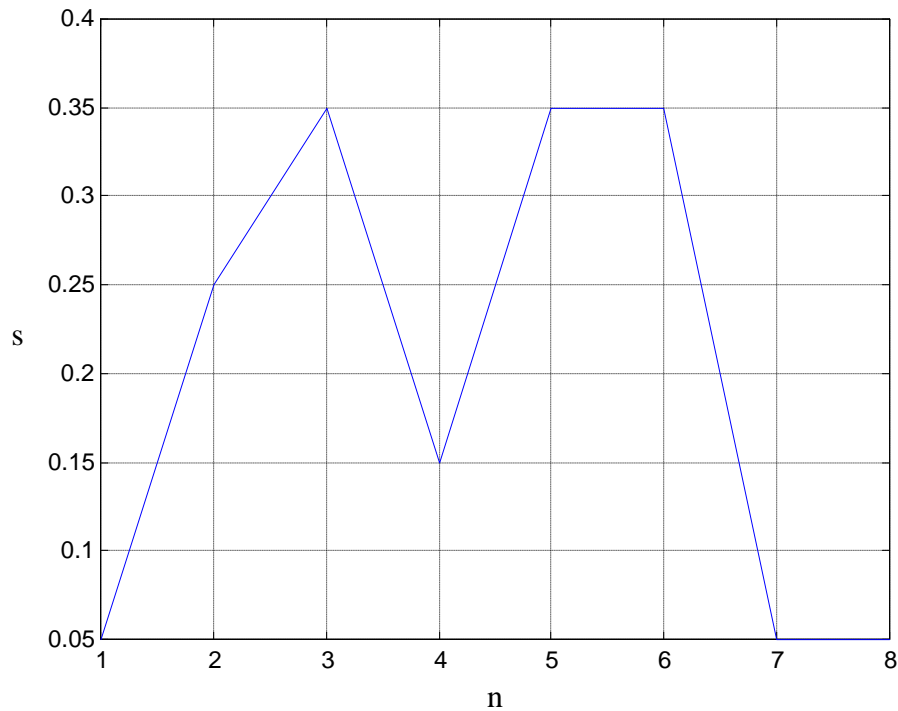


Figura B2 – Representação dos coeficientes de $F[3]$ após aplicação do Operador OMS ao sinal $S[3]$.

Agora, serão calculados os coeficientes $f_i, i=0, \dots, n$, sem o uso do valor absoluto em (B.2), ou seja, usando a fórmula

$$f_i = \frac{s_i + s_{i+1}}{2} \quad (\text{B.6})$$

Os cálculos efetuados usando a equação (B.6) são apresentados a seguir e a curva poligonal correspondente obtida, usando esses dados, é apresentada na Figura B3.

$$\begin{aligned}
 f_0 &= \frac{s_0 + s_1}{2} & \Rightarrow & \quad f_0 = \frac{-0,05 + 0,05}{2} & \Rightarrow & \quad f_0 = 0 \\
 f_1 &= \frac{s_1 + s_2}{2} & \Rightarrow & \quad f_1 = \frac{0,05 + 0,45}{2} & \Rightarrow & \quad f_1 = 0,25 \\
 f_2 &= \frac{s_2 + s_3}{2} & \Rightarrow & \quad f_2 = \frac{0,45 + 0,25}{2} & \Rightarrow & \quad f_2 = 0,35 \\
 f_3 &= \frac{s_3 + s_4}{2} & \Rightarrow & \quad f_3 = \frac{0,25 + 0,05}{2} & \Rightarrow & \quad f_3 = 0,15 \\
 f_4 &= \frac{s_4 + s_5}{2} & \Rightarrow & \quad f_4 = \frac{0,05 + (-0,65)}{2} & \Rightarrow & \quad f_4 = -0,3 \\
 f_5 &= \frac{s_5 + s_6}{2} & \Rightarrow & \quad f_5 = \frac{(-0,65) + (-0,05)}{2} & \Rightarrow & \quad f_5 = -0,35
 \end{aligned}$$

$$f_6 = \frac{s_6 + s_7}{2} \Rightarrow f_6 = \frac{(-0,05) + (-0,05)}{2} \Rightarrow f_6 = -0,05$$

$$f_7 = \frac{s_7 + s_6}{2} \Rightarrow f_7 = \frac{(-0,05) + (-0,05)}{2} \Rightarrow f_7 = -0,05$$

Com os cálculos efetuados, os elementos do conjunto que será denotado por $F'[3]$ utilizando a equação (B.6) são dados por:

$$F'[3] = \{0 \quad 0,25 \quad 0,35 \quad 0,15 \quad -0,3 \quad -0,35 \quad -0,05 \quad -0,05\} \quad (\text{B.6'})$$

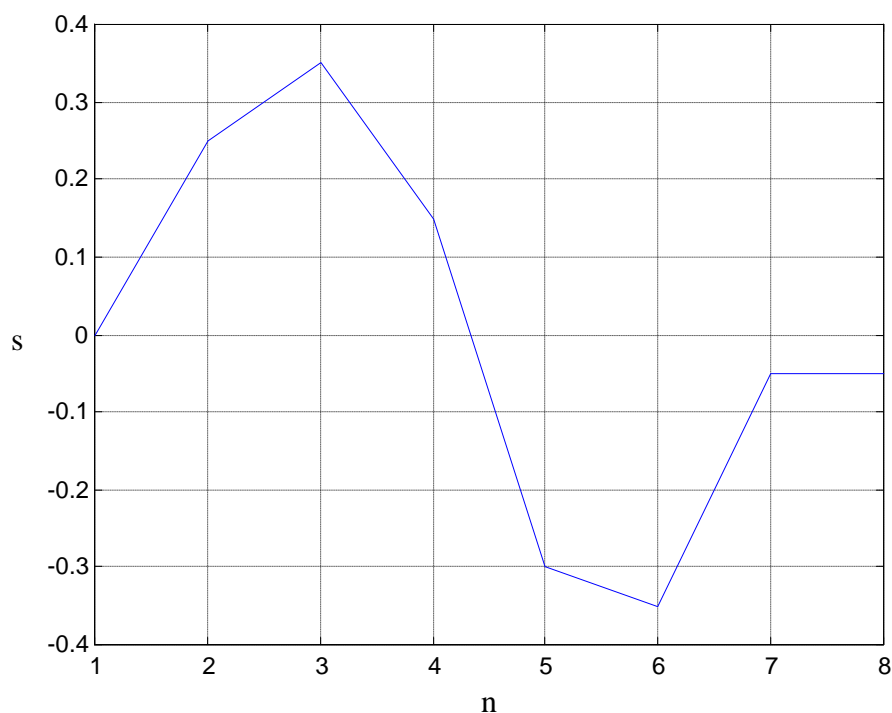


Figura B3 – Representação dos coeficientes de $F'[3]$ após aplicação do operador OMS ao sinal $S[3]$, sem utilização do valor absoluto.

O uso do operador OMS dada pela equação (B.2) se justifica pelo fato de que o produto entre os coeficientes do sinal original $S[3]$ pelos coeficientes de $F[3]$ apenas reduzirá o sinal processado, ao passo, que quando se faz o produto dos coeficientes do sinal $S[3]$ pelos coeficientes de $F'[3]$, além de reduzir o sinal processado, a fase de alguns coeficientes do sinal processado ficará com a sua fase invertida, e isto causará distorções no sinal processado.

Na Figura B4, para melhor comparação entre os sinais processados, apresentam-se as três curvas poligonais vistas nas figuras anteriores utilizando o sinal ruidoso original $S[3]$.

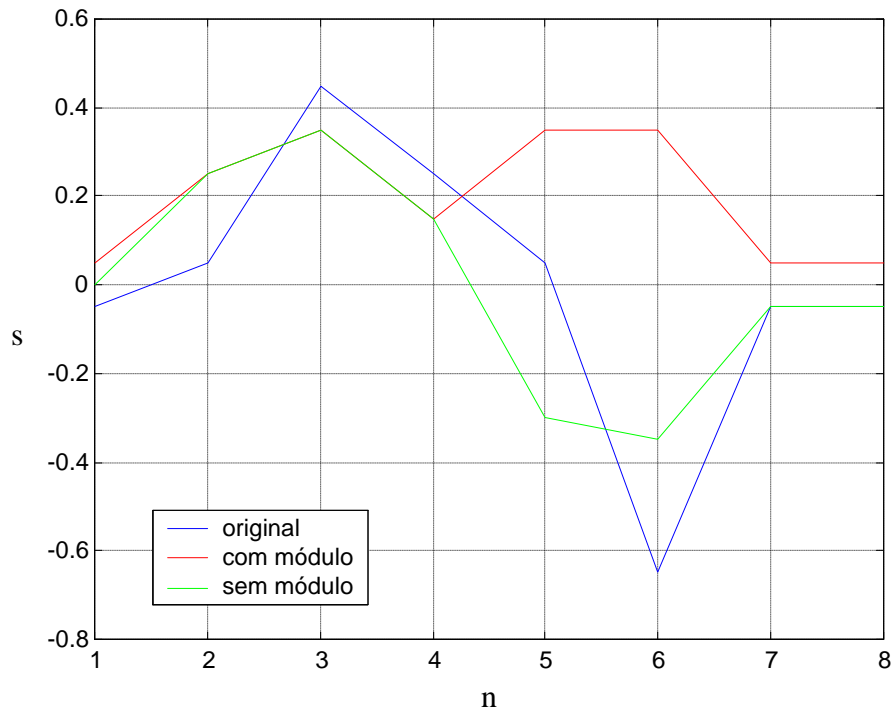


Figura B4 – Representação dos coeficientes de $S[3]$, $F[3]$ e $F'[3]$.

Observando a Figura B4, pode-se ver que a curva vermelha, cujos coeficientes são calculados usando o módulo, está totalmente acima do semiplano $\{(n, s): s > 0\}$ enquanto que a curva verde, que é calculada sem a utilização do módulo, tem partes abaixo do semiplano $\{(n, s): s > 0\}$. Se a curva verde fosse usada para a redução de ruído proposta, haveria inversão de coeficientes e, conseqüentemente, introdução de distorções no sinal. Esse fato justifica o uso do valor absoluto no cálculo do operador OMS.

A redução de ruído em sinais de voz proposta no domínio *wavelet*, para este processamento, é feita através da simples multiplicação dos respectivos elementos do sinal ruidoso, indicado pelo conjunto $S[n]$ dado em (B.1), pelos correspondentes elementos do operador OMS, indicado por $F[n]$ conforme a equação (B.5). Assim, tem-se o conjunto resultante:

$$Y[n] = \{s_0 \times f_0, s_1 \times f_1, s_2 \times f_2, \dots, s_n \times f_n\} \quad (\text{B.7})$$

Fazendo $y_i = s_i \times f_i$, a equação (B.7) pode ser escrita como:

$$Y[n] = \{y_0, y_1, y_2, \dots, y_n\} \quad (\text{B.8})$$

Os elementos do conjunto $Y[n]$ são coeficientes resultantes da caracterização da operação de redução de ruído.

Para o processamento denominado como OMS, os elementos do conjunto resultante $Y[3]$, usando $S[3]$ e $F[3]$ conforme a equação (B.8) são calculados a seguir.

$$\begin{array}{lll}
 y_0 = s_0 \times f_0 : & y_0 = -0,05 \times 0,05 & \Rightarrow y_0 = -0,0025 \\
 y_1 = s_1 \times f_1 : & y_1 = 0,05 \times 0,25 & \Rightarrow y_1 = 0,0125 \\
 y_2 = s_2 \times f_2 : & y_2 = 0,45 \times 0,35 & \Rightarrow y_2 = 0,1575 \\
 y_3 = s_3 \times f_3 : & y_3 = 0,25 \times 0,15 & \Rightarrow y_3 = 0,0375 \\
 y_4 = s_4 \times f_4 : & y_4 = 0,05 \times 0,35 & \Rightarrow y_4 = 0,0175 \\
 y_5 = s_5 \times f_5 : & y_5 = -0,65 \times 0,35 & \Rightarrow y_5 = -0,2275 \\
 y_6 = s_6 \times f_6 : & y_6 = -0,05 \times 0,05 & \Rightarrow y_6 = -0,0025 \\
 y_7 = s_7 \times f_7 : & y_7 = -0,05 \times 0,05 & \Rightarrow y_7 = -0,0025
 \end{array}$$

Assim, o sinal reduzido $Y[3]$ é dado por:

$$Y[3] = \{-0,0025 \ 0,0125 \ 0,1575 \ 0,0375 \ 0,0175 \ -0,2275 \ -0,0025 \ -0,0025\} \quad (\text{B.9})$$

Observar que o sinal reduzido $Y[3]$ tem amplitude absoluta de 0,2275 unidades de medida.

A curva poligonal passando pelos pontos $(i + 1, y_i)$ é apresentada na Figura B5.

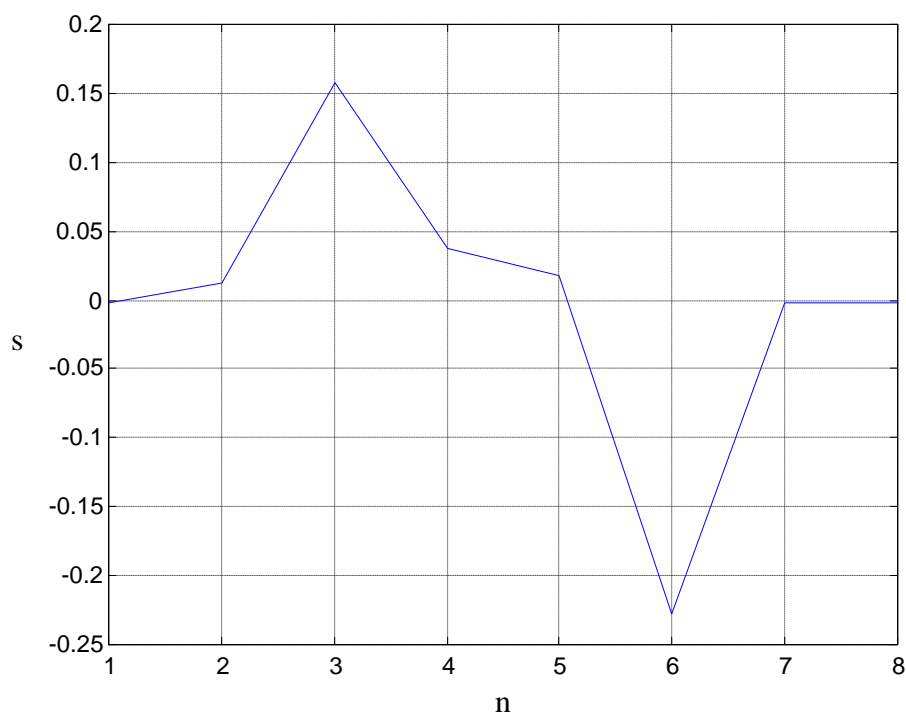


Figura B5 – Resultante da operação $y_i = s_i \cdot f_i$: representação dos coeficientes do sinal $Y[3]$.

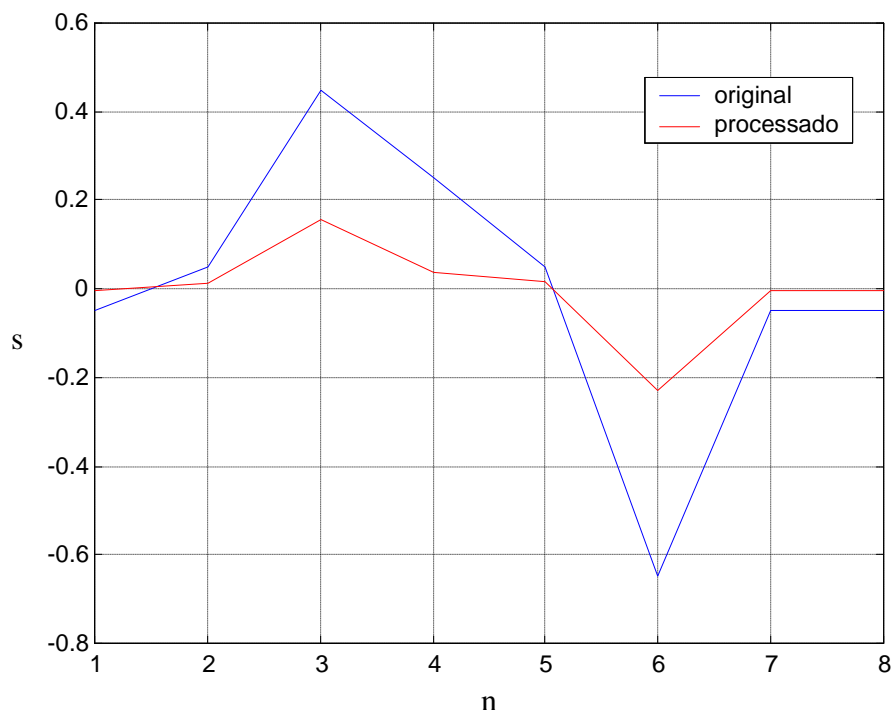


Figura B6 – Comparação entre as curvas (sinal original $S[3]$ e sinal processado $Y[3]$).

Analisando as duas curvas poligonais da Figura B6, apresentadas em B1 e B5, pode-se notar que, quando se analisa os coeficientes do sinal ruidoso original (em azul: Figura B1) e os coeficientes resultantes da redução de ruído através do operador OMS (em vermelho: Figura B5), esses coeficientes sofrem uma redução de 0,4225 ($0,65 - 0,2275$) unidades de medida no valor absoluto de suas amplitudes, havendo assim redução de ruído.

Como se observa na Figura B6 há uma diferença considerável entre os sinais original e processado, isto se deve ao fato de o sinal original ter amplitude no intervalo $[-1,1]$, desta forma o vetor $F[3]$ tem uma amplitude menor que o sinal original fazendo com que no produto de $F[3]$ por $S[3]$ a redução na amplitude do sinal processado seja forte. Essa redução em parte se deve ao fato de se tratar de um sinal sem ruído. Como não há componentes ruidosos o operador reduz coeficientes significativos do sinal original.

Observar que o processo de decomposição do sinal $s(n)$, no domínio do tempo, para o sinal $S[n]$, no domínio *wavelet* ou domínio transformado, é feito usando a transformada *wavelet*. Já o processo de reconstrução do sinal $s(n)$ a partir do sinal $S[n]$, é feito utilizando a transformada *wavelet* inversa. Esses processos de

decomposição e de reconstrução foram apresentados no Apêndice A, aplicados a um caso particular do sinal $s(3)$, dado em (A.1).

Por outro lado, no Apêndice B, foi processado o sinal $S[n]$, considerado no domínio transformado, para o sinal $Y[n]$, considerado também no domínio transformado. Esse processamento é feito usando o denominado operador OMS, definido pela equação (B.2). A execução do processamento foi exemplificado usando o sinal $S[3]$, dado em (B.4), para a obtenção do sinal $Y[3]$, dado em (B.9).

Agora deve-se decompor o sinal $Y[n]$, obtido via operador OMS e considerado no domínio transformado, para um sinal que será denotado por $\hat{y}(n)$ no domínio do tempo. Essa decomposição é feita usando a transformada *wavelet* inversa, seguindo os passos da reconstrução do sinal $s(n)$ a partir do sinal $S[n]$, conforme apresentados no Apêndice A. Mais especificamente, deve-se aplicar a transformada *wavelet* inversa de *Haar* ao sinal $Y[n]$ para calcular os sinais indicados por $\hat{y}_{(n-k)}$, para $k=1, \dots, n$, seguindo a sequência de passos

$$Y[n] \mapsto \hat{y}_{(n-1)} \mapsto \hat{y}_{(n-2)} \mapsto \dots \mapsto \hat{y}_{(n-n)}.$$

O sinal $\hat{y}_{(n-n)}$, obtido no último passo de decomposição, é indicado por $\hat{y}(n)$.

Usando o processo descrito acima, para exemplificar, será decomposto o sinal $Y[3]$, dado em (B.9), para a obtenção do sinal $\hat{y}(3)$, seguindo a seguinte sequência de passos

$$Y[3] \mapsto \hat{y}_{(3-1)} \mapsto \hat{y}_{(3-2)} \mapsto \hat{y}_{(3-3)} =: \hat{y}(3).$$

Esse processo de decomposição é apresentado no Apêndice C.

APÊNDICE C

C1. PROCESSO DE RECONSTRUÇÃO USANDO A TRANSFORMADA WAVELET INVERSA NO SINAL RUIDOSO

Para ilustrar as operações básicas da transformada *wavelet* inversa no sinal ruidoso, será utilizado o vetor no domínio transformado $Y[3]$ apresentado no Apêndice B. O processamento do vetor $Y[3]$ consiste em efetuar a transformada *wavelet* inversa (operações efetuadas no Apêndice A), obtendo um novo sinal no domínio do tempo, que será denotado como $\hat{Y}(3)$. Para a análise destes procedimentos serão efetuados os cálculos em cada nível de resolução, sendo ainda apresentadas as curvas poligonais do sinal original no domínio do tempo $s(3)$, apresentados no Apêndice A, e a reconstrução do sinal em cada fase do processamento para um melhor entendimento.

Nesta parte, usando a transformada *wavelet* inversa (vide Apêndice A), é realizada a operação de reconstrução do sinal $Y[3]$, apresentada no Apêndice B. Abaixo é apresentado o vetor $Y[3]$ obtido no Apêndice B.

$$Y[3] = [-0,0025 \quad 0,0125 \quad 0,1575 \quad 0,0375 \quad 0,0175 \quad -0,2275 \quad -0,0025 \quad -0,0025]$$

O processo de reconstrução do sinal é feito seguindo os passos apresentados no Apêndice A, conforme descritos a seguir. No caso particular, devem-se obter os coeficientes do vetor $\hat{Y}(3)$ a partir de $Y[3]$. Para a análise destes procedimentos serão

efetuados os cálculos em cada nível de resolução, sendo ainda, apresentadas as curvas poligonais, que representam os coeficientes do sinal reconstruído, em cada fase do processamento. Seguem então os procedimentos para efetuar a operação de reconstrução do sinal, usando a transformada *wavelet* inversa (vide Apêndice A).

Primeiro Passo: Cálculo do sinal $\hat{y}_{(3-1)}$

A operação de reconstrução do sinal ruidoso transformado para o domínio do tempo na primeira iteração é feita executando as duas operações algébricas descritas abaixo. O processo consiste na obtenção de dois novos coeficientes de aproximação s_0 e s_4 , que serão os novos coeficientes de aproximação, utilizados na segunda iteração do processo. Para esse fim, faz-se uso dos coeficientes de aproximação do sinal ruidoso transformado $Y[3]$, conforme os cálculos realizados a seguir.

$$\frac{s_0 + s_4}{2} = s_0^{(3-3)} \qquad \frac{s_0 - s_4}{2} = s_4^{(3-3)}$$

Substituindo os valores numéricos dos dois coeficientes de aproximação do sinal ruidoso transformado $s_0^{(3-3)}$ e $s_4^{(3-3)}$, resolve-se o sistema linear nas variáveis s_0 e s_4 para se obter os novos coeficientes de aproximação, que serão utilizados na sequência dos procedimentos para a obtenção do sinal ruidoso no domínio do tempo.

$$\frac{s_0 + s_4}{2} = -0,0025 \qquad \frac{s_0 - s_4}{2} = 0,0175$$

$$\begin{cases} s_0 + s_4 = -0,005 \\ s_0 - s_4 = 0,035 \end{cases} \quad \Rightarrow \quad s_0 = 0,015; \quad s_4 = -0,02$$

Com os novos valores para s_0 e s_4 a primeira iteração tem como resultante o vetor $\hat{y}_{(3-1)}$:

$$\hat{y}_{(3-1)} = [0,015 \quad 0,0125 \quad 0,1575 \quad 0,0375 \quad -0,02 \quad -0,2275 \quad -0,0025 \quad -0,0025]$$

Na Figura C1 apresentam-se duas curvas poligonais, sendo em azul a curva que representa os coeficientes do sinal original proposto $s(3)$ no domínio do tempo (vide Apêndice A) e em vermelho a curva que representa os coeficientes do vetor $\hat{y}_{(3-1)}$, obtidos usando a transformada *wavelet* inversa no sinal ruidoso, calculados na primeira iteração.

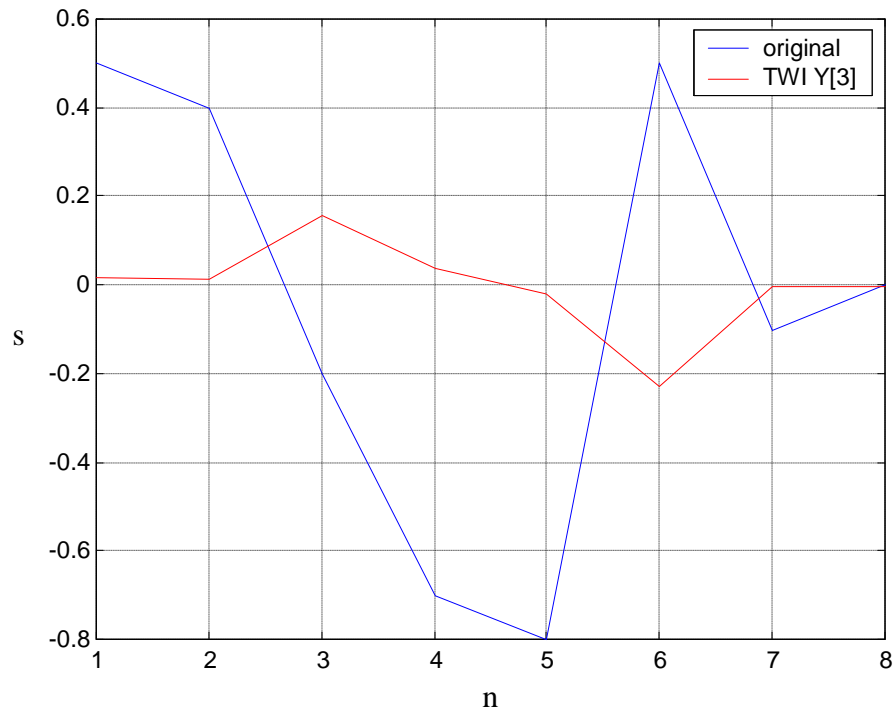


Figura C1 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $\hat{y}_{(3-1)}$, resultante do primeiro processamento da reconstrução do sinal ruidoso.

Segundo Passo: Cálculo do sinal $\hat{y}_{(3-2)}$

Na segunda iteração, o processo de reconstrução é feito executando as quatro operações algébricas apresentadas abaixo, operações essas para obter quatro novos coeficientes de aproximação s_0 , s_2 , s_4 e s_6 que serão os novos coeficientes de aproximação a serem utilizados no terceiro passo da reconstrução usando a transformada *wavelet* inversa. Para esse fim, faz-se uso dos coeficientes de aproximação do sinal $\hat{y}_{(3-1)}$, obtidos na iteração anterior, conforme as operações algébricas:

$$\frac{s_0 + s_2}{2} = s_0^{(3-2)} \quad \frac{s_0 - s_2}{2} = s_2^{(3-2)} \quad \frac{s_4 + s_6}{2} = s_4^{(3-2)} \quad \frac{s_4 - s_6}{2} = s_6^{(3-2)}$$

Substituindo os valores numéricos dos quatro coeficientes de aproximação do sinal ruidoso transformado, resultante da primeira iteração, $s_0^{(3-2)}$, $s_2^{(3-2)}$, $s_4^{(3-2)}$ e $s_6^{(3-2)}$, resolve-se o sistema linear e obtêm-se os novos coeficientes de aproximação s_0 , s_2 , s_4 e s_6 que serão utilizados na próxima iteração para a obtenção do sinal ruidoso processado no domínio do tempo. Tem-se

$$\begin{array}{cccc} \frac{s_0 + s_2}{2} = s_0^{(3-2)} & \frac{s_0 - s_2}{2} = s_2^{(3-2)} & \frac{s_4 + s_6}{2} = s_4^{(3-2)} & \frac{s_4 - s_6}{2} = s_6^{(3-2)} \\ \frac{s_0 + s_2}{2} = 0,015 & \frac{s_0 - s_2}{2} = 0,1575 & \frac{s_4 + s_6}{2} = -0,02 & \frac{s_4 - s_6}{2} = -0,0025 \end{array}$$

Das duas primeiras:

$$\begin{cases} s_0 + s_2 = 0,03 \\ s_0 - s_2 = 0,315 \end{cases} \Rightarrow s_0 = 0,1725; \quad s_2 = -0,1425$$

Das duas últimas:

$$\begin{cases} s_4 + s_6 = -0,04 \\ s_4 - s_6 = -0,005 \end{cases} \Rightarrow s_4 = -0,0225; \quad s_6 = -0,0175$$

Com os novos valores para s_0 , s_2 , s_4 e s_6 , a segunda iteração tem como resultante o vetor $\hat{y}_{(3-2)}$:

$$\hat{y}_{(3-2)} = [0,1725 \quad 0,0125 \quad -0,1425 \quad 0,0375 \quad -0,0225 \quad -0,2275 \quad -0,0175 \quad -0,0025]$$

Na Figura C2 apresentam-se duas curvas poligonais, sendo em azul a curva que representa os coeficientes do sinal original proposto $s(3)$ no domínio do tempo (vide Apêndice A) e em vermelho a curva que representa os coeficientes do vetor $\hat{y}_{(3-2)}$ obtidos usando a transformada inversa no sinal ruidoso efetuada na segunda iteração.

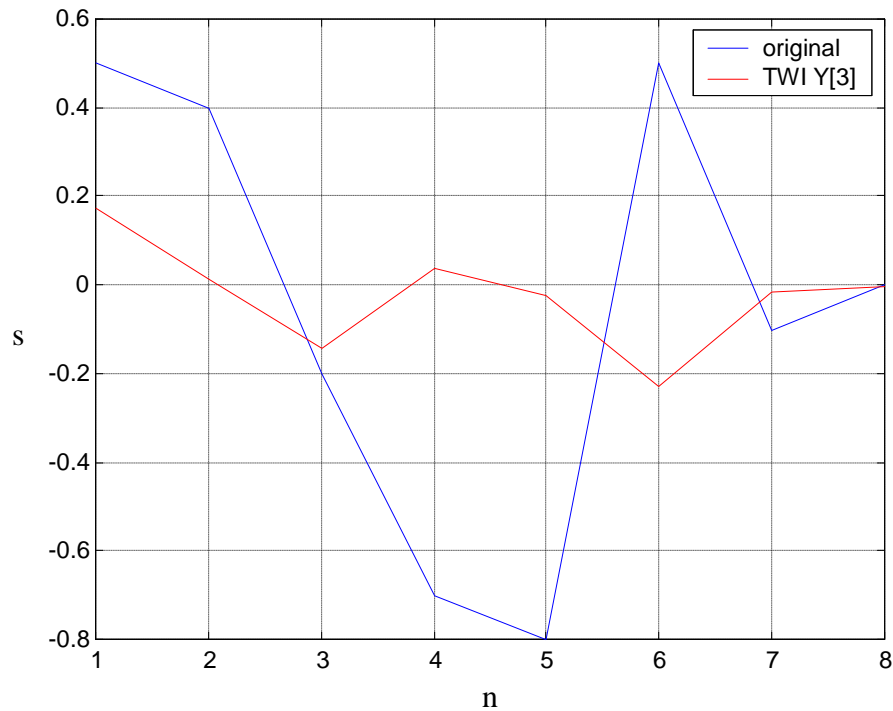


Figura C2 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $\hat{y}_{(3-2)}$, resultante do segundo processamento da reconstrução do sinal ruidoso.

Terceiro Passo: Cálculo do sinal $\hat{y}_{(3-3)} = \hat{y}(3)$

Neste último procedimento, as operações são efetuadas nos coeficientes de aproximação, assim como, nos coeficientes de detalhes. Para obter esses novos coeficientes usam-se todos os coeficientes do sinal transformado ruidoso $\hat{y}_{(3-2)}$ obtido no passo anterior, conforme expresso pelas equações algébricas descritas a seguir.

A seguir são apresentadas as operações algébricas que são usadas para a obtenção dos coeficientes do sinal ruidoso processado no domínio do tempo $\hat{y}(3)$.

$$\begin{array}{cccc} \frac{s_0 + s_1}{2} = s_0^{(3-1)} & \frac{s_0 - s_1}{2} = s_1^{(3-1)} & \frac{s_2 + s_3}{2} = s_2^{(3-1)} & \frac{s_2 - s_3}{2} = s_3^{(3-1)} \\ \frac{s_4 + s_5}{2} = s_4^{(3-1)} & \frac{s_4 - s_5}{2} = s_5^{(3-1)} & \frac{s_6 + s_7}{2} = s_6^{(3-1)} & \frac{s_6 - s_7}{2} = s_7^{(3-1)} \end{array}$$

Substituindo os valores numéricos dos coeficientes do sinal transformado resultante da segunda iteração, resolve-se o sistema linear e obtêm-se os novos coeficientes do sinal ruidoso processado no domínio do tempo $\hat{y}(3)$.

$$\begin{array}{cccc} \frac{s_0 + s_1}{2} = 0,1725 & \frac{s_0 - s_1}{2} = 0,0125 & \frac{s_2 + s_3}{2} = -0,1425 & \frac{s_2 - s_3}{2} = 0,0375 \end{array}$$

$$\frac{s_4 + s_5}{2} = -0,0225 \quad \frac{s_4 - s_5}{2} = -0,2275 \quad \frac{s_6 + s_7}{2} = -0,0175 \quad \frac{s_6 - s_7}{2} = -0,0025$$

$$\begin{cases} s_0 + s_1 = 0,345 \\ s_0 - s_1 = 0,025 \end{cases} \Rightarrow s_0 = 0,185; \quad s_1 = 0,16$$

$$\begin{cases} s_2 + s_3 = -0,285 \\ s_2 - s_3 = 0,075 \end{cases} \Rightarrow s_2 = -0,105; \quad s_3 = -0,18$$

$$\begin{cases} s_4 + s_5 = -0,045 \\ s_4 - s_5 = -0,455 \end{cases} \Rightarrow s_4 = -0,25; \quad s_5 = 0,205$$

$$\begin{cases} s_6 + s_7 = -0,035 \\ s_6 - s_7 = -0,005 \end{cases} \Rightarrow s_6 = -0,02; \quad s_7 = -0,015$$

Com os novos coeficientes calculados, o vetor resultante $\hat{y}(3)$ é o sinal ruidoso processado no domínio do tempo reconstruído usando a transformada *wavelet* inversa. Segue abaixo o sinal ruidoso processado no domínio do tempo:

$$\hat{y}(3) = [0,185 \quad 0,16 \quad -0,105 \quad -0,18 \quad -0,25 \quad 0,205 \quad -0,02 \quad -0,015]$$

Na Figura C3 apresentam-se duas curvas poligonais, sendo em azul a curva que representa os coeficientes do sinal original proposto $s(3)$ no domínio do tempo (vide Apêndice A) e em vermelho a curva que representa os coeficientes do vetor $\hat{y}(3)$ obtidos usando a transformada inversa no sinal ruidoso efetuada na terceira e última iteração.

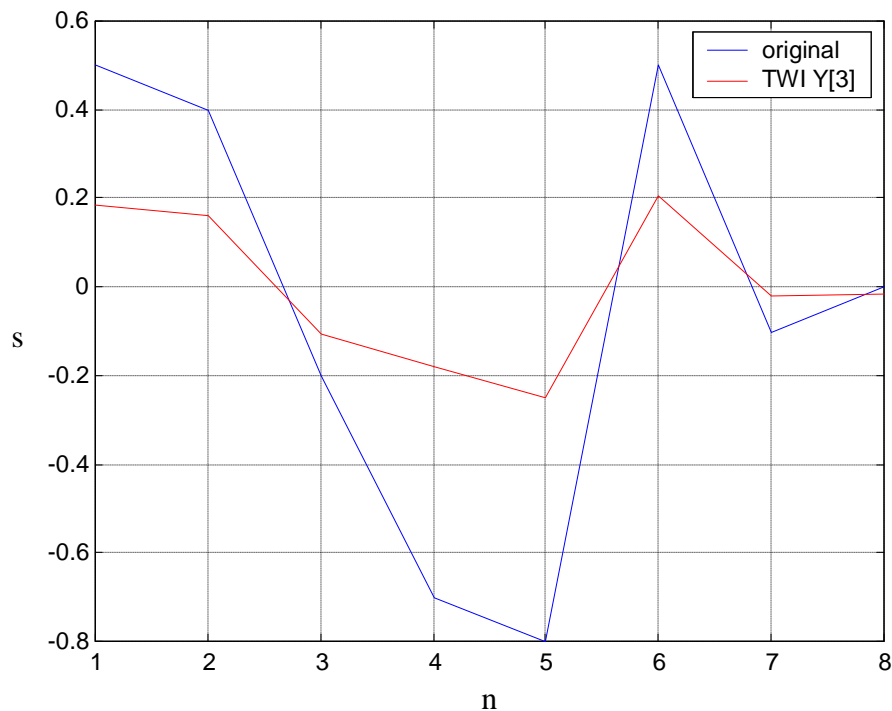


Figura C3 – Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $\hat{y}(3)$, resultante do terceiro processamento da reconstrução do sinal ruidoso.

Analisando as duas curvas poligonais da Figura C3, pode-se notar que, quando se analisa os coeficientes do sinal original (em azul: Figura A1) e os coeficientes resultantes da redução de ruído no sinal ruidoso (em vermelho: Figura C3), todos no domínio do tempo, esses coeficientes sofrem uma redução de 0,55 ($0,8 - 0,25$) unidades de medida no valor absoluto de suas amplitudes, havendo assim redução de ruído.

Como se observa na Figura C3, as duas curvas poligonais praticamente têm o mesmo formato, havendo uma diferença considerável entre suas amplitudes, fazendo com que a redução na amplitude do sinal processado seja forte. Consequentemente o sinal processado apresentará uma redução ou atenuação em seu ruído.

APÊNDICE D

D1. MÉTODO PROPOSTO

A metodologia de redução de ruído proposta neste trabalho consiste na aplicação de três operadores parciais e independentes. Para ilustrar as operações básicas da aplicação destes operadores, será utilizado o sinal $S[3]$ (vide (A.5) e (B.4)). Neste trabalho propõe-se um processamento que faz uma combinação polinomial dos três operadores, e em seguida, faz uma adequação de amplitude via ajuste sigmoidal.

O processamento via combinação polinomial combina as funções usadas nos três processamentos apresentadas nas equações (5.7), (5.8) e (5.9). Como dois dos processamentos apresentados reduzem significativamente o ruído, naturalmente eles introduzem fortes distorções nos trechos de voz, pois a redução também é forte nesses trechos. Assim, minimizando a redução de ruído, as distorções também serão minimizadas, pois o ruído se espalha de maneira uniforme pelo sinal.

Uma alternativa proposta para o uso deste processamento consiste em aumentar os valores das funções usadas em cada um dos processamentos (vide 5.2.1, 5.2.2 e 5.2.3) e considerar a soma das mesmas. Assim, dado α entre 0 e 1, indicando respectivamente, nas equações (D.1), (D.2) e (D.3)

$$y_i = \frac{|s_i| + |s_{i+1}|}{2}, \quad (\text{D.1})$$

$$z_i = \frac{|s_i|}{\alpha + |s_i|} e \quad (D.2)$$

$$w_i = \frac{|s_i|}{\alpha + |sr_i|} \quad (D.3)$$

a combinação polinomial proposta é a função $f(y_i, z_i, w_i)$ dada por:

$$f_i = y_i^3 + z_i^2 + w_i \quad (D.4)$$

A equação (D.1) é a equação (B.2) apresentada no Apêndice B. Para ilustrar a combinação proposta, os resultados obtidos serão apresentados sem detalhes dos procedimentos, pois estes já foram abordados nos três apêndices anteriores.

Em relação ao operador OMS, considerando o conjunto $S[3]=[s_0, \dots, s_7]$, dado em (A.5) e (B.4), e indicando por

$$Y^3[3]=[s_i x y_i^3 : i=0, \dots, 7]=[s_i x \left(\frac{|s_i| + |s_{i+1}|}{2} \right)^3 : i=0, \dots, 7]$$

tem-se (vide também (B.5')).

$$Y^3(3)=[-0,00000625 \quad 0,00078125 \quad 0,01929375 \quad 0,00084375 \quad 0,00214375 \quad -0,02786875 \\ -0,00000625 \quad -0,00000625]$$

Decompondo esse sinal, usando a transformada *wavelet* inversa de *Haar*, de modo análogo ao que foi feito no Apêndice C, obtém-se o sinal, denotado por $\hat{y}^3(3)$:

$$\hat{y}^3(3)=[0,0222125 \quad 0,02065 \quad -0,0163125 \quad -0,018 \quad -0,030025 \quad 0,0257125 \\ -0,00215 \quad -0,0021375]$$

Observe que o sinal reduzido $\hat{y}^3(3)$ tem amplitude absoluta de 0,030025 unidades de medida.

Analisando as duas curvas poligonais da Figura D1, pode-se notar que, quando se analisa os coeficientes do sinal original $s(3)$ (em azul: vide também a Figura A1) e os coeficientes resultantes da redução de ruído no sinal ruidoso da combinação proposta $\hat{y}^3(3)$ (em vermelho: Figura D1), todos no domínio do tempo, esses coeficientes sofrem uma redução de 0,769975 ($0,8 - 0,030025$) unidades de medida no valor absoluto de suas amplitudes, havendo assim redução de ruído.

Como se observa na Figura D1, as duas curvas poligonais praticamente mantêm as características dos coeficientes significativos, havendo uma diferença considerável entre suas amplitudes, fazendo com que a redução na amplitude do sinal processado seja forte. Consequentemente o sinal processado apresentará uma redução ou atenuação em seu ruído.

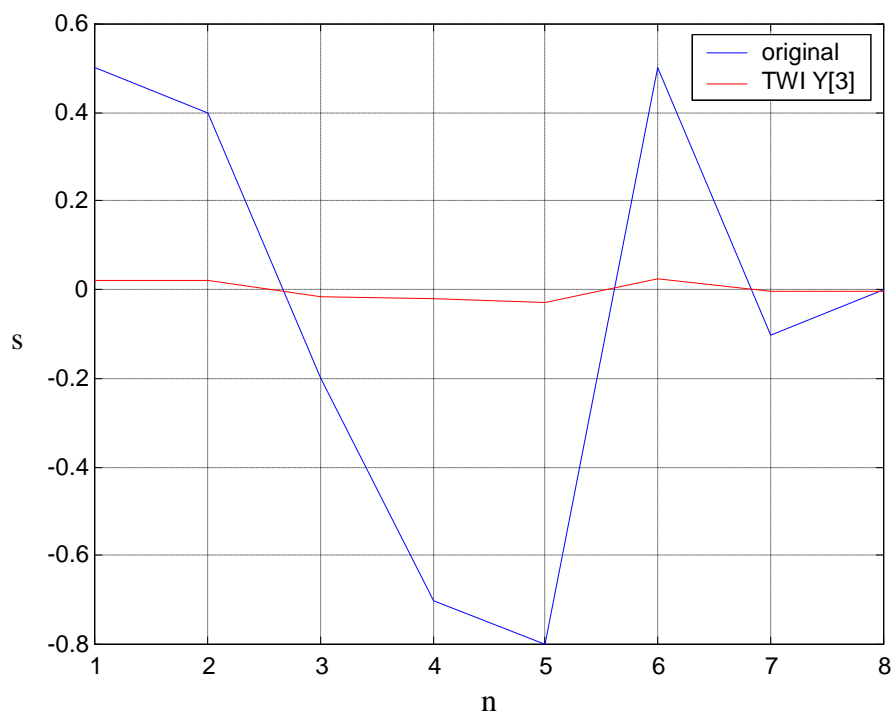


Figura D1: Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $\hat{y}^3(3)$, resultante do terceiro processamento da reconstrução do sinal ruidoso.

Na Figura D2, para melhor comparação entre os sinais original e processado, apresentam-se as três curvas poligonais. Em azul, apresenta-se o sinal original no domínio do tempo $s(3)$, apresentado no Apêndice A (vide Figura A1); em verde, representação do sinal ruidoso processado pelo operador OMS no domínio do tempo, denotado por $\hat{y}(3)$, apresentado no Apêndice C (vide Figura C3); e em vermelho apresenta-se o sinal ruidoso processado elevando ao cubo o operador OMS, sinal este denotado por $\hat{y}^3(3)$.

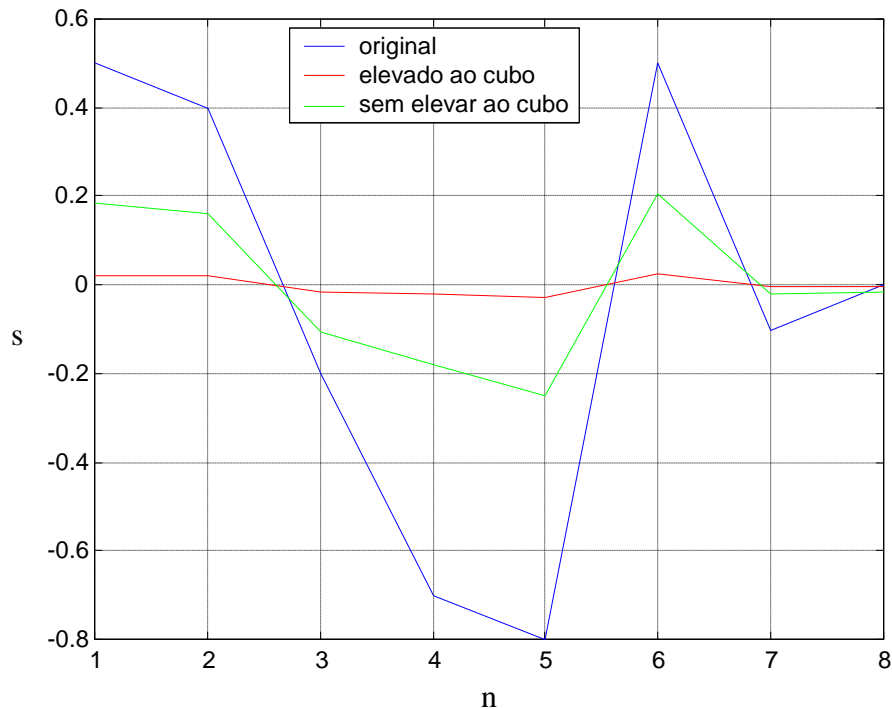


Figura D2: Representação dos coeficientes do sinal $s(3)$ (em azul); representação dos coeficientes do sinal $\hat{y}(3)$ (em verde) e representação dos coeficientes do sinal $\hat{y}^3(3)$ (em vermelho).

Observando a Figura D2, nota-se que os três sinais têm as mesmas características em seus coeficientes, pois as fases são mantidas, embora haja forte redução em suas amplitudes.

Para o segundo processamento denominado como operador SNRP (vide 5.2.2) os procedimentos utilizados para decomposição e reconstrução dos sinais foram os mesmos apresentados nos Apêndices A, B e C. Cabe ressaltar que para este processamento o operador utilizado é dado pela equação (5.8), sendo a mesma apresentada em (D.2).

Em relação ao operador SNRP, conforme (5.8), considerando novamente o conjunto $S[3]$, dado em (A.5) e (B.4), e indicando por

$$Z[3] = [s_i x z_i : i=0, \dots, 7] = \left[s_i x \left(\frac{|s_i|}{\alpha + |s_i|} \right) : i=0, \dots, 7 \right]$$

tem-se

$$Z[3] = [-0,041666667 \quad 0,041666667 \quad 0,440217392 \quad 0,240384616 \quad 0,041666667 \quad -0,640151515 \\ -0,041666667 \quad -0,041666667]$$

Decompondo esse sinal, via transformada *wavelet* inversa de *Haar*, obtém-se o sinal denotado por $\hat{z}(3)$:

$$\hat{z}(3) = [0,481884059 \quad 0,398550725 \quad -0,199832776 \quad -0,680602008 \quad -0,598484848 \\ 0,681818182 \quad 0,083333334 \quad 0,166666668]$$

Do mesmo modo, indicando por

$$Z^2[3] = [s_i \ x \ z_i^2 : i=0, \dots, 7] = [s_i \ x \ \left(\frac{|s_i|}{\alpha + |s_i|} \right)^2 : i=0, \dots, 7]$$

tem-se

$$Z^2[3] = [-0,034722222 \quad 0,034722222 \quad 0,430647449 \quad 0,231139054 \quad 0,034722222 \\ -0,63045225 \quad -0,034722222 \quad -0,034722222]$$

E decompondo esse sinal, do mesmo modo como nos casos anteriores, obtém-se o sinal, denotado por $\hat{z}^2(3)$:

$$\hat{z}^2(3) = [0,465369671 \quad 0,395925227 \quad -0,199508395 \quad -0,661786503 \quad -0,665174472 \\ 0,595730028 \quad 0,138888888 \quad 0,208333332]$$

Observar que o sinal reduzido $\hat{z}^2(3)$ tem amplitude absoluta de 0,665174472 unidades de medida.

Analisando as duas curvas poligonais da Figura D3, pode-se notar que, quando se analisa os coeficientes do sinal original $s(3)$ (em azul: vide também a Figura A1) e os coeficientes resultantes da redução de ruído no sinal ruidoso da combinação proposta $\hat{z}^2(3)$ (em vermelho: Figura D3), todos no domínio do tempo, esses coeficientes sofrem uma redução de 0,134825528 ($0,8 - 0,665174472$) unidades de medida no valor absoluto de suas amplitudes, havendo assim, redução de ruído.

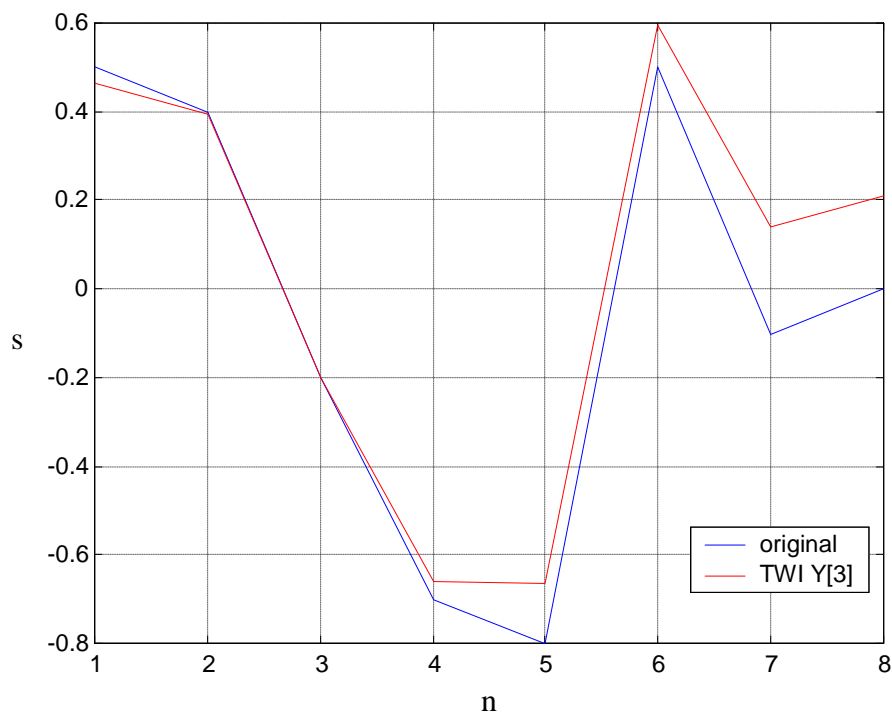


Figura D3: Representação dos coeficientes do sinal $s(3)$ e dos coeficientes do sinal $\hat{z}^2(3)$, resultante do terceiro processamento da reconstrução do sinal ruidoso.

Na Figura D4, para melhor comparação entre os sinais original e processado, apresentam-se as três curvas poligonais. Em azul, apresenta-se o sinal original no domínio do tempo $s(3)$, apresentado no Apêndice A (vide Figura A1); em verde, representação do sinal ruidoso processado pelo operador SNRP no domínio do tempo, denotado por $\hat{z}(3)$; e em vermelho apresenta-se o sinal ruidoso processado pela combinação proposta, denotado por $\hat{z}^2(3)$.

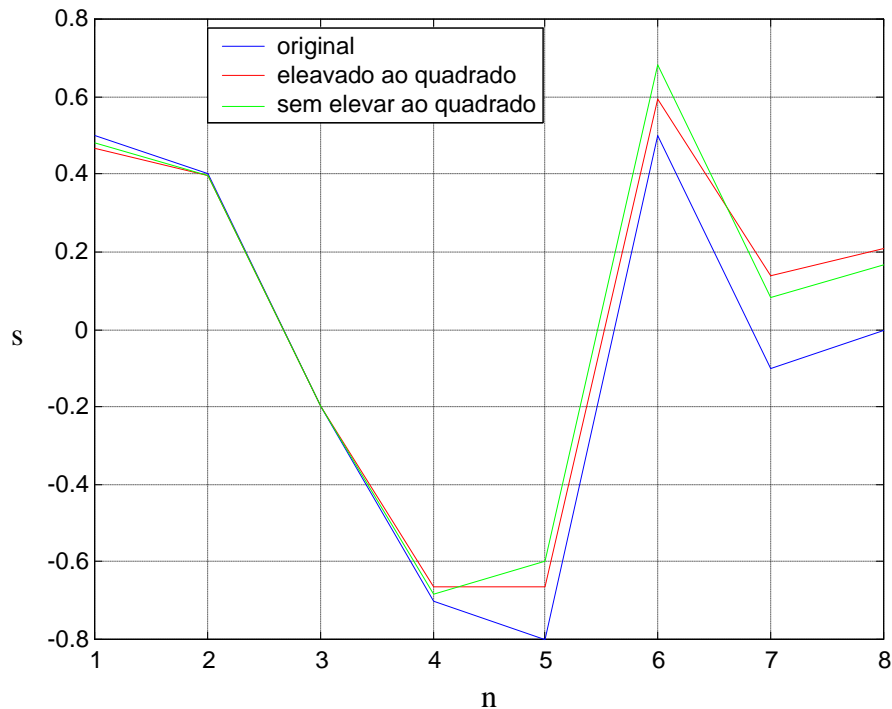


Figura D4: Representação dos coeficientes do sinal $s(3)$ (em azul); representação dos coeficientes do sinal $\hat{z}(3)$ (em verde) e representação dos coeficientes do sinal $\hat{z}^2(3)$ (em vermelho).

Agora será considerada a combinação dos operadores OMS e SNRP. Embora, a soma apresentada na equação (D.4) contenha uma terceira parcela, que é o operador SNRPR, para a reconstrução do sinal, o mesmo não foi usado, pois este operador é utilizado apenas quando o sinal original está contaminado por algum tipo de ruído. Como o mesmo trabalha com a estimativa do ruído, ele só é utilizado em sinais de voz onde se possa fazer a distinção entre silêncio e voz, sendo que o ruído é estimado nos trechos de silêncio.

Considerando o conjunto $S[3]$, dado em (A.5) e (B.4), e denotado por:

$$Y_2^3[3] := [s_i x (y_i^3 + z_i^2) : i=0, \dots, 7] = [s_i x \left\{ \left(\frac{|s_i| + |s_{i+1}|}{2} \right)^3 + \left(\frac{|s_i|}{\alpha + |s_i|} \right)^2 \right\} : i=0, \dots, 7]$$

tem-se

$$Y_2^3[3] = \begin{bmatrix} -0,034728472 & 0,035503472 & 0,449941199 & 0,231982804 & 0,036865972 \\ -0,658321 & -0,034728472 & -0,034728472 & & \end{bmatrix}$$

Decompondo esse sinal, de modo análogo ao que foi feito nos casos anteriores, obtém o sinal reduzido pela “combinação” $\hat{y}^3(3) + \hat{z}^2(3)$, sinal este denotado por $\hat{y}_2^3(3)$:

$$\hat{y}_2^3(3) = [0,487582171 \quad 0,416575227 \quad -0,215820895 \quad -0,679786503 \quad -0,764643916 \\ 0,551998084 \quad -0,071594444 \quad -0,0021375]$$

Na Figura D5, apresenta-se o sinal reconstruído, em vermelho, usando a combinação proposta e o sinal original, em azul, para que se possa compará-los.

A diferença de amplitude absoluta entre os dois sinais apresentados na Figura D5 é de 0,035356084 ($0,8 - 0,764643916$). Esta diferença seria menor caso existisse ruído no sinal, pois haveria a introdução da terceira parcela da equação (D4).

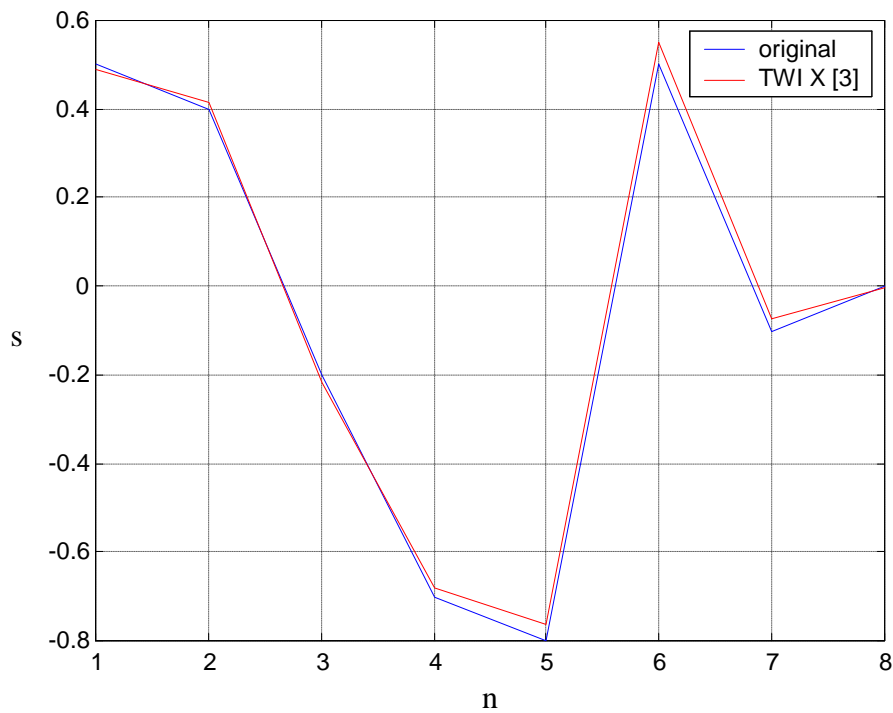


Figura D5: Sinal $s(3)$ (em azul) e $X[3] = \hat{y}_2^3(3)$, combinação de $\hat{y}^3(3)$ e de $\hat{z}^2(3)$ (em vermelho).

APÊNDICE E

E1. ANÁLISE COM UM SINAL DE VOZ

Nesta seção será feita uma análise com um trecho de um sinal de voz utilizando os conceitos que foram apresentados nos apêndices anteriores. Ainda como forma de ilustração serão apresentadas as figuras resultantes dos processamentos, utilizando um sinal de voz denominado de Sinal A, que é apresentado no Capítulo 6 deste trabalho.

Conforme os apêndices apresentados anteriormente, as análises serão feitas usando o operador OMS e a combinação proposta, sendo o mesmo procedimento a ser realizado para o Sinal A. Para processar o Sinal A, utilizando os processamentos propostos, foi feita a análise em um trecho de voz contendo 128 amostras, usando a *wavelet* de *Daubechies* (db10). Na Figura E1 apresenta-se um trecho do sinal de voz do Sinal A no domínio do tempo e na Figura E2 o respectivo trecho do sinal no domínio *wavelet* (domínio transformado).

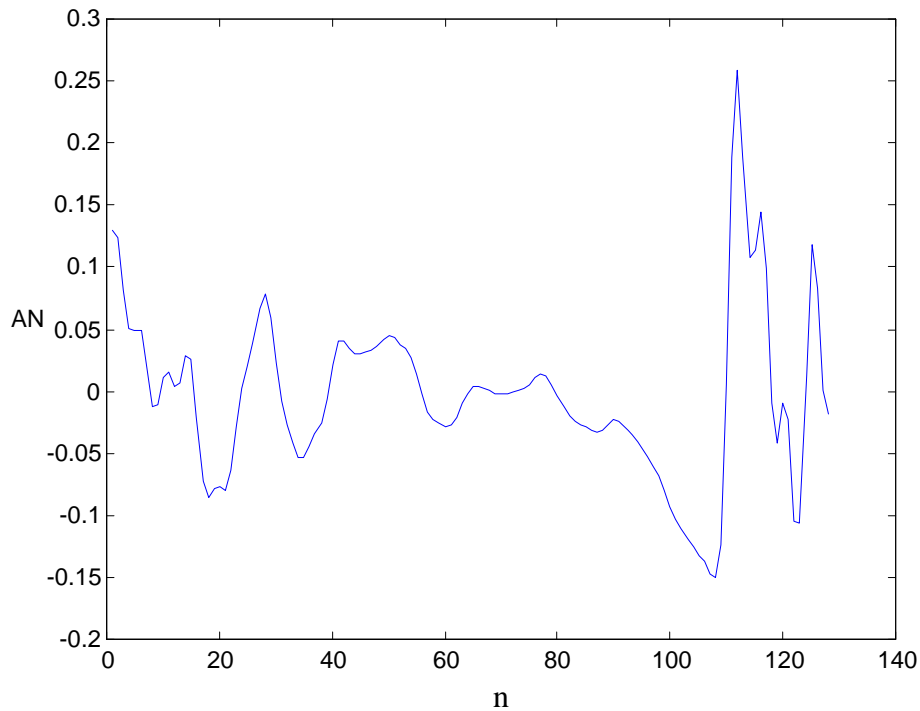


Figura E1: Trecho de um sinal de voz (Sinal A) no domínio do tempo.

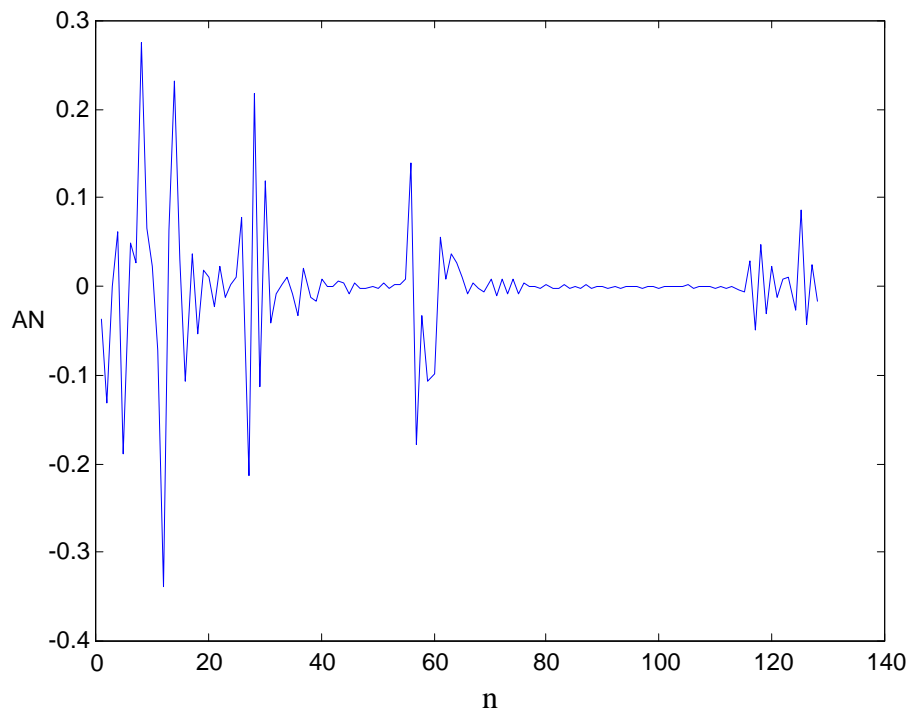


Figura E2: Trecho de um sinal de voz (Sinal A) no domínio *wavelet* (domínio transformado).

Como os processos de decomposição e reconstrução de um sinal já foram analisados nos apêndices anteriores, as figuras que serão apresentadas em seguida são resultantes da utilização destes conceitos juntamente com a aplicação do operador OMS

apresentado no Apêndice B. Como consequência disso tudo, omitir-se-ão tais demonstrações e apenas serão apresentadas as figuras resultantes de tais processamentos. Na Figura E3 apresenta-se o mesmo trecho do sinal no domínio *wavelet* processado utilizando o operador OMS e na Figura E4, o mesmo trecho do sinal reconstruído no domínio do tempo utilizando a transformada *wavelet* inversa.

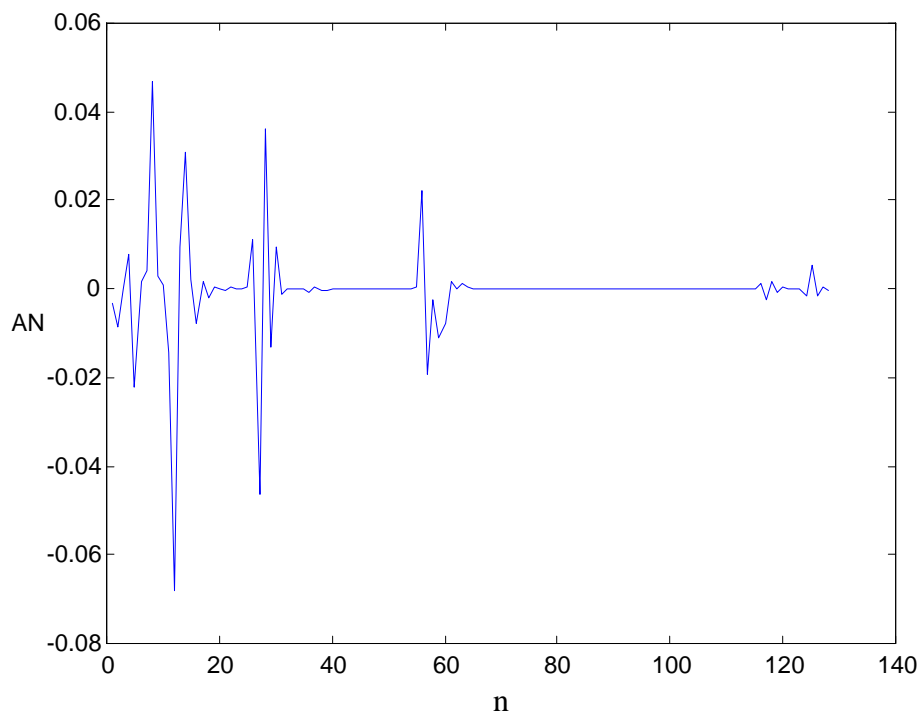


Figura E3: Trecho do sinal de voz (Sinal A) processado no domínio *wavelet* utilizando o operador OMS.

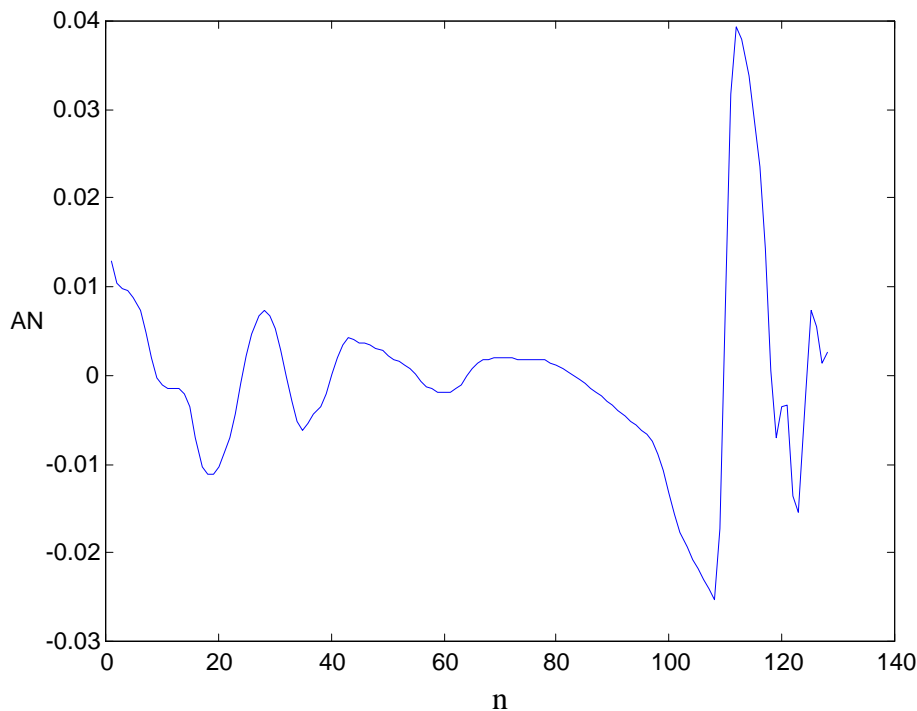


Figura E4: Reconstrução do trecho do sinal de voz (Sinal A) no domínio do tempo usando a TWI.

Pode-se verificar que a Figura E1 e a Figura E4 apresentam as mesmas características em suas formas de onda, diferenciando-se apenas em suas amplitudes. Analisando as duas formas de ondas, verifica-se que o trecho de sinal de voz utilizando o operador OMS reduz drasticamente a sua amplitude em comparação com o sinal original, o que se deve ao fato de o sinal original ter amplitude no intervalo $[-1,1]$, fazendo com que a redução na amplitude do sinal processado seja forte. Essa redução em parte se deve ao fato de se tratar de um trecho de sinal sem ruído. Como não há componentes ruidosos, o operador reduz coeficientes significativos do sinal original. Isto se pode notar na Figura E5, onde se faz a comparação das duas formas de onda apresentadas nas Figuras E1 (em azul) e na Figura E4 (em vermelho).

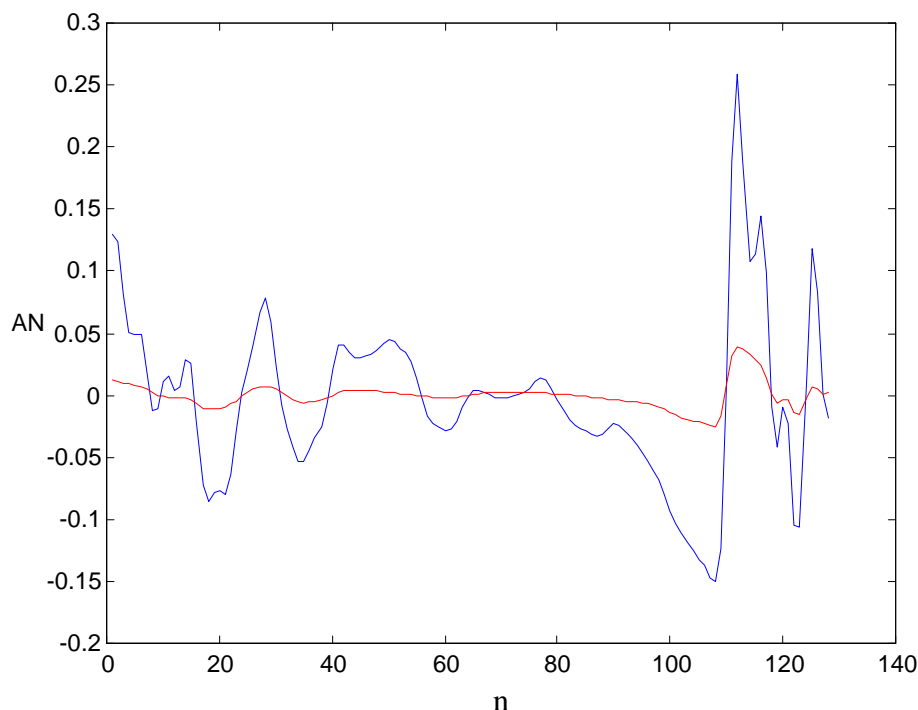


Figura E5: Trecho do sinal de voz (Sinal A) original no domínio do tempo (em azul: Figura E1) e trecho do sinal de voz (Sinal A) processado pelo operador OMS reconstruído no domínio do tempo (em vermelho: Figura E4).

Para ilustrar a combinação proposta, os resultados obtidos serão apresentados sem detalhes dos procedimentos, pois estes já foram abordados nos apêndices anteriores, principalmente no Apêndice D. Serão apenas apresentadas as figuras resultantes da combinação proposta para o trecho de sinal de voz (Sinal A).

Será considerada a combinação dos operadores OMS e SNRPR. Embora a soma apresentada na equação (D.4) contenha uma terceira parcela, que é o operador SNRPR, para a reconstrução do sinal, o mesmo não foi usado, pois este operador é utilizado apenas quando o sinal original está contaminado por algum tipo de ruído. Como o mesmo trabalha com a estimativa do ruído, ele só é utilizado em sinais de voz onde se possa fazer a distinção entre silêncio e voz, sendo que o ruído é estimado nos trechos de silêncio.

Então, na Figura E6 apresenta-se trecho do sinal de voz (Sinal A) processado no domínio *wavelet* usando a combinação proposta apresentada no Apêndice D.

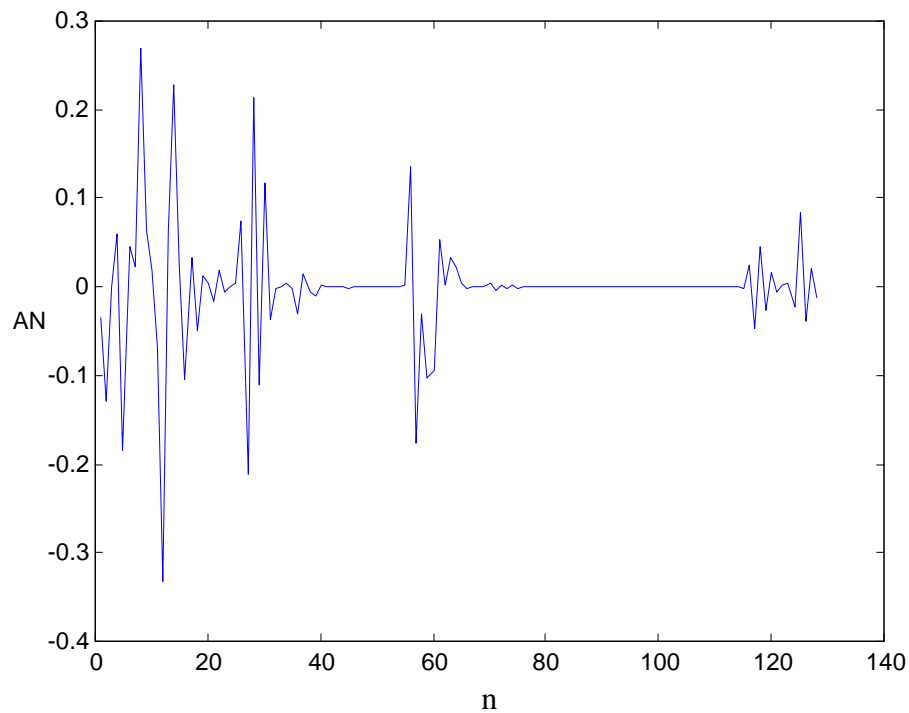


Figura E6: Trecho do sinal de voz (Sinal A) processado no domínio *wavelet* utilizando a combinação proposta.

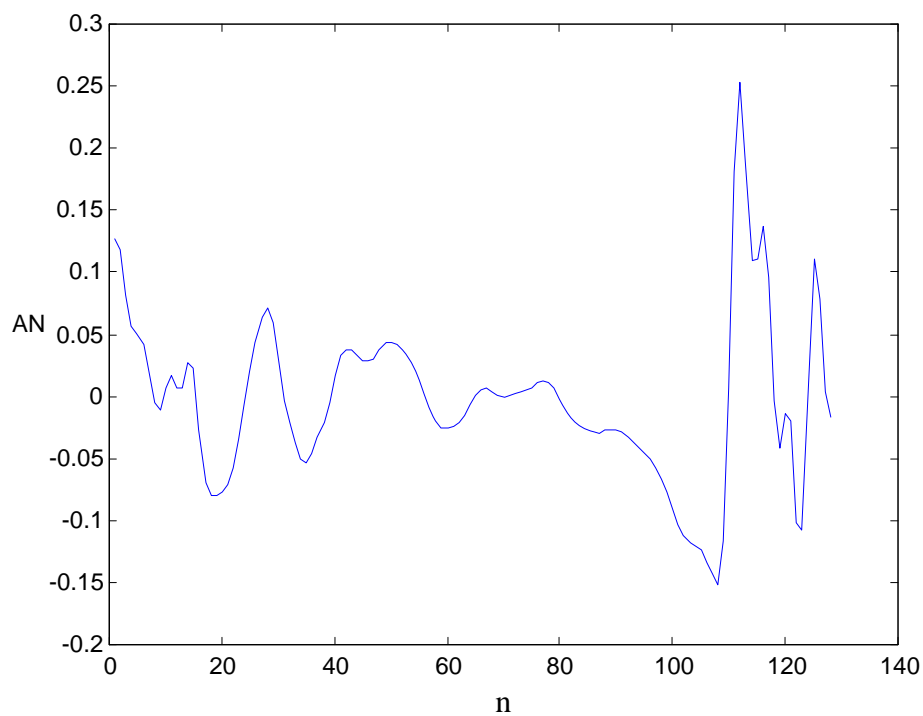


Figura E7: Reconstrução do trecho do sinal de voz (Sinal A) usando a TWI no domínio do tempo utilizando a combinação proposta.

Na Figura E7 apresenta-se o mesmo trecho do sinal de voz (Sinal A) reconstruído utilizando a TWI, no domínio do tempo, utilizando a combinação proposta.

Pode-se notar que quando se comparam a Figura E1 e a Figura E7, ambas no domínio do tempo, há uma pequena diferença em suas formas de onda. Esta diferença seria menor ainda caso existisse ruído no sinal, pois haveria a introdução da terceira parcela da equação (D4). Isto pode ser observado na Figura E8, onde se faz a comparação entre o trecho do sinal original (Sinal A) representado em azul e, em vermelho, o trecho do sinal (Sinal A) processado pela combinação proposta.

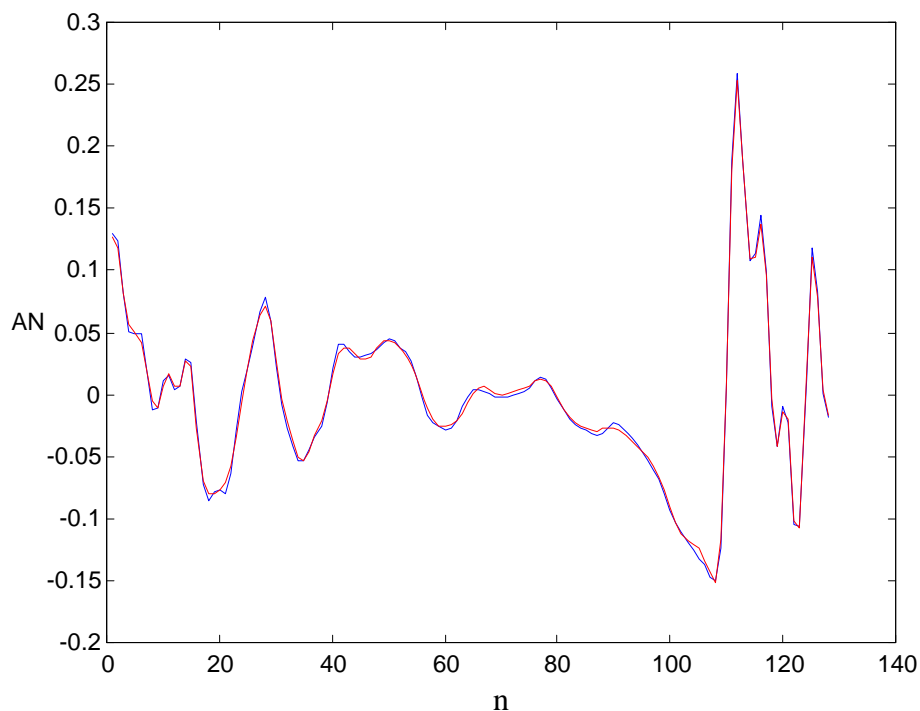


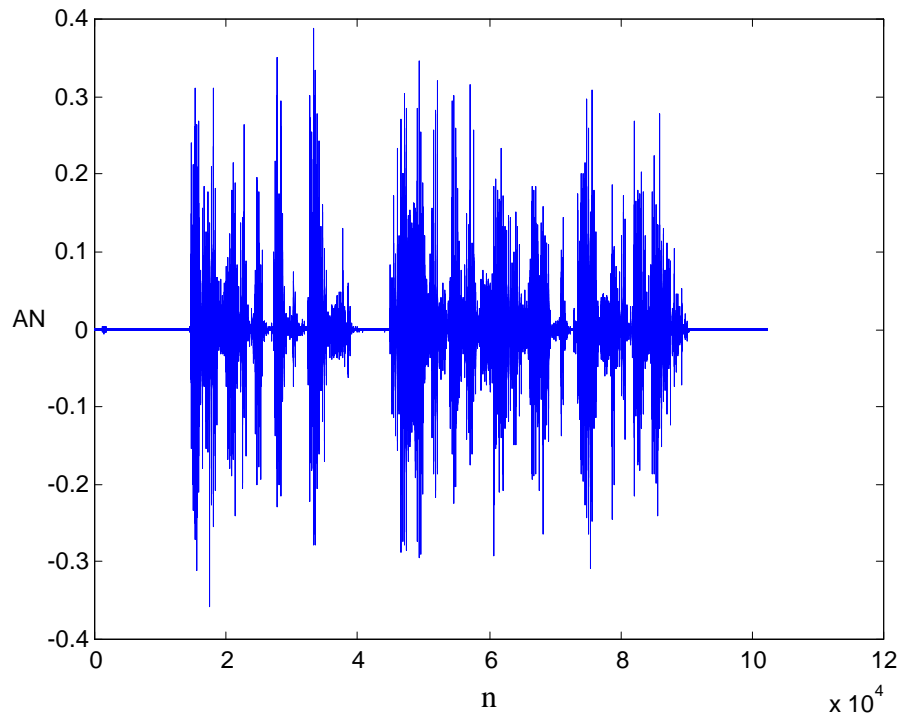
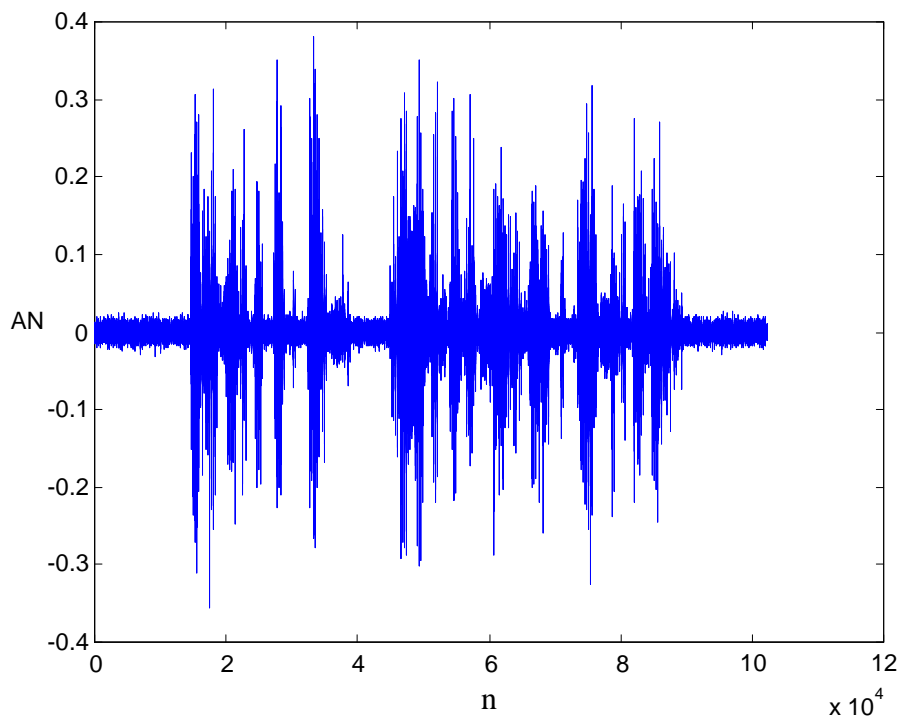
Figura E8: Trecho do sinal de voz (Sinal A) original no domínio do tempo (em azul: Figura E1) e trecho do sinal de voz (Sinal A) processado utilizando a combinação proposta no domínio do tempo (em vermelho: Figura E7).

APÊNDICE F

SINAIS NO DOMÍNIO *WAVELET*

O sinal usado no capítulo 5, para apresentação da metodologia proposta, é apresentado neste apêndice, no domínio *wavelet*. As figuras a seguir são relativas aos três processamentos e a combinação polinomial dos três.

F.1. SINAIS ANTES DO PROCESSAMENTO

Figura F1: Sinal limpo $X[n]$.Figura F2: Sinal contaminado com ruído branco $S[n]$.

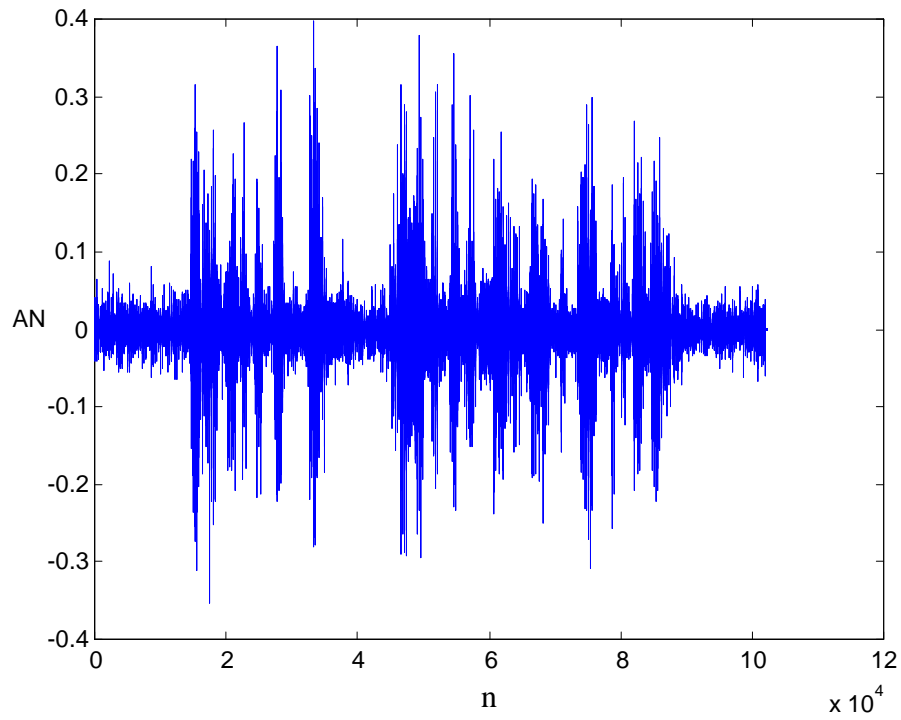


Figura F3: Sinal contaminado com ruído colorido $S_0[n]$.

F.2. OPERADOR OMS

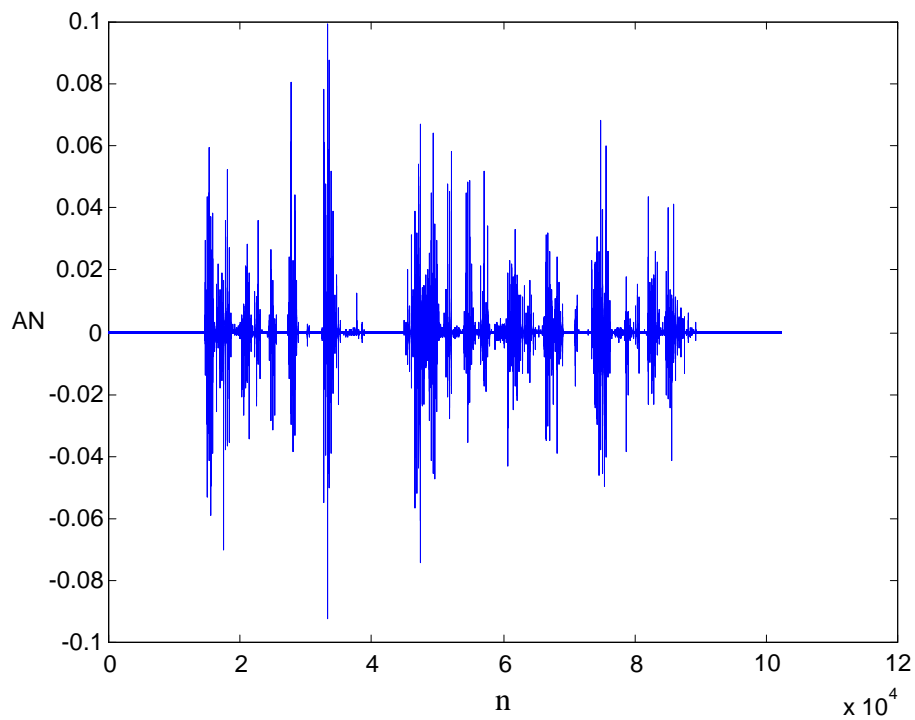


Figura F4: Sinal limpo processado pelo operador OMS, $Y_x[n]$.

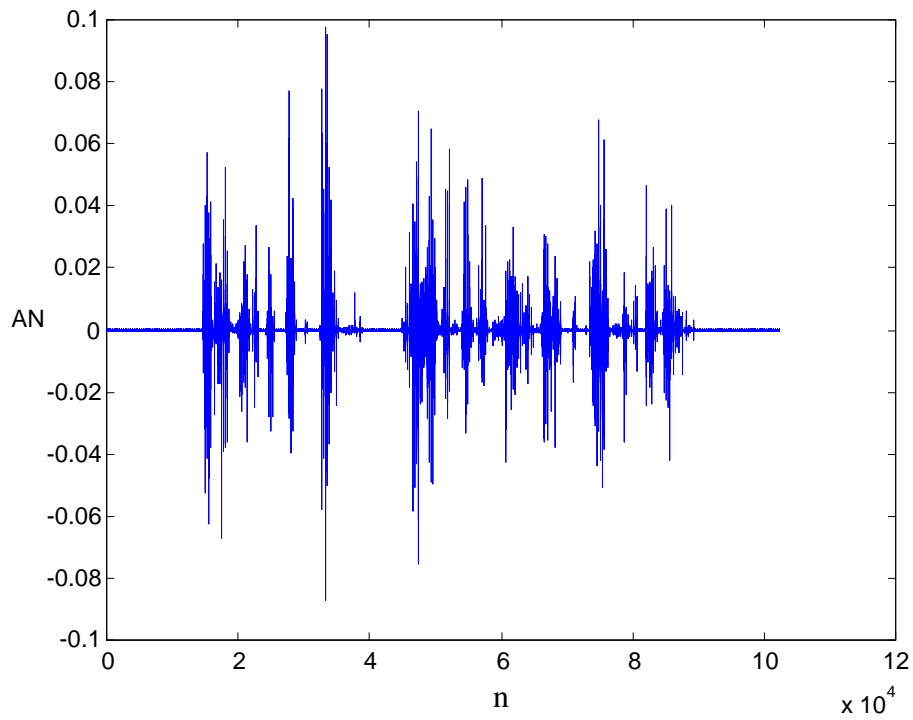


Figura F5: Sinal contaminado com ruído branco processado pelo operador OMS, $Y_s[n]$.

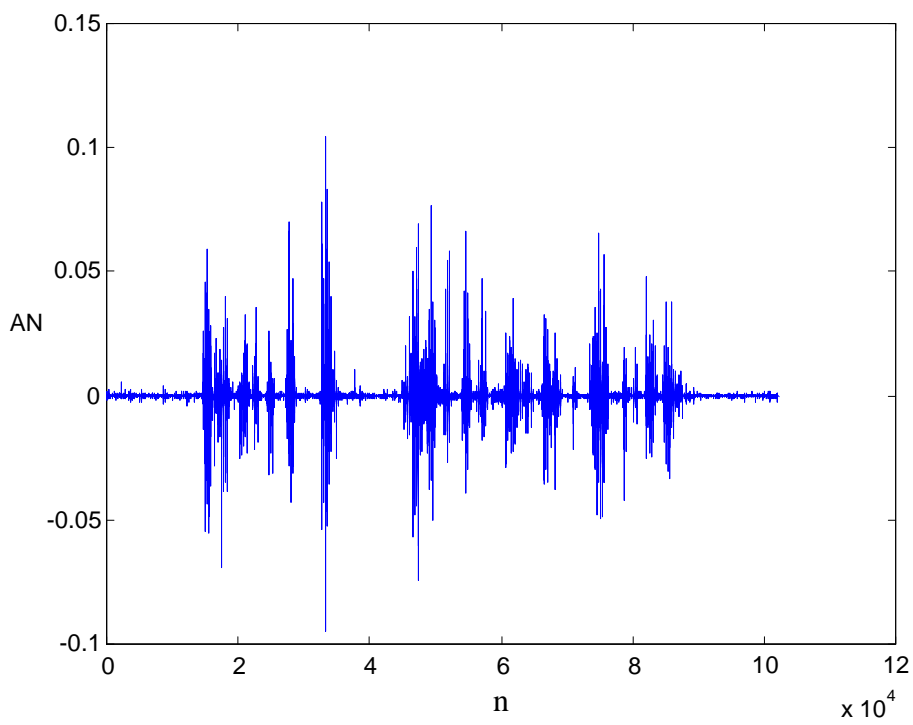


Figura F6: Sinal contaminado com ruído colorido processado pelo operador OMS, $Y_o[n]$.

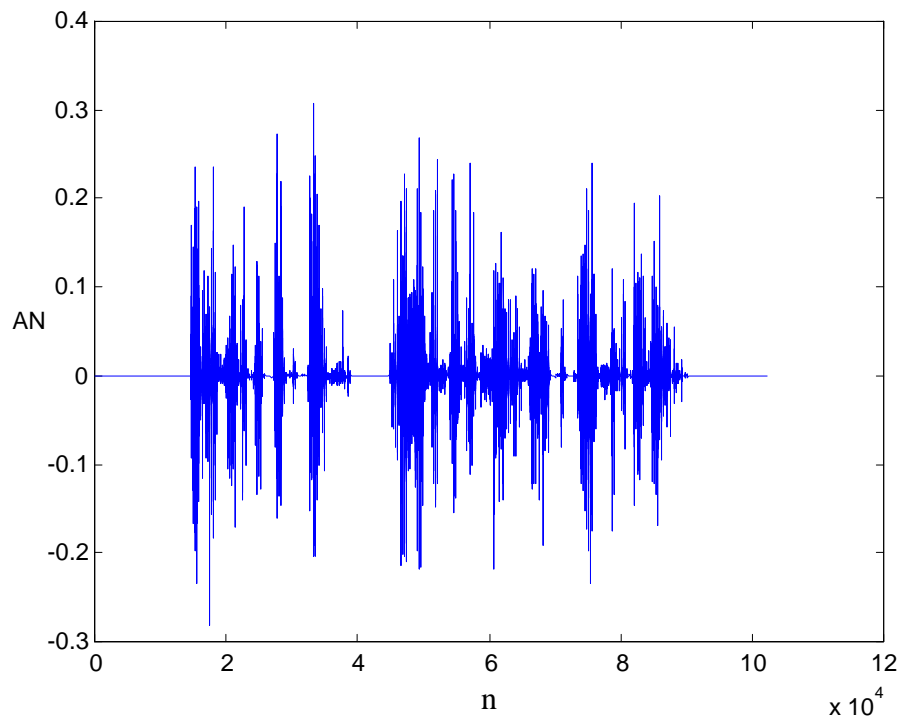
F.3. OPERADOR SNRP

Figura F7: Sinal limpo processado pelo operador SNRP, $Y_x[n]$.

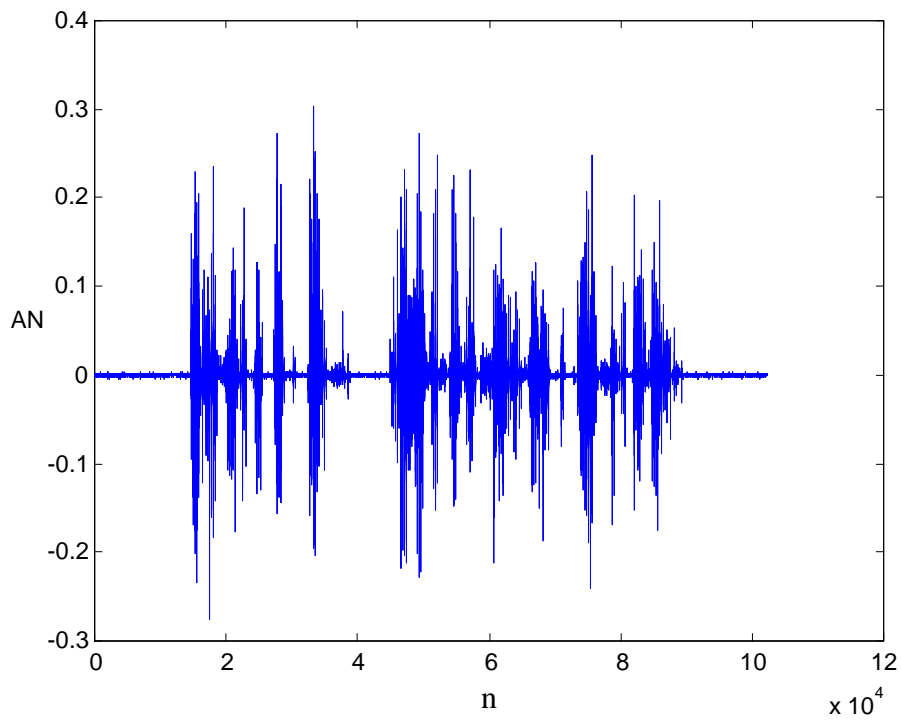


Figura F8: Sinal contaminado com ruído branco processado pelo operador SNRP, $Y_s[n]$.

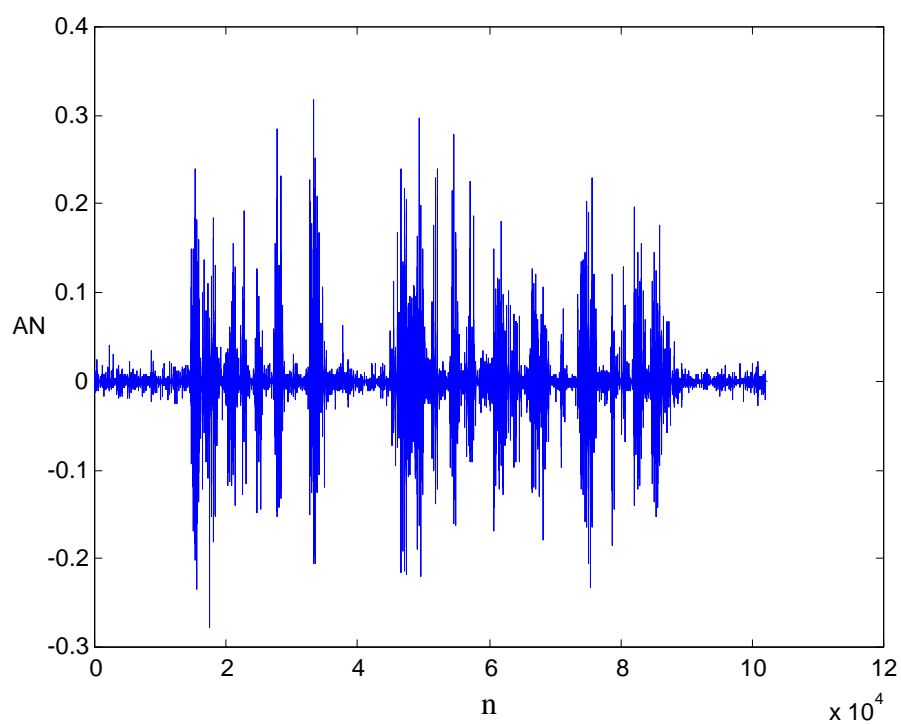


Figura F9: Sinal contaminado com ruído colorido processado pelo operador SNRP, $Y_0[n]$.

F.4. OPERADOR SNRPR

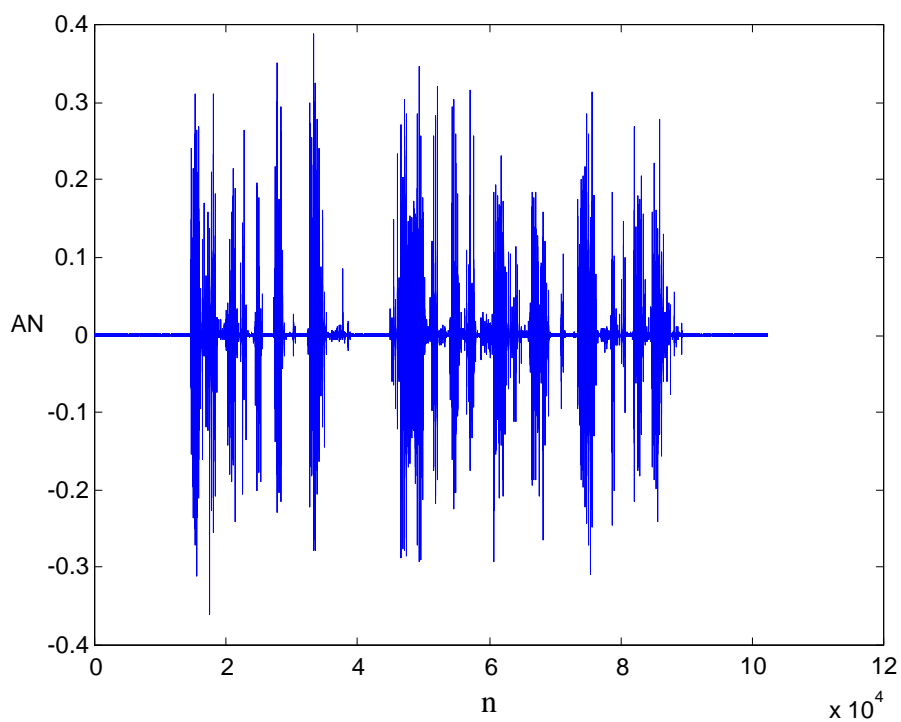


Figura F10: Sinal limpo processado pelo operador SNRPR, $Y_x[n]$.

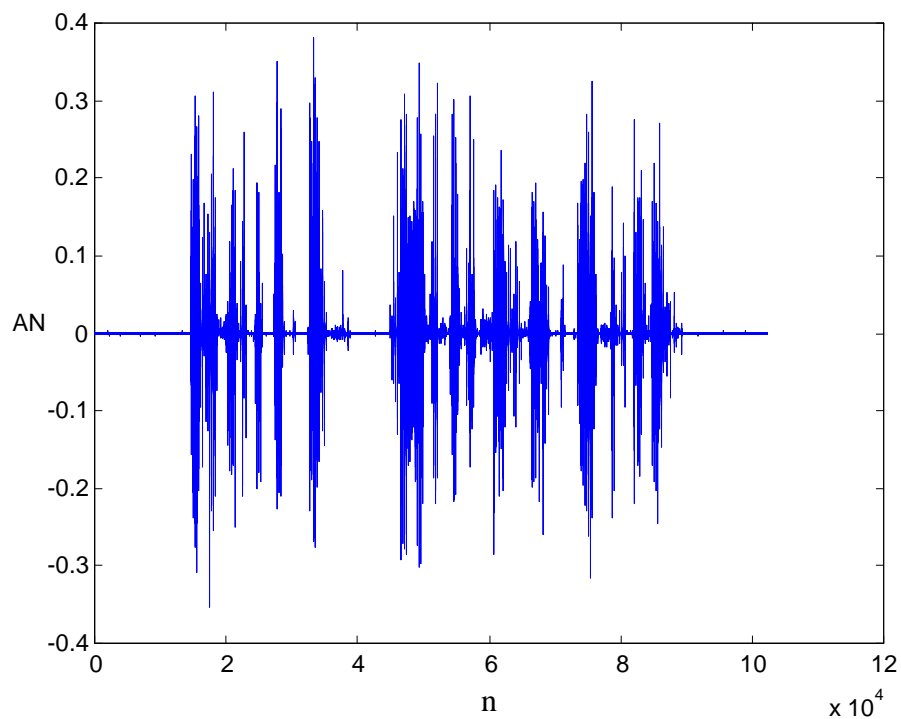


Figura F11: Sinal contaminado com ruído branco processado pelo operador SNRPR, $Y_s[n]$.

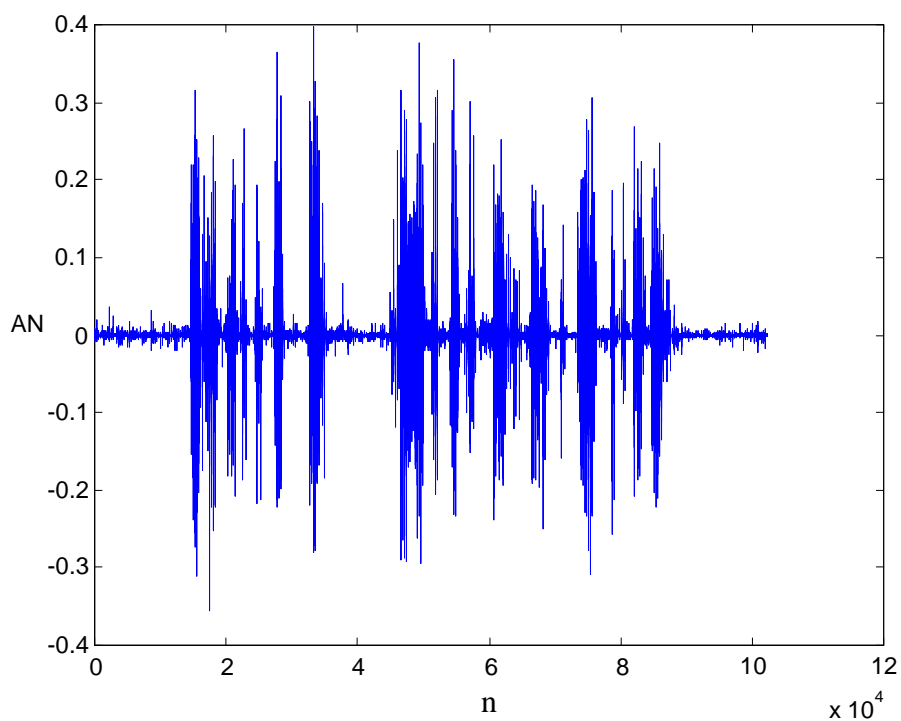
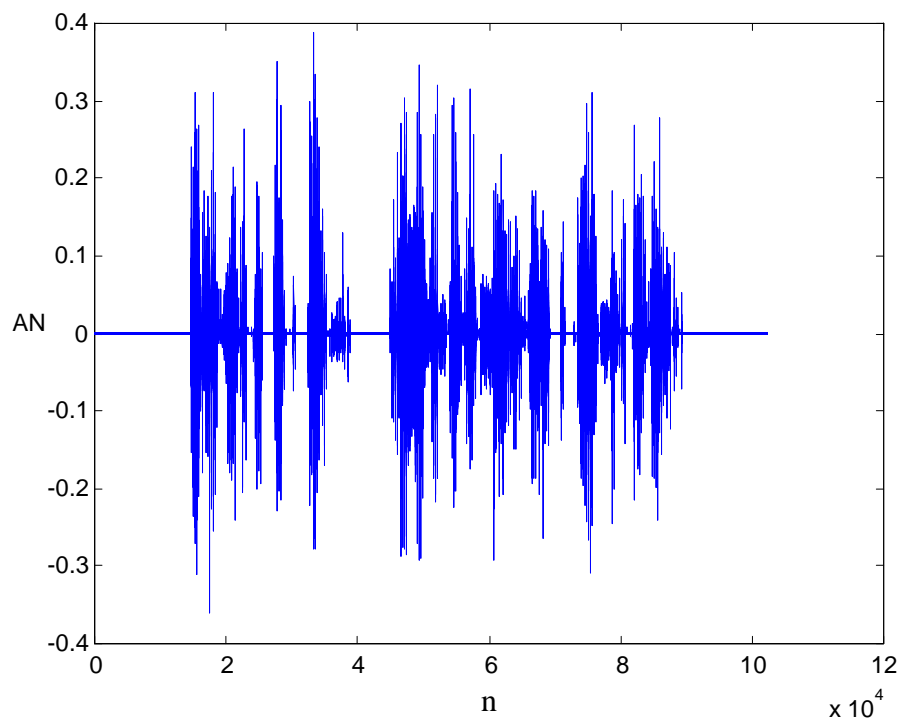
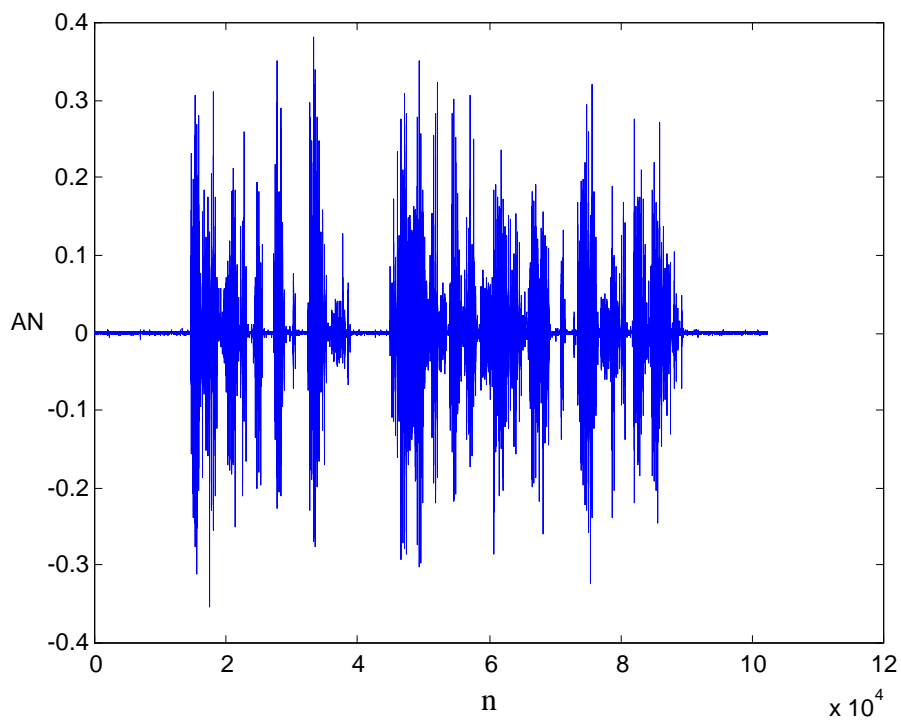


Figura F12: Sinal contaminado com ruído colorido processado pelo operador SNRPR, $Y_0[n]$.

F.5. COMBINAÇÃO PROPOSTAFigura F13: Sinal limpo processado pela Combinação Proposta, $Y_x [n]$.Figura F14: Sinal contaminado com ruído branco processado pela Combinação Proposta, $Y_s [n]$.

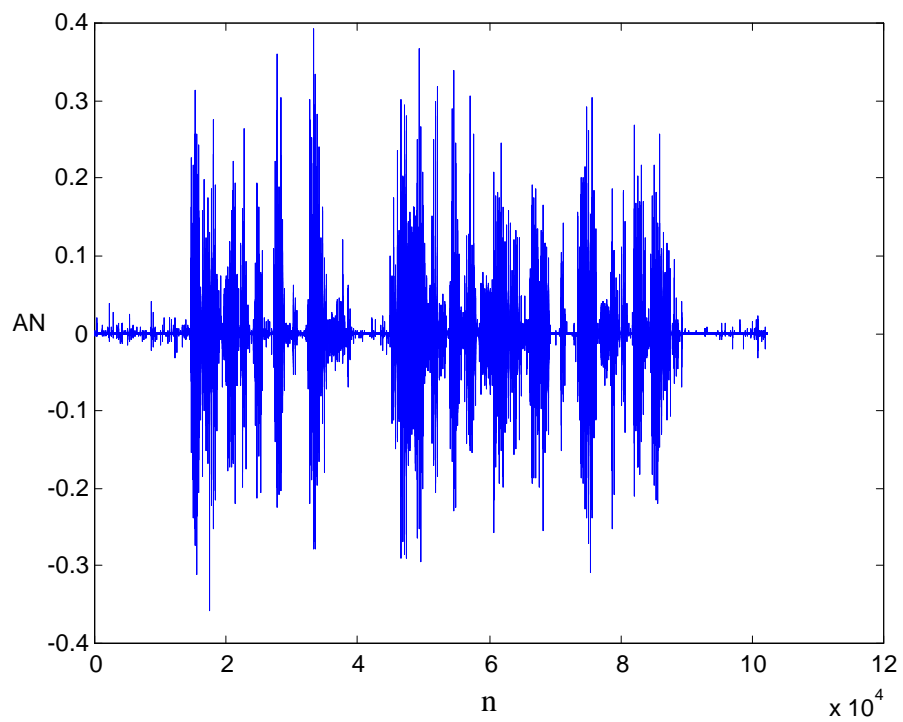


Figura F15: Sinal contaminado com ruído colorido processado pela Combinação Proposta, $Y_0[n]$.

APÊNDICE G

G1. PUBLICAÇÕES

Este apêndice apresenta uma lista de artigos resultantes dos trabalhos desenvolvidos em vínculos científicos com esta tese, até o momento.

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F., VIEIRA FILHO, J. Uma função de transferência para redução de ruído no domínio wavelet. In BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, DINCON'2009, 5, 2009, Bauru. **Proceedings of the...** Bauru (no prelo), 2009.

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Análise de métodos de redução de ruído por limiar no domínio wavelet. **TEMA: Tendências em Matemática Aplicada e Computacional**, São Carlos, v.9, n.3, p.471-480, 2008a.

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F., VIEIRA FILHO, J. Redução não-limiar de ruído no domínio wavelet. In BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, DINCON'2008, 5, 2008,

Presidente Prudente. **Proceedings of the...** Presidente Prudente: SBMAC, 2008b. p.961 – 967.

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Análise de métodos de redução de ruído em sinais de voz no domínio wavelet. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, DINCON'2007, 6, 2007, São José do Rio Preto. **Proceedings of the...** São José do Rio Preto: SBMAC, 2007a. p 918 – 923.

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Sobre a transformada wavelet discreta. In: SIMPÓSIO REGIONAL DE MATEMÁTICA E SUAS APLICAÇÕES, SRMAIS, 1, 2007, Ilha Solteira. **Simpósio...** Ilha Solteira: Unesp/Feis, 2007b. p.1-4, (CD-ROM).

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Analysis of thresholding methods in the wavelet domain. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, DINCON'2006, 6, 2006, Guaratinguetá. **Proceedings of the...** Guaratinguetá: SBMAC, 2006. (CD-ROM).

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Wavelets: teoria e aplicações. In: BRAZILIAN CONFERENCE ON DYNAMICS, CONTROL AND THEIR APPLICATIONS, DINCON'2005, 6, 2005, Bauru. **Proceedings of the...** Bauru: SBMAC, 2005.(CD-ROM).

SOARES, W. C.; DUARTE, M. A. Q.; VILLARREAL, F.; VIEIRA FILHO, J. Wavelets: Theory and Applications. In: ITALIAN-LATIN AMERICAN CONFERENCE ON APPLIED AND INDUSTRIAL MATHEMATICS, 5, 2004, Trujillo - Peru. **Resumenes do Italian-Latin American Conference on Applied and Industrial Mathematics.** Trujillo: Universidad Nacional de Trujillo – Peru. 2004. p 26.