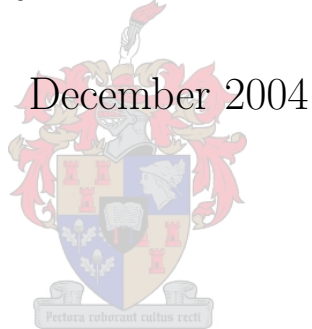


# Wavelet-based Speech Enhancement: A Statistical Approach

Wynand Harmse



Thesis presented in partial fulfilment  
of the requirements for the degree of

**Master of Engineering**

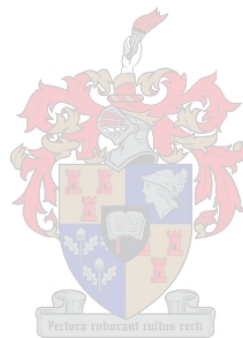
at the

**University of Stellenbosch**

*Supervisor:* **Ludwig Schwardt**

# Declaration

*I, the undersigned, hereby declare that the work contained in this thesis is my own original work and has not previously in its entirety or in part been submitted at any university for a degree.*



2004/11/25

---

Wynand Harmse

---

Date

# Abstract

Speech enhancement is the process of removing background noise from speech signals. The equivalent process for images is known as image denoising. While the Fourier transform is widely used for speech enhancement, image denoising typically uses the wavelet transform. Research on wavelet-based speech enhancement has only recently emerged, yet it shows promising results compared to Fourier-based methods. This research is enhanced by the availability of new wavelet denoising algorithms based on the statistical modelling of wavelet coefficients, such as the hidden Markov tree.

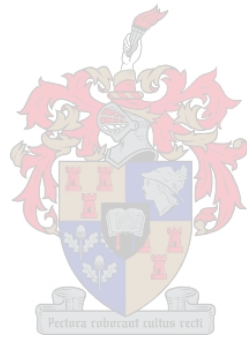
The aim of this research project is to investigate wavelet-based speech enhancement from a statistical perspective. Current Fourier-based speech enhancement and its evaluation process are described, and a framework is created for wavelet-based speech enhancement. Several wavelet denoising algorithms are investigated, and it is found that the algorithms based on the statistical properties of speech in the wavelet domain outperform the classical and more heuristic denoising techniques. The choice of wavelet influences the quality of the enhanced speech and the effect of this choice is therefore examined. The introduction of a noise floor parameter also improves the perceptual quality of the wavelet-based enhanced speech, by masking annoying residual artifacts. The performance of wavelet-based speech enhancement is similar to that of the more widely used Fourier methods at low noise levels, with a slight difference in the residual artifact. At high noise levels, however, the Fourier methods are superior.

# Opsomming

Spraaksuiwering is die proses waardeur agtergrondgeraas uit spraakseine verwyder word. Die ekwivalente proses vir beelde word beeldsuiwering genoem. Terwyl spraaksuiwering in die algemeen in die Fourier-domein gedoen word, gebruik beeldsuiwering tipies die golfie-transform. Navorsing oor golfie-gebaseerde spraaksuiwering het eers onlangs verskyn, en dit toon reeds belowende resultate in vergelyking met Fourier-gebaseerde metodes. Hierdie navorsingsveld word aangehelp deur die beskikbaarheid van nuwe golfie-gebaseerde suiweringsstegnieke wat die golfie-koëffisiënte statisties modelleer, soos die verskuilde Markov-boom.

Die doel van hierdie navorsingsprojek is om golfie-gebaseerde spraaksuiwering vanuit 'n statistiese oogpunt te bestudeer. Huidige Fourier-gebaseerde spraaksuiweringsmetodes asook die evalueringsproses vir sulke algoritmes word bespreek, en 'n raamwerk word geskep vir golfie-gebaseerde spraaksuiwering. Verskeie golfie-gebaseerde algoritmes word ondersoek, en daar word gevind dat die metodes wat die statistiese eienskappe van spraak in die golfie-gebied gebruik, beter vaar as die klassieke en meer heuristiese metodes. Die keuse van golfie beïnvloed die kwaliteit van die gesuiwerde spraak, en die effek van hierdie keuse word dus ondersoek. Die gebruik van 'n ruisvloer parameter verhoog ook die kwaliteit van die golfie-gesuiwerde spraak, deur steurende residuele artefakte te verberg. Die golfie-metodes vaar omtrent dieselfde as die klassieke Fourier-metodes by lae ruisvlakke, met 'n klein verskil in residuele artefakte. By hoë ruisvlakke vaar die Fourier-metodes egter steeds beter.

*to my parents and  
Ludwig, Jacques and Dorita*



# Acknowledgements

Free software used in this research:

- **STSA software** [61],  
Matlab STSA Toolbox for Audio Signal Noise Reduction  
- P. Wolfe.
- **Noise source software** [30],  
Additive Noise Sources, Version 1.0  
- J.H.L Hansen.
- **Distortion Measures software** [45],  
Objective Speech Quality Assessment, Version 1.0  
- B. Pellom.
- **Classical wavelet shrinkage software** [16],  
WaveLab 802 for Matlab 5.x  
- Donoho, Duncan, Huo and Levi.
- **HMT C software** [12],  
Software For Image Denoising Using Wavelet-Domain Hidden Markov Tree Models  
- H. Choi.

# Acronyms

AR	–	Autoregressive
CD	–	Compact Disk
DWT	–	Discrete Wavelet Transform
EM	–	Expectation-Maximisation
FIR	–	Finite Impulse Response
GMM	–	Gaussian Mixture Model
HMM	–	Hidden Markov Model
HMT	–	Hidden Markov Tree
HPF	–	Highpass Filter
IDWT	–	Inverse Discrete Wavelet Transform
iid	–	independent and identically distributed
IM	–	Independent Mixture
ISTFT	–	Inverse Short-Time Fourier Transform
LP	–	Linear Prediction
LPC	–	Linear Prediction Coding
LPF	–	Lowpass Filter
LSA	–	Log-Spectral Amplitude
MMSE	–	Minimum Mean-Square Error
MSE	–	Mean-square error
pdf	–	Probability Density Function
pmf	–	Probability Mass Function
SNR	–	Signal-to-Noise Ratio
STFT	–	Short-Time Fourier Transform
STSA	–	Short-Time Spectral Attenuation
SURE	–	Stein's Unbiased Risk Estimate
uHMT	–	Universal Hidden Markov Tree
WGN	–	White Gaussian Noise

# List of Symbols

## General

$x[n]$	–	discrete-time clean frame
$d[n]$	–	discrete-time noise frame
$y[n]$	–	discrete-time noisy frame
$\hat{x}[n]$	–	discrete-time denoised/enhanced frame
$n$	–	discrete-time index counter
$N$	–	length of analysis frame
$F_S$	–	sampling frequency
$\sigma_d^2$	–	noise variance
$\hat{\sigma}_d^2$	–	estimated noise variance
$P(\cdot)$	–	probability mass function (pmf)
$f(\cdot)$	–	probability density function (pdf)



## Short-time spectral attenuation (STSA)

$X_k$	–	clean short-time Fourier coefficient of bin $k$
$D_k$	–	noise short-time Fourier coefficient of bin $k$
$Y_k$	–	noisy short-time Fourier coefficient of bin $k$
$k$	–	index of the frequency bins
$K$	–	number of frequency bins
$R_k$	–	magnitude component of the noisy Fourier coefficient $Y_k$
$\vartheta_k$	–	phase component of the noisy Fourier coefficient $Y_k$
$A_k$	–	magnitude component of the clean Fourier coefficient $X_k$
$\alpha_k$	–	phase component of the clean Fourier coefficient $X_k$
$\sigma_x^2(k)$	–	clean variance of spectral bin $k$
$\sigma_d^2(k)$	–	noise variance of spectral bin $k$
$\sigma_y^2(k)$	–	noisy variance of spectral bin $k$



$\xi_k$	–	<i>a priori</i> signal-to-noise ratio
$\gamma_k$	–	<i>a posteriori</i> signal-to-noise ratio
$\nu_k$	–	intermediate variable containing $\xi_k$ and $\gamma_k$
$ \hat{X}_k $	–	spectral estimate of bin $k$ of the current frame
$ \hat{X}_k^{\text{pf}} $	–	spectral estimate of bin $k$ of the previous frame
$E[\cdot]$	–	expected value
$H_k$	–	suppression rule for bin $k$
$I_0(\cdot)$	–	Bessel function of the first kind of order 0
$I_1(\cdot)$	–	Bessel function of the first kind of order 1
$\alpha$	–	forgetting/weighting factor for the Ephraim-Malah decision-directed $\xi_k$ estimate

## Wavelet theory and filter bank design

$L_D / H_0$	–	lowpass decomposition wavelet filter
$H_D / F_0$	–	highpass decomposition wavelet filter
$L_R / H_1$	–	lowpass reconstruction wavelet filter
$H_R / F_1$	–	highpass reconstruction wavelet filter
$L$	–	length of a wavelet filter
$h_i$	–	filter coefficients of $H_0$
$f_i$	–	filter coefficients of $F_0$
$K$	–	delay in terms of samples
$P(z)$	–	polynomial used for spectral factorisation
$P_m(x)$	–	polynomial used to find $P(z)$
$m$	–	Herrmann order
$N$	–	total number of wavelet coefficients
$i$	–	index to wavelet coefficients
$j$	–	index to resolution levels
$N_j$	–	number of wavelet coefficients in resolution level $j$
$J$	–	number of decomposition levels
$J_{MAX}$	–	maximum number of decomposition levels
$J'$	–	effective number of resolution levels
$j_0$	–	low resolution cut-off level
$T$	–	number of binary trees of coefficients
$\psi_i$	–	wavelet atom (basis function)

## Wavelet-based denoising

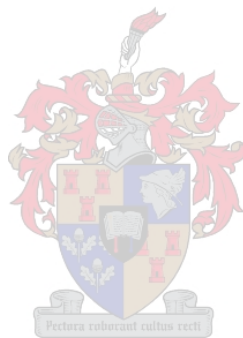
$\mathbf{x}$	– clean time signal in vector notation
$\mathbf{d}$	– noise time signal in vector notation
$\mathbf{y}$	– noisy time signal in vector notation
$\hat{\mathbf{x}}$	– enhanced time signal in vector notation
$\mathbf{z} / \mathbf{z}^*$	– zero-mean, unity variance white Gaussian noise
$\mathbf{w}$	– wavelet function
$\boldsymbol{\theta}$	– clean wavelet coefficients
$\hat{\boldsymbol{\theta}}$	– estimated wavelet coefficients
$\mathbf{W}$	– orthogonal matrix for the DWT
$\Theta^H(\mathbf{w})$	– hard shrinkage function
$\Theta^S(\mathbf{w})$	– soft shrinkage function
$\Theta^{1L}(\mathbf{w})$	– one-slope shrinkage function
$\Theta^{2L}(\mathbf{w})$	– two-slope shrinkage function
$\lambda$	– threshold for shrinkage functions
$\sigma_{j;y}^2$	– noisy variance of resolution level $j$
$w_i$	– individual wavelet coefficient
$s_i$	– hidden state variable associated with $w_i$
$M$	– number of possible states
$\mathcal{M}$	– GMM, HMM and HMT model parameter vector
$\mathcal{M}_y$	– noisy GMM, HMM and HMT model parameter vector
$\mathcal{M}^k$	– parameter vector of the current frame $k$
$\mathcal{M}^{k-1}$	– parameter vector of the previous frame $k - 1$
$\sigma_S^2$	– small variance
$\sigma_L^2$	– large variance
$\sigma_{j;S}^2$	– small clean variance of resolution level $j$
$\sigma_{j;L}^2$	– large clean variance of resolution level $j$
$\sigma_{j;S;y}^2$	– small noisy variance of resolution level $j$
$\sigma_{j;L;y}^2$	– large noisy variance of resolution level $j$
$\hat{\sigma}_{j;d}^2$	– estimated noise variance of resolution level $j$
$\sigma_{j;d}^2$	– actual noise variance of resolution level $j$
$\sigma_{j;x}^2$	– variance of clean coefficients of resolution level $j$
$P_S$	– probability to be in a small state
$P_L$	– probability to be in a large state

- $\epsilon_{i,p(i)}^{mr}$  – state transition probability
- $\pi(m)$  – initial probability
- $\sigma_{i,m}^2$  – variance parameter of coefficient  $w_i$  and state  $m$
- $p(i)$  – indices of the parent coefficient of  $w_i$
- $c(i)$  – indices of the children coefficient of  $w_i$
- $T_i$  – subtree containing  $w_i$  and all its descendants
- $T_{p(i)}$  – subtree containing  $w_{p(i)}$  and all its descendants
- $T_{p(i)\setminus i}$  – set of coefficients obtained by removing the subtree  $T_i$  from  $T_{p(i)}$
- $\alpha_i(m)$  – upward variable for HMT training
- $\beta_i(m)$  – downward variable for HMT training
- $\beta$  – noise floor parameter
- $\beta^W$  – wavelet-based noise floor parameter
- $\beta^F$  – Fourier-based noise floor parameter



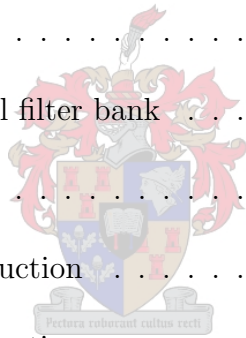
# Contents

<b>Declaration</b>	<b>i</b>
<b>Abstract</b>	<b>ii</b>
<b>Opsomming</b>	<b>iii</b>
<b>Acknowledgements</b>	<b>v</b>
<b>Acronyms</b>	<b>vi</b>
<b>List of Symbols</b>	<b>vii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 The denoising problem . . . . .	1
1.2 A generalised denoising system . . . . .	2
1.2.1 Noise estimation . . . . .	2
1.2.2 The denoising algorithm . . . . .	3
1.3 Literature study . . . . .	4
1.3.1 Fourier-based speech enhancement . . . . .	5
1.3.2 Wavelet-based signal/image denoising . . . . .	6
1.3.3 Wavelet-based speech enhancement . . . . .	8



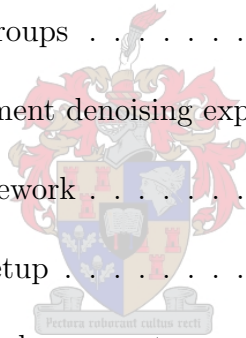
1.3.4	This study in perspective . . . . .	10
1.4	Objectives . . . . .	11
1.5	Contributions . . . . .	12
1.6	Overview of this study . . . . .	13
1.6.1	Theory . . . . .	13
1.6.2	Experiments . . . . .	13
<b>2</b>	<b>Current STSA speech enhancement</b>	<b>14</b>
2.1	Short-time spectral attenuation (STSA) . . . . .	14
2.1.1	Forward transformation . . . . .	15
2.1.2	Attenuation step . . . . .	15
2.1.3	Inverse transformation . . . . .	16
2.2	Attenuation . . . . .	16
2.2.1	The different STSA algorithms . . . . .	16
2.2.2	Estimating the <i>a priori</i> signal-to-noise ratio . . . . .	19
<b>3</b>	<b>Evaluation of speech enhancement</b>	<b>21</b>
3.1	Introduction . . . . .	21
3.2	Objective quality measures . . . . .	22
3.2.1	Segmental signal-to-noise ratio $d_{SEGSNR}$ . . . . .	24
3.2.2	Itakura-Saito distortion measure $d_{IS}$ . . . . .	25
3.3	Subjective listening tests . . . . .	26
3.3.1	Informal listening tests . . . . .	26
3.3.2	Formal listening tests . . . . .	26
3.4	Denosing artifacts . . . . .	26

3.4.1	Musical noise . . . . .	27
3.4.2	Speech distortion . . . . .	27
3.4.3	The trade-off . . . . .	27
<b>4</b>	<b>Wavelet theory and filter bank design</b>	<b>28</b>
4.1	Introduction . . . . .	28
4.2	Wavelet filter banks . . . . .	28
4.3	Designing the wavelet filters . . . . .	29
4.3.1	The wavelet filters . . . . .	29
4.3.2	The downsampler . . . . .	30
4.3.3	The upsampler . . . . .	30
4.3.4	The two-channel filter bank . . . . .	31
4.3.5	Regularity . . . . .	32
4.3.6	Perfect reconstruction . . . . .	32
4.3.7	Spectral Factorisation . . . . .	34
4.4	Decomposition levels . . . . .	43
4.4.1	Full wavelet decomposition . . . . .	43
4.4.2	$J$ -level decomposition . . . . .	44
4.4.3	The decomposition of speech . . . . .	45
4.5	Statistical properties of the DWT . . . . .	46
<b>5</b>	<b>Wavelet-based signal denoising</b>	<b>50</b>
5.1	General signal denoising . . . . .	50
5.1.1	Forward transformation . . . . .	51
5.1.2	Attenuation step . . . . .	51



5.1.3	Inverse transformation . . . . .	52
5.2	Attenuation . . . . .	52
5.3	The shrinkage functions . . . . .	53
5.3.1	Using the shrinkage functions . . . . .	56
5.4	VisuShrink . . . . .	56
5.5	SureShrink and HybridSure . . . . .	57
5.6	Wavelet-based Wiener denoising . . . . .	58
5.7	Statistical models in the wavelet domain . . . . .	59
5.7.1	The hidden state variable . . . . .	60
5.7.2	The low-resolution cut-off level $j_0$ . . . . .	60
5.8	Gaussian Mixture Models (GMMs) . . . . .	60
5.8.1	The GMM structure . . . . .	60
5.8.2	Modelling the GMM non-Gaussianity . . . . .	62
5.8.3	The GMM model parameters . . . . .	63
5.8.4	Training the GMM . . . . .	64
5.8.5	GMM denoising . . . . .	67
5.9	Hidden Markov Models (HMMs) . . . . .	68
5.9.1	The HMM structure . . . . .	68
5.9.2	Modelling the HMM non-Gaussianity . . . . .	69
5.9.3	The HMM model parameters . . . . .	69
5.9.4	Training the HMM . . . . .	70
5.9.5	HMM denoising . . . . .	74
5.10	Hidden Markov Trees (HMTs) . . . . .	74

5.10.1	The HMT structure . . . . .	74
5.10.2	Modelling the HMT non-Gaussianity . . . . .	76
5.10.3	The HMT model parameters . . . . .	77
5.10.4	HMT training via the EM algorithm . . . . .	78
5.10.5	HMT denoising . . . . .	84
5.11	Performance comparison of wavelet denoising algorithms . . . . .	84
<b>6</b>	<b>Wavelet-based speech enhancement</b>	<b>89</b>
6.1	Introduction . . . . .	89
6.2	Denoising of speech segments . . . . .	89
6.2.1	The phoneme groups . . . . .	90
6.2.2	The speech-segment denoising experiment . . . . .	93
6.3	The experimental framework . . . . .	97
6.3.1	Experimental setup . . . . .	98
6.4	Wavelet-based speech enhancement experiments . . . . .	100
6.4.1	Objective evaluation . . . . .	101
6.4.2	Subjective evaluation . . . . .	102
6.5	The $d_{IS}$ problem segments . . . . .	104
6.5.1	The spectral floor parameter $\beta$ . . . . .	107
6.5.2	Objective evaluation of the floor parameter . . . . .	108
6.5.3	LPC evaluation of the floor parameter . . . . .	111
6.5.4	Subjective evaluation of the floor parameter . . . . .	112
6.6	Choosing a good wavelet . . . . .	113
6.6.1	The Itakura-Saito distortion ( $d_{IS}$ ) evaluation . . . . .	115





6.6.2	The segmental signal-to-noise ratio ( $d_{SEGSNR}$ ) evaluation . . . . .	116
6.6.3	Subjective evaluation . . . . .	118
6.6.4	A good wavelet for speech . . . . .	119
6.7	Choosing the best frame size . . . . .	120
6.8	Choosing the best algorithm . . . . .	122
6.8.1	Objective evaluation . . . . .	122
6.8.2	Formal subjective evaluation . . . . .	125
6.9	Conclusions . . . . .	126
6.9.1	Denoising of speech-segments . . . . .	126
6.9.2	The noise floor parameter . . . . .	126
6.9.3	The wavelet . . . . .	127
6.9.4	The frame size . . . . .	127
6.9.5	Comparing HMT, HMM, GMM and Wiener speech enhancement . . . . .	128
<b>7</b>	<b>Comparing STSA with HMTs on speech</b>	<b>129</b>
7.1	Introduction . . . . .	129
7.2	The Ephraim-Malah algorithm . . . . .	130
7.2.1	Objective evaluation of the Ephraim-Malah algorithm . . . . .	130
7.2.2	Subjective evaluation of the Ephraim-Malah algorithm . . . . .	133
7.3	Comparing Fourier-based and wavelet-based speech enhancement . . . . .	134
7.3.1	Global objective measures . . . . .	135
7.3.2	Phoneme class objective measures . . . . .	138
7.3.3	Subjective evaluation . . . . .	140
7.3.4	Formal subjective evaluation . . . . .	141



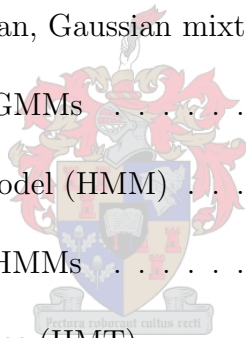
7.4	Conclusions . . . . .	142
<b>8</b>	<b>Conclusions</b>	<b>144</b>
8.1	Conclusions of this study . . . . .	144
8.2	Speech enhancement vs image denoising . . . . .	145
8.2.1	The training data of speech vs images . . . . .	146
8.2.2	Statistical properties of speech vs images . . . . .	146
8.2.3	The typical global SNR of speech vs images . . . . .	148
8.3	Future research . . . . .	149
8.3.1	Domain recommendations . . . . .	149
8.3.2	Sampling rate . . . . .	149
8.3.3	Clusters of speech . . . . .	150
8.3.4	Pitch tracking . . . . .	150
<b>A</b>	<b>The Itakura-Saito distortion measure</b>	<b>157</b>
<b>B</b>	<b>The TIMIT database</b>	<b>160</b>
B.1	The TIMIT192WGN core test set . . . . .	160
B.2	The TIMIT24WGN training set . . . . .	161
B.3	Informal listening tests . . . . .	161
B.4	Formal listening tests . . . . .	161



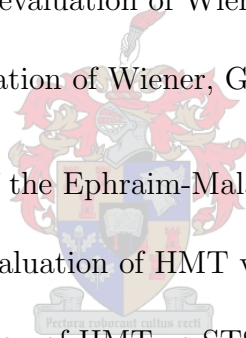
# List of Figures

1.1	The block diagram of signal enhancement . . . . .	2
1.2	The flowchart of a generalised denoising system . . . . .	4
2.1	The flowchart of STSA speech enhancement . . . . .	14
2.2	The flowchart of the STSA attenuation step . . . . .	17
3.1	The flowchart of objective speech enhancement evaluation . . . . .	22
3.2	The $d_{SEGSNR}$ distortion measure over time . . . . .	23
4.1	The DWT and IDWT filter banks . . . . .	29
4.2	The FIR filter . . . . .	29
4.3	The downsampler . . . . .	30
4.4	The upsampler . . . . .	30
4.5	The two-channel filter bank . . . . .	31
4.6	The new two-channel filter bank . . . . .	33
4.7	The pole-zero plot of a maximally flat symmetric lowpass filter . . . . .	36
4.8	The Daubechies 6 wavelet filters in the $z$ -plane . . . . .	37
4.9	The Biorthogonal 1 wavelet filters in the $z$ -plane . . . . .	39
4.10	The Symlet 6 wavelet filters in the $z$ -plane . . . . .	40
4.11	The Haar wavelet filters in the $z$ -plane . . . . .	41

4.12	The magnitude response of different wavelet filters . . . . .	42
4.13	The full wavelet decomposition . . . . .	44
4.14	A two-level wavelet decomposition . . . . .	45
4.15	The different views of the wavelet transform . . . . .	47
5.1	The flowchart of wavelet-based signal denoising . . . . .	50
5.2	The flowchart of the wavelet-based attenuation step . . . . .	52
5.3	The hard, soft, one-slope and two-slope shrinkage functions . . . . .	54
5.4	The hidden state variable $s_i$ and wavelet coefficient $w_i$ . . . . .	60
5.5	The Gaussian Mixture Model (GMM) . . . . .	61
5.6	The two-state, zero-mean, Gaussian mixture . . . . .	63
5.7	The EM flowchart for GMMs . . . . .	65
5.8	The Hidden Markov Model (HMM) . . . . .	69
5.9	The EM flowchart for HMMs . . . . .	71
5.10	The Hidden Markov Tree (HMT) . . . . .	75
5.11	The binary tree of connected state variables . . . . .	76
5.12	The EM flowchart for HMTs . . . . .	78
5.13	The different sets of wavelet coefficients used for HMT training . . . . .	80
5.14	Bumps and its wavelet coefficients . . . . .	85
5.15	Blocks and its wavelet coefficients . . . . .	85
5.16	Doppler and its wavelet coefficients . . . . .	85
5.17	HeaviSine and its wavelet coefficients . . . . .	86
6.1	Vowels in the time domain and wavelet domain . . . . .	91
6.2	Nasals in the time domain and wavelet domain . . . . .	92



6.3	Semivowels in the time domain and wavelet domain . . . . .	92
6.4	Stops in the time domain and wavelet domain . . . . .	92
6.5	Fricatives in the time domain and wavelet domain . . . . .	93
6.6	The Itakura-Saito problem segments . . . . .	104
6.7	The LPC responses and wavelet functions of a problem segment . . . . .	105
6.8	The global distortion measures for different values of $\beta$ . . . . .	109
6.9	The LPC response of a problem segment using a noise floor . . . . .	112
6.10	The $d_{IS}$ evaluation of different wavelets . . . . .	115
6.11	The $d_{SEGSNR}$ evaluation of different wavelets . . . . .	117
6.12	Comparative $d_{SEGSNR}$ evaluation of Wiener, GMM, HMM and HMM . . .	123
6.13	Comparative $d_{IS}$ evaluation of Wiener, GMM, HMM and HMM . . . . .	124
7.1	Objective evaluation of the Ephraim-Malah algorithm . . . . .	131
7.2	The global $d_{SEGSNR}$ evaluation of HMT vs STSA . . . . .	136
7.3	The global $d_{IS}$ evaluation of HMT vs STSA . . . . .	137
8.1	The Blocks signal compared to voiced speech . . . . .	147
8.2	The two-state ergodic model of the HMM . . . . .	148



# List of Tables

4.1	Maximally flat symmetric halfband filters . . . . .	35
5.1	The Donoho-Johnstone denoising experiment . . . . .	86
6.1	The five different phoneme groups with their TIMIT labels . . . . .	91
6.2	The $d_{MSE}$ evaluation of speech segments . . . . .	94
6.3	The $d_{SEGSNR}$ evaluation of speech segments . . . . .	95
6.4	The $d_{IS}$ evaluation of speech segments . . . . .	96
6.5	The $d_{SEGSNR}$ evaluation of the speech-only sections . . . . .	101
6.6	The $d_{IS}$ evaluation of the different phoneme groups . . . . .	102
6.7	The $d_{IS}$ evaluation using different noise floors . . . . .	111
6.8	The $d_{SEGSNR}$ evaluation using different noise floors . . . . .	111
6.9	The $d_{SEGSNR}$ evaluation using different frame sizes . . . . .	121
6.10	Statistical properties of the different algorithms . . . . .	122
6.11	First formal listening test results . . . . .	125
7.1	The $d_{SEGSNR}$ evaluation of Ephraim-Malah vs HMT . . . . .	139
7.2	The $d_{IS}$ evaluation of HMT vs STSA . . . . .	139
7.3	Second formal listening test results . . . . .	142
B.1	The <i>TIMIT core test set</i> . . . . .	160

B.2 The *TIMIT training set* . . . . . 161



# Chapter 1

## Introduction

Speech enhancement is the process of removing background noise from speech signals. This noise can vary from light microphone noise to the heavy background noise of speech in windy conditions. A lot of research has been done on developing different speech enhancement algorithms, most of these in the Fourier domain [7, 9, 23, 24, 40, 42, 55, 58, 59, 60], of which [60] gives a basic overview.

Image denoising is a very similar process, where noise, such as speckle, is removed from an image. Wavelet-based image denoising [11, 14, 18, 19, 20, 21, 49, 48, 47] has proven to be very successful.

Little research has been done on wavelet-based speech enhancement, all of which is very recent [3, 4, 13, 27, 35, 50], yet it shows promising results when compared to Fourier-based methods. None of these algorithms explicitly attempt to capture the statistical properties of the wavelet coefficients of speech.

The aim of this research project is to investigate wavelet-based speech enhancement, specifically from a statistical point of view, and then to compare this with Fourier-based speech enhancement.

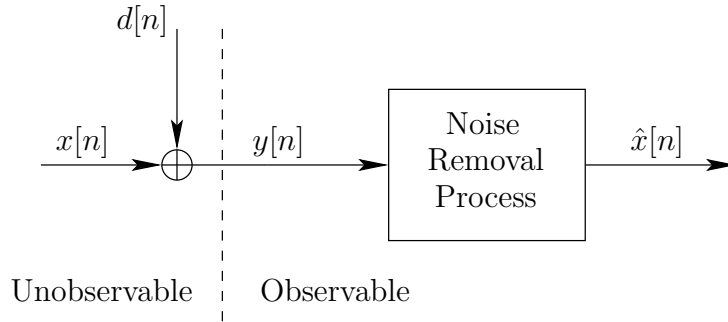
### 1.1 The denoising problem

A basic understanding of the general denoising problem is first required. Let  $x[n]$  be a discrete-time *clean* signal and  $d[n]$  a *noise* signal. If the noise is considered to be **additive**, then the *noisy* signal is represented by the additive observation model,

$$y[n] = x[n] + d[n] . \tag{1.1}$$



Figure 1.1 shows how  $y[n]$  represents the *observed* noisy signal at time index  $n$ ,  $x[n]$  represents the *unobserved* clean signal and  $d[n]$  represents the noise, uncorrelated with the clean signal. The goal of the *noise removal process* is to form an *estimate*  $\hat{x}[n]$  of the clean signal  $x[n]$  based on the observed signal  $y[n]$ .



**Figure 1.1:** The block diagram of signal enhancement in the case of additive noise.

The noise removal process is generally called *signal denoising*. It is also called an *estimator*, because it forms an estimate  $\hat{x}[n]$  of the underlying signal  $x[n]$ . In the case where  $x[n]$  is a speech signal, the noise removal process is referred to as *speech enhancement*.

## 1.2 A generalised denoising system



Any denoising system consists of two basic parts, namely a noise estimation process and a denoising algorithm, and they are described below.

### 1.2.1 Noise estimation

In most real-world problems, the noise signal is not directly known and has to be estimated. In image denoising, the noise has to be estimated from the noisy image itself. In speech enhancement, the noise is estimated from the portions of the sound recording which do not contain speech and therefore only consist of noise. Noise estimation in speech is therefore less of a problem than in images and is also more accurate. The better the noise estimate, the better the performance of the denoising system will be.

There are many types of noises that occur in real-world speech enhancement problems. Examples include the noise inside a car, helicopter or aeroplane cockpit, the noise inside an office or factory, the noise of a cooling or heating fan, and even the noise from

other speakers in the vicinity of the speaker under analysis. Several recordings of real-world noise sources have been made and are used for standard speech enhancement tests. These recordings are readily available on the Internet [30, 54] and include white Gaussian, speech babble (recordings of multiple speakers speaking simultaneously), car, helicopter, F16 cockpit, factory and office noises. The noise that occur in real-world problems is generally broadband in nature, implying that it is localised in neither time nor frequency and therefore difficult to remove [57]. Most research is done on the enhancement of speech corrupted by broadband noise, of which **White Gaussian noise** (WGN) is a good example.

If the noise is **stationary** (i.e. if its statistical character does not change over time), it follows that its estimated spectrum is constant over time. If it is non-stationary but changes its characteristics relatively slowly, it can be modelled as quasi-stationary. The noise is hereby assumed to be stationary within the time-span of two consecutive noise spectral estimates.

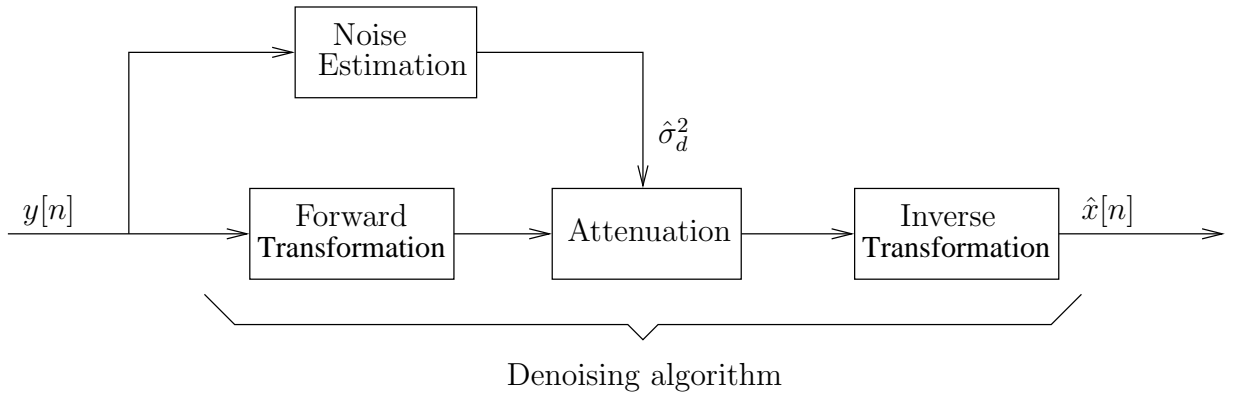
The noise in this research is therefore assumed to be additive, independent and identically distributed, stationary and white Gaussian, which are conditions typically used in most speech enhancement research.

The noise estimation process usually uses an algorithm that estimates the noise spectrum. For the purpose of this study, white Gaussian noise is generated and then added to the clean signal to produce the noisy signal. The noise spectrum can therefore be directly calculated from the noise, instead of being estimated from the real-world data. The desired global signal-to-noise ratio, which is of the form  $10 \log_{10} (\sigma_x^2 / \sigma_d^2)$  in decibels, is first specified. By scaling the noise to unity variance and therefore setting  $\sigma_d^2 = 1$ , the variance of the clean signal  $\sigma_x^2$  is computed and then used to scale the clean signal. Adding these scaled signals, as in Figure 1.1, the noisy signal then has the required global signal-to-noise ratio. This removes the effect of the noise estimation algorithm on the final result, and thereby focuses attention on the performance of the denoising algorithm.

## 1.2.2 The denoising algorithm

Figure 1.2 shows the flowchart of the noise removal process. The denoising algorithm is the basic mechanism of denoising. It relies on the noise estimate and is comprised of three parts:

1. **Forward transformation.** In the forward transformation step, the noisy signal  $y[n]$  is transformed into coefficients of a certain domain. Current state-of-the-art



**Figure 1.2:** *The flowchart of a generalised denoising system.*

image denoising is done in the wavelet domain [11, 14, 18, 19, 20, 21, 49, 48, 47], whereas current speech enhancement is generally done in the short-time Fourier domain [7, 9, 23, 24, 40, 42, 55, 58, 59, 60], although recent research has been done in the wavelet domain [50], the wavelet packet domain [3, 4, 13, 27] and the multitaper spectral domain [35].

2. **Attenuation.** The attenuation step is where the actual denoising is done. The noisy coefficients are attenuated by using a *suppression rule* (in Fourier-based speech enhancement) or a *shrinkage function* (in wavelet-based denoising) to form an estimate of the coefficients of the clean signal. This makes use of the noise estimate.
3. **Inverse transformation.** The inverse transform of the attenuated coefficients renders the estimated clean signal  $\hat{x}[n]$ .

### 1.3 Literature study

This study investigates wavelet-based speech enhancement algorithms and compares them with Fourier-based speech enhancement algorithms. It therefore requires knowledge of the following fields of research:

- Fourier-based speech enhancement.
- Wavelet-based signal/image denoising.
- Wavelet-based speech enhancement.

A short description of these fields is given below as well as a brief history of each, which highlights a selection of important papers in each domain.

### 1.3.1 Fourier-based speech enhancement

Speech enhancement algorithms make the assumption that speech is quasi-stationary, i.e. stationary within a short time-frame of analysis [36]. Speech is therefore denoised on a timeframe-by-timeframe basis. Each time-frame is transformed to the Fourier domain where the Fourier coefficients represent the signal as a number of frequency bins. Each bin is then classified as containing either signal or noise. If the bin predominantly represents the underlying signal, it is left unattenuated. If it contains mainly noise, it is shrunk towards zero. The inverse of this frame-by-frame Fourier transform produces the enhanced speech. A residual noise artifact typically encountered in Fourier-based speech enhancement is the so-called “musical noise” artifact [9]. Musical noise consists of tonal components at random frequencies. It has an unnatural structure and is perceptually annoying [55]. Important papers of Fourier-based speech enhancement are given below.

- 1978 — Lim and Oppenheim [40] proposed a speech enhancement method based on an iterative estimation of all-pole speech parameters. It uses a maximum *a posteriori* (MAP) estimate under the assumption that the speech signal is the response of an all-pole process.
- 1984 — Ephraim and Malah [23] derived a minimum mean-square error estimator (the MMSE STSA algorithm) as an extension of the maximum likelihood estimator of McAulay and Malpass [42]. It assumes that the Fourier coefficients of the clean signal and the noise may be modelled as statistically independent, zero-mean, Gaussian random variables.
- 1985 — Ephraim and Malah [24] derived the minimum mean-squared error log-spectral amplitude estimator (the MMSE-LSA algorithm). This algorithm is similar to [23], except that it minimises the mean-squared error of the log-spectra, instead of the spectra.
- 1991 — Hansen and Clements [31] further enhanced the all-pole model of Lim and Oppenheim [40] by introducing spectral constraints to ensure more speech-like formant trajectories.

- 1994 — Cappé [9] presented a study of the Ephraim-Malah MMSE STSA algorithm [23], demonstrating how this algorithm succeeds in eliminating the “musical noise” phenomenon.
- 1999 — Virag [55] proposed a subtractive-type algorithm which is based on masking properties of the human auditory system. It leads to a significant reduction of the unnatural structure of the residual noise.
- 2001 — Wolfe and Godsill [58, 59, 60] proposed three alternative suppression rules to the Ephraim-Malah suppression rule by using alternative Bayesian approaches. These suppression rules exhibit almost identical behaviour to that of the Ephraim-Malah suppression rule, but are computationally more efficient and yield a more intuitive interpretation.

### 1.3.2 Wavelet-based signal/image denoising

Unlike the Fourier transform, which represents the signal in frequency bins, the wavelet transform yields a multiresolution representation of the signal with fine frequency resolution at low frequencies and fine time resolution at high frequencies. This represents real-world signals such as images more compactly. The idea behind wavelet-based denoising is similar to that of Fourier-based speech enhancement, as coefficients are classified as representing either signal or noise, and attenuated accordingly. A brief history of wavelet-based denoising follows below.

- 1992 — Donoho and Johnstone [18] proposed wavelet shrinkage in the form of the RiskShrink algorithm. A mean-squared error (MSE) or “risk” approach is taken to obtain a threshold value for the soft shrinkage function (see Section 5.3). Wavelet coefficients with values above this threshold are attenuated only a little, whereas coefficients below this threshold are shrunk to zero.
- 1992 — Donoho and Johnstone [18] also proposed the VisuShrink algorithm, which uses the “universal” threshold for the soft shrinkage function. This threshold is a function of the signal length. VisuShrink results in an almost “noise-free” reconstruction, which is visually very smooth on images.

- 1994 — Donoho and Johnstone [19] proposed the SureShrink and HybridSure algorithms. Stein’s Unbiased Risk Estimate (SURE) [26] is computed for each possible threshold value. SureShrink uses the threshold that minimises this risk. HybridSure, which is specifically designed for signals with sparse wavelet coefficients, uses a combination of SureShrink and VisuShrink.
- 1997 — Chipman, Kolaczyk and McCulloch [11] proposed an algorithm which is a wavelet shrinkage approach that uses Bayesian priors. It is based on the “compression” property of wavelet coefficients, which implies that wavelet coefficients tend to have a non-Gaussian distribution. The prior of each coefficient consists of a mixture of two Gaussian distributions with different standard deviations. The parameters are chosen adaptively according to the resolution level of the coefficients, typically shrinking high resolution (frequency) coefficients more heavily.
- 1998 — Crause, Nowak and Baraniuk [14] proposed the Hidden Markov Tree (HMT) algorithm. They identified two “secondary” properties of wavelet coefficients of real-world signals, namely clustering and persistence, which imply that adjacent coefficients tend to have similar values. The HMT algorithm uses a two-state, zero-mean tree-structured Hidden Markov Model framework to capture the non-Gaussian statistics of the individual coefficients. This is similar to [11], but also captures the inter-coefficient dependencies (clustering and persistence). Crause et al. report superior denoising performance over the above-mentioned algorithms. The algorithm, however, suffers from a large number of model parameters and uses a computationally intensive Expectation-Maximisation algorithm.
- 1999 — Romberg [47] introduced a simpler model than the standard HMT algorithm [14], that attempts to capture the same statistical properties. It uses even further “tertiary” properties of wavelet coefficients of images, namely exponential decay across scale and strong persistence at finer scales. Within this framework, Romberg proposed an algorithm that uses a fixed set of parameters for the denoising of normalised grey-scale images. This is referred to as the universal Hidden Markov Tree (uHMT) algorithm. It produces results similar to the HMT algorithm on images, in spite of its comparative simplicity.

- 1999 — Wavelet-based image denoising frequently exhibit visual artifacts, usually in the form of “ringing” around edges. Ringing typically occurs when excessively long wavelet filters are used. Romberg, Choi and Baraniuk [49] proposed a more computationally intensive shift-invariant version of the uHMT, which uses circular rotation to reduce the ringing artifact.
- 2002 — Romberg, Choi, Baraniuk and Kingsbury [48] proposed using the HMT algorithm [14] on the complex wavelet transform. The complex wavelet transform has near shift-invariance and an improved angular resolution over the discrete wavelet transform. This method outperforms even the computationally expensive redundant uHMT algorithm [49], owing to its underlying transform.

### 1.3.3 Wavelet-based speech enhancement

Speech can be divided into two very different types of signals, namely voiced speech, such as vowels, and unvoiced speech, such as consonants [36]. Because voiced speech is produced by the oscillation of the vocal chords it is periodic in nature. The Fourier domain is well suited for such signals, and is widely used in speech applications such as phoneme recognition. Unvoiced sounds, however, are generally not periodic in nature and the Fourier domain may not be the best way to model such signals for denoising purposes. The success of wavelet-based signal/image denoising has led researchers to investigate the potential of wavelet-based speech enhancement which include using either the wavelet transform or the wavelet packet transform. The latter decomposes the signal into a larger number of subbands and produces a multiresolution framework that can have finer frequency resolution at high frequencies than the standard wavelet-transform [56]. Wavelet-based speech enhancement is similar to Fourier-based speech enhancement, but instead of calculating the Fourier transform of every consecutive frame, the wavelet transform is used. A selection of important wavelet-based speech enhancement papers is given below.

- 1997 — Seok and Bae [50] proposed a speech enhancement algorithm which thresholds the coefficients of speech in the wavelet domain. Thresholding speech in the wavelet domain can easily eliminate sections of speech, though, especially when denoising the noise-like unvoiced sounds [3, 4, 50]. The algorithm uses voiced/unvoiced detection to solve this problem. Unvoiced sections of speech are denoised by only attenuating the coefficients of the highest resolution level, whereas all coefficients are attenuated with voiced sounds. Seok and Bae report promising results on the cepstral distance distortion measure, despite the simplicity of the algorithm.
- 2001 — Bahoara and Rouat [3, 4] proposed a novel speech enhancement algorithm by using a time-adaptive threshold in a 16-subband uniform wavelet packet domain. The threshold is computed by applying an approximated Teager energy operator on the wavelet packet coefficients. The Teager energy operator is a nonlinear operator capable to extract the signal energy based on mechanical and physical considerations. This operator enhances coefficients that represent signal information among those that represent noise. This function is then modified to compute time-adaptive thresholds. Bahoara and Rouat report that their algorithm improves the global SNR more than the Ephraim-Malah MMSE STSA algorithm [23], even under heavy noise conditions.
- 2001 — Cohen [13] proposed an algorithm which uses a weighted Wiener filter to attenuate the coefficients of a non-uniform 84-subband redundant wavelet packet transform. The subband spacing approximates the bark frequency scale, which is a perceptual frequency scale generally used for audio compression purposes. The *a priori* SNR is estimated by a variation of the Ephraim-Malah decision-directed estimate [23]. Compared to Fourier-based speech enhancement, the algorithm leads to better results on the segmental signal-to-noise ratio distortion measure [33] and lower residual noise of enhanced speech.



- 2003 — Fu and Wan [27] proposed a method which uses Fourier-based and wavelet-based denoising techniques in a series combination. The Ephraim-Malah MMSE STSA speech enhancement algorithm [23] is used as a pre-processing step to eliminate some noise while still retaining speech quality. This enhanced speech signal is then transformed into the wavelet packet domain by using an 18-subband critical-band decomposition, similar to the decomposition in [13]. Time- and frequency-adaptive thresholds are computed for each subband and time frame by using a variation of the universal threshold (see Section 5.4). Denoising is done with a variation of the Ephraim Malah suppression rule [23]. Fu and Wan state that combining Fourier-based and wavelet-based denoising techniques eliminates a reasonable amount of “musical” noise while still retaining speech quality. The algorithm also shows promising results on the segmental signal-to-noise ratio distortion measure [33].
- 2003 — Hu and Loizou [35] proposed a different approach which also combines short-time spectral attenuation (STSA) and wavelet-based denoising techniques. Unlike the above-mentioned wavelet-based algorithms [3, 4, 13, 27, 50], which threshold the wavelet coefficients of the time signal, this algorithm denoises the log multitaper spectra [53]. The multitaper spectra have good bias and variance properties [53]. These spectral signals are then transformed to the wavelet domain, denoised with SureShrink [26] (see Section 5.5) and then finally inverse transformed back into the log multitaper spectral domain. Wavelet denoising of the log multitaper spectra leads to even better (low-variance) spectral estimates. These refined spectra are then used in an STSA speech enhancement algorithm, which is a variation of Wiener filtering (see Section 2.2.1). The actual speech denoising is done in the multitaper spectral domain, whereas the wavelet-based denoising step is only used to get more refined spectral estimates, which makes this algorithm an STSA speech enhancement algorithm. Hu and Loizou showed that their algorithm has little “musical” noise and it also preserves speech quality better than the Ephraim-Malah MMSE-LSA algorithm [24].

### 1.3.4 This study in perspective

The main investigation of this study involves the Hidden Markov Tree (HMT) algorithm [14]. Since the Hidden Markov Tree algorithm is very successful in denoising the Donoho-Johnstone test set [10, 11, 14] and also in denoising images, it is of specific interest. The algorithm attempts to capture the statistical properties of the wavelet coefficients. This is something that has been exploited in wavelet-based image/signal

denoising [11, 14, 49, 48, 47] and image compression [51, 52], but not yet in speech enhancement.

The statistical properties of speech in the wavelet domain therefore need to be investigated. It is expected that certain phonemes, such as stops and voiced phonemes, have non-Gaussianity, clustering and persistence. It is not known how strong these properties are for speech. It is also of interest to what extent the HMT algorithm is capable of capturing these properties of speech signals. Other statistical techniques, namely Wiener filters [44] and Gaussian Mixture Models (GMMs) [14], are also implemented to aid the investigation. A Hidden Markov Model (HMM) denoising algorithm has been proposed by Crause et al. [14]. This algorithm is implemented in this study as a speech enhancement algorithm and it specifically attempts to capture the clusters found in wavelet coefficients. These statistical algorithms are not as sophisticated as the HMT algorithm and differ in their approach to capture some of these statistical properties. As the level of the noise increases, it reduces the presence of these properties. It is therefore expected that these statistical algorithms will not yield desirable results under heavy noise conditions.

Although most wavelet-based speech enhancement is done in the wavelet packet domain [3, 4, 13, 27], this domain does not provide a natural binary tree structure in the time-frequency tiling view, which is a requirement for the HMT algorithm. Since the Wiener, GMM and HMM methods denoise each resolution level independently, they can easily be implemented in the wavelet packet domain, which will then be closely related to [3, 4, 13, 27]. For purposes of comparison, all methods in this study are implemented in the wavelet domain. This study is therefore closely related to that of [50], although no explicit voiced/unvoiced decisions or speech presence detection is done. The thresholds are rather chosen according to the statistical information of the wavelet coefficients of each frame, which suits the frame whether it is voiced/unvoiced or speech/silence.

## 1.4 Objectives

The objectives of this study are:

- To implement the HMT for speech denoising.
- To implement a Hidden Markov Model (HMM) denoising algorithm, which attempts to capture the clustering property of wavelet coefficients.
- To develop a framework for wavelet-based speech enhancement algorithms in which the Wiener, GMM, HMM and HMT algorithms are compared to each other.

- To choose a good wavelet for speech enhancement according to objective distortion measures and informal subjective listening tests.
- To choose the best frame size for the statistical speech enhancement algorithms.
- To compare statistical wavelet-based speech enhancement algorithms with Fourier-based techniques.

## 1.5 Contributions

The following contributions are made in this study:

- HMTs are used for speech denoising for the first time.
- A novel implementation of a wavelet-based Hidden Markov Model (HMM) denoising algorithm is done. This algorithm was proposed by Crause et al. [14], but it was not implemented, nor was it used in any experiments. It is found that the HMM algorithm outperforms the state-of-the-art Hidden Markov Tree [14] algorithm on the Donoho-Johnstone *Doppler* test signal. The Doppler signal in the wavelet domain does not have strong persistence, but has a single prominent cluster within each resolution level. Although these properties are not generally found in real-world images, they are typical of seismic, radar and sonar signals. The HMM algorithm also has an advantage over the HMT algorithm in that it can easily be implemented in the wavelet packet domain, which is becoming a popular domain for wavelet-based speech enhancement.
- The choice of wavelet has an influence on the quality and residual noise of the enhanced signal. No research has been found on this subject. In this study, experiments are done to choose a good wavelet for speech enhancement according to objective distortion measures and subjective listening tests. The Discrete Meyer and higher order Symlet (Herrmann order  $m \approx 20$ ) wavelets are found to be the best wavelets for speech enhancement.
- No algorithms have been proposed that explicitly attempt to capture the statistical properties of speech in the wavelet domain. This is investigated in this study by using four similar algorithms, namely Wiener, GMM, HMM and HMT, which all attempt to capture some of these properties. It is found that these properties are not as strong in speech as in images and therefore the statistical algorithms should only be used under light noise conditions. It is however possible that these models are not sufficient to fully capture the properties of the wavelet coefficients of speech.

- Very little speech enhancement is done in the wavelet domain, because of its poor frequency resolution. Segments of speech can easily be eliminated, which leads to gaps in the speech spectrogram and hence poor speech quality. It is found in this study that this effect leads to problem segments on the Itakura-Saito distortion measure, which is addressed by introducing a noise floor parameter in the algorithms. This eliminates these problem segments and also enhances perceived speech quality.

## 1.6 Overview of this study

This study consists of a theoretical discussion and an experimental analysis.

### 1.6.1 Theory

Chapter 2 discusses Fourier-based speech enhancement. The short-time spectral attenuation (STSA) approach is currently the most widely used speech enhancement method. Chapter 3 discusses the evaluation process of speech enhancement, which includes objective distortion measures and subjective listening tests. Chapter 4 discusses wavelet theory and filter bank design. This requires knowledge of how wavelets are designed by using filter banks and the properties of the different wavelets. The statistical properties of real-world signals in the wavelet domain are also discussed here. Chapter 5 describes wavelet-based denoising methods, which include the classical wavelet shrinkage algorithms (VisuShrink, SureShrink and HybridSure) and also the statistical methods (Wiener, GMM, HMM and HMT).

### 1.6.2 Experiments

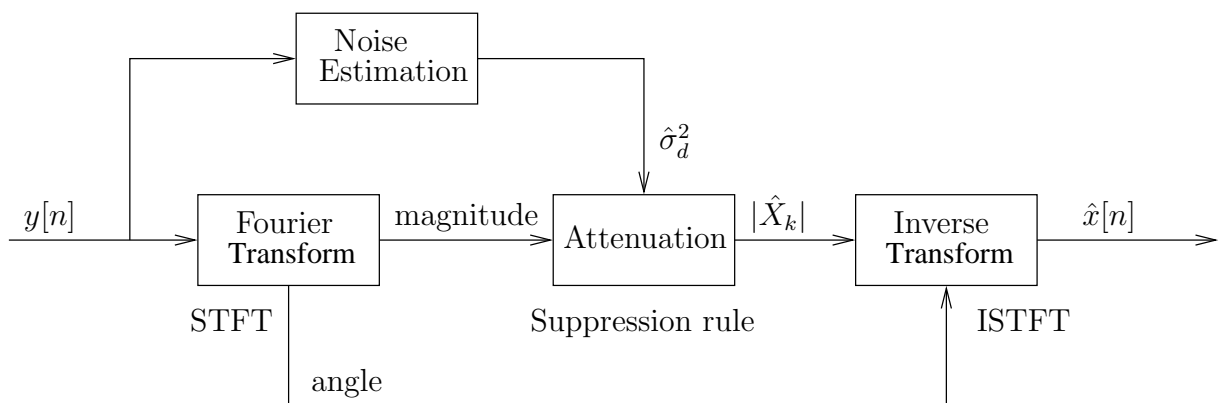
In Chapter 6 a framework is designed for wavelet-based speech enhancement in which the algorithm parameters are experimentally chosen. The Wiener, GMM, HMM and HMT algorithms are also compared to each other. In Chapter 7 an experimental comparison between wavelet-based and Fourier-based speech enhancement is done.

# Chapter 2

## Current STSA speech enhancement

### 2.1 Short-time spectral attenuation (STSA)

Short-time spectral attenuation is currently the most widely used speech enhancement technique. As described in Section 1.2, STSA consists of three steps, namely *forward transformation*, *attenuation* and *inverse transformation*. Figure 2.1 shows the flowchart of STSA speech enhancement and all such algorithms use this framework. These steps are described below and because the difference between the various STSA algorithms lies in the *attenuation* step, it is described in more detail in Section 2.2.



**Figure 2.1:** The flowchart of short-time spectral attenuation (STSA) speech enhancement. The forward transformation is the short-time Fourier transform (STFT) and the inverse transformation is the inverse short-time Fourier transform (ISTFT).

### 2.1.1 Forward transformation

In correspondence to Section 1.2.1, the noise  $d[n]$  is assumed to be additive, therefore the noisy signal is given, as in (1.1), by

$$y[n] = x[n] + d[n] . \quad (2.1)$$

STSA is Fourier-based and the forward transformation step is the *short-time Fourier transform* (STFT) of overlap-add analysis [36]. This is a process where an utterance of speech is separated into frames of short time-duration. These can be overlapping frames if a correctly-chosen time-window is multiplied by the time-frame. Each individual frame, which is assumed to be stationary, is then transformed into the Fourier domain where the analysis is done.

Because the Fourier transform is a linear transform, the coefficients  $Y_k$  can be written as [60]

$$Y_k = X_k + D_k . \quad (2.2)$$

STSA analysis is frame-based and (2.2) describes the Fourier coefficients of the current frame. These quantities are complex, with a magnitude and a phase component. The subscript  $k = 0, 1 \dots K - 1$  is an integer that indexes each of the  $K$  frequency bins associated with the Fourier coefficients. Because the Fourier transform is symmetric for real data, a length  $N$  frame results in  $K = N/2$  bins for even  $N$  and  $K = N/2 + 1$  bins for odd  $N$ .

In this research, the data has a sampling frequency of  $F_S = 8$  kHz. The chosen frame size is 32 ms, resulting in  $N = 256$  Fourier coefficients and  $K = 128$  frequency bins. Half-overlapping *Hanning* windows are used to reduce spectral leakage. These are widely used parameters [23, 55, 58, 59, 60].

### 2.1.2 Attenuation step

The attenuation step of STSA speech enhancement uses a *suppression rule* to form a spectral estimate  $|\hat{X}_k|$  of  $X_k$  by using  $|Y_k|$  and  $\hat{\sigma}_d^2$ . The attenuation step is applied to the magnitude only, leaving the phase unchanged. Ephraim and Malah [23] proved that the noisy phase  $\angle Y_k$  is the optimal output phase. The difference in STSA algorithms lies within this step and it is described in Section 2.2.

### 2.1.3 Inverse transformation

The spectral estimate  $\hat{X}_k$  is inverse-transformed to obtain the reconstruction of the time domain signal. The *inverse short-time Fourier transform* converts the Fourier coefficients of the individual time-frames back into the time-domain, whereafter they are added to create an utterance similar to the original [36]. Perfect reconstruction is possible, depending on the amount of overlapping and the time-window used.

## 2.2 Attenuation

The elements of  $X_k$  and  $D_k$  are modelled as independent, zero-mean, complex Gaussian random variables [60]. The respective *clean* and *noise* variances for the  $k$ th bin are  $\sigma_x^2(k) = E[|X_k|^2]$  and  $\sigma_d^2(k) = E[|Dk|^2]$  and in real-world speech enhancement both of these has to be estimated. The different STSA algorithms are given in terms of these variances and are described below.

### 2.2.1 The different STSA algorithms

The attenuation step of STSA methods consists of three parts, namely computing the *a posteriori* SNR  $\gamma_k$ , estimating the *a priori* SNR  $\xi_k$  and applying a suppression rule  $H_k$ . Figure 2.2 shows that  $\gamma_k$  is first calculated, then  $\xi_k$  is estimated, and finally the suppression rule is applied.

1. Computing the *a posteriori* SNR  $\gamma_k$

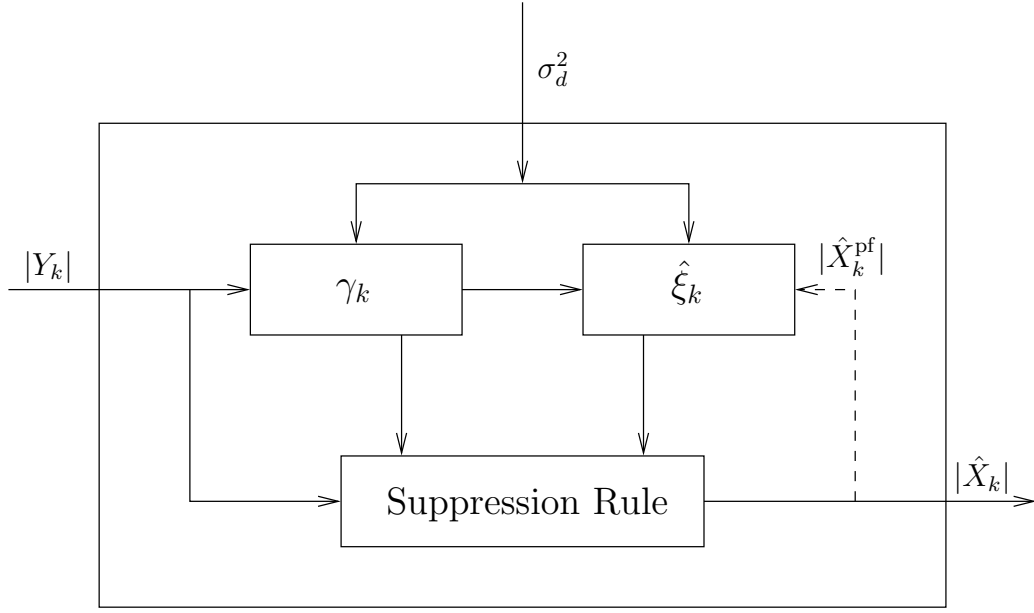
The *a posteriori* signal-to-noise ratio is not a signal-to-noise ratio in the classical sense. It is actually the ratio of the noisy signal power to the noise power, i.e.

$$\frac{\text{“clean signal + noise”}}{\text{“noise”}}.$$

It is observed and calculated as

$$\gamma_k = \frac{R_k^2}{\sigma_d^2(k)}, \quad (2.3)$$

where  $R_k = |Y_k|$  is the magnitude of the  $k^{\text{th}}$  bin noisy Fourier coefficient.



**Figure 2.2:** The flowchart of the STSA attenuation step. The parameters  $\gamma_k$  and  $\hat{\xi}_k$  are first computed. The spectral estimate of the previous frame  $|\hat{X}_k^{\text{pf}}|$ , is only used with the Ephraim-Malah decision-directed  $\xi_k$  estimate. The suppression rule attenuates  $|Y_k|$  to yield  $|\hat{X}_k|$ .

## 2. Estimating the *a priori* SNR $\xi_k$

The *a priori* signal-to-noise ratio is in the usual form of

$$\frac{\text{“clean signal”}}{\text{“noise”}}.$$

It is the unobserved signal-to-noise ratio of bin  $k$ , given by

$$\xi_k = \frac{\sigma_x^2(k)}{\sigma_d^2(k)}. \quad (2.4)$$

Because  $\sigma_x^2(k)$  is unobserved, it is estimated by estimating  $\xi_k$  directly. Two methods to estimate  $\xi_k$  are investigated:

- Maximum Likelihood  $\xi_k$  estimation, and
- the Ephraim-Malah decision-directed  $\xi_k$  estimate.

These  $\xi_k$  estimates are described in detail in Section 2.2.2.



### 3. Applying a suppression rule $H_k$

A suppression rule is a nonnegative real-valued gain  $H_k$  applied to each bin  $k$  of the observed signal spectrum  $Y_k$ . It forms an estimate  $|\hat{X}_k|$  of the the original spectrum by multiplying  $H_k$  with  $|Y_k|$ ,

$$|\hat{X}_k| = H_k |Y_k|. \quad (2.5)$$

The intermediate variable  $\nu_k$  is found in the suppression rules and it is a combination of  $\gamma_k$  and  $\xi_k$  given by [60],

$$\nu_k = \frac{\xi_k}{1 + \xi_k} \gamma_k. \quad (2.6)$$

From the substitution of (2.3) and (2.4) into (2.6),  $\nu_k$  can be written as

$$\nu_k = \left[ \frac{\sigma_x^2}{\sigma_d^2 + \sigma_x^2} \right] \frac{R_k^2}{\sigma_d^2}, \quad (2.7)$$

which can be interpreted as a scaled Wiener shrinkage rule.

Different suppression rules that have been proposed, all in terms of  $\gamma_k$ ,  $\xi_k$  and  $\nu_k$ , are:

- Power Spectral Subtraction [42],

$$H_k = \sqrt{\frac{\xi_k}{1 + \xi_k}}. \quad (2.8)$$

- The Wiener suppression rule [42],

$$H_k = \frac{\xi_k}{1 + \xi_k}. \quad (2.9)$$

- Maximum Likelihood Envelope Estimation [42],

$$H_k = \frac{1}{2} + \frac{1}{2} \sqrt{\frac{\xi_k}{1 + \xi_k}}. \quad (2.10)$$

- Ephraim-Malah MMSE amplitude suppression rule [23],

$$H_k = \left( \frac{\sqrt{\nu_k}}{\gamma_k} \right) \left( \frac{\sqrt{\pi}}{2} \right) \exp\left(-\frac{\nu_k}{2}\right) \left[ (1 + \nu_k) I_0\left(\frac{\nu_k}{2}\right) + \nu_k I_1\left(\frac{\nu_k}{2}\right) \right]. \quad (2.11)$$

- Joint MAP amplitude and phase suppression rule [58, 59, 60],

$$H_k = \frac{\xi_k + \sqrt{\xi_k^2 + 2(1 + \xi_k) \frac{\xi_k}{\gamma_k}}}{2(1 + \xi_k)}. \quad (2.12)$$

- MAP amplitude suppression rule [58, 59, 60],

$$H_k = \frac{\xi_k + \sqrt{\xi_k^2 + (1 + \xi_k) \frac{\xi_k}{\gamma_k}}}{2(1 + \xi_k)} . \quad (2.13)$$

- MMSE spectral power estimator [58, 59, 60],

$$H_k = \sqrt{\frac{\xi_k}{1 + \xi_k} \left( \frac{1 + \nu_k}{\gamma_k} \right)} . \quad (2.14)$$

Any  $\xi_k$  estimate can be used with any suppression rule. Certain combinations of these two are generally used together. Spectral subtraction techniques usually use the maximum likelihood  $\xi_k$  estimate. The widely used Ephraim-Malah speech enhancement algorithm uses the Ephraim-Malah MMSE amplitude suppression rule with the Ephraim-Malah decision-directed  $\xi_k$  estimate [23].

## 2.2.2 Estimating the *a priori* signal-to-noise ratio

### Maximum likelihood $\xi_k$ estimation

The maximum likelihood estimation approach is used to estimate the unknown  $\sigma_x^2(k)$  from  $Y_k$  which has a given probability density function  $f(Y_k)$  [42]. The parameter  $\sigma_x^2(k)$  is the variance of the  $k$ th spectral bin of the frame under analysis. The following derivation is taken from [42]. Only the current frame is used to estimate  $\xi_k$ . The observed spectral component  $Y_k$  is assumed to be a zero-mean complex Gaussian random variable. The variance of  $Y_k$  is defined as  $\sigma_y^2$ , therefore its real and imaginary parts are also Gaussian [42] with variance  $\sigma_y^2/2$ . The probability density function for  $Y_k$  is,

$$f(Y_k) = \left( \frac{1}{\pi \sigma_y^2(k)} \right) \exp \left[ - \frac{|Y_k|^2}{\sigma_y^2(k)} \right] . \quad (2.15)$$

The noise is assumed to be independent and identically distributed (iid), as described in Section 1.2.1. Since the signal and noise components are independent, the noisy variance  $\sigma_y^2(k)$  may be written as [42]

$$\sigma_y^2(k) = \sigma_x^2(k) + \sigma_d^2(k) . \quad (2.16)$$

Substituting (2.16) into (2.15) leads to

$$f(Y_k) = \left( \frac{1}{\pi [\sigma_x^2(k) + \sigma_d^2(k)]} \right) \exp \left[ - \frac{R_k^2}{[\sigma_x^2(k) + \sigma_d^2(k)]} \right] . \quad (2.17)$$

By maximising  $f(Y_k)$  with respect to  $\sigma_x^2(k)$ , the maximum likelihood estimate of  $\sigma_x^2(k)$  can be found to be

$$\hat{\sigma}_x^2(k) = R_k^2 - \sigma_d^2(k) . \quad (2.18)$$

Dividing both sides of (2.18) by  $\sigma_d^2(k)$  leads to the  $\xi_k$  estimate

$$\hat{\xi}_k = \gamma_k - 1 . \quad (2.19)$$

The maximum likelihood  $\hat{\xi}_k$  estimate (2.19) can be interpreted as being a signal-to-noise ratio which is estimated by subtracting the noise from the noisy observation as follows:

$$\frac{\text{“signal”}}{\text{“noise”}} = \frac{\text{“signal+noise”}}{\text{“noise”}} - 1.$$

### The Ephraim-Malah decision-directed $\xi_k$ estimate

Ephraim and Malah [23] proposed a different approach to estimate the *a priori* SNR  $\xi_k$ . For the current analysis frame, the decision-directed *a priori* SNR estimate  $\hat{\xi}_k$  is given by a geometric weighting of the SNR in the previous frame,  $|\hat{X}_k^{\text{pf}}|^2/\sigma_d^2(k)$ , and the current frame  $(R_k^2 - \sigma_d^2(k))/\sigma_d^2(k)$  and is given as

$$\hat{\xi}_k = \alpha \frac{|\hat{X}_k^{\text{pf}}|^2}{\sigma_d^2(k)} + (1 - \alpha) \max[\gamma_k - 1, 0], \quad \alpha \in [0, 1]. \quad (2.20)$$

The term  $|\hat{X}_k^{\text{pf}}|^2$  is the spectral estimate of the previous frame.

The parameter  $\alpha$  is a forgetting factor and is suggested by Ephraim and Malah to be  $\alpha = 0.98$ . This results in a residual noise which is *colourless* and much less annoying than the *musical* noise obtained with the maximum likelihood  $\xi_k$  estimate [23].

The proposed initial conditions [23] are given by

$$\hat{\xi}_k(0) = \alpha + (1 - \alpha) \max[0, \gamma_k(0) - 1] . \quad (2.21)$$

The term  $\hat{\xi}_k(0)$  is the estimated *a priori* SNR of the first frame and  $\gamma_k(0)$  the *a posteriori* SNR of the first frame. The initial conditions are chosen to minimise the initial transition effects in the enhanced speech [23].

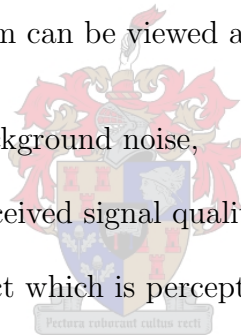
# Chapter 3

## Evaluation of speech enhancement

### 3.1 Introduction

A speech enhancement algorithm can be viewed as successful if it

1. suppresses perceivable background noise,
2. preserves or enhances perceived signal quality, and
3. produces a residual artifact which is perceptually acceptable.



Speech enhancement evaluation attempts to quantify these properties. This is no trivial task, since the performance of speech enhancement is influenced by the specific type of noise, the global SNR, the noise estimation, the algorithm framework and the algorithm parameter settings [33]. Although significant progress has been made in speech enhancement in recent years, the evaluation of the process has not yet been standardised. Hansen and Pellom [33] proposed a standardisation, which involves the speech enhancement of a standard speech database. They suggest using the 192 sentences of the TIMIT Core test set<sup>1</sup>, downsampled to  $F_s = 8$  kHz, with a set of different noise types. The evaluation of enhanced speech is done by using different objective distortion measures and subjective listening tests.

---

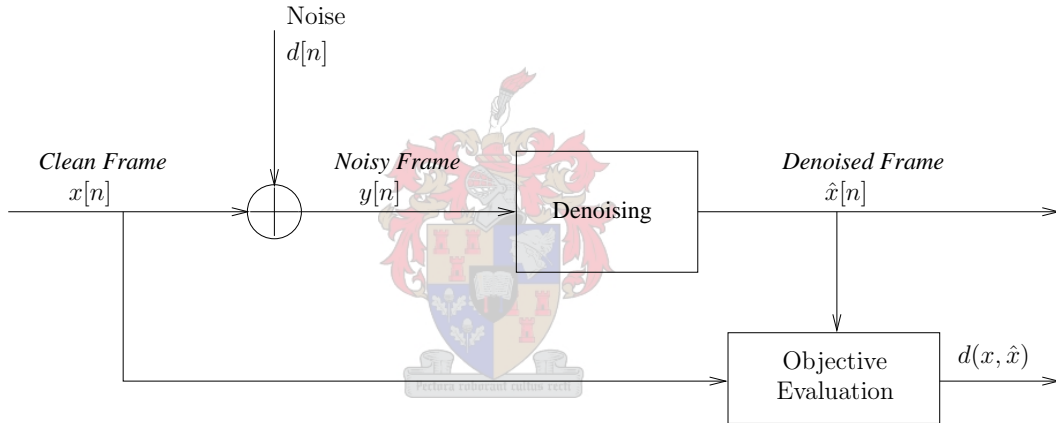
<sup>1</sup>See Appendix B.

## 3.2 Objective quality measures

A speech distortion measure is a single nonnegative number, that mathematically describes the quality and intelligibility of enhanced speech compared to the original speech. Objective evaluation also has the difficult task of quantifying the various residual artifacts. Any such objective measure has to correlate with subjective listening tests. It is difficult to satisfy all of these requirements with a single distortion measure.

Objective speech quality measures are computed on a frame-by-frame basis, with  $d(x, \hat{x})$  the distortion between clean frame  $x[n]$  and the denoised frame  $\hat{x}[n]$  with  $n = 1, 2, \dots, N$  and  $N$  being the number of samples in the frame.

Objective evaluation is only applicable in a laboratory environment where the original signal is available. The experimental setup is shown in Figure 3.1.



**Figure 3.1:** The flowchart of objective speech enhancement evaluation.

The distortion measure must be subjectively meaningful in the sense that a difference in the measure corresponds to a difference in perceived quality and intelligibility.

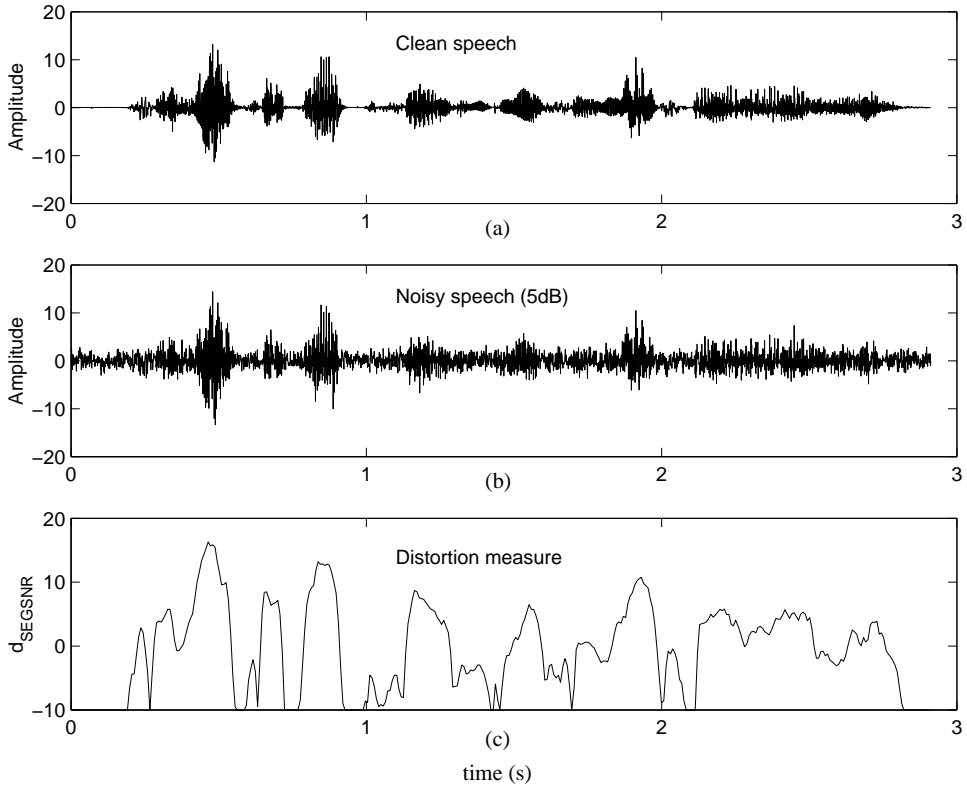
A common distortion measure is the mean-square error ( $d_{MSE}$ ). It is widely and successfully used in image denoising [47] and is given by

$$d_{MSE} = \frac{1}{N} \sum_{n=0}^{N-1} (x[n] - \hat{x}[n])^2 . \quad (3.1)$$

Because  $d_{MSE}$  is a subtractive measure, smaller values correspond to better quality. A large  $d_{MSE}$  distortion, however, does not necessarily imply poor speech quality. For example, a “shh” sound is essentially a white noise process and any typical waveform would sound the same, although the  $d_{MSE}$  will be large [29]. The MSE is therefore not a good distortion measure for speech.

There is a wide range of objective measures specifically designed for speech evaluation. Of these the *segmental signal-to-noise ratio* distortion measure  $d_{SEGSNR}$ , described in Section 3.2.1, and the *Itakura-Saito* measure  $d_{IS}$ , described in Section 3.2.2, are the most widely used and these are therefore chosen to be the objective measures used in this research, as in [31, 37, 55].

Figure 3.2 shows how the frame-based  $d_{SEGSNR}$  varies over time for noisy speech compared to clean speech. Since speech signals vary over time, due to the sequence of phonemes, the impact of background distortion will also vary.



**Figure 3.2:** (a) Clean speech signal, “She had your dark suit in greasy wash water all year”. (b) Noisy speech (5 dB global SNR), corrupted with white Gaussian noise. (c) The segmental signal-to-noise ratio distortion measure  $d_{SEGSNR}$  compares the clean and noisy speech. The values vary significantly over time. Phonemes with higher energy are far less effected by the noise.

A global objective measure is the average value of all the frame-based distortion measures. The global objective measures are calculated by using only the speech segments (i.e. discarding the distortion values of the silent regions just before and after the utterance) and discarding the worst 5% of the measures as proposed in [33].

### 3.2.1 Segmental signal-to-noise ratio $d_{SEGSNR}$

The overall signal-to-noise ratio can be computed as

$$d_{SNR} = 10 \log_{10} \frac{\sum_{m=0}^{M-1} x^2[m]}{\sum_{m=0}^{M-1} \{x[m] - \hat{x}[m]\}^2} \text{ dB}, \quad (3.2)$$

with  $x[m]$  the clean utterance and  $\hat{x}[m]$  the enhanced utterance. The index  $m = 0, 1, \dots, M-1$  is a sample counter with  $M$  being the number of samples within the whole sentence. The  $d_{SNR}$  measure, however, is of little value as an objective measure of speech quality because of the non-uniform impact of noise on enhanced speech quality, which can be seen in Figure 3.2(c). The  $d_{SNR}$  measure also correlates poorly with subjective tests [33].

The frame-based segmental signal-to-noise ratio, however, is a reasonable measure of speech quality. The segmental signal-to-noise ratio distortion measure is computed for each analysis frame and is given as [33]

$$d_{SEGSNR}(x, \hat{x}) = 10 \log_{10} \frac{\sum_{n=0}^{N-1} x^2[n]}{\sum_{n=0}^{N-1} \{x[n] - \hat{x}[n]\}^2} \text{ dB}, \quad (3.3)$$

with  $x[n]$  the clean frame and  $\hat{x}[n]$  the denoised frame under analysis. The index  $n = 0, 1, \dots, N-1$  is the sample counter with  $N$  being the number of samples within the frame.

The  $d_{SEGSNR}$  is typically in the range  $-10$  dB to  $35$  dB, with a higher  $d_{SEGSNR}$  corresponding to better performance. Frames with an SNR estimate above  $35$  dB do not differ perceptually from the clean frame, therefore an upper limit of  $35$  dB is set for frames with a value higher than this. Frames during periods of silence tend to have very large negative  $d_{SEGSNR}$  values. This is similarly not a true reflection of perception, and a lower limit of  $-10$  dB is set for  $d_{SEGSNR}$  values below this.

The global segmental signal-to-noise ratio  $d_{SEGSNR}$  is calculated by averaging the frame-based  $d_{SEGSNR}(x, \hat{x})$  distortion measures (3.3) [33]

$$d_{SEGSNR} = \frac{1}{K} \sum_{k=0}^{K-1} d_{SEGSNR}(x_k, \hat{x}_k). \quad (3.4)$$

The index  $k = 1, 2, \dots, K$  is a frame counter with  $K$  being the number of frames in the utterance. The clean and denoised signals  $x_k$  and  $\hat{x}_k$  are that of frame number  $k$ . It should be noted that the overall signal-to-noise ratio (3.2) differs from the global segmental signal-to-noise ratio (3.4), which is an average of logarithmic (dB) values.

### 3.2.2 Itakura-Saito distortion measure $d_{IS}$

The Itakura-Saito distortion measure,  $d_{IS}$ , is based on the LP power spectrum, which models the human speech production system. It describes the spectral matching properties of linear prediction and is influenced by the similarity or difference between the LP power spectra of the clean and denoised frames [29]. It is, as with the  $d_{SEGSNR}$  measure, calculated on a frame-by-frame basis, where  $d_{IS}(x, \hat{x})$  denotes the Itakura-Saito distortion between clean frame  $x[n]$  and denoised frame  $\hat{x}[n]$ . The global  $d_{IS}$  distortion is the average of the frame-based measures and is calculated similar to (3.4).

The Itakura-Saito distortion is derived in Appendix A and can be written as

$$d_{IS}(x, \hat{x}) = \frac{\mathbf{a}_d^T \mathbf{R}_c \mathbf{a}_d}{\mathbf{a}_d^T \mathbf{R}_d \mathbf{a}_d} + \ln \frac{\sigma_d^2}{\sigma_c^2} - 1, \quad (3.5)$$

or, as given in [33], as

$$d_{IS}(x, \hat{x}) = \left[ \frac{\sigma_c^2}{\sigma_d^2} \right] \left[ \frac{\mathbf{a}_d^T \mathbf{R}_c \mathbf{a}_d}{\mathbf{a}_c^T \mathbf{R}_c \mathbf{a}_c} \right] + \ln \frac{\sigma_d^2}{\sigma_c^2} - 1. \quad (3.6)$$

It should be noted that (3.5) and (3.6) is a comparison between a clean frame and a denoised frame of speech. Therefore, subscripts  $c$  and  $d$  refer to the clean and denoised frames, respectively. Variables  $\sigma^2$ ,  $\mathbf{a}$  and  $\mathbf{R}$  are taken from the ‘‘autocorrelation method’’ of short-term linear prediction analysis [15]. Variable  $\sigma^2$  is the prediction error power or all-pole gain. The matrix  $\mathbf{R}$  is the autocorrelation matrix in its Toeplitz form and  $\mathbf{a} = [1 \ a_1 \ a_2 \ \dots \ a_P]^T$  is the linear prediction coefficient vector with  $P$  the order.

The Itakura-Saito distortion measure penalises a mismatch in formant locations [37]. By looking at (3.6) it is seen that if  $\mathbf{a}_c \approx \mathbf{a}_d$  and  $\sigma_c^2 \approx \sigma_d^2$  then  $d_{IS}(x, \hat{x}) \approx 0$ , which implies low  $d_{IS}$  values for frames with similar LP power spectra. High  $d_{IS}$  values therefore imply that the denoised speech is of poor quality compared to the original speech. Errors in the location of spectral valleys do not contribute as heavily as a mismatch in formant peaks [37].

The  $d_{IS}$  measure is subjectively meaningful [29] and correlates well with subjective measures [37]. The typical range of  $d_{IS}$  values is from 1 to 10 with lower  $d_{IS}$  values corresponding to better performance. Frames containing non-speech might have unrealistically high distortion values and should not be incorporated. Hansen and Pellom [33] suggested discarding the highest 5% of the  $d_{IS}$  values in computing the global  $d_{IS}$  distortion measure.

The Itakura-Saito measure is implemented in this study by using the software of Pellom [45]. Half-overlapping frames of 32 ms are used. The frames first get shifted to have



a zero mean. Each frame is then multiplied with a *Hanning* window to reduce spectral leakage. A linear prediction filter order of  $P = 10$  is used.

### 3.3 Subjective listening tests

Two different types of subjective evaluation are done in this study, namely formal listening tests and informal listening tests.

#### 3.3.1 Informal listening tests

Informal listening tests are done throughout this study. It consists of listening to a few sentences of denoised speech and then commenting on its quality, intelligibility and residual artifacts. The purpose of informal tests is to support the objective evaluation when designing the different denoising algorithms. The sentences used for informal listening tests are shown in Appendix B.3.

#### 3.3.2 Formal listening tests

For the formal tests, an independent evaluator listens to two different denoised versions of a sentence and then chooses which of the two he prefers. This process, referred to as a *trial*, is repeated for a number of sentences and evaluators. The end result is a set of *preference counts*, which indicate how many times a specific model was preferred to another model. These preference counts are then combined to form overall rankings for the different denoising algorithms. The formal listening tests are used to compare different algorithms with each other. The experimental setup for these tests is described in Appendix B.4.

### 3.4 Denoising artifacts

As described in Chapter 1.1, speech enhancement may be viewed as

1. *forward transformation*, transforming the noisy signal into a particular domain,
2. *attenuation*, attenuating the noisy coefficients, and

3. *inverse transformation*, inverse-transforming these back to the time domain.

STSA speech enhancement algorithms generally produce two main undesirable effects, namely “musical” residual noise and speech distortion.

The attenuation step is a process which attempts to decompose the noisy coefficients into their signal and noise components. The clean signal coefficients are estimated from this classification. Algorithms will inevitably classify certain components incorrectly. These mistakes lead to different artifacts when they are transformed back to the time domain.

### 3.4.1 Musical noise

Musical noise is a frequently encountered residual noise artifact of STSA techniques [9]. If noise coefficients are incorrectly classified as signal coefficients, the actual sinusoidal basis functions are transformed back to the time domain. This results in isolated short-time windowed sinusoids. Musical noise is tonal components at random frequencies, has an unnatural structure and is perceptually annoying.

### 3.4.2 Speech distortion

At low signal-to-noise ratios it is difficult to suppress noise without introducing speech distortion and therefore decreasing intelligibility [55]. Speech is distorted if the coefficients containing signal energy are incorrectly attenuated. This happens if the enhancement algorithm mistakes signal components for noise components. Although typically not as annoying as “musical” noise, speech distortion can impair intelligibility.

### 3.4.3 The trade-off

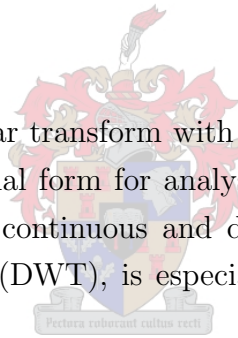
Most STSA algorithms have parameters that can be set to find the best trade-off between musical noise and speech distortion. For example, the  $\alpha$  parameter of the Ephraim-Malah decision-directed  $\xi_k$  estimate fulfils this role (see Section 2.2.2). Ephraim and Malah [23] propose  $\alpha = 0.98$  as subjectively the best value. This results in higher speech distortion, but lower musical noise. Lower  $\alpha$  values, however, lead to better  $d_{SEGSNR}$  and  $d_{IS}$  values, because the speech distortion is lower at the cost of higher musical noise.

# Chapter 4

## Wavelet theory and filter bank design

### 4.1 Introduction

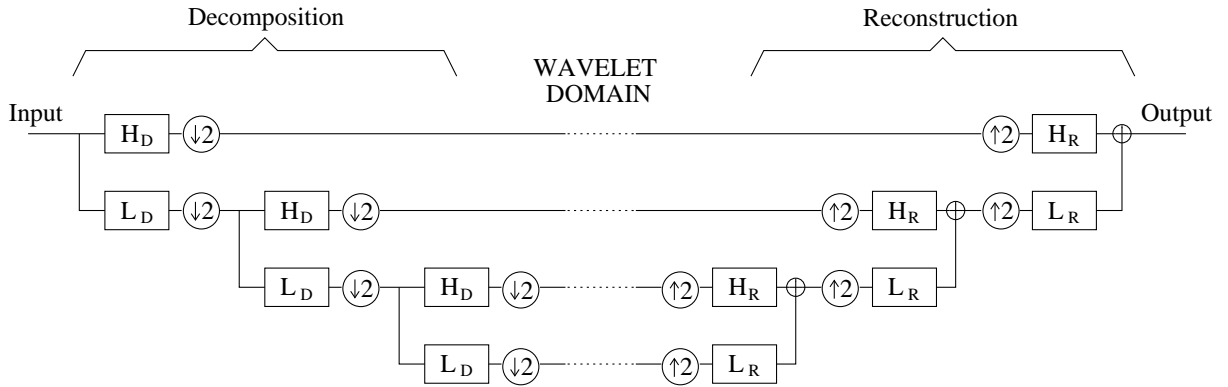
The wavelet transform is a linear transform with a hierarchical or multiresolution structure. It exists in one-dimensional form for analysing signals, and two-dimensional form for use with images. It has a continuous and discrete version. The latter, known as the discrete wavelet transform (DWT), is especially simple to implement, owing to its connection with filter banks.



### 4.2 Wavelet filter banks

The DWT is found by passing the data iteratively through a filter bank as shown in Figure 4.1. The output signal of each decomposition filter is downsampled by a factor of two to create the wavelet coefficients of the wavelet domain. The inverse discrete wavelet transform (IDWT) is found by upsampling with a factor of two and then filtering. The following sections are based on [52].

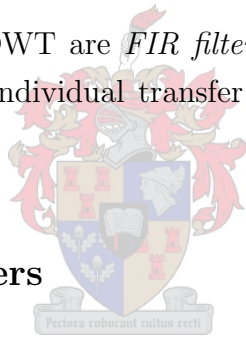
The decomposition bank shares the lowpass and highpass filters,  $L_D$  and  $H_D$ . Similarly, the reconstruction bank shares  $L_R$  and  $H_R$ . Each different wavelet has its own corresponding set of four filters. It seems unbelievable that perfect reconstruction is possible, since signal information is being thrown away in the downsampling step. However, by correctly designing the four filters to be wavelet filters, perfect reconstruction is achieved.



**Figure 4.1:** The discrete wavelet transform and inverse transform filter banks. The decomposition filter bank is on the left and the reconstruction filter bank is on the right. The dotted region in the middle is the wavelet domain.

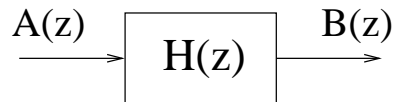
### 4.3 Designing the wavelet filters

The basic components of the DWT are *FIR filters*, *upsamplers* and *downsamplers*. To design the wavelet filters, the individual transfer functions of the components are first derived.



#### 4.3.1 The wavelet filters

Wavelet filters are *finite impulse response* (FIR) filters, which are described by its impulse response. If the input to the filter is an impulse, the output sequence or impulse response is given by  $\{h_0 \ h_1 \ h_2 \ \dots \ h_{L-1}\}$ , with  $L$  being the length of the filter. These numbers are also known as filter coefficients. In the  $z$ -domain, shown in Figure 4.2, the filter is described by the transfer function  $H(z) = h_0 + h_1 z^{-1} + h_2 z^{-2} + \dots + h_{L-1} z^{L-1}$ . The filter is called a FIR filter, because its response to an impulse is of finite duration.



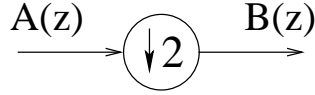
**Figure 4.2:** The FIR filter  $H(z)$ , with  $A(z)$  the input and  $B(z)$  the output.

If  $A(z)$  is the input and  $H(z)$  the filter transfer function, then the output  $B(z)$  is merely a multiplication of  $A(z)$  and  $H(z)$  in the  $z$ -domain,

$$B(z) = A(z)H(z) . \tag{4.1}$$

### 4.3.2 The downsampler

The downsampler, shown in Figure 4.3, discards every second sample of the incoming sequence. The output sequence  $B(z)$  now has half the number of samples compared to the input sequence  $A(z)$ .



**Figure 4.3:** *The downsampler discards every second sample of the input  $A(z)$ , to produce the output  $B(z)$ .*

If the input sequence (even length  $N$ ) is,

$$A(z) = a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + \cdots + a_{N-1}z^{N-1}, \quad (4.2)$$

and the output sequence (length  $N/2$ ) is,

$$B(z) = a_0 + a_2z^{-1} + a_4z^{-2} + a_6z^{-3} + \cdots + a_{N-2}z^{-(\frac{N-2}{2})}, \quad (4.3)$$

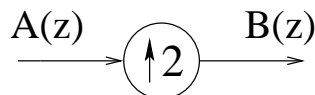
then the output of the downsampler can be written as

$$B(z) = \frac{A(\sqrt{z}) + A(-\sqrt{z})}{2}. \quad (4.4)$$

The discarding of samples leads to aliasing in the frequency domain, and in general it is not possible to determine  $A(z)$  from  $B(z)$ .

### 4.3.3 The upsampler

The upsampler, shown in Figure 4.4, inserts a zero between every two elements of the incoming sequence. Now the output sequence  $B(z)$  has twice as many samples as the input sequence  $A(z)$ .



**Figure 4.4:** *The upsampler inserts a zero between every sample of the input  $A(z)$ , to produce the output  $B(z)$ .*

If the input sequence (length  $N$ ) is,

$$A(z) = a_0 + a_1z^{-1} + a_2z^{-2} + a_3z^{-3} + \cdots + a_{N-1}z^{N-1}, \quad (4.5)$$

and the output sequence (length  $2N$ ) is,

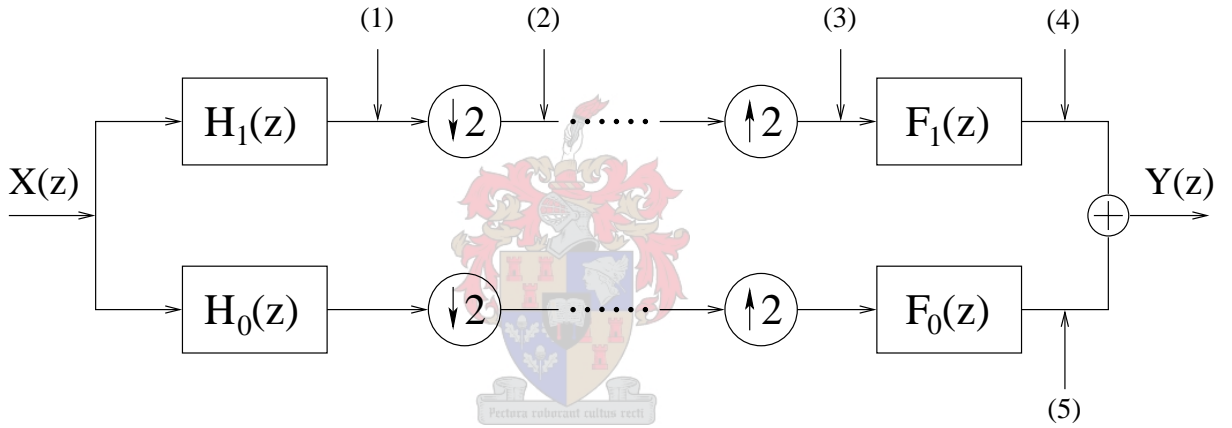
$$B(z) = a_0 + 0z^{-1} + a_1z^{-2} + 0z^{-3} + \dots + a_{N-1}z^{-2(N-1)}, \quad (4.6)$$

then the output of the upsampler can be written as

$$B(z) = A(z^2). \quad (4.7)$$

### 4.3.4 The two-channel filter bank

The basic building block of the DWT is the two-channel filter bank shown in Figure 4.5. Perfect reconstruction of the DWT, shown in Figure 4.1, is trivial if it is found for the two-channel filter bank. Carefully designing the FIR filters achieves perfect reconstruction.



**Figure 4.5:** The two-channel filter bank, with  $H_0$  the lowpass decomposition filter,  $H_1$  the highpass decomposition filter,  $F_0$  the lowpass reconstruction filter and  $F_1$  the highpass reconstruction filter.

Using the individual transfer functions from (4.1), (4.4) and (4.7) and following Figure 4.5, the transfer function of the two-channel filter bank is derived. The signals at various stages of the filter bank are given below:

$$\begin{aligned} \text{At (1):} & \quad X(z)H_1(z) \\ \text{At (2):} & \quad \frac{1}{2}[X(\sqrt{z})H_1(\sqrt{z}) + X(-\sqrt{z})H_1(-\sqrt{z})] \\ \text{At (3):} & \quad \frac{1}{2}[X(z)H_1(z) + X(-z)H_1(-z)] \\ \text{At (4):} & \quad \frac{1}{2}F_1(z)[X(z)H_1(z) + X(-z)H_1(-z)] \end{aligned}$$

Similarly,

$$\text{At (5):} \quad \frac{1}{2}F_0(z)[X(z)H_0(z) + X(-z)H_0(-z)]$$

Therefore,

$$\begin{aligned}
 Y(z) = & \underbrace{\frac{1}{2} \left[ H_0(z)F_0(z) + H_1(z)F_1(z) \right]}_{\text{Distortion}} X(z) \\
 & + \underbrace{\frac{1}{2} \left[ H_0(-z)F_0(z) + H_1(-z)F_1(z) \right]}_{\text{Aliasing term}} X(-z)
 \end{aligned} \tag{4.8}$$

The “transfer function” of the two-channel filter bank (4.8) lies at the heart of the filter design. The system will be a wavelet system if

1. the filters are regular, and if
2. the two-channel filter bank yields perfect reconstruction.

### 4.3.5 Regularity

The system has to be regular to be a wavelet system. This entails ensuring that  $H_0(z)$  is actually a lowpass filter and  $H_1(z)$  a highpass filter. The most basic requirement for regularity is for a highpass filter to fail to pass DC. This is achieved by making the filter coefficients of the highpass filter sum to zero. Alternatively, the lowpass filter should have zeros at  $z = -1$ . The number of zeros at  $z = -1$  determine the order of regularity with a higher order of regularity resulting in filters with a flatter magnitude response.

### 4.3.6 Perfect reconstruction

The filters of the two-channel filter bank are designed so that the filter bank as a whole has perfect reconstruction, i.e.

$$Y(z) = z^{-K} X(z). \tag{4.9}$$

The delay term  $z^{-K}$  introduces a delay of  $K$  samples. It should be noted that a delay still yields perfect reconstruction, since the output sequence can be circularly rotated to produce the input sequence. The requirements for perfect reconstruction are therefore to set the *aliasing* term in (4.8) to zero and the *distortion* term to a delay:

- aliasing term = 0,
- distortion term =  $z^{-K}$ .

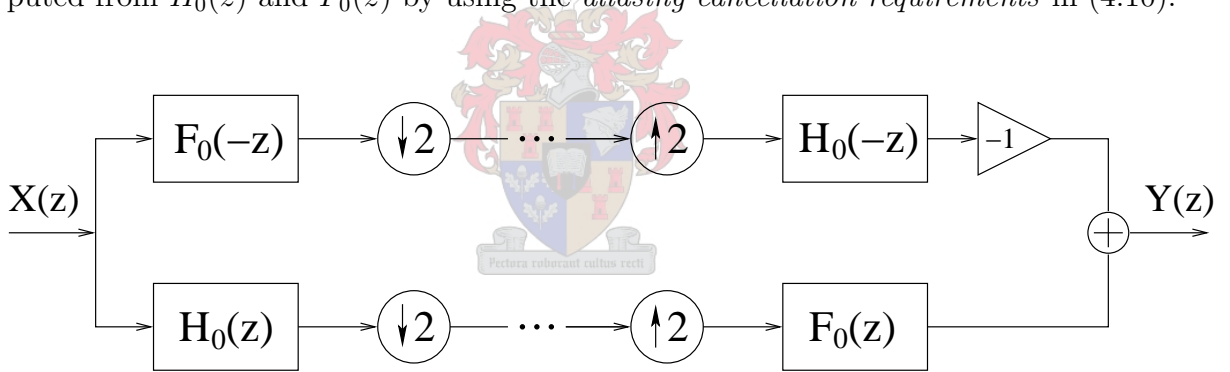
### Setting the aliasing term to zero

By setting the aliasing term of (4.8) to zero, the *aliasing cancellation requirements* are derived,

$$F_0(z) = H_1(-z) \quad \text{and} \quad F_1(z) = -H_0(-z). \quad (4.10)$$

The lowpass reconstruction filter  $F_0(z)$  and the highpass decomposition filter  $H_1(z)$  are equal in length and flipped versions of each other. The effect of the minus sign in  $H_1(-z)$  is to flip the frequency response of the filter around the imaginary axis in the complex  $z$ -plane. This changes the highpass filter into its equivalent lowpass filter. Similarly, the lowpass decomposition filter  $H_0(z)$  and the highpass reconstruction filter  $F_1(z)$  are equal in length and flipped versions of each other.

The new two-channel filter bank incorporates the aliasing cancellation requirements and is shown in Figure 4.6. Now only the two lowpass filters,  $H_0(z)$  and  $F_0(z)$ , have to be designed, compared to the four filters of Figure 4.5. The filters  $H_1(z)$  and  $F_1(z)$  are computed from  $H_0(z)$  and  $F_0(z)$  by using the *aliasing cancellation requirements* in (4.10).



**Figure 4.6:** *The new two-channel filter bank with the aliasing cancellation requirements incorporated.*

### Setting the distortion term to be a delay

The two lowpass filters are designed by setting the distortion term of (4.8) to a delay, and by setting the aliasing term to zero. This results in

$$H_0(z)F_0(z) + H_1(z)F_1(z) = 2z^{-K}, \quad (4.11)$$

Substituting the aliasing cancellation requirements from (4.10) into (4.11) yields

$$H_0(z)F_0(z) - H_0(-z)F_0(-z) = 2z^{-K}. \quad (4.12)$$



Developing the DWT now results in designing filters to satisfy (4.12). A solution to this is called the *biorthogonal* solution. This is a general solution since there are two unknowns namely  $H_0(z)$  and  $F_0(z)$ .

With  $h_i$  the coefficients of  $H_0$  and  $f_i$  those of  $F_0(z^{-1})$  the regularity requirement of both filters are given by [52]

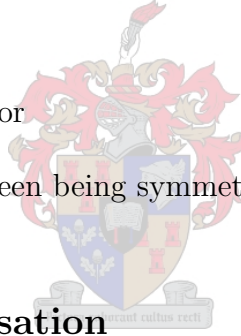
$$\sum_{i=0}^{N-1} (-1)^i h_i = 0 \quad , \quad \sum_{i=0}^{N-1} (-1)^i f_i = 0 \quad \text{and} \quad \sum_{i=0}^{N-1} h_i = \sqrt{2} . \quad (4.13)$$

For perfect reconstruction  $h_i$  and  $f_i$  must satisfy [52]

$$\sum_{i=0}^{N-1} h_i f_i = 1 \quad \text{and} \quad \sum_{i=0}^{N-1} h_i f_{i+2k} = 0 \quad \text{for} \quad k \neq 0 . \quad (4.14)$$

Biorthogonal filters can be designed so that

- filters are symmetric,
- filters are maximally flat, or
- filters are a trade-off between being symmetric and having minimum phase.



### 4.3.7 Spectral Factorisation

For perfect reconstruction with delay  $K$ , (4.12) has to be satisfied. Spectral factorisation can be used to do this, by defining

$$P(z) = H_0(z)F_0(z). \quad (4.15)$$

Now perfect reconstruction is achieved if

$$P(z) - P(-z) = 2z^{-K}. \quad (4.16)$$

Polynomial  $P(z)$  is first computed and then factorised as in (4.15). For perfect reconstruction  $P(z)$  is a polynomial with even powers of  $z$  only, except for a single odd power  $z^{-K}$  with a coefficient of 1. Therefore, with  $c_i$  the coefficients of the polynomial,  $P(z)$  can be written as

$$P(z) = \cdots + c_4 z^4 + c_2 z^2 + c_0 + c_{-2} z^{-2} + c_{-4} z^{-4} + \cdots + 1 z^{-K}. \quad (4.17)$$

Perfect reconstruction can easily be verified for odd  $K$  by substituting (4.17) into (4.16).

Since linear phase is important,  $P(z)$  is further restricted to be a symmetric halfband lowpass filter. Now  $P(z)$  has the form [34]

$$P(z) = z^{-K} \left[ 1 + \sum_i p_i (z^{-2i+1} + z^{2i-1}) \right] \quad \text{where } K \text{ is odd.} \quad (4.18)$$

An additional restriction is for  $P(z)$  to have a maximum order of regularity. This implies that  $P(z)$  should have the maximum number of zeros at  $z = -1$ .

Herrmann [34] proposed a solution to design  $P(z)$  by first choosing the Herrmann order  $m$  and then finding a polynomial  $P_m(x)$  given as

$$P_m(x) = (1-x)^m \sum_{\nu=0}^{m-1} \binom{m-1+\nu}{\nu} x^\nu \quad (4.19)$$

If  $P_m(x)$  has been chosen, the maximally flat symmetric lowpass filter  $P(z)$  may be found with the transformation

$$x = \frac{1}{2} \left( 1 - \frac{1}{2} (z + z^{-1}) \right). \quad (4.20)$$

All Herrmann filters satisfy (4.18) and hence (4.16). The Herrmann filter  $P(z)$  has  $4m - 2$  zeros in total. The filter has  $2m$  zeros at  $z = -1$  and there are  $2m - 2$  remaining zeros.

The first few maximally flat symmetric halfband filters are given in Table 4.1.

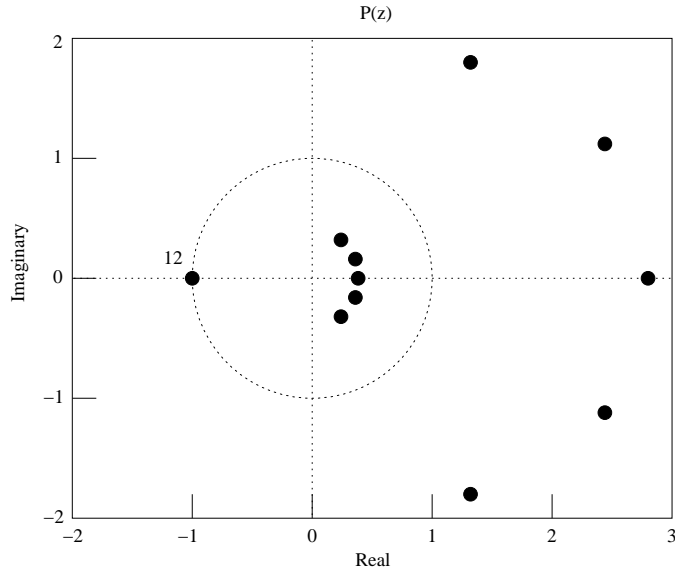
**Table 4.1:** *Maximally flat symmetric halfband filters of Herrmann orders  $m = 1$  to  $m = 4$ .*

$m$	$P(z)$
1	$\frac{1}{2}(1 + z^{-1})^2$
2	$\frac{1}{16}(1 + z^{-1})^4(-1 + 4z^{-1} - z^{-2})$
3	$\frac{1}{256}(1 + z^{-1})^6(3 - 18z^{-1} + 38z^{-2} - 18z^{-3} + 3z^{-4})$
4	$\frac{1}{2048}(1 + z^{-1})^8(-5 + 40z^{-1} - 131z^{-2} + 208z^{-3} - 131z^{-4} + 40z^{-5} - 5z^{-6})$

Spectral factorisation is described by using an example of  $P(z)$  with a Herrmann order of  $m = 6$ , so that

$$P(z) = \frac{1}{524288} (1 + z^{-1})^{12} \begin{pmatrix} -63 + 756z^{-1} - 4067z^{-2} + 12768z^{-3} \\ -25374z^{-4} + 32216z^{-5} - 25374z^{-6} \\ +12768z^{-7} - 4067z^{-8} + 756z^{-9} - 63z^{-10} \end{pmatrix}. \quad (4.21)$$

Factorising  $P(z)$  into its roots leads to the pole-zero plot in the  $z$ -plane shown in Figure 4.7. The black dots indicate the zeros of  $P(z)$ .



**Figure 4.7:** The Pole-zero plot of maximally flat symmetric lowpass filter  $P(z)$  with a Herrmann order of  $m = 6$ .

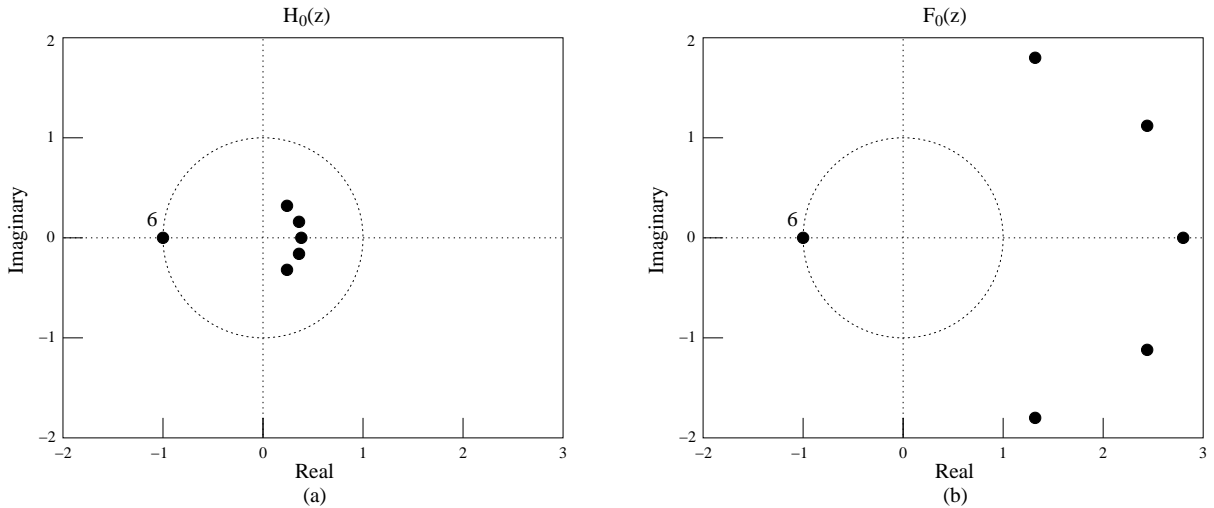
Now that  $P(z)$  is chosen, the problem of spectral factorisation described by (4.15) involves dividing the roots of  $P(z)$  between  $H_0(z)$  and  $F_0(z)$ . A few special cases are investigated, which leads to designing wavelets of a certain wavelet family.

### The Daubechies wavelet family

The Daubechies wavelets are obtained by choosing both  $H_0(z)$  and  $F_0(z)$  to be maximally flat. Looking at Table 4.1, all the Herrmann filters have an even number ( $2m$ ) of zeros at  $z = -1$ . These zeros are evenly divided between  $H_0(z)$  and  $F_0(z)$  to make them both maximally flat. The remaining zeros are divided between  $H_0(z)$  and  $F_0(z)$  by assigning the zeros inside the unit circle to the lowpass decomposition filter  $H_0(z)$ , thereby making it minimum phase. This causes  $F_0(z)$  to receive the zeros outside the unit circle. This is shown in Figure 4.8 for a Herrmann order of  $m = 6$ , where the zeros of  $H_0(z)$  and  $F_0(z)$  are indicated with black dots.

This choice results in the following properties of Daubechies wavelets:

- Both  $H_0(z)$  and  $F_0(z)$  are maximally flat filters.
- $H_0(z)$  is a minimum phase filter, while  $F_0(z)$  is maximum phase.
- Each zero  $z$  and its complex conjugate  $\bar{z}$  stay together, which ensures real-valued



**Figure 4.8:** *The Daubechies 6 wavelet filters in the  $z$ -plane. (a) Lowpass decomposition filter  $H_0(z)$ . (b) Lowpass reconstruction filter  $F_0(z)$ .*

filter coefficients.

- Both  $H_0(z)$  and  $F_0(z)$  are even length filters.
- $H_0(z)$  and  $F_0(z)$  have the same lengths.
- The zeros of  $H_0(z)$  are the inverses of the zeros of  $F_0(z)$ .
- The filter coefficients  $\{h_0\}$  are formed by reversing the filter coefficients  $\{f_0\}$  in time.
- Each Herrmann order produces one unique set of filters.

## The Biorthogonal wavelet family

The Biorthogonal wavelets are obtained by choosing both  $H_0(z)$  and  $F_0(z)$  to be symmetric, and thus having linear phase. Linear-phase filters preserve the position of signal details. Any combination of zeros at  $z = -1$  can be given to  $H_0(z)$  and  $F_0(z)$ , as long as both receive at least one zero at  $z = -1$  to satisfy regularity. The remaining zeros of symmetric halfband filters come in groups of four, i.e. if the filter has a zero at  $z$ , it will also have zeros at  $\bar{z}$  (complex conjugate),  $z^{-1}$  (inverse) and  $\bar{z}^{-1}$  (inverse of the complex conjugate). For the filters to have real coefficients, zeros  $z$  and  $\bar{z}$  must stay together. For the filters to be symmetric, zeros  $z$  and  $z^{-1}$  must stay together. Real-valued zeros come in groups of two,  $z$  and  $z^{-1}$ , which also have to stay together. Any group of four (or two) zeros can be given to either  $H_0(z)$  or  $F_0(z)$ , as long as the zeros within these groups stay together, to ensure real-valued coefficients and linear phase.

The Biorthogonal wavelets used in this denoising research project are all chosen to have a short lowpass decomposition filter  $H_0(z)$  and a longer lowpass reconstruction filter  $F_0(z)$  and can be used with any Herrmann order  $m$ . The longer lowpass reconstruction filter results in better smoothing [52]. Both filters are regular, but  $F_0(z)$  has a much higher order of regularity and therefore has a flatter response than  $H_0(z)$ .

- The **Biorthogonal 1** wavelet family.

The zeros of  $P(z)$  are divided between  $H_0(z)$  and  $F_0(z)$  so that  $H_0(z)$  is a short filter with only one zero at  $z = -1$ . The lowpass reconstruction filter  $F_0(z)$  receives all the other zeros at  $z = -1$  and all the remaining zeros of  $P(z)$ . An example of Biorthogonal 1 wavelet filters in the  $z$ -plane, with a Herrmann order of  $m = 6$ , is shown in Figure 4.9. The Biorthogonal 1 wavelets with Herrmann orders  $m = 1, 2$  and 3 are the Matlab [41] “rbio1.1”, “rbio1.3” and “rbio1.5” wavelets.

- The **Biorthogonal 2** wavelet family.

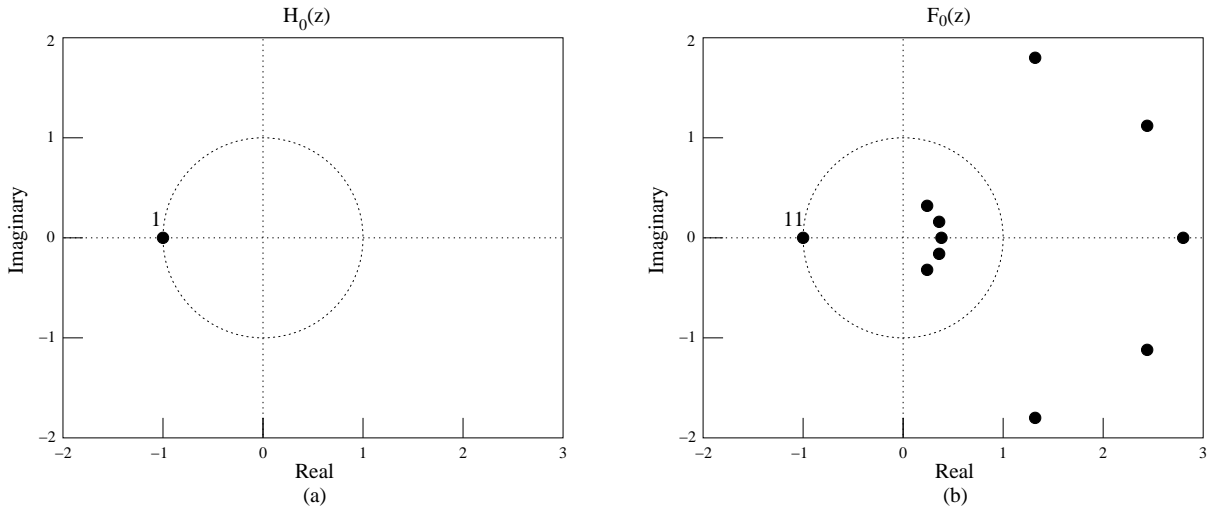
This family is similar to the Biorthogonal 1 wavelet family, except that  $H_0(z)$  has two zeros at  $z = -1$ , and  $F_0(z)$  has ten zeros at  $z = -1$ , for  $m = 6$ .  $F_0(z)$  also receives all the remaining zeros of  $P(z)$ . The Biorthogonal 2 wavelets with Herrmann orders  $m = 2, 3, 4$  and 5 are the Matlab [41] “rbio2.2”, “rbio2.4”, “rbio2.6” and “rbio2.8” wavelets.

- The **Biorthogonal 3** wavelet family.

This family is similar to the Biorthogonal 1 and Biorthogonal 2 wavelet families, except that  $H_0(z)$  has three and  $F_0(z)$  has nine zeros at  $z = -1$ , for  $m = 6$ . The Biorthogonal 3 wavelets with Herrmann orders  $m = 2, 3, 4$  and 5 are the Matlab [41] “rbio3.1”, “rbio3.3”, “rbio3.5” and “rbio3.7” wavelets.

The Biorthogonal wavelets used in this study have the following properties:

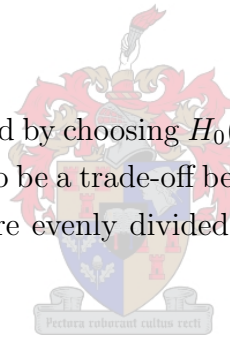
- Both  $H_0(z)$  and  $F_0(z)$  are symmetric, and therefore linear-phase filters.
- The filters  $H_0(z)$  and  $F_0(z)$  do not have to have the same lengths.
- Each zero  $z$  and its complex conjugate  $\bar{z}$  stay together, which produces real-valued filter coefficients.
- Each zero  $z$  and its inverse  $z^{-1}$  stay together, which produces linear-phase filters.
- There are many different choices of wavelet filters with the same Herrmann order.



**Figure 4.9:** The Biorthogonal 1 wavelet filters in the  $z$ -plane. (a) Lowpass decomposition filter  $H_0(z)$ . (b) Lowpass reconstruction filter  $F_0(z)$ .

### The Symlet wavelet family

The Symlet wavelets are obtained by choosing  $H_0(z)$  to be a trade-off between linear phase and minimum phase and  $F_0(z)$  to be a trade-off between linear phase and maximum phase. The zeros of  $P(z)$  at  $z = -1$  are evenly divided between  $H_0(z)$  and  $F_0(z)$  so that both can be maximally flat filters.



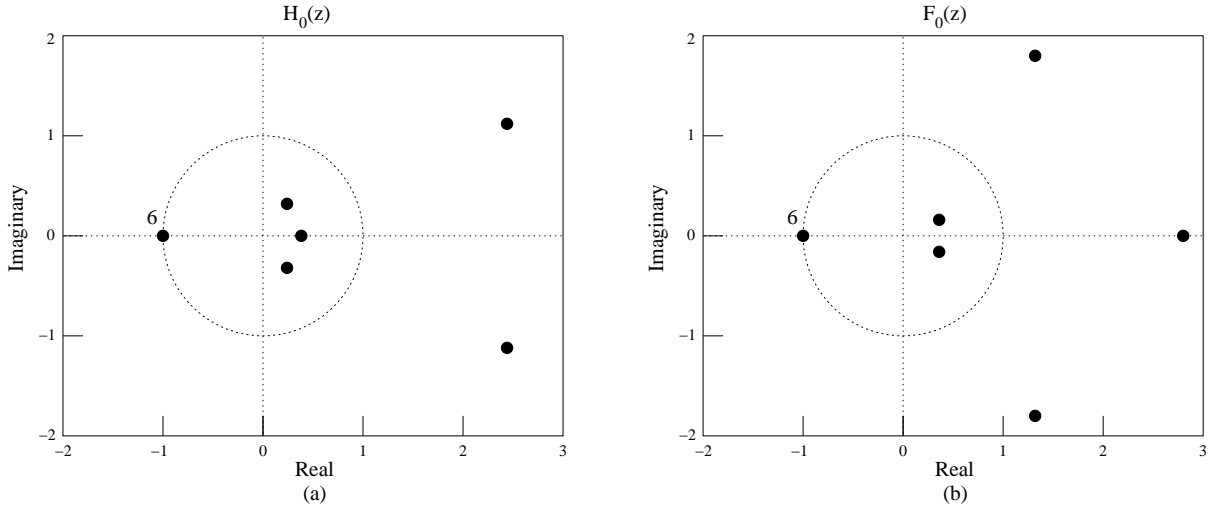
Symlets are produced by

- evenly dividing the remaining groups of four zeros of  $P(z)$  between  $H_0(z)$  and  $F_0(z)$ , and
- if there are any real-valued zeros, which will come in groups of two, the group is split between  $H_0(z)$  and  $F_0(z)$ .

An example of the Symlet wavelet filters for a Herrmann order of  $m = 6$  is shown in Figure 4.10.

Symlet wavelet filters have the following properties:

- Both  $H_0(z)$  and  $F_0(z)$  are maximally flat filters.
- Both  $H_0(z)$  and  $F_0(z)$  are almost symmetric.
- Each zero  $z$  and its complex conjugate  $\bar{z}$  stay together, which ensures real-valued filter coefficients.



**Figure 4.10:** The Symlet 6 wavelet filters in the  $z$ -plane. (a) Lowpass decomposition filter  $H_0(z)$ . (b) Lowpass reconstruction filter  $F_0(z)$ .

- Real-valued zeros  $z$  and their inverses  $z^{-1}$  do not stay together and are split between  $H_0(z)$  and  $F_0(z)$ .
- Both  $H_0(z)$  and  $F_0(z)$  are even length filters.
- $H_0(z)$  and  $F_0(z)$  have the same lengths.
- Each Herrmann order produces one unique set of filters.

## The Haar wavelet

There is only one Haar wavelet, which results from a Herrmann order of  $m = 1$ . The Herrmann filter  $P(z)$  has only two zeros, both at  $z = -1$ , which are divided between  $H_0(z)$  and  $F_0(z)$ .

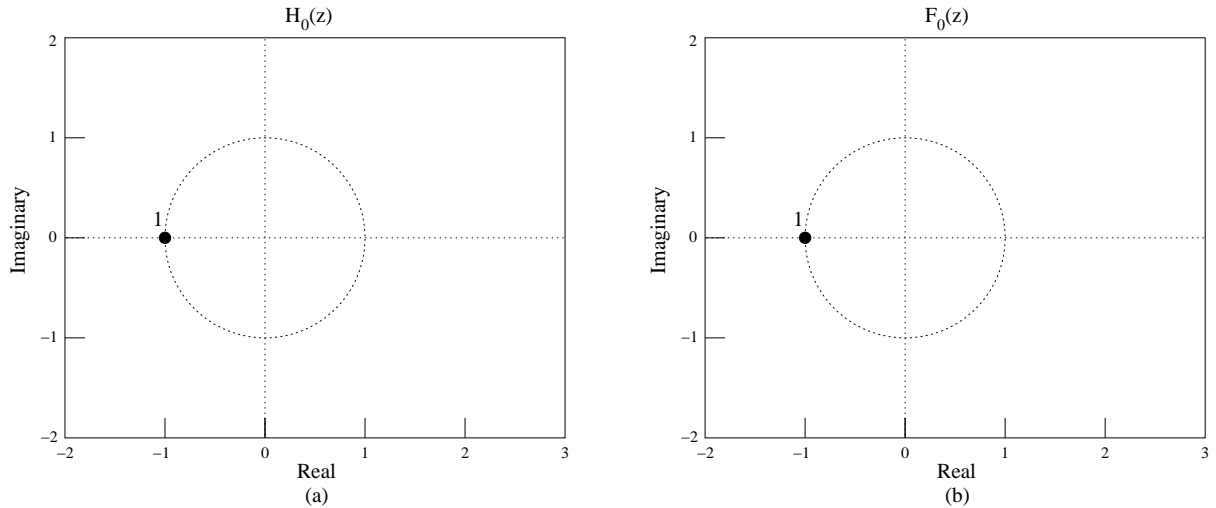
Choosing the Herrmann filter with a order of  $m = 1$

$$P(z) = \frac{1}{2}(1 + z^{-1})^2, \quad (4.22)$$

results in

$$H_0(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}) \quad \text{and} \quad F_0(z) = \frac{1}{\sqrt{2}}(1 + z^{-1}). \quad (4.23)$$

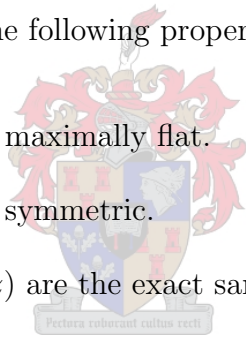
Figure 4.11 shows the Haar wavelet filters in the  $z$ -plane.



**Figure 4.11:** *The Haar wavelet filters in the  $z$ -plane. (a) Lowpass decomposition filter  $H_0(z)$ . (b) Lowpass reconstruction filter  $F_0(z)$ .*

The Haar wavelet filters have the following properties:

- Both  $H_0(z)$  and  $F_0(z)$  are maximally flat.
- Both  $H_0(z)$  and  $F_0(z)$  are symmetric.
- The filters  $H_0(z)$  and  $F_0(z)$  are the exact same filter.



## The Discrete Meyer wavelet

The Discrete Meyer wavelet, found in the MathWorks Wavelet Toolbox [41], is a FIR filter approximation of the Meyer wavelet. The discrete version has compact support in the time domain, unlike the original Meyer wavelet [22, 38]. Algorithms for implementing the Discrete Meyer wavelet transform are described in the thesis of Kolaczyk [38], but the MathWorks Wavelet Toolbox uses an algorithm described in a French book by Abry [1].

Figure 4.12 shows that the magnitude response of a Discrete Meyer wavelet filter has a steep cut-off gradient and is almost maximally flat in the bandpass region. Because it is an approximation of the Meyer wavelet which is symmetric [41], its phase response is almost linear. The Discrete Meyer wavelet has an equivalent Herrmann order of  $m \approx 31$  and is, because of the above-mentioned qualities, almost an ideal halfband filter, in comparison to other wavelets which generally have much shorter filter lengths.

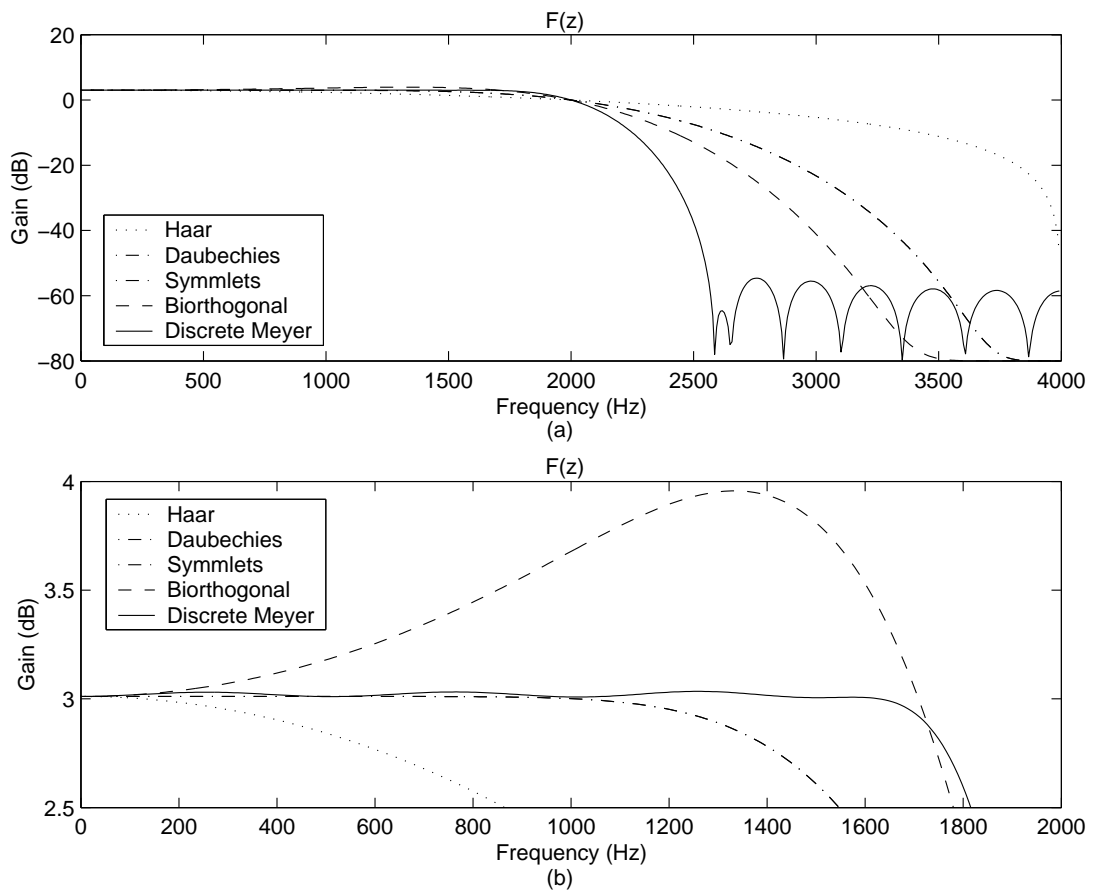


## Comparing the wavelet filters

The following wavelets are compared:

- The Haar wavelet (Herrmann order of  $m = 1$ ).
- The Daubechies 6 wavelet (Herrmann order of  $m = 6$ ).
- The Symlet 6 wavelet (Herrmann order of  $m = 6$ ).
- The Biorthogonal 1 wavelet (Herrmann order of  $m = 6$ ).
- The Discrete Meyer wavelet (which is equivalent to a Herrmann order of  $m \approx 31$ )

Figure 4.12 shows the magnitude response of the lowpass reconstruction filters  $F_0(z)$  of the different wavelets.



**Figure 4.12:** (a) The magnitude response of the  $F_0(z)$  filter of the different wavelets.  
(b) A closer look at the bandpass region.

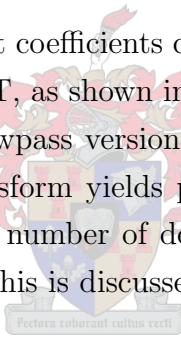
The sampling frequency in Figure 4.12 is chosen to be  $F_S = 8$  kHz. The gain in dB is shown twice, in Figure 4.12(a) to investigate the cut-off gradient and in Figure 4.12(b) to examine the flatness of the bandpass region.

Figure 4.12(a) shows that the Discrete Meyer filter has the steepest cut-off gradient, while the Haar wavelet has a gradual cut-off gradient. The Daubechies and Symlet filters have the exact same magnitude response, and differ only in their phase response.

Figure 4.12(b) shows that the Daubechies and Symlet wavelet filters have a maximally flat magnitude response, whereas the Biorthogonal filters are not nearly flat. The Discrete Meyer filter is almost maximally flat with a ripple clearly visible. Although the Haar wavelet filter is maximally flat, it is a poor halfband filter because of its short length.

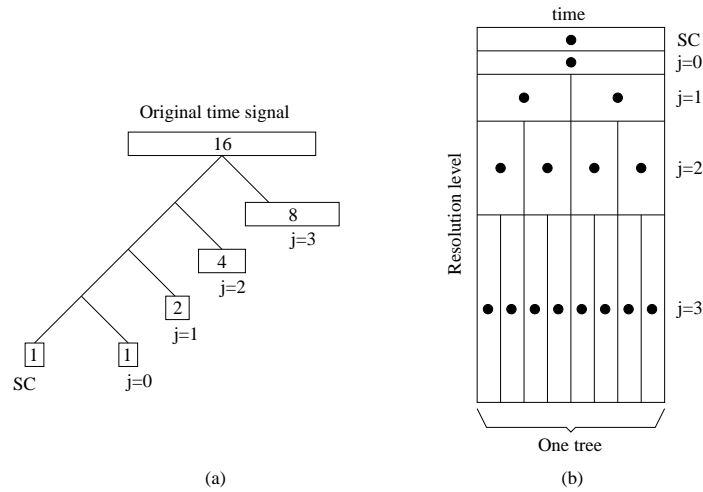
## 4.4 Decomposition levels

This section describes how wavelet coefficients can be interpreted in terms of the number of decomposition levels. The DWT, as shown in the Figure 4.1, splits the data a number of times into a highpass and a lowpass version. This is called the number of decomposition levels  $J$ . The wavelet transform yields perfect reconstruction for any number of decomposition levels. A different number of decomposition levels does however lead to different wavelet coefficients and this is discussed below.



### 4.4.1 Full wavelet decomposition

The maximum number of decomposition levels is  $J_{MAX} = \log_2 N$ , with  $N$  the total number of wavelet coefficients (equal to the number of samples in the signal). Figure 4.13(a) shows an example of the full decomposition tree of a discrete-time signal with the maximum number of decomposition levels  $J = J_{MAX}$ . The wavelet decomposition tree corresponds directly to the decomposition filter bank in Figure 4.1. The length of the example discrete-time signal is 16, which is halved after every decomposition because of downsampling. A full decomposition results in one scaling coefficient (SC). Figure 4.13(b) shows the corresponding *time-frequency view* of the coefficients. A full decomposition results in one binary *tree of coefficients*, which spans the total time-length of the original signal.



**Figure 4.13:** (a) The full wavelet decomposition with the maximum number of decomposition levels  $J = J_{MAX}$ . The length of the original signal is 16, which is halved after every decomposition. There is one scaling coefficient (SC). (b) The time-frequency tiling view of the DWT shown in (a) consists of one tree.

#### 4.4.2 $J$ -level decomposition

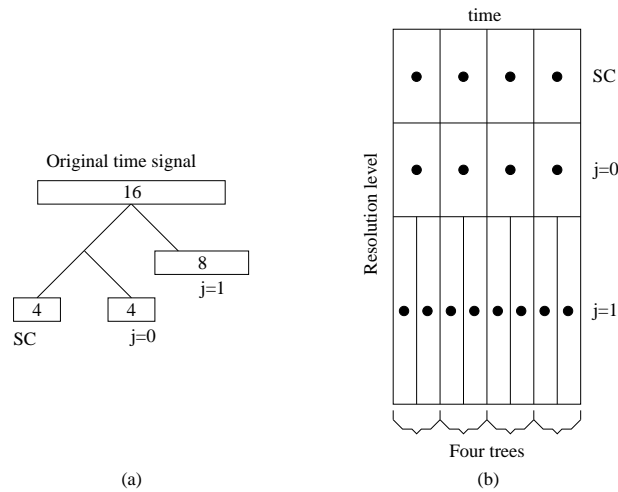
The number of binary trees of coefficients  $T$  depend on the chosen number of decomposition levels  $J$  and the total number of wavelet coefficients  $N$ , and is calculated as

$$T = \frac{N}{2^J} = \frac{2^{J_{MAX}}}{2^J} . \quad (4.24)$$

If the original signal is a time signal, the chosen number of decomposition levels  $J$  determine the time-span of each tree of coefficients.

Figure 4.14(a) shows an example of a wavelet decomposition with two decomposition levels. The number of scaling coefficients corresponds directly to the number of trees of coefficients, therefore there are four scaling coefficients according to from (4.24) and also a *forest* of four trees of coefficients as shown in Figure 4.14(b). Each tree only spans a quarter of the time-length of the original signal.

The time-frequency tiling views (Figures 4.13(b) and 4.14(b)) are constructed from the wavelet decomposition trees (Figures 4.13(a) and 4.14(a)), by using a resolution index  $j = 0, 1, \dots, J - 1$  and the scaling coefficients (SC). The number of wavelet coefficients are therefore equal to the number of samples in the discrete-time domain.



**Figure 4.14:** (a) This wavelet decomposition is two levels deep ( $J = 2$ ), resulting in four scaling coefficients. (b) The time-frequency tiling view of the DWT shown in (a) consists of a forest of four trees.

Note that the highest two resolution levels in Figure 4.13(b) and 4.14(b) are exactly the same coefficients. Introducing more levels of decomposition only changes the coefficients of the lower resolution levels. The surface areas of the tiles in the time-frequency view of Figure 4.13(b) and Figure 4.14(b) are equal because as time resolution increases, frequency resolution decreases. Figures 4.13(a) and 4.14(a) show that, independent of the number of decomposition levels, the original signal must have a length which is a power of two.

### 4.4.3 The decomposition of speech

Speech enhancement algorithms such as STSA techniques are frame-based, where each consecutive frame is transformed into the Fourier domain. It is necessary to develop a frame-based framework for wavelet-based speech enhancement. This is developed in Section 6.3.

It should be noted that the frames should not be windowed as with Fourier-based techniques. This is because of the multiresolution representation of the DWT, which has a fine time resolution at high resolution levels. We are looking for an equivalent to non-overlapping, rectangular windowed frames.

## Frame-by-frame full decompositions

One way to do a frame-based DWT on speech, similar to the frame-based Fourier techniques, is to divide the time signal into frames, which are then fully decomposed ( $J = J_{MAX}$ ) to create a tree of coefficients for every frame.

However, this will create unwanted edge-effects within the DWT of each segment. These edge-effects are introduced in the filtering step, because of the discontinuities associated with any form of extension. Because the methods analysed in this study rely on statistics, the edge-effects will influence the signal/noise classification of the algorithms. Attenuating the edge-effects is also not an option, as this disrupts the perfect reconstruction of the IDWT and leads to distortion. This method, however, can be implemented in real-time, where the frames are streamed to the denoising algorithm.

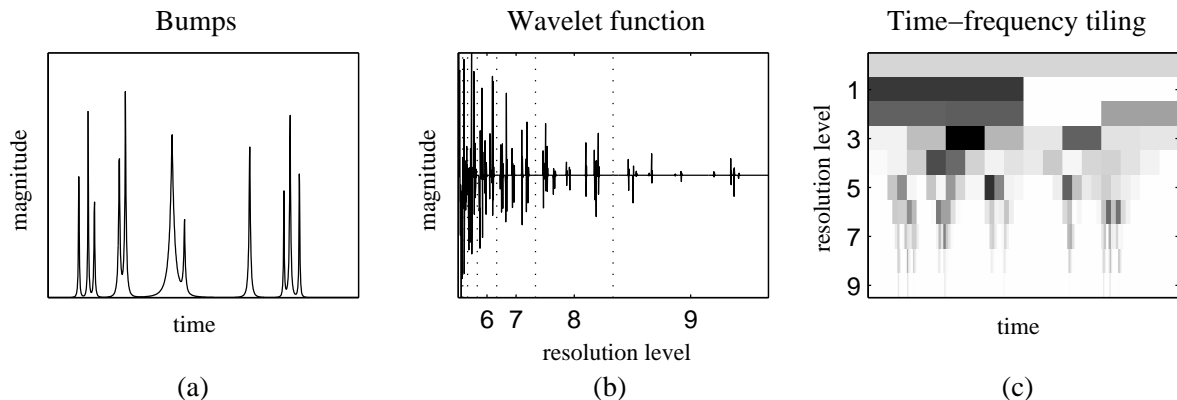
## Decomposition of a whole sentence

Another method to create a frame-based DWT for speech is to take the DWT of the whole sentence (zero-padded to have a length which is a power of two) with a chosen number of decomposition levels smaller than the maximum number,  $J < J_{MAX}$ . This results in a forest of trees of coefficients as shown in Figure 4.14(b) which are consecutive non-overlapping trees. This is equivalent to non-overlapping frames.

Since the DWT is computed on the whole sentence, the only place where edge-effects are introduced is at the beginning and end of the sentence. The sentence can easily be chosen to begin and end with silence. Edge-effects are not noticeable in almost zero-valued signals such as these silent regions. Edge-effects are therefore not a factor in such speech enhancement. This method, however, cannot be implemented in real-time, because it requires future knowledge of the sentence. Wavelet-based speech enhancement in this study is implemented by using the *decomposition of a whole sentence* because of the importance of statistics rather than real-time implementation.

## 4.5 Statistical properties of the DWT

The wavelet coefficients can be viewed in two ways. The first view is to see the coefficients as a *wavelet function* which is the output of the DWT and the second view is to sort the coefficients in a *time-frequency tiling* view. These are shown in Figure 4.15 and described below.



**Figure 4.15:** *The different views of the wavelet transform of a typical real-world signal. (a) The Bumps signal from the Donoho-Johnstone software. (b) The wavelet function view. (c) The time-frequency tiling view.*

The Donoho-Johnstone [16] *Bumps* test signal, viewed as a time signal and shown in Figure 4.15(a), is used as an example for this discussion. The Daubechies 4 (Herrmann order  $m = 4$ ) wavelet is used in a full decomposition DWT. The test signal has a length of 1024 samples which leads to  $J = 10$  resolution levels ( $j = 0, 1, \dots, 9$ ).

The wavelet function is shown in Figure 4.15(b). The highest resolution level is shown to the right of the rightmost dotted line. This resolution level contains half of the wavelet coefficients and represents the entire time-length of the signal. It is a filtered version of the original time signal. The other resolution levels, which is seen between the dotted lines, have a similar interpretation. Wavelet coefficients within a resolution level are filtered and compact versions of the original time signal. It is clearly seen that coefficients from higher resolution levels have much lower values than that of low resolution levels.

The time-frequency tiling view is shown in Figure 4.15(c). It should be noted that the time-frequency tiling view has a resolution level (or scale) axis instead of the normal frequency axis. This makes more sense, since the frequency responses of wavelet filters typically overlap. This is seen in Figure 4.12, where for example the Haar wavelet filters are far from being ideal symmetric halfband filters. Each resolution level can, however, be associated with a certain frequency band. All resolution levels in Figure 4.15(c) are, for viewing purposes, incorrectly displayed with equal widths. The coefficients are also normalised within resolution level, for displaying purposes.

The wavelet coefficients of real-world signals share certain properties. The description thereof is based on [14] and [47]. The DWT of real-world signals typically has the following two primary properties:

- **P1 Locality:**

Each wavelet atom (basis function)  $\psi_i$  is localised in time (or spatial location) and frequency. This can be seen in Figure 4.15(c), where each block (wavelet coefficient) is localised in time and frequency (resolution level).

- **P2 Multiresolution:**

The wavelet atoms  $\psi_i$  are shrunk or expanded to analyse the signal at a nested set of resolution levels. The atoms are shifted within each resolution level. This allows the DWT to match both short-duration and long-duration signal components at specific time locations. The DWT representation is narrow-band at low frequencies with longer time intervals. At high frequencies it is wide-band with shorter time intervals. The bandwidth of adjacent resolution levels differs with one octave. Figure 4.15(c) shows how the time resolution increases at higher frequencies.

Properties P1 and P2 lead to a natural arrangement of the wavelet coefficients in a binary tree structure<sup>1</sup>. The wavelet coefficients of real-world signals can be modelled as random variables, which tend to have certain properties. Looking at the individual coefficients, the third primary property of the DWT is deduced.

- **P3 Compression:** The DWT compresses real-world signals, therefore the wavelet coefficients tend to be sparse. There are a large number of small coefficients, and a small number of large coefficients. The wavelet coefficients are therefore non-Gaussian in nature (the histogram of a coefficient over a number of observations tend to be more peaky and heavy-tailed than a Gaussian density). Looking at Figure 4.15(b) the small number of large coefficients can be seen, especially at higher resolution levels.

An assumption can be made that the wavelet coefficients tend to be decorrelated. Although it is a fair assumption to view the DWT as a decorrelator, the transform cannot completely decorrelate a signal. A residual dependency structure remains between the coefficients, implying that they are not statistically independent.

This results in the *secondary properties* of the DWT. These describe the intercoefficient dependencies.

---

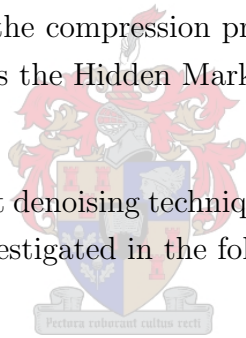
<sup>1</sup>In the 2-dimensional DWT of images, P1 and P2 lead to a quad-tree structure with 3 sub-bands representing horizontal, vertical and diagonal edges [47].

- **S1 Clustering:** If a coefficient is large/small, its neighbouring coefficients within the same resolution level also tend to be large/small. These clusters are clearly seen in Figure 4.15(b).
- **S2 Persistence:** Coefficients tend to propagate across scale. If a parent coefficient is large/small, its children coefficients also tend to be large/small. Figure 4.15(c) shows that large coefficients tend to have a pyramid shape. This type of structure in the time-frequency tiling view implies persistence.

Compression P3, clustering S1 and persistence S2 are the basic properties that Shapiro [51] captured in his revolutionary *zerotree wavelet image compression* technique<sup>2</sup>. This algorithm captures both the non-Gaussian statistics of the individual wavelet coefficients and the intercoefficient dependencies in compressing images.

Both the primary and secondary properties of the DWT are utilised in the different denoising techniques. Even the most basic denoising method of zeroing coefficients below a certain threshold makes use of the compression property. Highly computationally intensive training algorithms (such as the Hidden Markov Tree method) have been developed to capture persistence.

Different state-of-the-art wavelet denoising techniques, which make use of these properties in one way or the other, are investigated in the following chapter.




---

<sup>2</sup>The JPEG2000 image compression standard is based on this [39].



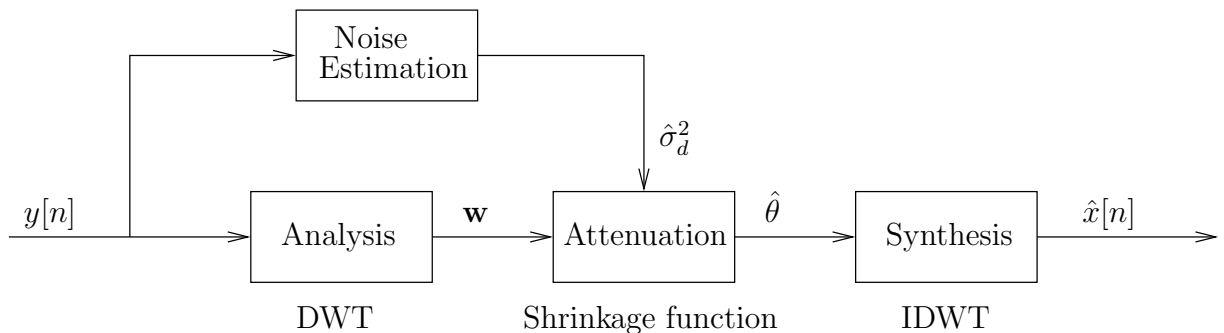
# Chapter 5

## Wavelet-based signal denoising

### 5.1 General signal denoising

This chapter is concerned with wavelet-based denoising techniques. Wavelet-based denoising is widely used for image denoising. This research, however, investigates wavelet-based speech denoising. The current wavelet-based denoising techniques for general signals are now described, and these are applied to speech in Chapter 6.

As described in Section 1.2, wavelet-based denoising consists of three steps, namely *forward transformation*, *attenuation* and *inverse transformation*. All wavelet denoising methods are described by this framework and they differ only in the attenuation step.



**Figure 5.1:** *The flowchart of wavelet-based signal denoising.*

### 5.1.1 Forward transformation

The noise  $d[n]$  is assumed to be additive, therefore the observed signal is modelled as in (2.1) as

$$y[n] = x[n] + d[n] , \quad (5.1)$$

or in vector notation,

$$\mathbf{y} = \mathbf{x} + \mathbf{d} , \quad (5.2)$$

where vectors  $\mathbf{y} = \{y[n]\}_{n=0}^{N-1}$ ,  $\mathbf{x} = \{x[n]\}_{n=0}^{N-1}$  and  $\mathbf{d} = \{d[n]\}_{n=0}^{N-1}$  represent the *noisy*, *clean* and *noise* discrete-time signals respectively, with  $N$  the length of the signals. The enhanced signal  $\hat{x}[n]$  is represented by  $\hat{\mathbf{x}}$  in vector notation. The noise  $d[n]$  is assumed to be zero-mean Gaussian noise. It is also assumed to be statistically independent and identically distributed (iid).

The forward transformation or analysis step of wavelet-based denoising is the *discrete wavelet transform* (DWT). The real-valued vector  $\mathbf{w}$  containing the noisy wavelet coefficients can be computed by multiplying orthogonal matrix  $\mathbf{W}$  with the noisy signal  $\mathbf{y}$ ,

$$\mathbf{w} = \mathbf{W}\mathbf{y} . \quad (5.3)$$

This process of computing the wavelet coefficients (the DWT) is described in Chapter 4. Because the DWT is a linear transform [11],

$$\begin{aligned} \mathbf{w} &= \mathbf{W}\mathbf{x} + \mathbf{W}\mathbf{d} \\ &= \boldsymbol{\theta} + \sigma_d \mathbf{W}\mathbf{z} , \end{aligned} \quad (5.4)$$

with  $\boldsymbol{\theta}$  the clean (unobserved) wavelet coefficients,  $\sigma_d$  the standard deviation of the noise and  $\mathbf{z}$  a vector of zero-mean unity variance Gaussian noise.

The noisy coefficients (5.4) can be written in a “signal” plus “noise” form [11], as

$$\mathbf{w} = \boldsymbol{\theta} + \sigma_d \mathbf{z}^* . \quad (5.5)$$

Here  $\mathbf{z}^*$  is also a zero-mean unity variance Gaussian noise process which is still uncorrelated with  $\boldsymbol{\theta}$ .

### 5.1.2 Attenuation step

The aim of wavelet-based denoising is to estimate the unobserved clean signal  $\boldsymbol{\theta}$ . The attenuation step of wavelet-based denoising takes the form of a *shrinkage function*. It

forms an estimate  $\hat{\boldsymbol{\theta}}$  of the clean wavelet coefficients from  $\mathbf{w}$  and  $\hat{\sigma}_d^2$ . The attenuation step is described in detail in Section 5.2.

### 5.1.3 Inverse transformation

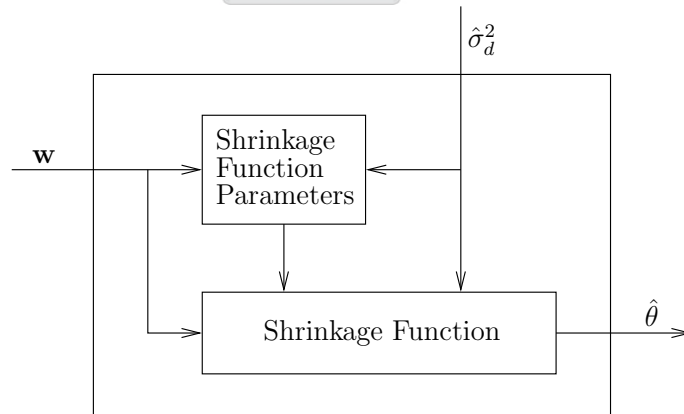
The inverse transformation or synthesis step is the *inverse discrete wavelet transform* (IDWT). It reconstructs the estimated clean signal from the modified coefficients  $\hat{\boldsymbol{\theta}}$ , as

$$\hat{\mathbf{x}} = \mathbf{W}^T \hat{\boldsymbol{\theta}} . \quad (5.6)$$

The matrix  $\mathbf{W}^T$  represents the IDWT which is described in Chapter 4.

## 5.2 Attenuation

A noisy signal is transformed into the wavelet domain, where the coefficients are attenuated on an individual basis, as shown in Figure 5.2. Large coefficients are assumed to contain mostly signal energy and are left unattenuated. Coefficients that are sufficiently small will typically be the noise components and are muted. The different denoising techniques make use of the properties of the wavelet coefficients of real-world signals which are described in Section 4.5.



**Figure 5.2:** *The flowchart of the wavelet-based attenuation step.*

Figure 5.2 shows that the attenuation step of wavelet-based denoising is twofold. The first step is to calculate the shrinkage function parameters via a denoising rule. The second step is to alter the noisy wavelet coefficients  $\mathbf{w}$  with the shrinkage function. Different shrinkage functions are described in Section 5.3, whereafter the shrinkage rules, namely

VisuShrink, SureShrink, HybridSure, Wiener, GMM, HMM and HMT are described in Sections 5.4 to 5.10.

### 5.3 The shrinkage functions

A shrinkage function forms an estimated clean coefficient  $\hat{\theta}_i$  from each noisy wavelet coefficient  $w_i$ ,

$$\hat{\theta}_i = \Theta(w_i) . \quad (5.7)$$

Four shrinkage functions are investigated<sup>1</sup>, namely the *hard*, *soft*, *one-slope* and *two-slope* shrinkage functions, and they are shown in Figure 5.3.

#### Hard shrinkage function $\Theta^H(\mathbf{w})$

The hard shrinkage function has a threshold parameter  $\lambda$  and is given by [26]

$$\Theta^H(\mathbf{w}) = \begin{cases} \mathbf{w}, & |\mathbf{w}| > \lambda \\ 0, & |\mathbf{w}| \leq \lambda \end{cases} . \quad (5.8)$$

Wavelet coefficients with a magnitude below the threshold  $\lambda$  are therefore zeroed, while the rest are left unchanged.

#### Soft shrinkage function $\Theta^S(\mathbf{w})$

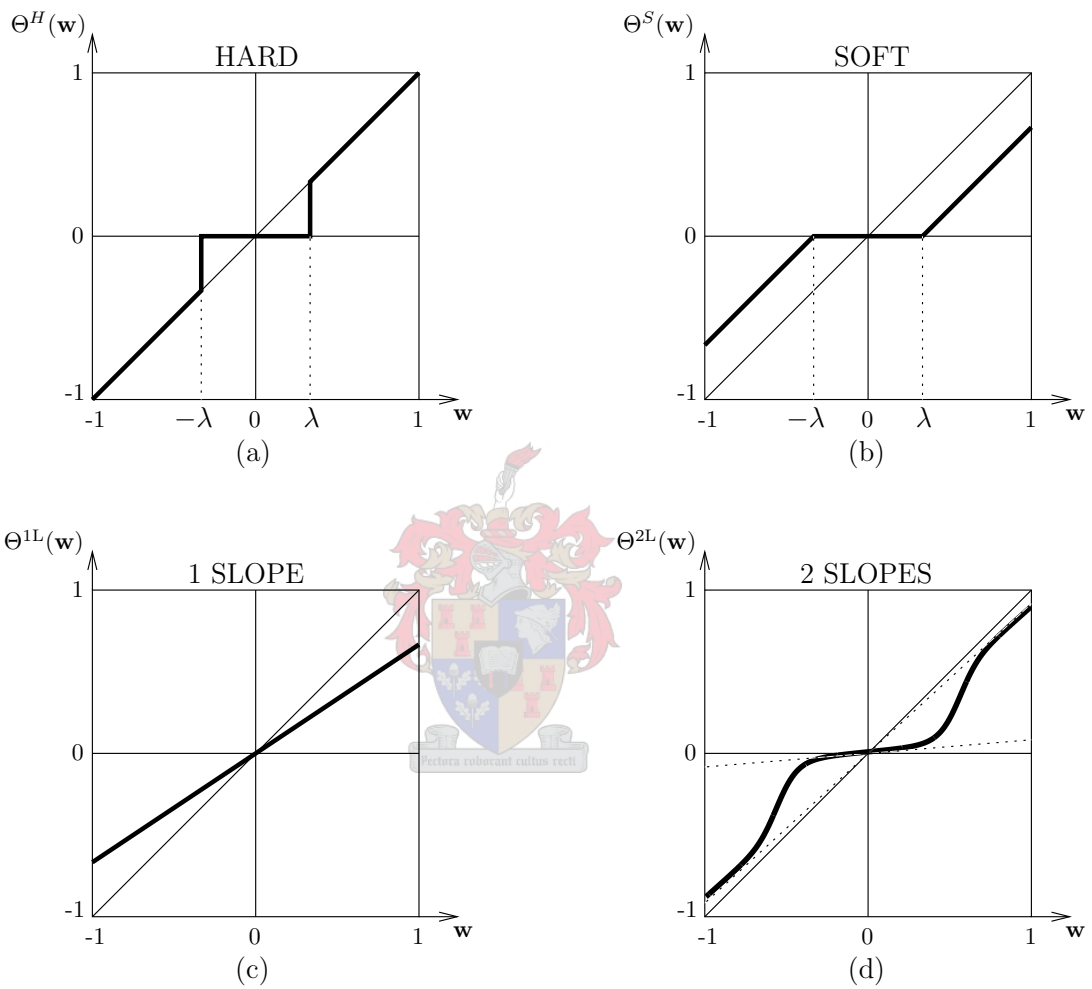
The soft shrinkage function has a threshold parameter  $\lambda$  and is given by [26]

$$\Theta^S(\mathbf{w}) = \begin{cases} \text{sign}(\mathbf{w}) (|\mathbf{w}| - \lambda), & |\mathbf{w}| > \lambda \\ 0, & |\mathbf{w}| \leq \lambda \end{cases} . \quad (5.9)$$

It is similar to the hard version, except that large coefficients are also attenuated.

---

<sup>1</sup>Other shrinkage functions include the *Semisoft* and the *Garrote* shrinkage functions [26].



**Figure 5.3:** (a) The hard shrinkage function  $\Theta^H(\mathbf{w})$  with threshold  $\lambda$ . (b) The soft shrinkage function  $\Theta^S(\mathbf{w})$  with threshold  $\lambda$ . (c) The one-slope shrinkage function  $\Theta^{1L}(\mathbf{w})$ . (d) The two-slope shrinkage function  $\Theta^{2L}(\mathbf{w})$  is an interpolation between the two dotted line slopes.

### One-slope shrinkage function $\Theta^{1L}(\mathbf{w})$

This shrinkage function scales the coefficients by  $\frac{\sigma_x^2}{\sigma_d^2 + \sigma_x^2}$ . It therefore has a signal variance parameter  $\sigma_x^2$  and a noise variance parameter  $\sigma_d^2$  and is given by

$$\Theta^{1L}(\mathbf{w}) = \left[ \frac{\sigma_x^2}{\sigma_d^2 + \sigma_x^2} \right] \mathbf{w} . \quad (5.10)$$

It does not distinguish between large and small coefficients, but suppresses all coefficients based on the signal-to-noise ratio.

### Two-slopes shrinkage function $\Theta^{2L}(\mathbf{w})$

This shrinkage function forms a smooth interpolation between two lines with slopes  $\frac{\sigma_S^2}{\sigma_d^2 + \sigma_S^2}$  and  $\frac{\sigma_L^2}{\sigma_d^2 + \sigma_L^2}$ . The slopes are based on the signal-to-noise ratios for large and small coefficients. This shrinkage function can be seen as a softer version of the hard shrinkage function, with parameters that are based on statistics rather than heuristics. The two-slope shrinkage function is given by

$$\Theta^{2L}(\mathbf{w}) = \left[ P_S(\mathbf{w}) \frac{\sigma_S^2}{\sigma_d^2 + \sigma_S^2} + P_L(\mathbf{w}) \frac{\sigma_L^2}{\sigma_d^2 + \sigma_L^2} \right] \mathbf{w} . \quad (5.11)$$

The parameters  $P_S(\mathbf{w})$  and  $P_L(\mathbf{w})$  are posterior probabilities and can be interpreted as the probability of a coefficient to be either small or large. Their computation differ in each statistical algorithm, such as the GMM, HMM and HMT.

By looking at Figure 5.3(d) it is seen that the posterior probabilities determine the interpolation between the two lines. A shrinkage function with a small  $P_S(\mathbf{w})$ , which implies a large  $P_L(\mathbf{w})$ , will increase the width of the interval about zero where the shrinkage function clings to the line with the smaller slope [11].

Small and large coefficients are represented by parameters  $\sigma_S^2$  and  $\sigma_L^2$ , respectively. These parameters and the noise variance  $\sigma_d^2$  determine the slopes of the two lines. If there is little difference between small and large coefficients, the two-slope shrinkage function approximates the one-slope shrinkage function. The two-slope shrinkage function is therefore specifically designed for signals that have a significant difference between small and large coefficients.

### 5.3.1 Using the shrinkage functions

The different denoising algorithms each use specific shrinkage functions. VisuShrink and SureShrink use either the hard or soft shrinkage function. Wiener denoising uses the one-slope shrinkage function, whereas the GMM, HMM and HMT denoising algorithms use the two-slope shrinkage function.

In practice, the noise is assumed to have unity variance in order to simplify the shrinkage function thresholds, and therefore noisy coefficients  $\mathbf{w}$  must be scaled properly. Based on the representation in (5.5), this is implemented as

$$\hat{\boldsymbol{\theta}} = \hat{\sigma}_d \Theta(\mathbf{w}/\hat{\sigma}_d) . \quad (5.12)$$

The input to the shrinkage function in (5.12) is the scaled noisy coefficients  $\mathbf{w}/\hat{\sigma}_d$ , while its output is multiplied by  $\hat{\sigma}_d$  to yield the estimated clean coefficients  $\hat{\boldsymbol{\theta}}$  [26].

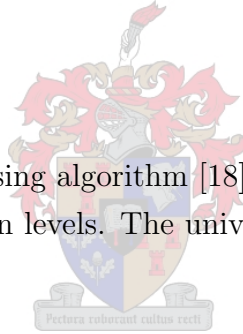
## 5.4 VisuShrink

The standard VisuShrink denoising algorithm [18] uses the soft threshold function and a single threshold for all resolution levels. The universal threshold for a length  $N$  signal is given as

$$\lambda = \sqrt{2 \ln N} . \quad (5.13)$$

This threshold is used for all resolution levels higher than the low-frequency cut-off level,  $j_0$ , which implies that coefficients in levels  $j < j_0$  are left unattenuated [18].

VisuShrink is an estimator that achieves low variance at the expense of bias [11]. The VisuShrink estimator uses a global threshold and does not reduce the mean-square error as much as adaptive thresholding techniques such as SureShrink, which uses separate thresholds for each resolution level [8, 25].



## 5.5 SureShrink and HybridSure

SureShrink and HybridSure [19] are very similar, since both choose a threshold value  $\lambda$  that minimises *Stein's Unbiased Risk Estimate* (SURE) [25, 26]. This threshold  $\lambda$  depends on the resolution level.

Based on the form of (5.5), the estimated clean wavelet coefficient vector  $\hat{\boldsymbol{\theta}}$  can be written as the sum of the observed noisy wavelet coefficient vector  $\mathbf{w}$  and a general  $R^N \rightarrow R^N$  function  $\mathbf{g}(\mathbf{w}) = \{g_i(\mathbf{w})\}_{i=1}^N$  [25]:

$$\hat{\boldsymbol{\theta}} = \Theta(\mathbf{w}) = \mathbf{w} + \mathbf{g}(\mathbf{w}) . \quad (5.14)$$

Stein showed that for almost any shrinkage function  $\Theta(\mathbf{w})$  and assuming unity variance noise ( $\hat{\sigma}_d^2 = 1$ ), the expected loss/risk is estimated as

$$E\{ \|\Theta(\mathbf{w}) - \boldsymbol{\theta}\|_2^2 \} = N + 2\{\nabla \cdot \mathbf{g}(\mathbf{w})\} + E\{ \|\mathbf{g}(\mathbf{w})\|_2^2 \} \\ \text{with } \nabla \cdot \mathbf{g}(\mathbf{w}) \equiv \sum_{i=0}^{N-1} \frac{\partial g_i}{\partial w_i} . \quad (5.15)$$

The formula for SureShrink depends on the chosen shrinkage function and the noise estimate. The soft shrinkage function is used in the following derivation. Recall from (5.9) that the soft shrinkage function can be written as

$$\Theta_i^S(w_i) = \begin{cases} w_i - \lambda \cdot \text{sign}(w_i), & |w_i| > \lambda \\ 0, & |w_i| < \lambda \end{cases} . \quad (5.16)$$

From (5.14),

$$g_i(w_i) = \begin{cases} -\lambda \cdot \text{sign}(w_i), & |w_i| > \lambda \\ -w_i, & |w_i| < \lambda \end{cases} \Rightarrow \frac{\partial g_i}{\partial w_i} = \begin{cases} 0, & |w_i| > \lambda \\ -1, & |w_i| < \lambda \end{cases} \quad (5.17)$$

$$\|g_i(w_i)\|_2^2 = \begin{cases} \lambda^2, & |w_i| > \lambda \\ |w_i|^2, & |w_i| < \lambda \end{cases} = \{\min(|w_i|, \lambda)\}^2 . \quad (5.18)$$

Substituting this into (5.15) yields

$$E\{ \|\Theta(\mathbf{w}) - \boldsymbol{\theta}\|_2^2 \} = \text{SURE}(\mathbf{w}, \lambda), \quad \text{with} \\ \text{SURE}(\mathbf{w}, \lambda) = N - 2\{\text{number of } w_i : |w_i| \leq \lambda\} \\ + \sum_{i=0}^{N-1} \{\min(|w_i|, \lambda)\}^2 . \quad (5.19)$$



SureShrink uses a different threshold for each resolution level. This threshold,  $\lambda_j$ , is chosen as the value that minimises  $\text{SURE}(\mathbf{w}_j, \lambda)$ , with  $\mathbf{w}_j$  being the wavelet coefficients of resolution level  $j$ . The threshold is computed as

$$\lambda_j = \arg \min_{\lambda \geq 0} \text{SURE}(\mathbf{w}_j, \lambda) . \quad (5.20)$$

SureShrink implies using this threshold in the soft shrinkage function  $\Theta^S(\mathbf{w}_j)$  on each resolution level.

If the wavelet coefficients within a resolution level are sparse, SureShrink performs poorly [25]. This usually occurs at high resolution levels where coefficients contain primarily noise. Therefore the sparseness of the resolution level has to be checked first.

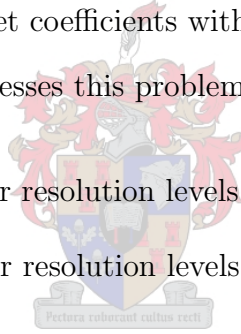
The specific resolution level is labelled as being sparse if

$$\frac{1}{N_j} \sum_{i=1}^{N_j} \left[ \left( \frac{w_i}{\hat{\sigma}_d} \right)^2 - 1 \right] \leq \frac{(\log_2 N_j)^{\frac{3}{2}}}{\sqrt{N_j}} . \quad (5.21)$$

Here  $N_j$  is the number of wavelet coefficients within the resolution level.

The HybridSure algorithm addresses this problem by

- using SureShrink (5.20) for resolution levels that are not sparse, and
- using VisuShrink (5.13) for resolution levels that are sparse.



## 5.6 Wavelet-based Wiener denoising

The wavelet-based Wiener denoising algorithm<sup>2</sup> models the wavelet coefficients as Gaussian random variables. The algorithm is implemented here as a resolution level dependent one-slope shrinkage function. The first step is to estimate  $\sigma_{j;y}^2$ , the variance of the *noisy* coefficients of resolution level  $j$ , as [25]

$$\hat{\sigma}_{j;y}^2 = \frac{1}{N_j} \sum_{i=1}^{N_j} w_i^2 , \quad (5.22)$$

with variable  $i = 1, 2, \dots, N_j$  referring to the wavelet coefficients within resolution level  $j$ , and  $N_j = 2^j$  the number of coefficients within resolution level  $j$ .

---

<sup>2</sup>The wavelet-based Wiener denoising algorithm is similar to the Wiener suppression rule which is derived in [58, 59, 60].

The clean variance  $\sigma_{j;x}^2$  is estimated, as in [14], by subtracting the noise variance  $\hat{\sigma}_d^2$  from the noisy variance  $\hat{\sigma}_{j;y}^2$ , as follows

$$\hat{\sigma}_{j;x}^2 = \max(\hat{\sigma}_{j;y}^2 - \hat{\sigma}_d^2, 0) . \quad (5.23)$$

Denoising is done via the one-slope shrinkage function by using Wiener filtering, as

$$\Theta^{1L}(w_i) = \frac{\hat{\sigma}_{j;x}^2}{\hat{\sigma}_{j;x}^2 + \hat{\sigma}_d^2} w_i . \quad (5.24)$$

## 5.7 Statistical models in the wavelet domain

The wavelet-based Wiener denoising method models the wavelet coefficients as Gaussian random variables. Since they typically contain a mixture of small and large values, the coefficients should be more accurately described by non-Gaussian statistics.

The dependencies between wavelet coefficients are completely characterised by the joint probability density function  $f(\mathbf{w})$  of all the wavelet coefficients  $\mathbf{w} = \{w_i\}$ . This complete joint density function has two major drawbacks. It is computationally intractable and it cannot be estimated robustly [14].

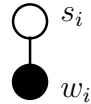
On the other extreme, it is simple to model the coefficients as statistically independent with  $f(\mathbf{w}) = \prod_i f(w_i)$  but it disregards the inter-coefficient dependencies.

The aim of a good statistical model is to capture only the key dependencies. The secondary properties of the DWT, described in Section 4.5, are the natural candidates. Clustering (S1) suggests that coefficients can have strong dependencies within resolution levels [14]. Persistence (S2) implies that wavelet coefficients are statistically dependent along the branches of the binary wavelet tree [48].

Three statistical models are described which capture the non-Gaussian statistics of wavelet coefficients. Two of these also model the key dependencies. The Gaussian Mixture Model (GMM) models the coefficients as non-Gaussian and independent. The Hidden Markov Model (HMM) models the coefficients as non-Gaussian and having clusters within the resolution levels. The Hidden Markov Tree Model (HMT) models the coefficients as non-Gaussian, having clusters within the resolution levels and having persistence across scale.

### 5.7.1 The hidden state variable

Each of the statistical methods (GMM, HMM and HMT) uses the concept of a hidden state variable  $s_i$  associated with each of the wavelet coefficients  $w_i$ . Figure 5.4 shows the wavelet coefficient (black dot)  $w_i$  as the real-valued observation,  $w_i \in R$ . The hidden state variable (white dot)  $s_i$  is unobserved and can only take on discrete values,  $s_i \in 1, 2, \dots, M$ , where  $M$  is the possible number of states.



**Figure 5.4:** *Associated with each wavelet coefficient (black dot)  $w_i$  is a hidden state variable (white dot)  $s_i$ .*

The statistical methods in this study assume two possible states for each wavelet coefficient, namely small (S) and large (L). The value of the state variable  $s_i$  influences the assumed density function of the coefficient  $w_i$ .

### 5.7.2 The low-resolution cut-off level $j_0$

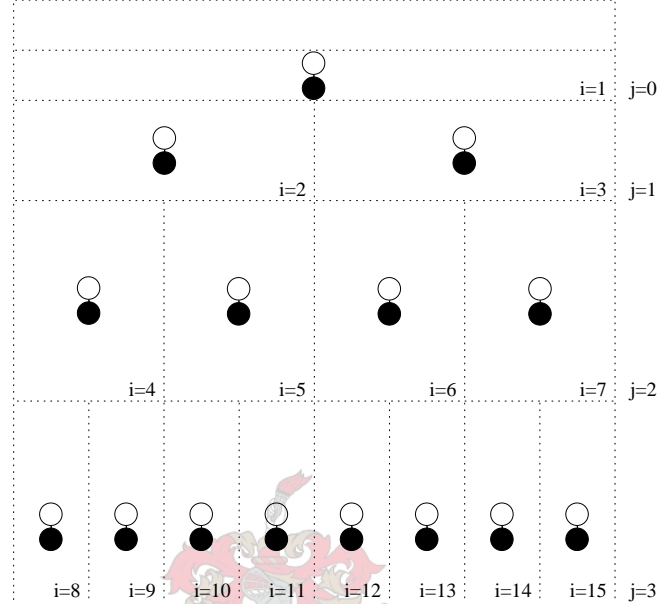
The scaling coefficient and the coefficients from lowest resolution levels  $j < j_0$  are typically not zero-mean and should not be shrunk towards zero [8, 14]. These coefficients are relatively noise free and are therefore used in their unprocessed form in the IDWT of the synthesis step [14]. In this study, the low-resolution cut-off level is chosen as in [14] to be  $j_0 = 3$ , which implies that the scaling coefficient and resolution levels  $j = 0, 1$  and  $2$  are not attenuated.

## 5.8 Gaussian Mixture Models (GMMs)

### 5.8.1 The GMM structure

Crause et al. [14] discuss an Independent Mixture (IM) model, where the wavelet coefficients are modelled as independent random variables. The IM model is implemented here as a Gaussian Mixture Model and is shown in Figure 5.5. The GMM algorithm attempts to capture the compression P3 property of real-world wavelet coefficients by modelling

them as being non-Gaussian. The coefficients are assumed to have a similar distribution within each resolution level, which is described by a two-state zero-mean Gaussian Mixture Model. This is an improvement on the Wiener denoising algorithm, described in Section 5.6, which models the coefficients with a single Gaussian density. A binary tree of coefficients is denoised by using a different GMM model for each resolution level.



**Figure 5.5:** *The GMM associates a hidden state variable (white dot) with each coefficient (black dot). There are no connections between the hidden states, because they are modelled as being independent.*

The GMM uses an indexing scheme as shown in Figure 5.5.

- The index  $i = 1, 2, \dots, 2^J - 1$  refers to all coefficients within the binary tree, apart from the scaling coefficient, with  $J$  being the number of resolution levels.
- The index  $j = 0, 1, \dots, J - 1$  refers to the  $J$  resolution levels.
- The index  $L_j = 2^j$  is the leftmost index of resolution level  $j$ , and  $R_j = 2^{j+1} - 1$  is the rightmost index of resolution level  $j$ .
- The set  $[j] = \{L_j, L_j + 1, \dots, R_j\}$  is defined as all values of index  $i$  within the resolution level  $j$ .
- The size of the set  $[j]$  is  $N_j = 2^j$ , which is the number of wavelet coefficients within resolution level  $j$ .
- The operator  $j = \ell(i)$  determines the resolution level  $j$  associated with index  $i$ .

## 5.8.2 Modelling the GMM non-Gaussianity

The non-Gaussianity is modelled by associating a discrete hidden state variable  $s_i \in \{1, 2, \dots, M\}$  with each coefficient, where  $M$  is the number of possible states. Each state is associated with a Gaussian probability density function. Coefficient  $w_i$  therefore has  $M$  conditional probability density functions,

$$\begin{aligned} f(w_i | s_i = m, \mathcal{M}) &= \left( \frac{1}{2\pi\sigma_{i;m}^2} \right)^{\frac{1}{2}} \exp \left[ -\frac{(w_i - \mu_{i;m})^2}{2\sigma_{i;m}^2} \right] \\ &= g(w_i; \mu_{i;m}, \sigma_{i;m}^2) . \end{aligned} \quad (5.25)$$

The vector  $\mathcal{M}$  contains the model parameters and is described in Section 5.8.3. The parameters  $\mu_{i;m}$  and  $\sigma_{i;m}^2$  are the mean and variance of the Gaussian distribution, with  $i$  the wavelet coefficient index and  $m$  the state of the hidden state variable  $s_i$ . The function  $g(\cdot)$  refers to the Gaussian distribution function and is defined in (5.25).

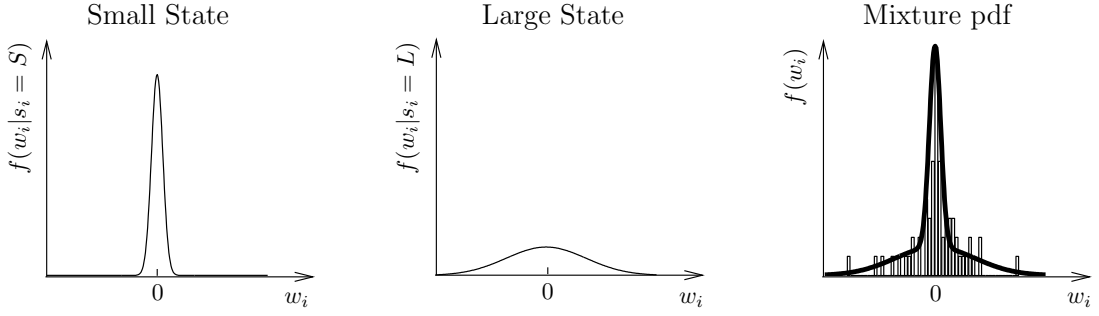
The state variable  $s_i$  also has an associated probability mass function (pmf)  $P(s_i = m)$ , with  $\sum_{m=1}^M P(s_i = m) = 1$ . This pmf can be described as the probability that state variable  $s_i$  is in state  $m$ . The marginal pdf  $f(w_i)$  of coefficient  $w_i$  is approximated by the Gaussian mixture,

$$f(w_i) = \sum_{m=1}^M P(s_i = m) f(w_i | s_i = m). \quad (5.26)$$

A two-state zero-mean Gaussian mixture model is an appropriate approximation of the non-Gaussian statistics of real-world wavelet coefficients [14]. This is because of the compression property of the DWT (P3). Most wavelet coefficients are small and are therefore in a *small* state ( $s_i = S$ ). These coefficients are responsible for a Gaussian distribution with a small variance  $\sigma_{i;S}^2$ . The few large coefficients are in a *large* state ( $s_i = L$ ). These coefficients are responsible for a Gaussian distribution with a large variance  $\sigma_{i;L}^2$ . From this definition, the Gaussian parameters have the following properties:

- $\sigma_{i;L}^2 > \sigma_{i;S}^2$
  - $\mu_{i;m} = 0$ , for all  $i$  and  $m$ , and
  - $P(s_i = S) + P(s_i = L) = 1$  for all  $i$ .
- (5.27)

The set of possible states  $\{1, 2, \dots, M\}$  is therefore replaced by the more intuitive set  $\{S, L\}$ . The conditional pdfs  $f(w_i | s_i = \{S, L\})$  are shown in Figure 5.6(a) and (b).



**Figure 5.6:** (a) The small-state conditional pdf  $f(w_i|s_i = S)$ . (b) The large-state conditional pdf  $f(w_i|s_i = L)$ . (c) The two-state Gaussian mixture model  $f(w_i)$  (thick line) is a good approximation of real-world wavelet coefficients (histogram). The histogram is that of resolution level 7 of the Bumps signal decomposed with the Daubechies 20 wavelet.

Figure 5.6(a) shows an example of a *small-state* low-variance ( $\sigma_{i,S}^2$ ) Gaussian conditional pdf  $f(w_i|s_i = S)$  of the set of coefficients  $w_i$  for  $i \in [j]$ . Figure 5.6(b) shows the corresponding *large-state* conditional pdf  $f(w_i|s_i = L)$ . Figure 5.6(c) shows a comparison between the marginal pdf  $f(w_i)$  from (5.26) and the histogram of real-world coefficients. This pdf has a large peak at zero (because of the large number of small wavelets) and heavy tails (because of the small number of large wavelets).

### 5.8.3 The GMM model parameters

Because the GMM models the wavelet coefficients as two-state zero-mean Gaussian independent random variables, the model parameters can be chosen as

- $P(s_i = L)$  for all  $i$ , and
- $\sigma_{i,m}^2$  for all  $i$  and  $m$ .

This results in three independent parameters per wavelet coefficient,  $P(s_i = L)$ ,  $\sigma_{i,S}^2$  and  $\sigma_{i,L}^2$ , which makes it difficult to train the model. The number of parameters can be reduced by assigning the above parameters per resolution level instead of per coefficient.

The GMM model parameters for each modelled resolution level  $j = j_0, j_0 + 1, \dots, J - 1$  are therefore chosen as:

- $\boxed{P_j(L)}$  where  $P_j(m) = P(s_i = m)$  with  $j = \ell(i)$  and  $m \in \{S, L\}$ . It is the probability mass function for state variables  $s_i$  within resolution level  $j$ . Parameter  $P_j(S)$  is calculated from (5.27) as  $P_j(S) = 1 - P_j(L)$ .

- $\sigma_{j;m}^2$  with  $m \in \{S, L\}$ . It is the variance parameters of the conditional probability density functions in (5.25) for resolution level  $j$ .

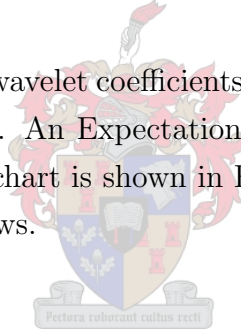
These parameters are grouped into a model parameter vector

$$\mathcal{M} = \{P_j(L), \sigma_{j;S}^2, \sigma_{j;L}^2\}, \text{ with } j = 0, 1, \dots, J - 1. \quad (5.28)$$

Because the parameters are tied within resolution level, the GMM model has two variance parameters  $\sigma_{j;S}^2$  and  $\sigma_{j;L}^2$  and one probability parameter  $P_j(L)$  per resolution level. As discussed in Section 5.7.2, coefficients within resolution levels  $j < j_0$  are left unattenuated. The GMM is therefore effectively only trained on  $J' = J - j_0$  resolution levels. Parameter  $J'$  is referred to as the effective number of resolution levels. The GMM thus has  $3J'$  parameters in total.

## 5.8.4 Training the GMM

The GMM algorithm views the wavelet coefficients within each resolution level as different observations of the same model. An Expectation-Maximisation (EM) algorithm is used to train the GMM, and its flowchart is shown in Figure 5.7. A description of each of the steps in the block diagram follows.



### Initialise the GMM model $\mathcal{M}$

Because the component pdfs are assumed to have zero means, a sophisticated initialisation algorithm such as binary-split or  $K$ -means, which focuses on the component means, is inappropriate. Initialisation is dependent on the resolution level  $j$  in accordance with the GMM model definition.

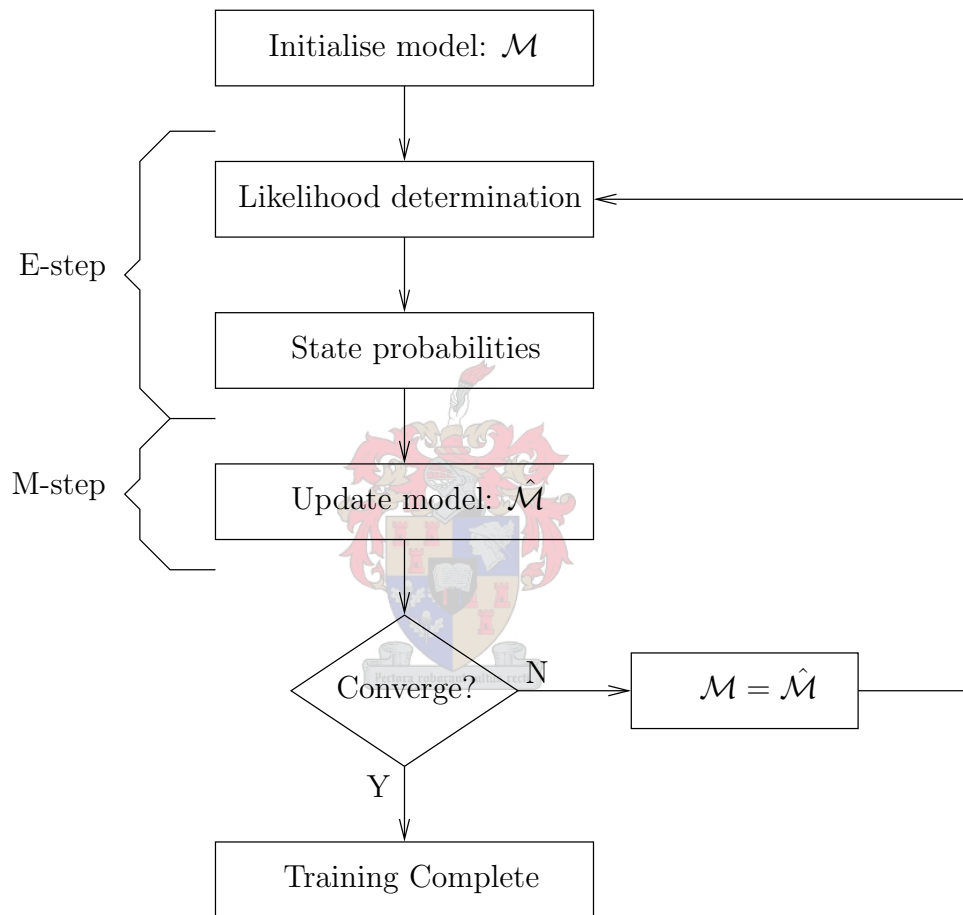
The initial probabilities are set to be equal, as

$$\left. \begin{array}{l} P_j(S) = 0.5 \\ P_j(L) = 0.5 \end{array} \right\} \text{ for all } j. \quad (5.29)$$

The variance parameters are initialised from the *noisy* variance  $\sigma_{j;y}^2$  of resolution level  $j$ , which is calculated using (5.22) as [12]

$$\left. \begin{array}{l} \sigma_{j;S}^2 = \sigma_{j;y}^2/2 \\ \sigma_{j;L}^2 = 2\sigma_{j;y}^2 \end{array} \right\} \text{ for all } j. \quad (5.30)$$

This ensures that  $\sigma_{j;S}^2$  starts off smaller than  $\sigma_{j;L}^2$ .



**Figure 5.7:** *The flowchart of the Expectation-Maximisation (EM) algorithm for Gaussian Mixture Models (GMMs).*



## GMM likelihood determination

The probability density function of each observed wavelet coefficient  $w_i$  given the model  $\mathcal{M}$  is

$$f(w_i|\mathcal{M}) = \sum_{m \in \{S,L\}} P_j(m) g(w_i; 0, \sigma_{j;m}^2), \quad \text{with } j = \ell(i). \quad (5.31)$$

## GMM state probabilities

The probability that the  $i$ th wavelet coefficient is in state  $m$ , given its observed value  $w_i$  and the GMM model  $\mathcal{M}$  is given by

$$P(s_i = m|w_i, \mathcal{M}) = \frac{P_j(m) g(w_i; 0, \sigma_{j;m}^2)}{f(w_i|\mathcal{M})}, \quad \text{with } j = \ell(i). \quad (5.32)$$

This posterior probability is used in the shrinkage rule to discriminate between large and small coefficients and is also used in the EM training algorithm.

## Updating the GMM model parameters

The model parameters are updated in the M-step of training, based on the posterior state probabilities and the coefficient values, as follows:

$$\hat{P}_j(m) = \frac{1}{N_j} \sum_{i \in [j]} P(s_i = m|w_i, \mathcal{M}) \quad (5.33)$$

$$\hat{\sigma}_{j;m}^2 = \frac{\sum_{i \in [j]} P(s_i = m|w_i, \mathcal{M}) w_i^2}{\sum_{i \in [j]} P(s_i = m|w_i, \mathcal{M})} \quad (5.34)$$

## Convergence

The GMM training is done independently for each resolution level. Let  $\mathbf{w}_j$  be the wavelet coefficients on resolution level  $j$ , while  $\mathcal{M}_j$  represents the GMM parameters associated with this level. The log-likelihood of the coefficients  $\mathbf{w}_j$ , given the model  $\mathcal{M}_j$ , is given by  $\log f(\mathbf{w}_j|\mathcal{M}_j) = \sum_{i \in [j]} \log f(w_i|\mathcal{M}_j)$ , since the coefficients are assumed to be independent. With Expectation-Maximisation training, each iteration produces an increase in log-likelihood, which is the difference of the log-likelihoods of the current and previous iterations,

$$\text{Increase in log-likelihood} = \log f(\mathbf{w}_j|\mathcal{M}_j^k) - \log f(\mathbf{w}_j|\mathcal{M}_j^{k-1}) . \quad (5.35)$$

Variable  $k$  refers to the training iteration index.. Vector  $\mathcal{M}_j^k$  is the model parameters of the current iteration, whereas  $\mathcal{M}_j^{k-1}$  is the model parameters of the previous iteration. As the EM algorithm converges to a local optimum, the difference in the log-likelihood decreases. Training is stopped when the difference falls below  $10^{-5}$ , as in [12].

### 5.8.5 GMM denoising

The GMM uses the two-slope shrinkage function to denoise a corrupted signal. As described in Section 5.8.4, the parameters of the shrinkage function are based on the unobserved clean signal. In practice, however, these parameters have to be estimated from the observed noisy data. The GMM model is therefore first trained on the noisy data. The clean GMM model is then estimated from the noisy model, which leads to the shrinkage function parameters<sup>3</sup>.

The noisy model

$$\mathcal{M}_y = \{P_{j;y}(L), \sigma_{j;S;y}^2, \sigma_{j;L;y}^2\} \quad \text{with } j = 0, 1, \dots, J-1 , \quad (5.36)$$

is trained on the noisy wavelet coefficients.

The model  $\mathcal{M}$  for the estimated clean speech is derived from this noisy model. The clean variance parameters are estimated by subtracting the estimated noise variance  $\hat{\sigma}_{j;d}^2$  from the variances of the noisy model. These variance parameters cannot be negative and are therefore calculated as [14]

$$\begin{aligned} \sigma_{j;S}^2 &= \max(\sigma_{j;S;y}^2 - \hat{\sigma}_{j;d}^2, 0) , \quad \text{and} \\ \sigma_{j;L}^2 &= \max(\sigma_{j;L;y}^2 - \hat{\sigma}_{j;d}^2, 0) . \end{aligned} \quad (5.37)$$

---

<sup>3</sup>This is also the case with HMM and HMT denoising which are described in Sections 5.9.5 and 5.10.5.

The parameters  $\sigma_{j;S}^2$  and  $\sigma_{j;L}^2$  are the *small* and *large* Gaussian variance parameters of the clean model. The clean probability  $P_j(L)$  is assumed to be unaffected by the noise [14], therefore  $P_j(L) = P_{j;y}(L)$ . The GMM model parameter vector for the underlying clean speech is therefore

$$\mathcal{M} = \{P_j(L), \sigma_{j;S}^2, \sigma_{j;L}^2\} \text{ , with } j = 0, 1, \dots, J - 1 \text{ .} \quad (5.38)$$

These clean GMM parameters are used in the two-slope shrinkage function, so that the clean wavelet coefficients are estimated as

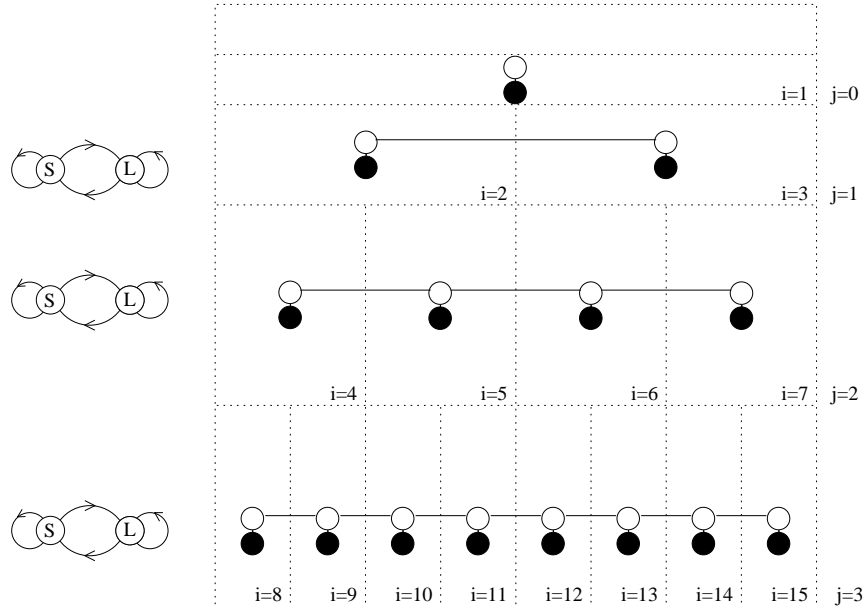
$$\Theta^{2L}(w_i) = \left[ P_j(S) \frac{\sigma_{j;S}^2}{\hat{\sigma}_{j;d}^2 + \sigma_{j;S}^2} + P_j(L) \frac{\sigma_{j;L}^2}{\hat{\sigma}_{j;d}^2 + \sigma_{j;L}^2} \right] w_i \text{ .} \quad (5.39)$$

$P_j(S) = P(s_i = S | \mathbf{w}, \mathcal{M})$  and  $P_j(L) = P(s_i = L | \mathbf{w}, \mathcal{M})$  with  $j = \ell(i)$  are the probabilities that the state variable  $s_i$  associated with coefficient  $i$  is in a *small* or *large* state, respectively. The noise variance estimate  $\hat{\sigma}_{j;d}^2$  is pre-estimated for each resolution level and is computed as described in Section 1.2.1. The two-slope shrinkage function in (5.39) is based on the weighted Wiener shrinkage rule derived in Section 5.6. The GMM is expected to be more accurate than Wiener denoising, as long as the underlying data has a zero-mean non-Gaussian nature.

## 5.9 Hidden Markov Models (HMMs)

### 5.9.1 The HMM structure

Crause et al. [14] proposed a *Hidden Markov Chain Model*, where the hidden state variables  $s_i$  are connected horizontally within each resolution level. Although proposed in [14], the implementation of the algorithm is novel in this study. The Hidden Markov Model is shown in Figure 5.8 with the state variables connected with first-order Markovian dependencies in a horizontal chain. This model treats the wavelet coefficients as dependent within the resolution level, but independent from scale to scale. The coefficients are modelled using a Hidden Markov Model (HMM) structure within each scale. This is shown on the left of Figure 5.8, where the two possible states, *small* and *large*, are connected in an ergodic structure. In practice, the wavelet-based HMM consists of a number of independent hidden Markov models, which depends on the number of resolution levels. The indexing of the coefficients  $w_i$  and the state variables  $s_i$  uses the same notation as that of the GMM, described in Section 5.8.1.



**Figure 5.8:** The HMM associates a hidden state variable (white dot) with each wavelet coefficient (black dot). The hidden state variables are connected horizontally to capture clustering. The two-state ergodic HMMs, used to model the coefficients within each resolution level, are shown on the left.

### 5.9.2 Modelling the HMM non-Gaussianity

The HMM models the non-Gaussian statistics in the wavelet domain in the same manner as the Gaussian Mixture Model. This is described in detail in Section 5.8.2.

### 5.9.3 The HMM model parameters

The HMM models the wavelet coefficients as two-state zero-mean Gaussian random variables, with a Markovian dependency structure within the resolution levels. The model parameters are defined for all resolution levels and given as [46]:

- $\boxed{\pi_j(L)}$  where  $\pi_j(m) = P(s_{L_j} = m)$ ,  $m \in \{S, L\}$  and  $\sum_{m \in \{S, L\}} \pi_j(m) = 1$ . The parameter  $\pi_j(m)$  is the initial state distribution, which is the probability for the leftmost coefficient to be in state  $m$ .
- $\boxed{a_{mn}^{(j)} = P(s_{i+1} = n | s_i = m)}$  with  $m, n \in \{S, L\}$ ,  $a_{mn}^{(j)} \geq 0$ ,  $\sum_{n \in \{S, L\}} a_{mn}^{(j)} = 1$ , and  $j \in \ell(i)$ . The parameter  $a_{mn}^{(j)}$  is the state transition probability that the given state  $s_i = m$  is succeeded by state  $s_{i+1} = n$  in resolution level  $j$ .

- $\sigma_{j,m}^2$  with  $m \in \{S, L\}$ . The parameter  $\sigma_{j,m}^2$  is the Gaussian variance parameter of the conditional distribution  $f(w_i | s_i = m)$  of (5.25) with  $j = \ell(i)$ .

These parameters are grouped into a model parameter vector

$$\mathcal{M} = \{ \pi_j(L), a_{mn}^{(j)}, \sigma_{j,m}^2 \}, \text{ with } n, m \in \{S, L\} \text{ and } j = 0, 1, \dots, J-1. \quad (5.40)$$

The HMM has five parameters per resolution level, which are the two parameters from  $\sigma_{j,m}^2$  and, because probabilities sum to one, the one parameter from  $\pi_j(m)$  and the two from  $a_{mn}^{(j)}$ . The HMM is effectively only trained on  $J'$  resolution levels, as with the GMM. The model  $\mathcal{M}$  therefore has  $5J'$  parameters in total.

### 5.9.4 Training the HMM

The HMM algorithm is similar to the GMM algorithm. It is level dependent and each wavelet coefficient within the resolution level is seen as a different observation. An Expectation-Maximisation (EM) algorithm is used in a forward-backward manner to train the HMM. This is known as Baum-Welch re-estimation [5, 43, 46].

The flowchart of the EM algorithm for HMMs is shown in Figure 5.9. Each block in the flowchart is described in detail in the following section, which describes the training of an HMM model for a specific resolution level.

#### Initialise the model $\mathcal{M}$

Initialisation is similar to the GMM method and also dependent on the resolution level. The initial state distributions are set to be equal for all states, as

$$\pi_j(m) = 0.5 \text{ for } m \in \{S, L\}. \quad (5.41)$$

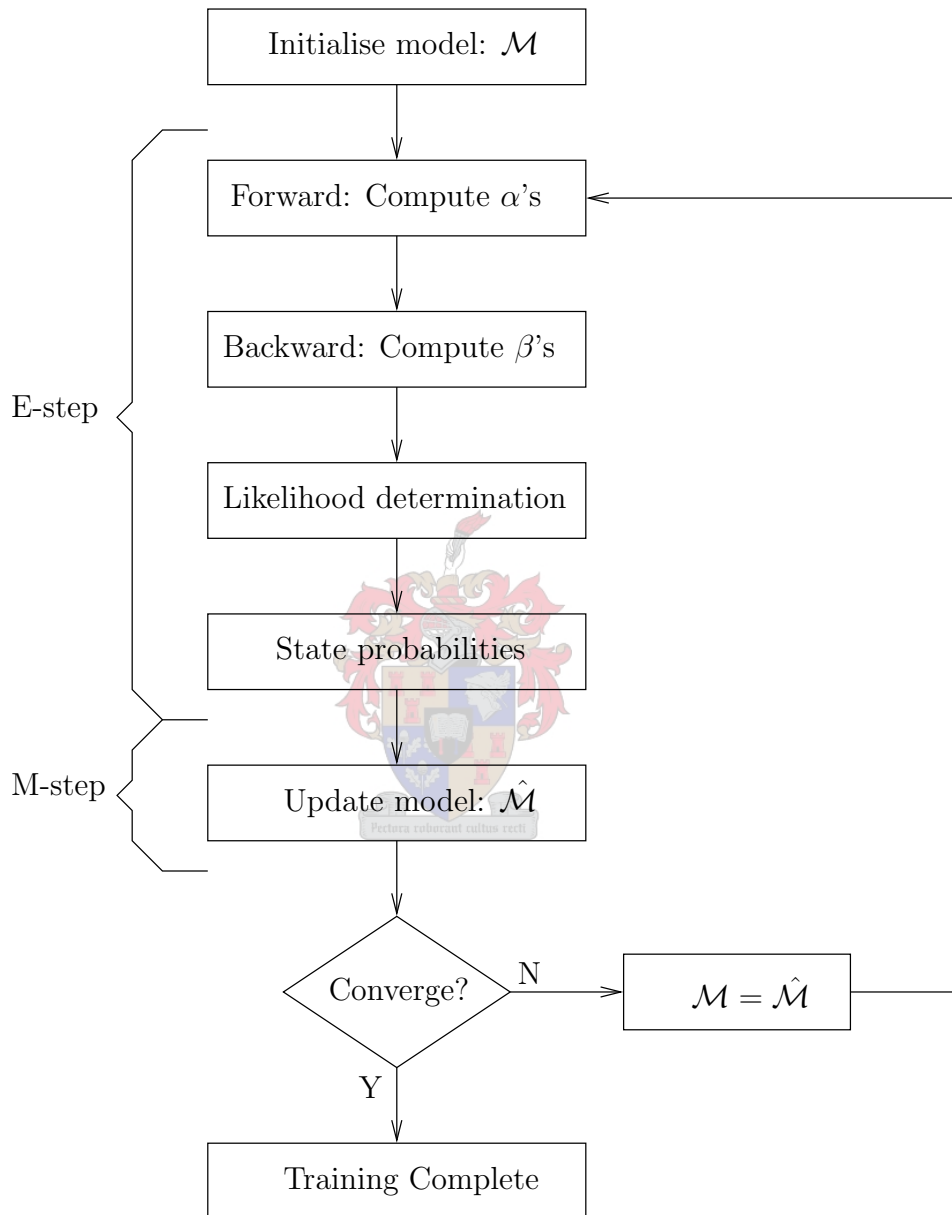
The initial probabilities are also set to be equal, as

$$a_{mn}^{(j)} = 0.5 \text{ for } m, n \in \{S, L\}. \quad (5.42)$$

The initial variance parameters are computed as for GMMs in Section 5.8.4, based on the *noisy* variance  $\sigma_{j;y}^2$  as [12]

$$\left. \begin{array}{l} \sigma_{j;S}^2 = \sigma_{j;y}^2/2 \\ \sigma_{j;L}^2 = 2\sigma_{j;y}^2 \end{array} \right\} \text{ for all } j. \quad (5.43)$$

As for GMMs, this ensures that  $\sigma_{j;L}^2 > \sigma_{j;S}^2$ .



**Figure 5.9:** *The flowchart of the Expectation-Maximisation (EM) algorithm for Hidden Markov Models (HMMs).*

### HMM forward variable $\alpha$

The forward variable  $\alpha$  is computed by moving from left to right within the resolution level. For each coefficient index  $i = L_j, L_j + 1, \dots, R_j$ , define the forward variable as

$$\alpha_i(m) = f(w_{L_j}, w_{L_j+1}, \dots, w_i, s_i = m | \mathcal{M}) . \quad (5.44)$$

This is the probability of the partial set of wavelet coefficients from coefficient  $w_{L_j}$  to  $w_i$  and state variable  $s_i = m$ , given the model  $\mathcal{M}$  [46].

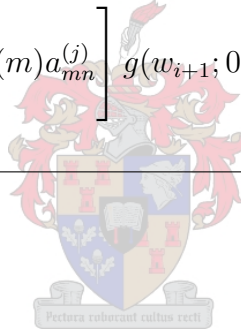
#### Forward step: Computing $\alpha$ 's

Initialisation:

$$\alpha_{L_j}(m) = \pi_j(m)g(w_{L_j}; 0, \sigma_{j;m}^2) , \text{ with } m \in \{S, L\} . \quad (5.45)$$

For all  $i = L_j, L_j + 1, \dots, R_j - 1$  with  $j \in \ell(i)$ , compute:

$$\alpha_{i+1}(n) = \left[ \sum_{m \in \{S, L\}} \alpha_i(m) a_{mn}^{(j)} \right] g(w_{i+1}; 0, \sigma_{j;n}^2) , \text{ with } n \in \{S, L\} . \quad (5.46)$$



### HMM backward variable $\beta$

For each coefficient index  $i = L_j, L_j + 1, \dots, R_j - 1$ , define the backward variable as

$$\beta_i(m) = f(w_{i+1}, w_{i+2}, \dots, w_{R_j} | s_i = m, \mathcal{M}) . \quad (5.47)$$

This is the probability of the partial set of wavelet coefficients from  $w_{i+1}$  to the rightmost coefficient  $w_{R_j}$ , given state  $s_i = m$  and the model  $\mathcal{M}$  [46].

#### Backward step: Computing $\beta$ 's

Initialisation:

$$\beta_{R_j}(m) = 1 , \text{ for } m \in \{S, L\} . \quad (5.48)$$

For all  $i = R_j - 1, R_j - 2, \dots, L_j$  with  $j \in \ell(i)$ , compute:

$$\beta_i(m) = \sum_{n \in \{S, L\}} a_{mn}^{(j)} \beta_{i+1}(n) g(w_{i+1}; 0, \sigma_{j;n}^2) , \text{ for } m \in \{S, L\} . \quad (5.49)$$

## HMM likelihood determination

Let  $\mathbf{w}_j$  be the wavelet coefficients  $\{w_{L_j}, w_{L_j+1}, \dots, w_{R_j}\}$  on resolution level  $j$ . The probability density function (pdf) of the observed coefficients  $\mathbf{w}_j$  given the model  $\mathcal{M}$  is [46],

$$f(\mathbf{w}_j|\mathcal{M}) = \sum_{m \in \{S, L\}} \alpha_i(m)\beta_i(m), \quad \text{with } j = \ell(i). \quad (5.50)$$

Since  $\alpha_i(m)$  accounts for the partial observation sequence  $w_{L_j}$  to  $w_i$  and state variable  $s_i = m$ , while  $\beta_i(m)$  accounts for the remainder of the observation sequence  $w_{i+1}$  to  $w_{R_j}$  given state  $s_i = m$ , the pdf  $f(\mathbf{w}_j|\mathcal{M})$  has the same value for any chosen  $i \in [j]$ . It is typically computed by setting  $i = R_j$ .

## HMM state probabilities

For all  $i \in [j]$  and  $m \in \{S, L\}$ , compute:

$$P(s_i = m|\mathbf{w}_j, \mathcal{M}) = \frac{\alpha_i(m)\beta_i(m)}{f(\mathbf{w}_j|\mathcal{M})}. \quad (5.51)$$

For all  $i = L_j, L_j + 1, \dots, R_j - 1$  and  $m, n \in \{S, L\}$ , compute:

$$P(s_i = m, s_{i+1} = n|\mathbf{w}_j, \mathcal{M}) = \frac{\alpha_i(m)a_{mn}^{(j)}\beta_{i+1}(n)g(w_{i+1}; 0, \sigma_{j;n}^2)}{f(\mathbf{w}_j|\mathcal{M})}. \quad (5.52)$$

## Updating the HMM model parameters

$$\hat{\pi}_j(m) = P(s_{L_j} = m|\mathbf{w}_j, \mathcal{M}) \quad (5.53)$$

$$\hat{a}_{mn}^{(j)} = \frac{\sum_{i=L_j}^{R_j-1} P(s_i = m, s_{i+1} = n|\mathbf{w}_j, \mathcal{M})}{\sum_{i=L_j}^{R_j-1} P(s_i = m|\mathbf{w}_j, \mathcal{M})} \quad (5.54)$$

$$\hat{\sigma}_{j;m}^2 = \frac{\sum_{i \in [j]} P(s_i = m|\mathbf{w}_j, \mathcal{M})w_i^2}{\sum_{i \in [j]} P(s_i = m|\mathbf{w}_j, \mathcal{M})} \quad (5.55)$$



## 5.9.5 HMM denoising

HMM denoising is similar to GMM denoising. It also estimates the underlying model parameters from the noisy data and uses the two-slope shrinkage function

$$\Theta^{2L}(w_i) = \left[ P(s_i = S | \mathbf{w}_j, \mathcal{M}) \frac{\sigma_{j;S}^2}{\sigma_d^2 + \sigma_{j;S}^2} + P(s_i = L | \mathbf{w}_j, \mathcal{M}) \frac{\sigma_{j;L}^2}{\sigma_d^2 + \sigma_{j;L}^2} \right] w_i, \quad (5.56)$$

with  $j = \ell(i)$ . The clean HMM variance parameters are estimated by subtracting the estimated noise variance  $\hat{\sigma}_{j;d}^2$  from the variances of the noisy model, as with GMMs in (5.37). The state probabilities of the noisy model are directly used for the clean model as in [14].

The posterior state probabilities,  $P(s_i = m | \mathbf{w}_j, \mathcal{M})$ , should be more accurate and refined than in the case of the GMM, as long as the HMM describes the data better, in which case the HMM should also improve denoising.

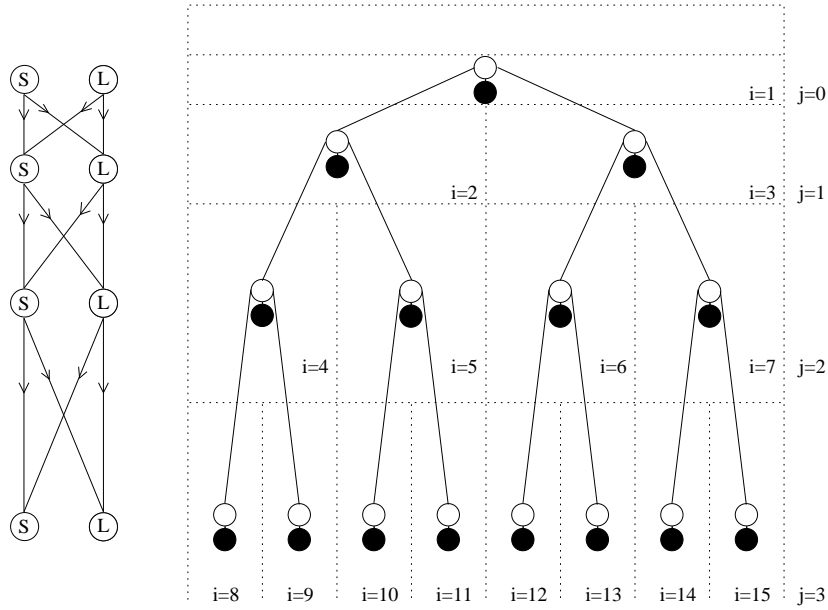
## 5.10 Hidden Markov Trees (HMTs)

### 5.10.1 The HMT structure

The Hidden Markov Tree (HMT) framework is proposed by Crause, Nowak and Baraniuk [14]. It uses a tree-structured Expectation-Maximisation (EM) algorithm to train the model. Once the model is trained, the HMT forms an estimate of  $f(\mathbf{w})$  which attempts to capture compression (P3), clustering (S1) and persistence (S2). The compression property of the DWT leads to non-Gaussian statistics of the individual wavelet coefficients. The HMT capture this in the same manner as GMMs and HMMs, by associating a discrete hidden state variable with each coefficient, which leads to modelling the coefficient as a Gaussian mixture. Again the two-state zero-mean assumption is made, which is described in Section 5.8.2. Persistence and clustering are captured by using the natural tree structure of the wavelet coefficients. The hidden state variables are connected with first-order Markovian dependencies in a binary tree structure, shown in Figure 5.10.

An abstract indexing scheme, shown in Figure 5.10, is used within the HMT framework. This is similar to that of the GMM and HMM and is summarised below:

- The index  $i = 1, 2, \dots, 2^J - 1$  refers to all coefficients within the binary tree, apart from the scaling coefficient, with  $J$  being the number of resolution levels.

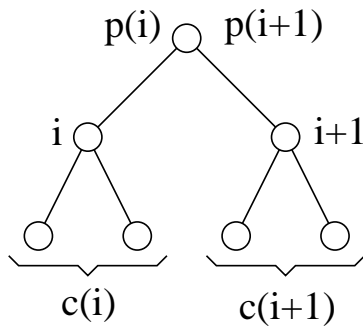


**Figure 5.10:** *The HMT associates a hidden state variable (white dot) with each wavelet coefficient (black dot). The hidden state variables are connected, with first-order Markovian dependencies (solid lines), in a binary tree structure to capture clustering and persistence. The two-state HMT model, used to model all coefficients, is shown on the left.*

- The total number of coefficients in the binary tree is  $N = 2^J - 1$ .
- The indices  $i = 2^j, 2^j + 1, \dots, 2^{j+1} - 1$  form the set  $[j]$ , which still represents the indices within resolution level  $j$ , as with GMMs and HMMs.
- The number of coefficients in  $[j]$  is  $N_j = 2^j$ .
- The function  $j = \ell(i)$  returns the resolution level  $j$  of index  $i$ , as with GMMs and HMMs.

Figure 5.11 shows that both node  $i$  and  $i + 1$  share the same parent node  $p(i) = p(i + 1)$ . Node  $i$  has two children, represented by the set of indexes  $c(i)$ . This binary tree structure is imposed on the hidden state variables and directly corresponds to the natural time-frequency tiling view of the wavelet coefficients. The node with no ancestor is the *root* node. The scaling coefficient sits above the root node and is not modelled within the HMT framework<sup>4</sup>. The root node corresponds to the coefficient representing the lowest

<sup>4</sup>The scaling coefficient is typically left unaltered in wavelet-based denoising and is therefore not included in the model [14].



**Figure 5.11:** *The binary tree of connected state variables. The white dots are the state variables, and the connecting lines represent the Markovian dependencies. The parent  $p(i)$  and child  $c(i)$  notation is shown.*

frequency band. Nodes with no children are the *leaf* nodes and correspond to the highest frequency band.

Clustering is captured by using the fact that each node, apart from the leaf nodes, has two child nodes<sup>5</sup>. The children share the same transition probabilities, allowing the model to capture clustering between these two child nodes. State variables  $s_i$  and  $s_{i+1}$  are dependent due to their joint interaction with their parent state variable  $s_{p(i)}$  [14]. This method of capturing clustering is different from the HMM. The HMT allows two neighbouring children coefficients to share statistical information, whereas the HMM uses first-order Markovian dependencies within the resolution levels.

### 5.10.2 Modelling the HMT non-Gaussianity

The HMT models the non-Gaussian statistics in the wavelet domain in a similar manner to GMMs and HMMs and this is described in Section 5.8.2.

#### Tying within scale

It is important to notice that in both image denoising and frame-based speech enhancement, there is only a single set of observed wavelet coefficients. This is the tree shown in Figure 5.10, or a two-dimensional version thereof in the case of images. Ideally, we would

---

<sup>5</sup>The 2-dimensional DWT, used for images, results in each node having 4 children. This is because of its quad-tree structure, as opposed to the binary tree structure of the 1-dimensional DWT.

like to have many such trees as training data, all with similar statistical properties. In our case, however, only one tree of coefficients is available as training data and therefore some form of averaging is needed to train the model. The coefficients within each resolution level are assumed to have similar statistical properties. An extra statistical averaging process is used to tie the coefficients within each resolution level. This averaging step is done when updating the model parameters in (5.77) and (5.80) and this is referred to as tying within scale [14].

### 5.10.3 The HMT model parameters

From the two-state zero-mean Gaussian mixture model of the individual coefficients and the Markovian binary tree structure on the hidden states, the HMT model parameters are,

- $\boxed{\pi(L)}$  where  $\pi(m) = P(s_1 = m)$ ,  $m \in \{S, L\}$  and  $\sum_{m \in \{S, L\}} \pi(m) = 1$ . The parameter  $\pi(m)$  is the state probability of the root node and is interpreted as the probability that the root node  $s_i$  is in state  $m$ .
- $\boxed{\epsilon_{i,p(i)}^{mn} = P(s_i = m | s_{p(i)} = n)}$  with  $m, n \in \{S, L\}$ ,  $\epsilon_{i,p(i)}^{mn} \geq 0$  and  $\sum_{n \in \{S, L\}} \epsilon_{i,p(i)}^{mn} = 1$ . The parameter  $\epsilon_{i,p(i)}^{mn}$  is the conditional probability that state variable  $s_i$  is in state  $m$ , given that its parent state variable  $s_{p(i)}$  is in state  $n$ . Since the root node has no parent,  $\epsilon_{1,p(1)}^{mn}$  is undefined and can be taken as 0. The transition probability is tied within scale, changing the parameter to  $\boxed{\epsilon_{(j)}^{mn} = \epsilon_{i,p(i)}^{mn}}$  for  $j = \ell(i)$ .
- $\boxed{\sigma_{j,m}^2}$  with  $m \in \{S, L\}$ . The parameter  $\sigma_{j,m}^2$  is the Gaussian variance parameter of the conditional distribution  $f(w_i | s_i = m)$  of (5.25) with  $j = \ell(i)$  (i.e. associated with resolution level  $j$ ).

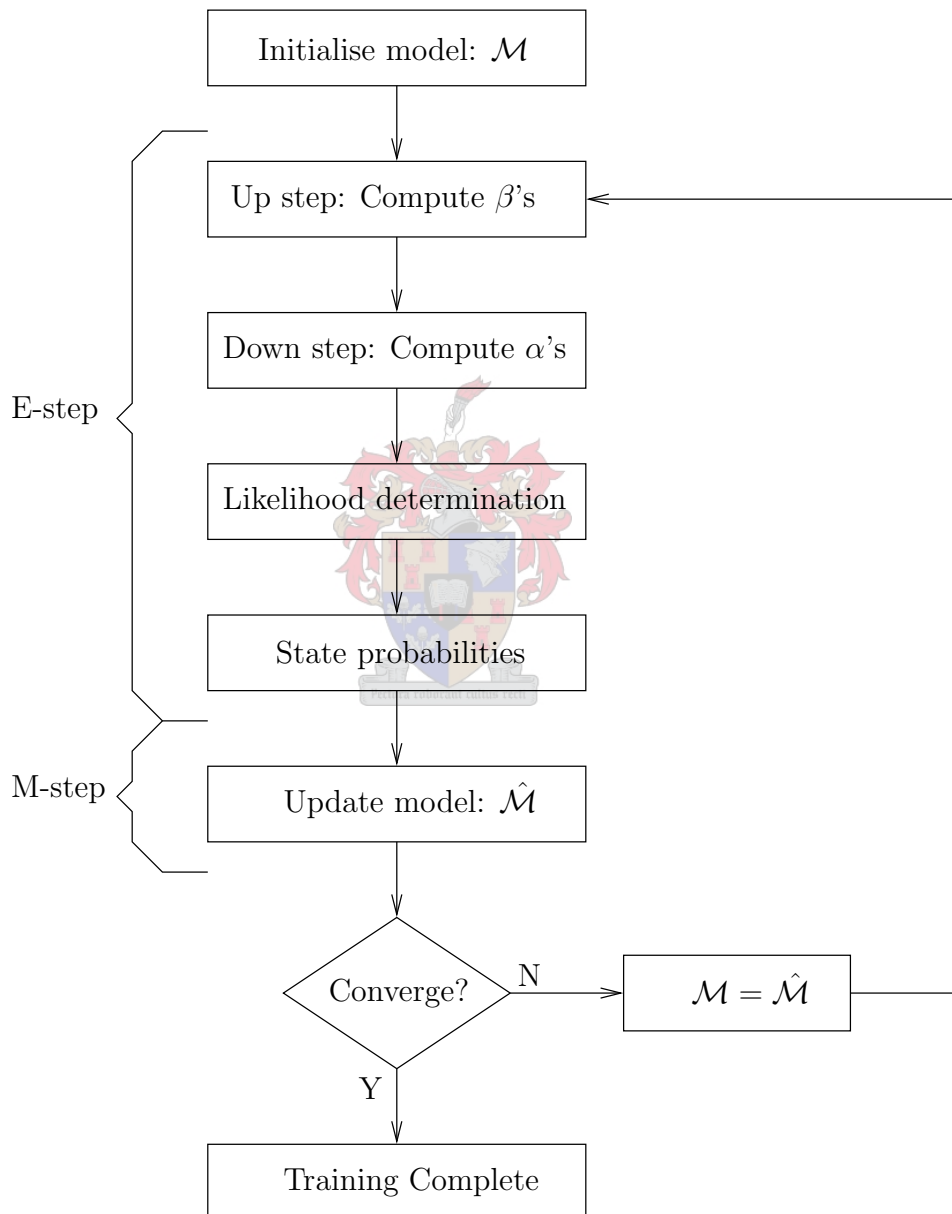
These parameters are grouped into a model parameter vector

$$\mathcal{M} = \{ \pi(L), \epsilon_{(j)}^{mn}, \sigma_{j,m}^2 \} , \text{ with } n, m \in \{S, L\} \text{ and } j = 0, 1, \dots, J-1. \quad (5.57)$$

The HMT model  $\mathcal{M}$  has  $2J$  variance parameters  $\sigma_{j,m}^2$ . Because probabilities sum to one, there are  $2(J-1)$  state transition probabilities  $\epsilon_{(j)}^{mn}$  and the single  $\pi(L)$  parameter. The HMT therefore has  $2J + 2(J-1) + 1 = 4J - 1$  parameters, where  $J$  is the total number of resolution levels. This differs from the HMM, which has  $5J'$  parameters, and the GMM, which has  $3J'$  parameters. Unlike the GMM and HMM, the HMT is trained on all resolution levels ( $j = 0, 1, \dots, J-1$ ), but coefficients within resolution levels  $j < j_0$  are left unattenuated.

### 5.10.4 HMT training via the EM algorithm

Unlike the GMM and HMM methods which are trained per resolution level, the HMT training is done on the whole binary tree of coefficients. An Expectation-Maximisation algorithm [14] is used in an *upward-downward* manner to train the HMT. The flowchart of the EM algorithm for HMTs is shown in Figure 5.12. Each block in the flowchart is described in the following section.



**Figure 5.12:** The flowchart of the Expectation-Maximisation (EM) algorithm for Hidden Markov Trees (HMTs).

## Initialise the model $\mathcal{M}$

The initialisation follows that of [12]. The state distribution of the root node  $s_1$  is set to be equal for all states, as

$$\pi(m) = 0.5 \text{ for all } m \in \{S, L\}. \quad (5.58)$$

The initial state transition probabilities are also set to be equal, as

$$\epsilon_{i,p(i)}^{mn} = 0.5 \text{ for all } m, n \in \{S, L\}. \quad (5.59)$$

The variance parameters are initialised as for GMMs and HMMs in Section 5.8.4, based on the *noisy* variance  $\sigma_{j;y}^2$  as

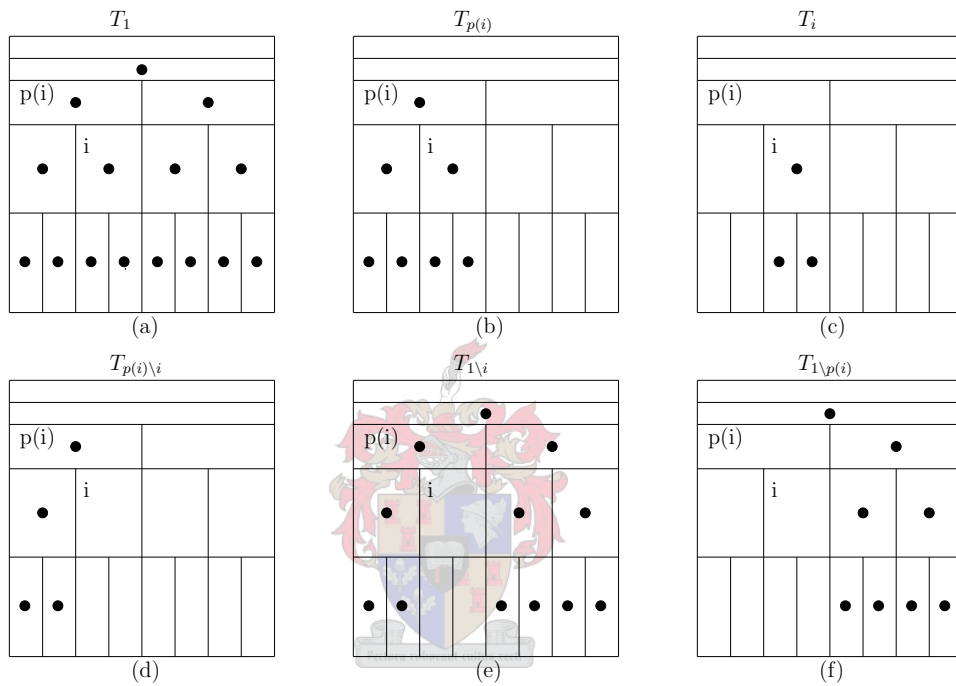
$$\left. \begin{aligned} \sigma_{j;S}^2 &= \sigma_{j;y}^2/2 \\ \sigma_{j;L}^2 &= 2\sigma_{j;y}^2 \end{aligned} \right\} \text{ for all } j. \quad (5.60)$$

As with GMMs and HMMs, this ensures that  $\sigma_{j;L}^2 > \sigma_{j;S}^2$ .

## E-step of the HMT

The EM algorithm for HMMs, described in Section 5.9.4, uses a *forward-backward* algorithm on the coefficients of each resolution level. The HMT training algorithm is similar to this, but it uses an *upward-downward* algorithm which involves all the wavelet coefficients. The forward-backward algorithm uses intermediate variables  $\alpha$  and  $\beta$ . Both of these are based on partial sets of observations. The intermediate variables for the upward-downward algorithm of HMTs also use partial sets of wavelet coefficients. These sets are arranged in a tree structure, which corresponds to the HMT model definition, and are shown in Figure 5.13.

The subtree of observed coefficients with its root at node  $i$  is defined as  $T_i$ . The subtree  $T_i$  shown in Figure 5.13(c) contains coefficient  $w_i$  and all its descendants. Thus,  $T_1$  shown in Figure 5.13(a) is the entire tree of observed wavelet coefficients, apart from the scaling coefficient which is not modelled. Also  $T_{p(i)}$ , shown in Figure 5.13(b), is the partial set containing coefficient  $w_{p(i)}$  and all of its descendants. The notation  $T_{p(i)\setminus i}$  indicates the set of observed wavelet coefficients obtained by removing the subtree  $T_i$  from  $T_{p(i)}$  and is shown in Figure 5.13(d). Similarly,  $T_{1\setminus i}$  and  $T_{1\setminus p(i)}$  are shown in Figures 5.13(e) and (f), respectively.



**Figure 5.13:** The different sets of wavelet coefficients used for HMT training (black dots). (a)  $T_1$ , the whole set of observed wavelet coefficients, apart from the scaling coefficient. (b)  $T_{p(i)}$ , the set of coefficients including  $w_{p(i)}$  and all its descendants. (c)  $T_i$ , the set of coefficients rooted at  $i$ . (d)  $T_{p(i)\setminus i}$ , the set of coefficients obtained by excluding  $T_i$  from  $T_{p(i)}$ . (e)  $T_{1\setminus i}$ , all coefficients apart from  $T_i$ . (f)  $T_{1\setminus p(i)}$ , all coefficients apart from  $T_{p(i)}$ .

*HMT upward ( $\beta$ ) variables*

Three intermediate  $\beta$  variables are used in the upward step of the EM algorithm for HMTs. For each subtree  $T_i$  with  $i = 1, 2, \dots, N$  and for all states  $m \in \{S, L\}$ , define the conditional likelihood

$$\beta_i(m) = f(T_i | s_i = m, \mathcal{M}), \quad (5.61)$$

which is the likelihood of partial set  $T_i$ , given state  $s_i = m$  and model  $\mathcal{M}$ , and

$$\beta_{i,p(i)}(m) = f(T_i | s_{p(i)} = m, \mathcal{M}), \quad (5.62)$$

which is the likelihood of partial set  $T_i$ , given parent state  $s_{p(i)} = m$  and model  $\mathcal{M}$ , and

$$\beta_{p(i)\setminus i}(m) = f(T_{p(i)\setminus i} | s_{p(i)} = m, \mathcal{M}), \quad (5.63)$$

which is the likelihood of partial set  $T_{p(i)\setminus i}$  given parent state  $s_{p(i)} = m$  and the model  $\mathcal{M}$ . These are calculated from the leaf nodes upwards to the root node. The likelihoods  $\beta_{i,p(i)}(m)$  and  $\beta_{p(i)\setminus i}(m)$  are undefined for  $i = 1$ .

**Up step: Computing the  $\beta$ 's**

Initialisation: For all leaf nodes ( $i \in [j]$ ;  $j = J - 1$ ) calculate

$$\beta_i(m) = g(w_i; 0, \sigma_{j;m}^2), \text{ with } m \in \{S, L\}. \quad (5.64)$$

Moving upwards in resolution levels ( $j = J - 2, J - 3, \dots, 1$ )

and for all  $i \in [j]$ , compute

$$\beta_{p(i)}(m) = g(w_{p(i)}; 0, \sigma_{j-1;m}^2) \prod_{i \in c(p(i))} \beta_{i,p(i)}(m), \quad (5.65)$$

$$\beta_{i,p(i)}(m) = \sum_{n \in \{S, L\}} \epsilon_{i,p(i)}^{nm} \beta_i(n), \text{ and} \quad (5.66)$$

$$\beta_{p(i)\setminus i}(m) = \frac{\beta_{p(i)}(m)}{\beta_{i,p(i)}(m)}. \quad (5.67)$$

The term  $i \in c(p(i))$  in (5.65) refers to the set containing index  $i$  and all of its siblings.

*HMT downward ( $\alpha$ ) variables*

For each subtree  $T_{1\setminus i}$  with  $i = 1, 2, \dots, N$  and for all states  $m \in \{S, L\}$ , compute the joint density

$$\alpha_i(m) = f(s_i = m, T_{1\setminus i} | \mathcal{M}), \quad (5.68)$$



which is the likelihood of the partial set  $T_{1 \setminus i}$  and that state variable  $s_i$  is in state  $m$ , given the model  $\mathcal{M}$ .

**Down step: Computing the  $\alpha$ 's**

Initialise: For the root node ( $i = 1; j = 0$ ), compute

$$\alpha_1(m) = \pi(m), \text{ with } m \in \{S, L\}. \quad (5.69)$$

For each resolution level, moving downwards ( $j = 1, 2, \dots, J - 1$ ) and for all  $i \in [j]$ , compute

$$\alpha_i(m) = \sum_{n \in \{S, L\}} \epsilon_{i, p(i)}^{mn} \alpha_{p(i)}(n) \beta_{p(i) \setminus i}(n). \quad (5.70)$$

**M-step of the HMT**

*HMT likelihood determination*

The likelihood  $f(\mathbf{w}|\mathcal{M})$  determines how well the given HMT model  $\mathcal{M}$  describes the observed wavelet coefficients  $\mathbf{w}$ . It is calculated with the help of the intermediate  $\alpha$  and  $\beta$  variables, similar to the HMM algorithm, and is used during training to test for convergence. The likelihood of  $w$  is

$$f(\mathbf{w}|\mathcal{M}) = f(T_1|\mathcal{M}) = \sum_{m \in \{S, L\}} f(s_i = m, \mathbf{w}|\mathcal{M}). \quad (5.71)$$

**HMT likelihood determination**

$$f(\mathbf{w}|\mathcal{M}) = \sum_{m \in \{S, L\}} \alpha_i(m) \beta_i(m) \quad (5.72)$$

The pdf  $f(\mathbf{w}|\mathcal{M})$  has the same value for any chosen  $i \in \{1, 2, \dots, N\}$ , because  $\alpha_i(m)$  accounts for the partial observation sequence  $T_{1 \setminus i}$  and  $\beta_i(m)$  accounts for the partial observation sequence  $T_i$ , therefore incorporating all possible state paths for any chosen  $i$ . It is computed in this project by using the root node  $i = 1$  as in [12].

*HMT posterior state probabilities*

Because of the first-order Markovian dependencies within the binary tree structure, the sets  $T_{1 \setminus i}$  and  $T_i$  are independent given  $s_i = m$  [14]. From this independence and the chain

rule of probability calculus, the state probabilities are calculated. The non-Gaussianity of the individual coefficients is modelled by using the conditional density

$$f(s_i = m, T_1 | \mathcal{M}) = \alpha_i(m) \beta_i(m) . \quad (5.73)$$

The inter-coefficient dependencies are modelled by the conditional density

$$f(s_i = m, s_{p(i)} = n, T_1 | \mathcal{M}) = \alpha_{p(i)}(n) \beta_{p(i) \setminus i}(n) \beta_i(m) \epsilon_{i,p(i)}^{mn} . \quad (5.74)$$

Here variable  $\alpha_{p(i)}(n)$  accounts for the observed set  $T_{1 \setminus p(i)}$  shown in Figure 5.13(f). Variable  $\beta_{p(i) \setminus i}(n)$  accounts for the set  $T_{p(i) \setminus i}$  shown in Figure 5.13(d) and variable  $\beta_i(m)$  handles the observations  $T_i$  shown in Figure 5.13(c). All possible state paths are therefore taken into account.

Now Bayes' rule is applied to (5.73) and (5.74) to produce the following conditional probabilities:

#### HMT conditional probabilities

$$P(s_i = m | \mathbf{w}, \mathcal{M}) = \frac{\alpha_i(m) \beta_i(m)}{f(\mathbf{w} | \mathcal{M})} \quad (5.75)$$

$$P(s_i = m, s_{p(i)} = n | \mathbf{w}, \mathcal{M}) = \frac{\alpha_{p(i)}(n) \beta_{p(i) \setminus i}(n) \beta_i(m) \epsilon_{i,p(i)}^{mn}}{f(\mathbf{w} | \mathcal{M})} \quad (5.76)$$

#### Updating the HMT model parameters

By tying across scale, the model  $\mathcal{M}$  is updated as follows:

#### Updating the HMT model parameters

$$\hat{\pi}(m) = P(s_1 = m | \mathbf{w}, \mathcal{M}) \quad (5.77)$$

$$P_j(m) = \frac{1}{N_j} \sum_{i \in [j]} P(s_i = m | \mathbf{w}, \mathcal{M}) \quad (5.78)$$

$$\hat{\epsilon}_{(j)}^{mn} = \frac{\frac{1}{N_j} \sum_{i \in [j]} P(s_i = m, s_{p(i)} = n | \mathbf{w}, \mathcal{M})}{P_{j-1}(n)} \quad (5.79)$$

$$\hat{\sigma}_{j;m}^2 = \frac{\sum_{i \in [j]} w_i^2 P(s_i = m | \mathbf{w}, \mathcal{M})}{\sum_{i \in [j]} P(s_i = m | \mathbf{w}, \mathcal{M})} \quad (5.80)$$

### 5.10.5 HMT denoising

HMT denoising is similar to the GMM and HMM denoising process. It estimates the clean model parameters from the noisy data and uses the two-slope shrinkage function

$$\Theta^{2L}(w_i) = \left[ P(s_i = S|\mathbf{w}, \mathcal{M}) \frac{\sigma_{j;S}^2}{\sigma_d^2 + \sigma_{j;S}^2} + P(s_i = L|\mathbf{w}, \mathcal{M}) \frac{\sigma_{j;L}^2}{\sigma_d^2 + \sigma_{j;L}^2} \right] w_i, \quad (5.81)$$

with  $j = \ell(i)$ . The clean HMT variance parameters are estimated by (5.37), as with GMMs and HMMs. The posterior state probabilities of the noisy model are directly used for the clean model, as with HMMs and following [14]. HMT denoising is expected to outperform HMM and GMM denoising if the wavelet coefficients have persistence in addition to sparseness and clustering.

## 5.11 Performance comparison of wavelet denoising algorithms

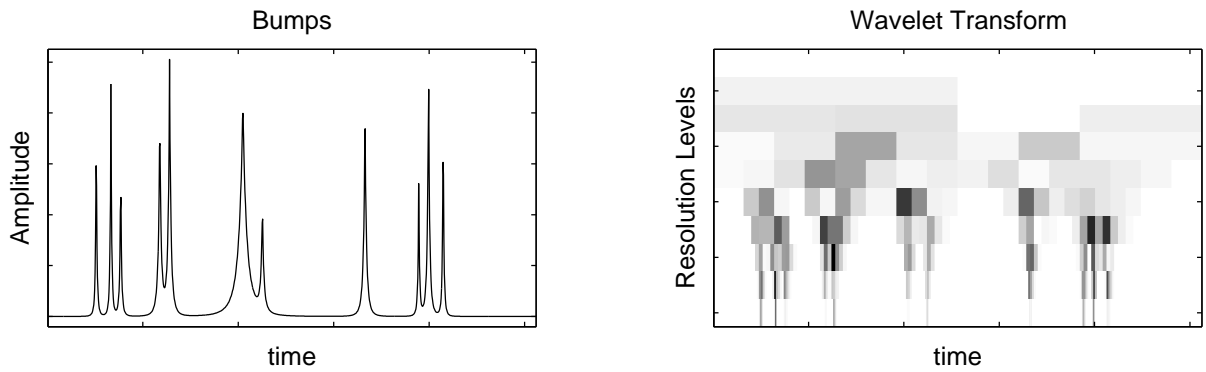
Chipman et al. [10, 11] did an experiment to compare the performance of different denoising algorithms. The same experiment was done by Crause et al. [14] and it can therefore be used as a benchmark denoising experiment. It uses the four Donoho-Johnstone standard test signals [19], namely *Bumps*, *Blocks*, *Doppler* and *HeaviSine*, which all contain elements typically found in real-world signals such as images. These signals are 1024 samples in length and are generated with Donoho and Johnstone's *WaveLab* software package [16]. White Gaussian noise is generated and added to the test signals to create noisy signals with a global signal-to-noise ratio of 17 dB<sup>6</sup>. Different algorithms are used to denoise 1000 noisy realisations of each signal. The mean-square error (3.1), is used to evaluate the denoised signals [14]. Averaging the 1000 measures results in a single measure for each of the algorithms and test signals.

This experiment, referred to as the Donoho-Johnstone benchmark experiment, is recreated here to evaluate VisuShrink, SureShrink, HybridSure and the GMM, HMM and HMT algorithms. The four test signals and their respective wavelet transforms are shown in Figures 5.14 to 5.17 and the mean-square error results are shown in Table 5.11.

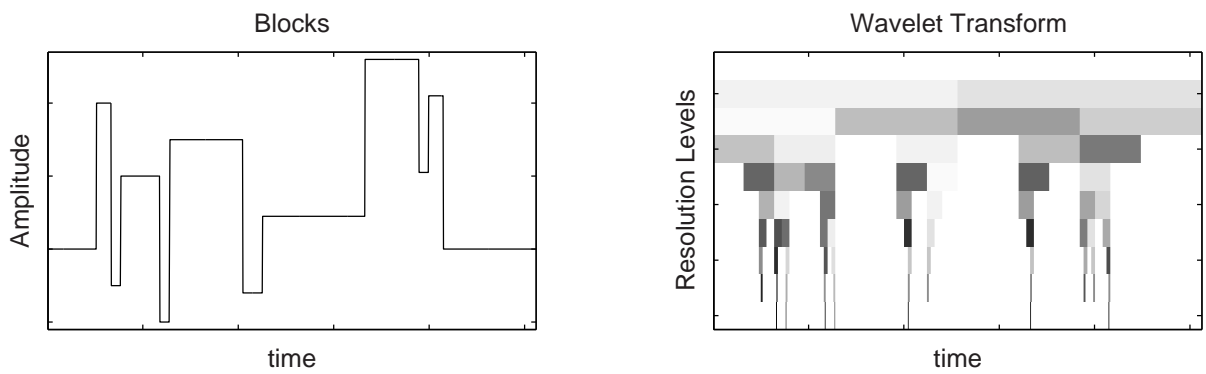
The VisuShrink and SureShrink results in Table 5.11 are in good agreement with that of Chipman et al. [10, 11] and the GMM and HMT results mirror that of Crause et al. [14].

---

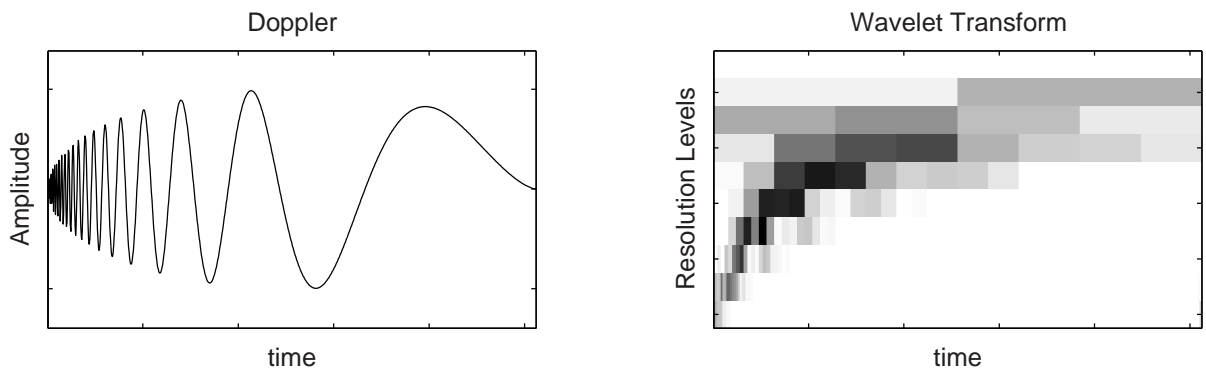
<sup>6</sup>A global signal-to-noise ratio of 17 dB is constructed by adding white Gaussian noise with power  $\sigma_n^2 = 1$  to the test signals.



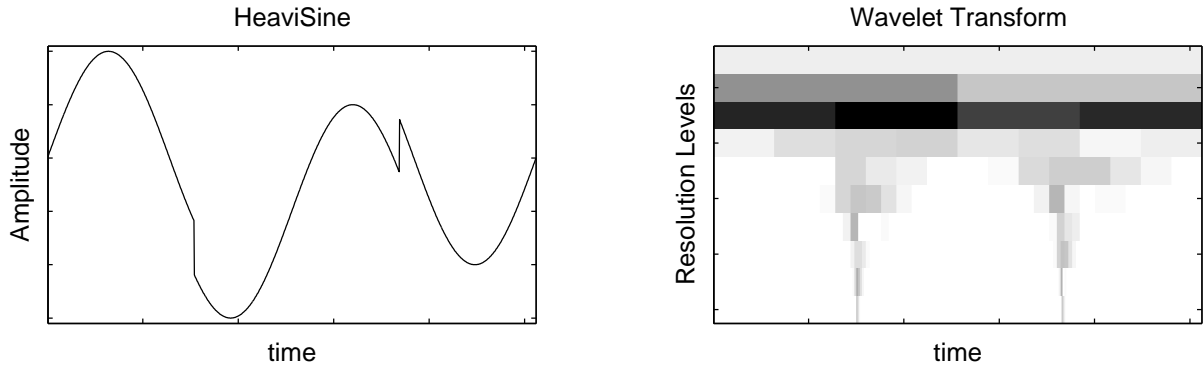
**Figure 5.14:** *Bumps and its wavelet coefficients (using the Daubechies 4 wavelet).*



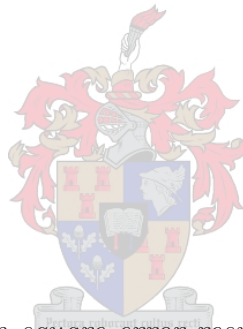
**Figure 5.15:** *Blocks and its wavelet coefficients (using a Haar wavelet).*



**Figure 5.16:** *Doppler and its wavelet coefficients (using a Daubechies 8 wavelet).*



**Figure 5.17:** *HeaviSine and its wavelet coefficients (using Daubechies 8 wavelet).*



**Table 5.1:** *The mean-square error results of the Donoho-Johnstone denoising experiment.*

Algorithm	mean-square error			
	Bumps	Blocks	Doppler	HeaviSine
Noisy Signal	1	1	1	1
VisuShrink	1.6304	0.6837	0.4873	0.1203
SureShrink	0.7348	0.5125	0.4400	0.2863
HybridSure	0.4795	0.2122	0.2339	0.0945
GMM	0.3383	0.1084	0.1780	0.0981
HMM	<b>0.2616</b>	0.1115	<b>0.1155</b>	0.0984
HMT	0.2715	<b>0.0802</b>	0.1421	<b>0.0861</b>

Because the noise is randomly generated, the Donoho-Johnstone benchmark experiment cannot be exactly recreated, which accounts for this small difference in results.

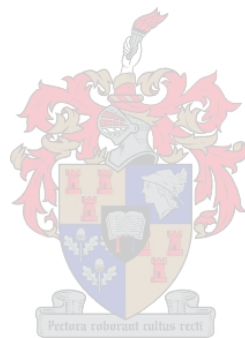
By examining the mean-square error results, shown in Table 5.11, the following observations are made:

- In most cases the algorithms have lower mean-square error values than that of the noise, implying that the algorithms successfully denoised the test signals. The only exception is VisuShrink on Bumps, which indicates that VisuShrink tends to remove too much high frequency content [11].
- The GMM, HMM and HMT algorithms outperform the classical and more heuristic Donoho and Johnstone methods. It is deduced that the explicit modelling of the non-Gaussianity is responsible for this enhancement.
- The HMT algorithm performs best on Blocks and HeaviSine. These are signals with strong persistence which can be seen in Figures 5.15 and 5.17. Because the HMT is designed to capture persistence, it performs best on such signals.
- Crause et al. [14] showed that significant mean-square error gains can be achieved by exploiting wavelet-domain dependencies via the HMT model. Table 5.11 shows that the HMM algorithm performs best on Bumps and Doppler and therefore outperforms the state-of-the art HMT mean-square error results. These novel results imply that the HMT model is not completely successful in its attempt to capture clustering and that the HMM is more successful in denoising these types of signals.

The HMM significantly improves the mean-square error results of the Doppler test signal. Figure 5.16 shows that the Doppler signal has only a single cluster of coefficients within each resolution level. The superior mean-square error results imply that the HMM successfully captures these single clusters. Figure 5.16 also shows that the vertical alignment of the Doppler coefficients between neighbouring resolution levels is not as strong as, say, that of Bumps or Blocks, indicating that the Doppler signal does not have strong persistence in the wavelet domain. These qualities of the Doppler signal explains the inferior HMT results.

The Blocks signal, on which the HMT algorithm excels, is representative of so-called “punctured smooth” signals typically found in real-world images [47]. The Doppler signal, however, is more representative of signals found in seismic, radar and sonar signals.. The HMM algorithm is therefore a better candidate than the HMT for denoising these signals.

It should also be noted that a signal-to-noise ratio of 17 dB is considered to be a low noise level. In speech enhancement experiments, the SNR typically ranges from 0 dB to 10 dB which are much higher noise levels. The statistical denoising algorithms are expected to perform poorer at high noise levels, where the statistical properties of the underlying clean signal is not as apparent as at low noise levels.



# Chapter 6

## Wavelet-based speech enhancement

### 6.1 Introduction

Wavelet-based speech enhancement is investigated, implemented and evaluated in this chapter. Section 6.2 investigates the potential of the different wavelet-based denoising algorithms for speech enhancement. This is done in Section 6.2.1 by first classifying speech into five different groups of phonemes with roughly similar statistics, namely vowels, nasals, semivowels, stops and fricatives. An experiment is then done in Section 6.2.2 on these phoneme groups in which the denoising algorithms are evaluated for their speech enhancement potential. A framework for wavelet-based speech enhancement is developed in Section 6.3 and the various parameters used throughout this research project are discussed in Section 6.3.1. In Section 6.4 the Wiener, GMM, HMM and HMT denoising algorithms, which are described in Sections 5.6 to 5.10, are implemented as speech enhancement algorithms. In Section 6.5.1 a noise floor parameter is investigated, which allows the enhanced speech to have a residual white noise artifact, which is perceptually pleasing and masks unwanted artifacts. Section 6.6 investigates the effect of using different wavelets for speech enhancement. Speech enhancement is frame-based, and the best frame size is chosen in Section 6.7. To complete the design of wavelet-based speech enhancement, the best algorithm is chosen in Section 6.8.

### 6.2 Denoising of speech segments

The Donoho-Johnstone denoising experiment [11, 14] investigates the potential of wavelet-based algorithms for denoising of signals such as images. It uses four benchmark test



signals, namely *Bumps*, *Blocks*, *Doppler* and *HeaviSine*. It was shown in Section 5.11, where the experiment was recreated, that HMT and HMM denoising perform well on the Donoho-Johnstone test signals. HMT denoising also works well on images [14, 47], because it captures the statistical properties of images in the wavelet domain. Real-world images are typically punctured smooth signals. The slowly varying localised shades are the *smooth* parts, while the less frequent abrupt changes in colour or shade form the *punctured* parts. This is reminiscent of the *Blocks* signal in the Donoho-Johnstone set, on which HMT denoising excelled. The question arises if wavelet-based denoising algorithms such as HMT and HMM denoising also work well on speech signals, which are typically not punctured smooth.

An experiment similar to the Donoho-Johnstone denoising experiment is done in Section 6.2.2, which investigates the potential of wavelet-based denoising algorithms for speech enhancement. Speech can be divided into different groups of phonemes. It is assumed that the phonemes within these groups have roughly similar statistical properties. The experiment uses five phoneme groups as test signals, namely *vowels*, *nasals*, *semivowels*, *stops* and *fricatives*.

In noisy speech, the signal-to-noise ratio of each frame varies dramatically from frame to frame, as shown in Figure 3.2. Certain phonemes, such as nasals, typically have a much lower segmental signal-to-noise ratio than phonemes such as vowels. The phonemes are therefore scaled in this experiment so that all segments have the same signal-to-noise ratio, creating an experimental setup which is similar to the Donoho-Johnstone denoising experiment. Because the test signals are short speech segments, the experiment is a simplified version of real speech enhancement. Although it is not the same as real speech enhancement, similar results are still expected.

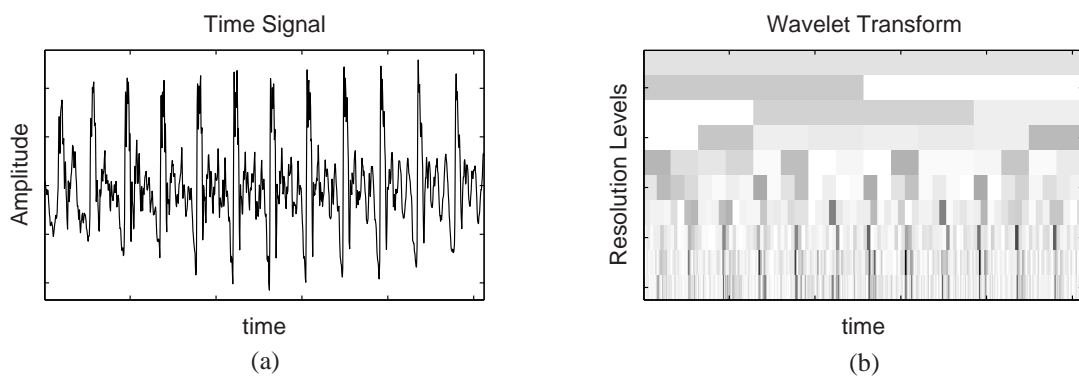
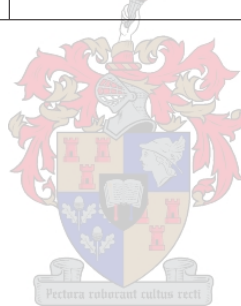
### 6.2.1 The phoneme groups

The chosen five phoneme groups, with their TIMIT phoneme labels [28], are given in Table 6.2.1 and correspond to those used in [31].

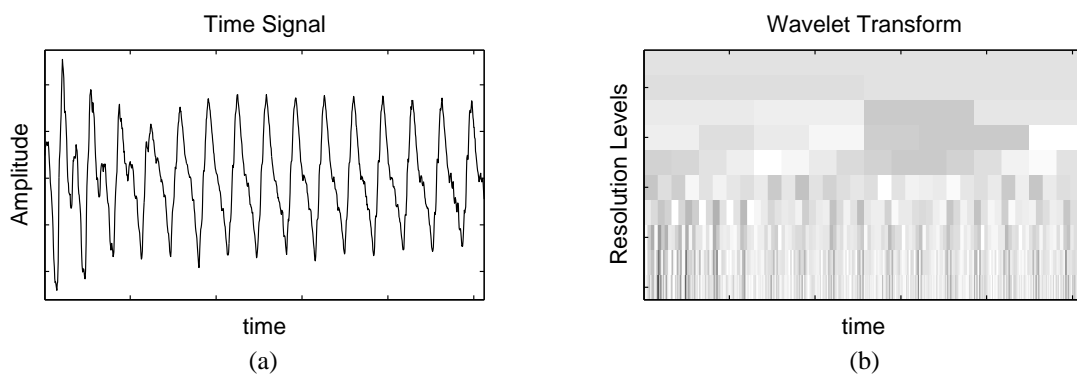
An example of each phoneme group is shown in the time domain and the wavelet domain in Figures 6.1 to 6.5. The time-domain view gives an indication of the harmonic content, the noise content and the abrupt changes of the different phoneme groups. The *time-frequency tiling* view of the wavelet domain gives an indication the statistical properties of the wavelet coefficients of the different phoneme groups.

**Table 6.1:** *The five different phoneme groups with their corresponding TIMIT phoneme labels.*

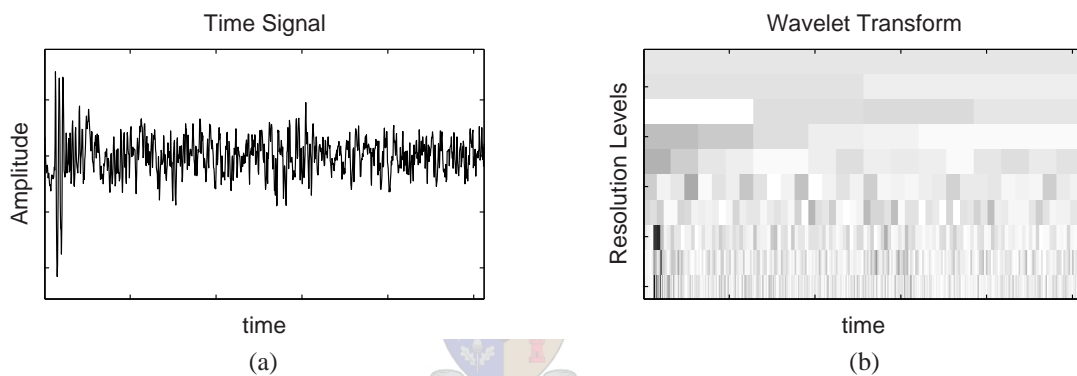
Phoneme group	TIMIT phoneme labels
Vowels	iy, ih, eh, ey, ae, aa, aw, ay, ah, ao, oy, ow, uh, uw, ux, er, ax, ix, axr, ax-h
Nasals	m, n, ng, em, en, eng, nx
Semivowels	l, r, w, y, el
Stops	b, d, g, p, t, k
Fricatives	s, sh, z, zh, f, th, v, dh, jh, ch



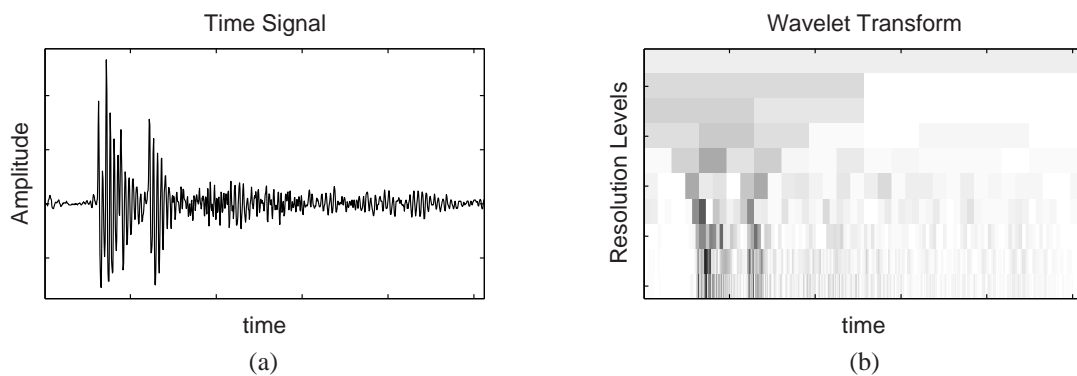
**Figure 6.1:** *(a) Vowels in the time domain have a strong harmonic content, but also high frequency components. (b) The wavelet domain shows them to have some degree of persistence and also clusters of almost equal lengths within the resolution levels.*



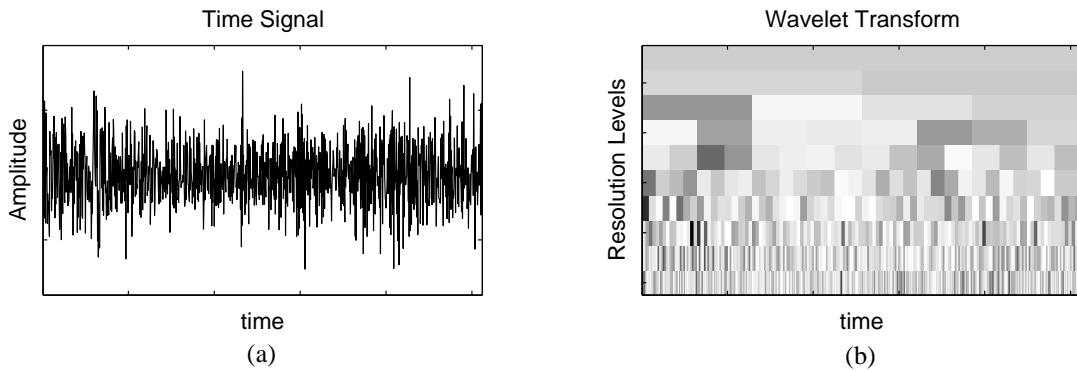
**Figure 6.2:** (a) *Nasals in the time domain have a very strong harmonic content and is almost sinusoidal.* (b) *Nasals in the wavelet domain have equal length clusters within certain resolution levels.*



**Figure 6.3:** (a) *Semivowels in the time domain have harmonic and noise components.* (b) *Semivowels in the wavelet domain have persistence hidden in the more dominant white noise coefficients.*



**Figure 6.4:** (a) *Stops in the time domain are signals with abrupt changes.* (b) *Stops in the wavelet domain have persistence.*



**Figure 6.5:** (a) *Fricatives in the time domain are almost white.* (b) *Fricatives in the wavelet domain are also very white.*

## 6.2.2 The speech-segment denoising experiment

This experiment is similar to the Donoho-Johnstone denoising experiment [11, 14] done in Section 5.11, except that 100 different segments within each phoneme group are used instead of a single Donoho-Johnstone test signal. The speech signals are taken from the TIMIT speech database [28], which contains high-quality speech recorded at  $F_S = 16$  kHz. The recordings are downsampled to have a sampling frequency of  $F_S = 8$  kHz, which is the most prevalent choice in speech enhancement research [33]. Phonemes vary in duration but a typical length is in the order of 32 ms. A data set for each phoneme group is created by extracting speech segments, which are 256 samples (32 ms) in length, from the recordings of the speakers in the TIMIT24WGN set<sup>1</sup>. The beginning, middle and end sections of each phoneme are included in the database of segments. White Gaussian noise is generated and added to each segment to create 100 noisy segments from each phoneme group, where each segment has an SNR of 10 dB. A full wavelet decomposition is done on each segment using the Discrete Meyer wavelet filters which are popular in current wavelet-based speech enhancement [8, 13].

The algorithms under investigation are those described in Sections 5.4 to 5.10 and are listed below:

- HMT - Hidden Markov Tree denoising.
- HMM - Hidden Markov Model denoising.
- GMM - Gaussian Mixture Model denoising.

---

<sup>1</sup>See Appendix B.

- WIE - Wiener denoising.
- SURE - SureShrink.
- HYB - HybridSure.
- VISU - VisuShrink.
- NSY - The un-enhanced original noisy segments.

The  $d_{MSE}$ ,  $d_{SEGSNR}$  and  $d_{IS}$  objective measures, which are widely used [14, 33] and described in Sections 3.2, 3.2.1 and 3.2.2, are used to evaluate the performance of the algorithms on the five phoneme groups. The noisy segments (NSY) are also evaluated and used as a benchmark, where the denoising algorithms are expected to outperform the noisy results.

The  $d_{MSE}$  measure views the difference between the clean and enhanced time signals as an error signal, whereas the  $d_{SEGSNR}$  measure views this difference as a noise signal and uses it to compute the signal-to-noise ratio. Both measures are computed in the time domain and yield similar results which are shown in Tables 6.2 and 6.3.

**Table 6.2:** *The mean-square error  $d_{MSE}$  evaluation of speech segments from different phoneme groups. Lower  $d_{MSE}$  values correspond to better performance.*

Phoneme group	Mean-square error $d_{MSE}$							
	HMT	HMM	GMM	WIE	SURE	HYB	VISU	NSY
Vowels	<b>0.534</b>	<b>0.534</b>	0.544	0.552	0.817	0.656	1.166	1
Nasals	0.504	<b>0.502</b>	0.517	0.542	0.865	0.728	0.896	1
Semivowels	0.480	<b>0.477</b>	0.493	0.504	0.719	0.566	1.115	1
Stops	0.732	<b>0.718</b>	0.767	0.872	0.994	0.805	2.651	1
Fricatives	0.794	<b>0.787</b>	0.807	0.820	0.994	0.871	3.392	1

The following conclusions can be made from the  $d_{MSE}$  and  $d_{SEGSNR}$  evaluation:

- All algorithms, apart from VisuShrink, outperform the original noisy signal. The poor performance of VisuShrink implies that the universal threshold is too high for speech signals. The values of most wavelet coefficients of especially fricatives, stops and vowels are below this threshold and therefore made zero. This leads to the poor performance of VisuShrink and implies that the wavelet coefficients of speech are not as sparse as that of images, for which the universal threshold is designed.

**Table 6.3:** The segmental signal-to-noise ratio  $d_{SEGSNR}$  evaluation of speech segments from different phoneme groups. Higher  $d_{SEGSNR}$  values correspond to better performance.

Phoneme group	Segmental signal-to-noise ratio $d_{SEGSNR}$							
	HMT	HMM	GMM	WIE	SURE	HYB	VISU	NSY
Vowels	12.703	<b>12.715</b>	12.578	12.364	11.384	11.778	8.507	10.532
Nasals	12.467	<b>12.492</b>	12.331	11.787	10.591	11.148	10.165	9.870
Semivowels	12.901	<b>12.935</b>	12.787	12.601	11.399	12.339	9.716	9.880
Stops	11.077	<b>11.146</b>	10.853	10.073	9.985	10.643	5.913	9.952
Fricatives	10.894	<b>10.930</b>	10.883	10.799	10.375	10.569	5.244	10.335

- The HMM algorithm performs the best on all phoneme groups, with the HMT algorithm performance only slightly inferior to that of the HMM. This is similar to the experiment done in Section 5.11, where the  $d_{MSE}$  results of the HMM outperformed that of the HMT on *Bumps* and *Doppler*, which are signals with stronger clustering (property S1) than persistence (property S2). Because the HMM outperforms the HMT, it can be deduced that persistence is not as strongly present in speech as in images.
- The GMM method does not perform as well as the HMT and HMM methods, but outperforms the Wiener method. Because the HMT, HMM and GMM model the non-Gaussianity of wavelet coefficients, it is deduced that the coefficients of speech signals do have a degree of sparsity (property P3).
- HybridSure is similar to SureShrink, apart from an extra step that checks the sparseness of the wavelet coefficients [25]. HybridSure performs better than SureShrink, which also implies that the wavelet coefficients of speech do possess some sparsity.
- The statistical methods, namely HMT, HMM, GMM and Wiener, all significantly outperform the classical techniques, namely SureShrink, HybridSure and VisuShrink. Certain phonemes, such as fricatives, contain signal energy which is very noisy and similar to white noise. The classical wavelet-based denoising techniques classify these coefficients as noise and attempt to denoise this vital part of speech signals. The statistical methods use the data itself to set the parameters of the shrinkage functions and therefore retain the noisy components of speech.

The  $d_{IS}$  measure compares the spectra of the clean and enhanced signals, as opposed to the  $d_{MSE}$  and  $d_{SEGSNR}$  measures which operate in the time domain. The results of the  $d_{IS}$  measure are shown in Table 6.4.

**Table 6.4:** *The Itakura-Saito  $d_{IS}$  evaluation of speech segments from different phoneme groups. Lower  $d_{IS}$  values correspond to better performance.*

Phoneme group	Itakura-Saito distortion $d_{IS}$							
	HMT	HMM	GMM	WIE	SURE	HYB	VISU	NSY
Vowels	0.501	<b>0.499</b>	0.524	1.391	7.539	937	1060	0.777
Nasals	<b>0.498</b>	0.580	0.552	1.956	0.590	4677	11967	0.756
Semivowels	<b>1.097</b>	1.374	1.490	25.098	33.21	1591	2191	1.537
Stops	0.027	<b>0.026</b>	<b>0.026</b>	0.033	0.037	0.029	0.956	0.038
Fricatives	0.052	<b>0.049</b>	0.052	0.058	0.067	2.485	4.267	0.082

The  $d_{IS}$  evaluation yield the following results:

- Only the HMT, HMM and GMM methods outperform the noisy signals for all phonemes.
- The performance of Wiener, SureShrink, HybridSure and VisuShrink on vowels, nasals and semivowels is very poor. The  $d_{IS}$  distortion values are typically in the order of 0-10 [55], which implies that these high  $d_{IS}$  values, such as the  $d_{IS} = 11967$  for VisuShrink on nasals, are unrealistically high. The  $d_{IS}$  values on these phonemes imply a great loss in characteristic information and can only result if the enhanced signal is totally different from the clean signal. This effect is referred to as “problem segments” and is further investigated in Section 6.5.
- While performing only slightly inferior on vowels, stops and fricatives, the HMT significantly outperforms the HMM and GMM on nasals and semivowels. The HMT is therefore the best method according to the  $d_{IS}$  measure.
- It is expected that stops have strong persistence (property S2) and that the HMT would therefore excel on them. The HMT, HMM and GMM, however, perform equally well on stops. This implies that the persistence property of stops is not as strong as would be expected. Although stops are signals with abrupt changes, they are not punctured smooth image-like signals which have strong persistence. This is confirmed by the  $d_{MSE}$  and  $d_{SEGSNR}$  results.

- Fricatives are very noisy with wavelet coefficients that are not sparse. This is verified by the fact that the HMT, HMM and GMM methods, which capture sparseness, perform very similar to the Wiener method which models the coefficients as Gaussian.

All three objective measures, shown in Tables 6.2, 6.3 and 6.4, confirm that the HMT, HMM and GMM algorithms perform the best over all phoneme groups. The Wiener method performs surprisingly well, considering its simplicity compared to the above-mentioned three methods.

It is suggested that HMT, HMM, GMM and Wiener denoising methods should be chosen as speech enhancement algorithms rather than SureShrink, HybridSure and VisuShrink. These last three algorithms are concluded to be inferior speech enhancement algorithms and are therefore not investigated any further.

### 6.3 The experimental framework

A general framework for wavelet-based speech enhancement and evaluation is discussed below. Several different aspects are pointed out in boldface; a selection must be made in each case when doing speech experiments.

Figure 1.1 suggests that two data sets are needed, namely **a speech database** containing high quality speech sentences, and **a noise database** containing realisations of various noise types. The noise may also be generated but then the experiment is not reproducible. The noisy speech is created by adding the noise to the clean sentences, which can now be assumed to contain additive noise. Depending on the power of the speech and noise signals, the noisy speech has a **global signal-to-noise ratio (SNR)**, which is usually expressed in decibels (dB).

Figure 5.1 shows the flowchart of wavelet-based speech enhancement with the analysis step being the DWT. Decomposition is done by using a particular set of **wavelet filters**, which should be chosen according to the specific class of signals that is denoised (which is *speech* in this case). The DWT also requires choosing the **number of decomposition levels**, which determines the size of the binary trees of coefficients and also the number of resolution levels.

The attenuation step of wavelet-based denoising, shown in Figure 5.2, requires choosing a **denoising algorithm** and its corresponding shrinkage function, which is used to



attenuate the noisy coefficients. The denoising algorithms can be optimised for certain applications by setting various **parameters**. The synthesis step of wavelet-based denoising is the IDWT and produces the estimated speech sentences.

The objective evaluation process of speech enhancement methods, which is shown in Figure 3.1, involves comparing the estimated speech with the clean speech by using various **distortion measures**. Subjective evaluation involves informal listening tests, which imply listening to the enhanced speech and commenting on the various denoising artifacts which the objective measures cannot highlight. In some experiments these are accompanied by formal listening tests, which involve several listeners expressing preferences for the models involved.

### 6.3.1 Experimental setup

All speech enhancement experiments in this research project use the framework discussed in Section 6.3. The chosen baseline experimental setup is discussed below:

- **Speech database:** Sentences from the TIMIT speech database [28] are used as the clean speech. Speech enhancement research is widely done on speech with a sampling frequency of  $F_S = 8$  kHz [33] and therefore the recordings are downsampled from  $F_S = 16$  kHz by discarding every second sample.
- **Noise database:** This study investigates wide-band noise reduction, therefore white Gaussian noise (WGN) is chosen from Hansen’s “Additive noise sources” [30] as the noise type used. This is a single WGN file with  $F_S = 8$  kHz which is added to every sentence in the TIMIT data set and it allows results to be reproduced, unlike generated noise.

Two sets of sentences are used in this research and are listed in Appendix B. The training set contains 24 sentences and is assumed to be large enough to determine the various model parameters. The training set corrupted with additive WGN is referred to as the TIMIT24WGN set. The test set suggested by [33] contains 192 sentences from the TIMIT core test set. The test set corrupted with additive WGN is referred to as the TIMIT192WGN set.

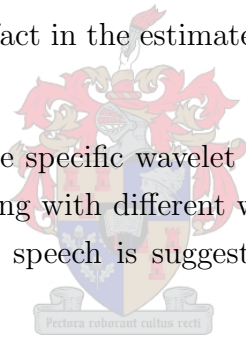
- **Global signal-to-noise ratio:** The noise and the clean speech can be scaled to produce different global signal-to-noise ratios. The silent sections before and after each sentence are included in the computation of the global SNR as was done in [32, 33]. The noise levels can be roughly categorised as follows:

- High noise levels :  $-5$  dB to  $0$  dB global SNR.
- Moderate noise levels :  $5$  dB to  $10$  dB global SNR.
- Low noise levels :  $15$  dB to  $20$  dB global SNR.

The statistical wavelet-based speech enhancement algorithms are expected to suffer at high noise levels where statistics are dominated by the noise. A moderate noise level of  $10$  dB is therefore assumed to be a good baseline global SNR.

In Section 6.2.2 it was suggested that the HMT, HMM, GMM and Wiener algorithms are superior speech enhancement algorithms compared to the classical methods. They are implemented as described in Sections 5.6 and 5.8 to 5.10. The following parameters of wavelet-based speech enhancement are investigated:

- **The algorithm parameters:** The HMT, HMM, GMM and Wiener algorithms can be modified to include a residual noise floor parameter  $\beta$ . This introduces a perceptually pleasant artifact in the estimated speech and is further investigated in Section 6.5.1.
- **The wavelet filters:** The specific wavelet used in the DWT has an effect on the estimated speech. Denoising with different wavelets are investigated in Section 6.6 where a good wavelet for speech is suggested, amongst a set of commonly used wavelets.



The Discrete Meyer wavelet can be expected to be a good wavelet for speech enhancement, because its lowpass filters are close to being ideal half-band lowpass filters in the bandpass and cut-off gradient regions and they have almost linear phase as described in Section 4.3.7. It is also used in current wavelet-based speech enhancement research [8, 13].

- **The number of decomposition levels:** The number of decomposition levels constrains the size of the analysis frames. The duration of these frames should be chosen for quasi-stationary conditions to hold and is investigated in Section 6.7.

An eight-level wavelet decomposition of speech with  $F_s = 8$  kHz results in 32-ms analysis segments (256 samples per segment) if implemented as described in Section 4.4.3. This is deemed a good choice because it yields the maximum number of training data, while the segments are still within the quasi-stationary range of speech. It is also used in current speech enhancement methods [23, 55, 58, 59, 60].

- **The different algorithms:** The HMT, HMM, GMM and Wiener denoising methods model the wavelet coefficients of speech signals with different approaches. They

are evaluated in Section 6.8 to find the superior algorithm for speech. The HMT and HMM algorithms are concluded to be the superior wavelet-based algorithms for speech enhancement, and based on the experiments done in Section 6.2.2, the HMT is chosen as the baseline algorithm.

The following evaluation process is followed:

- **The distortion measures:** The  $d_{IS}$  and  $d_{SEGSNR}$  distortion measures are chosen because they are widely used [33], and they are implemented as described in Chapter 3. Global distortion measures are computed on the
  - speech-only sections, by disregarding the non-speech segments at the beginning and end of each recording, and on the
  - phoneme groups, by first denoising the whole sentence and then using the TIMIT phoneme labels to average the distortion values of the phonemes within the particular phoneme group.
- **Informal listening tests:** It is necessary to comment on the enhanced speech of different techniques because the denoising artifacts cannot be fully represented by the distortion measures [33]. Two different denoising techniques which yield equivalent objective scores may sound completely different. Two TIMIT sentences are used for subjective evaluation and they are identified in Appendix B.3.
- **Formal listening tests:** Two formal listening tests are done in this study. The first is a comparison between the different wavelet-based speech enhancement algorithms, done in Section 6.8.2. The second is a comparison between the wavelet-based HMT algorithm and the Fourier-based Ephraim-Malah algorithm, done in Section 7.3.4. The experimental setup for these tests are discussed in Appendix B.4.

## 6.4 Wavelet-based speech enhancement experiments

The wavelet-based HMT, HMM, GMM and Wiener denoising algorithms are, as suggested in Section 6.2.2, implemented as speech enhancement algorithms.

A denoising experiment is done on the TIMIT24WGN set. White Gaussian noise is added to the clean speech to create noisy sentences with a global signal-to-noise ratio of 10 dB each. The Discrete Meyer wavelet is used in an eight-level wavelet decomposition, which leads to 32 ms non-overlapping segments. The four algorithms are used to denoise each sentence and are implemented in the framework described in Section 6.3.

### 6.4.1 Objective evaluation

The global  $d_{SEGSNR}$  evaluation on the speech-only sections for the different algorithms are shown in Table 6.5 and the following is deduced from the  $d_{SEGSNR}$  evaluation:

**Table 6.5:** Objective  $d_{SEGSNR}$  evaluation of the speech-only sections.

Algorithm	Speech-only $d_{SEGSNR}$
HMT	9.545
HMM	9.477
GMM	9.416
Wiener	9.357
Noisy	6.196

- All four algorithms clearly enhance speech compared to the un-enhanced (Noisy) version.
- The HMT clearly performs the best. This is to be expected because the HMT captures non-Gaussianity, clustering and persistence in wavelet coefficients. The HMM performs slightly worse, because it only captures non-Gaussianity and clustering. This is in contrast with the  $d_{SEGSNR}$  evaluation of the speech-segment experiment of Section 6.2.2, where the HMM slightly outperformed the HMT. This is ascribed to the artificial scaling of the speech-segments, where all segments had a SNR of 10 dB. In real speech enhancement, certain segments have a much lower SNR, which does not necessarily suit the HMM algorithm.
- The GMM performance is not as good as that of the HMM and HMT, because, although it models the non-Gaussianity of coefficients, it disregards intercoefficient dependencies.
- The Wiener method performs the least successfully, because it disregards non-Gaussian statistics and intercoefficient dependencies. However, its performance of  $d_{SEGSNR} = 9.357$  does not differ much from the HMT performance of  $d_{SEGSNR} = 9.545$ . When listening to the enhanced speech, the difference in  $d_{SEGSNR}$  of 0.2 is perceptually barely detectable.
- Not shown in Table 6.5 are the evaluation results of the algorithms with no training and initial conditions as described in Sections 5.8.4, 5.9.4 and 5.10.4. In this case the HMT, HMM and GMM methods all yield a distortion value of  $d_{SEGSNR} = 9.190$ . In

all three cases, the algorithm performance improves with fully trained models. This verifies that the model parameters are correct and more accurate if fully trained.

The  $d_{IS}$  evaluation on the speech-only sections for the five different phoneme groups are shown in Table 6.6.

**Table 6.6:** *Objective  $d_{IS}$  evaluation of the speech-only sections and of the different phoneme groups. Lower  $d_{IS}$  values correspond to better performance.*

Algorithm	$d_{IS}$					
	Speech	Vowels	Nasals	Semivowels	Fricatives	Stops
HMT	1.440	0.547	7.568	5.832	1.676	3.190
HMM	1.274	0.581	9.358	3.565	1.348	3.798
GMM	1.035	<b>0.517</b>	<b>2.336</b>	2.138	1.194	2.143
Wiener	13.115	1.268	1024	64	4.547	68
Noisy	<b>1.031</b>	0.660	2.406	<b>1.470</b>	<b>0.911</b>	<b>1.222</b>

The following observations are made from the  $d_{IS}$  evaluation:

- None of the four algorithms show an improvement over the un-enhanced signal on the speech-only sections.
- On a phoneme level, only the HMT, HMM and GMM methods show an improvement, and only on vowels. Nasals and semivowels seem to be extremely distortion-prone, which was also experimentally found in Section 6.2.2.
- Not shown in Table 6.6 is the evaluation of the HMT, HMM and GMM methods with no training, which yields  $d_{IS} = 0.493$  on the speech-only sections. This is a definite enhancement compared to the noisy  $d_{IS} = 1.031$ . According to the  $d_{IS}$  measure, full training leads to speech distortion, whereas the  $d_{SEGSNR}$  measure from Table 6.5 indicates enhancement. This contradiction is further investigated in Section 6.5.

## 6.4.2 Subjective evaluation

The  $d_{SEGSNR}$  measure shows that the HMT algorithm performs slightly better than the other algorithms, while the  $d_{IS}$  measure, in contrast, shows results with significant differences between the various algorithms. By listening to the enhanced speech, the differences between the algorithms can be evaluated subjectively.

The two sentences for subjective listening tests<sup>2</sup> are enhanced under the same conditions as for the objective evaluation. The wavelet-based Wiener method, and the HMT, HMM and GMM algorithms with both full and no training are investigated.

The following observations are made from informal listening tests:

- There is perceptually no difference between the enhanced speech of the fully trained HMT, HMM and GMM methods. There is, however, a slight difference, which is barely detectable, between the Wiener method and the other three algorithms. This correlates with the  $d_{SEGSNR}$  results, which also show similar performance for the different algorithms, with Wiener being slightly inferior. It also shows that the dissimilar  $d_{IS}$  results are not a true reflection of the similar performance of the different algorithms.
- There is also no perceptual difference between the enhanced speech of the HMT, HMM and GMM algorithms implemented with no training. This is to be expected, because these algorithms use the same initial conditions and are expected to produce enhanced speech that is almost identical.
- The enhanced speech of the HMT, HMM and GMM algorithms with no training has a residual noise artifact which is perceptually noisier than that of the fully trained models. This agrees with the  $d_{SEGSNR}$  results, which yielded a distortion of  $d_{SEGSNR} = 9.190$  for no training and  $d_{SEGSNR} = 9.545$  for the fully trained HMT model. The subjective evaluation suggests that fully trained models outperform those with no training. This is again in contrast with the  $d_{IS}$  evaluation which yielded superior results for no training.
- The enhanced speech of the HMT, HMM, GMM and Wiener has an annoying residual artifact. It can be described as a stuttering, scratchy and uneven artifact and is referred to as the *wavelet-based residual artifact*. The multiresolution representation of the wavelet coefficients and the particular wavelet in use are responsible for its unique sound.

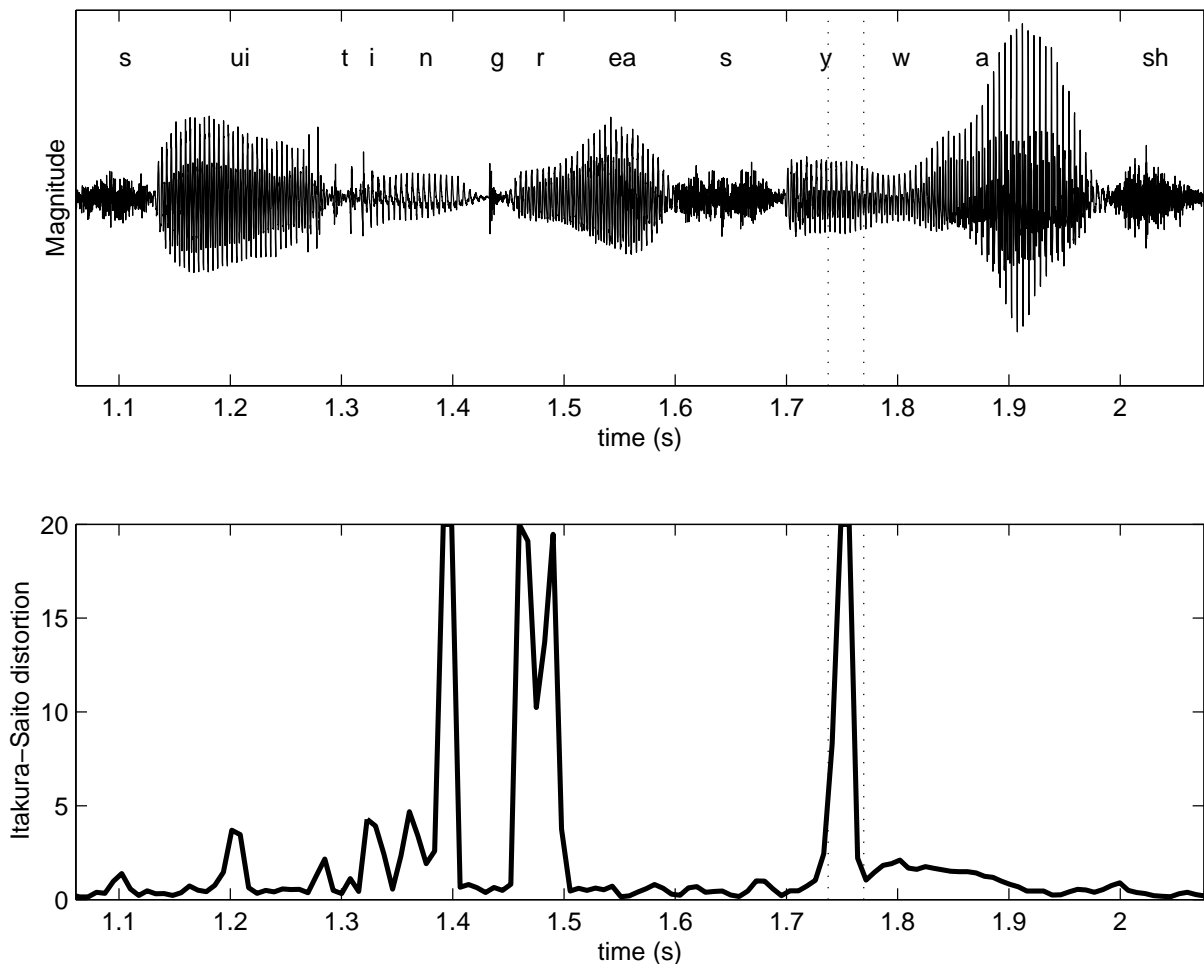
From the objective and subjective evaluation it is clear that the  $d_{IS}$  distortion values do not agree with the  $d_{SEGSNR}$  distortion values and subjective evaluation. A closer look at why the  $d_{IS}$  performance is poorer with full training is necessary and this is discussed in Section 6.5.

---

<sup>2</sup>See Appendix B.3.

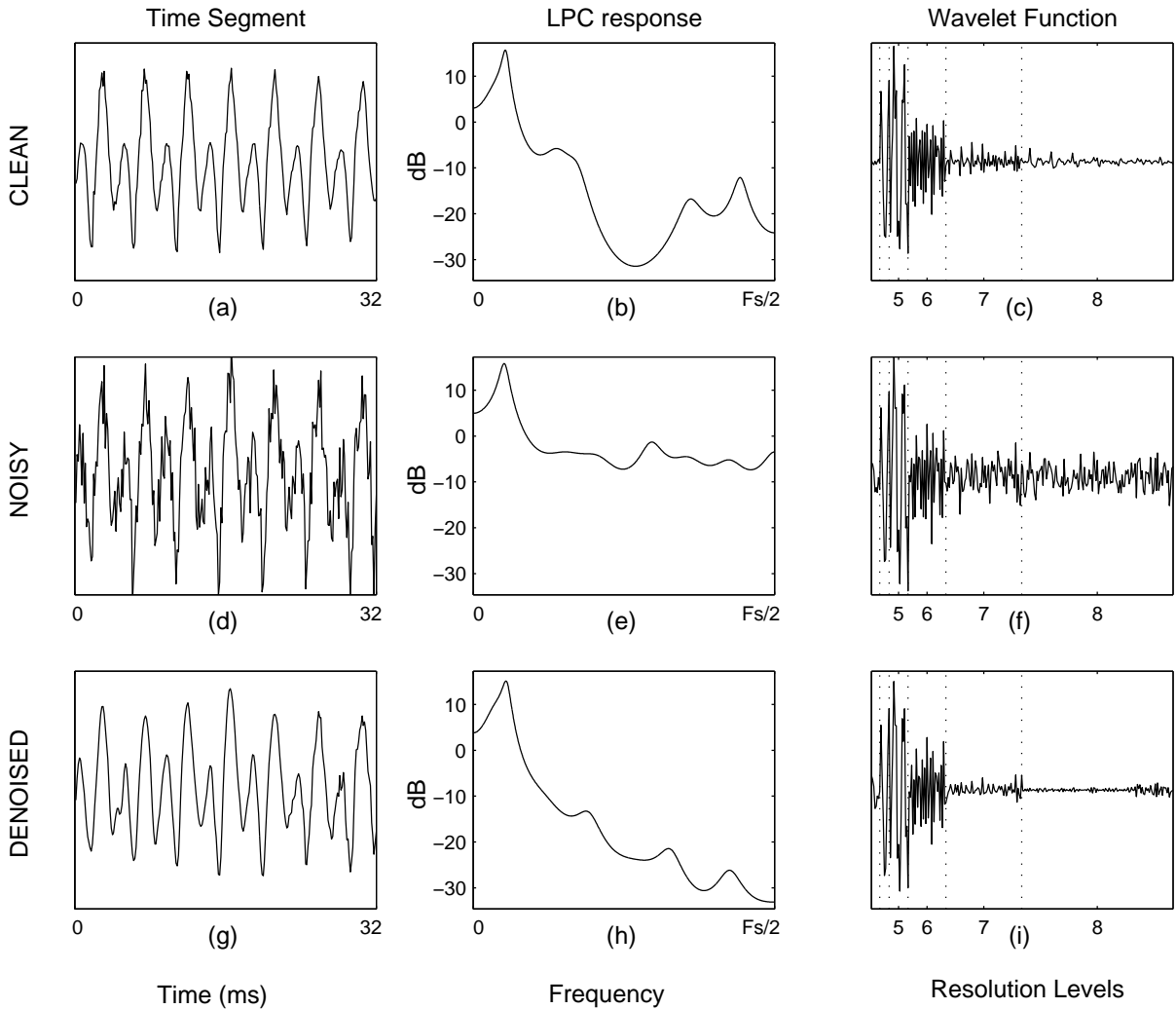
## 6.5 The $d_{IS}$ problem segments

Figure 6.6 shows an example of how the  $d_{IS}$  distortion values for enhanced speech vary over time. The sentence was corrupted with additive white Gaussian noise and had a global SNR of 10 dB before fully trained HMT enhancement. Most phonemes have low  $d_{IS}$  values, whereas certain segments are considered to be *problem segments* with extremely high  $d_{IS}$  values. This is verified in Table 6.6, where nasals and semivowels seem to be more sensitive to full training.



**Figure 6.6:** (a) A part of a TIMIT sentence, corrupted with 10 dB white Gaussian noise and enhanced with the fully trained HMT method. The speaker is *timit/train/dr1/fcjf0* and the sentence is *sa1.wav* (*She had your dark suit in greasy wash water all year*). (b) The Itakura-Saito distortion values show problem segments when evaluating enhanced speech.

Figure 6.7 shows an example of the time signals, the Linear Prediction Coefficient (LPC) spectra<sup>3</sup> and the wavelet functions of a problem segment. The clean, noisy and enhanced signals are shown. The segment is taken from the example in Figure 6.6 and is demarcated by the vertical dotted lines.



**Figure 6.7:** *The time segments, LPC responses and wavelet functions of a clean, noisy and enhanced problem segment.*

The clean, noisy and denoised time waveforms are shown in Figure 6.7(a),(d) and (g) respectively. Inspection of these three waveforms verifies that the denoised segment does not differ that much from the original, as confirmed by the  $d_{SEGSNR}$  measure.

---

<sup>3</sup>Linear prediction is a popular analysis method for speech [15]. It fits a low-order all-pole filter to a speech frame, based on its autocorrelation. The transfer function of this filter, known as the LPC spectrum, is a smoothed version of the power spectrum of the speech frame, and typically characterises the vocal tract configuration of the speaker.



The second column shows the LPC spectra of the corresponding segments. The clean segment, Figure 6.7(b), clearly shows formant activity in the higher frequency regions. The noisy spectrum, Figure 6.7(e), indicates that the additive noise dampens these spectral peaks and valleys such that the formants are barely detectable. The denoised spectrum, Figure 6.7(h), shows how the higher frequencies are almost completely cut out. The difference between the clean and denoised LPC spectra in the high frequency regions gives rise to the high level of  $d_{IS}$  distortion.

The third column shows the wavelet functions of the clean, noisy and denoised segments. A closer look at the highest resolution level (to the right of the last dotted line) is necessary. The clean wavelet function, Figure 6.7(c), shows signal activity in the highest resolution level. The noisy wavelet function, Figure 6.7(f), clearly shows that the noise energy in the highest resolution level is dominant over the signal energy. The denoised wavelet function, Figure 6.7(i), shows that the highest resolution level is made almost zero. The characteristic signal coefficients shown in Figure 6.7(c) are eliminated.

The problem segments are now identified as speech segments which have formant frequencies (characteristic signal information) of low energy inside the higher resolution levels. When these resolution levels contain coefficients which are meaningful but have small values, they are referred to as *problem resolution levels*.

The HMT is trained on the noisy coefficients and models the distribution of the noisy coefficients of each resolution level  $j$  with a small and a large variance parameter, namely  $\sigma_{j;S;y}^2$  and  $\sigma_{j;L;y}^2$ . The noise, which has an estimated Gaussian distribution of  $\hat{\sigma}_{j;d}^2$ , smothers the relatively small signal coefficients inside these problem resolution levels and is completely dominant as shown in Figure 6.7(f). This is in contrast with the HMT model definition, which assumes that large coefficients represent signal energy and small coefficients represent noise. Because the noise overpowers the signal coefficients, the observed noisy coefficients have a nearly Gaussian distribution with a variance almost equal to the noise variance. The HMT tries to model this single Gaussian distribution (variance  $\approx \hat{\sigma}_{j;d}^2$ ) using two Gaussian components (variances  $\sigma_{j;S;y}^2$  and  $\sigma_{j;L;y}^2$ ), which end up having almost equal variance parameters which are also almost equal to the noise variance, therefore  $\sigma_{j;S;y}^2 \approx \sigma_{j;L;y}^2 \approx \hat{\sigma}_{j;d}^2$ .

The clean HMT model has variance parameters  $\sigma_{j;S}^2$  and  $\sigma_{j;L}^2$  and they are estimated from the noisy HMT model as given in (5.37), as

$$\begin{aligned}\sigma_{j;S}^2 &= \max(\sigma_{j;S;y}^2 - \hat{\sigma}_{j;d}^2, 0) \\ \sigma_{j;L}^2 &= \max(\sigma_{j;L;y}^2 - \hat{\sigma}_{j;d}^2, 0) .\end{aligned}\tag{6.1}$$

The variance parameters of the clean model will therefore approximate zero, with  $\sigma_{j;S}^2 \approx 0$  and  $\sigma_{j;L}^2 \approx 0$ . It is important to notice that the variance parameters  $\sigma_{j;S}^2$  and  $\sigma_{j;L}^2$  are level dependent and they represent the distribution of all wavelet coefficients within resolution level  $j$ .

From the weighted Wiener-based shrinkage rule (5.81), given as

$$\Theta^{2L}(w_i) = \left[ P(s_i = S | \mathbf{w}, \mathcal{M}) \frac{\sigma_{j;S}^2}{\hat{\sigma}_{j;d}^2 + \sigma_{j;S}^2} + P(s_i = L | \mathbf{w}, \mathcal{M}) \frac{\sigma_{j;L}^2}{\hat{\sigma}_{j;d}^2 + \sigma_{j;L}^2} \right] w_i, \quad (6.2)$$

all estimated coefficients  $\hat{\theta}_i$  within the problem resolution level will approximate zero, therefore

$$\Theta^{2L}(w_i | w_i \in \text{problem resolution level}) \approx 0. \quad (6.3)$$

The problem resolution levels are usually the higher levels, which also contain most of the wavelet coefficients. If, for example, the highest resolution level is a problem level, then half of the wavelet coefficients within the segment will be shrunk to zero and characteristic signal information will be completely cut out. This characteristic signal information is, however, of very low magnitude and barely audible. The extremely high  $d_{IS}$  distortion values are therefore not representative of the small loss in speech quality, for which moderate  $d_{IS}$  distortion values would be expected.

It should be noted that STSA techniques, which denoise in the Fourier domain, analyse the signal with far more frequency bins. For the example shown in Figure 6.7, STSA would use 64 frequency bins just for the highest resolution level (which is only one bin in wavelet-based methods). It is therefore not disastrous to zero a single bin in STSA speech enhancement, because this will only shrink one coefficient to zero. The high  $d_{IS}$  values of problem segments are therefore more of a problem in wavelet-based speech enhancement because the higher resolution levels of the DWT correspond to very wide frequency bands.

### 6.5.1 The spectral floor parameter $\beta$

Berouti et al. [6] introduced a *spectral floor* in power spectral subtraction speech enhancement [42], which masks the “musical” residual noise artifact [23, 55]. It implies that the estimated variance can never be lower than a specified threshold value. Therefore, the use of a spectral floor overestimates the spectral variance [23].

Using a noise floor in wavelet-based speech enhancement might also prove useful. The purpose of denoising is to eliminate noise, whereas a noise floor, by contrast, reinserts some

residual noise into the enhanced speech. The aim of this noise is to mask the annoying wavelet-based residual artifact while remaining barely audible itself. This involves a compromise in selecting the value of the noise floor.

One way to implement a noise floor is to introduce a floor parameter  $\beta$  into (6.1) which estimates the variance parameters of the clean HMT model as follows:

$$\begin{aligned}\sigma_{j;S}^2 &= \max(\sigma_{j;S;y}^2 - \hat{\sigma}_{j;d}^2, \beta) \\ \sigma_{j;L}^2 &= \max(\sigma_{j;L;y}^2 - \hat{\sigma}_{j;d}^2, \beta) .\end{aligned}\tag{6.4}$$

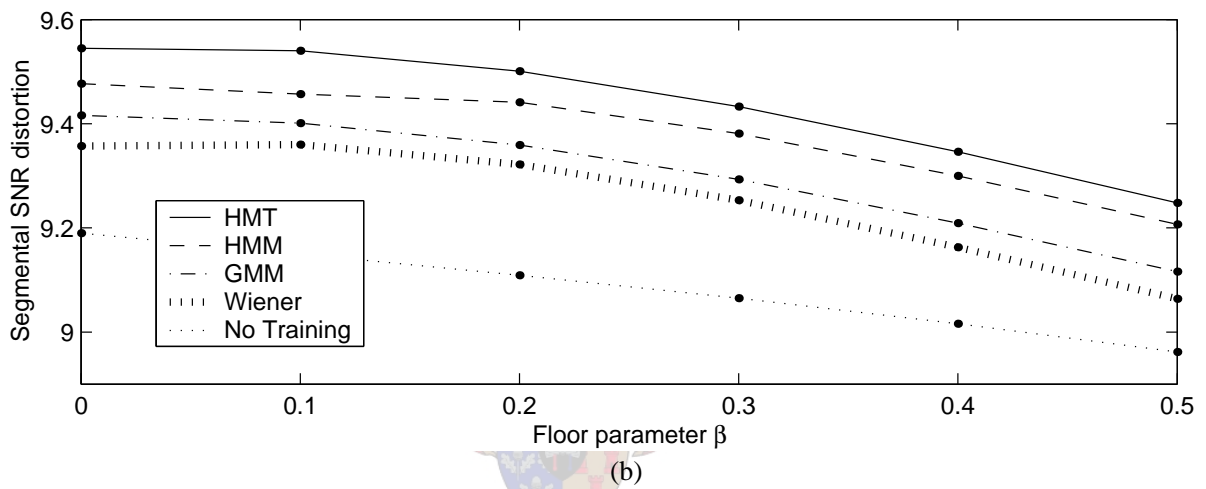
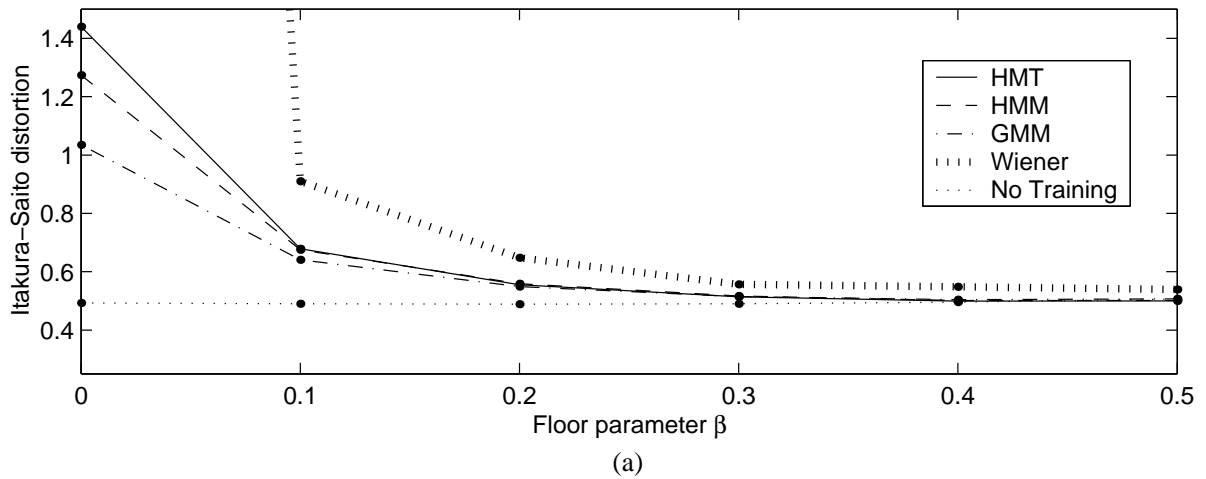
In Section 6.5 it was shown that the variance parameters of the noisy HMT model for problem resolution levels have values  $\sigma_{j;S;y}^2 \approx \sigma_{j;L;y}^2 \approx \hat{\sigma}_{j;d}^2$ . By using (6.4) with  $\beta > 0$ , the shrinkage function  $\Theta^{2L}(w_i)$  of problem segments can never be zero. By looking at (6.2), coefficients from problem resolution levels are not shrunk as much as when using no noise floor ( $\beta = 0$ ) and therefore the characteristic signal coefficients with small values and their surrounding noise coefficients are kept.

Using a noise floor in wavelet-based speech enhancement is a smoothing process. It proves to be perceptually appealing and also produces satisfactory  $d_{IS}$  distortion values.

### 6.5.2 Objective evaluation of the floor parameter

An experiment is done which objectively investigates the effect of using a floor parameter  $\beta$ . Introducing a noise floor is expected to improve the  $d_{IS}$  performance. However, it also increases the residual noise power in the enhanced signal and therefore decreases the segmental signal-to-noise ratio  $d_{SEGSNR}$ . The goal of this experiment is to find the optimum value for  $\beta$  which produces satisfactory performance on both the  $d_{IS}$  and  $d_{SEGSNR}$  distortion measures.

The same experimental setup of Section 6.4 is used here, except that different values of parameter  $\beta$  are investigated. The HMT, HMM, GMM and Wiener algorithms are used to denoise the TIMIT24WGN set with a global SNR of 10 dB. The algorithms are implemented with floor parameter values of  $\beta = \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ . The performance of the HMT, HMM and GMM algorithms with no training and initial conditions as described in Sections 5.8.4, 5.9.4 and 5.10.4 are also investigated for different values of  $\beta$ . Because initialisation is the same for all three algorithms, they produce almost the same distortion measure values if they are not trained. The  $d_{IS}$  and  $d_{SEGSNR}$  distortion values of the speech-only sections are shown in Figure 6.8.



**Figure 6.8:** The global distortion measures of the different algorithms for different values of  $\beta$ . (a) The  $d_{IS}$  results. (b) The  $d_{SEGSNR}$  results.

The  $d_{IS}$  evaluation in Figure 6.8(a) shows:

- Setting  $\beta = 0$  is equivalent to the results shown in Table 6.6, which shows poor  $d_{IS}$  performance with full training.
- As  $\beta$  increases, the  $d_{IS}$  performance drastically improves. With  $\beta > 0.2$  the  $d_{IS}$  of all four algorithms are almost equal to the favourable  $d_{IS}$  distortion of no training.
- With  $\beta > 0.2$  the  $d_{IS}$  measures of the HMT, HMM and GMM algorithms are nearly equal.
- The Wiener performance with  $\beta > 0.2$  is only slightly inferior to that of the other three algorithms.
- With  $\beta > 0.2$  the  $d_{IS}$  values of all four algorithms converges to a local minimum. The value of  $\beta$  should be chosen to lie within this region.

- Distortion values with  $\beta > 0.5$  are not shown in Figure 6.8(a) but become less desirable, eventually reaching  $d_{IS} = 1.031$  at  $\beta = 1$  (cf. Table 6.6), which is equivalent to no denoising.

The  $d_{SEGSNR}$  evaluation, shown in Figure 6.8(b), highlights the following:

- As  $\beta$  increases, the  $d_{SEGSNR}$  performance of all algorithms drop. By using no noise floor and therefore setting  $\beta = 0$ , the results shown in Table 6.5 are obtained and it is shown here to produce the best  $d_{SEGSNR}$  performance.
- At  $\beta = 0.2$  the gradient of the  $d_{SEGSNR}$  curves becomes steeper, which implies little difference in  $d_{SEGSNR}$  performance with the floor parameter within the range  $0 < \beta < 0.2$ . Setting  $\beta > 0.2$  noticeably decreases the  $d_{SEGSNR}$  performance.
- The performance ranking of the various algorithms is the same for all values of  $\beta$ , with the HMT being the superior algorithm, the HMM performing slightly worse, the GMM performing even less satisfactory and the Wiener method being the least desirable. The difference in the performance of the four algorithms is relatively small, however.
- The fully trained HMT, HMM and GMM algorithms noticeably improve the  $d_{SEGSNR}$  performance when compared to a model without training. With no training, the non-Gaussian distribution is not correctly estimated by the HMT, HMM and GMM initial parameters. Also, the state transition probabilities of the HMT and HMM models are not utilised.

From the  $d_{IS}$  and  $d_{SEGSNR}$  evaluation of enhancing speech with a global signal-to-noise ratio of 10 dB, the best floor parameter is chosen as  $\beta = 0.2$ . This choice results in satisfactory and stable  $d_{IS}$  values with only a slight drop in  $d_{SEGSNR}$  values. The question arises whether  $\beta = 0.2$  is also a good choice at different global signal-to-noise ratios.

An experiment is done which is similar to the previous experiment, where the effect of the floor parameter is investigated. The HMT method is used to denoise the TIMIT24WGN set at global signal-to-noise ratios of 0 dB and 20 dB. The  $d_{IS}$  and  $d_{SEGSNR}$  results are shown in Table 6.7 and 6.8.

The  $d_{IS}$  and  $d_{SEGSNR}$  evaluation at 0 dB and 20 dB also shows  $\beta = 0.2$  to be a good choice for the floor parameter.

**Table 6.7:** *The Itakura-Saito  $d_{IS}$  evaluation of HMT speech enhancement using different noise floors at different SNRs.*

SNR	Speech-only $d_{IS}$						$d_{IS}$ NSY
	$\beta = 0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	
0 dB	9.200	1.305	1.153	1.143	1.189	1.251	2.268
20 dB	0.242	0.208	0.185	0.174	0.168	0.165	0.319

**Table 6.8:** *The segmental signal-to-noise ratio  $d_{SEGSNR}$  evaluation of HMT speech enhancement using different noise floors at different SNRs*

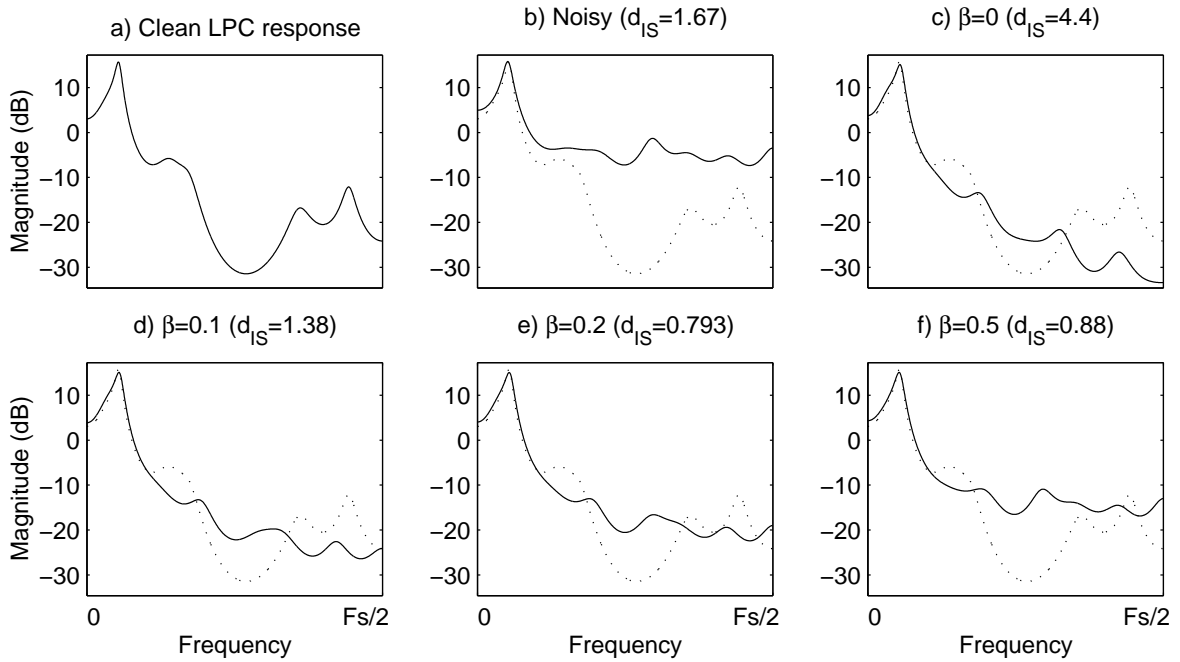
SNR	Speech-only $d_{SEGSNR}$						$d_{SEGSNR}$ NSY
	$\beta = 0$	$\beta = 0.1$	$\beta = 0.2$	$\beta = 0.3$	$\beta = 0.4$	$\beta = 0.5$	
0 dB	2.612	2.543	2.359	2.106	1.830	1.556	-2.279
20 dB	17.490	17.494	17.491	17.480	17.464	17.444	16.086

### 6.5.3 LPC evaluation of the floor parameter

The effect of the floor parameter  $\beta$  on the LPC spectrum with a fully trained HMT model is evaluated here. Figure 6.9 shows the LPC spectra associated with different noise floors for the same problem segment as shown in Figure 6.7. The top row is taken from Figure 6.7 for comparison purposes. The clean spectrum is superimposed onto the noisy and enhanced spectra in Figure 6.9(b)-(f). A closer fit with the clean spectrum results in better enhancement and thus lower  $d_{IS}$  values.

The clean LPC spectrum in Figure 6.9(a) shows significant peaks (formant frequencies) and valleys. Especially note the two peaks in the higher frequency region. The noisy LPC spectrum in Figure 6.9(b) shows the effect of additive broadband noise. The spectrum becomes flatter as the noise dampens the peaks and valleys.

Figure 6.9(c) shows the LPC spectrum with  $\beta = 0$ . There is a big difference between the enhanced and clean spectra in the high frequency region and the  $d_{IS}$  penalises such a mismatch in formant location. The magnitude of the enhanced spectra in this region is in the order of  $-30$  dB, which implies very little signal energy. The  $d_{IS}$  is thus especially harsh on an enhanced spectrum of very low magnitude, and this results in the high distortion of  $d_{IS} = 4.4$ . Note, however, that although the magnitude is small, the peaks and valleys are more noticeable than in any of the other enhanced spectra. This observation



**Figure 6.9:** (a) The clean LPC response. (b) The noisy LPC response. (c)-(f) The denoised LPC response, with  $\beta$  increasing from 0 to 0.5.

implies that the  $d_{IS}$  measure is not necessarily the most accurate way to evaluate speech enhancement.

Figure 6.9(d)-(f) show that as  $\beta$  increases from  $\beta = 0.1$  to  $\beta = 0.5$ , the magnitude of the higher frequencies increases (which is good), while the peaks and valleys gets dampened (which is bad). From an LPC viewpoint, setting  $\beta = 0.2$  is a good trade-off between these two factors, where the matching of the enhanced and clean spectra seems to be best. The experiment done in Section 6.5.2, which investigate the objective measures, verifies this.

#### 6.5.4 Subjective evaluation of the floor parameter

By subjectively listening to the enhanced speech, the effect of different noise floors is evaluated. The aim of the investigation is to subjectively find the noise floor which masks the annoying residual artifact without being too annoying itself.

The two sentences for subjective listening tests<sup>4</sup> are enhanced under the same conditions as the objective evaluation in Section 6.5.2, based on the HMT algorithm. Different values for the floor parameter are investigated and chosen from the set  $\beta \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5\}$ .

<sup>4</sup>See Appendix B.3.

From informal listening tests the following observations are made:

- Setting  $\beta = 0$  results in speech enhancement with no noise floor. The wavelet-based residual artifact is very scratchy and stuttering and therefore perceptually annoying.
- By setting  $\beta = 0.1$ , the residual artifact is still very annoying and the noise floor barely detectable.
- Setting  $\beta = 0.2$  introduces a slight noise floor in the enhanced speech. The noise floor is slightly audible but it masks the annoying residual artifact. The new artifact sounds like white noise and is perceptually pleasing.
- Setting  $\beta \geq 0.3$  results in a noise floor which is audible and even disturbing. This defeats the purpose of denoising.

From the informal listening tests, a noise floor of  $\beta = 0.2$  is preferred over no noise floor. This confirms the objective and LPC evaluation results of Sections 6.5.2 and 6.5.3.

## 6.6 Choosing a good wavelet

The choice of the wavelet filters can have a large effect on speech enhancement, since it determines the decomposition and reconstruction filter banks used. Experiments are done with a variety of discrete wavelet filters to answer the question: “Which wavelet produces the best speech enhancement?”

Different wavelet families are investigated, all of which are described in Section 4.3.7 and given below:

- **The Daubechies wavelet family**

The Daubechies wavelet family consists of several wavelets (or sets of wavelet filters) which are distinguished by their Herrmann order. Daubechies wavelets have maximally flat filters of equal lengths, but do not have linear phase. Daubechies wavelets are used by Seok et al. [50] in their wavelet-based speech enhancement approach, which is similar to the approach of this research project.

- **The Symlet wavelet family**

The Symlet wavelet family is similar to the Daubechies family and also consists of several wavelets of differing Herrmann order. The filters are maximally flat, of equal lengths and have almost linear phase. Symlet wavelets are expected to outperform



Daubechies wavelets when denoising with the HMT algorithm, because linear phase preserves the location of signal details and therefore enhances persistence. Symlets are suggested by [25] for image denoising and are used by [35] in speech enhancement.

- **The Biorthogonal wavelet families**

There are several Biorthogonal wavelet families which are determined in spectral factorisation filter design by the distribution of the remaining zeros. The Biorthogonal wavelets used in this study have linear phase which enhances persistence and are therefore good for HMT denoising. The Biorthogonal 1, Biorthogonal 2 and Biorthogonal 3 families are investigated because they have short highpass and long lowpass reconstruction filters, which was shown to be favourable for image compression [52]. There are several wavelets within each Biorthogonal family, as determined by their Herrmann order.

- **The Haar wavelet**

The Haar wavelet is a single wavelet which is the Daubechies, Symlet and Biorthogonal 1 wavelet with a Herrmann order of  $m = 1$ . It is the most basic wavelet because its lowpass filters only have a single zero at  $z = -1$ . The Haar wavelet is expected to be inferior to the other wavelets because the filters are far from being ideal halfband filters, as shown in Figure 4.12.

- **The Discrete Meyer wavelet**

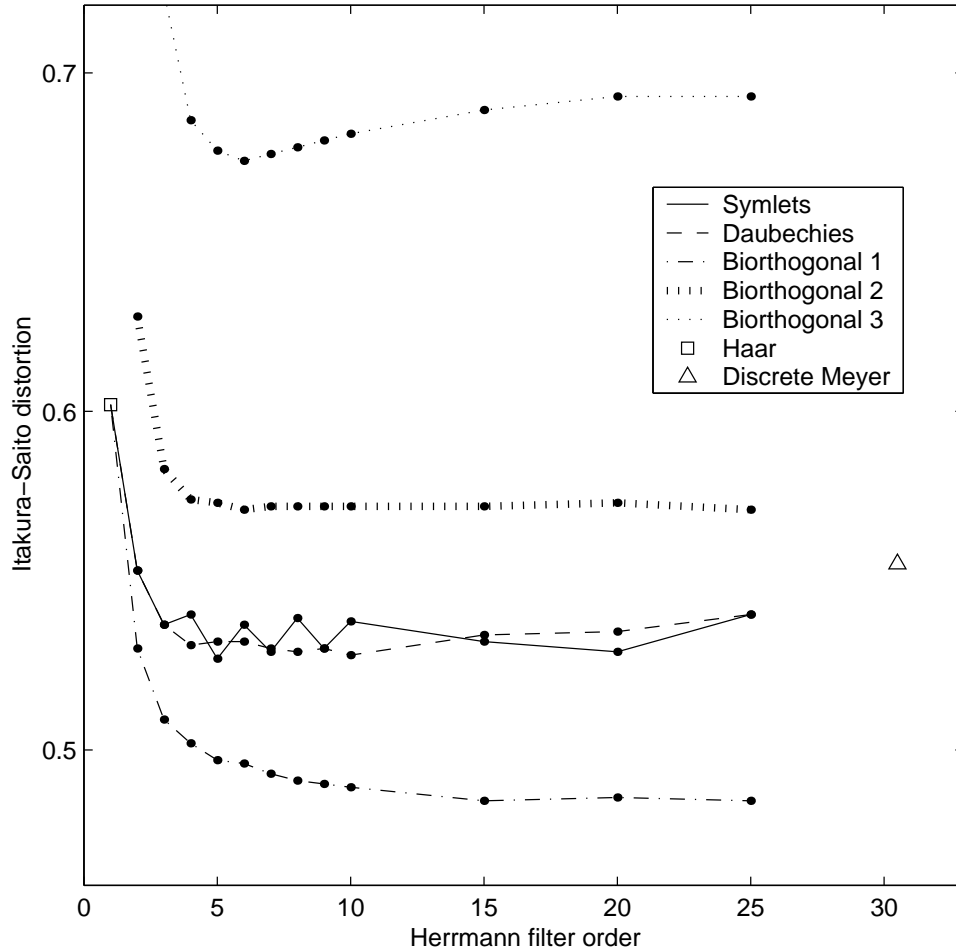
The Discrete Meyer wavelet is a single wavelet which is not designed by spectral factorisation but can be viewed, from its filter lengths, to have an equivalent Herrmann order of  $m \approx 31$ . The Discrete Meyer wavelet filters are close to being ideal in the bandpass and cut-off gradient region (see Figure 4.12). The Discrete Meyer wavelet is used by [8, 13] in wavelet-based speech enhancement.

An experiment is done which investigates the performance of different wavelets. The same experimental setup as described in Section 6.3.1 is used, where the noisy TIMIT24WGN set with a global SNR of 10 dB is enhanced with the HMT algorithm with a floor parameter of  $\beta = 0.2$ . This experiment is done for each of the investigated wavelets and results in a single distortion value for both the global  $d_{IS}$  and  $d_{SEGSNR}$  measures. The Daubechies, Symlet, Biorthogonal 1, Biorthogonal 2 and Biorthogonal 3 wavelet families are investigated by choosing the Herrmann order from the set  $m \in \{2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20, 25\}$ . This produces twelve wavelets per family, from which a distortion curve is generated for each of the five families. Because there is only one Haar and Discrete Meyer wavelet, enhancing speech with them produces only a single  $d_{IS}$  and  $d_{SEGSNR}$  distortion value.

The  $d_{IS}$  and  $d_{SEGSNR}$  distortion measures are implemented as described in Chapter 3 and

the results are shown in Figures 6.10 and 6.11.

### 6.6.1 The Itakura-Saito distortion ( $d_{IS}$ ) evaluation



**Figure 6.10:** The Itakura-Saito  $d_{IS}$  evaluation of different wavelet families with different filter orders. The Itakura-Saito distortion of the noisy speech is  $d_{IS} = 1.031$ .

From Figure 6.10, which shows the  $d_{IS}$  distortion, the following observations are made:

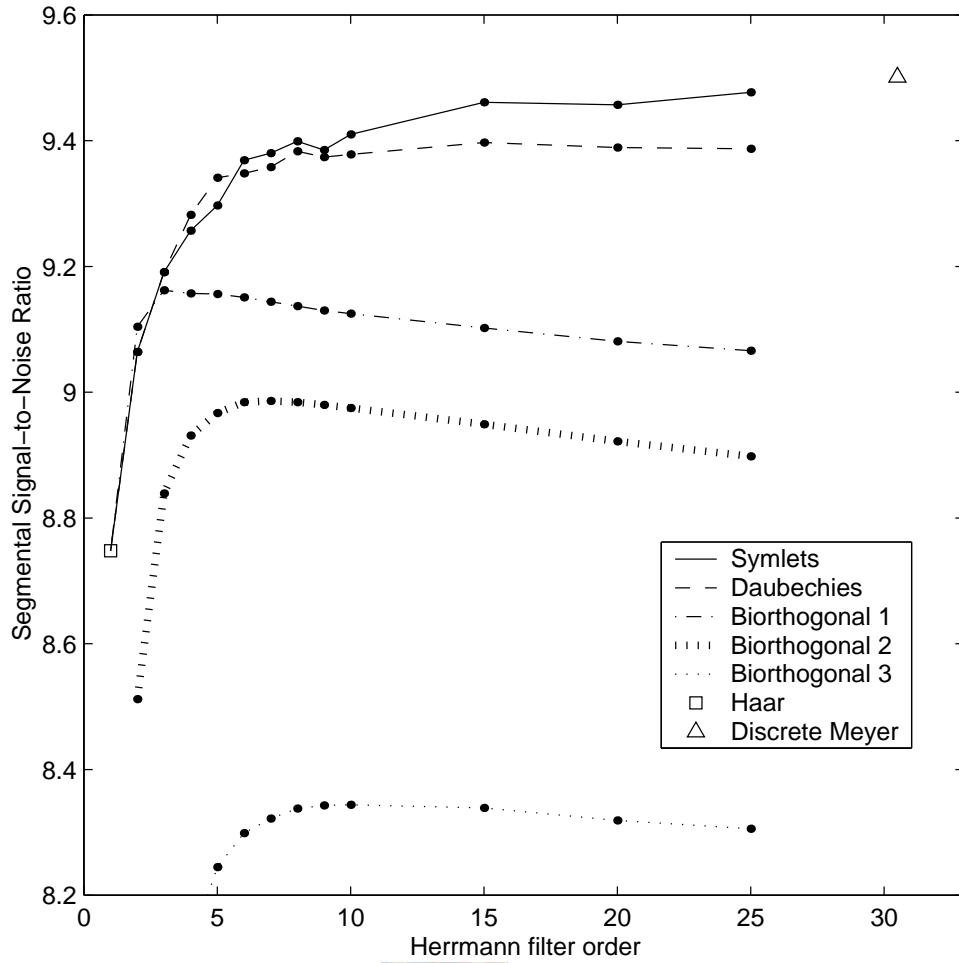
- All wavelets clearly enhance speech because it outperforms the noisy distortion value of  $d_{IS} = 1.031$  which is not shown in Figure 6.10.
- For most wavelets, the quality of enhanced speech starts to converge from a Herrmann filter order of  $m > 10$ . As the Herrmann order increases, the wavelet filters become closer to being ideal halfband filters and therefore perform better.

- The Haar wavelet does not perform as well as the higher-order Daubechies, Symlet and Biorthogonal 1 wavelets. This is because the Haar wavelet has a Herrmann order of  $m = 1$  and is therefore not an ideal halfband filter.
- The Daubechies and Symlet wavelets show similar performance, which is expected as they are very similar wavelets.
- The Biorthogonal 1 wavelet performance is superior to that of all other wavelets used in this study. These filters have a short highpass and a long lowpass reconstruction order. In image compression, the short highpass filter avoids ringing, whereas the long lowpass filter produces good smoothing [52]. According to the  $d_{IS}$  measure, these are also qualities wanted for speech enhancement. It should be noted that the Biorthogonal 1 filter is not maximally flat (see Figure 4.12) and its satisfactory performance is surprising.
- The Biorthogonal 2 wavelet performance is inferior to that of the Biorthogonal 1 wavelet, and the Biorthogonal 3 wavelet performs even worse. The Biorthogonal 2 and Biorthogonal 3 wavelet filters deviate from being halfband filters, which leads to an increase in errors in the reconstruction step of the DWT. Their magnitude responses are also not nearly flat, which further induces errors.
- The Discrete Meyer wavelet performs slightly less desirably than the higher-order Daubechies and Symlet wavelets and significantly more poorly than the Biorthogonal 1 wavelet. This is unexpected, since the Discrete Meyer wavelet filters are closer to being ideal in the bandpass and cut-off gradient regions. Figure 4.12(a) shows that the magnitude response of the Discrete Meyer wavelet filters is more distorted in the stopband region, compared to that of the Daubechies, Symlet and Biorthogonal 1 wavelets. This may be the reason for the poorer performance of the Discrete Meyer wavelet on the  $d_{IS}$  measure.

### 6.6.2 The segmental signal-to-noise ratio ( $d_{SEGSNR}$ ) evaluation

By looking at the  $d_{SEGSNR}$  results shown in Figure 6.11 and comparing it to the  $d_{IS}$  results of Figure 6.10, the following observations are made:

- All wavelets, as with the  $d_{IS}$  evaluation, drastically improve on the noisy distortion of  $d_{SEGSNR} = 6.196$  which is not shown in Figure 6.11.
- Apart from the Biorthogonal 1 family (which performs worse on  $d_{SEGSNR}$ ) and the Discrete Meyer wavelet (which shows superior  $d_{SEGSNR}$  performance), the  $d_{IS}$  and



**Figure 6.11:** The segmental signal-to-noise ratio  $d_{SEGSNR}$  evaluation of different wavelet families with different filter orders. The noisy  $d_{SEGSNR} = 6.196$ .

$d_{SEGSNR}$  results show similar performance for the different wavelets.

- The Biorthogonal 1 wavelet does not perform as well on the  $d_{SEGSNR}$  measure as on the  $d_{IS}$  measure. This is ascribed to the magnitude response of the Biorthogonal 1 filters which is not nearly flat in the bandpass region. A residual noise is introduced if coefficients are wrongfully attenuated. This noise can be further amplified because of resonant peaks (non-flatness) in the magnitude response of synthesis filters. The poor  $d_{SEGSNR}$  performance is ascribed to this enlargement in the residual noise power which lowers the signal-to-noise ratio.
- The Discrete Meyer  $d_{SEGSNR}$  performance is the best of all the wavelets. This is because its magnitude response is almost maximally flat in the bandpass region and it has the steepest cut-off gradient. This produces the smallest residual noise power and hence the best  $d_{SEGSNR}$  results.

- The Discrete Meyer and high-order Daubechies and Symlet wavelets significantly outperform the Biorthogonal wavelets. This shows the importance of using filters with a magnitude response which is very flat in the bandpass region.
- Only Symlet wavelets show an increase in quality as the Herrmann order increases with  $m > 10$ . The performance of Daubechies wavelets converges, whereas the Biorthogonal performance decreases.

As the Herrmann order increases, the magnitude response of the maximally flat Daubechies and Symlet wavelet filters approximates an ideal halfband response. Symlets still have a nearly linear phase response, whereas the Daubechies phase response become more non-linear. Symlet wavelets are superior to Daubechies wavelets at high Herrmann orders because they preserve persistence, which the HMT utilises.

The magnitude response of Biorthogonal filters becomes more distorted with higher Herrmann orders, which introduces more noise and explains the decrease in  $d_{SEGSNR}$  quality.

### 6.6.3 Subjective evaluation

Enhancing speech with different wavelets produces different noise artifacts which are not necessarily captured by the objective measures. Listening to the enhanced speech is a good way to evaluate how perceptually annoying these artifacts are.

The two sentences for subjective listening tests<sup>5</sup> are enhanced under the same conditions that yielded the distortion curves of Figures 6.10 and 6.11. A Herrmann order of  $m = 15$  is used for the Daubechies, Symlet, Biorthogonal 1, Biorthogonal 2 and Biorthogonal 3 wavelet families. Using these wavelets and also denoising with the Haar and Discrete Meyer wavelets lead to denoising with seven different wavelets.

Informal listening tests show that a difference in  $d_{IS}$  of more than 0.05 and in  $d_{SEGSNR}$  of more than 0.5 can be perceived by the ear. Certain wavelets have similar residual artifacts and three such groups can clearly be detected, which is also apparent in the  $d_{IS}$  and  $d_{SEGSNR}$  distortion curves.

The three groups are given below in rank of perceptual preference, with the best listed first. The results of the informal listening tests are compared to that of the objective

---

<sup>5</sup>See Appendix B.

evaluation.

### 1. **The Daubechies/Symlets/Discrete Meyer group**

The enhanced speech using the Daubechies, Symlets and Discrete Meyer wavelets are indistinguishable to the ear. The residual artifact sounds like white noise which is perceptually pleasing. This agrees with both the  $d_{IS}$  and  $d_{SEGSNR}$  distortion measures, where these three wavelets produced similar results. This group is perceptually the most pleasing and is preferred above the other wavelets.

### 2. **The Biorthogonal wavelets**

The enhanced speech using Biorthogonal wavelets share the same type of artifact, which sounds like coloured noise with a strong high frequency content. This artifact is perceptually more annoying than that of the Daubechies/Symlets/Discrete Meyer group.

The residual noise level of the Biorthogonal 2 wavelet is noticeably higher than that of the Biorthogonal 1 wavelet. The Biorthogonal 3 wavelet has the highest perceived noise level. This agrees with both the  $d_{IS}$  and  $d_{SEGSNR}$  results, where the same pattern is clearly seen.

The Biorthogonal 1 wavelet has the best  $d_{IS}$  values, which is in contrast with this subjective test and the  $d_{SEGSNR}$  results, where the Daubechies/Symlets/Discrete Meyer group is superior. This observation verifies the fact that good performance is dependent on both objective measures, which indicates that the Biorthogonal 1 wavelet is not a favourable choice.

### 3. **The Haar wavelet**

The enhanced speech using the Haar wavelet sounds very scratchy and by far the worst of all the families. The residual artifact of the Haar wavelet is a blocky, step-like signal which can clearly be heard and is very annoying. This does not show up in the objective evaluation, where the Haar wavelet is given moderate ratings.

## 6.6.4 A good wavelet for speech

From the objective and subjective evaluation, it is clear that the Haar wavelet and the Biorthogonal wavelets are inferior to the Daubechies/Symlets/Discrete Meyer group. Although the Daubechies and Symlet wavelets are perceptually indistinguishable, Symlets outperform Daubechies on the  $d_{SEGSNR}$  objective measure, and are therefore superior to them.

Perceptually there is little difference between the Discrete Meyer and Symlet wavelets. Symlets slightly outperform the Discrete Meyer wavelet on the  $d_{IS}$  measure, whereas the Discrete Meyer wavelet has superior  $d_{SEGSNR}$  results. Therefore there are little to choose between the high-order ( $m \approx 20$ ) Symlets and the Discrete Meyer wavelet. The Discrete Meyer wavelet is used by both Bron [8] and Cohen [13] in recent wavelet-based speech enhancement research, which leads to its choice as the best wavelet amongst those considered for speech enhancement.

## 6.7 Choosing the best frame size

A signal is called stationary if its statistical properties do not change over time [36]. Speech signals are not stationary because speech consists of a sequence of phonemes each having different properties. However, it is reasonable to assume that sections of phonemes are stationary [36]. The *quasi-stationary assumption* states that segments of speech are stationary within a frame of analysis. Most speech processing applications therefore use a short-time approach based on frames of speech, with the frame size chosen to satisfy the quasi-stationary assumption.

In STSA speech enhancement, which is Fourier-based, a longer frame size produces higher frequency resolution, which is desirable. The frame size must, however, be short enough to be inside the quasi-stationary range. Most STSA speech enhancement algorithms use a frame size of 32 ms for speech sampled at  $F_S = 8$  kHz as in [23, 55, 58, 59, 60]. Shorter frame sizes may be used at higher sampling frequencies, such as the 25.6-ms frame size for  $F_S = 10$  kHz speech used in [40]. Both of these lead to 128 frequency bins per frame, which provides fine enough frequency resolution for speech enhancement.

In addition to the quasi-stationary requirement of Fourier-based speech enhancement, the statistical wavelet-based speech enhancement algorithms such as the HMT, HMM and GMM also need enough training data to produce accurate models. The amount of training data decreases as the analysis frame becomes shorter, which places a lower limit on the frame size. If the frame is too long, the segment cannot be assumed to be stationary and information from neighbouring phonemes will be used to model a phoneme with completely different statistical characteristics.

An experiment is done which investigates the effect of using different frame sizes. The TIMIT24WGN with a global SNR of 10 dB is enhanced with the HMT algorithm and implemented as described in Section 6.3.1. The wavelet transform restricts the frame size to be a power of two, as seen in Section 4.4.3. The sampling frequency therefore

plays a role in choosing the frame size. Enhancing speech with  $F_S = 8$  kHz leads to a possible frame size within the set {8ms, 16ms, 32ms, 64ms, 128ms, 256ms, 256ms} for this experiment. Speech enhancement based on a frame size outside this set is expected to perform poorly, since a shorter frame size implies very little training data and a longer frame size suffers from non-stationarity.

Table 6.9 shows the  $d_{SEGSNR}$  evaluation of the enhanced speech by using different frame sizes. The speech-only sections (Speech) and the different phoneme groups are evaluated.

**Table 6.9:** *The segmental signal-to-noise ratio  $d_{SEGSNR}$  evaluation of HMT speech enhancement using different frame sizes.*

Segment size	$d_{SEGSNR}$					
	Speech	Vowels	Nasals	Semivowels	Fricatives	Stops
64 (8 ms)	9.098	13.110	4.613	12.925	2.998	2.433
128 (16 ms)	9.312	13.228	4.974	13.097	3.364	2.825
256 (32 ms)	9.501	13.335	5.387	13.259	3.641	3.107
512 (64 ms)	<b>9.559</b>	<b>13.364</b>	<b>5.547</b>	<b>13.319</b>	<b>3.717</b>	<b>3.190</b>
1024 (128 ms)	9.422	13.212	5.431	13.122	3.619	3.117
2048 (256 ms)	9.266	13.047	5.279	12.896	3.421	3.002
Noisy	6.211	11.248	-0.766	10.045	-0.255	-1.806

The  $d_{SEGSNR}$  distortion values show that the optimum frame size for all phoneme groups is 64 ms. This is slightly unexpected compared to a 32-ms frame size, since 64-ms frames may include neighbouring coefficients, whereas 32-ms frames are unlikely to contain more than one phoneme. The larger amount of training data with 64-ms frames is likely responsible for the superior  $d_{SEGSNR}$  distortion values.

It should be noted that the performance of 32-ms frames is only slightly inferior to 64-ms frames, which still makes it a good choice. Frame sizes of 8 ms and 16 ms are too short and produce very little training data. The model parameters cannot be trained accurately and therefore result in poor performance. Frame sizes of 128 ms and 256 ms are too long and result in the analysis of non-stationary signals. The model parameters are incorrectly trained based on coefficients from neighbouring phonemes. These inaccurate model parameters lead to the poorer performance. It is therefore concluded that a 64-ms frame size is the best choice for speech with  $F_S = 8$  kHz.



## 6.8 Choosing the best algorithm

The statistical properties of wavelet coefficients of real-world signals are discussed in Chapter 4.5. It would be interesting to see how successful the HMT, HMM and GMM methods are at capturing the statistical properties of speech in the wavelet domain. It is not a trivial task to quantify these properties or measure to what extent these properties are present in speech signals. However, some indication of their presence is found by examining the denoising performance of statistical wavelet-based techniques that exploit these properties.

The four statistical techniques, namely HMT, HMM, GMM and basic Wiener, are employed. Table 6.10 indicates how the different algorithms capture the statistical properties of wavelet coefficients.

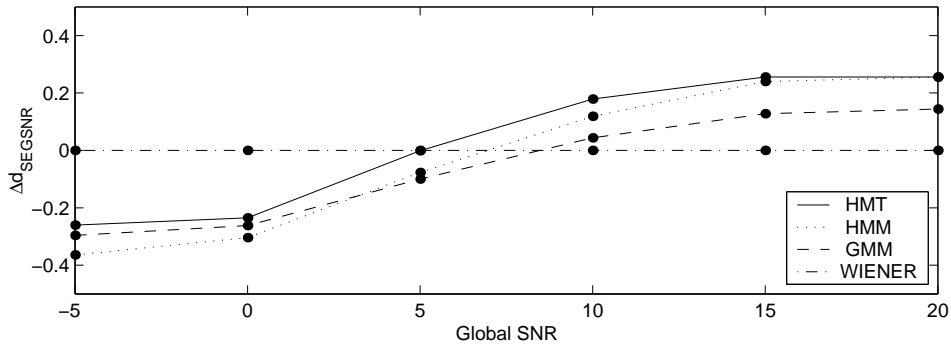
**Table 6.10:** *The four statistical wavelet-based speech enhancement algorithms each exploit the indicated statistical properties of the wavelet coefficients of speech.*

Method	Sparsity (P3)	Clustering (S1)	Persistence (S2)
Wiener	No	No	No
GMM	Yes	No	No
HMM	Yes	Yes	No
HMT	Yes	Yes	Yes

### 6.8.1 Objective evaluation

An experiment which uses the framework discussed in Section 6.3.1 is done which investigates the different algorithms over different global signal-to-noise ratios. The TIMIT24WGN set is enhanced with the Wiener and fully trained HMT, HMM and GMM algorithms at global SNRs of  $-5$  dB,  $0$  dB,  $5$  dB,  $10$  dB,  $15$  dB and  $20$  dB. An eight-level Discrete Meyer wavelet transform is used, resulting in  $32$  ms non-overlapping analysis segments.

Figures 6.12 and 6.13 compare the  $d_{SEGSNR}$  and  $d_{IS}$  performance between the Wiener, GMM, HMM and HMT speech enhancement algorithms over a range of global signal-to-noise ratios. It is found that the distortion values are very close to each other and therefore the Wiener method is used as a reference and the difference in  $d_{SEGSNR}$  and  $d_{IS}$  between the HMT, HMM and GMM methods and the Wiener method is shown.



**Figure 6.12:** Comparative segmental signal-to-noise ratio  $d_{SEGSNR}$  evaluation of the speech-only sections using the Wiener, GMM, HMM and HMM algorithms.

The  $d_{SEGSNR}$  evaluation in Figure 6.12 shows two definite regions:

- **Low noise levels (Global SNR > 5 dB).**

At low noise levels, most of the signal coefficients are large and most noise coefficients are small, which satisfies the HMT, HMM and GMM modelling assumptions. The non-Gaussianity and intercoefficient dependencies of speech in the wavelet domain can be detected by the HMT, HMM and GMM algorithms.

The HMT, HMM and GMM methods outperform the Wiener method, which indicates that they do succeed in capturing non-Gaussianity. The HMT and HMM methods also outperform the GMM method, which implies that clustering and persistence are present. The HMT, however, only slightly outperforms the HMM method, which implies that persistence is not as strong in speech as might be expected. The above-mentioned observations agree with Table 6.10, where algorithm performance is expected to become more satisfactory as it captures more statistical properties.

It should be noted that the differences between the algorithms are very small and barely audible, which suggests that the sparsity, clustering and persistence properties of speech in the wavelet domain are not very strong.

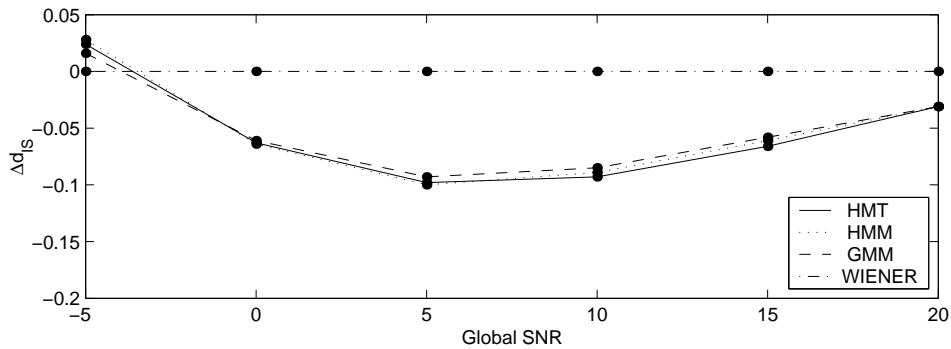
- **High noise levels (Global SNR < 5 dB).**

At high noise levels Wiener denoising outperforms the more computationally intensive statistical methods. Under these conditions, most of the large coefficients result from noise instead of from the clean signal. The observed noisy coefficients become more Gaussian and the intercoefficient dependencies are also lost.

The less satisfactory  $d_{SEGSNR}$  performance of the HMT, HMM and GMM methods at high noise levels can be ascribed to the statistical methods attempting to find

patterns within the noise. The non-linear shrinkage function allows the HMT, HMM and GMM methods to incorrectly keep the large coefficients (which is noise in this case) and to attenuate the small coefficients (which might be signal coefficients). The Wiener method is based on linear shrinkage and all coefficients are shrunk by using a single multiplier. This is more desirable for the denoising of signals with a Gaussian distribution.

The  $d_{IS}$  evaluation in Figure 6.13 shows the following:



**Figure 6.13:** *Comparative Itakura-Saito  $d_{IS}$  evaluation of the speech-only sections using the Wiener, GMM, HMM and HMM algorithms.*

- The HMT, HMM and GMM methods outperform the Wiener method for nearly all global signal-to-noise ratios. This differs from the  $d_{SEGSNR}$  results, which show superior performance only for global SNRs higher than 5 dB. The  $d_{IS}$  results do however suggest that the HMT, HMM and GMM methods are better at preserving perceived speech quality than the Wiener method.
- The HMT, HMM and GMM  $d_{IS}$  performances are almost equal. The capturing of intercoefficient dependencies does not show up on the  $d_{IS}$  distortion measure.

The  $d_{IS}$  and  $d_{SEGSNR}$  evaluation suggests that the HMT method is the best wavelet-based speech enhancement algorithm to use amongst those evaluated, but only with global signal-to-noise ratios of 5 dB and higher. The HMM method, which is a simpler model than the HMT, is only slightly inferior to the HMT method on the  $d_{SEGSNR}$  measure and is also suggested to be superior to the GMM and Wiener methods. For SNRs of 5 dB and lower the Wiener method should be used.

## 6.8.2 Formal subjective evaluation

A formal subjective evaluation is done to compare the different wavelet-based denoising algorithms, as described in Appendix B.4. Because the objective evaluation done in Section 6.8.1 suggests that the wavelet-based denoising methods are more suited for low noise levels, we do the formal listening test on noisy speech recordings with a signal-to-noise ratio of 10 dB.

The results of the listening test are shown in Table 6.11. The four statistical denoising algorithms (HMT, HMM, GMM and Wiener) are compared. The original clean (CLN) and un-enhanced noisy (NSY) signals are also included, to serve as reference. The results show the number of times (out of 48 trials) that the method shown on the left in Table 6.11 is preferred to the method shown on top. Statistically significant preferences (at 5% significance level) are indicated in boldface, following the procedure described in Appendix B.4. The coarse model scores in the rightmost column of Table 6.11 are used to rank the algorithms.

**Table 6.11:** *The listening test results show the number of times (out of 48) that a row method is preferred to a column method. Significant preferences are shown in boldface. The total number of times that a method is preferred is shown in the Total column.*

Preferred	CLN	HMT	WIE	HMM	GMM	NSY	Total
CLN	-	<b>47</b> /48	<b>46</b> /48	<b>48</b> /48	<b>48</b> /48	<b>47</b> /48	236/240
HMT	<b>1</b> /48	-	25/48	22/48	29/48	<b>43</b> /48	120/240
WIE	<b>2</b> /48	23/48	-	25/48	29/48	<b>40</b> /48	119/240
HMM	<b>0</b> /48	26/48	23/48	-	25/48	<b>43</b> /48	117/240
GMM	<b>0</b> /48	19/48	19/48	23/48	-	<b>37</b> /48	98/240
NSY	<b>1</b> /48	<b>5</b> /48	<b>8</b> /48	<b>5</b> /48	<b>11</b> /48	-	30/240

The following observations can be made:

- All the algorithms significantly improve the un-enhanced noisy signal, since their outputs are preferred to the noisy signal in nearly all the trials. We can therefore safely say that the wavelet-based algorithms enhance noisy speech.
- The original clean signal is significantly preferred to all other signals, being preferred in 236 of the 240 trials. This shows that there is still room for improvement of the denoising algorithms.

- The differences between the HMT, HMM, GMM and Wiener algorithms are not statistically significant. Based on this test, these algorithms can be considered to be indistinguishable. The near-equal preference counts indicate that the evaluators were indecisive in their choices between these algorithms.
- Although not quite significant, the GMM algorithm appears to have worse quality than the rest of the wavelet-based methods. This difference can only be confirmed by expanding the listening test to include more evaluators and sentences.

## 6.9 Conclusions

Experiments were done on a few aspects of wavelet-based speech enhancement, namely the denoising of speech-segments, the floor parameter, the wavelet, the frame size and a comparison between different algorithms. The conclusions of these experiments are summarised below.

### 6.9.1 Denoising of speech-segments

Speech signals are divided into five different groups of phonemes which contain similar statistics, namely vowels, nasals, semivowels, fricatives and stops. In Section 6.2.2 an experiment, similar to the Donoho-Johnstone denoising experiment [11, 14], was done on speech segments from these phoneme groups. It was found that the statistical algorithms, namely Wiener, GMM, HMM and HMT denoising, are superior to the classical methods, namely VisuShrink, SureShrink and HybridSure. These statistical methods have the most potential to be implemented as speech enhancement algorithms.

### 6.9.2 The noise floor parameter

The wavelet domain represents the signal in octave frequency bands, and does not have the fine frequency resolution of the Fourier domain. This results in a classical problem of speech enhancement in the discrete wavelet transform (DWT), where segments of speech are easily eliminated when attenuating wavelet coefficients. This creates gaps in the speech spectrogram and thus high speech distortion. It was shown in Section 6.5 that this effect clearly shows up in the Itakura-Saito  $d_{IS}$  distortion measure, where these gaps in the spectrogram are represented by extremely high sporadic distortion values. By using

a noise floor, these  $d_{IS}$  problem segments are eliminated, while the noise floor also masks residual noise artifacts.

The effect of the floor parameter  $\beta$  was investigated objectively in Section 6.5.2, subjectively in Section 6.5.4 and from an LPC viewpoint in Section 6.5.3. From the above-mentioned experiments, the floor parameter is chosen to be  $\beta = 0.2$ .

### 6.9.3 The wavelet

The specific wavelet used in the DWT and IDWT is of importance. Although perfect reconstruction is possible for all wavelets, the chosen wavelet influences the following:

1. **The statistical properties of the wavelet coefficients.**

An example of this is wavelet filters with linear phase. These filters preserve persistence (property S2), because they preserve the alignment of the coefficients across resolution levels. Such filters should be used when denoising with the HMT algorithm, which attempts to utilise persistence.

2. **The type of residual artifact.**

Wrongfully attenuated wavelet coefficients result in a residual noise which is directly characterised by the form of the wavelet itself. An example of this is the Haar wavelet. It is a blocky wavelet and results in a perceptually annoying “scratchy” residual artifact.

3. **The level of the residual noise.**

Wavelet filters which are not maximally flat in the bandpass region, enhance wrongfully attenuated coefficients, and hence the level of the residual noise. Biorthogonal wavelet filters are examples of such filters.

By taking the above-mentioned factors into account, it is suggested in Section 6.6 that the Discrete Meyer or higher-order Symlet wavelets (Herrmann order  $\approx 20$ ) should be used for speech enhancement.

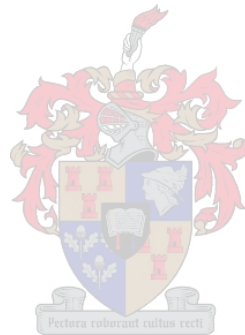
### 6.9.4 The frame size

Speech enhancement is frame-based and the effect of different frame sizes was investigated in Section 6.7. A long frame size leads to more training data and therefore more accurate model parameters. If the frame size is too long, the speech frame cannot be assumed to

be stationary. It is experimentally shown that the best frame size for speech sampled at  $F_S = 8$  kHz is 64 ms, although a 32-ms frame size is also a good choice.

### 6.9.5 Comparing HMT, HMM, GMM and Wiener speech enhancement

Because the HMT algorithm is a good image denoising algorithm [14, 47] and because it attempts to capture all three statistical properties (sparsity, clustering and persistence), it was expected to outperform the other denoising methods. It is, in contrast, shown in Section 6.8, that the objective evaluation of the HMT, HMM and GMM algorithms show similar results and subjectively there is very little difference between them. The HMT, HMM and GMM algorithms are suggested to be good speech enhancement algorithms, but only when the global signal-to-noise ratio is not too low (or the noise level not too high).



# Chapter 7

## Comparing STSA with HMTs on speech

### 7.1 Introduction

In this chapter the statistical wavelet-based speech enhancement algorithms developed in Chapter 6 are compared to the current state-of-the-art Fourier-based STSA techniques discussed in Chapter 2. A widely-used STSA technique is the Ephraim-Malah MMSE STSA algorithm [23], which is chosen as representative of the Fourier-based techniques. The algorithms are evaluated by objective measures and subjective listening tests, as described in Chapter 3.

In Section 7.2 a noise floor is introduced into the Ephraim-Malah algorithm. The noise floor parameter  $\beta$  and the Ephraim-Malah decision-directed weighting factor  $\alpha$  is objectively and subjectively evaluated and chosen in Sections 7.2.1 and 7.2.2, respectively.

The Ephraim-Malah algorithm and the wavelet-based algorithms are experimentally compared with each other in Section 7.3. In Section 7.3.1 global objective measures are used for the comparison. In Section 7.3.2 the algorithms are compared on a phoneme group level. The Ephraim-Malah algorithm and the wavelet-based algorithms are subjectively compared in Section 7.3.3. In Section 7.4 the conclusions of this chapter are briefly summarised.



## 7.2 The Ephraim-Malah algorithm

The standard Ephraim-Malah speech enhancement algorithm [23] is implemented by using the Ephraim-Malah MMSE amplitude suppression rule described in Section 2.2.1 and the Ephraim-Malah decision-directed  $\xi_k$  estimate described in Section 2.2.2.

One way to mask the musical noise artifact of STSA speech enhancement based on the power spectral subtraction rule [42], is to use a noise floor which overestimates the *a priori* SNR [23]. A noise floor may also be introduced in the Ephraim-Malah algorithm by inserting a floor parameter  $\beta$  into the Ephraim-Malah decision-directed  $\xi_k$  estimate (2.20). The new *a priori* SNR estimate is given as

$$\hat{\xi}_k = \alpha \frac{|\hat{\mathbf{X}}_k^{\text{pf}}|^2}{\sigma_d^2(k)} + (1 - \alpha) \max[\gamma_k - 1, \beta], \quad \alpha \in [0, 1]. \quad (7.1)$$

Using (7.1) with  $\beta > 0$ , the *a priori* SNR is overestimated in the case where  $\gamma_k < \beta + 1$ . It is also seen in (7.1) that the algorithm has two parameters, namely a weighting factor  $\alpha$  and a noise floor parameter  $\beta$ , that has to be chosen before implementation. The weighting factor is suggested by Ephraim and Malah [23] to be  $\alpha = 0.98$ , which they found to produce the least annoying residual artifact. The floor parameter  $\beta$  should be large enough to mask the musical noise, but also small enough to produce only a slight residual noise floor.

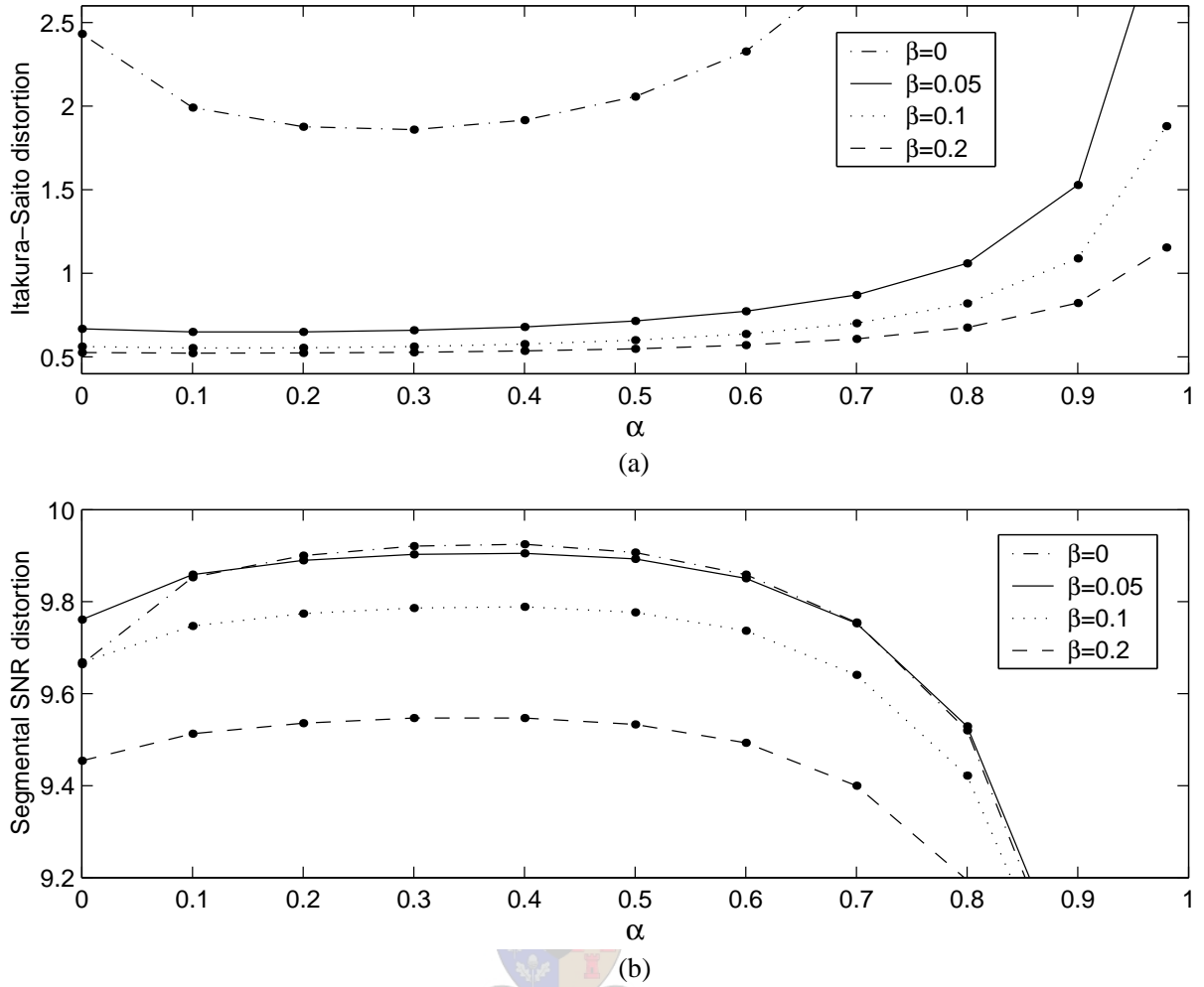


### 7.2.1 Objective evaluation of the Ephraim-Malah algorithm

An experiment is done which investigates the effect of the weighting factor  $\alpha$  and the noise floor parameter  $\beta$  on objective distortion measures. The aim of this experiment is to find the combination of  $\alpha$  and  $\beta$  which produces the most desirable distortion values.

The TIMIT24WGN set, with a global SNR of 10 dB, is enhanced with the Ephraim-Malah algorithm using half-overlapping 32 ms frames and Hanning windows as in [23]. The weighting factor, which has a range of  $\alpha \in [0, 1)$ , is chosen from set  $\alpha \in \{0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.98\}$ . Because the floor parameter cannot be too large, it is chosen to be in the set  $\beta \in \{0, 0.05, 0.1, 0.2\}$ . The  $d_{IS}$  and  $d_{SEGSNR}$  distortion measures (see Section 3.2) are used to evaluate the speech-only sections of the enhanced speech. By choosing a certain noise floor and then changing the value of  $\alpha$ , a distortion curve is gained for each value of  $\beta$ .

The  $d_{IS}$  and  $d_{SEGSNR}$  evaluation results are shown in Figure 7.1.



**Figure 7.1:** Objective evaluation of the Ephraim-Malah algorithm by using different values for  $\alpha$  and  $\beta$ . (a) Itakura-Saito distortion ( $d_{IS}$ ). (b) Segmental signal-to-noise ratio distortion ( $d_{SEGSNR}$ ).

### Choosing $\alpha$ objectively

Distortion curves for the weighting factor  $\alpha$  are created for each of the investigated noise floor values ( $\beta \in \{0, 0.05, 0.1, 0.2\}$ ) and for both objective measures. These curves, which are shown in Figure 7.1, all have a similar form and yield the same conclusions in terms of the choice of  $\alpha$ .

The  $d_{SEGSNR}$  measure indicates a suitable choice of the weighting factor within the range  $0.2 \leq \alpha \leq 0.5$ , while the  $d_{IS}$  measure shows desirable performance when  $0.1 \leq \alpha \leq 0.4$ . The weighting factor is therefore objectively chosen as  $\alpha = 0.3$ .

Setting  $\alpha = 0$  is equivalent to the maximum likelihood  $\xi_k$  estimation approach described in

Section 2.2.2. From especially the  $d_{SEGSNR}$  evaluation it can be seen that as  $\alpha$  increases, the Ephraim-Malah decision-directed approach outperforms the maximum likelihood approach. The weighting factor  $\alpha$  should therefore not be too small.

For high values of the weighting factor ( $\alpha > 0.7$ ), the *a priori* SNR  $\xi_k$  is mainly estimated from the previous frame of analysis and the observed SNR of the current frame is neglected. The distortion measures show that this leads to high speech distortion and the weighting factor  $\alpha$  should therefore not be too large.

### Choosing $\beta$ objectively

As the noise floor increases from  $\beta = 0$  to  $\beta = 0.2$ , the  $d_{SEGSNR}$  measure shown in Figure 7.1(b) decreases, which imply an increase in distortion. This is expected, as a higher noise floor leads to a higher noise power and hence a smaller signal-to-noise ratio. The best  $d_{SEGSNR}$  performance is therefore obtained with no noise floor ( $\beta = 0$ ).

On the other hand, the  $d_{IS}$  measure, shown in Figure 7.1(a), shows unfavourable performance with no noise floor. This is related to the discussion on  $d_{IS}$  problem segments of wavelet-based denoising in Section 6.5, where characteristic information from certain frequency bands are shown to be eliminated.

Setting  $\beta = 0.05$  barely reduces the  $d_{SEGSNR}$  performance but it shows a dramatic increase in  $d_{IS}$  performance as these values become smaller. This observation clearly shows that the  $d_{IS}$  distortion measure is especially harsh on certain frames, as described in Section 6.5. It also gives an indication of the masking effect, where a slight noise floor, which is barely audible (seen by the nearly equal  $d_{SEGSNR}$  performance), masks the musical noise artifact (seen by the great difference in the  $d_{IS}$  performance).

With  $\beta = 0.1$ , the  $d_{IS}$  distortion values show a slight increase in performance, whereas the  $d_{SEGSNR}$  distortion values become less desirable, indicating that the noise floor is on the verge of being too high.

With  $\beta = 0.2$ , the noise floor is too high and the  $d_{SEGSNR}$  performance shows a dramatic decrease with little gain in  $d_{IS}$  performance. As mentioned in Section 6.5.4, an excessive noise floor level defeats the purpose of denoising because the residual artifact becomes very noisy.

According to the  $d_{IS}$  and  $d_{SEGSNR}$  distortion measures, the noise floor parameter should therefore be chosen to be in the range  $0.05 \leq \beta \leq 0.1$ .

## 7.2.2 Subjective evaluation of the Ephraim-Malah algorithm

In Section 7.2.1 the weighting factor  $\alpha$  and the noise floor parameter  $\beta$  of the Ephraim-Malah algorithm were objectively investigated. Two experiments are done here which investigate the subjective effect of these two parameters, via informal listening tests.

### Choosing $\alpha$ subjectively

From the objective evaluation in Section 7.2.1 it was suggested that  $\alpha = 0.3$  should be used, whereas Ephraim and Malah [23] subjectively suggests a totally different value, namely  $\alpha = 0.98$ .

The effect of the weighting factor is evaluated by enhancing the two sentences for subjective evaluation<sup>1</sup>. The noisy sentences are corrupted with WGN to have a global SNR of 10 dB and are then enhanced by the Ephraim-Malah algorithm. Enhancement is done without a noise floor ( $\beta = 0$ ), because the various residual artifacts, which a noise floor would mask, are of interest here.

The following values for the weighting factor  $\alpha$  are implemented and evaluated:

- With  $\alpha = 0$ , the musical noise residual artifact is strong and very annoying.
- With  $\alpha = 0.98$ , the residual noise is colourless and much less annoying than with  $\alpha = 0$ , which was also found by Ephraim-Malah in [23]. Although there is barely any “musical” noise, the enhanced speech is distorted and sounds very “hollow” (as if spoken into a bottle) and therefore results in a reduction in speech quality.
- Setting  $\alpha = 0.3$  produces high speech quality but also a level of musical noise. The musical noise is less annoying than with  $\alpha = 0$ , however.

The weighting factor  $\alpha$  can therefore be seen as a parameter which creates a trade-off between musical noise and “hollow” speech distortion. Comparing this to the objective evaluation of Section 7.2.1, it is seen that satisfactory  $d_{IS}$  and  $d_{SEGSNR}$  performance correspond to high speech quality but also a certain level of “musical” noise.

---

<sup>1</sup>See Appendix B.

### Choosing $\beta$ subjectively

The aim of using a noise floor is to set its level such that it masks musical noise while being only slightly audible itself.

The two sentences for subjective evaluation are again corrupted with WGN to have a global SNR of 10 dB. These are then enhanced with the Ephraim-Malah algorithm, using a weighting factor of  $\alpha = 0.3$  and varying the floor parameter within the set  $\beta \in \{0, 0.05, 0.1, 0.2\}$ .

The informal listening tests show that a noise floor parameter of  $0.05 \leq \beta \leq 0.1$  is successful in masking musical noise. This corresponds to the objective evaluation of Section 7.2.1.

## 7.3 Comparing Fourier-based and wavelet-based speech enhancement

In this section the Fourier-based Ephraim-Malah MMSE STSA [23] algorithm and the statistical wavelet-based algorithms (Wiener, GMM, HMM and HMT) are compared with each other.

It is expected that the Ephraim-Malah algorithm will outperform the wavelet-based methods because of its finer frequency resolution and its smoother half-overlapping analysis technique.

The wavelet transform represents the signal in octave frequency bands, some of which are very large compared to that of the Fourier domain. Although the poor frequency resolution of the wavelet domain is not optimal for speech enhancement, the wavelet domain has a fine time resolution in the larger frequency bands. The wavelet-based algorithms might therefore perform better than Fourier-based algorithms on segments of speech which change abruptly, which may increase intelligibility. The Fourier-based and wavelet-based speech enhancement methods are also expected to have very different residual artifacts.

The algorithms are implemented with a noise floor parameter. The HMT algorithm uses an eight-level discrete Meyer wavelet decomposition, which results in 32-ms non-overlapping analysis frames. The wavelet-based floor parameter of  $\beta^W = 0.2$  is implemented in (6.4). These parameters are chosen from the experiments done in Sec-

tions 6.5.2 to 6.5.4. The Ephraim-Malah algorithm is implemented as in Section 7.2, with a weighting factor of  $\alpha = 0.3$  and a noise floor parameter of  $\beta^F = 0.05$ .

The Itakura-Saito  $d_{IS}$  and segmental signal-to-noise ratio  $d_{SEGSNR}$  measures are used as objective evaluation. The objective evaluation is done in two parts, as suggested by [33]:

1. In Section 7.3.1 global distortion measures are computed for a range of different global signal-to-noise ratios. This evaluates algorithm performance under different noise levels.
2. In Section 7.3.2 distortion measures are computed for the five different phoneme groups. This evaluates algorithm performance on a phoneme level.

### 7.3.1 Global objective measures

In this experiment global distortion measures are calculated over a range of global SNRs to investigate how the algorithms perform under different levels of noise.

The four statistical wavelet-based speech enhancement algorithms, namely Wiener, GMM, HMM and HMT (see Sections 5.6 to 5.10), are compared to the Ephraim-Malah MMSE STSA algorithm [23]. The wavelet-based algorithms are not expected to perform well under high noise levels. This is because the observed signal loses its non-Gaussianity and inter-coefficient dependencies, which the statistical algorithms (HMT, HMM and GMM) attempt to capture. The wavelet-based algorithms might perform satisfactory and even outperform the Ephraim-Malah algorithm for medium to low noise levels, where the model definition of the wavelet-based algorithms correspond to the statistics of the observed signal.

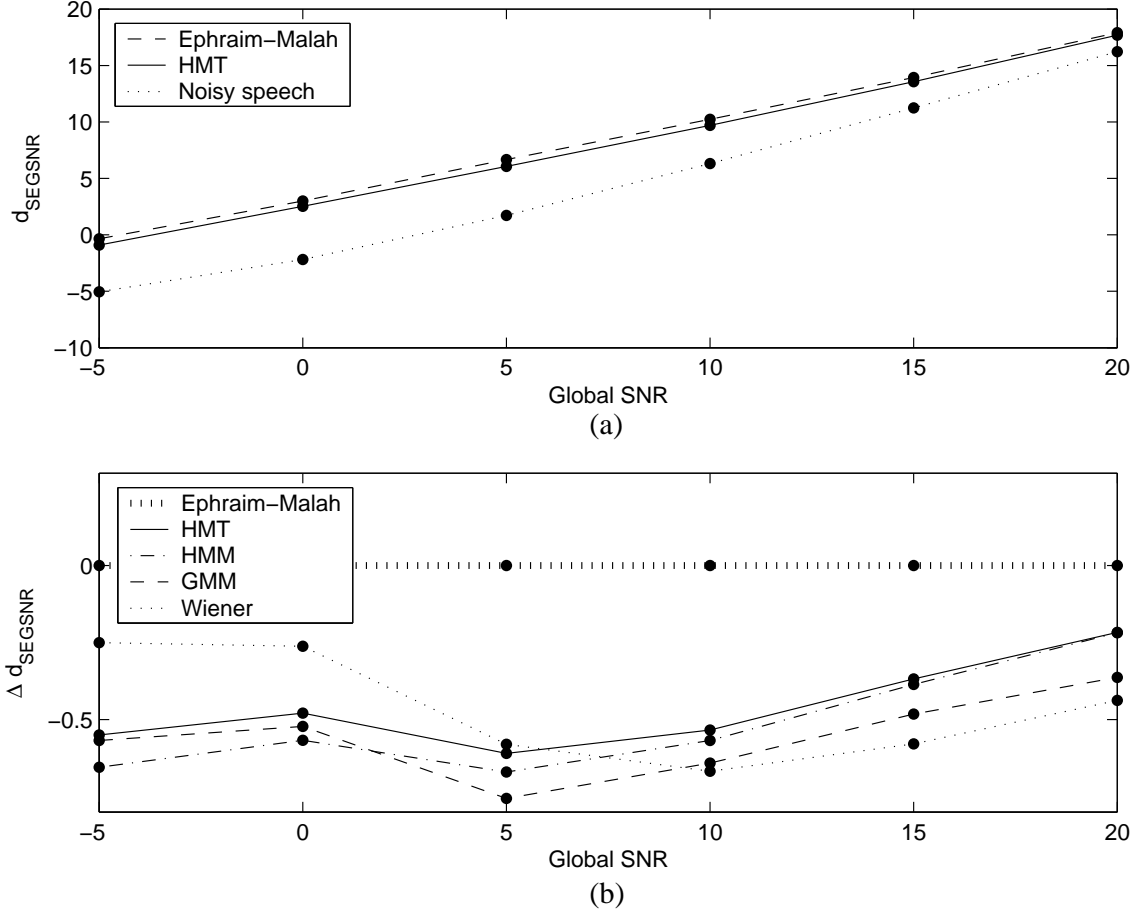
The TIMIT192WGN set<sup>2</sup>, which contains 192 sentences with global signal-to-noise ratios in the set  $\{-5 \text{ dB}, 0 \text{ dB}, 5 \text{ dB}, 10 \text{ dB}, 15 \text{ dB}, 20 \text{ dB}\}$ , is enhanced using the Ephraim-Malah algorithm and wavelet-based algorithms, all having parameter values as described in Section 7.3.

Figures 7.3.1 and 7.3.1 show the  $d_{SEGSNR}$  and  $d_{IS}$  global objective evaluation of the different algorithms over a range of global signal-to-noise ratios. Because the performance of the wavelet-based algorithms are very similar, the difference between the wavelet-based methods and the Ephraim-Malah algorithm is also shown (in Figures 7.3.1(b) and 7.3.1(b)) to

---

<sup>2</sup>See Appendix B.

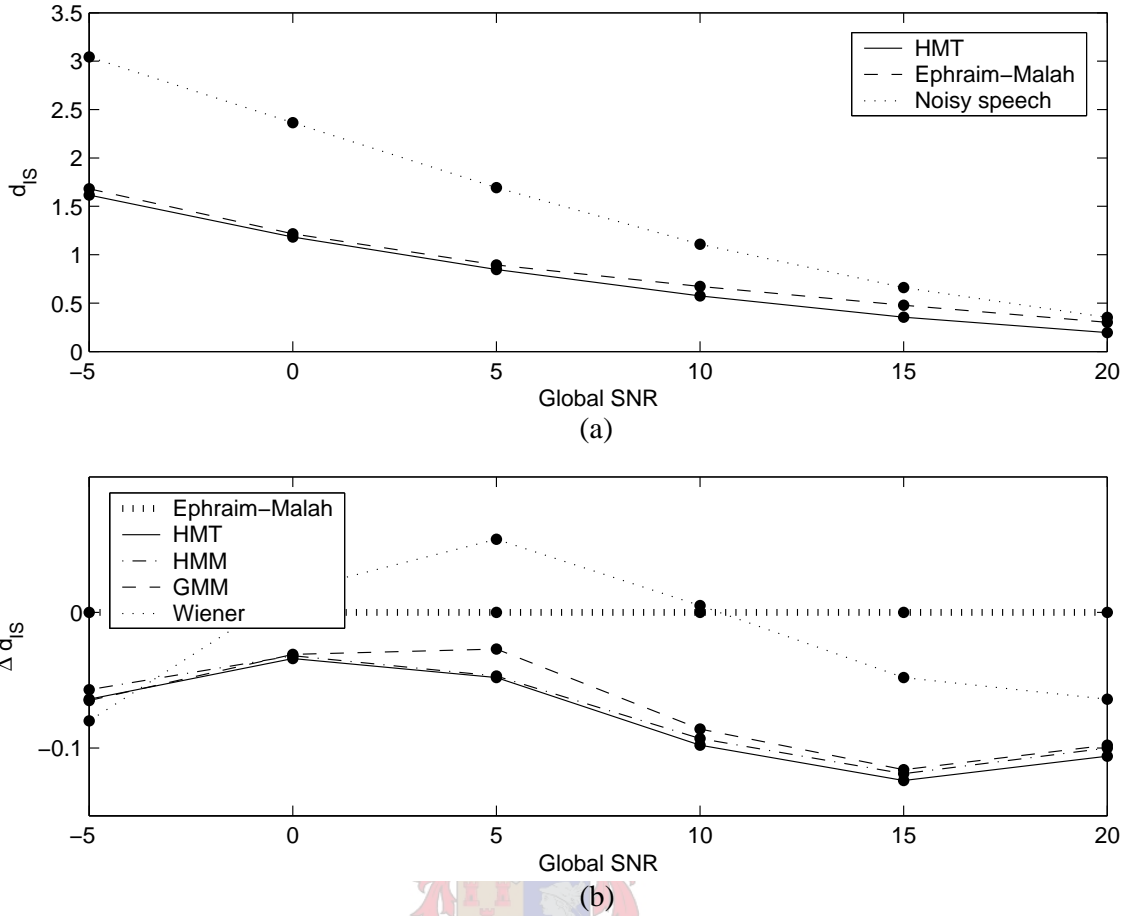
visually enlarge the differences between them. It should also be noted that the segmental signal-to-noise ratio of the noisy speech (dotted line) in Figure 7.3.1(a) differs from the global signal-to-noise ratio (y-axis) because the  $d_{SEGSNR}$  measure is averaged in the log-domain rather than the linear domain.



**Figure 7.2:** (a) The global  $d_{SEGSNR}$  evaluation over a range of global signal-to-noise ratios. Higher  $d_{SEGSNR}$  values imply better performance. (b) The wavelet-based algorithms relative to the Ephraim-Malah algorithm

The following observations are made from the  $d_{SEGSNR}$  and  $d_{IS}$  experimental results:

- Both the Ephraim-Malah and the wavelet-based algorithms clearly enhance speech for all noise levels. As the noise level decreases (which is an increase in global SNR) the algorithms perform only slightly better than the unprocessed noisy signal. This is expected since additive WGN of low magnitude does not distort speech significantly.
- The most interesting observation is that the wavelet-based methods outperform the Ephraim-Malah algorithm on the  $d_{IS}$  measure (lower is better), whereas the opposite



**Figure 7.3:** (a) The global  $d_{IS}$  evaluation over a range of global signal-to-noise ratios. Lower  $d_{IS}$  values imply better performance. (b) The wavelet-based algorithms relative to the Ephraim-Malah algorithm

is true for the  $d_{SEGSNR}$  measure (higher is better), which shows the Ephraim-Malah algorithm to be superior. It is deduced that the Ephraim-Malah algorithm reduces the noise strongly (seen from the superior  $d_{SEGSNR}$  results) at the cost of speech quality (seen from the inferior  $d_{IS}$  results). Because of the poor frequency resolution of the wavelet domain, the wavelet-based algorithms need a larger noise floor to retain speech quality. In this experiment, the noise floor is chosen to produce the best trade-off between noise reduction and speech quality from the experiments done in Sections 6.5 and 7.2.

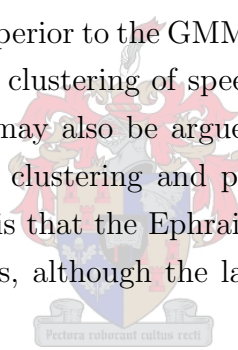
- At low noise levels (15 dB to 20dB) the Ephraim-Malah algorithm only slightly outperforms the wavelet-based methods on the  $d_{SEGSNR}$  measure, whereas the wavelet-based methods outperform the Ephraim-Malah algorithm on the  $d_{IS}$  measure. This shows that the wavelet-based methods are able to detect the non-Gaussianity and inter-coefficient dependencies of the unobserved clean signal. It implies that the



HMT, HMM and GMM methods are superior to the Ephraim-Malah algorithm in retaining speech quality at low noise levels.

- As the noise level increases from moderate to high levels (decrease in global SNR), the Ephraim-Malah method starts to noticeably outperform the wavelet-based methods on the  $d_{SEGSNR}$  measure, while their  $d_{IS}$  values are almost equal. This is expected, since the inter-coefficient dependencies and non-Gaussian statistics of the underlying clean signal become less apparent.
- The  $d_{SEGSNR}$  distortion measure shows that the HMT and HMM algorithms perform better than the Wiener and GMM algorithms at low noise levels. This difference is very small, however. The HMT, HMM and GMM perform equally well on the  $d_{IS}$  distortion measure. This is unexpected, since a bigger difference between the performance of these algorithms would be expected.

From the global objective measures it is deduced that the HMT and HMM algorithms perform similarly, are slightly superior to the GMM algorithm and outperforms the Wiener algorithm. The persistence and clustering of speech coefficients are not strong and only present at low noise levels. It may also be argued that the statistical methods are not suitable to capture the type of clustering and persistence found in speech coefficients. The main conclusion, however, is that the Ephraim-Malah algorithm reduces more noise than the wavelet-based methods, although the latter produces better speech quality at low noise levels.



### 7.3.2 Phoneme class objective measures

In this experiment the algorithm performance is evaluated on a phoneme group level. Speech is divided into the five phoneme groups described in Section 6.2.1, namely vowels, nasals, semivowels, fricatives and stops.

Because the Fourier domain uses sinusoidal basis functions, it is suitable to represent signals which have a harmonic nature. Such signals are represented by only a few large Fourier coefficients. It is expected that speech enhancement in the Fourier domain will perform well on harmonic phonemes such as nasals and semivowels. The wavelet domain has a multiresolution representation and it is therefore expected that the wavelet-based algorithms might outperform the Fourier-based Ephraim-Malah algorithm on non-harmonic phonemes such as fricatives and stops.

The TIMIT192WGN set with a global signal-to-noise ratio of 10 dB is enhanced with the

Ephraim-Malah and HMT algorithms, with parameter values as described in Section 7.3. By using the TIMIT phoneme labels, a single global objective measure is computed for each of the five phoneme groups.

The  $d_{SEGSNR}$  and  $d_{IS}$  distortion values are shown in Tables 7.1 and 7.2. The *Speech* columns refer to the global distortion measures for the speech-only sections which correspond to Figures 7.3.1 and 7.3.1. The number of analysis frames are shown in brackets.

**Table 7.1:** *The phoneme group  $d_{SEGSNR}$  evaluation of the Ephraim-Malah, HMT and noisy speech for the TIMIT192WGN set with a global SNR of 10 dB.*

Algorithm	$d_{SEGSNR}$ (# of Frames)					
	(55075) Speech	(27227) Vowels	(4516) Nasals	(5904) Semivow.	(11860) Fric.	(3752) Stops
Ephraim-Malah	10.232	<b>14.107</b>	<b>5.957</b>	<b>13.270</b>	3.735	2.963
HMT method	9.698	13.425	5.249	12.321	3.621	2.951
Noisy speech	6.320	11.045	-0.961	8.788	-0.167	-1.870

**Table 7.2:** *The phoneme group  $d_{IS}$  evaluation of the Ephraim-Malah, HMT and noisy speech for the TIMIT192WGN set with a global SNR of 10 dB.*

Algorithm	$d_{IS}$ (# of Frames)					
	(55075) Speech	(27227) Vowels	(4516) Nasals	(5904) Semivow.	(11860) Fric.	(3752) Stops
Ephraim-Malah	0.672	0.496	1.045	0.773	0.799	0.922
HMT method	0.574	<b>0.429</b>	1.002	0.776	<b>0.567</b>	<b>0.696</b>
Noisy speech	1.109	0.797	2.484	1.738	0.924	1.240

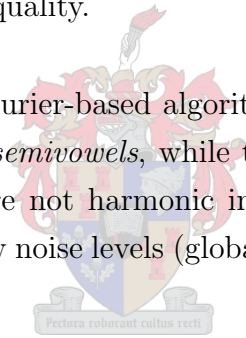
Tables 7.1 and 7.2 highlight the following:

- Both the Ephraim-Malah and wavelet-based methods clearly enhance speech for all phoneme groups. This is seen by comparing algorithm performance with that of the unprocessed noisy speech.
- The  $d_{SEGSNR}$  values of Table 7.1 show that the Ephraim-Malah algorithm clearly outperforms the wavelet-based methods on nasals and semivowels, whereas the performance of fricatives and stops are very similar. This is to be expected, since nasals

and semivowels are phonemes with a high harmonic content which suits the Fourier-based Ephraim-Malah algorithm. The wavelet-based algorithms have wavelets as basis functions, which are not ideal to model harmonic signals.

- The  $d_{IS}$  values of Table 7.2 show that the wavelet-based algorithms outperform the Ephraim-Malah algorithm on fricatives and stops, whereas the performance of nasals and semivowels are very similar. The wavelet-based algorithms perform better on non-harmonic signals such as stops, which are signals with abrupt changes, and fricatives, which are essentially white.
- In the case of vowels, the wavelet-based methods perform better according to  $d_{IS}$ , whereas the Ephraim-Malah algorithm perform better according to  $d_{SEGSNR}$ . Because vowels are far more frequent than any of the other phoneme groups, this pattern is also seen in the performance of the speech-only sections. This observation confirms the experimental discussion of Section 7.3.1, which states that the Ephraim-Malah algorithm eliminates more noise than the wavelet-based algorithms, but at the cost of speech quality.

The conclusion is made that Fourier-based algorithms perform better on harmonic-type phonemes, such as *nasals* and *semivowels*, while the wavelet-based algorithms are more suitable for phonemes which are not harmonic in nature, such as *stops* and *fricatives*. However, this is only true for low noise levels (global SNR > 5 dB) which suit the wavelet-based algorithms.



### 7.3.3 Subjective evaluation

To comment on the residual artifacts of the Fourier-based and wavelet-based algorithms, it is necessary to make a subjective comparison. It is also a good way to investigate the level of background noise and speech distortion.

An informal subjective listening test is done, where the two sentences for subjective evaluation are corrupted by WGN to have a global SNR of 10 dB, and enhanced with the Ephraim-Malah MMSE STSA algorithm and the HMT algorithm. The algorithms use the parameters as chosen in Section 7.3.

From the informal listening tests, there are three main differences between the algorithms:

1. Their residual artifacts.
2. Their level of noise reduction and speech quality.

3. Their capability to enhance voiced and unvoiced phonemes.

### **The residual artifacts**

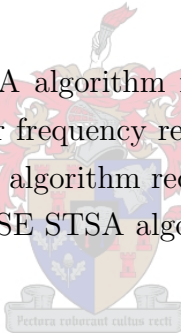
The residual artifacts of the two types of algorithms differ greatly.

The Ephraim-Malah MMSE STSA algorithm is Fourier-based, and its residual artifact is the “musical” noise commonly observed in STSA techniques. The residual noise is random sinusoids spanning the total length of a frame.

The wavelet-based HMT algorithm has no “musical” noise, but rather a “scratchy” artifact because of its multiresolution decomposition. The residual noise consists of actual wavelets of variable time spans, depending on the resolution level.

### **Noise reduction and speech quality**

The Ephraim-Malah MMSE STSA algorithm reduces more noise than the HMT algorithm. This is because of the finer frequency resolution of the Fourier domain compared to the wavelet domain. The HMT algorithm requires a large noise floor parameter, compared to the Ephraim-Malah MMSE STSA algorithm, to retain speech quality.



### **Voiced and unvoiced sounds**

The wavelet-based HMT algorithm leads to crisper unvoiced sounds compared to the Ephraim-Malah MMSE STSA algorithm. This is due to the multiresolution framework of the wavelet domain.

The voiced phonemes are of higher quality with the Fourier-based Ephraim-Malah MMSE STSA algorithm than with the HMT algorithm. This is because of the fine frequency resolution of the Fourier domain, which represents harmonic signals more compactly than the wavelet domain.

## **7.3.4 Formal subjective evaluation**

A formal subjective evaluation is done to compare the wavelet-based HMT denoising algorithm [14] with the Fourier-based Ephraim-Malah algorithm [23], as described in

Appendix B.4. The listening test formed part of the test described in Section 6.8.2 and all recordings therefore have a signal-to-noise ratio of 10 dB.

The results of the listening test are shown in Table 7.3. The two algorithms Hidden Markov Tree (HMT) and Ephraim-Malah (STSA) are compared. The original clean (CLN) and un-enhanced noisy (NSY) signals are also included, to serve as reference. The results show the number of times (out of 48 trials) that the method shown on the left in Table 7.3 is preferred to the method shown on top. Statistically significant preferences (at 5% significance level) are indicated in boldface, following the procedure described in Appendix B.4. The coarse model scores in the rightmost column of Table 7.3 are used to rank the algorithms.

**Table 7.3:** *The listening test results show the number of times (out of 48) that a row method is preferred to a column method. Significant preferences are shown in boldface. The total number of times that a method is preferred is shown in the Total column.*

Preferred	CLN	HMT	STSA	NSY	Total
CLN	-	<b>47/48</b>	<b>47/48</b>	<b>47/48</b>	141/144
HMT	<b>1/48</b>	-	27/48	<b>38/48</b>	66/144
STSA	<b>1/48</b>	21/48	-	<b>35/48</b>	57/144
NSY	<b>1/48</b>	<b>10/48</b>	<b>13/48</b>	-	24/144

The following observations can be made:

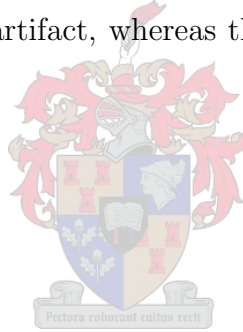
- The results found in Section 6.8.2 are verified in this experiment, where all the algorithms significantly improve the un-enhanced noisy signal and the original clean signal is significantly preferred to all other signals.
- The difference between the HMT and STSA algorithms is not quite statistically significant. Based on this test, these algorithms can be considered to be equivalent, although the HMT algorithm appears to be slightly more preferred to STSA. This difference can only be confirmed by expanding the listening test to include more evaluators and sentences.

## 7.4 Conclusions

In this chapter the parameters for the Ephraim-Malah algorithm are chosen and this algorithm is then compared to the statistical wavelet-based speech enhancement algorithms.

The parameters for the Ephraim-Malah algorithm are chosen in Section 7.2. The noise floor parameter is selected as  $0.05 \leq \beta \leq 0.1$  and the weighting factor as  $\alpha = 0.3$ , based on objective distortion measures. These values are then subjectively verified and it is also seen that the weighting factor  $\alpha$  produces a trade-off between “musical” noise and speech distortion.

The Ephraim-Malah algorithm and the wavelet-based algorithms are compared in Section 7.3. From the experiments using global objective measures in Section 7.3.1, it is deduced that the HMT and HMM algorithms perform almost equally well and slightly better than the GMM algorithm. These wavelet-based methods does not reduce the noise as much as the Ephraim-Malah algorithm, but they retain speech quality better under light noise conditions. In Section 7.3.2 the algorithms are compared on a phoneme group level. The Fourier-based Ephraim-Malah algorithm performance is superior on harmonic-type phonemes, such as *nasals* and *semivowels*, while the wavelet-based algorithms are more suitable for phonemes which are not harmonic in nature, such as *stops* and *fricatives*. The subjective evaluation in Section 7.3.3 shows that the Ephraim-Malah algorithm has a “musical” noise residual artifact, whereas the wavelet-based algorithms produce a “scratchy” residual artifact.



# Chapter 8

## Conclusions

The conclusions of the theoretical and experimental research done in this study are discussed in this chapter. Section 8.1 briefly summarises the conclusions of the experiments done in this study. Several differences between speech enhancement and image denoising are discussed in Section 8.2. Recommendations for future research are given in Section 8.3.

### 8.1 Conclusions of this study

The wavelet-based Hidden Markov Model (HMM) [14] denoising algorithm is implemented in Chapter 5. We propose that the HMM denoising algorithm outperforms the state-of-the-art Hidden Markov Tree [14] algorithm on the Donoho-Johnstone [19] *Doppler* test signal. Although the Doppler signal is not representative of typical images, it is similar to seismic, radar and sonar signals.

In Chapter 6, different wavelet-based speech enhancement algorithms were investigated. The statistical speech enhancement algorithms, namely Wiener, GMM, HMM and HMT, are superior to the classical methods, namely VisuShrink, SureShrink and HybridSure. The use of a noise floor eliminates problem segments and also masks residual noise artifacts. The noise floor is suggested to be  $\beta = 0.2$ . The specific wavelet used in the wavelet transform influences the statistical properties of the wavelet coefficients, the type of residual artifact and the level of the residual noise. It is suggested that the Discrete Meyer or higher-order Symlet wavelets (Herrmann order  $\approx 20$ ) should be used for speech enhancement. The best frame size for speech sampled at  $F_S = 8$  kHz is 64 ms. This produces the maximum amount of training data while still being within the quasi-stationary range of speech. It is found that the HMT, HMM and GMM algorithms yield similar

results for speech enhancement and should only be used under light noise conditions.

In Chapter 7, wavelet-based speech enhancement is compared to standard Fourier-based speech enhancement. It is found that the Fourier-based algorithms outperform the wavelet-based methods in very noisy conditions (global SNR  $< 5$  dB). At low noise levels, the Fourier-based algorithms perform better with harmonic-type phonemes, whereas the wavelet-based algorithms are more suitable for phonemes which are not harmonic in nature. The Fourier-based Ephraim-Malah MMSE STSA algorithm [23] produces “musical” noise, whereas the wavelet-based methods produce speech with a “scratchy” residual noise.

## 8.2 Speech enhancement vs image denoising

There are a number of differences between speech enhancement and image denoising:

- Speech signals are one-dimensional, whereas images are two-dimensional.
- A whole image is denoised in a single step, whereas speech enhancement is frame-based, which implies a separate denoising step for each frame.
- The typical global SNR differs between noisy speech signals and images.
- Statistical properties of the wavelet coefficients of speech signals and images differ.

This raises a few questions:

1. Is there enough *training data* in a speech frame to train the parameters of statistical models accurately?
2. Are the HMT, HMM and GMM suitable models to capture the *statistical properties* of the wavelet coefficients of speech?
3. Is the *global SNR* found in speech enhancement problems too low to capture the statistical properties?

These questions are answered in the following sections.



## 8.2.1 The training data of speech vs images

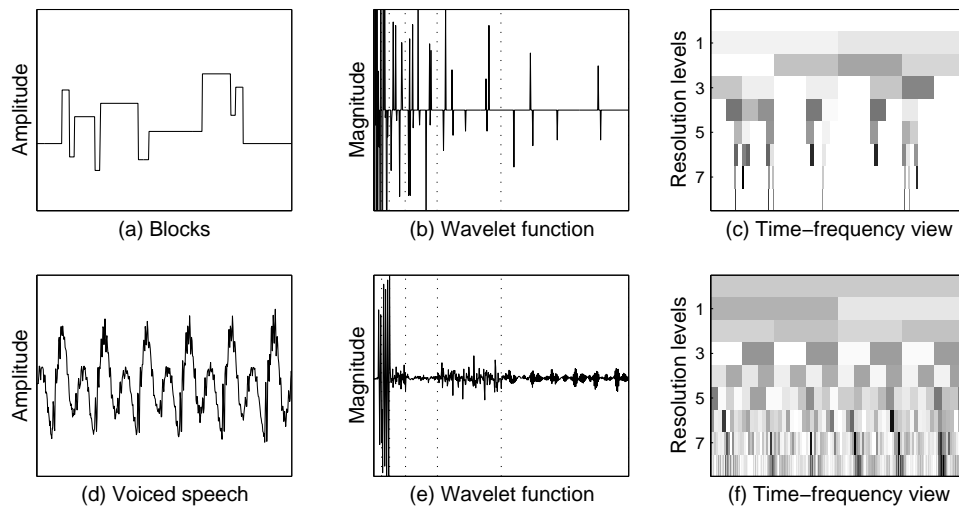
Because speech signals are one-dimensional and the enhancement process is frame-based, there is far less training data available for speech enhancement than for image denoising. For example, a frame of speech containing 256 samples results in 256 wavelet coefficients. The highest resolution level contains 128 samples, the second highest resolution level contains 64 samples, and so forth. In a  $256 \times 256$  image, the highest resolution level contains  $128 \times 128 = 16384$  samples and the second highest resolution level contains  $64 \times 64 = 4096$  samples. Clearly, the two-dimensional image yields far more training data compared to a one-dimensional speech frame. Although HMT denoising works well for images, it may be argued that there is too little training data for the statistical speech enhancement algorithms to have accurate model parameters.

It is, however, shown in Section 5.11 that the statistical HMT and HMM methods are still the superior wavelet-based denoising algorithms for the Donoho-Johnstone test signals. These signals are one-dimensional and contain a mere 1024 samples, which is similar to a speech frame. Because of this, it is deduced that there is enough training data in a speech frame, at least at the higher resolution levels. It is also expected that the statistical algorithms will perform better if the speech signals are recorded at a higher sampling rate.

## 8.2.2 Statistical properties of speech vs images

The HMT, HMM and GMM methods focus on three main statistical properties of wavelet coefficients, namely sparsity, clustering and persistence. It is concluded in Section 6.8 that these properties are not as strongly present in speech as in images. However, under light noise conditions these properties have enough presence to be useful.

Figure 8.1 shows the clean Blocks signal, its wavelet decomposition and the time-frequency tiling view of the wavelet coefficients, compared to that of clean voiced speech. The Blocks signal from the Donoho-Johnstone test signals [16] is a so-called “punctured smooth signal”, a quality typically associated with images. The example of voiced speech is from the TIMIT [28] sentence “timit/train/dr1/fcjf0/sa1.wav”. The three wavelet properties are discussed separately below.



**Figure 8.1:** *The Blocks signal compared to voiced speech. (a) The Blocks signal. (b) Wavelet function of Blocks (Haar wavelet decomposition). (c) Time-frequency tiling view of the coefficients of Blocks (normalised across scale). (d) An example of voiced speech. (e) Wavelet function of voiced speech (Discrete Meyer wavelet decomposition). (f) Time-frequency tiling view of the coefficients of voiced speech (also normalised).*

## Sparsity

It is seen in Figure 8.1(b) that the coefficients of Blocks are very sparse. There are a small number of large coefficients and a large number of small coefficients, with the large coefficients evenly spread around zero. This leads to the zero-mean, two-state model of the GMM, HMM and HMT algorithms. Figure 8.1(e) shows coefficients of voiced speech. These coefficients do have a non-Gaussian distribution, but they are not as sparse as those of Blocks.

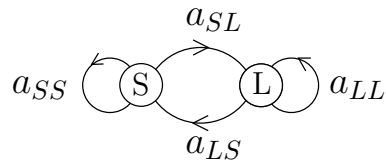
## Clustering

The HMT algorithm attempts to capture clustering by averaging statistical information over pairs of two neighbouring wavelet coefficients. The HMM algorithm is more sophisticated in its attempt to capture clustering. It models first-order Markovian dependencies within each resolution level.

It is seen in Figure 8.1(b) that the clusters of Blocks are of short duration and large in magnitude compared to surrounding coefficients. The clusters of voiced phonemes differ greatly from that of the Blocks signal (or the typical image). These clusters, which are

seen in Figure 8.1(e), are spread over a large number of coefficients and they are very noisy. Their spacing is also related to the fundamental pitch period. This is seen by comparing the six pitch pulses of Figure 8.1(d) with the six clusters in the highest resolution level (to the right of the last dotted line) of Figure 8.1(e).

Certain properties of the clusters found in voiced speech cannot be captured by the HMT algorithm, but are utilised by the HMM algorithm. This ability can be understood from its two-state ergodic model, shown in Figure 8.2.



**Figure 8.2:** *The two-state ergodic model of the HMM, with the two states, small and large, and the state probabilities shown.*

If both the self-loop transition probabilities  $a_{SS}$  and  $a_{LL}$  are large, the HMM effectively describes coefficient sequences containing consecutive runs of large and small coefficients. The strength of  $a_{LL}$  gives an indication of the cluster length (number of consecutive large coefficients), while  $a_{SS}$  describes the spacing between clusters (number of consecutive small coefficients). This inherent ability to model periodic clusters can explain why the HMM is more suitable than the HMT for describing the clusters found in speech.

## Persistence

From Figure 8.1(c) and (f) it is seen that persistence of voiced speech is not as strong as that of Blocks (or images). However, the persistence of speech does coincide with the pitch pulses, as seen by comparing Figure 8.1(d) with Figure 8.1(f).

### 8.2.3 The typical global SNR of speech vs images

In the Donoho-Johnstone denoising experiment (see Section 5.11), which is a standard test for wavelet denoising algorithms, the global signal-to-noise ratio is approximately 17 dB. Speech enhancement research focuses more on denoising speech which is corrupted with a high level of noise (global SNR in the order of 0 dB). Speech enhancement is therefore a more difficult problem, since it implies denoising under heavier noise conditions. The

high noise levels reduce the presence of the statistical properties of the underlying clean speech signal. It is found in Section 6.8 that the statistical methods should only be used when enhancing speech corrupted by light noise, with a global SNR of 5 dB and higher.

## 8.3 Future research

### 8.3.1 Domain recommendations

The wavelet packet domain has recently proved to be successful in speech enhancement [3, 4, 13, 27]. It has a multiresolution structure with many possible decompositions, ranging from the wavelet decomposition (which is a special case of the wavelet packet decomposition) to the uniform decomposition (which is closely related to the Fourier domain). A critical-band wavelet packet decomposition, which approximates a Bark or Mel scale, has also been successfully used [3, 4, 13]. The GMM and HMM wavelet-based speech enhancement methods developed in this study can easily be implemented in a critical-band wavelet packet domain. The HMT algorithm cannot be implemented in the wavelet packet domain, because it needs a binary tree structure in the time-frequency view of the coefficients, which only the wavelet domain provides.

It is suggested that the GMM and HMM could be used to exploit the non-Gaussianity and clustering of the coefficients of the wavelet packet domain. A problem with this is that the wavelet packet domain has a finer frequency resolution than the wavelet domain, and therefore provides less training data per resolution level. The redundant bark-scaled wavelet packet domain of Cohen [13] yields more training data for each resolution level and might therefore be a better transform for the statistical methods.

### 8.3.2 Sampling rate

All the experiments in this study are done on speech recordings with a sampling rate of  $F_S = 8$  kHz, because it is the practice in most research done on speech enhancement [23, 55, 58, 59, 60]. However, this is not the only standard used in speech research. Other popular sampling rates include 16 kHz as used in TIMIT [28], 20 kHz as used by Bagshaw [2], and the Compact Disc (CD) rate of 44.1 kHz.

Since a higher sampling rate increases the amount of training data for the GMM, HMM and HMT algorithms, they might perform even better. It is recommended that exper-

iments should be done with speech signals recorded at a higher sampling rate than the  $F_S = 8$  kHz used in this study.

### 8.3.3 Clusters of speech

It is suggested that future research should use more sophisticated methods to model the clusters found in speech. Because the speech-like clusters are very noisy, as seen in Figure 8.1(e), the following suggestions are made for the HMM algorithm in either the wavelet domain or the wavelet packet domain:

- Use an HMM with higher-order Markovian dependencies to account for the speech-like clusters, which are noisy and spread over a large number of coefficients.
- Train the HMM model on a smooth lowpass version of the observed coefficients and then use this model to attenuate the observed noisy coefficients.
- Enhance the clusters with the Teager energy operator as in [3, 4] and then train the HMM model on these altered coefficients.

### 8.3.4 Pitch tracking

As seen in Figure 8.1(d) and (f), the persistence in the wavelet domain coincides with the fundamental period of voiced speech. The frame-based HMT algorithm can be used to extract the position of the pitch pulses by using the persistence property of the coefficients. The HMT model could be trained on the clean speech signal, which will then enable the use of the HMT conditional probabilities, (5.75) and (5.76). The conditional probability  $P(s_i = L | \mathbf{w}, \mathcal{M})$  is the probability that a coefficient is large, whereas the probability  $P(s_i = L, s_{p(i)} = L | \mathbf{w}, \mathcal{M})$  gives an indication of the persistence of large coefficients. It is expected that the wavelet coefficients for which these probabilities are high, represent the position of the pitch pulses. The HMT can therefore be used as the basis of a pitch tracker.



# Bibliography

- [1] ABRY, P., *Ondelettes et Turbulence - Multirésolutions, Algorithmes de Décomposition, Invariance d'Echelle et Signaux de Pression*. Paris: Diderot, Editeur des Sciences et des Arts, 1997. English Translation: “Wavelets and Turbulence”.
- [2] BAGSHAW, P. C., HILLER, S. M., and JACK, M. A., “Enhanced Pitch Tracking and the Processing of F0 Contours for Computer Aided Intonation Teaching.” in *Proceedings of the European Conference on Speech Communication (Eurospeech)*, (Berlin), pp. 1003–1006, 1993.
- [3] BAHOURA, M. and ROUAT, J., “A New Approach for Wavelet Speech Enhancement.” *Proceedings of Eurospeech*, September 2001, pp. 1937–1940.
- [4] BAHOURA, M. and ROUAT, J., “Wavelet Speech Enhancement based on the Teager Energy Operator.” *IEEE Signal Processing Letters*, January 2001, Vol. 8, No. 1, pp. 10–12.
- [5] BENGIO, Y., “Markovian Models for Sequential Data.” *Neural Computing Surveys*, 1999, Vol. 2, pp. 129–162. <http://www.icsi.berkeley.edu/~jagota/NCS>.
- [6] BEROUTI, M., SCHWARTZ, R., and MAKHOUL, J., “Enhancement of Speech Corrupted by Acoustic Noise.” *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, April 1979, pp. 208–211.
- [7] BOLL, S. F., “Suppression of Acoustic Noise in Speech Using Spectral Subtraction.” *IEEE Transactions on Acoustics, Speech and Signal Processing*, April 1979, Vol. 2, No. 2, pp. 113–120.
- [8] BRON, A., “Wavelet-Based Denoising of Speech.” Master’s thesis, The Technion - Israel Institute of Technology, June 2000.
- [9] CAPPÉ, O., “Elimination of the Musical Noise Phenomenon with the Ephraim and Malah Noise Suppressor.” *IEEE Transactions on Speech and Audio Processing*, April 1994, Vol. 2, No. 2, pp. 345–349.

- [10] CHIPMAN, H., KOLACZYK, E., and McCULLOCH, R., “Signal De-noising Using Adaptive Bayesian Wavelet Shrinkage.” in *Proceedings of Third International Symposium on Time-Frequency and Time-Scale Analysis (TFTS-96)*, (Paris, France), pp. 225–228, 1996.
- [11] CHIPMAN, H., KOLACZYK, E., and McCULLOCH, R., “Adaptive Bayesian Wavelet Shrinkage.” *Journal of the American Statistical Association*, 1997, Vol. 92, pp. 1413–1421.
- [12] CHOI, H., “Software for Image Denoising using Wavelet-Domain Hidden Markov Tree Models.” Internet, September 2004.  
<http://www-dsp.rice.edu/software/whmt.shtml>.
- [13] COHEN, I., “Enhancement of Speech Using Bark-Scaled Wavelet Packet Decomposition.” *Proceedings of the 7th European Conference on Speech, Communication and Technology, EUROSPEECH-2001*, September 2001, pp. 1933–1936.
- [14] CRAUSE, M. S., NOWAK, R. D., and BARANUIK, R. G., “Wavelet-Based Statistical Signal Processing Using Hidden Markov Models.” *IEEE Transactions on Signal Processing*, April 1998. Special Issue on Wavelets and Filterbanks.
- [15] DELLER, J. R., HANSEN, J. H. L., and PROAKIS, J. G., *Discrete-time Processing of Speech Signals*. New York: Macmillan Publishing Company, 2000.
- [16] DONOHO, D. L., DUNCAN, M. R., HUO, X., and LEVI, O., “Wavelab 802 for Matlab 5.x.” Internet, September 2004. <http://www-stat.stanford.edu/~wavelab/>.
- [17] DONOHO, D. L. and JOHNSTONE, I. M., “Ideal Denoising in an Orthonormal Basis Chosen from a Library of Bases.” *Comp. Rend. Academy of Science Paris Ser.*, 1994, Vol. A 319, pp. 1317–1322.
- [18] DONOHO, D. L. and JOHNSTONE, I. M., “Ideal Spatial Adaptation by Wavelet Shrinkage.” *Biometrika*, 1994, Vol. 81, pp. 425–455.
- [19] DONOHO, D. L. and JOHNSTONE, I. M., “Adapting to Unknown Smoothness via Wavelet Shrinkage.” *Journal of the American Statistical Association*, December 1995, Vol. 90, pp. 1200–1224.
- [20] DONOHO, D. L. and JOHNSTONE, I. M., “Minimax Estimation via Wavelet Shrinkage.” *Annals of Statistics*, 1998, Vol. 26, pp. 879–921.

- [21] DONOHO, D. L., JOHNSTONE, I. M., KERKYACHARIAN, G., and PICARD, D., “Wavelet Shrinkage: Asymptotia?.” *Journal of the Royal Statistical Society*, 1995, Vol. B57, pp. 301–369.
- [22] ELDÈN, L., BERNTSSON, F., and REGÍNSKA, T., “Wavelet and Fourier Methods for Solving the Sideways Heat Equation.” *SIAM Journal on Scientific Computing*, 2000, Vol. 21, pp. 2187–2205. Also Tech. Report LiTH-MAT-R-97-22.
- [23] EPHRAIM, Y. and MALAH, D., “Speech Enhancement Using a Minimum Mean Square Error Short-Time Spectral Amplitude Estimator.” *IEEE Transactions on Acoustics, Speech and Signal Processing*, December 1984, Vol. 32, No. 6, pp. 1109–1121.
- [24] EPHRAIM, Y. and MALAH, D., “Speech Enhancement Using a Minimum Mean Square Error Log-Spectral Amplitude Estimator.” *IEEE Transactions on Acoustics, Speech and Signal Processing*, April 1985, Vol. 33, No. 2, pp. 443–445.
- [25] FODOR, I. K. and KAMATH, C., “On Denoising Images Using Wavelet-based Statistical Techniques.” *Lawrence Livermore National Laboratory technical report UCRL-JC-142357*, 2001. Center for Applied Scientific Computing, University of California.
- [26] FODOR, I. K. and KAMATH, C., “Denoising Through Wavelet Shrinkage: An Empirical Study.” *Journal of Electronic Imaging*, January 2003, Vol. 12, pp. 151–160.
- [27] FU, Q. and WAN, E. A., “A Novel Speech Enhancement System Based on Wavelet Denoising.” *Center of Spoken Language Understanding, OGI School of Science and Engineering at OHSU*, February 2003.  
<http://speech.bme.ogi.edu/research/nsel.htm>.
- [28] GAROFOLO, J. S., “Getting started with the DARPA TIMIT CD-ROM: An Acoustic Phonetic Continuous Speech Database.” National Institute of Standards and Technology (NIST), Gaithersburg, MD, 1988.
- [29] GRAY, R. M., BUZO, A., GRAY, A., and MATSUYAMA, Y., “Distortion Measures for Speech Processing.” *IEEE Transactions of Acoustics, Speech, and Signal Processing*, August 1980, Vol. 28, No. 4, pp. 367–376.
- [30] HANSEN, J. H. L., “Additive Noise Sources, Version 1.0.” Internet, September 2004. <http://cslr.colorado.edu/rspl/rspl-software.html>.



- [31] HANSEN, J. H. L. and CLEMENTS, M. A., “Constrained Iterative Speech Enhancement with Application to Speech Recognition.” *IEEE Transactions on Signal Processing*, April 1991, Vol. 39, No. 4, pp. 795–805.
- [32] HANSEN, J. H. L. and LEVENT, M. A., “Robust Feature-Estimation and Objective Quality Assessment for Noisy Speech Recognition Using the Credit Card Corpus.” *IEEE Transactions on Speech and Audio Processing*, 1995, Vol. 3, pp. 169–184.
- [33] HANSEN, J. H. L. and PELLON, B., “An Effective Quality Evaluation Protocol for Speech Enhancement Algorithms.” *Proceedings of the International Conference On Speech and Language Processing*, December 1998, Vol. 6, pp. 2819–2822.
- [34] HERRMANN, O., “On the Approximation Problem in Nonrecursive Digital Filter Design.” *IEEE Transactions on Circuit Theory*, May 1971, Vol. 18, No. 3, pp. 411–413.
- [35] HU, Y. and LOIZOU, P. C., “Speech Enhancement Based on Wavelet Thresholding the Multitaper Spectrum.” *IEEE Transactions on Speech Audio Processing*, January 2004, Vol. 12, No. 1, pp. 59–67.
- [36] HUANG, X., ACERO, A., and HON, H.-W., *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*. Upper Saddle River, NJ: Prentice-Hall, April 2001.
- [37] KAMATH, S. D., “A Multi-band Spectral Subtraction Method for Speech Enhancement.” Master’s thesis, The University of Texas at Dallas, December 2001.
- [38] KOLACZYK, E. D., *Wavelet Methods for the Inversion of Certain Homogeneous Linear Operators in the Presence of Noisy Data*. PhD thesis, Department of Statistics, Stanford University, Stanford, CA 94305-4065, September 1994.
- [39] LI, J., “Image Compression: The Mathematics of JPEG 2000.” *Modern Signal Processing MSRI Publications*, 2003, Vol. 46.
- [40] LIM, J. S. and OPPENHEIM, A. V., “All-Pole Modeling of Degraded Speech.” *IEEE Transactions of Acoustics, Speech and Signal Processing*, June 1978, Vol. 26, No. 3, pp. 197–210.
- [41] MATLAB, “The Mathworks Wavelet Toolbox version 2.2.” Internet, September 2004. <http://www.mathworks.com/products/wavelet/>.

- [42] McAULAY, R. J. and MALPASS, M. L., "Speech Enhancement Using a Soft-Decision Noise Suppression Filter." *IEEE Transactions on Acoustics, Speech and Signal Processing*, April 1980, Vol. 28, No. 2, pp. 137–145.
- [43] MOON, T. K., "The Expectation-Maximization Algorithm." *IEEE Signal Processing Magazine*, November 1996, pp. 47–60.
- [44] PEEBLES, P. Z., *Probability, Random Variables, and Signal Principles*. 3<sup>rd</sup> edition. McGraw-Hill, 1993.
- [45] PELLOM, B., "Objective Speech Quality Assessment, Version 1.0." Internet, September 2004. [http://cslr.colorado.edu/rspl/rspl\\_software.html](http://cslr.colorado.edu/rspl/rspl_software.html).
- [46] RABINER, L. R., "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition." *Proceedings of the IEEE*, February 1989, Vol. 77, No. 2, pp. 257–286.
- [47] ROMBERG, J. K., "A Universal Hidden Markov Tree Image Model." Master's thesis, Rice University, June 1999.
- [48] ROMBERG, J. K., CHOI, H., BARANIUK, R., and KINGSBURY, N., "Hidden Markov Tree Models for Complex Wavelet Transforms." *Submitted to IEEE Transactions on Signal Processing*, May 2002. <http://cmc.rice.edu/docs/docinfo.aspx?doc=Rom2002May1HiddenMark>.
- [49] ROMBERG, J. K., CHOI, H., and BARANIUK, R., "Shift-Invariant Denoising Using Wavelet-Domain Hidden Markov Trees." *Conference Record of The Thirty-Third Asilomar Conference on Signals, Systems and Computers*, October 1999.
- [50] SEOK, J. W. and BAE, K. S., "Speech Enhancement with Reduction of Noise Components in the Wavelet Domain." *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1997, pp. 1323–1326.
- [51] SHAPIRO, J. M., "Embedded Image Coding Using Zerotrees of Wavelet Coefficients." *IEEE Transactions on Signal Processing*, December 1993, Vol. 41, No. 12, pp. 3445–3462.
- [52] SHERLOCK, B. G., "Wavelets and Filter Banks." University of North Carolina, Charlotte USA. Notes on a course given at the University of Stellenbosch, August 2002.
- [53] THOMSON, D. J., "Spectrum Estimation and Harmonic Analysis." *Proceedings of the IEEE*, September 1982, Vol. 70, No. 9, pp. 1055–1096.

- [54] VARGA, A. and STEENEKEN, H. J. M., “Assessment for Automatic Speech Recognition: II, NOISEX-92: A Database and an Experiment to Study the Effect of Additive Noise on Speech Recognition Systems.” *Speech Communication*, July 1993, Vol. 12, No. 3, pp. 247–251. [http://spib.rice.edu/spib/select\\_noise.html](http://spib.rice.edu/spib/select_noise.html).
- [55] VIRAG, N., “Single Channel Speech Enhancement Based on Masking Properties of the Human Auditory System.” *IEEE Transactions on Speech and Audio Processing*, March 1999, Vol. 7, No. 2, pp. 126–137.
- [56] WICKERHAUSER, M. V., “Lectures on Wavelet Packet Algorithms.” Department of Mathematics, Washington University, St. Louis, MO, November 1991. <http://citeseer.ist.psu.edu/124574.html>.
- [57] WOLFE, P. J. and GODSILL, S. J., “Towards a Perceptually Optimal Spectral Amplitude Estimator for Audio Signal Enhancement.” *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing, Istanbul, 2000*, Vol. 2, pp. 821–824.
- [58] WOLFE, P. J. and GODSILL, S. J., “On Bayesian Estimation of Spectral Components for Broadband Noise Reduction in Audio Signals.” August 2001. <http://citeseer.ist.psu.edu/453331.html>.
- [59] WOLFE, P. J. and GODSILL, S. J., “Simple Alternatives to the Ephraim and Malah Suppression Rule for Speech Enhancement.” *Proceedings of the 11<sup>th</sup> IEEE Workshop on Statistical Signal Processing*, 2001, pp. 496–499.
- [60] WOLFE, P. J. and GODSILL, S. J., “Efficient Alternatives to the Ephraim and Malah Suppression Rule for Audio Signal Enhancement.” *EURASIP Journal on Applied Signal Processing, Special Issue on Digital Audio for Multimedia Communications*, February 2003.
- [61] WOLFE, P. J., “Matlab STSA Toolbox for Audio Signal Noise Reduction.” Internet, September 2004. <http://www-sigproc.eng.cam.ac.uk/~pjlw47/>.

# Appendix A

## The Itakura-Saito distortion measure

The Itakura-Saito distortion measure [29, 33]  $d_{IS}$  is calculated on a frame-by-frame basis, where  $d_{IS}(x, \hat{x})$  denotes the Itakura-Saito distortion between clean frame  $x[n]$  and denoised frame  $\hat{x}[n]$ .

The all-pole (or LP or AR) model of the current frame under analysis models the power spectral density of the frame as [29]

$$f(\omega) = \frac{\sigma^2}{|A(e^{j\omega})|^2}, \quad (\text{A.1})$$

where  $f(\omega)$  is referred to as the *linear prediction (LP) power spectrum*. It is a non-negative even function of  $\omega$ , which is the normalised frequency ranging from  $-\pi$  to  $\pi$ , where  $\pi$  corresponds to half the sampling frequency  $F_S/2$ . The polynomial  $A(e^{j\omega}) = \sum_{k=0}^P a_k e^{-jk\omega}$  with  $a_0 = 1$  is the transfer function of the linear prediction analysis filter of order  $P$  and the term  $\sigma^2$  is the all-pole gain or prediction error power [15].

The Itakura-Saito distortion measure  $d_{IS}$  describes the spectral matching properties of linear prediction. It is influenced by the similarity or difference between the LP power spectrum of the clean frame

$$f_c(\omega) = \frac{\sigma_c^2}{|A_c(e^{j\omega})|^2}, \quad (\text{A.2})$$

and the LP power spectrum of the denoised frame

$$f_d(\omega) = \frac{\sigma_d^2}{|A_d(e^{j\omega})|^2}. \quad (\text{A.3})$$

The Itakura-Saito distortion is given as [29]

$$d_{IS}(x, \hat{x}) = \left\| \frac{f_c(\omega)}{f_d(\omega)} - \ln \frac{f_c(\omega)}{f_d(\omega)} - 1 \right\|_1. \quad (\text{A.4})$$

The  $L_1$ -norm in (A.4) is [29]

$$\|g(\omega)\|_1 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |g(\omega)| d\omega . \quad (\text{A.5})$$

Using (A.5) and noting that for any real  $u$ ,  $u - \ln u - 1 \geq 0$ , the  $d_{IS}$  measure in (A.4) can be written as

$$d_{IS}(x, \hat{x}) = \frac{1}{2\pi} \int_{-\pi}^{\pi} \left\{ \frac{f_c(\omega)}{f_d(\omega)} - \ln \frac{f_c(\omega)}{f_d(\omega)} - 1 \right\} d\omega . \quad (\text{A.6})$$

The prediction error power  $\sigma^2$  can be written as [29]

$$\sigma^2 = \exp \left\{ \frac{1}{2\pi} \int_{-\pi}^{\pi} \ln [f(\omega)] d\omega \right\} . \quad (\text{A.7})$$

Substituting (A.7) into (A.6) leads to

$$\begin{aligned} d_{IS}(x, \hat{x}) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{f_c(\omega)}{f_d(\omega)} d\omega + \ln \frac{\sigma_d^2}{\sigma_c^2} - 1 \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} \frac{\sigma_c^2 |A_d(e^{j\omega})|^2}{\sigma_d^2 |A_c(e^{j\omega})|^2} d\omega + \ln \frac{\sigma_d^2}{\sigma_c^2} - 1 \end{aligned} \quad (\text{A.8})$$

The autocorrelation vector  $\mathbf{r} = [r_{xx}(-P) \dots r_{xx}(-1) \ r_{xx}(0) \ r_{xx}(1) \dots r_{xx}(P)]$  is used to create the autocorrelation matrix  $\mathbf{R}$ , as

$$\mathbf{R} = \begin{bmatrix} r_{xx}(0) & r_{xx}(1) & \dots & r_{xx}(P) \\ r_{xx}(-1) & r_{xx}(0) & \dots & r_{xx}(P-1) \\ \vdots & \vdots & \ddots & \vdots \\ r_{xx}(-P) & r_{xx}(-P+1) & \dots & r_{xx}(0) \end{bmatrix} \quad (\text{A.9})$$

The linear prediction coefficient (LPC) vector is given as

$$\mathbf{a} = [a_0 \ a_1 \ \dots \ a_P]^T . \quad (\text{A.10})$$

A Toeplitz matrix, such as  $\mathbf{R}_c$ , can be written in a Toeplitz form  $T_c(\mathbf{a}_d)$  which is associated with the LP power spectrum  $f_c(\omega)$  [29]

$$\begin{aligned} T_c(\mathbf{a}_d) &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |A_d(e^{j\omega})|^2 f_c(\omega) d\omega \\ &= \frac{1}{2\pi} \int_{-\pi}^{\pi} |A_d(e^{j\omega})|^2 \frac{\sigma_c^2}{|A_c(e^{j\omega})|^2} d\omega \\ &= \mathbf{a}_d^T \mathbf{R}_c \mathbf{a}_d . \end{aligned} \quad (\text{A.11})$$

Using (A.11), the Itakura-Saito in (A.8) can be written as

$$d_{IS}(x, \hat{x}) = \frac{\mathbf{a}_d^T \mathbf{R}_c \mathbf{a}_d}{\sigma_d^2} + \ln \frac{\sigma_d^2}{\sigma_c^2} - 1 . \quad (\text{A.12})$$

Because [15, p328]

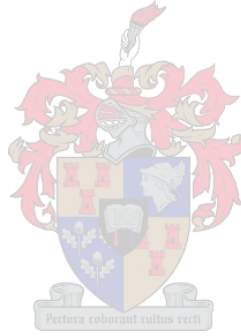
$$\sigma_c^2 = \mathbf{a}_c^T \mathbf{R}_c \mathbf{a}_c \quad \text{and} \quad \sigma_d^2 = \mathbf{a}_d^T \mathbf{R}_d \mathbf{a}_d , \quad (\text{A.13})$$

the Itakura-Saito distortion can also be written as

$$d_{IS}(x, \hat{x}) = \frac{\mathbf{a}_d^T \mathbf{R}_c \mathbf{a}_d}{\mathbf{a}_d^T \mathbf{R}_d \mathbf{a}_d} + \ln \frac{\sigma_d^2}{\sigma_c^2} - 1 , \quad (\text{A.14})$$

or, as given in [33], as

$$d_{IS}(x, \hat{x}) = \begin{bmatrix} \sigma_c^2 \\ \sigma_d^2 \end{bmatrix} \begin{bmatrix} \mathbf{a}_d^T \mathbf{R}_c \mathbf{a}_d \\ \mathbf{a}_c^T \mathbf{R}_c \mathbf{a}_c \end{bmatrix} + \ln \frac{\sigma_d^2}{\sigma_c^2} - 1 . \quad (\text{A.15})$$



# Appendix B

## The TIMIT database

### B.1 The TIMIT192WGN core test set

The TIMIT core test set is defined in [28] to be 192 sentences from the */timit/test/* directory in the database. The data contains sentences from 24 speakers, two male and one female, from each dialect region. The set contains all *si* and *sx* sentences from the speakers shown in Table B.1. When white Gaussian noise is added to these sentences, the data set is referred to as the TIMIT192WGN set.

**Table B.1:** *The TIMIT core test set are 192 selected sentences from /timit/test/. It includes all si and sx sentences from the shown speakers.*

Region	Female	Male	
dr1	felc0	mdab0	mwbt0
dr2	fpas0	mtas1	mwew0
dr3	fpkt0	mjmp0	mlnt0
dr4	fjlm0	mlll0	mtls0
dr5	fnlp0	mbpm0	mklt0
dr6	fmgd0	mcmj0	mjdh0
dr7	fdhc0	mgrt0	mnjm0
dr8	fmlD0	mjlN0	mpam0

## B.2 The TIMIT24WGN training set

The TIMIT24WGN training set used in this study is similar to the TIMIT core test set. It is a smaller set and contains the 24 sentences from the */timit/train/* directory which are shown in Table B.2.

**Table B.2:** *The TIMIT training set are 24 selected sentences from /timit/train/.*

Region	Female	Male	
dr1	fcjf0/si1027	mcpm0/si1194	mdac0/si1261
dr2	faem0/si1392	marc0/si1188	mbjv0/si1247
dr3	falk0/si1086	madc0/si1367	makb0/si1016
dr4	falr0/si1325	maeb0/si1411	marw0/si1276
dr5	fbjl0/si1552	mbgt0/si1341	mchl0/si1347
dr6	fapb0/si1063	mabc0/si1620	majp0/si1074
dr7	fblv0/si1058	madd0/si1295	maeo0/si1326
dr8	fbcg1/si1612	mbcg0/si2217	mbsb0/si1353

## B.3 Informal listening tests



The two sentences used for informal listening tests are read by a male and female speaker from the TIMIT database. They both utter the sentence “She had your dark suit in greasy wash water all year”. The filenames containing the sentences are given below:

- Female: *timit/train/dr1/fcjf0/sa1.wav*
- Male: *timit/train/dr1/mcpm0/sa1.wav*

## B.4 Formal listening tests

For the formal listening tests, an evaluator listens to two different denoised versions of a sentence and then chooses which of the two she prefers. This process, referred to as a *trial*, is repeated for several sentences and evaluators, to improve the statistical significance of the test results.



We use 42 sentences from 14 different speakers taken from the TIMIT192WGN set. The speakers are 7 males and 7 females from dialect regions *dr1* to *dr7*, that are listed in the second and third columns of Table B.1. The 42 sentences used for the formal subjective evaluation consist of the 3 *si* recordings of each of these 14 speakers. We use 24 independent evaluators (18 male and 6 female) to listen to these sentences.

Each trial contains a sentence denoised by two different algorithms, referred to as a *model pair*. In order to keep the test unbiased, it is necessary to evaluate each possible combination of denoising algorithms. Two listening tests were done, as described in Sections 6.8.2 and 7.3.4. The first test compared 6 models (i.e. HMT, HMM, GMM, Wiener, noisy speech and clean speech), which implies 15 possible model pairs. The second test compared 4 models (i.e. HMT, Ephraim-Malah, noisy speech and clean speech), which results in another 6 model pairs.

The ordering of a model pair is relevant. When listening to two consecutive recordings, the last recording tends to have a greater impact on the listener. This potential bias can be removed by including both orderings of each model pair in the test.

Each evaluator therefore performs  $2 \times (15 + 6) = 42$  trials, in random order. For each trial, he listens to the two versions of the sentence (without knowing which two algorithms performed the denoising), and decides which version he prefers. Each pair of models is ultimately evaluated in 48 trials (24 evaluators times 2 orderings per pair). The number of these trials in which a specific model was preferred to the other, is referred to as a *preference count* for that model. The output of the listening test is a set of preference counts, two per model pair.

Since each trial involves a yes/no decision, it is easy to quantify the statistical significance of the test results, based on the binomial distribution [44, p. 52]. If the outputs of the two denoising algorithms in a model pair are really equivalent for listeners, the preference counts associated with this pair ought to be binomially distributed, with  $N = 48$  and  $p = 0.5$ . The probability that the counts lie outside the range [17...31] is less than 5% under this assumption. If a preference count is therefore observed to lie outside this range, the hypothesis of equivalent models is rejected (at a significance level of 5%) and the observed preference is considered to be statistically significant.

The overall ranking of a model can be estimated by summing all the preference counts for that model. This gives an indication of how many times the given model was preferred to the rest of the models in the test. However, it is difficult to assign statistical significance to this ranking, and it should therefore be interpreted as a rough indicator only.