Independent Component Analysis for Environmentally Robust Speech Recognition

Von der Fakultät Elektrotechnik und Informatik

der Technischen Universität Berlin

zur Verleihung des akademischen Grades

Doktor-Ingenieur

genehmigte Dissertation

vorgelegt von Dipl.-Ing. Dorothea Kolossa

Berlin 2008

D 83

Independent Component Analysis for Environmentally Robust Speech Recognition

Von der Fakultät Elektrotechnik und Informatik

der Technischen Universität Berlin

zur Verleihung des akademischen Grades

Doktor-Ingenieur

genehmigte Dissertation

vorgelegt von Dipl.-Ing. Dorothea Kolossa

Promotionsausschuss:

Vorsitzender: Prof. Dr. Klaus-Robert Müller

Gutachter: Prof. Dr.-Ing. Reinhold Orglmeister

Gutachter: Prof. Dr. Te-Won Lee

Tag der wissenschaftlichen Aussprache: 21. Dezember 2007

Berlin 2008

D 83

Acknowledgments

The work on this dissertation was carried out mainly at the TU Berlin, but it has also profited greatly from the visits and collaborations which I have had the chance to participate in. I wish to thank everyone who has been involved in and helped during the last years.

I am grateful to Prof. Orglmeister for the supervision and support and for the many helpful comments and I would also like to extend many thanks to Prof. Lee for accepting the co-supervision and for the detailed suggestions which have helped greatly to improve the thesis. Prof. Klaus-Robert Müller, who has agreed to become head of the committee, has also helped me gain a deepened understanding by his interested and inquisitive questions, and I am thankful to him for this impulse towards improvement.

Also, many discussions have taken place here and much of what I have learned is due to my colleagues. Especially the collaboration with Wolf Baumann and Bert-Uwe Köhler, Ramon Fernandez Astudillo, Alexander Vorwerk and Diep Huynh has helped the dissertation along greatly, and I appreciate their contributions very much.

Likewise, the collaboration with DaimlerChrysler has helped immensely. They supported the data acquisition by providing not only equipment but very much active help in what was a truly time-consuming process, and I am grateful to Dr. Linhard, to Dr. Class and Dr. Bourgeois for the support and for allowing me to use DaimlerChrysler's speech recognition system for the evaluation.

Much support has also been extended to me in the joint work with NTT. I am indebted to Dr. Makino, Dr. Miyoshi, Dr. Sawada, Dr. Araki, Dr. Delcroix, Dr. Nakatani and Dr. Kinoshita for their kind invitation and for letting me stay in their lab and participate in their efforts. I am looking forward to all future collaborations and I am very thankful for the joint work. Very many thanks also go to Prof. Huo at the University of Hong Kong for welcoming me for two visits, during which many helpful discussions have taken place. I have been given many an idea and much motivation during these stays.

Finally, what has made the time spent here a period that I like to think back to are my friends and colleagues. I want to thank especially Steffen Zeiler for the patience and great support, and I wish to thank everyone whom I worked with during the last seven years for making our institute a great and friendly place to be at.

Table of Contents

Chapter 1

Introduction

It has long been a dream of many to be able to speak to a computer and be understood. Whereas this dream will remain in the realm of fantasy for a while, there are some applications which appear worthwile as well as achievable. One of those is speech recognition in car environments, useful, as it may be used to control electronic devices like the phone, radio or navigation system, without needing to take the hands and the eyes off the wheel and the road, and achievable, as the necessary recognition vocabulary is sufficiently limited, which makes recognition far easier than general, large vocabulary applications.

However, the speech is distorted in the car by background noise and it is undesirable to make all drivers wear close-talking microphones, so that it is necessary to cope with noisy speech and with an unknown transfer function from the speaker's mouth to the microphone.

This necessitates signal processing methods, which will remove noise and other possible interference from the speech signal and compensate for changes in the speech signal caused by the room impulse response.

In the following two chapters, the effects of this environment on the speech recognition system will be described. For this purpose, a short introduction to current speech recognition system is given, followed by a mathematical analysis of the effects of noise and reverberation in the car. Subsequently, Chapter 4 describes approaches to make speech recognition systems perform more robustly under these conditions. These approaches can be grouped into three classes:

- Processing the speech signal so as to get a good estimate of the clean speech
- Using speech features in the recognition system, which are invariant under or not strongly affected by noise and reverberation

- Adapting the speech models used in recognition so that these include all disturbances introduced by the environment.

After summarizing the state of the art, here, two routes of improvement are suggested. Firstly, Chapter 5 deals with possible improvements of frequency-domain independent component analysis (ICA) by using statistical models of speech incidence angles as well as by time-frequency masking. However, time-frequency masking alone can actually be detrimental to recognition results, due to inherent feature distortions. In order to avoid these negative effects, Chapter 6 describes a suggested new framework for integrating speech processing and recognition by means of what will be referred to as uncertain feature transformation. These suggested enhancements for processing and recognizing mixed speech signals are tested on two platforms in Chapter 7. Whereas ICA-performance is measured using SNR and recognition rates on a state-of-the-art recognizer provided by DaimlerChrysler, the new method of integrating processing and recognition can be tested only on a platform, where source code is available for modification. Therefore, uncertain feature transformation was evaluated on a Matlab recognizer which was programmed and modified based on the standard algorithms of HMM speech recognition. This recognizer and its results are shown and described also in the evaluation in Chapter 7. Finally, conclusions are drawn in Chapter 8.

Chapter 2

Speech Recognition by Hidden Markov Models

Whereas many approaches for speech recognition have been discussed in the past, currently all major systems show great similarity in the signal processing and pattern matching they perform.[1]

The structure, which is widely used, is shown in Figure 2.1.

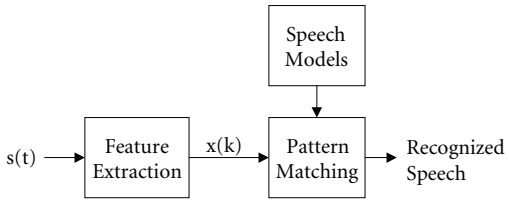

Figure 2.1: Basic recognizer structure.

In the subsequent sections, the stages of feature extraction, speech modeling and pattern matching will be explained in more detail.

2.1 Feature Extraction

The feature extraction stage transforms the speech signal to another representation, which accentuates the relevant characteristics of words or phonemes, while reducing irrelevant or redundant information. Two concepts are found in

[1] Previously discussed approaches include the use of Time Delay Neural Networks [Wai1989], hybrid HMM/Neural-Network approaches ([Mor1995],[Tre2003]) and so-called phonetic speech recognition systems, which detect speech features such as pitch and formant frequencies and use those to segment and label the utterances, finally parsing the segment lattice to obtain the most likely sequence of words [Rab1993].

many relevant representations and they are combined in the currently most widely spread speech representation, so they are described in some detail in the following section. The first of these concepts is production modeling, forming the basis e.g. of linear predictive coding and the second is incorporating models of human perception, which is attempted e.g. in perceptual linear prediction (PLP) and motivates the use of the mel-scale in cepstral speech representations.

2.1.1 Production Modeling

2.1.1.1 Source-Filter Models

A common model of human speech production, as it is described for example in [Rab1978], consists of two alternative sources of sound and one variable filter. The sound sources are either a noise source, which is active in the production of fricatives, or a periodic source, which is used when the speech is voiced. More physiologically speaking, these sources represent the turbulent stream of air that is caused by very narrow passages on the one hand, and the harmonic excitation of the vocal cords on the other hand. In either case, the stream of air has to pass through the vocal tract, where the spectral characteristics are shaped by poles and zeros of the vocal tract transfer function. Finally, further filtering of the signal is due to the radiation characteristics. A block diagram of this model can be seen in Figure 2.2.

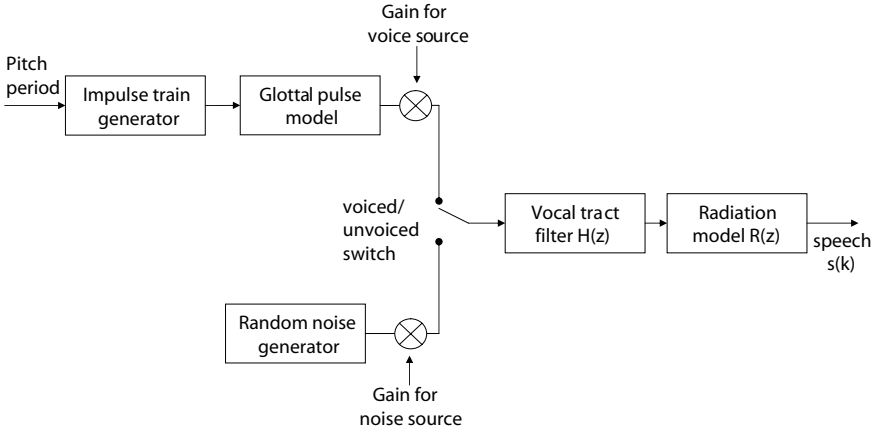

Figure 2.2: Source-filter model for speech production.

2.1.1.2 LPC

The two parts of the production model, lungs and vocal cords as the speech source and the vocal tract as its filter, can be influenced separately and in wide parts independently by the speaker. Linear Predictive Coding (LPC) [Ata1982, Mar1974] attempts to recreate the two distinct influences on the sound by using the following mathematical expression for the model:

$$s(t) = n(t) * h_v(t), \tag{2.1}$$

where h_v stands for the vocal tract impulse response and $n(t)$ represents the excitation signal. Using a discrete time IIR approximation of the system model, as shown in Figure 2.3, an optimum set of FIR predictor coefficients for the FIR prediction structure displayed in 2.4 is determined.

For this purpose, LPC computes the autocorrelation vector $\mathbf{r_{ss}}$, which is defined entrywise for a T-sample segment by

$$\mathbf{r_{ss}}(d) = \sum_{k=0}^{T-1-d} s(k)s(k+d), \tag{2.2}$$

and the autocorrelation matrix given as

$$\mathbf{R_{ss}} = \begin{bmatrix} r_{ss}(0) & r_{ss}(1) & r_{ss}(2) & \dots & r_{ss}(n-1) \\ r_{ss}(1) & r_{ss}(0) & r_{ss}(1) & \dots & r_{ss}(n-2) \\ r_{ss}(2) & r_{ss}(1) & r_{ss}(0) & \dots & r_{ss}(n-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{ss}(n-1) & r_{ss}(n-2) & r_{ss}(n-3) & \dots & r_{ss}(0) \end{bmatrix}.$$

It then finds an optimum set of filter coefficients $\hat{\mathbf{a}}$ by solving

$$\hat{\mathbf{a}} = \mathbf{R_{ss}^{-1}} \mathbf{r_{ss}}. \tag{2.3}$$

Since the autocorrelation matrix is a Toeplitz matrix, efficient methods for its inversion are available and one very efficient algorithm, which uses the Toeplitz structure to find a solution quickly is the Levinson-Durbin Recursion [Kai2000]. This algorithm starts with a first order predictor and in each iteration and with each new piece of data increases the predictor order by one, so that even during initialization, an optimum model of maximum attainable order is always available.

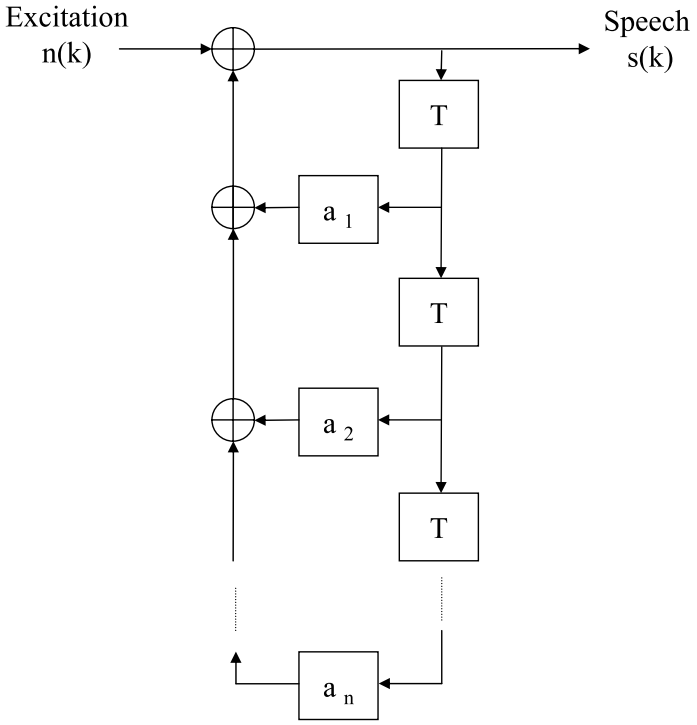

Figure 2.3: Discrete-time IIR Speech Production Model for LPC, with k as the discrete time index.

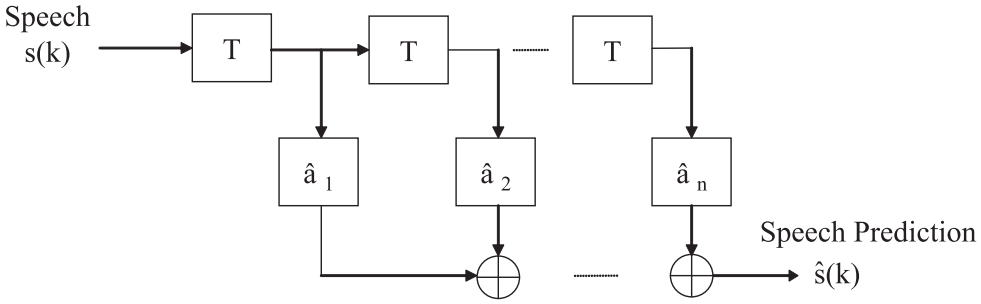

Figure 2.4: FIR Linear Prediction.

2.1.1.3 Homomorphic Signal Processing and the Cepstrum

Similarly to LPC, cepstral processing [Bog1963] also aims to reconstruct a model for speech production based on (2.1). Also similarly to LPC, this model is used in an attempt to separate the vocal tract transfer function, which is crucial for speech recognition, from the excitation signal, which does not carry relevant information for recognition, aside from the binary question of whether it is voiced or not.[2] But furthermore, and in addition to LPC, the cepstrum offers a possibility to separate all components of an extended model

$$x(t) = n(t) * h_v(t) * h_{ir}(t), \tag{2.4}$$

where $h_{ir}(t)$ denotes the room impulse response. It is based on one very popular idea of the 60ies and 70ies, so-called homomorphic signal processing [Opp2004]. The rationale behind it is the wide range of available linear signal processing methods which are easy to apply and in many cases can be shown to yield optimal results (in terms of minimum-mean square error, minimum absolute error, and regarding as many other criteria as may be cast into a linear optimization framework.) The chief limit of applicability for linear models, thus, lies in reality, which does not always behave linearly and the idea of homomorphic processing is to find a transform to another domain of representation, where linearity is re-established. Then, linear processing can be carried out in this linear domain, and, if necessary, the inverse transform can bring the signal back to the original, physical domain. This, it may be argued, is also the approach of convolutive ICA (where a convolutive mixing model is transformed to the approximately linear frequency domain, see the Section 4.2.2.2 and e.g. [Sma1997]). For the case of convolutive signal distortions, that is, when the desired signal $x(t)$ is modulated by an undesired impulse response $h(t)$, a simple, linear superposition of the source and the impulse response representation

[2] This is true for a wide range of languages, with the notable exception of *tonal* languages, where the frequency f_0 of the excitation signal is also distinctive.

is obtained, when the distorted output $y(t) = x(t) * h(t)$ is processed as follows:

$$\begin{aligned} y(t) &= x(t) * h(t) \\ &\circ\!\!-\!\!\bullet \\ Y(\omega) &= X(\omega) \cdot H(\omega) \\ &\Downarrow \log \|\cdot\|^2 \\ Y^l(\omega) &= X^l(\omega) + H^l(\omega) \end{aligned} \tag{2.5, 2.6}$$

Thus, signals which are convolved in the time domain are simply added in the log-spectral domain, and so at first view, the log-spectral domain could be seen as an adequate, homomorphic transform. However, many frequency components of the log-spectrum are affected even when the signal is just superimposed by one reflection. That is, when

$$y(t) = x(t) + \alpha x(t - \tau), \tag{2.7}$$

the log-spectrum is obtained via

$$\begin{aligned} y(t) &= x(t) * (\delta(t) + \alpha\delta(t - \tau)) \qquad (2.8) \\ &\circ\!\!-\!\!\bullet \\ Y(j\omega) &= X(j\omega)(1 + \alpha \exp(-j\omega\tau)) \\ |Y(j\omega)|^2 &= X(j\omega)(1 + \alpha \exp(-j\omega\tau)) \cdot X(-j\omega)(1 + \alpha \exp(j\omega\tau)) \\ &= |X(j\omega)|^2(1 + \alpha^2 + 2\alpha \cos(\omega\tau)) \qquad (2.9) \end{aligned}$$

and, after taking the logarithm, becomes

$$Y^l(\omega) = X^l(\omega) + \log\left[1 + \alpha^2 + 2\alpha \cos(\omega\tau)\right]. \tag{2.10}$$

As can be seen from (2.10), the effect of this convolution is that of adding a periodic disturbance in the log-spectrum domain. Applying another Fourier transform, or equivalently, a discrete cosine transform (DCT) [3], in order to find the periodicities in the log-spectrum will bring the signal to another domain of representation, where the independent variable is in units of time again and the analysis values correspond to periodicities in the spectrum. Thus, the analysis from the above example would

[3] Both are equivalent here, since the log-spectrum is an even function.

lead to a peak at "time delay" τ. A similar situation arises for periodic signals, i.e., when a signal is periodic with pitch period τ_p, the DCT of the log-spectrum will show a peak at τ_p. This form of signal representation, defined via

$$\begin{aligned} Y^c(\tau) &= \Im^{-1}\left(Y^l(\omega)\right) \\ &= \frac{1}{2\pi}\int_{-\pi}^{\pi} Y^l(\omega)e^{j\omega\tau}d\omega \end{aligned} \tag{2.11}$$

and referred to as the signal's *cepstrum*, thus turns out to have three advantages. Firstly, it allows to separate a nonstationary zero-mean source from a stationary filter by subtracting the cepstral mean value :

$$\begin{aligned} Y^c(\tau) &= X^c(\tau) + H^c(\tau) \\ E\left[Y^c(\tau)\right] &= E\left[X^c(\tau)\right] + E\left[H^c(\tau)\right] \\ &= H^c(\tau) \\ \Rightarrow Y^c(\tau) - E\left[Y^c(\tau)\right] &= X^c(\tau) \end{aligned} \tag{2.12}$$

or by including temporal derivatives of the cepstrum as additional features, and secondly, regarding voiced speech segments, periodic excitation signals can be separated from the vocal tract. Thus the cepstral speech representation allows an extended speech production model of the form

$$x(t) = (n(t) + v(t)) * h_v(t) * h_{ir}(t), \tag{2.13}$$

where $v(t)$ represents the voiced excitation, to be separated into its constituent components and especially also to focus on the most relevant one, being the vocal tract response $h_v(t)$. Finally, it has been observed that this set of speech features yields an approximately uncorrelated set of features. Whereas the Fourier coefficients of adjacent bins are highly correlated for speech signals, necessitating the use of full covariance matrices in construction of stochastic models, the cepstral coefficients are only very weakly correlated, so that computationally much simpler diagonal covariance models will often suffice.

2.1.2 Perception of Speech

2.1.2.1 The hearing system

Human hearing takes part in two stages: in the peripheral auditory system (i.e. the ear) and the auditory nervous system (i.e. the auditory nerves and the brain) [Zwi1999]. In the ear, variations in air pressure are transformed into mechanical vibrations on the basilar membrane. These are then represented by impulse trains traveling along the auditory nerve to the brain. While all stages of the processing are relevant for perception, the focus here is on those parts of processing which take place before the auditory nerve is stimulated. The ear consists of three parts, which are the outer ear, the middle and the inner ear, as shown in Figure 2.5. The outer ear consists of the visible part and of the auditory canal along which the sound travels. It ends at the eardrum, where air pressure variations make this thin membrane vibrate. The middle ear is an air filled cavity, in which three small bones, the ossicles called malleus, incus and stapes, transmit and amplify the sound between the eardrum and the cochlea. The transmission to the inner ear takes place at the oval window, a small membrane which can be compressed towards the cochlea by the stapes. In the cochlea, belonging to the inner ear, perception takes place insofar, as the auditory nerve is stimulated at this point. The cochlea itself is a spiral tube of about 3.5 cm length which coils about 2.6 times. It is divided into two fluid-filled chambers by the basilar membrane. The outputs of the cochlea are ordered by location and the location where an incoming sound wave excites the cochlea depends on the frequency, so the cochlea can be said to effect a frequency-to-location transformation. Filters close to the base are excited by high frequencies and lower frequencies affect points further on in the organ. However, the functional dependency between frequency and perceived pitch is not a linear one. Rather, the cochlea acts like a bank of filters whose center frequencies are approximately exponential (that is, the logarithm of the center frequencies behave approximately linear). Also, the critical bandwidth of the filters is not constant, but rather increases with growing frequency. Physiological experiments have led to the Bark scale, which models human perception through 24 critical bands of hearing, with the bark scale defined by

$$b(f) = 13 \arctan(0.00076 f) + 3.5 \arctan(\frac{f}{7500}^2). \tag{2.14}$$

Another perceptually motivated frequency scale is the mel scale, which is approximately linear below 1000Hz and logarithmic above. This scale has been designed

based on experiments with sinusoidal tones, in which the subjects were asked to divide a given frequency range into perceptually equal intervals. The mel scale can be approximated by

$$M(f) \approx 1125 \ln(1 + f/700). \tag{2.15}$$

As shown in Figure 2.6, the mel and the bark scale behave similarly and both

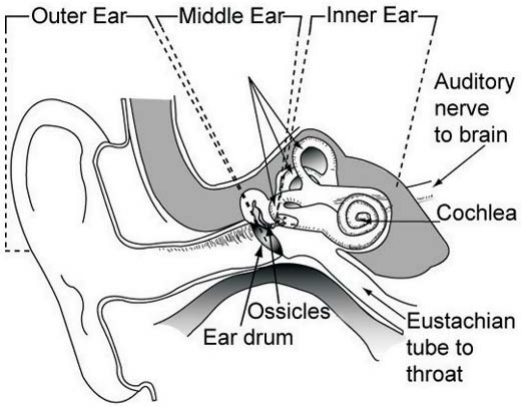

Figure 2.5: Physiology of the ear, from [Cou2002].

have been used extensively for perceptual speech representation and recognition (see. e.g. [Dav1980, Hua2001, Her1990, Syr1986, Yap2003]. Aside from its nonlinear frequency resolution, the human ear and hearing system also contains other non-linearities such as non-linear loudness perception, which also depends non-linearly on frequency [Zwi1999]. These effects are modeled explicitly in some more recent experiments in perceptually motivated speech parametrization, see e.g. [Tch1999, Sko2004], but they were not implemented here, and thus will not be considered in more detail in the following. Also, human perception has inherently an ability to perform grouping between auditory cues in such a way, as to allow even single channel source separation, also referred to as auditory streaming [Bre1999]. Since this can most likely be attributed to higher levels of processing beyond that found in the inner ear itself, it also does not form a part of this chapter, but will rather be considered in more detail in Section 4.2.2.3.

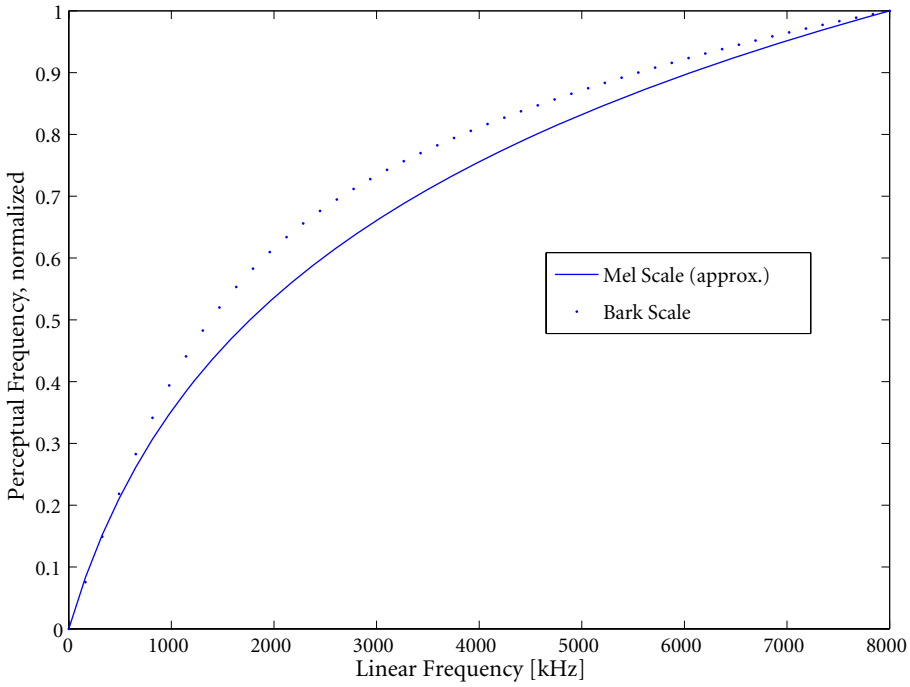

Figure 2.6: Bark and mel frequency scale.

2.1.2.2 Perceptual Linear Prediction

In perceptual linear prediction (PLP), the knowledge about the human hearing system is used to process the recorded waveform in such a way that only the perceptually relevant details remain. This processing is carried out in the frequency domain, where a critical band analysis is followed by equal-loudness preemphasis and intensity to loudness compression [Her1990]. In this way, the warped frequency perception is modeled as well as the nonlinear and frequency-variant human loudness perception. After processing, the signal is transformed back to the time domain, where regular linear prediction analysis is carried out as described above. The block diagram of this highly perceptually motivated feature extraction method is shown in Figure 2.7.

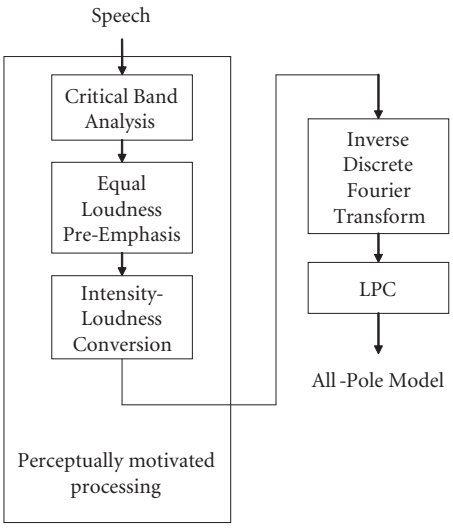

Figure 2.7: Perceptual linear prediction, adapted from [Her1990].

2.1.2.3 Mel-Frequency Cepstrum

While PLP uses perceptual considerations to enhance the LPC parameters, the mel-frequency cepstrum [Dav1980] is an equivalent enhancement of the basic linear cepstrum. Figure 2.8 shows the flow of processing. As can be seen there, the mel-frequency cepstrum and the cepstrum differ only in an additional warping of the frequency scale, which is transformed from linear to mel-scaled by a bank of filters. This bank of filters is just a weighted summation over linear spectral values, with the center frequencies chosen equidistant on the mel-scale defined in (2.15). The summation is carried out as follows.

$$S_{mel}(m,k) = \sum_{\Omega=1}^{nf} H_m(\Omega)|S(\Omega,k)| \tag{2.16}$$

Here, nf denotes the number of frequency bins, $S_{mel}(m,k)$ is the value of m'th bank of the mel spectrum at frame number k and $H_m(\Omega)$ is the weight of frequency band Ω in the calculation of the m'th bank. Literature differs on whether the M filters should have equal peak heights or energies ([Del1993]), but as long as training and recognition take place on the same parameterization, no significant changes can be expected between the two choices. Finally, to compute the filterbank coefficients, it

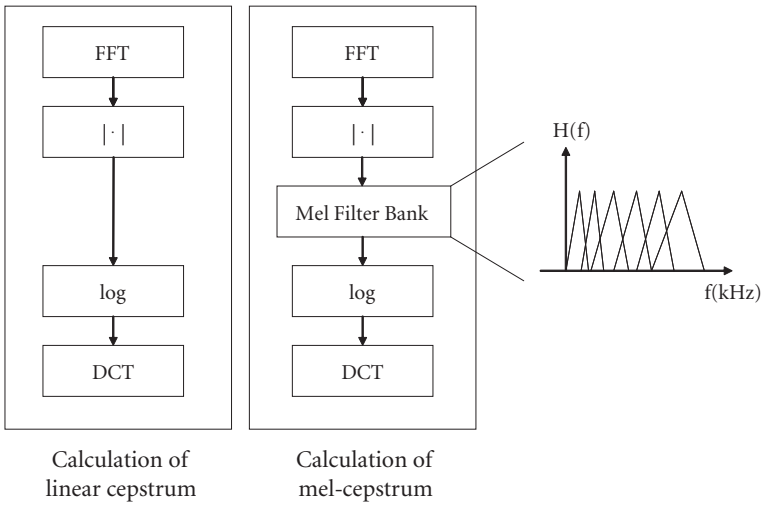

Figure 2.8: Cepstrum versus mel cepstrum.

is typical to use triangular filters with an overlap of 50%. Thus, for the case of equal band energies, the filter coefficients for obtaining M mel-scaled filterbank-outputs can be obtained in the following manner.

For m=1 : M

$$H_m(\Omega) = \begin{cases} 0 & \text{for } 0 \le \Omega \le f(m-1), \\ \frac{2(\Omega - f(m-1))}{(f(m+1)-f(m-1))(f(m)-f(m-1))} & \text{for } f(m-1) < \Omega \le f(m); \\ \frac{2(f(m+1)-\Omega)}{(f(m+1)-f(m-1))(f(m+1)-f(m))} & \text{for } f(m) < \Omega \le f(m+1); \\ 0 & \text{for } f(m+1) < \Omega. \end{cases} \tag{2.17}$$

Here, the frequency $f(m)$ is the FFT-index of the center frequency associated with the m'th band. From these filterbank outputs, finally, cepstrum values (the MFCC's) are obtained via an inverse Fourier transform or, due to symmetry, equivalently via a DCT.

2.1.3 Speech Parameterization for recognition

From different approaches to speech coding, some of which were more intended as production models, others as perceptually motivated representations, a number of different feature sets has come to be suggested for speech recognition purposes. The most important of these are

- Mel Scaled Spectrum
- Linear Prediction Spectrum or equivalently PARCOR-coefficients [Fel1984]
- Linear Prediction Cepstrum [Rab1993]
- Mel Scaled Cepstrum together with its first and second temporal derivative [Dav1980]
- Perceptual Linear Prediction Coefficients [Her1990]

Of these, the mel-scaled cepstrum with its derivatives has emerged as the typical parameterization for current state-of-the-art speech recognition systems [Hua2001, You2002] and even though they are not as well founded in either theoretical properties or closeness to human perception as other parameters, they have proven practical enough to be used as references in current speech recognition standard implementations (e.g. in the basic and as well as advanced versions of the current ETSI standard [ETS2003A] and [ETS2003A]), for recognition tasks at conferences [Hir2002] and currently developed large-scale systems, e.g. by IBM and Daimler-Chrysler [Cla1993, Mac2005, Hai2001] and have therefore been chosen for use in this thesis.

2.2 Speech Models

When the speech features have been extracted from the waveform, they need to be compared to a reference, whatever form this may take. A few years ago, there was decidedly more disagreement on the right form of a speech model, and a few widely differing routes were taken, such as neural networks, especially time delay neural networks (TDNN) [Wai1989], generating speech templates combined with dynamic time warping (DTW) [Mye1981] and knowledge based approaches

focusing on extracting and matching high level phonetic information such as pitch, voicedness, frication and others ([Rab1993]). Currently, the Hidden Markov Model (HMM) has emerged as the primary means for representing the static and dynamic properties that combine to make up the characteristics of a speech sound [Hua2001].

A Markov model is a stochastic process model[4]. In the original mathematical sense it is comprised of a set of states, which a stochastic process may assume, and a probability matrix which defines the probability of the state at the next point in time $p(q(t+1))$ given the current state $q(t)$. The following graph shows the basic structure and introduces the notation.

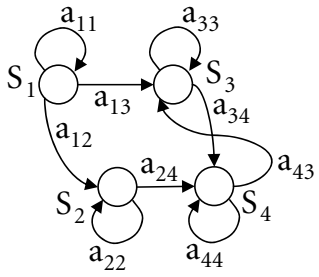

Figure 2.9: A Markov chain with four states and selected transitions.

In a hidden Markov model (HMM), the state of the process is assumed hidden from the observer, but there exists an observable variable **o**, vector valued in general, which can give implicit information about the state of the process. To describe this dependency, each state is assigned a probability distribution function to describe the process output when the process is currently in that state. Thus, mathematically speaking, the defining parameters Λ of a Markov model are:

- The initial state probabilities $a_0(i)$
- The transition matrix A and
- The output probability distributions $b_i(\mathbf{o})$

[4]A stochastic process $\{Q(t), t \in T\}$ is a collection of random variables [Ros1997], where each value describes the state of the process at index t, and t is often interpreted as the time.

The initial state probabilities describe the probability, that the Markov model will start in state i at time zero, so

$$a_0(i) = p(q(0) = i). \tag{2.18}$$

The transition matrix consists of the probabilities that the state j is assumed at time $t + 1$, given that the state at time t was i:

$$a_{ij} = p\left(q\left(t + 1\right) = j | q\left(t\right) = i\right). \tag{2.19}$$

In most speech recognizers, this very general structure is abandoned in favor of pure left-right models as the one shown in Figure 2.10. These models only allow transitions from a state i to states $j \geq i$.

Finally, the output probability distribution $b_i(\mathbf{o})$ gives the probability that observation vector $\mathbf{o}$ will occur at time t, when the Markov model is known to be in state i at that time, so:

$$b_i(\mathbf{o}) = p(\mathbf{o}(t) = \mathbf{o} | q(t) = i). \tag{2.20}$$

A shorthand notation, with $\mathbf{o}_t \stackrel{def}{=} \mathbf{o}(t)$ and $q_t \stackrel{def}{=} q(t)$, leads to the equivalent expression

$$b_i(\mathbf{o}) = p(\mathbf{o}_t = \mathbf{o} | q_t = i). \tag{2.21}$$

This specification of an HMM allows to compute the probability of an observation sequence for any given state sequence by combining transition probabilities a_{ij} and observation probabilities $b(\mathbf{o})$. In the example shown in Figure 2.10, the associated probability of the observation sequence $\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_1, \ldots, \mathbf{o}_5]$ would be computed via

$$p(\mathbf{O}) = a_0(1) \cdot a_{12} \cdot b_2(\mathbf{o}_1) \cdot a_{22} \cdot b_2(\mathbf{o}_2) \cdot a_{23} \cdot b_3(\mathbf{o}_3) \cdot a_{35} \cdot b_5(\mathbf{o}_4) \cdot a_{55} \cdot b_5(\mathbf{o}_5). \tag{2.22}$$

Regarding the output probability distribution, two alternative approaches are viable, one continuous and one discrete. For the first case, a multivariate mixture of Gaussian (MOG) model of the following form is customary for continuous observation vectors. It consists of a set of M single Gaussians

$$p_m(\mathbf{o}_t = \mathbf{o} | q_t = i) = \frac{1}{\sqrt{(2\pi)^n |\Sigma_m|}} \exp((\mathbf{o} - \mu_m)^T \Sigma_m^{-1} (\mathbf{o} - \mu_m)), \tag{2.23}$$

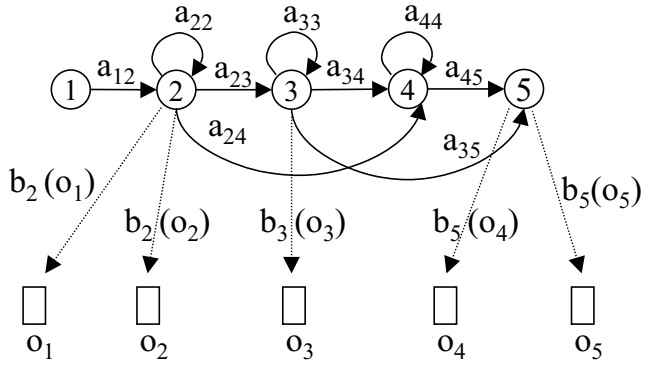

Figure 2.10: Example of a left-right HMM with output probabilities of an observation sequence.

where n is the dimensionality of the observation vector, μ_m is the mean and Σ_m the covariance of the m'th mixture. These single Gaussians are combined in a weighted sum using the mixture weights γ_m:

$$p(\mathbf{o}_t = \mathbf{o}|q_t = i) = \sum_{m=1\ldots}^{M} \gamma_m \cdot \mathrm{p}_m(\mathbf{o}_t = \mathbf{o}|q_t = i). \tag{2.24}$$

An alternative is the use of vector quantization, in which each of the set of possible values is assigned its own probability.

Having defined the model structure, three problems need to be solved in order to use HMMs for speech recognition ([Rab1989]):

- Evaluating the probability of observing a given sequence of speech features $\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_1, \ldots, \mathbf{o}_T]$ given the model parameters (the *evaluation* problem).
- Finding the most likely state sequence $Q = [\mathbf{q}_1, \mathbf{q}_1, \ldots, \mathbf{q}_5]$ according to $\hat{Q} = \arg\max_Q \mathrm{P}(Q|\mathbf{O}, \Lambda)$, given the model and the observation sequence (the *decoding* problem).
- Estimating the model parameters Λ (the *estimation* problem).

The probability of an observation sequence $\mathbf{O}$ is composed of a sum of the probabilities over all possible state sequences of appropriate lengths:

$$\begin{aligned} \mathrm{P}(\mathbf{O}|\Lambda) &= \sum_{\mathbf{Q}} \mathrm{P}(\mathbf{O}, \mathbf{Q}|\Lambda) \\ &= \sum_{\mathbf{Q}} \mathrm{P}(\mathbf{Q}|\Lambda) \cdot \mathrm{P}(\mathbf{O}|\mathbf{Q}, \Lambda) \\ &= \sum_{\mathbf{Q}} a_0(q_0) \prod_{t=1}^{T} a_{qt-1,qt} b_{qt}(\mathbf{o}_t). \end{aligned} \tag{2.25}$$

For its computation, the straightforward implementation is numerically prohibitive, therefore, this evaluation problem is usually handled with a dynamic programming approach especially adapted for this purpose, which is the forward-algorithm.

The decoding problem is usually solved with the help of the Viterbi algorithm as described in Section 2.3 and for the estimation problem, highly efficient maximum likelihood estimation procedures, such as the Baum-Welch algorithm, exist, which compute the model parameters Λ in such a way that $\Lambda = \arg\max_{\lambda} \mathrm{P}(\mathbf{O}_1, \ldots, \mathbf{O}_N|\lambda)$, when N observation sequences are available [Rab1989].

2.3 Pattern Matching

Speech recognition can be viewed as a communication problem: the human brain formulates a word which is transmitted via the speakers articulatory tract and the transmission channel from the mouth to the measured waveform. Recognition then is the search for that word or word sequence W which is most likely, given the observation sequence $\mathbf{O}$.

The Viterbi algorithm [Vit1967] steps through the observation sequence time frame by time frame. At each time it regards all states of the model and computes that path which leads up to the given state and time with the least cost, i.e. with the greatest probability. All other paths except for the one with optimum score are discarded. A trellis diagram serves to illustrate this approach: Thus, the Viterbi algorithm can be used to find the most likely sequence through a given model. This property is used, when a recognizer for continuous speech is needed. In that case,

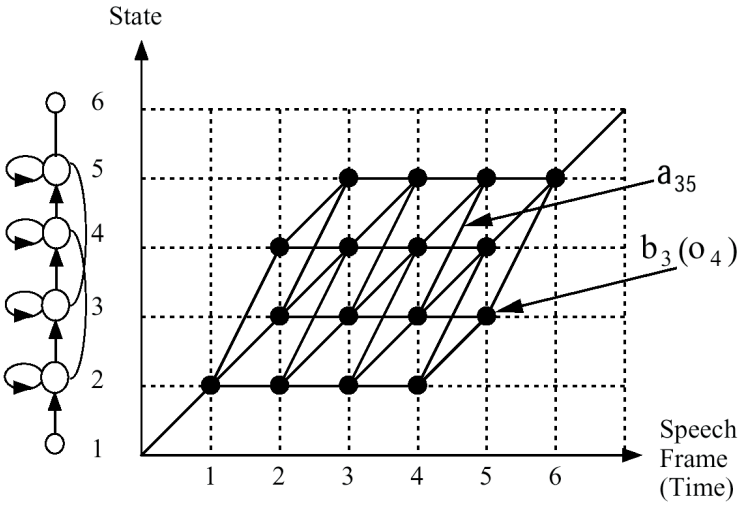

Figure 2.11: Finding the optimum state sequence with the Viterbi algorithm, from [You2002].

the simplest solution consists in modeling single words [5] and concatenating those models as shown in Figure 2.12 to obtain a complete "'speech model"'. Then, for a given utterance, the best path through the speech model is sought and found with the Viterbi algorithm or another dynamic programming approach, and since the annotation for each state is stored with the model, the best state sequence can easily be converted into the desired speech transcription.

[5] For large vocabularies, word models are constructed from phoneme or triphone models.

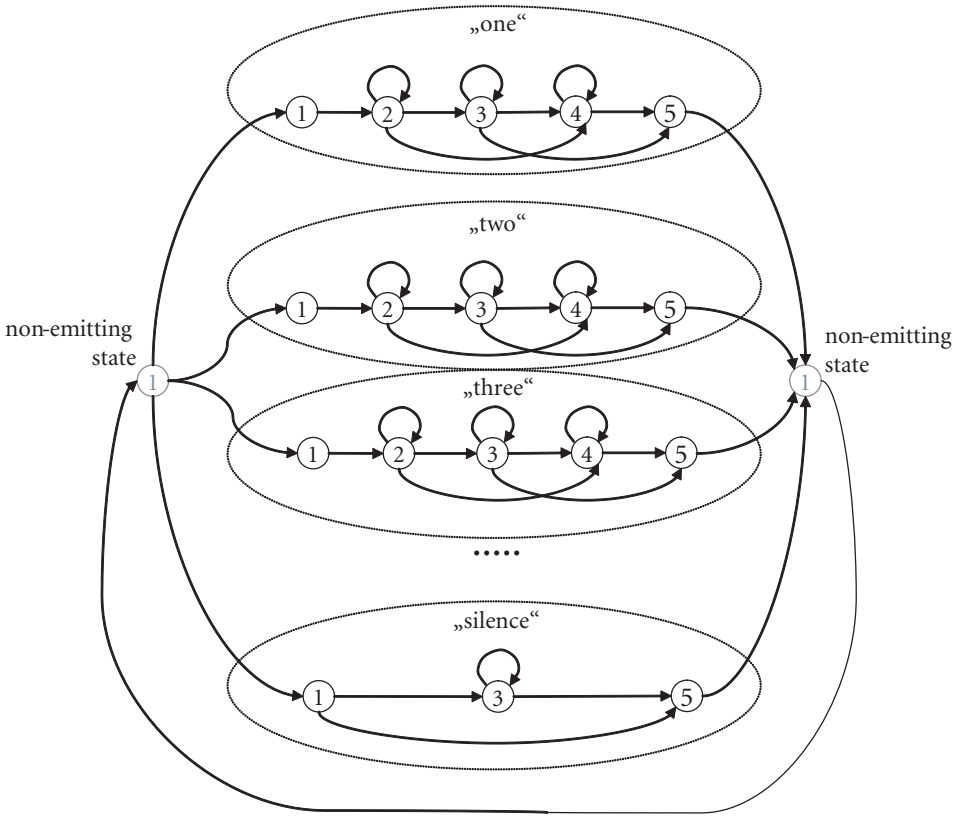

Figure 2.12: Possible HMM structure for recognition of continuously spoken digits.

Chapter 3

Environmental Effects on Speech Signals

3.1 Ideal Model

Speech recorded by one microphone in a noisy environment can be approximately described by the system model shown in Figure 3.1. The source signals are described by the s_i, and they are modified by their associated room impulse responses h_i. As all systems are linear, the sources signals add at the microphone, together with possible sensor noise n, to form the microphone signal x.

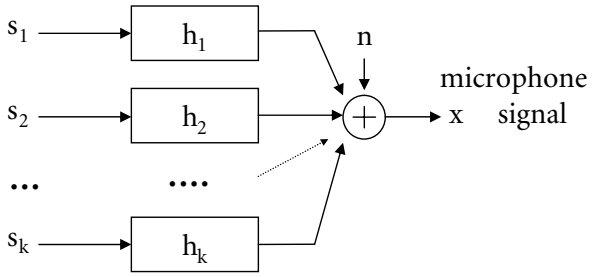

Figure 3.1: Noisy, convolutive mixing model.

This model can be extended to two microphones, where some simplification is achieved by concentrating on one directional interferer only, which is referred to as d, while subsuming all other, directional as well as non-directional, disturbances in the sensor noise terms n_1 and n_2. The thus obtained system model, shown in Figure 3.2, forms the basis of all subsequently described algorithms.

Here, one desired speaker signal, s, is convolved with the room impulse response h_s^i before reaching the sensors. Furthermore, one interfering speaker or

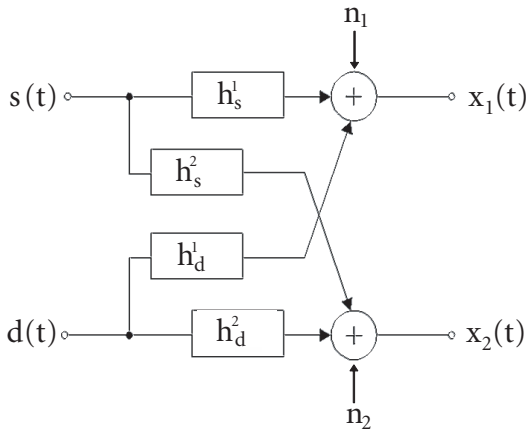

Figure 3.2: Noisy, convolutive mixing model for the case of two directional sources and two microphones.

noise from one localized disturbance source, is also modeled by convolution with a room impulse response, h_d^i. These three kinds of signals, desired and interfering speaker as well as directional noise, may be modeled by independent sources which arrive at the sensors, filtered with their associated room impulse responses. These responses, from origin to sensor i, will be referred to as h_s^i for the desired signal, and h_d^i for interfering speech or direction noise signals, in the following. In addition to the directional sources, whose effect is modeled by a wavefront impinging on the array from the source direction, there is also incoherent noise of two types - sensor noise n_s and the ambient noise field, caused in part by driving, n_a. These two types of noise are subsumed in the sensor noise terms n_i.

3.1.1 Frequency Domain Formulation

Concentrating on just one speaker and one directional interferer d (which may model a speaker or a coherent ambient noise component), source separation can use the model

$$x_i(t) = h_d^i(t) * d(t) + h_s^i(t) * s(t), \tag{3.1}$$

which has an especially simple equivalent in the frequency domain in $X_i(\omega) = H_d^i(\omega) \cdot D(\omega) + H_s^i(\omega) \cdot S(\omega)$. This equation is better known in matrix form, as

in

$$\mathbf{X} = \mathbf{A} \cdot \mathbf{S}, \tag{3.2}$$

where $\mathbf{X} = [X_1(\omega)\ X_2(\omega)]^T$, $\mathbf{S} = [S(\omega)\ D(\omega)]^T$ (so the vector of sources $\mathbf{S}$ consists of two parts, the desired speaker S and the interfering signal D) and

$$\mathbf{A}(\omega) = \begin{bmatrix} H_s^1(\omega) & H_d^1(\omega) \\ H_s^2(\omega) & H_d^2(\omega) \end{bmatrix}. \tag{3.3}$$

This model, with its linear dependence of measured signals $\mathbf{X}$ on the sources S and D, forms the basis of convolutive source separation. In the ideal case that is described by the model, it is possible to blindly identify the so-called mixing matrix A, to invert it and to obtain the desired speech signal S. However, in contrast to the idealized model, real environments, such as cars or office rooms, pose a number of challenges, which are not adequately described by a simple linear model. These are the subject of the following subsections.

3.2 Non-ideal Effects

3.2.1 Long Room Impulse Responses

One of the best-known properties of frequency domain signal representation is its translation of convolutions to multiplications[1]. However, this only holds ideally, when the analysis frame is infinitely long (i.e. for the Fourier Transform.) Speech signals are not stationary and therefore are better described by the short time Fourier transform (STFT), which splits the signal into short, overlapping segments, so-called analysis frames. These analysis frames have a typical length of 10 to 20ms, reflecting the length of approximate stationarity of speech signals [Hua2001]. However, when the impulse response is longer than N, the segment length of the STFT, the effect of the convolution extends over more than one frame of the STFT, with the first part of the impulse response influencing the current signal STFT, the second block of the impulse response describing the effect of the most recent past signal frame on the current observation frame and so forth. When the signal is thus analyzed, the room transfer function effect in the STFT-domain needs to be

[1] In contrast, cepstral representations are "homomorphic" in the ideal case, which means that a time domain convolution will be represented by summation in the cepstral domain.

modeled by a segmentwise convolution between the STFT of the signal and the STFT-coefficients of the impulse response. This is mathematically described by

$$\mathbf{W} \cdot \mathbf{f} \cdot \mathbf{x}(n) = \mathbf{F} \sum_{k=0..K-1} \mathbf{S}(n - kN) \cdot \mathbf{H}_{(k)} \tag{3.4}$$

where W is the DFT matrix composed of $\mathbf{W}_{kl} = \exp(-j2\pi kl/2N)$, $\mathbf{f}$ denotes a diagonal matrix of the window $(f_0, ...f_{N-1})$ defined by $\mathbf{f} = \text{diag}(0, 0, ..., f_0, ..f_{N-1})$, $\mathbf{x}(n)$ is the block of samples $[x(n - N), x(n - N + 1), \ldots, x(n + N - 1)]^T$. $\mathbf{F}$ is a $2N$x$2N$ matrix which contains the DFT coefficients of the zero padded window f, $\mathbf{S}(n - kN)$ is the $2N$x$2N$ diagonal Matrix of DFT-coefficients of the block of input samples $[s(n - kN - N), s(n - kN - N + 1), \ldots, s(n - kN + N - 1)]^T$ and $\mathbf{H}_{(k)}$ is the vector of DFT-coefficients of the k'th segment of the impulse response $[h(kN), h(kN + 1), \ldots, h(kN + N - 1), 0, \ldots, 0]^T$ padded with N zeros [Ser2003]. Thus, in oder to be able to apply (3.2), the framelength of the STFT should be chosen greater than the length of the impulse response. In general, this leads to the best source separation results, but it is not always applicable when ICA is used as preprocessing for speech recognition, since reverberation times can become so long that the computational effort would be prohibitive.

3.2.2 Effects of Ambient and Sensor Noise

In addition to the directed noise component d, ambient noise $n_{i,a}$ and sensor noise $n_{i,s}$ is also measured on all microphones. These noise signals are assumed to be independent on both sensors. The effect of additive noise on the signal (disregarding the directional noise component) can be described in the time domain by:

$$x_i(t) = s(t) * h^i(t) + n_{i,a}(t) + n_{i,s}. \tag{3.5}$$

Combining both ambient and sensor noise into a single term for isotropic noise $n_i(t)$, the power spectral domain representation of the i^{th} sensor signal at the angular frequency ω_k becomes

$$|X_i(\omega_k)|^2 = |S(\omega_k)|^2 \cdot |H^i(\omega_k)|^2 + |N_i(\omega_k)|^2, \tag{3.6}$$

under the assumption of short impulse responses. Since speech recognition systems typically use cepstral features for signal representation, it is interesting to analyze the effect that additive isotropic noise has on the cepstrum of the signal.

Cepstral coefficients are obtained from the spectrum of a signal $X(\omega)$ via

$$\mathbf{x_c} = \mathbf{C}\left[\ln|X(\omega_0)|^2 \ln|X(\omega_1)|^2 \ldots \ln|X(\omega_{M-1})|^2\right]^T \tag{3.7}$$

for a length-M spectral representation of the signal, where $\mathbf{C}$ is the DCT matrix defined by

$$\mathbf{C} = \begin{bmatrix} \cos\left(\frac{0\cdot\pi(1+1/2)}{M}\right) & \cos\left(\frac{0\cdot\pi(2+1/2)}{M}\right) & \ldots & \cos\left(\frac{0\cdot\pi(M-1+1/2)}{M}\right) \\ \cos\left(\frac{\pi(1+1/2)}{M}\right) & \cos\left(\frac{\pi(2+1/2)}{M}\right) & \ldots & \cos\left(\frac{\pi(M-1+1/2)}{M}\right) \\ \vdots & \vdots & \ddots & \vdots \\ \cos\left(\frac{k\pi(1+1/2)}{M}\right) & \cos\left(\frac{k\pi(2+1/2)}{M}\right) & \ldots & \cos\left(\frac{k\pi(M-1+1/2)}{M}\right) \end{bmatrix}. \tag{3.8}$$

and k is the desired number of cepstral coefficients.

Taking the logarithm in equation (3.6) results in

$$\begin{aligned} \ln|X_i(\omega_k)|^2 &= \ln\left(|S(\omega_k)|^2 \cdot |H^i(\omega_k)|^2 \cdot (1 + \frac{|N_i(\omega_k)|^2}{|S(\omega_k)|^2 \cdot |H^i(\omega_k)|^2})\right) \\ &= \ln|S(\omega_k)|^2 + \ln|H^i(\omega_k)|^2 \ldots \\ &+ \ln\left(1 + e^{(\ln|N_i(\omega_k)|^2 - \ln|S_i(\omega_k)|^2 - \ln|H^i(\omega_k)|^2)}\right). \end{aligned} \tag{3.9}$$

Multiplying the entire log spectra with the DCT-Matrix thus yields

$$\mathbf{C}\ln|\mathbf{X_i}|^2 = \mathbf{C}\ln|\mathbf{S}|^2 + \mathbf{C}\ln|\mathbf{H^i}|^2 + \mathbf{C}\ln\left(\mathbf{1} + e^{(\ln|\mathbf{N_i}|^2 - \ln|\mathbf{S_i}|^2 - \ln|\mathbf{H^i}|^2)}\right). \tag{3.10}$$

When the speech features $\mathbf{s_c}$ are denoted by

$$\mathbf{s_c} \stackrel{def}{=} \mathbf{C}\ln|\mathbf{S}|^2 \tag{3.11}$$

and $\mathbf{x_c}$,$\mathbf{h_c}$ and $\mathbf{n_c}$ are defined accordingly, (3.10) can be written somewhat more concisely:

$$\begin{aligned} \mathbf{x_c} &= \mathbf{s_c} + \mathbf{h_c} + \mathbf{C}\ln\left(1 + e^{(\mathbf{C}^{-1}\mathbf{n_c} - \mathbf{C}^{-1}\mathbf{s_c} - \mathbf{C}^{-1}\mathbf{h_c})}\right) \\ &= \mathbf{s_c} + \mathbf{h_c} + \mathbf{C}\ln\left(1 + e^{\mathbf{C}^{-1}(\mathbf{n_c} - \mathbf{s_c} - \mathbf{h_c})}\right). \end{aligned} \tag{3.12}$$

which, combined with equation (3.7), and defining

$$\mathbf{g}(\mathbf{u}) = \mathbf{C}\ln\left(1 + e^{\mathbf{C}^{-1}\mathbf{u}}\right) \tag{3.13}$$

results in

$$\mathbf{x_c} = \mathbf{s_c} + \mathbf{h_c} + \mathbf{g}(\mathbf{n_c} - \mathbf{s_c} - \mathbf{h_c}). \tag{3.14}$$

This equation thus describes the approximate effect of additive noise on the cepstral speech features. In practice, the matrix $\mathbf{C}$ is not usually square, since the higher cepstral coefficients are irrelevant for the purpose of speech recognition. Thus, Equation (3.12) can only be calculated approximately via the pseudo inverse of $\mathbf{C}$. This results in some smoothing and approximation in the spectral domain, but it has been shown to work reasonably well in practice [Hua2001]. As can be seen from Eq. (3.14), the effect of noise on the cepstrum is a highly non-linear one. This firstly motivates using denoising and source separation in the frequency domain, where linear methods are available, secondly, it also shows that any method seeking cepstral representations from uncertain values in the spectrum will need to be able to deal with non-linearities in order to succeed.

Chapter 4

Robust Speech Recognition

Approaches to robust speech recognition, generally speaking, have four points at which they may attempt improvements:

- Finding more robust speech features for better representation of the speech signal,
- cleaning the speech signal from interference,
- adapting recognition models online or offline, so as to capture the speech characteristics in the given situation and finally
- modifying the recognition process itself to include information about feature uncertainty.

A short overview of these approaches will be given in the following sections.

4.1 Robust Features

4.1.1 Mel Frequency Cepstral Coefficients

The currently most widely used representation of speech signals in the area of speech recognition is the mel-frequency domain cepstrum with its associated time derivatives, sometimes in conjunction with the signal's log-energy. A reason why it has been widely accepted is its relatively good performance, which may be due to its concentration on perceptually relevant features [Dav1980]. Also, high robustness can be attained with respect to environmental changes and to the speaker by adequate preprocessing. For attaining more robustness in these two respects, vocal-tract length normalization can help to normalize the characteristics of different speakers and cepstral mean normalization can compensate for changes in the room impulse response.

4.1.1.1 Vocal Tract Length Normalization

Different speakers are distinguishable especially by different shapes of their vocal tracts, which shift the resonances and thus the formant frequencies of their speech. In order to compensate for such shifts, a transformation of the frequency axis can be used. Often, it is linear (e.g. [Gar2005]) or piecewise linear ([Zha1997, You2002]), as shown in Figure 4.1. The depicted frequency warping is mathemati-

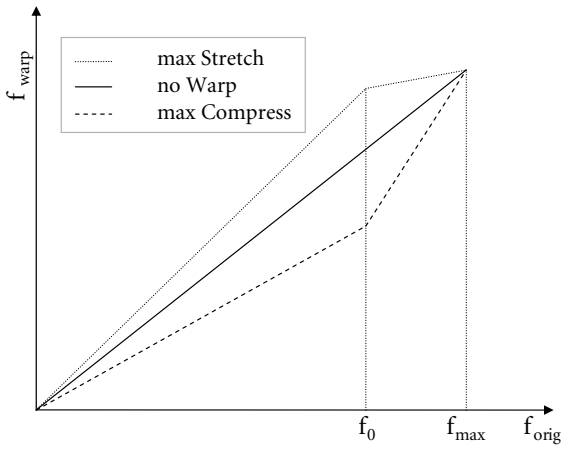

Figure 4.1: Piecewise linear frequency warping curves for vocal tract length normalization.

cally formulated and implemented as

$$f_{warp} = \begin{cases} \alpha^{-1} f_{orig} & \text{for } f_{orig} \leq f_0, \\ b f_{orig} + c & \text{for } f_{orig} > f_0, \end{cases} \tag{4.1}$$

where α is the warping factor and b and c are determined from α and f_0. As vocal tract length normalization thus amounts to a linear or nonlinear transformation of the frequency axis, this method of improving robustness is actually available for all spectrum-based speech parameterizations. In this way, the speaker dependence of a system can be reduced significantly ([Zha1997]).

4.1.1.2 Cepstral Mean Subtraction

Cepstral features are obtained by taking the logarithm of spectral features and subsequently applying a DCT, as described in Section 3.2.2:

$$\mathbf{x}_{cep}(\tau) = DCT\left(\log(|\mathbf{X}(\omega)|^2)\right). \tag{4.2}$$

Therefore, convolutions with the room impulse response, which have a multiplicative effect in the frequency domain, now become additive disturbances. As shown on page 9, a separation of the speech signal from the influence of the room impulse response thus becomes possible at least in an approximate sense by subtracting the mean value of the cepstrum from all cepstral features frame by frame via

$$\mathbf{x}_{cms}(\tau, k) \quad = \quad \mathbf{x}_{cep}(\tau, k) - \overline{\mathbf{x}}_{cep}(\tau), \tag{4.3}$$

with k denoting the frame index. Thus, the mismatch caused by the channel from mouth to microphone will be compensated at least under stationary conditions. Real time implementations of this method replace the subtraction of the cepstral mean $\overline{\mathbf{x}}_{cep}(\tau)$ itself by an adaptive estimate $\hat{\mathbf{x}}_{cep}(\tau, k)$, as in

$$\hat{\mathbf{x}}_k = \alpha \mathbf{x}_{cep}(\tau, k) + (1 - \alpha)\hat{\mathbf{x}}_{cep}(\tau, k - 1). \tag{4.4}$$

This is equivalent to a high pass filtering of the cepstral features, and thus also related to the so called *RASTA methods* [Hua2001] described below.

4.1.2 RASTA Processing

RASTA stands for "relative spectral" processing. The basic idea is to utilize the fact that speech usually changes at a different rate than environmental conditions such as background noise or the room impulse response. Thus, using band-pass filters for each frequency component, only those parts of the spectrum are retained, whose rate-of-change is in the same range as the rate-of-change of typical vocal tract parameters. This enables RASTA processing, introduced by Hermansky in [Her1994], to treat changed room and channel conditions as well as additive or convolutional background noise within the same framework. In principle, this also works by filtering of the speech features, but whereas cepstral mean subtraction

filters the cepstrum according to (4.3), RASTA processing utilizes more complex bandpass filters on the speech spectrum, e.g.

$$X_{RASTA}(\Omega,k) = \quad 0.2\cdot X(\Omega,k) + 0.1\cdot X(\Omega,k-1) - 0.1\cdot X(\Omega,k-3) - \\ 0.2\cdot X(\Omega,k-4) + 0.98\cdot X_{RASTA}(\Omega,k-1) \tag{4.5}$$

in the version described in [Her1994]. However, since the method is tuned specifically to exclude non-speech signals and retain speech components, it is expected to perform poorly for speaker-speaker separation and was therefore not considered for the problem at hand.

4.1.3 Long term features

The core operations used in deriving a speech representation for recognition have essentially remained unchanged over the last decades. They consist of computing a speech feature vector in some way from the spectral envelope of the signal over frames of around 20-30ms and with a frame shift of around 10ms [Mor2005]. However, human phoneme recognition is significantly degraded over such short segments, which has motivated experiments on longer term representations. These can be, for example, low-dimensional feature projections derived from a number of frames [Hae1994, Hun1989], Gabor features [Kle2002] with their multiscale characteristics, long term evolution of critical band energies [Che2004] or 2D features resulting from the application of linear prediction over both frequency and time to model time-frequency peaks accurately [Ath2004]. However, even though there exist some promising results for incorporating long term features into the recognition process, they are not widely used at the moment, which may be due to their need for a new statistical framework, possibly based on neural networks or more complex graphical models than customary HMMs [Mor2005].

4.1.4 Auditory based features

Another idea which is currently under much discussion is how to tune speech recognizers more specifically to those features which are dominant in human perception. This appears attractive, since human learning of speaking is also tuned to perceptible features so that speech can be expected to vary little in terms of the

characteristics which are accessible to human hearing. Therefore, the use of hearing models for robust speech recognition has been proposed e.g. in [Den2004, Lee2003, Hem2004, Mak2004, Tch1999, Wan2001]. However, at the moment, speech recognition rates of large vocabulary, speaker independent MFCC-based systems are not yet matched by other, auditory based features, so this route was also not pursued further for the current purpose, which is to obtain best possible recognition rates under strong and quickly varying directional interference.

4.2 Signal Preprocessing

It is convenient to distinguish single-channel and multi-channel techniques for speech processing. Regarding single-channel techniques, the following sections will show how spectral subtraction and related techniques process the signal based on simple models of the noise signal obtained in pauses of speech activity [McA1980, Eph1984]. These methods, described in 4.2.1.1 and 4.2.1.2, generally rely on the noise being at least sufficiently stationary to remain approximately the same during periods of speech. In contrast, the so-called model-based approaches, e.g. by Deng [Den2001], by Attias [Att1998], by Benaroya [Ben2003] and by Seltzer and Stern [Sel2003], discussed in 4.2.1.3, are capable of handling instationary noise also, but for this purpose they have to rely on the availability of an additional speech model.

Regarding multi-microphone signal processing, it is not easy to find a classification mechanism distinguishing ICA methods and beamforming. Still, these areas will be treated in separate sections. Here, all those adaptive, multi-microphone methods will be described as ICA, which rely on simultaneous diagonalization of cross-statistics or directly on information theoretic criteria to obtain speech signal estimates. In contrast, the section on beamforming subsumes systems with fixed directional characteristics and those adaptive methods which rely solely on second order statistics to adapt their directional characteristics.

Finally, after ICA and beamforming, the section also contains a short overview of nonlinear source separation, the so-called *CASA*-methods which have recently emerged as an alternative in two-channel, and possibly underdetermined, source separation. These methods rely on spatial, temporal or time-frequency characteristics of speech and noise signals to design an optimal one/zero-mask or smooth

mask, which ideally retains all speech-dominated time-frequency bins and masks all interferer bins.

4.2.1 Single-Microphone Techniques

For stationary noise environments, it is possible to obtain an estimate of the noise spectrum during speech pauses and to process subsequent segments of speech in principle by subtracting the noise spectrum, as described in the following two sections. For further improvements, a dynamic model of speech or noise signals is a prerequisite, which is described in more detail in Section 4.2.1.3.

4.2.1.1 Spectral Subtraction

Spectral subtraction deals with additive, stationary noise, see e.g. [Bol1979, Koe2005]. It is always necessary to assume that an estimate of the noise spectrum $N(\omega)$ is available, be it from a reference microphone, from a reliable voice activity detection algorithm, or from using an online capable statistical noise estimation method such as the minimum statistics described in [Mar1994] and [Coh2002]. When statistical independence of speech and noise spectrum is assumed, this results in

$$E(|X(\omega)|^2) = E(|S(\omega)|^2) + E(|N(\omega)|^2), \tag{4.6}$$

and the speech spectrum can be obtained from

$$E(|S(\omega)|^2) = E(|X(\omega)|^2) - E(|N(\omega)|^2). \tag{4.7}$$

This can also be written as

$$E(|S(\omega)|^2) = E(|X(\omega)|^2)(1 - \frac{E(|N(\omega)|^2}{E(|X(\omega)|^2)}). \tag{4.8}$$

For implementation purposes, expectations of the signals are generally replaced by instantaneous values and the noise value must be replaced by its estimate, which is often taken to be the long term average $\overline{N}(\omega)$, resulting in

$$|\hat{S}(\omega, k)|^2 = |X(\omega, k)|^2(1 - \frac{|\overline{N}(\omega)|^2}{|X(\omega, k)|^2}). \tag{4.9}$$

A general problem of spectral subtraction is *musical noise*, which can have a degrading effect on recognition performance. Using minimum mean square error (MMSE) estimation for noise suppression alleviates this effect, as described in the following Section 4.2.1.2. Alternatively, rather than resorting to MMSE estimation, it is also possible to improve spectral subtraction by choosing

$$|\hat{S}(\omega,k)|^2 = |X(\omega,k)|^2 - \alpha|\hat{N}(\omega,k)|^2 \tag{4.10}$$

with $\alpha > 1$ and $\hat{N}(\omega,k)$ as the estimated noise spectrum, and by compensating any physically impossible negative values. This approach, called *oversubtraction*, reduces the residual peaks responsible for musical noise. The basic version of spectral subtraction, with $\alpha = 1$, can be shown to yield a maximum likelihood estimate of the signal's Fourier coefficients, when a Gaussian distribution is assumed for the complex Fourier coefficients of speech and noise [McA1980]. However, spectral subtraction cannot be assumed to be applicable for multi-speaker scenarios due to its inherent inability to cope with multiple microphones and instationary noise environments.

4.2.1.2 Optimum Spectral Estimation

Statistical estimation can be used to obtain a speech signal estimate, which is optimal with respect to some cost function. In many cases, this approach results in a nonlinear gain function $G(\omega,k)$, which is applied to the noisy signal spectrum $|X(\omega,k)|^2$ in the form

$$\hat{S}(\omega,k) = G(\omega,k) \cdot X(\omega,k) \tag{4.11}$$

to yield the speech signal estimate $\hat{S}(\omega,k)$.

In maximum likelihood estimation, for this purpose, the clean speech spectrum $S(\omega,k)$ is parameterized via

$$S(\omega,k) = A(\omega,k)\exp^{-j\alpha(\omega,k)} \tag{4.12}$$

and a maximum likelihood estimate $\hat{A}$ is obtained from a Gaussian noise model and is given by

$$\hat{A}(\omega,k) = \frac{1}{2}\Big[|X(\omega,k)| + \sqrt{|X(\omega,k)|^2 - \sigma_N^2}\Big], \tag{4.13}$$

where σ_N^2 corresponds to the noise variance [Koe2005].

Similarly, the Ephraim-Malah filter operates on the amplitude spectrum of the measured signals. In this domain, it gives a minimum mean square error estimate of the speech amplitude, based on modeling both speech and noise as Gaussian distributed with means $\mu_S(\Omega), \mu_N(\Omega)$ and variances $\sigma_S(\Omega), \sigma_N(\Omega)$. The variances are estimated online for each frequency bin Ω, whereas the mean values are assumed to be zero. Defining variables u_N for the real part and v_N for the imaginary part of the noise Fourier coefficients (and u_S and v_S, respectively, for the speech signal) gives a joint distribution

$$\mathrm{p}(u_N, v_N) = \mathcal{N}(0, \sigma_N^2/2) \cdot \mathcal{N}(0, \sigma_N^2/2) = \frac{1}{\pi\sigma_N^2} \exp\left(-\frac{u_N^2 + v_N^2}{\sigma_N^2}\right). \tag{4.14}$$

This can be transformed to polar coordinates with magnitude A_N and phase α_N according to

$$\mathrm{d}u_N \mathrm{d}v_N = A_N \mathrm{d}A_N \mathrm{d}\alpha_N, \tag{4.15}$$

yielding [Koe2005]

$$\mathrm{p}(A_N, \alpha_N) = \frac{A_N}{\pi\sigma_N^2} \exp\left(-\frac{A_N^2}{\sigma_N^2}\right) \tag{4.16}$$

for the noise signal and a similar equation for the speech magnitude and phase. Furthermore, from (4.14) the following distribution of the noisy signal Fourier coefficients X_Ω is obtained [Eph1984]:

$$\mathrm{p}(X_\Omega | A_\Omega, \alpha_\Omega) = \frac{1}{\pi\sigma_N(\Omega)^2} \exp\left(-\frac{|X_\Omega - A_\Omega \exp(j\alpha_\Omega)|^2}{\sigma_N^2(\Omega)}\right). \tag{4.17}$$

with A_Ω and α_Ω denoting the clean signal magnitude and phase in frequency bin Ω. Minimum mean square error estimation leads to $\hat{A_\Omega} = E(A_\Omega | X_\Omega)$, which can be computed analytically using Bayes' law and integration of the conditional probabilities $\mathrm{p}(X_\Omega | A_\Omega, \alpha_\Omega)$ over all A_Ω and α_Ω [Eph1984]. In this way, a nonlinear gain function $G_{MMSE}(\Omega)$ is derived, which is used as shown in (4.11).

As an alternative to MMSE estimation, an optimal estimate may also be obtained by noting that the speech covariance matrix has distinct eigenvectors with pronounced eigenvalues, which allows decomposition of a noisy signal into a speech and a noise subspace, as derived in [Eph1995]. As a further alternative, maximum a posteriori (MAP) estimation rather than MMSE estimation has also been

used in the power spectrum domain, which allows incorporation of prior information about the noise power spectrum [Jia2003]. However, a disadvantage of the original Ephraim-Malah-filter, as well as of the more recent subspace and MAP estimation methods, is their reliance on time or spectral domain speech models. Alternatively, it is also possible to use optimum estimation in the log-spectrum domain via $\log \hat{A} = E(\log A|X)$ as described in [Eph1985]. What all these methods have in common, however, is their reliance on a noise spectrum estimate. Recently, methods for estimation of noise even during speech presence have been used increasingly to cope with this disadvantage [Coh2001]. However, noise from such sources as interfering speakers is still too instationary to differentiate it from the desired signal, thus the above methods cannot be expected to be successful for the task of speaker-speaker separation.

4.2.1.3 Model Based Processing

In this situation, is seems reasonable to use as much information as is available, and especially for speech recognition, where a good model of the speech signal is needed in any case, model based preprocessing has been applied in many cases.

State-Based Filtering

In a simple manner, a way to use speech model information is given by the Wiener filtering equations. When a speech HMM is available, each state of the HMM can be associated with its average speech autocorrelation function, where autocorrelations $\mathbf{r}_i(\tau)$ for HMM-state i are obtained after HMM-training by

- estimating the time-/state-alignment of the training dataset to the HMM using the Viterbi algorithm
- and for each state i computing the autocorrelation of the sequences which have been aligned with this state.

From this procedure, each HMM-state i is then equipped with knowledge about the associated speech autocorrelation function $\mathbf{r}_i(\tau)$. Thus, once an estimate of the noisy speech autocorrelation matrix $\mathbf{R}_x$ is available, a Wiener filter for each of the

HMM states $i = 1 \ldots N$ can be calculated using

$$\mathbf{R}_x \mathbf{h}_{opt,i} = \mathbf{r}_i \tag{4.18}$$

and it can then be used during recognition via filtering of the input signal, separately for each of the HMM states [Bea1991]. Thus, each of the states gets its own, optimum observation estimate $\hat{o}_i(t)$ for recognition, which is obtained by its own associated Wiener filter

$$\hat{o}_i(t) = o(t) * \mathbf{h}_{opt,i}. \tag{4.19}$$

As can be seen from Equation (4.19), filtering is carried out here on the time domain signal. The alternative of computing MMSE filers in the feature domain (e.g. the cepstrum domain) is also viable, but it requires a significant increase in complexity, since here, the combination function for speech and noise is nonlinear, as shown in Equation (3.14) for the case of cepstral features. This problem must be dealt with by techniques such as linearization[1], but is highly successful also in nonstationary noise environments [Cou2000].

Model-Based Optimum Estimation

However, the approach of state-based filtering only works well, when the number of states is not too large, so that separate filtering and also separate design of a Wiener filter for each state and in the presence of each new noise signal is practical. Relaxing this requirement, another way to go about integrating prior knowledge is via optimum estimation techniques, similar to the above discussed methods of minimum mean square error estimation. In this framework, the clean speech log-spectrum $\mathbf{S}_l$ can be estimated from the noisy spectrum $\mathbf{X}_l$ via

$$\begin{aligned} \hat{\mathbf{S}}_{l,MMSE} &= \mathrm{E}(\mathbf{S}_l|\mathbf{X}_l) && (4.20) \\ &= \mathbf{X}_l - \mathrm{E}(\mathbf{X}_l - \mathbf{S}_l|\mathbf{X}_l). && (4.21) \end{aligned}$$

The term $\mathrm{E}(\mathbf{X}_l - \mathbf{S}_l|\mathbf{X}_l)$ describes the difference between noisy and clean spectrum that is due to noise $\mathbf{N}_l$. Since the log spectrum changes nonlinearly when noise

[1] For example, the vector Taylor series can be used.

is added, it is necessary to model the nonlinearity explicitly to obtain the MMSE estimate via

$$\hat{\mathbf{S}}_{l,MMSE} = \mathbf{X}_l - \int_{\mathbf{S}_l} g(\mathbf{S}_l, \mathbf{N}_l, \mathbf{H}_l) \mathrm{p}(\mathbf{S}_l|\mathbf{X}_l) \mathrm{d}\mathbf{S}_l. \tag{4.22}$$

Here g stands for the environmental model

$$g(s, n, h) = \mathbf{H}_l + 10 \log_{10}(\mathbf{1} + 10^{\mathbf{N}_l - \mathbf{S}_l - \mathbf{H}_l}) \tag{4.23}$$

as described in [Mor1996] and the noise and channel models $\mathbf{N}_l$ and $\mathbf{H}_l$ are learned as the hidden variables of an EM algorithm. This environmental model is the log-spectrum equivalent of (3.13) and (3.14), which are derived in detail in Chapter 3. Using a vector Taylor series approximation of (4.23), MMSE estimation can then be carried out using the log-spectrum domain speech model (a standard Gaussian mixture model)

$$\mathrm{p}(\mathbf{S}_l) = \sum_{m=0}^{M-1} p_k \mathcal{N}(\mathbf{S}_l, \mu_m, \mathbf{\Sigma}_m) \tag{4.24}$$

in the MMSE framework via

$$\hat{\mathbf{S}}_{l,MMSE} = \mathbf{X}_l - \sum_{m=0}^{M-1} \mathrm{P}(k|\mathbf{X}_l) g(\mu_m, \mathbf{N}_l, \mathbf{H}_l). \tag{4.25}$$

But while this method is successful in improved estimation of log-spectrum speech, a cepstrum domain estimator is often even more desirable, since the cepstrum is a major domain for speech recognition applications. One such algorithm that is implemented directly in the cepstral domain is the so called "SPLICE-algorithm" [2] [Den2001]. This algorithm uses stereo training data[3] to learn a joint probability distribution of the clean and the distorted signal. A theoretically plausible way to formulate this joint probability distribution $\mathrm{p}(\mathbf{s_c}, \mathbf{x_c})$ would be:

$$\mathrm{p}(\mathbf{s_c}, \mathbf{x_c}) = \mathrm{p}(\mathbf{s_c}|\mathbf{x_c}) \mathrm{p}(\mathbf{x_c}). \tag{4.26}$$

But since the speech cepstrum has a nonlinear dependence on the observed cepstrum $\mathbf{x_c}$, this learning task is ill-posed. Therefore, a piecewise linear approximation of the joint probability is obtained via introducing an auxiliary variable i. This

[2] SPLICE: Stereo-Based Piecewise Linear Compensation for Environments

[3] Stereo data containing one channel of clean and one of distorted speech

variable partitions the space of observations into regions, in which the clean and the distorted speech have a linear dependence. Thus, it is possible to write:

$$p(\mathbf{s_c}, \mathbf{x_c}) = \sum_i p(\mathbf{s_c}|\mathbf{x_c}, i)p(\mathbf{x_c}|i)p(i). \tag{4.27}$$

For each of the partitions i, the dependence of the clean speech on the observed signal is

$$\mathbf{s_c} = \mathbf{x_c} + \mathbf{r_i}. \tag{4.28}$$

where $\mathbf{r_i}$ is the cepstral correction vector for partition i. One additional assumption - namely, the Gaussianity of the cepstral coefficients, allows calculating the probability distribution of the clean conditional on the noisy speech:

$$p(\mathbf{s_c}|\mathbf{x_c}) = \sum_i \mathcal{N}(\mathbf{s_c}, \mathbf{x_c} + \mathbf{r_i}, \Gamma_i)p(i) \tag{4.29}$$

The minimum mean square error estimate of a random variable is its conditional mean [Vas2001]. Therefore, $\mathbf{s_c}$ is estimated via:

$$\begin{aligned}
\hat{\mathbf{s}}_c &= \mathrm{E}(\mathbf{s_c}|\mathbf{x_c}) && (4.30)\\
&= \sum_i \mathrm{E}\left(\mathcal{N}(\mathbf{s_c}, \mathbf{x_c} + \mathbf{r_i}, \Gamma_i)p(i)|\mathbf{x_c}\right) && (4.31)\\
&= \sum_i (\mathbf{x_c} + \mathbf{r_i})p(i|\mathbf{x_c}) && (4.32)\\
&= \mathbf{x_c} + \sum_i \mathbf{r_i}p(i|\mathbf{x_c}). && (4.33)
\end{aligned}$$

Thus, it is possible to obtain a minimum mean square error estimate, as soon as the noise correction vectors $\mathbf{r_i}$ are known and once the partition probabilities $p(i|\mathbf{x_c})$ have been determined. Both these quantities are trained, respectively calculated, from stereo training data in which one dataset contains the clean and the other the noisy speech. To find the correction vectors, the maximum likelihood criterion is used [Dro2002]. For this purpose, the expected value of the joint likelihood of $\mathbf{x_c}$ and $\mathbf{s_c}$ given $\mathbf{r_c}$ can be maximized with respect to $\mathbf{r_c}$, which leads to

$$\mathbf{r_i} = \frac{\sum_k p(i|\mathbf{x_c}(k))(\mathbf{s_c}(k) - \mathbf{x_c}(k))}{\sum_k p(i|\mathbf{x_c}(k))}. \tag{4.34}$$

The noise type probability here can be calculated via Bayes' Rule:

$$\mathrm{p}(i|\mathbf{x_c}(k)) = \frac{\mathrm{p}(i, \mathbf{x_c}(k))}{\mathrm{p}(\mathbf{x_c}(k))} = \frac{\mathrm{p}(i)\mathrm{p}(\mathbf{x_c}(k)|i)}{\sum_i \mathrm{p}(i)\mathrm{p}(\mathbf{x_c}(k)|i)}. \tag{4.35}$$

Whereas this kind of speech processing requires a great amount of work in the training phase, as noise correction vectors must be learned for as many kinds of noise and distortion as possible, it is highly successful in applications with additive noise and convolutional distortions. However, it is not a good candidate for multi-speaker environments, since it would be necessary to obtain training data for all possible kinds of interfering sounds and in all kinds of convolutive environments, which makes the approach impractical for this scenario.

4.2.2 Multi-Microphone Techniques

4.2.2.1 Beamforming

Microphone arrays have been used "traditionally" for source separation. Their use preceded information theoretic methods by nearly a quarter of a century (see e.g. [Gri1969]). The main difference between ICA methods and "classical" beamforming lies in the fact, that neither the array geometry nor the position of the speaker relative to the array is required for ICA methods, whereas array methods in their original, non-adaptive formulation, do need this information for the design. Adaptive beamforming methods can also infer the speaker position, as ICA methods do, however, they still require advance knowledge regarding the array geometry.

Propagation Models for Beamforming

In general, beamformers rely on a model of signal generation where the source signal $s(t)$ propagates as an acoustic wave from the source to the sensor. Propagation can take place along a circular wavefront in the nearfield or along straight line in the farfield model. Both these models are shown in Figure 4.2.

In the farfield model, which is commonly used for source-to-microphone distances greater than $2L^2/\lambda$, plane wave propagation is often assumed.[4] For a

[4] L is the largest array dimension, see e.g. [Ken1998].

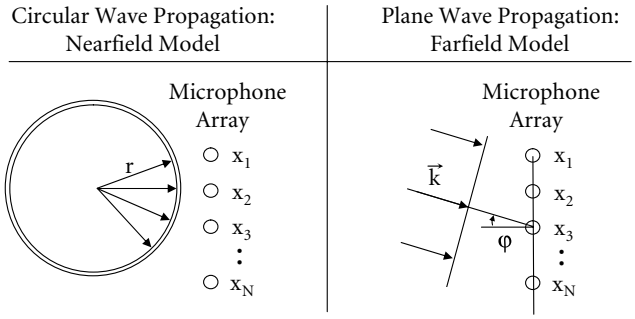

Figure 4.2: Nearfield (left) and farfield (right) model of sound propagation.

monochromatic wave, the amplitude as a function of space and time is given by

$$s(\mathbf{x}, t) = Ae^{j(\omega t - \mathbf{k}\cdot\mathbf{x})}, \tag{4.36}$$

where $\mathbf{x}$ denotes the location and $\mathbf{k}$ stands for the wavenumber vector [Joh1993], which has the direction of wave propagation and the magnitude $2\pi/\lambda$, with λ denoting the wavelength.

For the nearfield model, again in the monochromatic case, wave propagation is described by

$$s(r, t) = \frac{A}{r}e^{j(\omega t - kr)}, \tag{4.37}$$

where r is the distance to the sound source and the wavenumber k is given by ω/c.

With both these models, the measured signal at the sensors will correspond to the original signal $s(t)$, which arrives at all sensors $x_i, i = 1 \ldots N$ with a given delay δ_i and attenuation a_i, so

$$\mathbf{x}(t) = \begin{bmatrix} a_1 s(t - \delta_1) \\ a_2 s(t - \delta_2) \\ \vdots \\ a_N s(t - \delta_N) \end{bmatrix}. \tag{4.38}$$

In the frequency domain, this is equivalent to

$$\mathbf{X}(j\omega) = \begin{bmatrix} a_1 S(j\omega) e^{-j\omega\delta_1} \\ a_2 S(j\omega) e^{-j\omega\delta_2} \\ \vdots \\ a_N S(j\omega) e^{-j\omega\delta_N} \end{bmatrix}, \tag{4.39}$$

which can also be written in matrix form via

$$\mathbf{X}(j\omega) = \mathbf{r}(\omega) \cdot S(j\omega), \tag{4.40}$$

where $\mathbf{r}(\omega)$ is the steering-vector comprising the amplitude attenuations and the phase shifts incurred by the signal between source and sensor and is defined by

$$\mathbf{r}(j\omega) = \begin{bmatrix} a_1 e^{-j\omega\delta_1} \\ a_2 e^{-j\omega\delta_2} \\ \vdots \\ a_N e^{-j\omega\delta_N} \end{bmatrix}. \tag{4.41}$$

For the farfield model, the delays δ_i depend on the direction of arrival (DOA) φ, measured relative to broadside, via

$$\delta_i = \frac{d_i \sin(\varphi)}{c}, \tag{4.42}$$

where d_i stands for the distance between microphone i and the array origin. Therefore, $\mathbf{r}$ may also be expressed as a function of source-DOA, which is denoted as

$$\mathbf{r}(j\omega, \varphi) = \begin{bmatrix} a_1 e^{-j\omega\delta_1(\varphi)} \\ a_2 e^{-j\omega\delta_2(\varphi)} \\ \vdots \\ a_N e^{-j\omega\delta_N(\varphi)} \end{bmatrix}. \tag{4.43}$$

Delay and Sum Beamforming

Among the classical beamformers, a very popular one is the delay and sum beamformer. It attempts to emphasize the signal by phase-correct summation of the sensor signals, thus, for the model given in Equation (4.38), computing

$$\hat{s}(t) = \sum_{i=1}^{N} x_i(t - (\delta_T - \delta_i)), \tag{4.44}$$

with $\delta_T \geq \max_i \delta_i$, can exactly compensate the incurred signal delays. Due to this phase-correct addition, $\hat{s}$ consists of the sum of the amplitudes of the received source signals, whereas for incoherent noise, summation adds the energies. Thus, the desired signal energy in $\hat{s}$ grows with N^2, whereas the noise energy increases only with the factor N, yielding an SNR-improvement of 3dB for each doubling of N, or

$$SNR_{post} - SNR_{pre} = 10 \log_{10} N. \tag{4.45}$$

However, the delay and sum beamformer performance is hampered by the breadth of its main lobe at low frequencies, see Figure 4.3, whereas at high frequencies, spatial aliasing occurs due to undersampling of the incoming sound waves. This effect sets in as soon as $d > \frac{\lambda}{2}$ where d is the distance between the microphones. The effect is illustrated in the second subplot of Figure 4.3, where $d = 5\text{cm} > \frac{\lambda}{2} \approx 3.4\text{cm}$, so that the spatial sampling frequency is less than twice the spatial frequency, leading to two distinct nulldirections instead of only one as desired.

Filter and Sum Beamforming

More design parameters offer more flexibility in shaping beampatterns. This is the basic reason for employing a filter and sum beamformer. In this method, the simple delays of the delay-and-sum beamformer are replaced by filters $w_i(k)$, as shown in Figure 4.4. This allows a separate design of the discrete time frequency response for each frequency range, since the contribution and delay of each microphone can be adjusted over frequency. When the conjugate complex filter transfer functions W_i^* are composed in a row vector $\mathbf{W}^*(j\omega) = [W_1^*(j\omega), \ldots, W_N^*(j\omega)]$ according to [Gan2001], the array response can now be computed as the product of two linear systems, the room transfer function modeled by the steering vector in (4.39) in series with the filter responses, so that

$$\hat{S}(j\omega) = \mathbf{W}^*(j\omega)\mathbf{r}(\omega, \varphi) \cdot S(j\omega) \tag{4.46}$$

and the transfer function $\Psi(\omega, \varphi)$ from source to beamformer output is obtained from

$$\Psi(\omega, \varphi) = \mathbf{W}^*(j\omega)\mathbf{r}(\omega, \varphi). \tag{4.47}$$

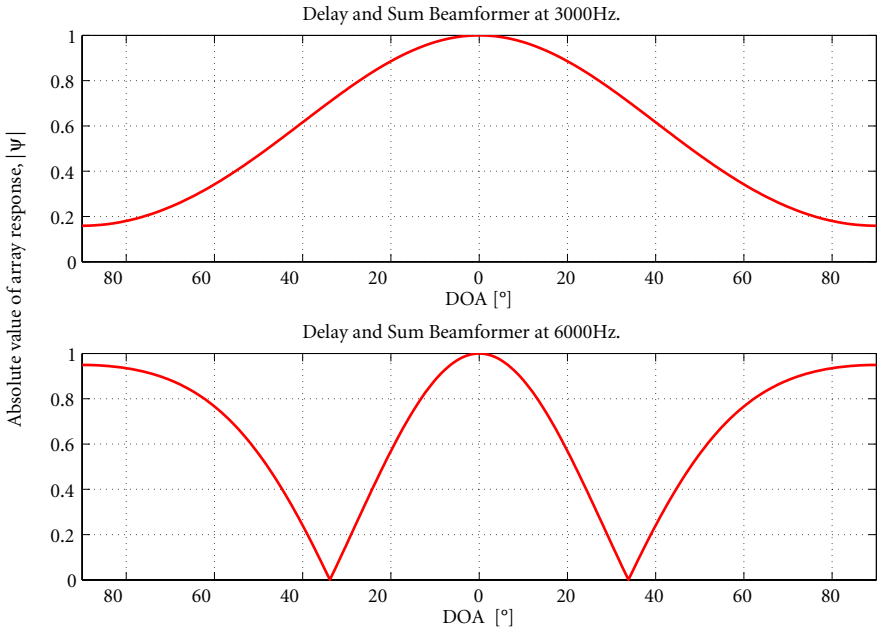

Figure 4.3: Directional response of a delay and sum beamformer for a microphone distance $d = 5$cm, at $f = 3000$Hz (top) and $f = 6000$Hz (bottom). See (4.47) for a definition of the array response Ψ.

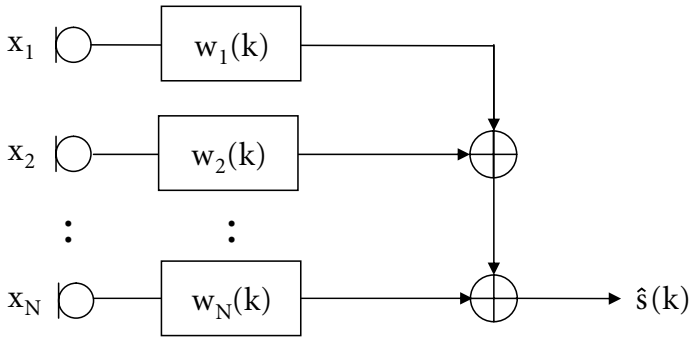

Figure 4.4: Structure of a filter and sum beamformer.

Therefore, the directional response (or beampattern) in general varies over frequency and DOA φ, so that signals arriving from different directions are effectively filtered by different $\Psi(\omega)$. However, if an equal response of the array is desired for all frequencies, this can be approximated by so-called *frequency invariant beamformers*, which strive to keep the array response constant over an arbitrarily wide bandwidth [War1996].

Optimal Beamforming

To adapt beamformers to arbitrary source and interferer directions, three criteria are significant.

- One is the minimization of noise energy,
- the second is keeping directional interference to a minimum,
- and the third is the constraint of maintaining an undistorted signal.

For attaining an optimum with respect to these criteria, two principally different approaches exist, the deterministic optimization approach and Bayesian (stochastic) estimation techniques.

In deterministic optimization, one of the most influential adaptive beamformers was designed by Frost [Fro1972]. In this approach, noise energy and directional interferers are combined in a joint optimization criterion, namely, the array output energy, subject to the constraint of undistorted signals arriving from the target direction φ_s. This leads to the cost function

$$J_{MVDR} = \mathrm{E}\left(|\hat{S}(j\omega)|^2\right) + \lambda\left(\mathbf{W}^*(j\omega)\mathbf{r}(\omega,\varphi_s) - 1\right), \tag{4.48}$$

which can be optimized to obtain minimum signal energy, subject to the constraint that $\mathbf{W}^*(j\omega)\mathbf{r}(\omega,\varphi_s) = 1$ for signals coming from the desired look direction φ_s. The resulting beamformers are also referred to as minimum variance distortionless response (MVDR) beamformers and are described in more detail below.

A major enhancement to MVDR beamforming has been incorporated in the optimization by Griffiths and Jim [Gri1982]. They describe an extension of the beamformer structure to include zeroing interferer directions. For this purpose, they

combine two beamformers, a conventional delay and sum beamformer to enhance the target signal and an adaptive path, from which the target signal is first removed by means of a blocking matrix **W**. In the adaptive, lower path, unconstrained adaptation is employed to find an optimum estimate of the interferer signal $y_A(k)$, which is then subtracted from the conventional beamformer output $y_C(k)$ to yield an optimum signal estimate $\hat{s}(k) = y_C(k) - y_A(k)$. Since the lower path suppresses energy impinging from any but the target sources, the method is known not only as the *Griffiths-Jim Beamformer* but alternatively also as the *Generalized Sidelobe Canceler*. Its structure, which is shown in detail in Figure 4.5, has been extended in

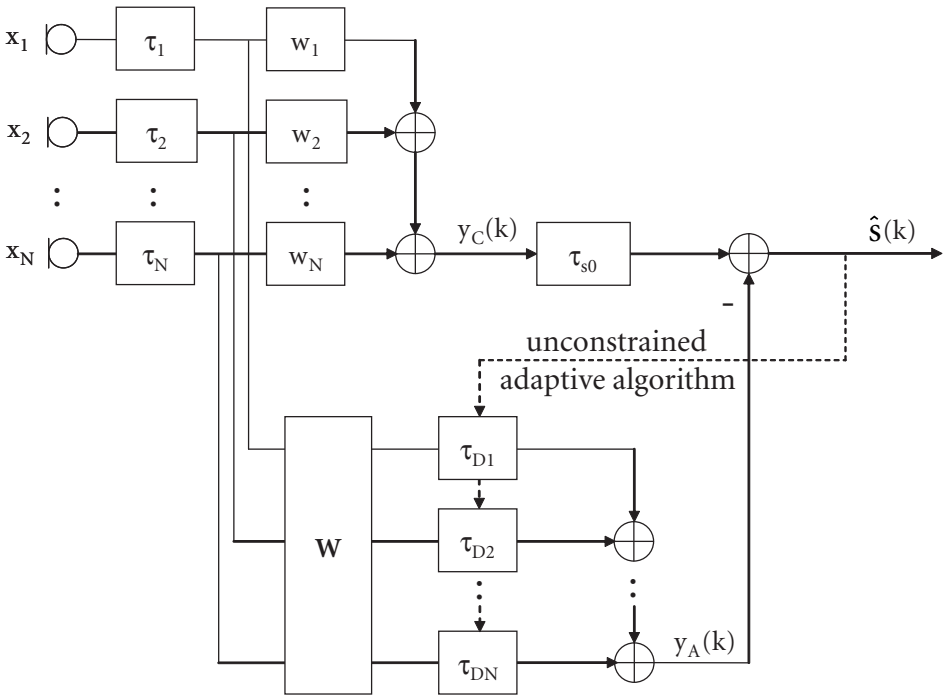

Figure 4.5: Structure of the generalized sidelobe canceller.

[Gan2001] to become applicable also to reverberant mixtures.

Aside from deterministic optimization, the second principal approach is stochastic signal estimation. It was first formulated for the beamforming problem in a

maximum likelihood sense as

$$\hat{s} = \arg\max_{s} p(\mathbf{x}|s) \tag{4.49}$$

by Capon, e.g. in [Cap1979]. While optimization by the minimum variance principle only needs the autocorrelation function of the measured signal - desired signal plus noise and interference - the maximum likelihood approach requires the covariance matrix of noise and interference, so that an additional voice activity detection is required for the system.

MVDR Beamforming

Minimum variance distortionless response beamforming is based on minimizing the beamformer output $\mathbf{W}^*(\omega)\mathbf{X}(\omega)\mathbf{X}(\omega)^H\mathbf{W}(\omega)^T$ with $\mathbf{W}^*(\omega)$ as the filter transfer functions, subject to the constraint $\mathbf{W}^*(\omega)\mathbf{r}(\omega, \varphi_s) = 1 \ \forall\omega$, which means that the signal arriving from the desired direction φ_s is not attenuated [Fro1972]. Stating these requirements as an optimization problem, as seen in (4.48), and dropping the frequency index for notational simplicity leads to

$$\min_{\mathbf{W}} \left(\mathbf{W}^* \mathbf{R}_{xx} \mathbf{W}^T + \lambda \cdot (\mathbf{W}^* \mathbf{r}(\varphi_s) - 1) \right), \tag{4.50}$$

where λ is the Lagrange multiplier for this constrained optimization problem and $\mathbf{R}_{xx}$ is the sensor autocorrelation matrix $\mathrm{E}\left(\mathbf{X}(\omega)\mathbf{X}(\omega)^H\right)$. To obtain the optimum with respect to the complex parameter $\mathbf{W}$, the gradient of (4.50) with respect to $\mathbf{W}^*$ must be used, as shown in [Bra1983]. Taking the derivative and equating it with zero results in

$$\mathbf{R}_{xx}\mathbf{W}^T + \lambda \mathbf{r}(\varphi_s) = \mathbf{0}. \tag{4.51}$$

Therefore,

$$\mathbf{W}^T = -\lambda \mathbf{R}_{xx}^{-1} \mathbf{r}(\varphi_s) \tag{4.52}$$

and inserting the distortionlessness criterion $\mathbf{W}^*\mathbf{r}(\varphi_s) = 1$ leads to

$$\begin{aligned} 1 = \mathbf{W}^*\mathbf{r} &= -\left(\mathbf{W}^T\right)^H \mathbf{r} \\ &= -\left(\lambda \mathbf{R}_{xx}^{-1} \mathbf{r}(\varphi_s)\right)^H \mathbf{r}(\varphi_s) \\ &= -\lambda \mathbf{r}(\varphi_s)^H \mathbf{R}_{xx}^{-1} \mathbf{r}(\varphi_s), \end{aligned} \tag{4.53}$$

where the last equality follows from the fact that $\mathbf{R}_{xx}^{-1}$ is a symmetric, real matrix. Then, Equation (4.53) gives λ to be

$$\lambda = -\frac{1}{\mathbf{r}(\varphi_s)^H \mathbf{R}_{xx}^{-1} \mathbf{r}(\varphi_s)}. \tag{4.54}$$

Finally, this can be plugged into (4.52) to obtain the optimal filter transfer functions

$$\mathbf{W}^T = \frac{\mathbf{R}_{xx}^{-1} \mathbf{r}(\varphi_s)}{\mathbf{r}(\varphi_s)^H \mathbf{R}_{xx}^{-1} \mathbf{r}(\varphi_s)}. \tag{4.55}$$

A similarly structured equation results, when the noise plus non-directed interference autocorrelation is used in the optimization:

$$\mathbf{W}^T = \frac{\mathbf{R}_{nn}^{-1} \mathbf{r}(\varphi_s)}{r(\varphi_s)^H \mathbf{R}_{nn}^{-1} \mathbf{r}(\varphi_s)}. \tag{4.56}$$

In that case, the output of the array will contain the minimum amount of noise and interference, instead of the minimum energy [Cap1967]. This is desirable in practice, however, it requires an estimate not of the sensor autocorrelation (which is easy to obtain), but of the noise autocorrelation, so that a voice activity detection will be required.

Bayesian Approach to MVDR Beamforming

When the direction of arrival of the desired signal φ_s is not known in advance, it also needs to be estimated. An erroneous estimation can be shown to be highly detrimental for the resultant SNR improvement [Kna2003]. With increasing angular error, the target signal is increasingly attenuated, and due to the frequency dependent beamformer response, is also filtered. Additionally, suppression of the interferer is reduced to such a degree, that even at relatively small angular errors the interferer energy can already exceed the target signal energy. To minimize the detrimental effects of not knowing the source direction, Bell et. al. suggest adapting the beamformer to the current source direction in [Bel2000]. For this purpose, the DOA is assumed to be a discrete random variable, and a beamformer can then be designed as a weighted sum of minimum variance distortionless response (MVDR) beamformers, of which each is weighted by the a posteriori probability of its associated target DOA.

Beamforming for Robust Speech Recognition

A variety of publications deal with the application of beamforming to robust speech recognition. Both principal approaches, deterministic and stochastic estimation, have been applied successfully.

Bayesian estimation has been applied to magnitude (*short time spectral amplitude estimation*) and phase estimation in the spectral domain by Balan and Rosca in [Bal2002]. This results in an extension of the Ephraim-Malah-filter to microphone arrays. Bayesian estimation of Log Spectra is described in detail in [Ban2003], where it is employed for noisy in-car speech recognition and results in an average word error rate reduction of 43% over a single microphone, when averaged over a number of noise conditions.

The basic deterministic beamformer structures are investigated by Omologo et. al. in [Omo1997], where a delay and sum beamformer is used, or by Saruwatari et. al. in [Sar1999]. Here, complementary beamforming (a nonlinear extension of the generalized sidelobe canceller shown in 4.5) is shown to improve speech recognition rates by more than 20% in low-SNR-conditions.

Regarding deterministic optimization, postfiltering of the beamformer output has emerged as a quasi-standard in recent publications. Since the achievable noise reduction rate using a beamforming system is restricted, and since superdirective beamformers, focused on removal of directional interferers, may even increase the incoherent noise part in the beamformer output, postfiltering of the beamformer output is an interesting option. With beamforming plus postfiltering, McCowan et. al. achieve an error rate reduction from 41% down to 9% in a reverberant room [McC2000] with incoherent noise and undesired background speech. Speech recognition improvements for beamforming with a denoising postfilter are also reported by Meyer and Simmer, who use delay and sum beamforming with spectral subtraction and Wiener filtering as postfilters in [Mey1997], by Maganti et. al., who employ filter and sum beamforming with postfiltering in [Mag2005], or by Moore and McCowan [Moo2003], who add a postfilter to a delay and sum beamformer.

Using speech characteristics for beamformer adaptation has also been investigated in different publications. For this purpose, [Yam1996] use the pitch-harmonic structure of speech to obtain a speaker direction estimate for adaptive delay and sum beamforming. Seltzer et. al. [Sel2003] integrate speech recognition directly in the adaptation by optimizing log-likelihoods of the speech data in

all subbands, while [Nis2003] incorporate the average speech spectrum in beamformer adaptation, for this purpose allowing small deviations from the otherwise desired distortionless response.

Beamforming in conjunction with recognizer relevant features has been evaluated in detail by Stern et. al. [Ste1992], who use a 23-microphone delay and sum beamformer. As shown, this can provide complementary benefits when used in conjunction with code dependent cepstral normalization for Mel-Frequency Cepstral Coefficients. Also, it was tested in the same setup as a preprocessor for an auditory type feature extraction (the Seneff haircell model [Sen1984]), where it gave inconclusive results.

Also, good VAD is essential for adaptation control of adaptive beamformers. This was shown in the work of Van Campernolle [Van1990], who investigated a switched Griffiths & Jim beamformer with robust VAD to prohibit incorrect adaptation of the noise reference during speech presence. Improved VAD was also a central aim in [Lun1999], which compares a generalized sidelobe canceller with soft adaptation during speech absence to spectral subtraction, and comes to the conclusion that beamforming offers a recognition rate improvement superior to spectral subtraction, which corresponds well with its lower cepstral distortion measures.

However, even though many publications deal with the application of beamforming to noisy speech recognition, it is rarely employed for speech-speech separation and recognition. McCowan et. al. in [McC2000] make an exception, where background speech is also successfully removed. Among the cited publications on beamforming, only [Moo2003] and [Mag2005] explicitly deal with 0dB SNR speech-speech-mixtures, which can be expected to pose additional problems due to the spectral similarities, making recognition under residual crosstalk more challenging. However, they still find microphone arrays significantly inferior to close-talk microphones in realistic scenarios, where an error rate of 19% is the best reported performance on a number recognition task after beamforming and maximum a posteriori (MAP) HMM-adaptation.

4.2.2.2 ICA

In contrast to beamforming, independent component analysis (ICA) has often been employed for speech-speech segregation.

In the original ICA methods, much emphasis was on the *blindness* of source separation, that is, on obtaining the correct unmixing system without resorting to any assumption regarding the characteristics of the source signals. Usually, one of the three following properties is used [Car2001]

Non-Gaussianity No more than one of the sources is allowed to adhere to a Gaussian probability distribution.

Non-Stationarity The sources must change in some of their statistical properties over time.

Non-Whiteness Sources are not allowed to have a flat spectrum.

As soon as at least one of these properties is given for the mixed sources (except for one source, which is allowed to be Gaussian, white and stationary), mixtures can be segregated without needing further information regarding the source characteristics. If that is the case, the separation process can be carried out by estimating and subsequently inverting the mixing matrix according to a suitable cost function, which, often indirectly, measures deviation of the sources from Gaussianity, their whiteness or stationarity. The mathematical background of these original, linear ICA methods is described in some more detail in Appendix B. But since the original methods of ICA, like the Infomax algorithm and JADE, are designed for purely additive mixtures of signals like

$$\begin{aligned} x_1(t) &= a_{11}s_1(t) + a_{12}s_2(t) \\ x_2(t) &= a_{21}s_1(t) + a_{22}s_2(t), \end{aligned} \tag{4.57}$$

they are not applicable in the convolutive case, where each source signal i is convolved with the room impulse response $h_{ij}(t)$ before it reaches sensor j:

$$\begin{aligned} x_1(t) &= h_{11}(t) * s_1(t) + h_{12}(t) * s_2(t) \\ x_2(t) &= h_{21}(t) * s_1(t) + h_{22}(t) * s_2(t). \end{aligned} \tag{4.58}$$

Therefore, the frequency domain formulation is often taken as a basis of convolutive source separation. Here, convolutions with the room impulse response reduce to multiplications with scalar (but complex) weights at each frequency:

$$\begin{aligned} X_1(\omega) &= H_{11}(\omega)S_1(\omega) + H_{12}S_2(\omega) \\ X_2(\omega) &= H_{21}(\omega)S_1(\omega) + H_{22}S_2(\omega). \end{aligned} \tag{4.59}$$

In this way, for each frequency bin the convolutive problem is reduced to a set of additive mixtures similar to (4.57), at least when sufficiently long frames can be used.[5] However, the ordering and scaling of ICA outputs is arbitrary and may change from frequency bin to frequency bin as detailed in Appendix B. Therefore, special care must be taken in assigning the ICA outputs to the sources and in determining their scale. Regarding the scaling problem, one possible solution consists of constraining all direct paths in the convolutive mixing model to one. This results in the modified convolutive mixing model shown in Figure 4.6. Such a model does not allow identification or compensation of the direct path room transfer functions. However, statistical independence alone is an insufficient criterion to identify these transfer functions, so no significant information is lost by this model choice as long as there are no additional sources of information [Bau2005].

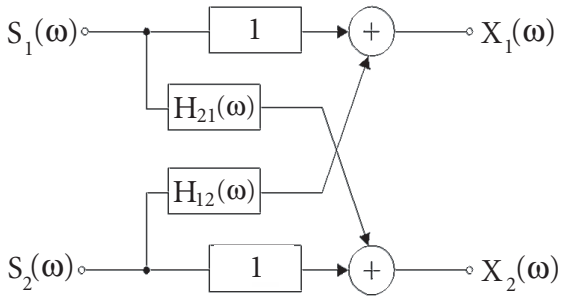

Figure 4.6: Constrained convolutive mixing model.

Regarding the permutation problem, a number of different solutions have been applied, three of which, namely

- utilizing the similarity of the transfer function in adjacent frequency bins,
- optimizing criteria which consider the independence of a number of frequency bands simultaneously and

[5] When the room impulse response is longer than the frame length, a signal which was emitted in frame number τ will also influence later frames $\tau + 1$ etc., so that the filtering effect can only be described approximately as a multiplication but in general must be modeled by a convolution in the frequency domain [Ser2003].

- aligning frequency bins according to their directivity characteristics

are described in more detail here.

Constraining adjacent frequency bands to be similar leads directly to flatness criteria, which evaluate the similarity between neighboring bins and are exploited e.g. by [Bau2001, Cap1995], and indirectly to a constraint of short time domain mixing filters, since flatness in the frequency domain directly corresponds to shortness in the time domain [Par2000]. However, such methods are difficult to apply in reverberant environments, where unmixing filters need to be long for good separation performance. In this case, choosing the parameter for constraint enforcement, e.g. the maximum filter length in Parra's method, is difficult, because the optimal trade-off between errors caused by insufficiently long unmixing filters and errors due to permutations is often difficult to find.

Joint optimization of the output cross correlation across frequency bins is another strategy, which is often successful for reducing frequency permutations. In its implementation by Anemüller [Ane2001], the cost function c for this purpose is the *amplitude modulation correlation*, which is defined by

$$\mathrm{c}(Y_1(\tau, f_k), Y_2(\tau, f_l)) = \mathrm{E}\{(|Y_1(\tau, f_k)| - \mathrm{E}\,(|Y_1(\tau, f_k)|))\,(|Y_2(\tau, f_l)| - \mathrm{E}\,(|Y_2(\tau, f_l)|))\} \tag{4.60}$$

as the correlated deviation of the short time spectral amplitudes $|Y_1(t, f_k)|$ and $|Y_2(t, f_l)|$ from their mean values. When a matrix is constructed from this basic cost function by computing the cost for each pair of frequencies according to

$$[C(Y_1, Y_2)]_{kl} = \mathrm{c}(Y_1(\tau, f_k), Y_2(\tau, f_l)), \tag{4.61}$$

the Frobenius norm of this matrix[6] is a suitable cost function for minimizing permutations, since a frequency permutation will lead to correlations between different output signals at some frequencies. However, this procedure carries a large computational burden since evaluation of the cost function is time-consuming and its local optima require multiple sweeps of the optimization procedure. An alternative consists in constructing a cost function which models output distributions p($\mathbf{y}$)

[6]The Frobenius norm of a matrix is given by $\|M\|_{Fro} = \sqrt{\sum_{i,j} |m_{ij}|^2}$.

as multivariate, taking into account different frequencies and time lags simultaneously. Minimizing the mutual information between estimated output distributions [Buc2003] represented by the cost function

$$C(m) = \sum_{i=0}^{m} \beta(i,m) \cdot \frac{1}{N} \cdot \ldots \tag{4.62}$$
$$\sum_{j=0}^{N-1} \{\log\left(\hat{p}_{PD}(\mathbf{y}_1(i,j),\ldots,\mathbf{y}_P(i,j))\right) - \log\left(\hat{p}_D(\mathbf{y}_1(i,j)) \cdot \ldots \cdot \hat{p}_D(\mathbf{y}_P(i,j))\right)\}$$

leads to an inclusion of across frequency permutations in a single cost function, where $\mathbf{y}_n(i,j)$ is the n'th output signal in block i with the time shift j within the block, N is the block length, β is a window function[7] and $\hat{p}_D$ and $\hat{p}_{PD}$ are the estimated or assumed probability distributions of the dimension D (which is the number of considered time lags), respectively PD (with P as the number of sensors). Also, with an appropriate form of probability distribution $\hat{p}$, one can exploit properties of the sources such as short-time stationarity and higher order statistics.

Another technique for the explicit modelling of inter-frequency dependencies is to move from scalar complex valued *independent component analysis* to a complex vector valued source model in *independent vector analysis* [Kim2006, Lee2006]. In this approach, the output of the source separation algorithm is a set of N vectors $\mathbf{y_i}, i = 1 \ldots N$, each containing all frequency bands of the desired source i. Then, it is possible to construct priors for these vector valued random variables, which include the constraints of sparseness as well as variance dependencies between entries in the source vectors (i.e. source spectra). Using such priors, the cost function C will be independence of the entire vectors as in

$$C = \mathcal{KL}\left(\mathrm{p}(\mathbf{y_1},\ldots,\mathbf{y_N}) \middle\| \prod_i \mathrm{p}(\mathbf{y_i})\right), \tag{4.63}$$

where $\mathcal{KL}$ denotes Kullback-Leibler divergence. This cost function can be used as the optimality criterion for gradient descent methods, which will then lead to most independent *vectors* rather than independent *frequency band outputs*, thus avoiding opermutations explicitly due to the applied cost function.

[7] β can be used for blockwise weighting of the data, which allows for online adaptive implementations.

Finally, it is also possible to consider the ICA unmixing matrix in the frequency domain as a beamformer to obtain its impact on sources as a function of frequency and DOA. For this purpose, it is useful to analyze the effect, which a weighted summation of microphone signals has on the output signal. This effect has already been derived for beamformers in Section 4.2.2.1, resulting in Equation (4.47). Thereby, the beampattern of an arbitrary frequency domain unmixing matrix $\mathbf{W}(j\omega)$ can be computed over frequency and angle of incidence by concentrating on one column i of the unmixing matrix at a time, resulting in

$$\Psi_i(\omega, \varphi) = \mathbf{r}(\omega, \varphi)^T \mathbf{W}_{\cdot,i}(j\omega). \tag{4.64}$$

for the i'th unmixing filter. Application of this equation results in n beampatterns for an $n \times n$ unmixing matrix. An example of such patterns is shown in Figure 4.7.

As can be seen, these beampatterns show strong attenuations of a small range of directions for most frequencies. Since the unmixing structure of ICA is equivalent to that of a frequency variant nullbeamformer [Kna2003], the minimum directions of an ICA unmixing filter can be interpreted as zero directions, which point to the interferer directions upon convergence. This idea directly leads to the last concept for permutation correction, the analysis of directivity characteristics. With this principle, ICA unmixing filters are sorted in such a way as to align minimum directions. Different implementations for this principle exist. For example, in [Kur2000], first, source directions are estimated from an analysis of the minimum direction statistics and in a second step, unmixing filters are sorted in accordance with these estimated source directions, and in [Saw2004] the null directions are computed analytically from W and mixing matrix columns are sorted by null directions for all those frequencies, at which a high confidence null direction estimate is available, whereas other frequencies are assigned by inter-frequency correlations. In [Muk2004] the principle is extended to find zero directions together with zeroing distances based on a nearfield beamforming model. This also allows for separating speakers which stand in the same angular position relative to the microphones. One further approach for assigning permutations based on directivity patterns, which leads to the permutation corrected unmixing filters shown in Figure 4.8, is discussed in Section 5.1.1.

Aside from time-frequency processing, another approach for carrying out independent component analysis is based on an analysis method that is closer to

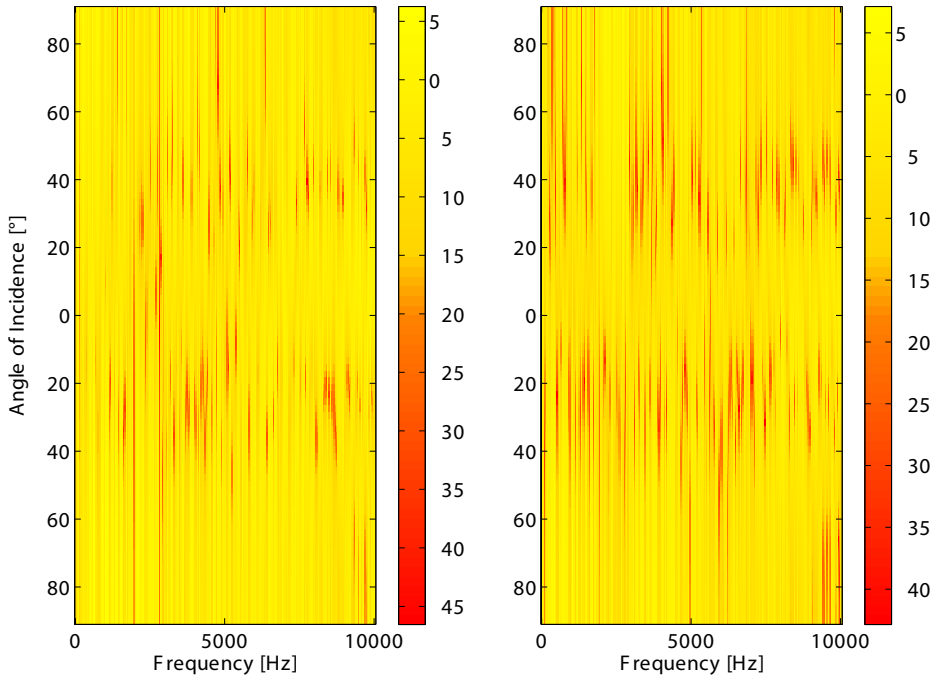

Figure 4.7: Beampatterns computed as $20\log_{10}|\Psi_i(\omega,\varphi)|$ for both columns of an ICA unmixing matrix W before permutation correction.

human auditory processing than the Fourier analysis described above. Here, a set of so-called correlograms is used, which are believed to offer a more accurate representation of the features extracted in the human hearing process than the spectrogram [Sla1993].

Correlograms are computed directly from the time domain source signals. First, the sensor signals are analyzed in a filter bank which models the human filtering process in the cochlea. The filter bank outputs are windowed, and then, for each windowed filter bank output, sets of autocorrelation coefficients are computed for all pairs of filterbank outputs, as described first in [Sla1993]. An example of a correlogram is shown in Figure 4.9.

After the signal is thus preprocessed, standard ICA techniques can also be

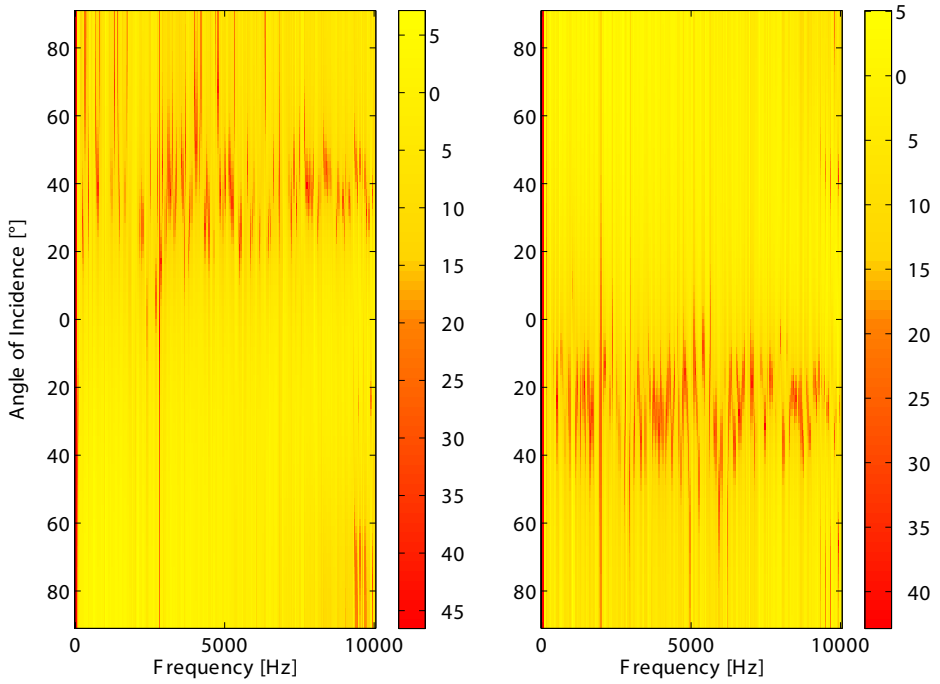

Figure 4.8: Beampatterns computed as $20 \log_{10} |\Psi_i(\omega, \varphi)|$ for both columns of an ICA unmixing matrix W after permutation correction.

applied to the correlograms, yielding a new set of demixed correlograms. Finally, these unmixed correlograms of the source signals are transformed back into the time domain, leading to an alternative solution for the source separation problem [Rut2001].

Finally, probabilistic models for speech and noise can also be included to perform source separation. One such method of incorporating the speech model in the source separation algorithm was developed by Attias in [Att1998]. The principal idea is to describe speech by its probability distribution, e.g. by an MOG model in the time domain:

$$p_{s_i}(s_i) = \sum_{m=1}^{M} \gamma_m \cdot \mathcal{N}(s_i, \mu_{i,m}, \sigma_{i,m}), \tag{4.65}$$

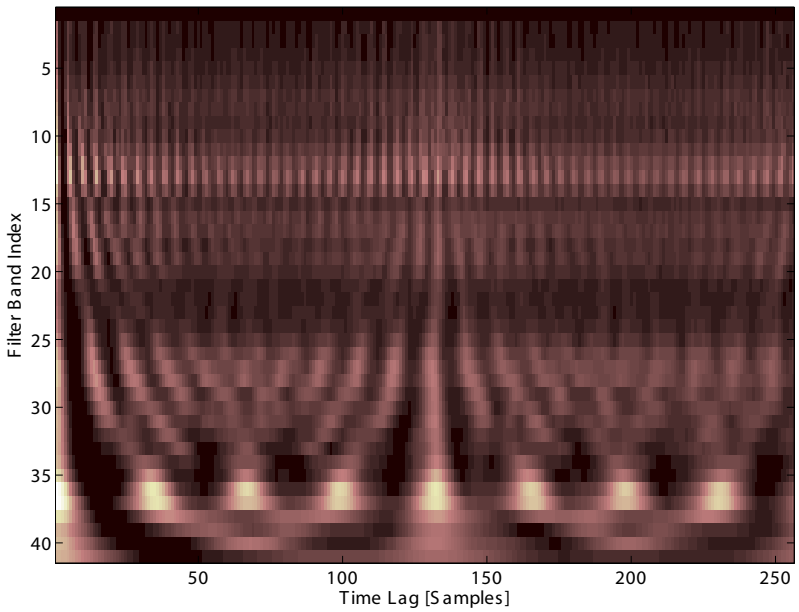

Figure 4.9: A correlogram shows the autocorrelation of auditory based filter bank outputs over time lag.

where $\mathcal{N}(\cdot, \mu, \sigma)$ denotes the normal distribution with mean μ and standard deviation σ. With this model, the mixing matrix A is estimated together with all source distribution parameters $\mu = \mu_1 \ldots \mu_M$ and $\sigma = \sigma_1 \ldots \sigma_M$ to obtain the closest match between the probability distribution $\mathrm{p}_{\hat{s}}(\hat{\mathbf{s}})$ of the unmixed signal

$$\hat{\mathbf{s}} = \hat{\mathbf{A}}^{-1}\mathbf{x} \tag{4.66}$$

and the postulated source distribution $\mathrm{p}_{s_i}(s_i)$ from (4.65). To achieve this optimal match, one suitable approach is to find that parameter set $\{\hat{\mathbf{A}}, \hat{\mu}, \hat{\sigma}\}$ which makes the observed data as likely as possible:

$$[\hat{\mathbf{A}}, \hat{\mu}, \hat{\sigma}] = \arg\max_{\mathbf{A},\mu,\sigma} p(x|\mathbf{A}, \mu, \sigma). \tag{4.67}$$

An expectation maximization algorithm can be used for this purpose, however, the problem becomes intractable as soon as noise is present in addition to speech, in

which case computationally expensive variational methods are the only currently applicable learning strategy, see, e.g. [Att2001, Lee2005].

ICA for Robust Speech Recognition

ICA has been applied successfully for speech/speech separation. However, the still not completely solved permutation problem as well as insufficient demixing performance in reverberant environments still pose problems for the recognition of overlapping speech signals. Still, a number of publications already show a significant improvement in recognition performance, when ICA is used as a preprocessing method.

For this purpose, it is necessary to explicitly deal with two problems, reverberation as well as permutations.

Once that is achieved, significant performance improvements can be obtained. For example, Asano et. al. [Asa2003] reduce the error rate from 50% to 32% in a reverberant room with t_{rev} = 400ms by means of ICA with additional beamforming as a preprocessor to reduce room reflections impinging from other than the source directions. In their approach, the detrimental effect of permutations is reduced by sorting ICA outputs according to the nulldirections of the unmixing filters as described above. Another approach for limiting permutations is by requiring smooth frequency domain unmixing filters, or equivalently, short time domain unmixing filters. This principle has been utilized by [Par2000] to reduce the word error rate on the Wall Street Journal task of continuous large vocabulary recognition from 43.7% to 34.5%. A further option to avoid permutations is a search for that assignment of sources in subbands, which leads to the smallest overall correlation between unmixed outputs. This principle has been applied successfully for speech recognition, where the error rate on a connected digits task was reduced from about 35% to 20% at 0dB SNR [Ane2001], but it is an expensive option, computationally.

Rather than complete demixing, sometimes it is sufficient to utilize a deflationary approach, which extracts the dominant target source first. This can also be used advantageously in reverberant environments, as shown in [Kou2002], with the advantage of lower computational complexity than conventional ICA. Alternatively, computational effort may be reduced by using second order criteria for separation. For example, Ehlers and Schuster in [Ehl1997], employ multiple decorrelation of

output signals at different time delays to obtain an error rate reduction from 52% down to 9% on a connected digit recognition task in a $20m^2$ reverberant room.

Noise is another significant problem. Here, it has been shown to be beneficial, when ICA is followed by a denoising stage. In this way Visser and Lee have applied ICA to noisy in-car speech mixtures, followed by an ICA-based and a second, wavelet-coefficient-shrinkage denoising stage and have achieved a reduction of error rates from 43% to 20% at 0-4dB SNR, when the baseline error was 10% [Vis2003]. In noisy scenarios, it is also very helpful, when a reliable voice activity detection is available, since this can be used for further denoising and recognition rate improvements [Vis2003b].

In contrast to the described methods, in which ICA is followed by feature extraction for the recognizer, it is also possible to apply ICA directly on the recognizer features, as shown in [Par1999]. In Park's approach, ICA is directly applied to auditory filter bank outputs on a mel-scale. However, this approach is not directly practicable for recognizer features with a non-linear dependence on spectral features, since the mixing model becomes intractable to estimate and non-linear ICA methods would be required, which is probably an inordinately difficult task for features such as MFCCs, where source and interference are mixed as in (3.14).

4.2.2.3 Computational Auditory Scene Analysis

When two speakers are simultaneously talking, it is impossible to separate the signals either by time segmentation or by "frequency segmentation" (i.e. filtering). However, two speech signals tend to be almost orthogonal in a time-frequency representation derived with the parameters usual for speech processing. This property is referred to as "W-disjoint orthogonality" of the signals, with the "W" standing for "windowed". This concept was introduced by Jourjine et al. in [Jou2000]. The property of approximate W-disjoint orthogonality leads directly to the idea of using spectral characteristics to obtain a binary mask for source segregation, and in this form, it has been used to obtain real time source separation for more sources than sensors in [Ric2001], [Yil2004], [Sri2004] and [Rom2003]. Reliance on a constant delay and attenuation between microphones is widely used in these examples of so-termed *computational auditory scene analysis* (CASA) methods. For

example, the mixing model used by Yilmaz et. al. in [Yil2004] is

$$x_1(\omega, k) = \sum_{j=1}^{N} s_j(\omega, k) \tag{4.68}$$

$$x_2(\omega, k) = \sum_{j=1}^{N} a_j s_j(\omega, k) \exp(-j\omega\delta_j) \tag{4.69}$$

where a_j is the attenuation and δ_j the delay of source j on sensor 2 compared to sensor 1. This model is used to derive estimators for the two parameters delay

$$\hat{\delta}(\omega, k) = -\frac{1}{\omega} \angle \left(\frac{x_2(\omega, k)}{x_1(\omega, k)} \right) \tag{4.70}$$

and damping

$$\hat{a}(\omega, k) = \left| \frac{x_2(\omega, k)}{x_1(\omega, k)} \right|, \tag{4.71}$$

which are subsequently used to derive appropriate masking functions.

A shortcoming of this delay- and attenuation-based approach is its reliance on a constant angle of incidence over frequency (equivalent to a constant delay) as well as a constant damping factor, which can lead to problems in reverberant source separation. However, considering more reflections in the model is not always helpful. As shown in [Bal2001], using greater numbers of reflections in the same structure shows performance comparable to the simple model of Equations (4.70) and (4.71). Also, the search algorithm employed to find the predominant angles of incidence and damping factors will often fail to converge, when the 2D-Histogram of amplitude and delay does not show the required sharp peaks. This is also the case for in-car recordings, for which histograms of noisy and clean data are shown in Figure 4.10 and 4.11.

However, there are a number of speech characteristics, which can be used to improve the separation performance of the CASA approach. Such cues, indicators of how the time-frequency points should be grouped to form the different sources, include synchronization cues such as common onset or common amplitude modulation, and harmonicity cues. Since these features help to include more speech-specific characteristics, which are also less sensitive to noise and reverberation than the above mentioned amplitude and delay differences, it can be expected

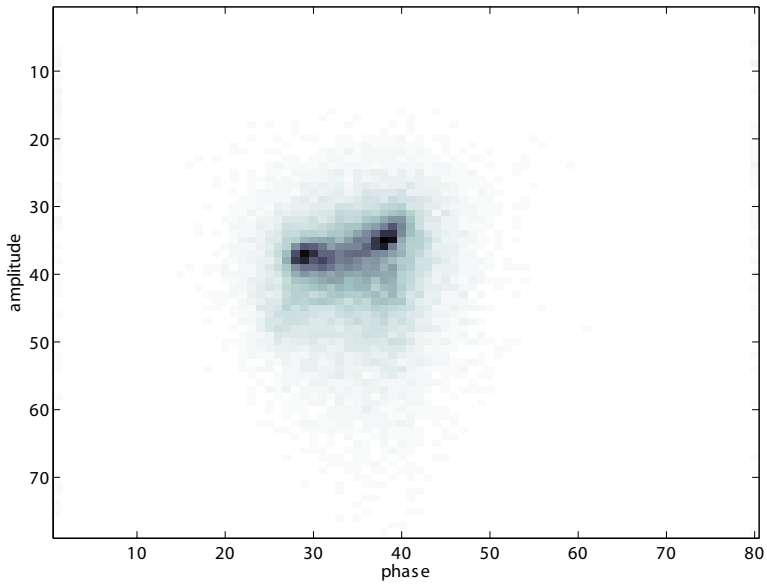

Figure 4.10: Histogram of two speakers in quiet surroundings.

that a combination of these criteria with the currently used methods will lead to a more robust solution of the auditory grouping problem, and much work is currently in progress to prove this point (e.g. [Bre1999, Row2000, Div2005, Ell2006]).

4.3 Adaptive Recognition

Changing environmental conditions can be dealt with on two sides. Either the feature estimate or the recognizer model can be adapted to the changing situations. *Adaptation of the Feature Vector* is another expression for *adaptive preprocessing* methods, many of which are already described in 4.2, so this chapter will focus on model but not on feature adaptation in Section 4.3.1 and 4.3.2.

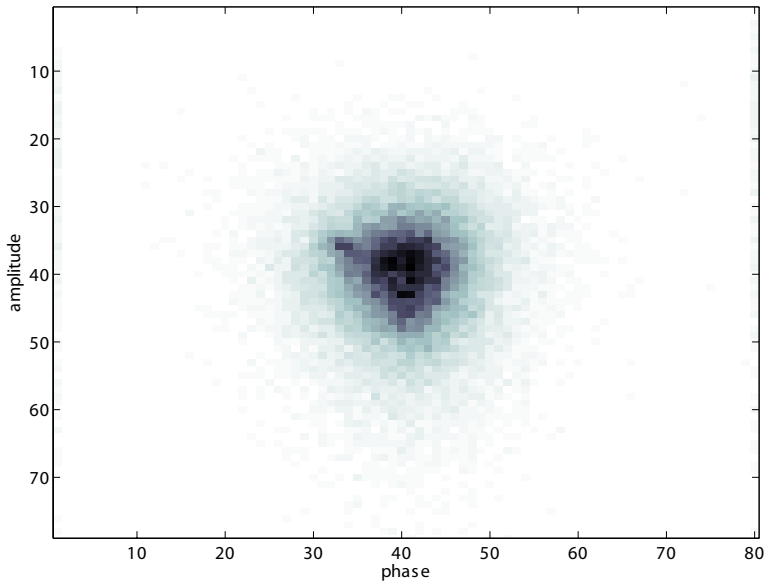

Figure 4.11: Histogram with equal speaker positions and amplitudes but with added noise.

4.3.1 Parallel Model Combination

Parallel model combination (PMC) offers a way of combining noise models and speech models to obtain noisy speech models. The principle behind PMC is the following: the speech model in the recognition system describes the probability distribution of speech in the cepstral domain, but the effects of noise and reverberation are most easily modeled in the linear spectral domain. Therefore, the models of the speech recognition system are transformed back to the spectral domain, the noise and reverberation (which must be known or estimated) are added to, respectively convolved with, the clean speech model, and a subsequent transformation of these models back to the cepstral features results in models of speech as it would be observed in the given noisy and reverberant environment. Figure 4.12, from [Gal1995], shows the associated data flow.

PMC is often successfully used, when the noise is purely additive and stationary or when the only effect to be compensated is that of an unknown transfer

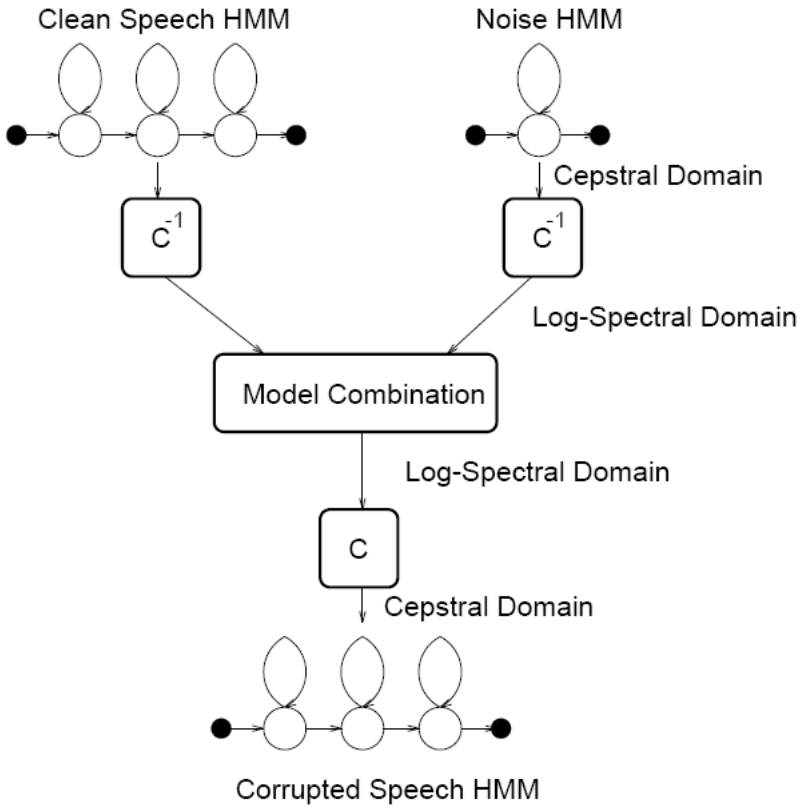

Figure 4.12: Structure of Parallel Model Combination [Gal1995].

function. However, for convolutive mixing of noise and speech, the offered extensions are problematic and nonstationarity of the noise can only be handled by having a multitude of models which needs a large amount of storage and computational effort at the same time. This problem can be alleviated for mildly nonstationary noise environments by computing online the updated models. For this purpose, a fast version of PMC has also been proposed [Gal1995b].

Also, PMC can be combined with noise reduction methods to give a further performance improvement. For example, it has been combined with spectral subtraction to obtain improvements over spectral subtraction alone [Nol1993] as well as over only parallel model combination [Nol1994]. Still, its need for an accurate

noise estimate makes it unsuitable for very quickly changing noise scenarios. To deal with this scenario, recently, PMC has been used in combination with an HMM noise model by jointly recognizing the speech state and the noise state. While this approach requires a pre-trained noise model for all possible scenarios, it is applicable also for situations, in which a fast noise model update is necessary [Kri2001]. However, joint recognition of speech and interfering speech is still computationally far too intensive for real-time or embedded applications and was therefore not considered here.

4.3.2 Maximum Likelihood Linear Regression

With maximum likelihood linear regression (MLLR), a linear transformation model is set up which can be used to compensate for

- added noise
- changed channel or
- speaker characteristics.

For this, the model parameters are assumed to be transformed linearly, thus, a linear and hopefully inverse transformation is learned [Leg1994]. When the parameters to be adapted are the means of a Gaussian mixture model, the following transform equation is used:

$$\mu' = \mathbf{T}\mu^{+}. \tag{4.72}$$

Here, $\mathbf{T}$ is the learned transformation matrix and μ^{+} stands for an augmented mean vector $\mu^{+} = [\omega, \mu_1, ...\mu_n]$ which contains a constant ω in order to allow for a bias. The transformation matrix is learned by maximizing the observation likelihood $p(o|\mu')$ in an EM algorithm[8]. Additionally, the compensation of variances is also possible for improved adaptation performance, see [Leg1995]. Recently, an alternative to the maximum likelihood criterion has emerged for learning optimal linear regression parameters. This is the minimum classification error criterion, which has been shown to outperform maximum likelihood based adaptation consistently even in large-vocabulary tasks [Wu2002, He2003].

[8]EM has been used as a shorthand both for *expectation maximization* and *estimate maximize*, with the first explanation being far more common.

4.4 Incorporation of Uncertainties

Additionally, instead of adapting the model or feature, there is the alternative of estimating the model or feature *uncertainty*. Thus, the methods employed for uncertain recognition do not need to directly estimate the model distortion or feature distortion, as long as they can provide an estimate of the probability that the observed feature or noisy model can deviate from the clean feature or model by a certain amount. Then, the Viterbi search[9] which is performed during recognition, will make an online estimate of the true model or feature, along with its estimate of the best HMM state-sequence. This leaves more work but also more flexibility for the search algorithm, but it also allows dealing with distortions more quickly, for example on a frame by frame basis, than it would be possible purely with model adaptation. Model domain uncertainty processing is described in Section 4.4.1 through 4.4.1.3, whereas the methods of feature domain uncertainty processing are explained in more detail starting from Section 4.4.2.

4.4.1 Model Domain Uncertainty

Since the data input to the recognizer is only known with a limited accuracy, some robust recognition systems aim at incorporating the uncertainty into the decoding process by calculating a new, uncertain speech HMM [Huo2000, Mer1993, Jia1999]. These approaches are subsequently termed ”uncertainty modeling methods”, as they incorporate uncertainty information into a flexible speech model with uncertain parameters. An HMM-based speech model usually is described by the following parameters:

- The initial state probabilities $a_0(i)$
- The transition matrix A and
- The parameters of the output probability distributions $b_i(o)$

which are introduced in Section 2.2. When a clean speech signal is corrupted by noise, it can be assumed that the probability of each word and the speaking speed

[9] The same is also true for any other HMM-based dynamic programming search strategy such as beam search or token passing.

stay approximately the same, only the speech features are changed by the added noise signal.[10] Thus, in order to describe how a noisy speech model would deviate from a clean speech model, the change in the state- and transition probabilities is often neglected, but it is always necessary to describe the influence of the noise and interference on the output probability distributions $b_i(o)$. This influence can be described by focusing on the parameters of the pdfs $b_i(o)$. In the following, these parameters will be denoted by the parameter vector $\mathbf{\Lambda}$ and they can have a number of concrete forms, but most likely will take the shape of means, variances and mixture probabilities of a Gaussian mixture model as defined in (2.24).

When a known noise signal is added to the speech signal, it is possible to compute new values $\mathbf{\Lambda}'$ for these parameters and to use those for recognition, as it is done in model adaptation techniques, e.g. in Parallel Model Combination, see 4.3.1. However, the noise is usually not known precisely but rather, it is itself a random variable. Thus, it is not precisely possible to obtain new model parameters and instead, it is often useful to consider the parameters $\mathbf{\Lambda}$ also as random variables.

To compute the probability of a certain model parameter taking a given value, another probability model, the so-called *prior pdf* $p_\Lambda(\mathbf{\Lambda})$, is set up. There are different ways to define this pdf, and thus describe model uncertainty. For example, in [Hub1965], model uncertainty is described nonparametrically by letting the probability distribution be a mixture of the nominal model pdf P_{nom} learned from training data and another arbitrary probability measure P_{dist}, as in

$$P_{robust}(o) = \epsilon P_{nom}(o) + (1 - \epsilon) P_{dist}(o). \tag{4.73}$$

Alternatively, the uncertainty of the speech model can also be described parametrically, by allowing the model parameters to vary within a certain fixed amount around the parameters estimated from the training data. This model will depend on the training data, which gives the first estimate of $\mathbf{\Lambda}$, and on the noise process estimate, which indirectly gives information about how far each parameter Λ_i may deviate from the original estimate.

[10] This model actually ignores all changes in speech characteristics caused by the speaker also perceiving his noisy environment, which in reality leads to modified speech amplitude and pitch as well as a slight change in formants, the so-called *Lombard effect*, see e.g. [Jun1993].

Once the model uncertainties have been defined, either parametrically or nonparametrically, this uncertain model can be used for recognition. At this point, there are three alternatives.

4.4.1.1 Maximum a Posteriori Adaptation

In many ways similar to maximum likelihood linear regression, maximum a posteriori (MAP) adaptation is also used for noise, channel and speaker adaptation. In addition to the likelihood of the observation $\mathbf{o}$ given the adapted model parameters, $p(\mathbf{o}|\mathbf{\Lambda}^{+})$, MAP adaptation also considers prior probabilities for the model parameters in form of a *prior pdf* $p_{\Lambda}(\mathbf{\Lambda})$ [Gau1994]. This allows a maximization of the model parameter probabilities via

$$\begin{aligned} \Lambda' &= \arg\max_{\Lambda} P(\Lambda|\mathbf{o}, \mathcal{P}) \\ &= \arg\max_{\Lambda} \frac{p(\mathbf{o}|\mathbf{\Lambda}) p_{\Lambda}(\mathbf{\Lambda}|\mathcal{P})}{p(\mathbf{o})}, \end{aligned} \tag{4.74}$$

where $\mathcal{P}$ stands for the prior information. Due to the use of this Bayesian formulation, MAP adaptation itself is sometimes also referred to as *Bayesian adaptation*. If no prior probabilities are used or a non-informative prior is employed, the MAP solution corresponds to that of maximum likelihood linear regression, since, with a noninformative prior $p_{\Lambda}(\mathbf{\Lambda})$

$$\begin{aligned} \Lambda' &= \arg\max_{\Lambda} \frac{p(\mathbf{o}|\mathbf{\Lambda}) p_{\Lambda}(\mathbf{\Lambda})}{p(\mathbf{o})} \\ &= \arg\max_{\Lambda} p(\mathbf{o}|\mathbf{\Lambda}), \end{aligned} \tag{4.75}$$

which is exactly the maximum likelihood solution. However, in contrast to MLLR, MAP adaptation updates every component of a model separately, whereas MLLR can estimate a single transformation for entire phonetic classes, which makes MLLR more efficient, when few adaptation data is available. In contrast, MAP begins to outperform MLLR as soon as sufficient adaptation information has been obtained, since then, its detailed granularity and principled inclusion of prior information begin to give this method an advantage [You2002]. However, as can be seen from these considerations already, MAP adaptation is applicable only for fairly stationary situations, where neither noise nor channel change too quickly.

4.4.1.2 Minimax Classification

The minimax classification rule strives to minimize the upper bound on the worst case classification error. This results ([Mer1993]) in the classification rule

$$\hat{W} = \arg\max_{W}[\max_{\Lambda\in\Omega} \mathrm{p}(\mathbf{o}|\Lambda, W) \cdot \mathrm{p}_{\Gamma}(W)], \tag{4.76}$$

where $\mathrm{p}_{\Gamma}(W)$ is the language model for the task under consideration and Ω is the range of the uncertain parameters.

4.4.1.3 Bayesian Predictive Classification

Alternatively, in Bayesian predictive classification ([Jia1999]), the probability of a given feature vector $\mathbf{o}$ is computed in two stages. First, the so-called *predictive pdf* is computed for all words, using

$$\hat{p}(\mathbf{o}|W) = \int_{\Omega} \mathrm{p}(\mathbf{o}|W, \mathbf{\Lambda})\mathrm{p}(\mathbf{\Lambda}|W)d\mathbf{\Lambda} \tag{4.77}$$

where W is the word to be recognized and $\mathrm{p}(\mathbf{\Lambda}|W)$ is the, parametrical or nonparametrical, prior pdf learned from training data. Once this probability distribution has been obtained, recognition amounts to finding

$$\hat{W} = \arg\max_{W} \mathrm{p}(W|\mathbf{o}) = \arg\max_{W} \left(\hat{\mathrm{p}}(\mathbf{o}|W) \cdot \mathrm{p}_{\Gamma}(W)\right) \tag{4.78}$$

where $p_{\Gamma}(W)$ stands, again, for the language model. As can be seen, unlike the above described minimax approach, here, not only a single set of parameter values is taken into account, but rather, the entire function of the prior distribution can be utilized for decision making.

4.4.2 Feature Domain Uncertainty Processing

All preceding approaches to processing distorted speech work in two distinct stages, as shown in Fig. 4.13. First, signal preprocessing calculates a point estimate of the speech signal, subsequently, this point estimate is used in the recognition process. During speech preprocessing, a model of the speech signal may well be obtained, for example, an ICA algorithm might estimate signal statistics up to order four, but this statistical information is discarded at the output of the preprocessing stage.

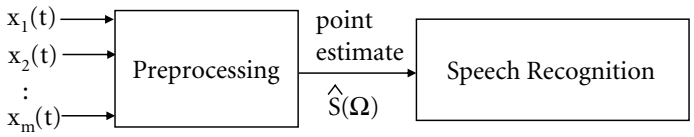

Figure 4.13: Structure of Preprocessing and Recognition.

4.4.2.1 Missing Data Techniques

When the speech data is considered uncertain, so-called missing data techniques can be applied. These can be grouped roughly into two approaches:

- *Imputation Methods* which attempt to recover missing features and
- *Marginalization Techniques* which perform recognition using only the available information.

For marginalization techniques, it is possible either to classify all features into two groups, the reliable and the unreliable ones, or, alternatively, to assign to each unreliable feature $\mathbf{o}_u$ an *uncertainty range* $[\mathbf{o}_{u,min}, \mathbf{o}_{u,max}]$, within which the true value of the feature is expected to lie. In the first case, all those speech features $\mathbf{o}_u$ which have been labeled as unreliable by the preprocessing stage, are disregarded in the recognition process. This can be effected by a modified computation of the feature vector likelihood. In customary speech recognizers, the likelihood that a given speech feature $\mathbf{o}$ was produced by an M-Component MOG is found by

$$p(\mathbf{o}) = \sum_{m=1}^{M} \gamma_m \cdot \mathcal{N}(\mathbf{o}, \mu_m, \Sigma_m). \tag{4.79}$$

Here, γ_m are the mixture weights, and the Gaussian probability associated with mixture m is $\mathcal{N}(\mathbf{o}, \mu_m, \Sigma_m)$. When the observed variables are vectors $\mathbf{o}$, consisting of elements o_i, they can be partitioned into reliable and unreliable components $\mathbf{o}_r$ respectively $\mathbf{o}_u$. The probability of the joint vector $\mathbf{o} = (\mathbf{o}_r, \mathbf{o}_u)$ can then be obtained via integration over all unreliable components as in

$$p(\mathbf{o}) = \int_{-\infty}^{\infty} \sum_{m=1}^{M} \gamma_m \cdot \mathcal{N}(\mathbf{o}, \mu_m, \Sigma_m) d\mathbf{o}_u. \tag{4.80}$$

An actual implementation of this equation causes a great computational effort though, as Equation (4.80) needs to be evaluated at every frame for every HMM state. One major improvement can be obtained when the features are modeled as independent random variables. This simplification allows replacing Equation (4.80) by the expression

$$p(\mathbf{o}) = \int_{-\infty}^{\infty} \sum_{m=1}^{M} \gamma_m \cdot \mathcal{N}(\mathbf{o}, \mu_m, \Sigma_m) d\mathbf{o}_u \tag{4.81}$$

$$= \int_{-\infty}^{\infty} \sum_{m=1}^{M} \gamma_m \cdot \mathcal{N}(\mathbf{o_r}, \mu_{m,r}, \Sigma_{m,r}) \mathcal{N}(\mathbf{o_u}, \mu_{m,u}, \Sigma_{m,u}) d\mathbf{o}_u \tag{4.82}$$

$$= \sum_{m=1}^{M} \gamma_m \cdot \mathcal{N}(\mathbf{o_r}, \mu_{m,r}, \Sigma_{m,r}) \int_{-\infty}^{\infty} \mathcal{N}(\mathbf{o_u}, \mu_{m,u}, \Sigma_{m,u}) d\mathbf{o}_u. \tag{4.83}$$

When the integral in (4.83) is evaluated over the entire range of possible values for $\mathbf{o}_u$, the expression simplifies to

$$p(\mathbf{o}) = \sum_{m=1}^{M} \gamma_m \cdot \mathcal{N}(\mathbf{o_r}, \mu_{m,r}, \Sigma_{m,r}), \tag{4.84}$$

so by integration, the uncertain features are removed completely from calculations. This is the most efficient approach computationally, but using the uncertainty range $[\mathbf{o}_{u,min}, \mathbf{o}_{u,max}]$, a further increase in recognition performance can be obtained [Coo2001]. In this case, $p(\mathbf{o})$ becomes

$$p(\mathbf{o}) = \int_{\mathbf{o}_{u,min}}^{\mathbf{o}_{u,max}} \sum_{m=1}^{M} \gamma_m \cdot \mathcal{N}(\mathbf{o}, \mu_m, \Sigma_m) d\mathbf{o}_u, \tag{4.85}$$

where one reasonable setting for the uncertainty range for spectral features might be

$$0 \leq \mathbf{o}_u \leq \|x\| \tag{4.86}$$

with $\|x\|$ as the observed energy in the speech noise mixture.

4.4.2.2 Marginalization versus Imputation

The idea of marginalization has been proposed by Barker, Green et. al. [Bark2001]. It has been used in conjunction with spectral [Sri2004, Ren2000,

Raj2005] and cepstral [Van2004] features. In marginalization, preprocessing is used to obtain a set of admissible values for the speech features and integration is carried out over this set of values in the recognition stage. In contrast, data imputation can also be employed, which does carry out a point estimation of the speech features, but it is done not at the output of the preprocessing stage (where all input data is known but no prior knowledge on the speech signal is available) but rather inside the recognition stage, using the full speech models. This also allows imputation to be used in speech enhancement as well as in recognition, as it is only necessary to transform the point estimates back into the time domain to compose a maximum likelihood model based speech estimate [Coo2001, Raj2005]. For this purpose, the bounded marginalization equation (4.85) is replaced by an estimate of the unreliable components via

$$\hat{\mathbf{o}}_u = \arg \max_{\mathbf{o}_{u,min} \leq \mathbf{o}_u \leq \mathbf{o}_{u,max}} P(\mathbf{o}_u|\mathbf{o}_r) \tag{4.87}$$

to reconstruct the full feature vector $\mathbf{o}$. This reconstructed vector can then be used for preprocessing as well as for recognition.

4.4.2.3 Uncertainty Decoding

Another way of incorporating feature level uncertainties is described under the name *uncertainty decoding* for example in [Den2005]. In contrast to the model domain uncertainty processing described in Section 4.4.1 ff., and similar to the marginalization and imputation methods above, this approach uses feature level uncertainties, which are computed by an arbitrary preprocessing method in the cepstrum domain. In contrast to the feature domain methods from Section 4.4.2, however, they are incorporated in the recognition process via a strict Bayesian formulation. For this purpose, first, a feature level uncertainty must be described in terms of a probabilistic model $p(\mathbf{s}_{cep}|\mathbf{x}_{cep})$, where the $\mathbf{s}_{cep}$ are the true cepstral features of the clean speech signal whose probabilities are conditioned on the noisy observation vectors $\mathbf{x}_{cep}$. Such a model can be obtained by all statistical processing methods defined on the cepstral domain. The approaches that have been followed so far to obtain the probabilistic model $p(\mathbf{s}_{cep}|\mathbf{x}_{cep})$ are

- a parametric distortion model as in [Den2005] or

- a pre-trained joint distribution of the clean speech cepstrum $\mathbf{s}_{cep}$ and the noisy cepstrum $\mathbf{x}_{cep}$ obtained from stereo training data, in which one channel contains clean and the other noisy data [Dro2002].

Using this conditional feature probability distribution, the word probability can be estimated and maximized via

$$\hat{W} = \arg\max_{W} \int_{\mathbf{s}_{cep}} \mathrm{p}(\mathbf{x}_{cep}|\mathbf{s}_{cep}, \Lambda, W)\mathrm{p}(\mathbf{s}_{cep}|\Lambda, W)\mathrm{d}\mathbf{s}_{cep} \cdot \mathrm{p}_{\Gamma}(W) \tag{4.88}$$

in order to recognize the noisy data. Here, Λ are the fixed model parameters and $\mathrm{p}_{\Gamma}(W)$ stands for the language model. In contrast to imputation, this procedure has the advantage of interpolating between the processing and the recognition model and is closely related to the approach suggested in subsequent chapters.

Chapter 5

ICA for Multi-Speaker Speech Recognition

Whereas multi-channel noise reduction only profits from sensor arrays with M elements via

$$SNR_{post} = SNR_{pre} + 10\log_{10} M \tag{5.1}$$

when the noise is uncorrelated and the desired signal is available and the same at all sensors [Joh1993], directional interferers can in the ideal case be completely canceled when multi-sensor processing is applied.

For this purpose, independent component analysis has emerged as a powerful tool in the past decade [Car1999, Rob2001, Muk2003, Div2005]. Frequency domain ICA can be employed to obtain estimates of clean speech signals in reverberant environments, as is described in more detail in Section 4.2.2.2.

In the following, two enhancements for ICA will be introduced, which allow the incorporation of additional knowledge regarding the structure of speech signals in real environments. The first of these, an EM-algorithm for permutation correction, is described in Section 5.1.1, and the second, time-frequency masking based on ICA results, is presented in Section 5.1.2.

5.1 ICA using prior knowledge

Independent component analysis is one area in the wider field of so called "blind source separation". It is called *blind* because it does not rely on any prior knowledge regarding the structure of the mixing system or the properties of the signal. However, introducing some additional knowledge regarding these two constituents can effectively improve the separation result and to some extent additional information becomes a necessity as soon as real-room signals rather than artificial mixtures are to be separated. Here, two enhancements to ICA are applied, using properties of speech signals as well as the mixing system in car environments:

- In cars, the angle of incidence for each speaker is approximately constant over time as well as over frequency.
- Speech signals are approximately disjoint orthogonal in the frequency domain [Ric2001].

5.1.1 Constant Direction of Arrival

5.1.1.1 Beampattern Analysis

In frequency domain ICA, an estimate of the mixing system (i.e. the room transfer function) is obtained and subsequently inverted for each frequency band Ω. Applying this unmixing system to the signals yields estimates of the short time spectra of the speech signals $\hat{S}_1(\Omega, k) \ldots \hat{S}_N(\Omega, k)$.

In this approach, the major problem consists of permutations between frequency bands, due to which the unmixing performance of the entire system is often negligible before permutation correction. A number of solutions for this problem exist, however, which are discussed in more detail in Section 4.2.2.2.

As also described in that section, a valuable tool for permutation correction is the directional response of the unmixing system, determined by

$$\begin{aligned} \Psi_1(f, \varphi) &= [W_{11}(f)\ W_{12}(f)] \cdot \mathbf{r}(f, \varphi) \\ \Psi_2(f, \varphi) &= [W_{21}(f)\ W_{22}(f)] \cdot \mathbf{r}(f, \varphi) \end{aligned} \tag{5.2}$$

for the two-by-two case and in general, with N sources and M sensors, by

$$\begin{aligned} \Psi_1(f, \varphi) &= [W_{11}(f) \ldots W_{1M}(f)] \cdot \mathbf{r}(f, \varphi) \\ \Psi_2(f, \varphi) &= [W_{21}(f) \ldots W_{2M}(f)] \cdot \mathbf{r}(f, \varphi) \\ &\vdots \\ \Psi_N(f, \varphi) &= [W_{N1}(f) \ldots W_{NM}(f)] \cdot \mathbf{r}(f, \varphi). \end{aligned} \tag{5.3}$$

The directional characteristics of the unmixing filters $\mathbf{W}$ at all frequencies are thus captured in the beampatterns $\Psi_n(f, \varphi)$, which describe the transfer function of the series connection of array response $\mathbf{r}(f, \varphi)$ and unmixing system $W_{n,\cdot}(f)$ for each output n.

Here, the array response is given in the form of the steering vector $\mathbf{r}(f, \varphi)$, which is computed as

$$\mathbf{r}(f, \varphi) = \begin{bmatrix} 1 \\ \frac{a_2}{a_1} e^{-j2\pi f(\delta_2(\varphi) - \delta_1(\varphi))} \\ \vdots \\ \frac{a_M}{a_1} e^{-j2\pi f(\delta_M(\varphi) - \delta_1(\varphi))} \end{bmatrix}. \tag{5.4}$$

This is just the array response given in (4.43), with the only difference that it is normalized with respect to a reference sensor.

In order to obtain the delays $\delta_m(\varphi)$, assuming farfield sound propagation as shown in Figure 5.1 is often a practical solution.

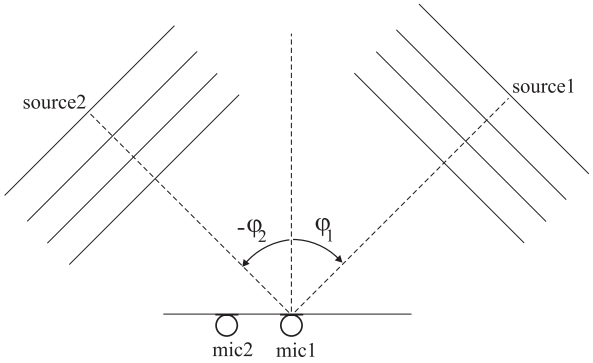

Figure 5.1: Farfield beamforming model

This results in

$$\delta_m(\varphi) - \delta_1(\varphi) = \frac{\mathbf{k}\mathbf{d}_m}{2\pi f} \tag{5.5}$$

with $\mathbf{k}$ as the wavenumber vector and $\mathbf{d}_m$ as the vector connecting sensor 1 to sensor m. For a linear array, with the angle φ measured relative to broadside and an intersensor spacing d, this is equal to

$$\delta_m(\varphi) - \delta_1(\varphi) = \frac{\sin(\varphi) m d}{c} \tag{5.6}$$

for the m'th microphone, resulting in

$$\mathbf{r}(f,\varphi) = \begin{bmatrix} 1 \\ \frac{a_2}{a_1}\exp\left(-j2\pi f\frac{\sin(\varphi)d}{c}\right) \\ \vdots \\ \frac{a_M}{a_1}\exp\left(-j2\pi f\frac{\sin(\varphi)(M-1)d}{c}\right) \end{bmatrix}. \tag{5.7}$$

for the steering vector.

Putting together (5.3) and (5.7), the directivity patterns of a demixing filter $\mathbf{W}(\Omega)$ in the frequency band Ω, $\Omega = 0 \ldots NFFT-1$, can be calculated as a function of the frequency $f = \frac{F_s}{NFFT}\Omega$ and the angle of incidence of the signal via

$$\Psi_n(\Omega,\varphi) = \sum_{k=1}^{M} W_{nk}(\Omega)\frac{a_k}{a_1}\exp\left(-j2\pi\frac{F_s kd\sin(\varphi)\Omega}{NFFT\cdot c}\right) \tag{5.8}$$

Here, $NFFT$ is the number of frequency bins and F_s is the sample rate.

When such beampatterns are computed for the unmixing filters determined by ICA, the result is often similar to the one shown in Figure 4.7. As can be seen, there tends to be a fairly strong directional minimum in each of the bands. This is consistent with the observation, that the unmixing structure of ICA is the same as that of a set of parallel filter and sum beamformers, see Figure 5.2 for a comparison. The only major difference lies in the criterion by which these beamformers are adjusted. In conventional beamforming, adjustment takes place by second order criteria such as minimum interference energy for known interferer directions, whereas ICA unmixing filters are adjusted with a cost function which leads to minimum statistical dependences between outputs under a unit norm constraint. If ICA converges, it often results in a filter and sum beamformer that is effectively a nullbeamformer pointing to the strongest interferer direction(s). Due to these structural similarities, ICA can often give results equivalent to a frequency variant nullbeamformer, with the null directions being adjusted blindly by an information theoretic criterion [Ara2001, Bau2003, Kna2003].

Therefore, important parameters that can be extracted from the beampatterns are the spatial minima, determined by

$$\phi_n(\Omega) = \arg\min_{\varphi} \Psi_n(\Omega,\varphi), \tag{5.9}$$

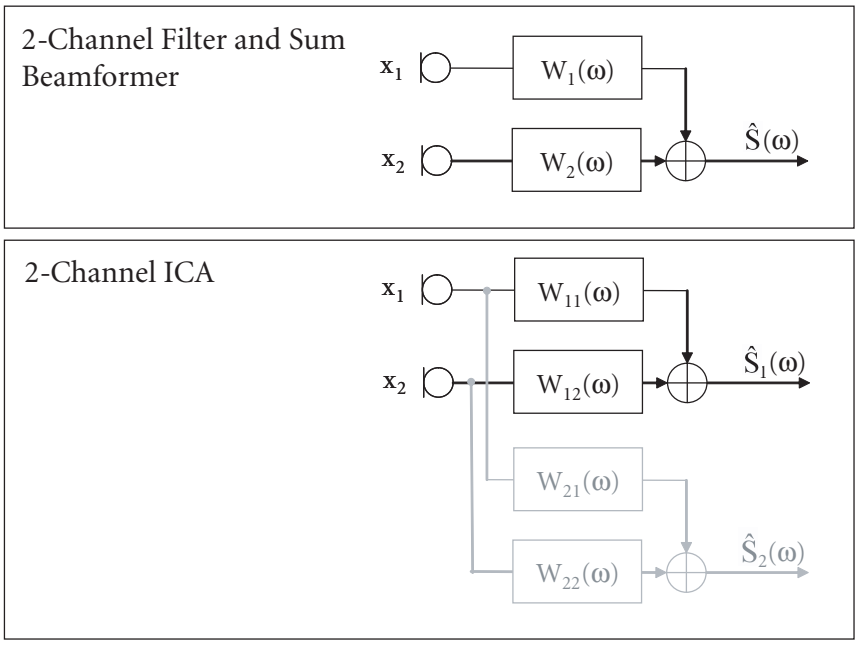

Figure 5.2: Structural comparison between two-channel ICA and beamforming. The parts shown in gray comprise the additional estimator for a further output signal of ICA, which can be seen to be equivalent to a parallelly connected, additional filter and sum beamformer.

which can additionally be mapped to the range in front of the array by

$$\phi' = \max[-90°, \min(\phi, 90°)]. \tag{5.10}$$

Alignment of these directional minima ϕ_n or ϕ'_n is a criterion which has already been successfully used for permutation correction, see e.g. [Kur2000]. However, it is customary to sort frequency bins with a deterministic approach, which consists of putting the directional minima in one set order. Further details of these approaches are given in Section 4.2.2.2. Here, instead, a stochastic direction-of-arrival (DOA) model is employed, which will allow also the computation of confidence values for ICA results in each frequency bin, as well as the use of source directivity models with different spatial characteristics and different degrees of fluctuation around their mean.

5.1.1.2 Permutation Correction by Statistical DOA-Modeling

The result of frequency domain ICA is a set of estimated unmixing matrices $\hat{W}(\Omega)$, one for each frequency bin. The directivity patterns of $\hat{W}(\Omega)$, determined by (5.8), should ideally have one minimum for each output in the two-source two-sensor case. This minimum should point to the direction of the main interfering signal, making unmixing equivalent to frequency domain adaptive nullbeamforming, as was shown e.g. by [Kur2000]. This fact can be used to find and correct permutation errors, which are inherent in ICA.

However, due to phase shifts of the transfer function as well as due to noise, the zero-direction which corresponds to each of the sources is not fixed, but can better be described as the result of a stochastic process. Therefore, it is suggested to use a statistical model for learning the zero direction distribution associated with each source, and, subsequently, to use maximum likelihood classification for detecting permutations and unreliable frequency bands in the unmixing system. To obtain such a model, two choices are necessary, firstly, the choice of a statistical model for the zero directions and secondly, that of an appropriate learning algorithm.

Statistical Model for Null Directions

Once the zero directions of the ICA filters have been obtained from beampattern analysis, each frequency band can be characterized by its vector of zero directions $\mathbf{\Phi}_\Omega = [\phi_{1\Omega} \dots \phi_{N\Omega}]$. For the two-by-two-case, which is considered in the following, the filters of the ICA stage can be grouped into one of three classes - those which extract the sources in the correct order, i.e. in the order they are extracted in the first frequency bin, those which extract the signals in the opposite order, and those that, possibly due to sparsity of signal energy, contain look directions which point to neither source. In order to model the probability density functions for both the permuted and the non-permuted look direction vectors, Gaussian

distributions are assumed, i.e.

$$p_p = \frac{m_p}{2\pi\sqrt{|\Sigma_p|}} \cdot \exp\left(-\frac{1}{2}(\phi - \mu_p)\Sigma_p^{-1}(\phi - \mu_p)^T\right) \tag{5.11}$$

$$p_{np} = \frac{m_{np}}{2\pi\sqrt{|\Sigma_{np}|}} \cdot \exp\left(-\frac{1}{2}(\phi - \mu_{np})\Sigma_{np}^{-1}(\phi - \mu_{np})^T\right) \tag{5.12}$$

$$\tag{5.13}$$

where p_p denotes the probability of $\mathbf{\Phi} = [\phi_1, \phi_2]$ belonging to the class of permuted filters C_p and equivalently p_{np} is the probability of it belonging to the non-permuted class C_{np}. The terms m_{np} and m_p stand for the prior probabilities of non-permuted and permuted bins, respectively, and μ_p, μ_{np} and Σ_p, Σ_{np} are the mean vectors and covariance matrices of the pdfs. An example of such distributions is shown in Figure 5.3 for the mean vectors $\mu_p = [25°, 70°]$ and $\mu_{np} = [70°, 25°]$. As the histograms

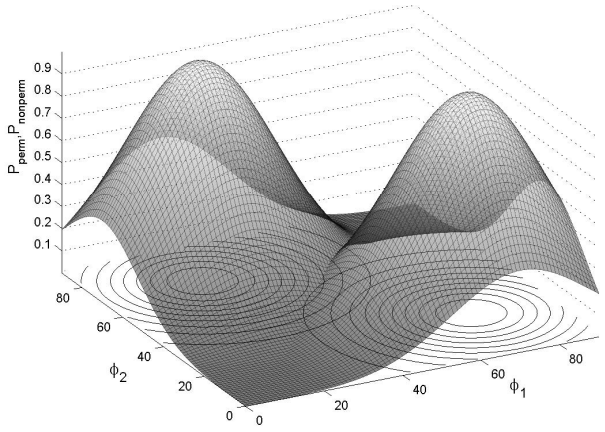

Figure 5.3: Probability density functions of permuted and non-permuted zero-directions.

obtained in noisy and reverberant environments show, cf. Figure 5.4, a Gaussian model is not an exact match. However, since a mixture of Gaussian distributions can approximate an arbitrary distribution [Als1972], and since, with the EM algorithm, a method is readily available for parameter estimation, it was considered as a reasonable choice for a zero direction model.

What can also be seen in Figure 5.4, is that there are outliers at $\phi = -90°$ and 90°, which are significant enough to justify explicit modeling. These outliers are caused by explicit mapping of all angles to the physically reasonable range $-90° \leq \phi \leq 90°$ by (5.10), whereby many of the incorrectly converged frequency bands, such as noisy bands, low energy bands, and frequencies, at which spatial aliasing occurs, all are assigned values of ±90°. In order to be able to model such outliers, the number of Gaussian components in the zero-direction model was increased to four for the two-by-two case. Of these four components, two are the source zero direction models given in (5.11), with the parameters (m_p, μ_p, Σ_p) and $(m_{np}, \mu_{np}, \Sigma_{np})$, and the other two components form a Gaussian mixture model for the unreliable bins, with the probability of the unreliable component computed by

$$p_u = \sum_{i=1}^{2} \frac{m_{ui}}{2\pi\sqrt{|\Sigma_{ui}|}} \cdot \exp\left(-\frac{1}{2}(\phi - \mu_{ui})\Sigma_{ui}^{-1}(\phi - \mu_{ui})^T\right). \tag{5.14}$$

In order for this probability model to be appropriately normalized, the weights m_p, m_{np} and m_{ui} have to fulfill the condition $m_p + m_{np} + \sum_{i=1}^{2} m_{ui} = 1$.

Learning Algorithm

In order to learn the zero direction models, the EM algorithm is used, since it is guaranteed to converge at least to a local optimum [Dem1977]. Because the algorithm needs an initial estimate and convergence to local optima can be avoided by proper choice of a starting point, initialization is an important step.

Initialization

To initialize the means and covariances of the distributions, the histogram of all found zero directions $\phi_{i\Omega}$, $i = 1 \ldots N, \Omega = 1 \ldots NFFT$, is analyzed to obtain the zero directions associated with its two maxima, ϕ_{max1} and ϕ_{max2}. These two angles are used for initialization of the means via $\mu_{np} = [\phi_{max1}, \phi_{max2}]$ and $\mu_p = [\phi_{max2}, \phi_{max1}]$. Subsequently, the first stage of permutation correction estimates the arrangement of all frequency bands Ω by a least squared error criterion, so that all frequency bands are assigned to either the correct or the permuted class by $\min(|\mathbf{\Phi}_\Omega - \mu_p|^2, |\mathbf{\Phi}_\Omega - \mu_{np}|^2)$. From this assignment, the covariance matrices Σ_p

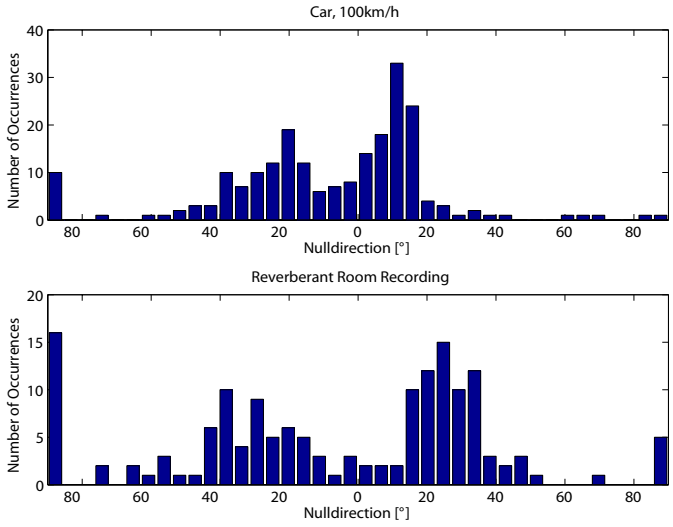

Figure 5.4: Histograms of estimated angles of incidence in a car traveling at 100 km/h, the experiments are described in Section 7.1.2 (top). Histograms of estimated angles of incidence in lab room, configuration A, with t_{rev} = 300ms, the experiments are described in Section 7.1.3 (bottom).

and Σ_{np} of the class distributions p_p and p_{np} can be calculated by

$$\Sigma_p^2 = \sum_{\Omega \in C_p} \frac{(\mathbf{\Phi}_\Omega - \mu_p)(\mathbf{\Phi}_\Omega - \mu_p)^T}{N_p} \quad \text{and} \tag{5.15}$$

$$\Sigma_{np}^2 = \sum_{\Omega \in C_{np}} \frac{(\mathbf{\Phi}_\Omega - \mu_{np})(\mathbf{\Phi}_\Omega - \mu_{np})^T}{N_{np}}, \tag{5.16}$$

with N_p and N_{np} as the numbers of frequency bands Ω assigned to class C_p and C_{np}, respectively. Finally, the unreliable mixtures are initialized to the extrema, thus, $\mu_{u1} = [-90°, 90°]$ and $\mu_{u2} = [90°, -90°]$ and all mixture weights are initialized to $m_p = m_{np} = m_{ui} = 1/4$.

EM Algorithm

The EM-algorithm is used to arrive at three classes of frequency bands, those that are either permuted or not permuted, and those, where the ICA unmixing system

must be considered unreliable. For this purpose, the log-likelihoods of the two source classes, permuted and not permuted, are obtained from

$$\ln \mathrm{p}(\mathbf{\Phi}_\Omega|p) = \ln m_p - \ln 2\pi \sqrt{|\Sigma_p|} - \frac{1}{2}(\mathbf{\Phi}_\Omega - \mu_p)^T \Sigma_p^{-1}(\mathbf{\Phi}_\Omega - \mu_p) \tag{5.17}$$

$$\ln \mathrm{p}(\mathbf{\Phi}_\Omega|np) = \ln m_{np} - \ln 2\pi \sqrt{|\Sigma_{np}|} - \frac{1}{2}(\mathbf{\Phi}_\Omega - \mu_{np})^T \Sigma_{np}^{-1}(\mathbf{\Phi}_\Omega - \mu_{np}) \tag{5.18}$$

for each frequency bin Ω and a Gaussian mixture model with two mixture components is used for the unreliable class via

$$\ln \mathrm{p}(\mathbf{\Phi}_\Omega|u) = \ln \sum_{i=1}^{2} \frac{m_{ui}}{2\pi \sqrt{|\Sigma_{ui}|}} \cdot \exp\left(-\frac{1}{2}(\mathbf{\Phi}_\Omega - \mu_{ui})^T \Sigma_{ui}^{-1}(\mathbf{\Phi}_\Omega - \mu_{ui})\right). \tag{5.19}$$

The iterations are then carried out as follows:

Calculation of Expected Class Labels For each frequency band Ω, calculate the probability $\mathrm{p}(p|\mathbf{\Phi}_\Omega)$ of it belonging to the set of permuted frequency bands, and the probabilities $\mathrm{p}(np|\mathbf{\Phi}_\Omega)$ of the non-permuted and $\mathrm{p}(u|\mathbf{\Phi}_\Omega)$ of the unreliable class. These probabilities are obtained using Bayes' law for each of the three classes $j = \{p, np, u\}$ via

$$\begin{aligned} p(j|\mathbf{\Phi}_\Omega) &= \frac{p(\mathbf{\Phi}_\Omega|j)\mathrm{p}(j)}{\mathrm{p}(\mathbf{\Phi}_\Omega)} \\ &= \frac{p(\mathbf{\Phi}_\Omega|j)\mathrm{p}(j)}{\sum_j \mathrm{p}(\mathbf{\Phi}_\Omega|j)\mathrm{p}(j)}. \end{aligned}$$

To evaluate this term, (5.17), (5.18) and (5.19) are used, and the class prior probabilities $\mathrm{p}(j)$ are equal to the corresponding mixture weights, so $\mathrm{p}(p) = m_p$, $\mathrm{p}(np) = m_{np}$ and $\mathrm{p}(u) = \sum_{i=1}^{2} m_{ui}$.

Density Estimation - Likelihood Maximization Estimate the new probability distributions for the permuted, non-permuted and unreliable class via maximum likelihood estimation. This would not have been possible directly in one step, since the true classes of the frequency bins $\Omega = 1 \ldots NFFT$ are not known, but it becomes possible in the two-step process, where the degree of assignment of each frequency band to each of the classes is estimated by the probabilities $p(j|\mathbf{\Phi}_\Omega)$ from the expectation step above. With these probabilities, the maximum likelihood estimation of all parameters can be carried out in three steps [Vas2001]:

- Update mean values for all classes j via

$$\mu_j^{new} = \frac{\sum_{\Omega=1}^{NFFT} \mathbf{\Phi}_\Omega \mathrm{p}(j|\mathbf{\Phi}_\Omega)}{\sum_{\Omega=1}^{NFFT} p(j|\mathbf{\Phi}_\Omega)} \tag{5.20}$$

- Update covariance values for all classes j via

$$\mathbf{\Sigma}_j^{new} = \frac{\sum_{\Omega=1}^{NFFT} (\mathbf{\Phi}_\Omega - \mu_j^{new})(\mathbf{\Phi}_\Omega - \mu_j^{new})^T \mathrm{p}(j|\mathbf{\Phi}_\Omega)}{\sum_{\Omega=1}^{NFFT} p(j|\mathbf{\Phi}_\Omega)} \tag{5.21}$$

- Update mixture probabilities m_p, m_{np}, m_{ui} for all classes j via

$$m_j^{new} = \frac{\sum_{\Omega=1}^{NFFT} \mathrm{p}(j|\mathbf{\Phi}_\Omega)}{NFFT} \tag{5.22}$$

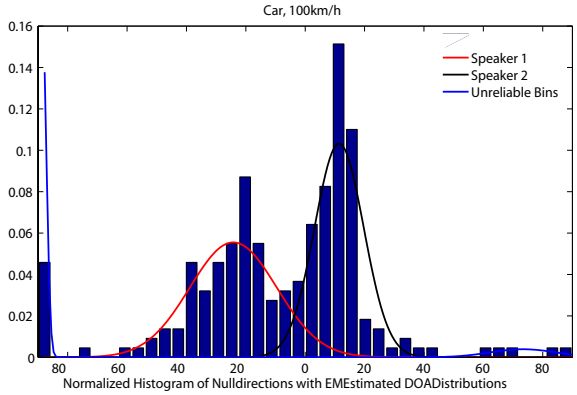

Figure 5.5: Histogram of zero directions for noisy car environment, shown with pdf estimated by EM algorithm.

These steps are iterated until the likelihood converges, which usually happens after 4 to 7 iterations for the considered real-room recordings. After the algorithm has converged, those frequency bands, for which neither of the classes p and np reaches a set confidence threshold γ, so that $\ln \mathrm{p}(\mathbf{\Phi}_\Omega|p) < \gamma$ and $\ln \mathrm{p}(\mathbf{\Phi}_\Omega|np) < \gamma$, are considered unreliable. In these frequency bands, the unmixing filters are replaced by unmixing filters of the nearest frequency band with a reliable permutation estimate. Figures 5.5 and 5.6 show examples of converged pdf estimates for noisy and reverberant recordings.

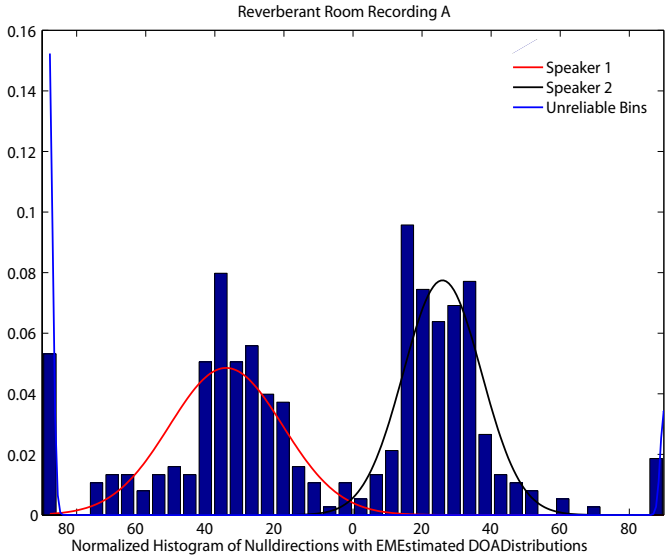

Figure 5.6: Histogram of zero directions for reverberant room recordings, shown with pdf estimated by EM algorithm.

5.1.2 Time-Frequency Masking

5.1.2.1 Theoretical Considerations

Two speech signals tend to have relatively little overlap, when they are transformed into a suitable time-frequency representation. For a more formal definition of this fact, the concept of *W-disjoint orthogonality* is useful.

Two signals $s_1(t)$ and $s_2(t)$ are called *W-disjoint orthogonal*, when the support of their windowed Fourier transforms do not overlap, i.e. when

$$S_1(\Omega, k)S_2(\Omega, k) = 0 \quad \forall k, \Omega, \tag{5.23}$$

for the window function $W(t)$ used to compute the short time spectra $S_1(\Omega, k)$ and $S_2(\Omega, k)$, where k refers to the frame number and Ω to the frequency bin number, ranging from 0 to $\frac{NFFT}{2} + 1$.

This condition does not hold exactly for overlapping speech signals, however, it is true approximately for an appropriate choice of time-frequency repre-

sentation, as shown in [Yil2004]. This property is also referred to as approximate disjoint orthogonality of speech.

Due to this characteristic of speech signals, time-frequency masking (TF-masking), i.e. the multiplication of the spectrum of mixed speech signals with binary masks $M_n(\Omega, k)$, can separate multiple sources from a single mixture $X(\Omega, k)$ according to

$$\begin{aligned} \hat{S}_1(\Omega, k) &= M_1(\Omega, k) \cdot X(\Omega, k) \\ \hat{S}_2(\Omega, k) &= M_2(\Omega, k) \cdot X(\Omega, k) \\ &\vdots \\ \hat{S}_N(\Omega, k) &= M_N(\Omega, k) \cdot X(\Omega, k) \end{aligned} \tag{5.24}$$

without causing excessive distortion.

In order for this technique to be effective, an appropriate time-frequency representation needs to be chosen, for which condition (5.23) is approximately true. In order to find such an appropriate representation, the degree of separability by means of masking needs to be assessed. For this purpose, the two criteria of achievable signal-to-interference-ratio (SIR) improvement ΔSIR and of consequential speech distortion (measured by the signal to distortion ratio SDR) need to be considered simultaneously.

The second of these criteria can be measured for each source $n = 1 \ldots N$ from

$$\hat{s}_n(t) = \text{IFFT}\left(M_n(\Omega, k) \cdot S_n(\Omega, k)\right) \tag{5.25}$$

by

$$SDR_n = 10 \log_{10} \frac{\left\langle s_n(t)^2 \right\rangle}{\left\langle (s_n(t) - \hat{s}_n(t))^2 \right\rangle} \tag{5.26}$$

with $\langle \cdot \rangle$ denoting time averaging.[1] The first criterion is obtained via

$$\Delta SIR = SIR_{out,n} - SIR_{in,n}. \tag{5.27}$$

[1] This computation implicitly assumes that source n is the source extracted on output n, a condition which can not be guaranteed due to the permutation problem. However, it is assumed here that an arbitrary reordering of outputs is not problematic, and that it is hence acceptable to define *source n* as *the source extracted on output n*.

The input SIR $SIR_{in,n}$ is kept fixed at 0dB for the following experiments and the output SIR is determined using the residual interference signals after masking

$$\tilde{s}_m(t) = \text{IFFT}\left(M_n(\Omega, k) \cdot S_m(\Omega, k)\right) \text{ for } m \neq n \tag{5.28}$$

in computing

$$SIR_{out,n} = 10 \log_{10} \frac{\left\langle \hat{s}_n(t)^2 \right\rangle}{\left\langle \sum_{m=1, m\neq n}^{N} \tilde{s}_m(t)^2 \right\rangle}. \tag{5.29}$$

In order to find good parameters for the time-frequency representation, it is interesting to consider the achievable trade-off between possible SIR improvement and consequential distortion in terms of SDR. In order to find such trade-off curves, artificial mixtures have been created by adding speech signals from the TIDigits speech database [LDC1993]. For this purpose, twelve different pairs of speakers have been used, four of them with two male, four with two female and four with speakers of mixed gender. Since the source signals are known, it is then possible to separate them again with an ideal masking function. This ideal mask is based on knowledge of the true source signals and determined via

$$M_n = \begin{cases} 1 & \text{if } 20 \log \frac{|S_n(\Omega,k)|}{|S_m(\Omega,k)|} > \theta_{dB} \ \forall m \neq n \\ 0 & \text{otherwise} \end{cases} \tag{5.30}$$

for various thresholds θ_{dB}.

The resulting trade-off curves depend both on the speaker characteristics and on the chosen time-frequency-representation. To give a concrete example, three such trade-off curves are shown for a 1024-point Hamming window in Figure 5.7. As can be seen, the change in SIR and SDR is smooth, and the choice of a threshold θ_{dB} allows to vary the mask performance smoothly between very effective masking with high distortion and between low-distortion masks which are less aggressive and show a reduced performance in terms of source separation capability. Overall, it can be noted that, at least for the TIDigits database, high SNR gains are possible with little consequential distortion. While this performance depends also on the similarity or dissimilarity between the speakers, as can be seen in the differing results in Figure 5.7 as well as in the spread of results over all cases, displayed in

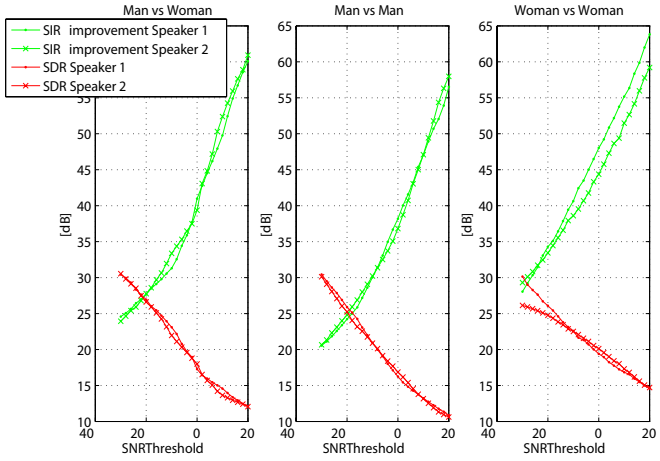

Figure 5.7: Degree of source separation achievable by ideal masking, depicted by SIR-improvement and SDR versus θ_{dB} and tested on speech data of two male speakers, two female speakers and speakers with mixed gender at a frame length of 1024 samples, corresponding to 51.2ms, calculated with a Hamming window.

Figure 5.8, it is very strongly influenced by the choice of time-frequency representation.

As it is also intuitively clear, inappropriate window functions as well as unsuitable frame lengths can have a negative impact on the source separation potential of time-frequency masking. Figure 5.9 shows the influence of the chosen window function and in Figure 5.10, the influence of the frame size is visible. Here, it is also apparent, that the influence of the frame length on the achievable separation is far greater than the difference between the commonly used window functions. Using such information, it is possible to make a selection of a time-frequency representation that is suitable for TF-masking. Here, and for the following experiments, a Hamming window, together with a 1024-sample frame size has been chosen, since the Hamming window has shown good potential performance and the 1024 frame size gives a good trade-off between separation performance and computational effort at the given sample rate.

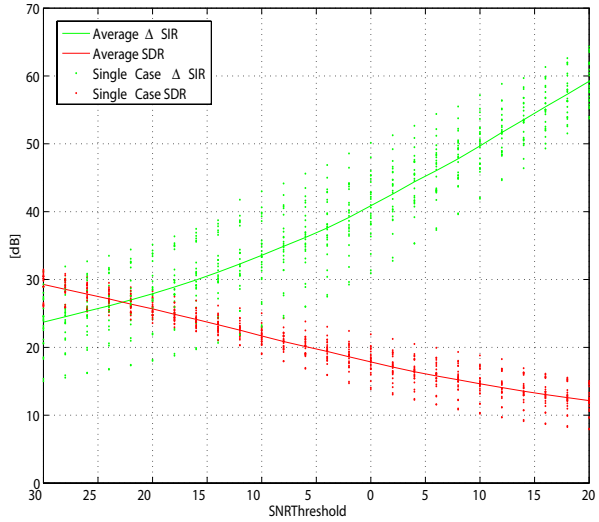

Figure 5.8: Degree of source separation achievable by ideal masking. The spread over all 24 considered speakers can be seen for 1024-sample Hamming window.

5.1.2.2 Practical Implementation

However, an ideal mask such as the one used in the above experiments is not available in practice. Therefore, it is suggested here to compute the time-frequency mask based on results of ICA. This has two advantages. First, ICA gives a set of results, namely the estimated output signals, the estimated mixing and unmixing system per frequency and possibly an estimated DOA, which are useful in deciding which of the sources of interest (if any) is active. These ICA results are obtained by averaging over many time frames, thus they are fairly robust to spurious noise and distortion.

Secondly, using ICA gives separated results for the non-sparse time-frequency points, thus making sparseness less of a requirement for the quality of source separation. This is especially important in the case of noisy mixtures, where it is often very difficult to find good criteria for computing time-frequency masks, but where ICA, due to its inherent robustness to noise, can help to pre-separate sources so that computation of a mask also becomes possible.

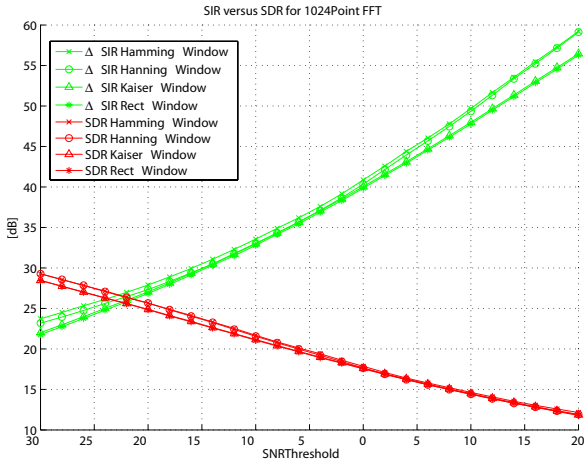

Figure 5.9: Degree of source separation achievable by ideal masking with a window length of 1024 points. The influence of the chosen window function is visible, as can be observed, the Hamming window gives marginally better performance than the other used window functions.

This can be seen especially in phase-damping histograms, which are used to compute masks in some CASA techniques (e.g. [Jou2000]). As already shown in Figures 4.10 and 4.11, whereas such histograms can show very clear peaks for the different speakers, their quality degrades markedly in noisy conditions. In contrast, ICA results obtained using higher order statistics are not affected by the Gaussian noise components at all.

Therefore, ICA is carried out as a first processing step, and it is followed by time-frequency masking to additionally increase separation performance. This two-stage architecture for blind source separation, which is introduced here, is shown in Figure 5.11.

As seen there, the ICA-based estimate is further enhanced by time-frequency masking. For this purpose, in each frequency bin Ω and at each frame k, it is estimated which of the sources is dominant. Based on the assumption of disjoint orthogonality, only one of the outputs should need to have a non-zero value at any given frame and bin. Therefore, only the frequency bin with the dominant source is retained, the other frequency bins are set to zero.

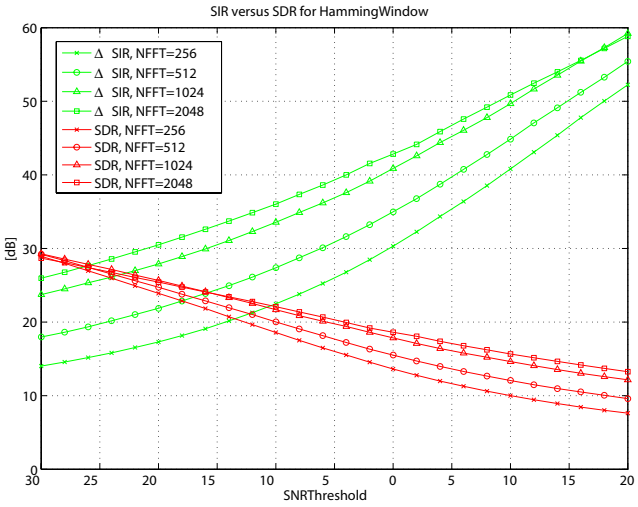

Figure 5.10: Degree of source separation achievable by ideal masking, plotted for different frame lengths which correspond to 12.8ms, 25.6ms, 51.2ms and 102.4ms at the sampling rate of 20kHz.

As a criterion of source dominance, two alternatives were compared.

Amplitude Masking Under the assumption of disjoint support for the speech signals, that source, which is active, should have a larger amplitude than all other sources. Thus, in the first method, the magnitudes of the energy normalized ICA outputs are compared.

Phase Angle Masking If the assumption of disjoint support for the speech signals is correct, the phase angle between the two microphone signals $X_1(\Omega, k)$ and $X_2(\Omega, k)$ should closely correspond to one of the zero directions of the beampatterns for the different speakers. Thus, the current frame angle is compared with all zero directions to find the index of the most probably active source.

5.1.2.3 Overall Strategy

First, the microphone signals are transformed into the time-frequency domain via STFT using a Hamming window of 1024 samples, with a frame shift

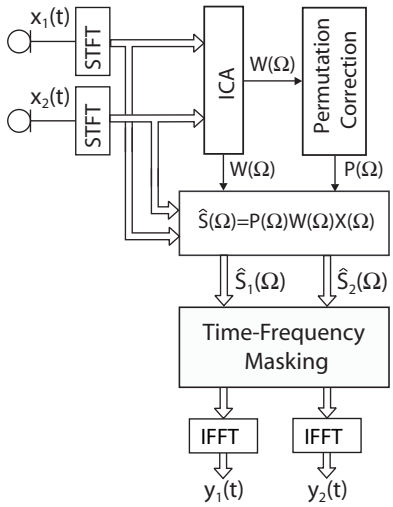

Figure 5.11: Overview of the algorithm

of 25%. In the ICA stage, the unmixing filters $\mathbf{W}(\Omega)$ are determined for each frequency bin. This can be accomplished with any ICA algorithm, provided it operates on complex data. For this work, three different ICA approaches were tested, one of them Parra's algorithm for convolutive source separation described in [Par2000], and two implementations of JADE in the frequency domain with different permutation correction strategies. The unmixing filters, determined by ICA, are applied to the microphone signals to obtain initial speech estimates $\hat{S}(\Omega, k)$. The result of this procedure is, in each channel, a linear, filtered combination of the input signals. Subsequently, masking is carried out according to one of the strategies described below, either Amplitude Masking (see Section 5.1.2.4) or Phase Angle Masking (described in 5.1.2.5). Finally, the unmixed signals $\mathbf{Y}(\Omega, k)$ are transformed back into the time domain using the overlap-add method.

5.1.2.4 Amplitude Masking

In amplitude masking, a time-frequency mask determined from the relative magnitudes of the pre-separated signals is applied to the ICA outputs, as shown in Figure 5.12 for the special case of two signals. More specifically, the time-

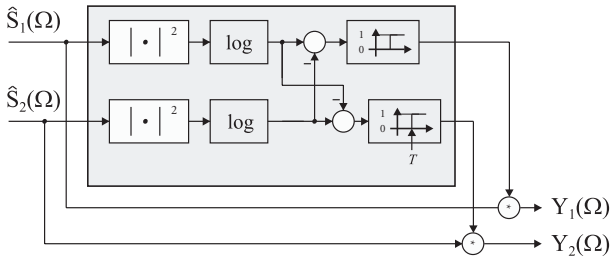

Figure 5.12: Postprocessing for the 2x2 case

frequency mask is determined from the ratio of demixed signal energies, which provides an estimate of the local SNR. The masking function

$$M_i = \Psi\left(\log\left(|\hat{S}_i(\Omega)|^2\right) - \max_{\forall j \neq i} \log\left(|\hat{S}_j(\Omega)|^2\right) - \frac{T}{10}\right) \tag{5.31}$$

is obtained by comparing this SNR-estimate to an acceptance threshold T, with Ψ defined by

$$\Psi(x) = \begin{cases} 0 & \text{for } -\infty \leq x \leq 0, \\ 1 & \text{for } 0 < x < \infty. \end{cases} \tag{5.32}$$

The threshold T was varied between -3dB and 5dB, with higher thresholds leading to better SNR gains but in some test cases to musical noise.

The contrast between this proposed method and the DUET technique introduced in [Jou2000] can be seen from the following Figure 5.13.

5.1.2.5 Phase Angle Masking

An overview of this strategy, which is similar in nature to the strategy described in [Saw2006], is given in Figures 5.14 and 5.15. As can be seen, it also consists of a first stage of ICA and beampattern-based permutation correction. However, rather than the amplitude difference, the phase angle difference between both input signals is used as a criterion for the post-masking function. Thus, the phase-angle-difference is computed first in the masking block, Figure 5.15, and it is converted to an estimated angle of incidence in accordance with the beamforming

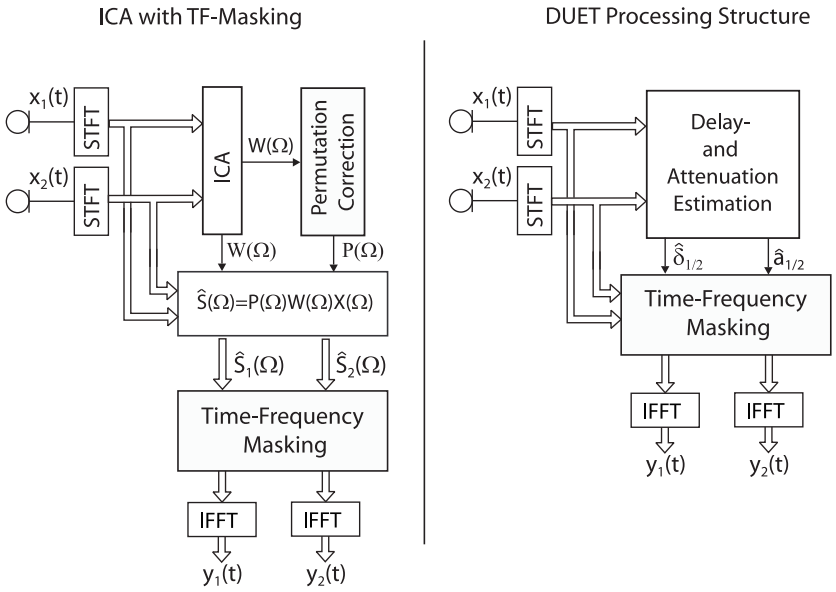

Figure 5.13: Relationship of ICA-based Amplitude-Masking to DUET algorithm.

model (5.4). As seen from (5.4), the phase angle difference between the signal X_1, recorded at the reference microphone, and microphone signal X_2 at frequency f is

$$\arg(X_1(f)) - \arg(X_2(f)) = 2\pi f(\delta_2(\varphi) - \delta_1(\varphi)). \tag{5.33}$$

The time delay difference $\delta_2(\varphi) - \delta_1(\varphi)$ is given by (4.42) as

$$\begin{aligned} \delta_2 - \delta_1 &= \frac{d_2 \sin(\varphi)}{c} - \frac{d_1 \sin(\varphi)}{\mathrm{c}} \\ &= \frac{d \sin(\varphi)}{\mathrm{c}}, \end{aligned} \tag{5.34}$$

where the distances between the two sensors and the reference location, d_1 and d_2, are zero and d respectively, since microphone 1 has been used as the reference point. Therefore

$$\begin{aligned} \arg(X_1(\Omega, k)) - \arg(X_2(\Omega, k)) &= 2\pi f \, \frac{d \sin(\varphi(\Omega, k))}{\mathrm{c}} \\ &= \frac{2\pi F_s}{NFFT} \Omega \, \frac{d \sin(\varphi(\Omega, k))}{\mathrm{c}}. \end{aligned} \tag{5.35}$$

And thus,

$$\begin{aligned}\sin(\varphi(\Omega,k)) &= \frac{\mathrm{c}\left(\arg(X_1(\Omega,k)) - \arg(X_2(\Omega,k))\right)}{\frac{2\pi F_s}{NFFT}\Omega d} \\ \Rightarrow \varphi &= \operatorname{asin}\left(\frac{\mathrm{c}\left(\arg(X_1(\Omega,k)) - \arg(X_2)(\Omega,k)\right)}{\frac{2\pi F_s}{NFFT}\Omega d}\right).\end{aligned}$$

Due to the sparsity assumption, this estimated angle of incidence should correspond well to the angle of incidence of the dominant source signal. Therefore, the decision of which source is likely dominant is made by comparing the estimated DOA in every time-frequency bin with the DOA-models for the sources, which have already been estimated for the purpose of permutation correction in Section 5.1.1.2 by means of the EM-algorithm. These DOA-models are characterized by a mean value μ_n and a variance $\sigma_n^2 = \Sigma_{nn}$ for each source n. Using both the mean and

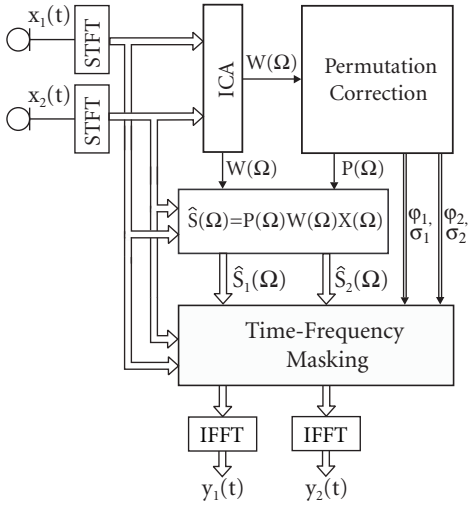

Figure 5.14: Overview of Phase-Masking

the variance, the likelihood of source activity for each time-frequency bin can be assessed via the Mahalanobis distance $d_m(\varphi, i)$ between the observed angle of arrival and the i'th source model

$$p(S_i \text{ active in } (\Omega,k)) \propto d_m\left(\varphi(\Omega,k), i\right)^2 = \left(\varphi(\Omega,k) - \varphi_i\right)^2 / \sigma_i^2. \tag{5.36}$$

By comparing $p(S_i \text{ active in } (\Omega, k))$ for all sources, it is possible to mask out the output value for less likely sources. More precisely, this decision was made by comparing the Mahalanobis distance d_m of the current estimated DOA with the probability model associated with each of the sources according to

$$d_m(\varphi(\Omega, k), i) = (\varphi(\Omega, k) - \varphi_i)/\sigma_i. \tag{5.37}$$

Based on the comparison of Mahalanobis distances, only those sources are retained, whose Mahalanobis distance is not greater than the smallest Mahalanobis distance

$$d_m(\varphi(\Omega, k), \text{opt}) = (\varphi(\Omega, k) - \varphi_{opt})/\sigma_{opt} = \min_i d_m(\varphi(\Omega, k), i). \tag{5.38}$$

by more than a threshold factor Thr. In the case when $Thr = 1$, this actually approximates a maximum likelihood decision, since the source with the smallest Mahalanobis distance is retained and all others are discarded. The entire procedure is also illustrated in Figure 5.15 for the case of two sources.

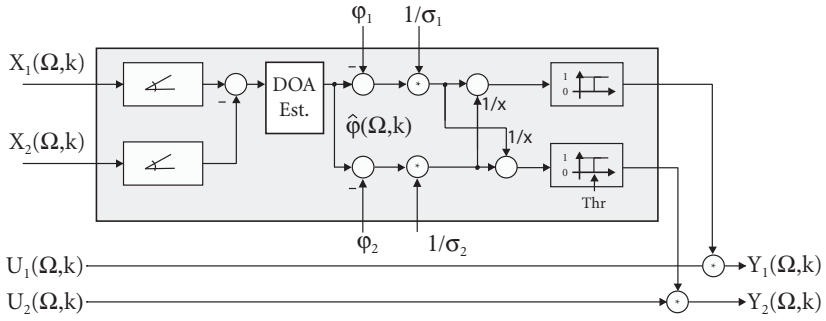

Figure 5.15: Postprocessing by DOA-based masking for the 2x2 case.

Chapter 6

Connecting Speech Processing and Recognition by Uncertain Feature Transformation and Missing Data Techniques

6.1 Framework

The most common approach for recognizing distorted speech consists of two stages. First, the speech signal is enhanced, and if array processing is applied, this is the stage where all input signals are combined into a single speech estimate $\hat{s}$. Speech recognition then runs independently of the first processing stage as illustrated in 6.1.

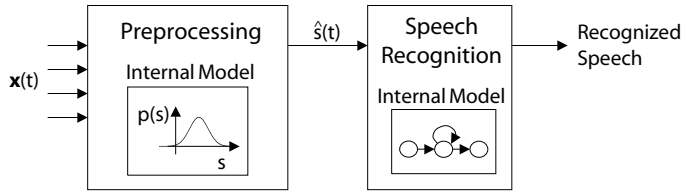

Figure 6.1: Signal Preprocessing and Recognition

While this approach has its advantages in the independent design of both subsystems and in easy, separate block testing, it is suboptimal in its use of available information. Many of the most successful approaches for speech processing are based on stochastic speech modeling, that is, they learn a model of the speech and possibly noise probability distribution from the data in order to obtain an optimal signal estimate. This estimate is often calculated using statistically motivated optimality criteria, such as MMSE or maximum likelihood, and is this way, a point estimate $\hat{s}$ of the speech signal is obtained, which is optimal with respect to the sensor data [Bit1999, Aal2002, Kou2002, Seg2002, Asa2003]. In the next step,

the point estimate of speech is passed on to the recognition engine, which has its own reference, in the form of another stochastic speech model. The point estimate from the preprocessing (won from a statistical model) is thus compared to another statistical model.

In this way, much of the information gathered in the preprocessing stage is disregarded in the decoding phase and at the same time, the speech recognizer models, which could contribute to make preprocessing more robust, are disregarded in the signal processing. In this thesis, it is suggested to use an integrated approach to speech processing and recognition in the form shown in Figure 6.2. Here, rather than carrying out point estimation in the preprocessing stage, instead, a statistical model of the speech signal is passed from the signal preprocessing to the recognizer. Thus, it is possible to utilize knowledge about the varying degree of estimation quality in the recognition stage, for example by regarding highly accurate segments as more significant for recognition, and discarding unreliable parts. Also, this structure allows estimation of the speech signal to take place in the recognition stage, where information from signal preprocessing can be integrated with information from the often detailed recognition models. In the following, the focus will be only on the speech recognition itself, though, but it should be mentioned here, that this possibility of improved speech estimation is a second advantage of this *soft interface* between signal processing and speech recognition.

In the following chapter, first, this extended interface between signal processing and speech recognition is described in more detail in Sections 6.2 through 6.3, and subsequently, the use of the statistical speech model for recognition purposes is detailed in Section 6.4.

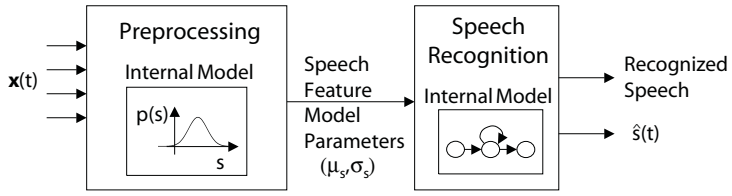

Figure 6.2: Conserving statistical information in the interface between signal preprocessing and recognition

6.2 Interfacing Signal Processing and Speech Recognition

When the conventional processing structure shown in 6.1 is used, i.e. when signal processing is used to estimate the speech waveform or spectrum, which is then passed on to the recognizer as a best point estimate, recognition rates can suffer from nonlinear postprocessing. The following Figures 6.3 and 6.4 show the development of SDR, SIR and recognition accuracy for three exemplary datasets, more data can be found in the quantitative evaluation in Chapter 7.

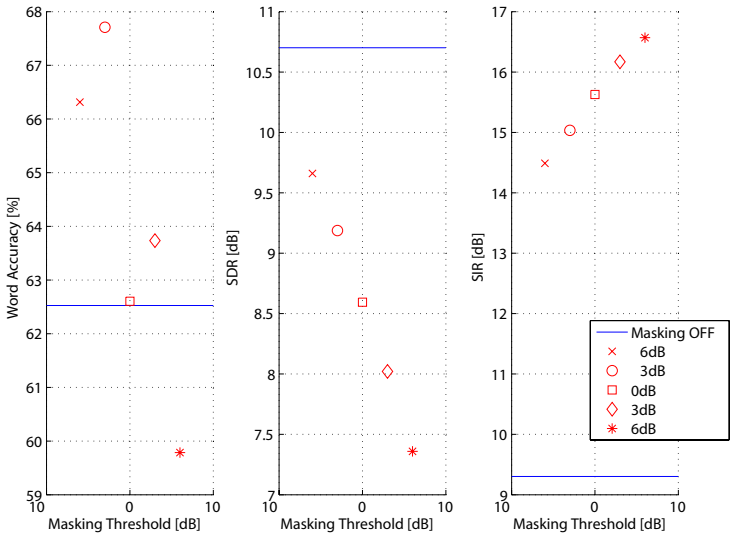

Figure 6.3: Effect of amplitude masking on recognition accuracy, signal to distortion ratio and signal to interference ratio, obtained for mixtures of a male and a female speaker, recorded in a car at standstill (dataset 2c).

As can be seen, despite the noticeable improvement in separation performance, as measured by the SIR, for some datasets the best recognition rates are achieved when no nonlinear postprocessing is applied at all. However, postprocessing does increase the SIR significantly, so the cause for this effect most probably lies in the additionally introduced distortions, i.e. in the inaccurate estimates of the speech signal in those time-frequency points which are masked. Figure 6.5 shows

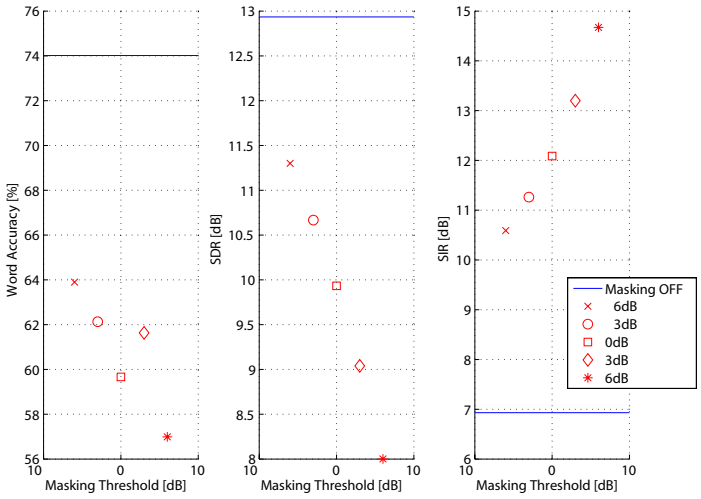

Figure 6.4: Effect of amplitude masking on recognition accuracy, signal to distortion ratio and signal to interference ratio, obtained for an artificial, anechoic mixture of two male speakers (dataset 1a).

the effect of time-frequency masking on the spectrum of speech, which visibly illustrates the effect.

A way to deal with this problem can be found in the use of missing data techniques for speech recognition, which are described in Subsection 4.4.2.1. As it is detailed at that point, missing data recognizers use two quantities for calculating the probability of a given set of speech features,

- the feature estimate itself
- and the associated reliability of each feature.

In classic binary missing data recognition, a distinction is made between reliable and unreliable features, and only those features which are considered reliable are used to evaluate the likelihood of the current speech feature vector. In the extended approach of *bounded marginalization*, the degree, to which a feature vector component influences the recognition result, increases gradually with the associated

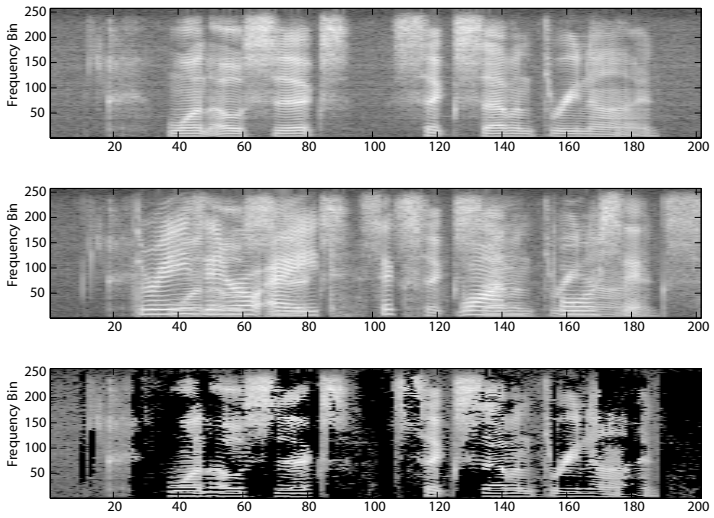

Figure 6.5: Effect of time-frequency masking on speech spectrogram.

reliability of the feature component [Coo2001]. Successful application of missing data techniques have been described for example in [Bark2001, Coo2001, Sri2004, Ren2000, Raj2005]. However, there is one significant difficulty which arises when missing data techniques are used.

- Which features are missing or uncertain is usually determined in a suitable time-frequency representation. This is true for noise reduction, for array processing and for source separation methods. Thus, missing feature recognition is also carried out in time-frequency domain, where it is known which features are certain (for binary missing data methods) or to which degree they are uncertain (for bounded marginalization).

- Speech recognition is most robust on mel-frequency cepstral coefficients [Dav1980], PLP cepstra [Her1990] or other specialized features (e.g. [Kle2002, Mor2005]). This is due to a large part to a greater robustness of cepstral or PLP features to variations in the room impulse response and the vocal tract characteristics of the speaker, as well as to a reduction of redundant and irrelevant information together with a concentration on auditorily relevant characteristics of the speech signal [Hua2001].

- Thus, missing data techniques are usually applied in a domain which is suboptimal for speech recognition. This leads to overall reduced performance, when missing data techniques applied in the spectrum domain are compared to feature reconstruction in the spectrum domain together with recognition in the mel-frequency cepstrum [Raj2004].

The alternative which is suggested here is the following:

- Preprocessing yields uncertainty or variance values for each feature. From these values, which are given in the spectrum domain, the variances of features in the recognition domain can be calculated, when the spectrum domain features are considered as random variables, and when the effect of the transformation of these variables from the spectrum to the recognition domain is determined. In this way, uncertainty information from the preprocessing stage can be passed onto the recognizer, and still recognition can be carried out with an appropriate feature set, e.g. MFCCs plus δ and acceleration coefficients.

- In the recognition stage, features are assumed to be approximately Gaussian. Using the feature mean and variance, modified imputation is proposed as an alternative to marginalization by integration.

An overview of the processing stages suggested here is given in Figure 6.6. Both

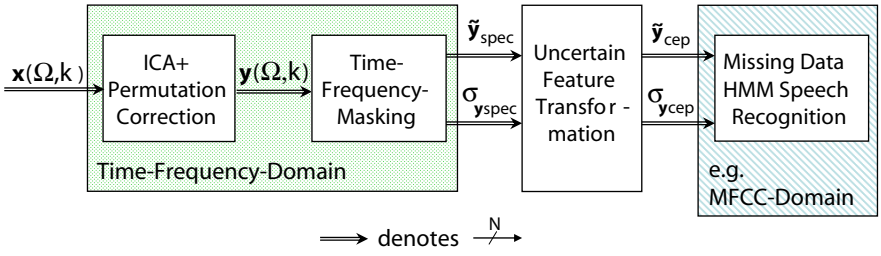

Figure 6.6: Overview of the data flow for ICA, time-frequency masking, uncertain feature transformation and recognition.

uncertainty-related methods, the transformation of uncertain features from the domain of preprocessing to the domain of speech recognition, and the subsequent use

of uncertainty values in modified imputation, are described in the following Sections 6.2.1 through 6.4.

6.2.1 Interfacing ICA to Missing Data Speech Recognition

ICA is carried out in the frequency domain, therefore, its outputs are two or more speech signal estimates at each time-frequency point $\hat{S}_1(\Omega,\tau)$,$\hat{S}_2(\Omega,\tau)$. At this point, there are two possible strategies, hard masking and choice of a soft mask.

In the first case, the means and variances could be estimated as follows:

$$\left.\begin{array}{ll}\mu_{S1} & = \hat{S}_1 \\ \sigma_{S1} & = 0\end{array}\right\} \text{if } 20\log_{10}|\hat{S}_1| > 20\log_{10}|\hat{S}_2| - \theta_{dB}$$
$$\left.\begin{array}{ll}\mu_{S1} & = 0 \\ \sigma_{S1} & = \infty\end{array}\right\} \text{if } 20\log_{10}|\hat{S}_1| \leq 20\log_{10}|\hat{S}_2| - \theta_{dB} \qquad (6.1)$$

and equivalently for μ_{S2} and σ_{S2}, where θ_{dB} is the threshold for amplitude masking defined in (5.30). This is equivalent to the algorithm proposed by Rickard et al. [Ric2001], where dominated frequency bands are masked completely.

Alternatively, the following more general strategy may be used to estimate means and variances:

$$\left.\begin{array}{ll}\mu_{S1} & = \hat{S}_1 \\ \sigma_{S1} & = 0\end{array}\right\} \text{if } 20\log_{10}|\hat{S}_1| > 20\log_{10}|\hat{S}_2| - \theta_{dB}$$
$$\left.\begin{array}{ll}\mu_{S1} & = d_m \cdot \hat{S}_1 \\ \sigma_{S1} & = p_e \cdot \hat{S}_1\end{array}\right\} \text{if } 20\log_{10}|\hat{S}_1| \leq 20\log_{10}|\hat{S}_2| - \theta_{dB} \qquad (6.2)$$

with p_e standing for the probability of erroneously masking out any given time-frequency point and d_m a mask damping factor. In this case, the estimates in all masked time-frequency points are augmented by uncertainty values sigma, which can also be considered as reliability measures specifically computed for each feature. Thus, at each point in time and frequency, not only an estimate of the signal but also one of the signal quality is available.

6.3 Feature Transformation

In the transformation stage, spectral domain representations are transformed to the chosen speech recognition features. Here,

- mel frequency cepstral coefficients (MFCCs) together with
- delta coefficients and
- acceleration coefficients

were chosen due to their robustness and widespread use [Hua2001]. The following flow diagram (Fig. 6.7) shows the necessary computations. Since the spectral

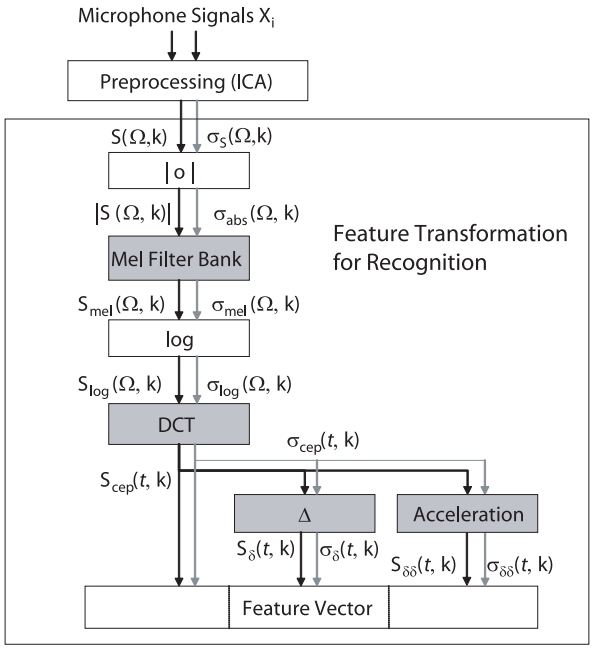

Figure 6.7: Feature Extraction for Recognition

features are not known exactly, the feature transformation is not carried out on numerical values but rather on estimated probability distributions. The most general formulation of this problem would start out with an arbitrary probability distribution and proceed to compute the probability distributions of all features by numerical integration, i.e. if the feature vector $\mathbf{m}$ is transformed to a vector $\mathbf{n}$ by the generally nonlinear function T via

$$\mathbf{n} = T(\mathbf{m}), \tag{6.3}$$

then the cdf of $\mathbf{n}$ can be obtained via

$$\mathrm{P}(\mathbf{n} < N) = \int_{\mathbf{m}:T(\mathbf{m})<N} \mathrm{p}_m(\mathbf{m})\mathrm{d}\mathbf{m} \tag{6.4}$$

and

$$\mathrm{p}_n(\mathbf{n}) = \frac{\mathrm{dP}(\mathbf{n} < N)}{\mathrm{d}N}, \tag{6.5}$$

when differentiability can be assumed. For invertible transformations T with continuous partial derivatives and nonzero Jacobians J, this can also be obtained via

$$p_n(\mathbf{n}) = p_m(\mathbf{m})|J(T(\mathbf{m}))|^{-1} \tag{6.6}$$

cf. [Ros1997].

However, carrying out these computations and working with entire pdfs in an online algorithm is infeasible. Therefore, two alternatives were investigated. Both approximations assume Gaussian distributions of speech vectors around their mean, so that all probability distributions are adequately characterized using only the first two moments, mean and variance. Thus, the computations consist of calculating the effect that each transformation stage has on the first two moments of a Gaussian input signal. This is simple for the case of linear transforms, which are shown shaded in Figure 6.7, and in order to be as precise and efficient as possible, the linear transformations were explicitly calculated as described in Section 6.3.1. But for nonlinear transforms, (6.5) is not easy to solve in general, therefore the effect of these transformations was approximated in two different ways, once by using Monte Carlo simulation and secondly by applying the unscented transform. The necessary equations are given for Monte Carlo simulation in 6.3.2 and for the unscented transform in Section 6.3.3. But before the computations are detailed for the transformation of random variables from the domain of preprocessing to the domain of the speech recognition features, the remainder of this section details the computations that are necessary to obtain MFCC features from spectral features when no uncertainties need to be considered.

Absolute Value

Frequency domain source separation and noise reduction techniques work on the complex speech spectrum. Thus, the result is a complex-valued estimate

$\mathbf{S}(\Omega, k)$ together with a real-valued estimate for the variance $\sigma^2(\Omega, k)$. The complex ICA output values are then transformed to yield absolute value mean $|\mathbf{S}(\Omega, k)|$ and variance $\sigma^2_{abs}(\Omega, k)$.

Mel-Scaled Filter Bank

The mel-scaled filter bank is used to transform the signal representation from a linear scale to perceptually motivated one, in which human hearing characteristics are modeled (c.f. Section 2.1.2.1). In effect, this is equivalent to a matrix multiplication of the spectral feature vector $|\mathbf{S}(\Omega, k)|$ at each time index k with the mel filterbank parameters:

$$\mathbf{S}_{mel}(\omega_{mel}, k) = \mathbf{M} \cdot \mathbf{S}(\Omega, k), \tag{6.7}$$

with $\mathbf{M}$ defined as the matrix of frequency weights. These are triangle shaped filters shown in Figure 6.8.

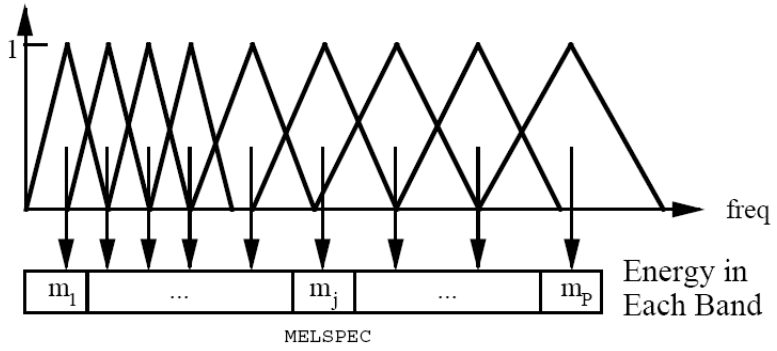

Figure 6.8: Shape of subband filters used for mel-scaled filterbank, from [You2002].

6.3.0.1 Logarithm

After the mel-scaled signal $\mathbf{S}_{mel}(\omega_{mel}, k)$ has been computed, its logarithm is taken.

$$\mathbf{S}_{log}(\omega_{mel}, k) = \log\left(\mathbf{S}_{mel}(\omega_{mel}, k)\right). \tag{6.8}$$

DCT

The mel-scale signals are then transformed into the cepstral (or "quefrency") domain by multiplying them with the DCT matrix $\mathbf{C}$ defined above in (3.8):

$$\mathbf{S}_{cep}(\cdot, k) = \mathbf{C} \cdot \mathbf{S}_{log}(\cdot, k), \tag{6.9}$$

again for each time index k. The matrix $\mathbf{C}$ need not be square. Since a DCT is applied to a signal in a spectrum domain, it is customary to regard the output dimension of the cepstral transformation as a dimension of time[1], therefore, the cepstrum is denoted as $S_{cep}(t, k)$ in the following.

6.3.0.2 Delta and Acceleration Coefficients

Delta coefficients show how quickly speech features change and are of importance in differentiating transients from stationary phonemes. They are defined by

$$\mathbf{S}_{\delta}(t, k) = \frac{\mathbf{S}_{cep}(t, k+1) - \mathbf{S}_{cep}(t, k-1)}{2} \tag{6.10}$$

when only two frames are used for their computation. In a more general way, they can also be computed from a set of 2Θ frames using regression. In that case

$$\mathbf{S}_{\delta}(t, k) = \frac{\sum_{\theta=1}^{\Theta} \theta \cdot (\mathbf{S}_{cep}(t, k+\theta) - \mathbf{S}_{cep}(t, k-\theta))}{\sum_{\theta=1}^{\Theta} \theta^2} \tag{6.11}$$

is the linear regression formula (see [Bro2001], p. 802). The acceleration coefficients $\mathbf{S}_{\delta\delta}(t, k)$ are computed in a similar manner, only using the delta coefficients rather than the static features as inputs.

6.3.1 Analytic Computation for Linear Transformations

The effect of a matrix multiplication on mean and variance of a stochastic variable $\mathbf{z} = \mathbf{M}\mathbf{y}$ are:

$$\mu_{\mathbf{z}} = \mathbf{M} \cdot \mu_{\mathbf{y}}, \tag{6.12}$$

[1] This can be justified well for the case of a cepstrum computed on log spectral, rather than log mel-spectral features. In that case, the cepstrum can be interpreted as a transformation to the domain of *time delays*, as derived e.g. in [Opp2004].

and

$$\mathbf{\Sigma_z} = \mathbf{M}\mathbf{\Sigma_y}\mathbf{M}^T, \tag{6.13}$$

as derived in Appendix A.2.

6.3.2 Monte Carlo Sampling

At each stage of the feature transformations, a probability distribution for the feature of interest $p_y(\mathbf{y})$ is given and the probability distribution of the transformed feature $\mathbf{z} = T(\mathbf{y})$ is desired. In the above approach for linear transformations, the computations for $p_z(\mathbf{z})$ are carried out explicitly. This has the advantage of low computational effort, however, it is inflexible insofar as

- only Gaussian distributions for the features are considered at all stages of the transformation and
- while for the linear stages of transformation (i.e. the mel filterbank, the DCT and all calculations of derivatives) mathematical analysis is efficient and simple, as shown in Appendix A.2, all nonlinear transformations cause the difficulty of calculating $p_z(\mathbf{z})$ according to (6.4) and (6.5).

Monte Carlo sampling offers an alternative solution by randomized simulations of the nonlinear transformation stages. For this purpose, a number N of samples is generated which share the probability distribution $p_y(\mathbf{y})$ of the input features. Since a Gaussian model for the noise is assumed, it is easy to generate samples with the given probability distribution. These samples are then transformed through the given nonlinearity (which is the absolute value and the logarithm) and the desired statistics of $p_z(\mathbf{z})$ are calculated from the result of the transformations. When second order statistics are sufficient, mean and variance of $\mathbf{z}$ are the quantities of interest, and can be estimated by approximating the expectations

$$\mathrm{E}\left(T(\mathbf{y})\right) \approx \frac{1}{N}\sum_{n=1}^{N} T(\mathbf{y}) \tag{6.14}$$

and

$$\mathrm{E}\left(T(\mathbf{y})^2\right) \approx \frac{1}{N}\sum_{n=1}^{N} T(\mathbf{y})^2 \tag{6.15}$$

and using

$$\begin{aligned} \hat{\mu}_z &= \mathrm{E}\left(T(\mathbf{y})\right) \text{ and} \\ \hat{\sigma}_z &= \mathrm{E}\left(T(\mathbf{y})^2\right) - \mathrm{E}\left(T(\mathbf{y})\right)^2 \end{aligned} \tag{6.16}$$

as estimators. These estimators for the mean and variance are unbiased and their variance decreases with N via

$$\mathrm{var}(\hat{\mu}_z) \propto \mathrm{var}(\mu_z)/N \tag{6.17}$$

and

$$\mathrm{var}(\hat{\sigma}_z) \propto \mathrm{var}(\sigma_z)/N \tag{6.18}$$

according to [Mac2003].

6.3.3 Unscented Transform

Finally, a more efficient approach for transforming features was also investigated. The so-called *Unscented Transform* has been suggested as a way of computing the effect which nonlinear transformations have on stochastic variables. Originally, it has been used in a control systems context, where it was first used as an alternative to the Extended Kalman Filter by Julier and Uhlmann [Jul1996].

In this context, the unscented transform was used for the nonlinear state estimation problem, where a system state $\mathbf{x}(t)$ is often estimated up to second order moments and a prediction for the next time step is needed. Thus, an n-dimensional random variable $\mathbf{x}(t)$, describing the system state with mean μ_x and covariance Σ_{xx} is transformed by the nonlinear system via $\mathbf{x}(t+1) = g(\mathbf{x}(t))$ and the statistics of the state $\mathbf{x}(t+1)$ are desired.

This is very similar to the problem encountered when uncertain speech features $\mathbf{x}$ are transformed from the preprocessing domain to the recognition domain, so the unscented transform was also tested here. In detail, it is comprised of the following steps:

- Given the n-dimensional distribution of processing features $\mathbf{x}$, a set of so-called *sigma points* is calculated, which capture the statistics of the features up to the desired order. For this purpose,

– determine the scaled matrix square root P_{xx} of the input covariance matrix Σ_{xx} via $P_{xx} = \sqrt{(n+\kappa)\Sigma_{xx}}$, where n stands for the dimension of the vector $\mathbf{x}$ and κ is an adjustable scale parameter,

– denote the i'th column of P_{xx} by $(P_{xx})_{\cdot,i}$,

– select $2n+1$ so-called sigma points $\mathcal{S}_i = \{\mathcal{X}_i, \mathcal{W}_i\}$ with their associated values $\mathcal{X}_i$ and weights $\mathcal{W}_i$ via

$$\begin{aligned} \mathcal{X}_0 &= \overline{\mathbf{x}} & \mathcal{W}_0 &= \tfrac{\kappa}{n+\kappa} & i &= 0 \\ \mathcal{X}_i &= \overline{\mathbf{x}} + (P_{xx})_{\cdot,i} & \mathcal{W}_i &= \tfrac{1}{2(n+\kappa)} & i &= 1,\ldots,n \\ \mathcal{X}_i &= \overline{\mathbf{x}} - (P_{xx})_{\cdot,i-n} & \mathcal{W}_i &= \tfrac{1}{2(n+\kappa)} & i &= n+1,\ldots,2n. \end{aligned}$$

This set of $2n+1$ points has weighted mean μ_x and covariance Σ_{xx}. For a proof, see Appendix A.3.

- The sigma points are propagated through the nonlinearity to form a set of transformed points $\mathcal{Y} = g(\mathcal{X})$.

- The second order statistics of $\mathbf{y} = g(\mathbf{x})$ are then approximated by the weighted mean $\overline{\mathcal{Y}}$ and covariance $\Sigma_{\mathcal{Y}\mathcal{Y}}$ of the transformed set $\mathcal{Y}$ according to

$$\overline{y} \approx \sum_{i=0}^{2n} \mathcal{W}_i \mathcal{Y}_i \tag{6.19}$$

$$\Sigma_{yy} \approx \sum_{i=0}^{2n} \mathcal{W}_i (\mathcal{Y}_i - \overline{y})(\mathcal{Y}_i - \overline{y})^T. \tag{6.20}$$

This approach is also illustrated in Figure 6.9.

Compared to Monte Carlo simulation, this algorithm has the advantage of efficiency: Whereas a large set of points needs to be simulated to obtain low errors in Monte Carlo simulation, using the unscented transform, only $2n+1$ points are simulated for each feature vector, where n is the feature vector size. In the concrete case of transforming features from a 512-dimensional spectrum with diagonal covariances via 29-dimensional mel spectra to 39-dimensional cepstral features with

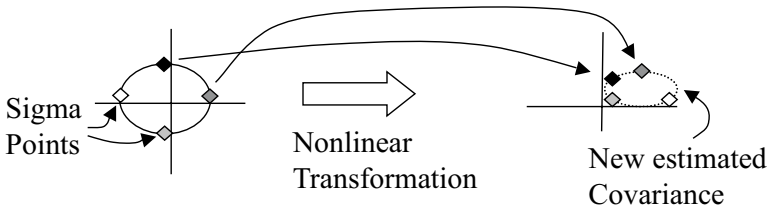

Figure 6.9: The sigma points of the signal probability distribution are transformed to obtain an estimate of the statistics after the transformation.

first and second derivatives, the improvement in computational effort was measured with the MATLAB profiler to be a reduction by the factor 14.7.

The cost of these savings are increased estimation errors in the higher order terms. However, when the scaling factor κ is optimized, the errors can be reduced, and with a higher number of generated sigma points, they can even be eliminated up to an arbitrary order at the cost of additional computational effort. An estimate of the errors can be found from the Taylor series expansion of $g(\mathbf{x})$ around μ_x, and, as shown in [Jul1996], for $\kappa = 3 - n$ they are minimal and of sixth order, i.e. the statistics up to order four, being mean, covariance and kurtosis, of $g(\mathbf{x})$ are matched. Further reductions of higher order errors may also be achieved by the scaled unscented transform [Mer2000]. However, since knowledge of higher order statistics must be used for optimization of the scale parameter in this method, and since higher order statistics for speech feature vectors are strongly dependent on the (unknown) phoneme, this method was not considered for implementation here.

6.4 Recognition

In HMM speech recognition, the probability of a given vector of speech features is evaluated at each frame for all HMM states, as detailed in Section 2.2. For this purpose, the model states are equipped with output probability distributions. These distributions are denoted by b_q, where $b_q(\mathbf{o})$ gives the probability that observation vector $\mathbf{o}$ will occur at time t, when the Markov Model is known to be in state q at that time, so:

$$b_q(\mathbf{o}) = \mathrm{p}(\mathbf{o}_t = \mathbf{o} | q_t = q). \tag{6.21}$$

For the recognition of given, fixed observation vectors $\mathbf{o}$, the probability distribution b_q can be evaluated for the available vector $\mathbf{o}$. This is the customary computation of output probabilities, denoted by $p_{o|q}(\mathbf{o}|q)$. With additional information from the preprocessing stage, however, rather than only the observation $\mathbf{o}$, its entire pdf $p_{o|x}(\mathbf{o}|\mathbf{x})$ is known.

Thus, a new approach for calculating observation likelihoods is needed, which incorporates all available information: the output probability distributions of the states as well as the observation probability distributions obtained from the preprocessing stage. For this purpose, a new method is suggested, which is termed *modified imputation* and described in the following section.

6.4.1 Modified Imputation

To evaluate the likelihood of an HMM state, it is necessary to combine the likelihood of the current observation $p_{o|q}(\mathbf{o}|q)$ given all possible HMM states q with the likelihood of the observation given the preprocessing model $p_{o|x}(\mathbf{o}|\mathbf{x})$.

A Gaussian distribution for the speech features was assumed in all previously described stages of processing. Thus, a model of the observation probability has been obtained, which gives $p_{o|x}(\mathbf{o}|\mathbf{x})$ as a function of the observed signal vector $\mathbf{x}$ in the following form:

$$p_{o|x}(\mathbf{o}|\mathbf{x}) = \mathcal{N}(\mathbf{o}, \mu_s, \sigma_s). \tag{6.22}$$

Also, the Hidden Markov Model defines observation likelihoods for each state, so that, given a hypothesized state q, it is possible to calculate $p_{o|q}(\mathbf{o}|q)$ as a function of the model state q.

Thus, two models for the speech feature $\mathbf{o}$ are available and in order to make maximum use of all available information, it is now suggested to obtain a combined model $\mathrm{p}(\mathbf{o}|\mathbf{x}, q)$. With this probability distribution, it will be possible to evaluate

$$\hat{\mathbf{o}} = \arg\max_{\mathbf{o}} \mathrm{p}(\mathbf{o}|\mathbf{x}, q) \tag{6.23}$$

at each frame and to use this maximum likelihood estimate $\hat{\mathbf{o}}$ of the feature vector for recognition as well as for speech processing.

In order to obtain the desired probability distribution, Bayes' law [2] can be applied as follows:

$$\begin{aligned} \mathrm{p}(\mathbf{o}|\mathbf{x}, q) &= \frac{\mathrm{p}(\mathbf{o}, \mathbf{x}, q)}{\mathrm{p}(\mathbf{x}, q)} \\ &= \frac{\mathrm{p}(\mathbf{o}, \mathbf{x}|q)\mathrm{p}(q)}{\mathrm{p}(\mathbf{x}|q)\mathrm{p}(q)} \\ &= \frac{\mathrm{p}(\mathbf{o}, \mathbf{x}|q)}{\mathrm{p}(\mathbf{x}|q)} \\ &= \frac{\mathrm{p}(\mathbf{o}|q)\mathrm{p}(\mathbf{x}|\mathbf{o}, q)}{\int_{o'} \mathrm{p}(\mathbf{x}|\mathbf{o}', q)\mathrm{p}(\mathbf{o}'|q)\mathrm{d}\mathbf{o}'}, \end{aligned} \tag{6.24}$$

where the law of total probabilities has been used in the last step. All statistical dependencies between the microphone signals $\mathbf{x}$ and the HMM state q are assumed to be captured in the feature vector $\mathbf{o}$. This makes $\mathrm{p}(\mathbf{x}|\mathbf{o}, q) = \mathrm{p}(\mathbf{x}|\mathbf{o})$, that is, the statistics of $\mathbf{o}$ are considered sufficient statistics for the input signal $\mathbf{x}$. Therefore

$$\begin{aligned} \mathrm{p}(\mathbf{o}|\mathbf{x}, q) &= \frac{\mathrm{p}(\mathbf{o}|q)\mathrm{p}(\mathbf{x}|\mathbf{o}, q)}{\int_{o'} \mathrm{p}(\mathbf{x}|\mathbf{o}', q)\mathrm{p}(\mathbf{o}'|q)\mathrm{d}\mathbf{o}'} \\ &= \frac{\mathrm{p}(\mathbf{o}|q)\mathrm{p}(\mathbf{x}|\mathbf{o})}{\int_{o'} \mathrm{p}(\mathbf{x}|\mathbf{o}', q)\mathrm{p}(\mathbf{o}'|q)\mathrm{d}\mathbf{o}'}. \end{aligned} \tag{6.25}$$

In this formulation, $\mathrm{p}(\mathbf{x}|\mathbf{o})$ is required, i.e., the probability of having observed $\mathbf{x}$ given that the features are $\mathbf{o}$. This value is hard to obtain in practice, therefore, Bayes' law is applied again to yield

$$\mathrm{p}(\mathbf{o}|\mathbf{x}, q) = \frac{\mathrm{p}(\mathbf{o}|q)\mathrm{p}(\mathbf{o}|\mathbf{x})\mathrm{p}(\mathbf{x})}{\mathrm{p}(\mathbf{o}) \int_{o'} \mathrm{p}(\mathbf{x}|\mathbf{o}', q)\mathrm{p}(\mathbf{o}'|q)\mathrm{d}\mathbf{o}'}. \tag{6.26}$$

In Equation (6.26), the integral $\int_{o'} \mathrm{p}(\mathbf{x}|\mathbf{o}', q)\mathrm{p}(\mathbf{o}'|q)\mathrm{d}\mathbf{o}'$ in the denominator as well as the term $\mathrm{p}(\mathbf{x})$ are independent of the feature vector $\mathbf{o}$. But since the equation will only be needed for the optimization problem stated in (6.23), they can be considered invariant scale factors. Defining a likelihood function p' via

$$p'(\mathbf{o}|\mathbf{x}, q) = \frac{\mathrm{p}(\mathbf{o}|q)\mathrm{p}(\mathbf{o}|\mathbf{x})}{\mathrm{p}(\mathbf{o})} \propto \mathrm{p}(\mathbf{o}|\mathbf{x}, q) \tag{6.27}$$

[2] It is used here in the two forms $\mathrm{p}(a|b) = \frac{\mathrm{p}(a,b)}{\mathrm{p}(b)}$ and $\mathrm{p}(a, b|c) = \mathrm{p}(a|c)\mathrm{p}(b|a, c)$.

allows to reformulate and simplify the problem as follows:

$$\hat{\mathbf{o}} = \arg\max_{\mathbf{o}} \mathrm{p}(\mathbf{o}|\mathbf{x}, q) = \arg\max_{\mathbf{o}} p'(\mathbf{o}|\mathbf{x}, q). \tag{6.28}$$

In order to simplify the following calculations, an uninformative, uniform prior is assumed for $\mathrm{p}(\mathbf{o})$. Thus, the term to be maximized is

$$\hat{\mathbf{o}} = \arg\max_{\mathbf{o}} \mathrm{p}(\mathbf{o}|q)\mathrm{p}(\mathbf{o}|\mathbf{x}), \tag{6.29}$$

which is the product of the processing model and the recognition model. The latter model may in general be chosen to be an MOG density. However, for clarity, the derivation is first shown for the plain Gaussian density in both models. In the processing model, this is

$$\mathrm{p}_{o|x}(\mathbf{o}|\mathbf{x}) = \frac{1}{\sqrt{(2\pi)^n ||\Sigma_s||}} \exp\left(-\frac{1}{2}(\mathbf{o}-\mu_s)^T \Sigma_s^{-1}(\mathbf{o}-\mu_s)\right) \tag{6.30}$$

and in the recognition model, the output distribution for a given state q is again described by

$$\mathrm{p}_o(\mathbf{o}|q) = \frac{1}{\sqrt{(2\pi)^n ||\Sigma_q||}} \exp\left(-\frac{1}{2}(\mathbf{o}-\mu_q)^T \Sigma_q^{-1}(\mathbf{o}-\mu_q)\right) \tag{6.31}$$

where μ_q and Σ_q are the mean and variance of the output distribution for state q.

Therefore, the optimization problem is:

$$\hat{\mathbf{o}} = \arg\max_{\mathbf{o}} e^{-\frac{1}{2}\left((\mathbf{o}-\mu_s)^T \Sigma_s^{-1}(\mathbf{o}-\mu_s)+(\mathbf{o}-\mu_q)^T \Sigma_q^{-1}(\mathbf{o}-\mu_q)\right)}. \tag{6.32}$$

Defining the log l of the cost function by

$$l = -\frac{1}{2}\left((\mathbf{o}-\mu_s)^T \Sigma_s^{-1}(\mathbf{o}-\mu_s) + (\mathbf{o}-\mu_q)^T \Sigma_q^{-1}(\mathbf{o}-\mu_q)\right), \tag{6.33}$$

the maximum likelihood estimate can be found by setting $\mathrm{dp}(\mathbf{o})/\mathrm{d}\mathbf{o}$ equal to zero

$$\frac{\mathrm{dp}(\mathbf{o})}{\mathrm{d}\mathbf{o}} \propto \frac{\mathrm{d}e^l}{\mathrm{d}\mathbf{o}} \stackrel{!}{=} 0. \tag{6.34}$$

Since e^l is always positive, the extremum must meet the condition

$$\frac{\mathrm{d}l}{\mathrm{d}\mathbf{o}} \stackrel{!}{=} 0. \tag{6.35}$$

Because of

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\mathbf{x}\mathbf{A}\mathbf{x}^T = \mathbf{A}\mathbf{x} + \mathbf{A}^T\mathbf{x}, \tag{6.36}$$

which, for symmetrical matrices, becomes

$$\frac{\mathrm{d}}{\mathrm{d}\mathbf{x}}\mathbf{x}\mathbf{A}\mathbf{x}^T = 2\mathbf{A}\mathbf{x}, \tag{6.37}$$

and since covariance matrices are symmetric,

$$\begin{aligned} \frac{\mathrm{d}l}{\mathrm{d}\mathbf{o}} &= 2\Sigma_s^{-1}(\mathbf{o}-\mu_s) \quad + \quad 2\Sigma_q^{-1}(\mathbf{o}-\mu_q) \\ \Leftrightarrow \Sigma_s^{-1}(\mathbf{o}-\mu_s) &\stackrel{!}{=} -\Sigma_q^{-1}(\mathbf{o}-\mu_q). \end{aligned} \tag{6.38}$$

In the case of diagonal covariance matrices, this means that the maximum will be obtained for that value $\mathbf{o}$ which lies on a straight line connecting the mean given by the processing model μ_s and the mean μ_q of the q'th state recognition model. Also, it is separated from the processing model mean by the same factor of processing standard deviations σ_s as it is apart from the recognition model mean in terms of recognition model standard deviations. This case is illustrated for the 1-dimensional scenario in Figure 6.10.

Finally, for computing the maximum likelihood estimate $\hat{\mathbf{o}}$, with

$$\hat{\mathbf{o}} = \arg\max_o \mathrm{p}_{o|q}(\mathbf{o}|q),$$

the following derivation

$$\begin{aligned} \Sigma_s^{-1}(\hat{\mathbf{o}}-\mu_s) &\stackrel{!}{=} -\Sigma_q^{-1}(\hat{\mathbf{o}}-\mu_q) \\ \Leftrightarrow (\Sigma_s^{-1}+\Sigma_i^{-1})\hat{\mathbf{o}} &= \mu_i\Sigma_i^{-1} + \mu_s\Sigma_s^{-1} \\ \Leftrightarrow \hat{\mathbf{o}} &= (\Sigma_s^{-1}+\Sigma_i^{-1})^{-1}(\mu_i\Sigma_i^{-1} + \mu_s\Sigma_s^{-1}) \end{aligned} \tag{6.39}$$

is used.

In the limit, it can be seen that this method also has a desirable behavior, since

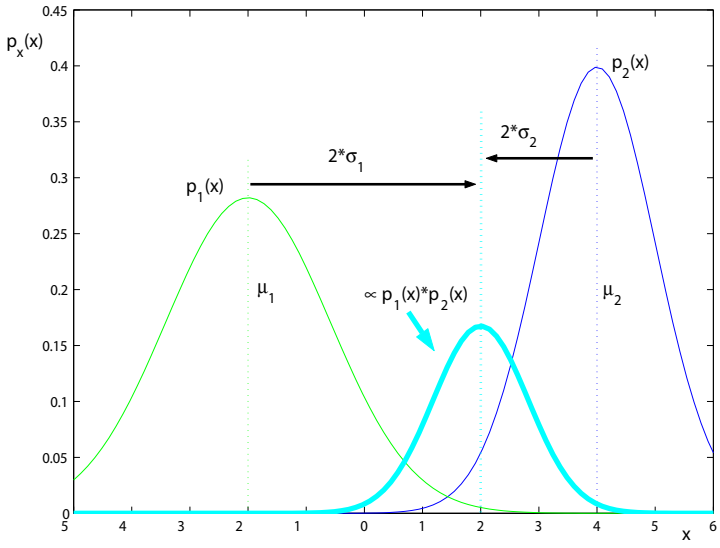

Figure 6.10: Composition of preprocessing model and recognition model with maximum likelihood estimate

- in the case of completely uncertain features, where Σ_s tends toward infinity, the method will choose exactly the recognition model mean as $\hat{\mathbf{o}} = \mu_q$,
- and when features are completely certain, the recognition model will not carry any weight and $\hat{\mathbf{o}} = \mu_s$ will hold.

This result can be plugged into the expression for the likelihood l given in Equation (6.31) and used for speech recognition in the same way as Equation (2.21) is applied for speech recognition when the features are considered given and fixed.

For Gaussian mixture models in the HMM state distribution, the same optimization problem needs to be carried out in principle, however, the exact solution needs to be found by numerical optimization. This can be seen from taking the processing model probability from Equation (6.30) and combining it with the MOG recognition model

$$p_{o|q}(o|q) \quad = \quad \sum_m \gamma_m \mathcal{N}_m(o, \mu_m, \sigma_m) \tag{6.40}$$

with the mixture index m. Taking these expressions and combining them via (6.29) yields the new optimization problem

$$\hat{\mathbf{o}} = \arg\max_{\mathbf{o}} \frac{1}{\sqrt{(2\pi)^n ||\Sigma_s||}} \exp(-\frac{1}{2}(\mathbf{o}-\mu_s)^T \Sigma_s^{-1}(\mathbf{o}-\mu_s)) \sum_m \gamma_m \mathcal{N}_m(o, \mu_m, \sigma_m). \quad (6.41)$$

Rather than using numerical optimization it is also possible to use a two-stage optimization, where rather than optimization over all summed mixtures, (6.39) is evaluated for all mixtures, of which in a second step, the most likely one is selected. This is equivalent to the optimization problem

$$\hat{\mathbf{o}} = \arg\max_m \arg\max_{\mathbf{o}} \frac{1}{\sqrt{(2\pi)^n ||\Sigma_s||}} \exp(-\frac{1}{2}(\mathbf{o}-\mu_s)^T \Sigma_s^{-1}(\mathbf{o}-\mu_s)) \sum_m \gamma_m \mathcal{N}_m(o, \mu_m, \sigma_m). \quad (6.42)$$

Compared to (6.41), this has the advantage of computational efficiency, since it scales linearly with the number of mixtures, and it can then also be used to evaluate the likelihood in (6.40).

6.5 Summary

A new method for using uncertainty information has been introduced. It consists of a two-stage process, where uncertainty values, interpreted as feature variances, are transformed together with the features themselves, taking them from the domain of preprocessing, usually the time-frequency domain, to an appropriate domain of speech recognition. After transformation, which can be carried out using analytical integration, Monte Carlo simulation or the unscented transform, these features can be used for missing feature recognition. For the purpose of uncertain recognition, a new method has been derived, which is termed *modified imputation*. This method leads to an estimate $\hat{\mathbf{o}}$ of the speech features, which considers both the information from the preprocessing stage and from the often very detailed recognition model, and leads to a smooth interpolation between these models. The following chapter will show results, which were obtained with the newly suggested approach, using ICA and time-frequency masking as a preprocessing method.

Chapter 7

Evaluation

7.1 Data Generation

7.1.1 Artificial Mixtures

In order to measure the algorithm performance on ideal and noise-corrupted mixtures, artificial, anechoic mixtures were generated with two different sets of speakers at four different noise levels. In all cases, the mixture was obtained by

$$\begin{bmatrix} X_1(j\omega) \\ X_2(j\omega) \end{bmatrix} = \begin{bmatrix} 1 & d_2 \mathrm{e}^{-j\omega\delta_2} \\ d_1 \mathrm{e}^{-j\omega\delta_1} & 1 \end{bmatrix} \begin{bmatrix} S_1(j\omega) \\ S_2(j\omega) \end{bmatrix} + \begin{bmatrix} N_1(j\omega) \\ N_2(j\omega) \end{bmatrix} \tag{7.1}$$

with the damping parameters $d_1 = d_2 = 0.8$. This model assumes that microphone 1 is a reference microphone for speaker 1 and microphone 2 for speaker 2. The delay parameters were calculated for the configuration shown in Figure 7.1. A microphone distance d=5cm and angles of incidence $\varphi_1 = -\varphi_2 = 30°$, relative to broadside, were chosen, yielding

$$\delta_1 = \delta_2 = \frac{d \cdot \sin(\varphi_1)}{c} \approx 39\mu s \tag{7.2}$$

for $c = 330$m/s. The damping factors were arbitrarily set to $d_1 = d_2 = 0.8$. Table 7.1 shows the datasets that were created in this setup. As noise signals, two recordings were used, which were obtained during the in-car data collection (see Section 7.1.2) at 100km/h.

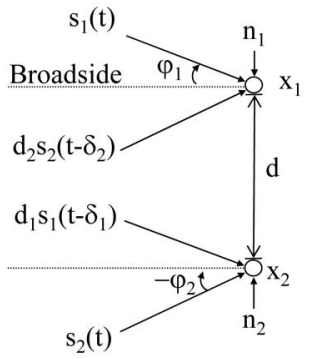

Figure 7.1: Signals are delayed according to their angles of incidence, which corresponds to a farfield model. Signal 1 has the angle of incidence φ_1 and the angle of incidence for signal 2 is φ_2, both relative to broadside. Damping with the damping factors d_1 and d_2 is also introduced, and realistic noise signals, which were recorded in a driving car on two microphones d cm apart, are added to both microphones.

Table 7.1: Artificial Dataset Indices

Dataset Label	**Speakers**	**Noise Level**
1a	2 male	noise free
2a	male+female	noise free
3a	2 male	10dB SNR
4a	male+female	10dB SNR
5a	2 male	0dB SNR
6a	male+female	0dB SNR
7a	2 male	-10dB SNR
8a	male+female	-10dB SNR

7.1.2 Car Data

In collaboration with DaimlerChrysler, a database of speech mixtures was recorded in a Mercedes S 320. For this purpose, two artificial heads were used to reproduce a subset of the TIDigits database, a database of continuously spoken digit

sequences [LDC1993], both separately as well as simultaneously.

7.1.2.1 Speaker Positioning

Recordings were made with the artificial heads at four different positions, of which the driver position was only available at standstill. Figure 7.2 shows the possible speaker locations for the recordings. Recordings were made for the combinations given in Table 7.2.

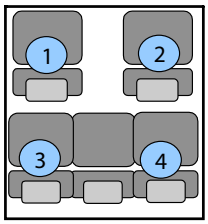

Figure 7.2: Dummy positions in car recordings.

Table 7.2: Speaker Combinations

Situations	Recorded Dummy Positions
Standstill, single speaker	1,2,3,4
Standstill, two speakers	(1,2), (2,3), (2,4)
Driving, single speaker	2,3,4
Driving, two speakers	(2,3), (2,4)

7.1.2.2 Microphone Installation

A microphone array, consisting of 8 omnidirectional microphones, was installed at the center of the ceiling near the interior light and the rear view mirror. Also, four cardioid microphones, one above each seat, were oriented toward the mouth of each respective speaker. Finally, two channels recorded the electrical signals at the loudspeakers of the artificial heads as reference signals. The microphone

positions are given in Figure 7.3 and an overview of the actual microphone installation is given in Figure 7.4.

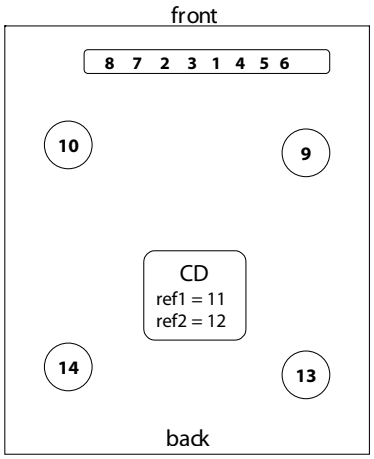

Figure 7.3: Channel arrangement in car recordings.

7.1.2.3 Recording Setup

The eight channel microphone array, the four cardioid microphones and both reference signals were simultaneously recorded on a 16-channel DAT recorder at 12khz with 16-bit resolution. Table 7.3 gives an overview of all parameters.

Array for recording

Cardioid microphone, above back right seat

Cardioid microphone, above front right seat

Cardioid microphone, above back left seat

Figure 7.4: Microphone placement for car recordings.

Table 7.3: Recording Setup

Microphones Array Ceiling	8 omnidirectional Sennheiser ME 102 4 cardioid
Pre-Amplifier	MidiMan
Recorder	16-channel DAT
File Format	headerless 16bit 12000Hz PCM
Artificial Heads	HEAD acoustics

7.1.2.4 Recordings of Single Speakers

To obtain reference signals for the evaluation, four speakers were selected out of the TI-Digits database. For these speakers, whose codes are given in Ta-

ble 7.4, all their 77 utterances from the TI-Digits database were concatenated and recordings were made for all four possible speaking positions.

Table 7.4: Single Speaker Recordings

Speaker Code	Gender
AR	Male
FM	Male
GA	Male
II	Female

7.1.2.5 Recordings of Multiple Speakers

To allow for the evaluation of ICA under clean and noisy conditions, recordings were made of two simultaneous speech signals. For this purpose, two simultaneously active artificial heads were used, both during standstill, and driving at different speeds and in different noise situations. Because of the low amplitude of the artificial heads, a number of noisy recordings exhibit excessive noise levels and are unsuitable for speech recognition tests, see also Table 7.7. Therefore, only the least noisy of the datasets, recorded at standstill and at 100km/h, were chosen for ICA and missing feature recognizer evaluation. Table 7.5 shows these selected datasets.

Table 7.5: Dataset Indices for In-Car Recordings

Dataset Label	Speakers	Driving Situation	Speaker Positions
1c	2 male	standstill	1+2
2c	male+female	standstill	1+2
3c	2 male	100km/h	2+3
4c	male+female	100km/h	2+3

7.1.2.6 Room Conditions

Using the TSP-technique described in Appendix C, the room impulse responses from all four speaker positions to all sensors were measured and the reverberation time[1] RT_{60} was calculated. Table 7.6 shows the reverberation times that were obtained.

Table 7.6: Reverberation Times

Speaker	Overhead Microphone	Array Center
Driver	21.1ms	26.5ms
Co-Driver	15.2ms	23.0ms
Left Back-Seat	17.3ms	38.2ms
Right Back-Seat	14.9ms	22.5ms

Also, the signal to noise ratio of the recordings was estimated, by comparing the energies of each noise-free reference recording with the energy of the noise-only signal recorded in the corresponding driving situation. This gave the results shown in Table 7.7, which also show why the recordings at 140 km/h were considered too noisy for source separation and recognition. Since the very low SNR is due to the low maximum amplitude of the artificial heads, these recordings would not be indicative of algorithm performance in realistic driving scenarios.

7.1.3 Data Collection in a Reverberant Environment

In order to obtain a reverberant test set, recordings were made in a lab room with dimensions of about 10×15m. The distance between the loudspeakers and the two microphones (Behringer ECM 8000) was set to one meter. At this distance, the reverberation time RT_{60} was measured to be 300ms. Again, speech signals from the TIDigits were used; in this case two male speakers, with speaker codes "AR" and "GA", were played back and recorded, once simultaneously and once separately, in two different setups of loudspeakers.

[1] Reverberation time is the time required for a sound in a room to decay by 60 dB.

Table 7.7: SNR at Array Center, Microphone 3.

Dataset	SNR
1c	30.4dB
2c	27.5dB
3c	-8.5dB
4c	-9.5dB
Recording at 140 km/h	-14.5dB

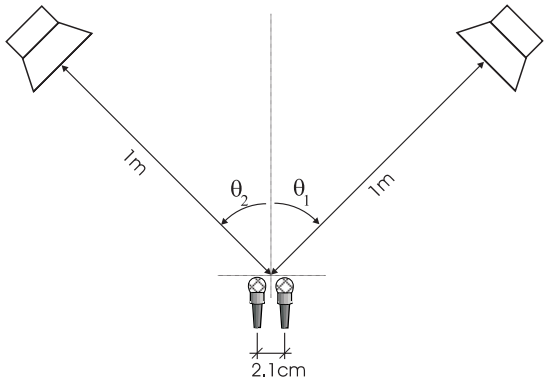

Figure 7.5: Experimental setup

Each of the signals was recorded with the loudspeaker positions varied according to the angles θ_1 and θ_2. The experimental setup is shown in Figure 7.5 and the following table gives an overview of the configurations.

Table 7.8: Overview of Reverberant Room Datasets

Dataset Label	Loudspeaker Setup θ_1 , θ_2
1r	45°, 25°
2r	10°, 25°

7.2 Evaluation

7.2.1 Performance Measures

To measure the efficiency of a speech signal processing approach, no one objectively ideal evaluation method can exist, and the evaluation will have to depend on the use that is to be made of the processed signal.

While telephony applications will naturally aim for understandability of the speech, as determined in listening tests, and for a good perceptual speech quality, which should also be determined by human listeners, speech recognition applications have recognition accuracy as their primary measure of success.

In both cases, for human listeners as for machine recognizers, evaluation of the signal processing success is a tedious process. On the one hand, listening tests are costly and time consuming, on the other hand, speech recognition results depend strongly on the recognizer that is used in the evaluation and need to be carried out on a large dataset in order to be representative.

Therefore, other measures for signal quality improvement are often used to complement the evaluation, even though they give results which are misleading in some cases.

A primary measures is the signal to interference ratio improvement ΔSIR. It is calculated for each source signal $n = 1 \ldots N$ by comparing its input SIR $SIR_{in,n}$ at a reference sensor i by

$$SIR_{in,n} = 10 \log_{10} \frac{\left\langle |x_{i,n}|^2 \right\rangle}{\left\langle \sum_{k=1,k\neq n}^{N} |x_{i,k}|^2 \right\rangle} \tag{7.3}$$

with the output SIR $SIR_{out,n}$

$$SIR_{out,n} = 10 \log_{10} \frac{\left\langle |y_{n,n}|^2 \right\rangle}{\left\langle \sum_{k=1,k\neq n}^{N} |y_{n,k}|^2 \right\rangle}. \tag{7.4}$$

Here, $\langle \cdot \rangle$ denotes time averaging and $y_{n,k}$ stands for the component of source k appearing in output signal n. This k^{th} source component in output n, $y_{n,k}$, is obtained

by using a recording in which only source k is active and demixing it with the unmixing system calculated on the mixed data.[2]

From these input and output SIRs, the SIR improvement due to the unmixing system, ΔSIR, is obtained for signal n via

$$\Delta SIR_n = SIR_{out,n} - SIR_{in,n}. \tag{7.5}$$

However, especially for time-frequency masking, the SIR improvement is not a sufficient measure of success. This can be seen from the fact that the degree of achievable separation performance is very high for a hypothetical algorithm which only retains those time-frequency bins that belong with very high probability to the speaker of interest only. Such an algorithm would suppress the interference energy almost completely. Still, it could lead to unacceptable performance in terms of perceived quality and understandability, since it could be discarding significant parts of the desired speaker signal.

Thus, the right balance between SIR improvement and consequential signal distortion is much more significant for the choice of algorithm and parameters than the SIR alone, and therefore, the signal to distortion ratio is also included in the evaluation. It is defined by

$$SDR_n = 10 \log_{10} \frac{\left\langle |x_n(t)|^2 \right\rangle}{\left\langle \left| (x_n(t) - y_{n,n}(t)) \right|^2 \right\rangle}, \tag{7.6}$$

where $x_n(t)$ is the microphone signal n obtained by recording only source n and $y_{n,k}$ again stands for the n^{th} ICA output, which is obtained by applying the ICA unmixing system and possible post-masking to a recording of only source k.

Since ICA and masking usually change the signal amplitude and since exact time-alignment between the reference signal and the processed output signal is often unavailable, in general, it is necessary to obtain a scale- and time-shift-invariant

[2]This formulation of SIR requires permutations of the output signals to be resolved, either by correlation with the reference signals or manually, i.e. the n^{th} output must correspond to the n^{th} source signal. This is no limitation to generality, since reference signals are needed in any case to determine SIR and SDR.

SDR measure. For source n, this can be found by

$$SDR_n = 10 \log_{10} \frac{\left\langle |\alpha_n x_n(t - D_n)|^2 \right\rangle}{\left\langle \left| (\alpha_n x_n(t - D_n) - y_{n,n}(t)) \right|^2 \right\rangle} \tag{7.7}$$

where delay D_n and amplitude factor α_n are adjusted to give optimum alignment between the reference signal $x_n(t)$ and the ICA output $y_{n,n}(t)$ [Saw2006].

Finally, recognition accuracy and recognition correctness are significant measures of recognizability. To determine these, the number of reference labels (R), substitutions (S), insertions (I) and deletions (D) is counted. From them, both criteria can be calculated:
The *correctness* is the percentage of correctly hypothesized words

$$PC = \frac{R - D - S}{R}. \tag{7.8}$$

Correctness has one disadvantage for judging ICA performance though. Since it ignores insertion errors, it will not penalize clear audibility of the interfering speaker during periods, when the desired speaker is silent. Therefore, a more important criterion for the success of ICA is recognition *accuracy*, defined as

$$PA = \frac{R - D - S - I}{R}. \tag{7.9}$$

Thus, PA will be used in the following evaluations of ICA performance, whereas, for judging the robustness of the entire system both criteria will be given, so that the recognizability of the processed signal can be judged by PC and the ability to distinguish the desired speaker from interferers is evaluated by PA.

7.2.2 Evaluation on DaimlerChrysler Speech Recognition System

In order to evaluate the recognizability of the processed speech signals, it was decided to use a standard, established recognizer with high robustness regarding environmental distortions. The DaimlerChrysler speech recognition engine (DCSR) is one such system. It is in use for command and control applications such as voice dialing in DaimlerChrysler cars in the so-called Linguatronic system, which has been on the market since 1996 [Hei2001]. It is described in detail e.g. in

[Hue2002]. Its recognition engine is comparable to the recognizer used in later evaluation in that it is also based on hidden Markov models of cepstral speech features. However, the features are not used directly, but rather, a vector quantization process is employed to obtain discrete valued speech features, requiring a minimum of storage space. Also, the system is adapted to the environment and to the speaker by means of cepstral mean subtraction (see Section 4.1.1.2) and vocal tract length normalization, described in 4.1.1.1.

7.2.2.1 Grammar Specification

For the purpose of recognizing the TI-Digits dataset, the sentence grammar was specified as an arbitrary-length sequence of digits with optional silence insertions. Figure 7.6 shows a graph representation of this grammar.

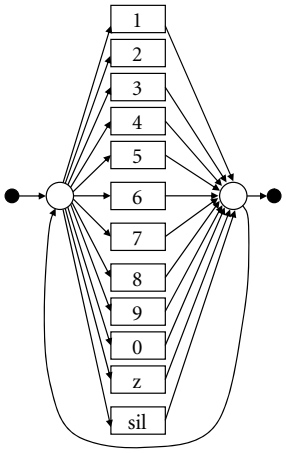

Figure 7.6: Specified grammar for recognition of digit strings. Here, rectangles stand for the corresponding word models while circles are link nodes, linking the last state of one HMM with the first state of another one. The black nodes represent the entry and exit states of the compound model. Regarding the word models, *1* through *0* stand for the HMMs of the words *one* to *oh*, *z* stands for the HMM of the word *zero* and *sil* is the silence model, which has the purpose of capturing background noise.

7.2.2.2 File Preparation

The DaimlerChrysler speech recognition engine accepts three different sample rates, 8, 12 and 16khz. Since the speech HMMs for English were only available for 16kHz speech data, it was necessary to resample all data to 16khz.

7.2.2.3 Parameter Adjustment

In order to find suitable settings for the system, speech recordings from the quiet and noisy in-car datasets, recorded of only single speakers at standstill and 100km/h respectively, were used as an input to the speech recognition engine.

For obtaining the best baseline recognition rate on this single-speaker data, different preprocessing algorithms were tested:

- no preprocessing
- spectral subtraction and
- vocal tract length normalization.

Table 7.9 shows the results which were obtained with these possible preprocessing settings, all measured on a total of 920 words. The evaluation is performed in terms of the word error rate[3], and the number of substitutions (SUB), insertions (INS) and deletions (DEL) is also given for the clean dataset. Performance was generally low on the noisy dataset, which is due to the low SNR of -9dB, for example, the word error rate was 99.1% without preprocessing. Thus, the performance on the noisy data was also evaluated, and it is given in the final column of Table 7.9 as the "noisy word error rate" (WERN).

With both vocal tract length normalization and spectral subtraction, best results were achieved, 7.8% with 85 errors, 22 substitutions, 3 insertions and 60 deletions on the clean and 43.3% on the noisy dataset.

Thus, both vocal tract length normalization as well as spectral subtraction were used for all further experiments, with their internal parameters adjusted to give

[3]The word error rate is related to the previously introduced word accuracy by $WER = 100 - PA$. It is the quality measure output by the DaimlerChrysler recognition system.

Table 7.9: Recognition results for different recognizer settings

Setting	WER	SUB	INS	DEL	WERN
No Preprocessing	11.3%	22	1	81	99.1%
Vocal Tract Norm.	8.5%	18	2	58	99.3%
Spectral Subtraction	13.2%	40	5	76	44.1%
Spectral Subtraction + Vocal Tract Norm.	7.8%	22	3	60	43.3%

optimum performance on the noisy dataset with acceptable accuracy on the clean data.

7.2.2.4 Baseline Recognition Performance

After the optimum recognizer settings were determined, baseline recognition rates were obtained for all datasets, using the set of parameters shown in Table 7.10.

Table 7.10: Recognizer Settings

Parameter	Value
Sampling Frequency	16000Hz
Cepstral Mean Adaptation	on
Decoding	Compound Word Decoding
N Best	50
Automatic Gain Control	on
Vocal Tract Normalization	Framewise
Spectral Subtraction	on
Cepstral Mean Normalization	on

The resultant recognition rates in terms of recognition accuracy are given in Table 7.11. In the first column, it shows the recognition rate for the noise free

recordings of only single speakers in the respective environments, with the dummy positioned on the correct seat where car recordings are concerned, and in the second column the performance for the mixed data is displayed. These word error rates give the reference, to which all subsequent recognition results can be compared.

Table 7.11: Baseline results on reference recordings and mixtures for all datasets.

DATASET Type	**Label**	**Percent Acc. Clean Data**	**Percent Acc. Mixture**
Artificial, Noise free	1a	94.070	26.030
Artificial, Noise free	2a	97.100	23.200
Artificial,10dB SNR	3a	94.070	25.760
Artificial,10dB SNR	4a	97.100	21.705
Artificial, 0dB SNR	5a	94.070	24.015
Artificial, 0dB SNR	6a	97.100	22.320
Artificial,-10dB SNR	7a	94.070	26.100
Artificial,-10dB SNR	8a	97.100	23.580
Car, Quiet	1c	90.580	20.410
Car, Quiet	2c	78.740	22.160
Car, Noisy	3c	90.480	19.620
Car, Noisy	4c	93.100	23.495
Reverberant Room	1r	48.330	21.065
Reverberant Room	2r	50.000	14.535

7.3 Results of ICA-Methods

The methods for improved speech source separation were tested on all 14 available datasets. Below, the results for both maximum likelihood permutation correction and for ICA-based time-frequency masking are presented in Section 7.3.1 and 7.3.3. For purpose of comparison, two ICA methods were used, on the one hand an algorithm by Parra and Spence, which works by inverse filtering in the time-domain, avoiding the permutation problem by constraining the filter length of the unmixing filters [Par2000], and on the other hand, a frequency domain implementation of the JADE algorithm, with permutation correction based on the assumption of a flat transfer function, as described in [Bau2001].

7.3.1 Results for Beampattern Based Permutation Correction

All experiments were carried out at a frequency resolution of NFFT=512 with a Hamming window and an overlap of 3/4, which is the minimum overlap needed to avoid undersampling, when a Hamming window is used.

In the reverberant environments, as seen in Table 7.12, best performance is achieved with maximum likelihood permutation correction. This is true for the achieved recognition accuracy as well as for the signal to interference and signal to distortion ratios.

In contrast to the reverberant recordings, for artificial delayed but not convolved mixtures, the Flatness criterion works better at times, as can be seen in the following Tables 7.13 through 7.16. This is to be expected, since very short impulse responses correspond to very flat transfer functions, which are exploited in flatness-based permutation correction. However, the added flexibility of maximum likelihood permutation correction does not pose a big disadvantage either, which can be seen from the rather similar performance, and in the noisy scenarios, the error-robust statistics-based approach is actually advantageous. In contrast, Parra's algorithm suffers from a low performance in this scenario. This may be due to the fact, that the parameters were not adjusted from scenario to scenario, so that the unmixing filter length, which is a constant in Parra's algorithm, was inappropriately long for the artificial data, since it had been chosen as $NFFT/2$ which is the parameter default in Parra's implementation and gave best overall performance.

Table 7.12: Results of different ICA algorithms on reverberant room recordings. *Maximum Likelihood* stands for the new algorithm with the EM-algorithm permutation correction described in Section 5.1.1.2, *Parra* stands for Parra's and Spence's algorithm [Par2000] and *Flatness* signifies the results obtained with frequency domain JADE and flatness-based permutation correction. Bold print indicates best performance for the considered dataset.

Algorithm	**Label**	**Percent Accuracy**	**SDR**	**SIR**
Flatness	1r	26.8	5.6	4.0
Flatness	2r	23.3	5.9	3.7
Maximum Likelihood	1r	**50.0**	**7.7**	**5.7**
Maximum Likelihood	2r	**42.3**	**7.3**	**4.8**
Parra	1r	41.2	-1.3	3.3
Parra	2r	34.4	1.3	3.6

Table 7.13: Results of different ICA algorithms on artificial noisefree mixtures

Algorithm	**Label**	**Percent Accuracy**	**SDR**	**SIR**
Flatness	1a	**94.3**	**15.7**	**10.1**
Flatness	2a	**95.9**	**14.3**	**8.4**
Maximum Likelihood	1a	**94.3**	**15.7**	**10.1**
Maximum Likelihood	2a	**95.9**	**14.3**	**8.4**
Parra	1a	24.2	0.5	5.6
Parra	2a	34.1	0.8	5.7

Table 7.14: Results of different ICA algorithms on artificial mixtures with added white noise at 10dB SNR.

Algorithm	Label	Percent Accuracy	SDR	SIR
Flatness	3a	**96.8**	**15.6**	**9.3**
Flatness	4a	98.4	14.5	8.1
Maximum Likelihood	3a	96.6	11.2	8.9
Maximum Likelihood	4a	**98.7**	**14.8**	**11.9**
Parra	3a	24.9	1.4	4.3
Parra	4a	26.3	2.2	4.6

Table 7.15: Results of different ICA algorithms on artificial mixtures with added white noise at 0dB SNR.

Algorithm	Label	Percent Accuracy	SDR	SIR
Flatness	5a	**94.8**	**13.1**	6.6
Flatness	6a	**96.7**	8.8	5.4
Maximum Likelihood	5a	94.4	11.0	**10.1**
Maximum Likelihood	6a	96.4	**9.6**	**7.4**
Parra	5a	26.8	2.8	5.6
Parra	6a	29.1	3.1	3.7

Table 7.16: Results of different ICA algorithms on artificial mixtures with added white noise at -10dB SNR.

Algorithm	Label	Percent Accuracy	SDR	SIR
Flatness	7a	**67.7**	**5.7**	0.1
Flatness	8a	40.4	**6.7**	0.1
Maximum Likelihood	7a	65.4	**5.7**	**6.2**
Maximum Likelihood	8a	**72.1**	5.5	**5.0**
Parra	7a	37.1	5.6	3.0
Parra	8a	49.7	5.3	1.1

In the car recordings, the overall best performance was again achieved with the maximum likelihood algorithm, as seen in Tables 7.17 and 7.18. The noise robustness of the separation quality for Parra's algorithm is also very high, as Table 7.18 shows, but while a good improvement in signal to interference ratio and a high signal to distortion ratio is obtained, noise is also accentuated in one of the outputs, which explains the low recognition rates achieved by Parra's algorithm even in the case where its other performance figures are best.

Table 7.17: Quiet car environment

Algorithm	**Label**	**Percent Accuracy**	**SDR**	**SIR**
Flatness	1c	65.3	4.6	0.7
Flatness	2c	75.1	**11.9**	8.4
Maximum Likelihood	1c	**69.0**	**8.3**	**4.4**
Maximum Likelihood	2c	**76.0**	11.3	**8.4**
Parra	1c	28.2	4.7	0.1
Parra	2c	30.4	1.2	1.2

Table 7.18: Noisy car environment, 100km/h, approx. -10dB SNR

Algorithm	**Label**	**Percent Accuracy**	**SDR**	**SIR**
Flatness	3c	48.4	5.7	3.0
Flatness	4c	46.6	4.3	3.1
Maximum Likelihood	3c	**60.7**	**7.1**	**4.4**
Maximum Likelihood	4c	**54.0**	4.4	2.6
Parra	3c	38.6	3.5	2.8
Parra	4c	34.8	**4.5**	**3.5**

7.3.2 Discussion

Beampattern based permutation correction has been performed with the help of the EM algorithm and a simple probabilistic DOA model. The above results show that

- the proposed approach gives best recognition rates for all considered datasets,
- and it also leads to the best average SDR and SIR for all recorded mixtures.
- For artificial mixtures, flatness based permutation correction gives a better signal quality (SDR), but the separation quality, measured as average SIR, is again best for the newly suggested algorithm in all scenarios.

These results come at the price of an increase in computational effort. A number of EM iterations are required, usually between 3 and 7 for the considered datasets. When all related aspects, including the beampattern computation, are considered as a part of the permutation correction, the algorithm requires about 25.1% of the entire effort of source separation. Whether this is an acceptable computational effort will rely very much on the mixing situation and the available resources, but since the robustness is increased significantly, in many cases this trade-off appears to be a positive one.

Additionally, beampattern based permutation correction results in a direction of arrival model for each of the sources. Aside from ICA, this may be useful for other steps of signal processing such as DOA-based VAD, and especially also for time-frequency masking, which is described in the following section.

7.3.3 Results Time-Frequency Masking

Time-Frequency masking was performed both with the amplitude mask described in Section 5.1.2.4 and the phase mask defined in Section 5.1.2.5. The amplitude mask is applicable to all ICA methods, since only the output amplitudes are needed in order to determine a mask. In contrast, the phase-based mask relies on a direction-of-arrival model, as it is estimated in the EM algorithm of the new maximum likelihood permutation correction described in Section 5.1.1.2.

7.3.3.1 Amplitude Mask

First, results were determined for Parra's algorithm. Since the results on artificial datasets in the evaluation of only linear ICA were not satisfactory with one setting for the filter length, it was decided here to adjust the filter length for the

artificial datasets to achieve optimum SDR. The value experimentally determined was Q=16. For all other scenarios, the filter length Q=512 was still used.

As Figure 7.7 shows for all artificial datasets, the effect of time-frequency masking is an increase of SIR and a simultaneous decrease in SDR, as was to be expected. Both changes are fairly smooth over masking threshold, and the fact, that the change continuously goes in the same direction of SIR improvement indicates, that the correct frequency points are chosen for deletion, so that amplitudes of ICA outputs appear to be a valid criterion for determining the mask. This is also visible from Tables 7.19 through 7.21, which cover all available datasets, i.e. the artificial, the reverberant room and the in-car recordings.

The point, at which best recognition rates were achieved is not consistent across datasets, though. In the cases in Tables 7.19 and 7.20, masking is very advantageous for recognition accuracy, with the best results attained at 0dB masking threshold, and at 78.9% very much better value than 49.2% without masking for the artificial data. In Table 7.21, however, best results are achieved at 3dB, where again, an improvement is visible, this time from 37.8% to 53.6% on average.

Table 7.19: Effect of amplitude-based mask on results of Parra's ICA algorithm, shown for all artificial datasets.

LABEL	**Percent Accuracy**	**SDR**	**SIR**
mean Mask OFF	49.2	16.1	7.8
mean Mask ON, Thresh = -6dB	60.3	11.2	9.3
mean Mask ON, Thresh = -3dB	68.8	9.2	10.3
mean Mask ON, Thresh = 0dB	78.9	6.3	11.8
mean Mask ON, Thresh = 3dB	66.1	3.5	12.4
mean Mask ON, Thresh = 6dB	50.6	2.0	12.8

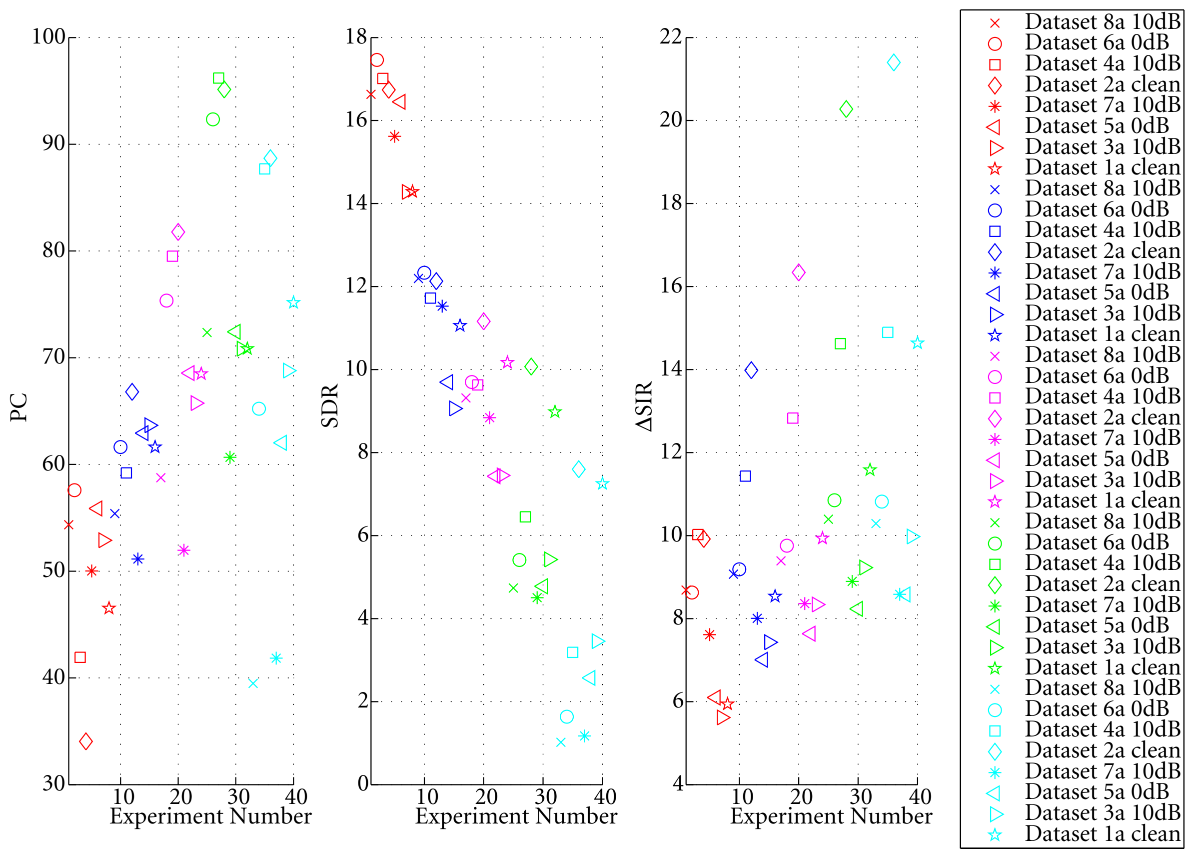

Figure 7.7: Effect of amplitude-based mask on results of Parra's ICA algorithm, shown for all artificial datasets. The red symbols show performance when no mask is applied. All other results were obtained with amplitude-based postmasking, where blue symbols indicate an amplitude threshold θ_{dB}=-6dB, purple symbols correspond to -3dB, for green symbols, the threshold was set at 0dB and the turquoise symbols correspond to θ_{dB}=3dB. Equal symbols denote equal datasets.

Table 7.20: Effect of amplitude-based mask on results of Parra's ICA algorithm, shown for all in-car datasets.

Method	Dataset	Percent Accuracy	SDR	SIR
Mask OFF	1c	28.8	0.8	0.0
Mask OFF	2c	32.1	2.4	1.0
Mask OFF	3c	37.9	4.7	3.6
Mask OFF	4c	29.3	4.5	2.9
Mask OFF	**mean**	**32.0**	**3.1**	**1.9**
Mask ON, Thresh = -6dB	1c	29.4	5.0	0.2
Mask ON, Thresh = -6dB	2c	45.9	2.1	2.2
Mask ON, Thresh = -6dB	3c	45.9	3.5	3.4
Mask ON, Thresh = -6dB	4c	39.2	3.9	4.4
Mask ON, Thresh = -6dB	**mean**	**40.1**	**3.6**	**2.5**
Mask ON, Thresh = -3dB	1c	32.5	4.7	0.3
Mask ON, Thresh = -3dB	2c	49.2	2.0	2.9
Mask ON, Thresh = -3dB	3c	56.7	3.3	4.2
Mask ON, Thresh = -3dB	4c	41.6	3.7	4.9
Mask ON, Thresh = -3dB	**mean**	**45.0**	**3.4**	**3.1**
Mask ON, Thresh = 0dB	1c	29.1	4.5	0.4
Mask ON, Thresh = 0dB	2c	49.4	1.9	4.4
Mask ON, Thresh = 0dB	3c	60.1	2.9	5.5
Mask ON, Thresh = 0dB	4c	48.9	3.6	5.6
Mask ON, Thresh = 0dB	**mean**	**46.8**	**3.2**	**4.0**
Mask ON, Thresh = 3dB	1c	28.5	4.2	0.5
Mask ON, Thresh = 3dB	2c	46.8	1.5	5.6
Mask ON, Thresh = 3dB	3c	60.8	2.4	6.7
Mask ON, Thresh = 3dB	4c	42.3	3.5	6.7
Mask ON, Thresh = 3dB	**mean**	**44.6**	**2.9**	**4.9**
Mask ON, Thresh = 6dB	1c	32.2	3.9	0.5
Mask ON, Thresh = 6dB	2c	37.9	1.1	6.7
Mask ON, Thresh = 6dB	3c	51.7	1.6	8.1
Mask ON, Thresh = 6dB	4c	37.0	3.2	8.2
Mask ON, Thresh = 6dB	**mean**	**39.7**	**2.5**	**5.9**

Table 7.21: Effect of amplitude-based mask on results of Parra's ICA algorithm, shown for the reverberant datasets.

Method	Dataset	Percent Accuracy	SDR	SIR
Mask OFF	1r	41.2	-1.3	3.3
Mask OFF	2r	34.4	1.3	3.6
Mask OFF	**mean**	**37.8**	**0.0**	**3.4**
Mask ON, Thresh = -6dB	1r	49.0	-1.3	4.0
Mask ON, Thresh = -6dB	2r	39.7	1.0	4.3
Mask ON, Thresh = -6dB	**mean**	**44.3**	**-0.2**	**4.1**
Mask ON, Thresh = -3dB	1r	52.0	-1.4	4.3
Mask ON, Thresh = -3dB	2r	40.5	0.9	4.7
Mask ON, Thresh = -3dB	**mean**	**46.2**	**-0.2**	**4.5**
Mask ON, Thresh = 0dB	1r	52.3	-1.6	4.9
Mask ON, Thresh = 0dB	2r	49.4	0.8	5.4
Mask ON, Thresh = 0dB	**mean**	**50.9**	**-0.4**	**5.1**
Mask ON, Thresh = 3dB	1r	54.2	-1.7	5.7
Mask ON, Thresh = 3dB	2r	52.9	0.6	5.9
Mask ON, Thresh = 3dB	**mean**	**53.6**	**-0.6**	**5.8**
Mask ON, Thresh = 6dB	1r	44.2	-2.0	6.9
Mask ON, Thresh = 6dB	2r	48.0	0.4	6.6
Mask ON, Thresh = 6dB	**mean**	**46.1**	**-0.8**	**6.7**

For JADE with flatness-based permutation correction, the situation is fairly similar regarding the SDR and SIR improvement. Figure 7.8 again shows the same tendencies as above, again the effect of time-frequency masking is an increase of SIR and a simultaneous, smooth decrease in SDR. Tables 7.23 through 7.24 illustrate the results for the car-data and the reverberant room recordings, which are again similar in nature.

However, in contrast to Parra's algorithm, JADE with flatness-based permutation correction does not always profit from time-frequency masking, where recognition accuracy is concerned, as can be seen e.g. from Table 7.22 and Table 7.24. Still, as Table 7.23 shows, there are also some cases, where improvements can be achieved, especially the noisy cases, where an improvement from 75.1 to 80.2% takes place in one of the scenarios.

But overall, it can be concluded that time-frequency masking as postprocessing is only suitable for the flatness-based algorithm when SIR improvements are the goal, but as long, as there is no way to better deal with the missing features in this case, it would not appear as a reasonable postprocessing method, when high recognition accuracy is needed.

Table 7.22: Effect of amplitude-based mask on results of JADE with flatness-based permutation correction algorithm, shown for all artificial datasets.

LABEL	**Percent Accuracy**	**SDR**	**SIR**
mean Mask OFF	85.6	11.8	6.0
mean Mask ON, Thresh = -6dB	83.4	11.0	12.1
mean Mask ON, Thresh = -3dB	81.6	10.5	13.0
mean Mask ON, Thresh = 0dB	80.0	9.9	14.0
mean Mask ON, Thresh = 3dB	77.3	9.1	15.2
mean Mask ON, Thresh = 6dB	75.2	8.3	16.5

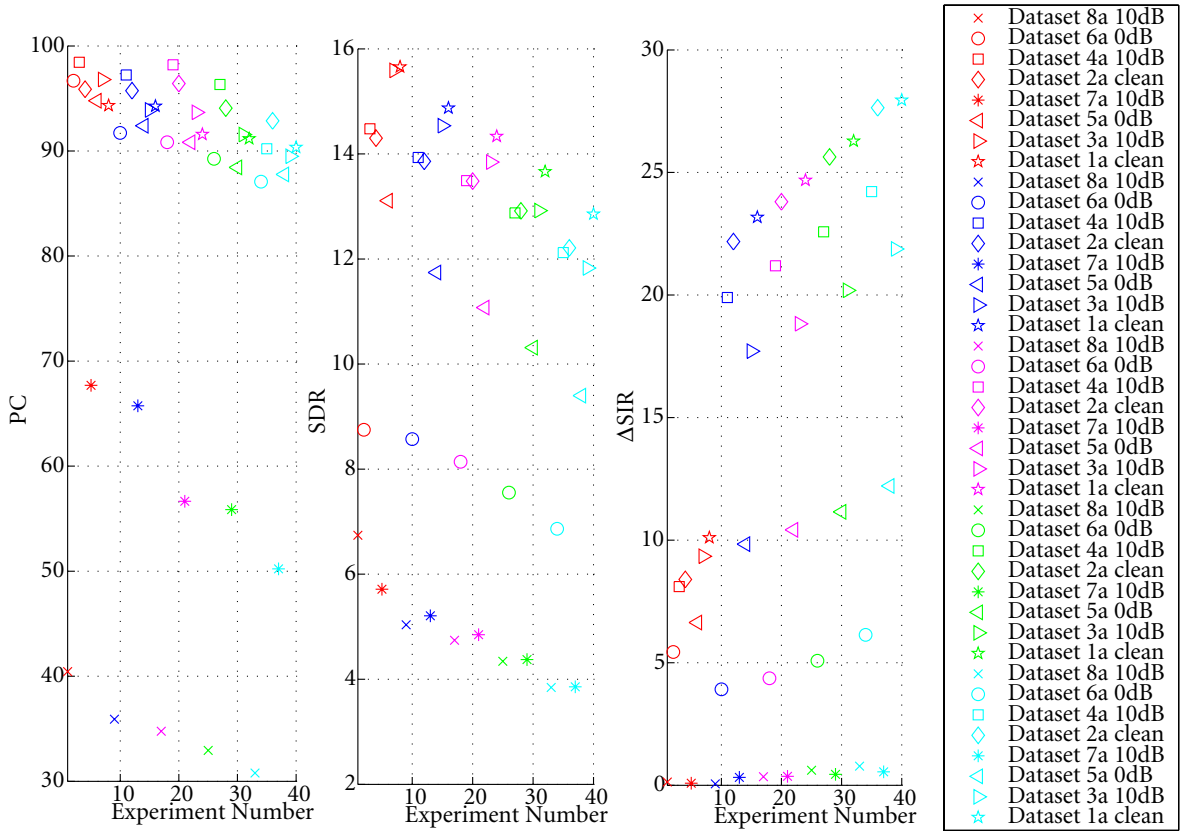

Figure 7.8: Effect of amplitude-based mask on results of JADE with flatness-based permutation correction algorithm, shown for all artificial datasets. The red symbols show performance when no mask is applied. All other results were obtained with amplitude-based postmasking, where blue symbols indicate an amplitude threshold θ_{dB}=-6dB, purple symbols correspond to -3dB, for green symbols, the threshold was set at 0dB and the turquoise symbols correspond to θ_{dB}=3dB. Equal symbols denote equal datasets.

Table 7.23: Effect of amplitude-based mask on results of JADE with flatness-based permutation correction algorithm, shown for all in-car datasets.

Method	Dataset	Percent Accuracy	SDR	SIR
Mask OFF	1c	65.3	4.6	-0.7
Mask OFF	3c	48.4	5.7	3.0
Mask OFF	2c	75.1	11.9	8.4
Mask OFF	4c	46.6	4.3	3.1
Mask OFF	**mean**	**58.9**	**6.6**	**3.5**
Mask ON, Thresh = -6dB	1c	47.2	3.5	-0.6
Mask ON, Thresh = -6dB	3c	44.5	5.2	3.7
Mask ON, Thresh = -6dB	2c	80.2	10.6	13.7
Mask ON, Thresh = -6dB	4c	47.4	5.5	4.2
Mask ON, Thresh = -6dB	**mean**	**54.8**	**6.2**	**5.3**
Mask ON, Thresh = -3dB	1c	48.1	3.4	-0.6
Mask ON, Thresh = -3dB	3c	42.1	5.0	3.8
Mask ON, Thresh = -3dB	2c	79.4	10.0	14.4
Mask ON, Thresh = -3dB	4c	46.7	5.7	4.5
Mask ON, Thresh = -3dB	**mean**	**54.1**	**6.0**	**5.5**
Mask ON, Thresh = 0dB	1c	46.6	3.2	-0.7
Mask ON, Thresh = 0dB	3c	40.0	4.7	3.9
Mask ON, Thresh = 0dB	2c	71.4	9.4	15.1
Mask ON, Thresh = 0dB	4c	45.1	5.8	4.8
Mask ON, Thresh = 0dB	**mean**	**50.8**	**5.8**	**5.8**
Mask ON, Thresh = 3dB	1c	46.4	3.0	-0.9
Mask ON, Thresh = 3dB	3c	35.2	4.4	4.0
Mask ON, Thresh = 3dB	2c	66.6	8.7	15.8
Mask ON, Thresh = 3dB	4c	42.6	5.6	5.1
Mask ON, Thresh = 3dB	**mean**	**47.7**	**5.4**	**6.0**
Mask ON, Thresh = 6dB	1c	44.4	2.8	-1.0
Mask ON, Thresh = 6dB	3c	30.6	3.9	4.0
Mask ON, Thresh = 6dB	2c	64.5	8.0	16.5
Mask ON, Thresh = 6dB	4c	37.0	5.1	5.3
Mask ON, Thresh = 6dB	**mean**	**44.1**	**4.9**	**6.2**

Table 7.24: Effect of amplitude-based mask on results of JADE with flatness-based permutation correction algorithm, shown for the reverberant datasets.

Method	Dataset	Percent Accuracy	SDR	SIR
Mask OFF	1r	26.8	5.6	4.0
Mask OFF	2r	23.3	5.9	3.7
Mask OFF	**mean**	**25.0**	**5.8**	**3.9**
Mask ON, Thresh = -6dB	1r	20.4	5.5	4.9
Mask ON, Thresh = -6dB	2r	15.1	5.5	4.1
Mask ON, Thresh = -6dB	**mean**	**17.8**	**5.5**	**4.5**
Mask ON, Thresh = -3dB	1r	17.0	5.3	5.2
Mask ON, Thresh = -3dB	2r	14.7	5.1	4.2
Mask ON, Thresh = -3dB	**mean**	**15.9**	**5.2**	**4.7**
Mask ON, Thresh = 0dB	1r	12.6	5.0	5.4
Mask ON, Thresh = 0dB	2r	13.8	4.7	4.4
Mask ON, Thresh = 0dB	**mean**	**13.2**	**4.8**	**4.9**
Mask ON, Thresh = 3dB	1r	13.7	4.5	5.7
Mask ON, Thresh = 3dB	2r	16.3	4.1	4.6
Mask ON, Thresh = 3dB	**mean**	**15.0**	**4.3**	**5.1**
Mask ON, Thresh = 6dB	1r	15.6	4.0	6.0
Mask ON, Thresh = 6dB	2r	7.3	3.5	4.9
Mask ON, Thresh = 6dB	**mean**	**11.5**	**3.7**	**5.5**

In the case of the new EM-based algorithm, time-frequency masking once again leads to large improvements in the SIR, for example an improvement from 8.5 to 14.9dB can be achieved on average for the artificial datasets. Also, there is only one among 14 considered cases, in which the SIR does not improve from application of the mask, making the method a good candidate for suppressing interfering speakers further than ICA alone could do.

However, the results for recognition accuracy are once again inconclusive. In some cases, for example in the reverberant room (see Table 7.27), significant improvements result from application of the amplitude-based masking function. On the other hand, on average the car-data becomes less recognizable after masking, as is to be seen in Table 7.26.

Table 7.25: Effect of amplitude-based mask on results of JADE with new EM-based permutation correction algorithm, mean calculated over all artificial datasets.

LABEL	**Percent Accuracy**	**SDR**	**SIR**
Mask OFF	89.2	11.0	8.5
Mask ON, Thresh = -6dB	87.4	10.7	11.6
Mask ON, Thresh = -3dB	86.0	10.3	12.3
Mask ON, Thresh = 0dB	83.6	9.8	13.1
Mask ON, Thresh = 3dB	80.7	9.2	13.9
Mask ON, Thresh = 6dB	77.9	8.6	14.9

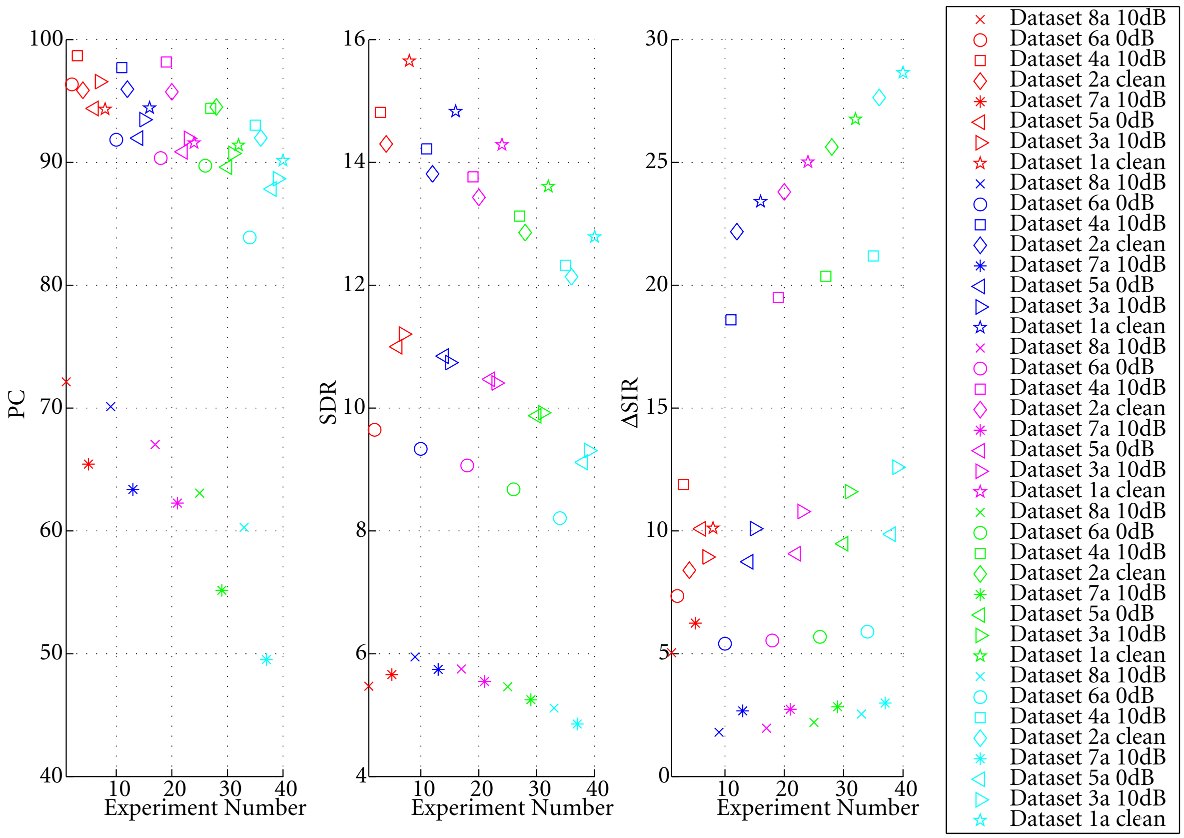

Figure 7.9: Effect of amplitude-based mask on results of JADE with new EM-based permutation correction algorithm, shown for all artificial datasets. The red symbols show performance when no mask is applied. All other results were obtained with amplitude-based postmasking, where blue symbols indicate an amplitude threshold θ_{dB}=-6dB, purple symbols correspond to -3dB, for green symbols, the threshold was set at 0dB and the turquoise symbols correspond to θ_{dB}=3dB. Equal symbols denote equal datasets.

Table 7.26: Effect of amplitude-based mask on results of JADE with new EM-based permutation correction algorithm, shown for all in-car datasets.

Method	Dataset	Percent Accuracy	SDR	SIR
Mask OFF	1c	69.0	5.6	1.9
Mask OFF	2c	76.0	11.3	8.4
Mask OFF	3c	60.7	7.1	4.4
Mask OFF	4c	53.9	4.4	2.6
Mask OFF	**mean**	**64.9**	**7.1**	**4.3**
Mask ON Thresh = -6dB	1c	61.5	4.2	2.2
Mask ON Thresh = -6dB	2c	81.2	10.3	13.7
Mask ON Thresh = -6dB	3c	64.6	6.7	6.6
Mask ON Thresh = -6dB	4c	44.9	5.1	3.5
Mask ON, Thresh = -6dB	**mean**	**63.1**	**6.6**	**6.5**
Mask ON , Thresh = -3dB	1c	60.7	4.1	2.6
Mask ON , Thresh = -3dB	2c	78.0	9.7	14.4
Mask ON , Thresh = -3dB	3c	63.0	6.6	7.1
Mask ON , Thresh = -3dB	4c	45.4	5.2	3.7
Mask ON, Thresh = -3dB	**mean**	**61.8**	**6.4**	**7.0**
Mask ON, Thresh = 0dB	1c	64.3	3.9	3.1
Mask ON, Thresh = 0dB	2c	75.2	9.1	15.0
Mask ON, Thresh = 0dB	3c	61.6	6.3	7.5
Mask ON, Thresh = 0dB	4c	44.0	5.1	3.8
Mask ON, Thresh = 0dB	**mean**	**61.3**	**6.1**	**7.4**
Mask ON, Thresh = 3dB	1c	60.2	3.7	3.5
Mask ON, Thresh = 3dB	2c	68.7	8.6	15.6
Mask ON, Thresh = 3dB	3c	62.3	5.8	8.0
Mask ON, Thresh = 3dB	4c	43.7	4.9	3.9
Mask ON, Thresh = 3dB	**mean**	**58.7**	**5.8**	**7.8**
Mask ON, Thresh = 6dB	1c	58.8	3.5	3.9
Mask ON, Thresh = 6dB	2c	64.7	7.9	16.1
Mask ON, Thresh = 6dB	3c	55.8	5.3	8.5
Mask ON, Thresh = 6dB	4c	41.7	4.4	4.1
Mask ON, Thresh = 6dB	**mean**	**55.2**	**5.3**	**8.2**

Table 7.27: Effect of amplitude-based mask on results of JADE with new EM-based permutation correction algorithm, shown for the reverberant datasets.

Method	Dataset	Percent Accuracy	SDR	SIR
Mask OFF	1r	50.0	7.7	5.8
Mask OFF	2r	42.3	7.3	4.8
Mask OFF	**mean**	**46.1**	**7.5**	**5.3**
Mask ON Thresh = -6dB	1r	61.6	6.4	7.4
Mask ON Thresh = -6dB	2r	39.0	6.2	5.6
Mask ON, Thresh = -6dB	**mean**	**50.3**	**6.3**	**6.5**
Mask ON , Thresh = -3dB	1r	58.2	6.1	8.0
Mask ON , Thresh = -3dB	2r	43.4	5.7	5.9
Mask ON, Thresh = -3dB	**mean**	**50.8**	**5.9**	**6.9**
Mask ON, Thresh = 0dB	1r	64.6	5.7	8.8
Mask ON, Thresh = 0dB	2r	51.2	5.1	6.2
Mask ON, Thresh = 0dB	**mean**	**57.9**	**5.4**	**7.5**
Mask ON, Thresh = 3dB	1r	68.0	5.1	9.8
Mask ON, Thresh = 3dB	2r	49.5	4.3	6.4
Mask ON, Thresh = 3dB	**mean**	**58.8**	**4.7**	**8.1**
Mask ON, Thresh = 6dB	1r	58.9	4.5	11.0
Mask ON, Thresh = 6dB	2r	53.8	3.4	6.6
Mask ON, Thresh = 6dB	**mean**	**56.3**	**3.9**	**8.8**

7.3.3.2 Phase Mask

The phase-based mask can only be applied, when there is a possibility of distinguishing the most likely source according to the phase angle. Among the considered ICA algorithms, this is only the case for JADE with maximum likelihood permutation correction, thus, this is the only algorithm for which results are available here. Firstly, Figure 7.10 gives an overview of the achieved results for the artificial datasets. As can be seen, the implemented phase-based mask only works reliably for those datasets with an SNR above 0dB. This is evident from the fact that only the SIR of the low-noise mixtures is reliably increased, but this is increased by a large degree. Also, it can be seen from the fact that both the clean in-car datasets (see Table 7.29) as well as the noise-free reverberant room recordings (in Table 7.30) can be recognized with significantly better accuracy with the mask than without it, whereas for the noisy in-car recordings, the SIR improves marginally but at the cost of a significantly lowered SDR.

Therefore, overall, the phase-based mask as implemented here can be considered as an alternative to amplitude-based masking, but it is only suitable for scenarios, where noise does not have a large impact on the signal. In further recognition tests, only the amplitude-based mask was selected for evaluation, since it is more robust to noise and reverberation and thus more suitable overall for the cases considered in this thesis.

Table 7.28: Effect of phase-based mask on results of JADE with new EM-based permutation correction algorithm, mean calculated over all artificial datasets.

LABEL	Percent Accuracy	SDR	SIR
Mask OFF	89.2	11.0	8.5
Mask ON, Likelihood Ratio 0.9	89.2	8.4	12.6
Mask ON, Likelihood Ratio 1.0	89.4	8.4	12.5
Mask ON, Likelihood Ratio 1.3	90.1	8.6	12.2
Mask ON, Likelihood Ratio 1.5	89.5	8.7	12.1

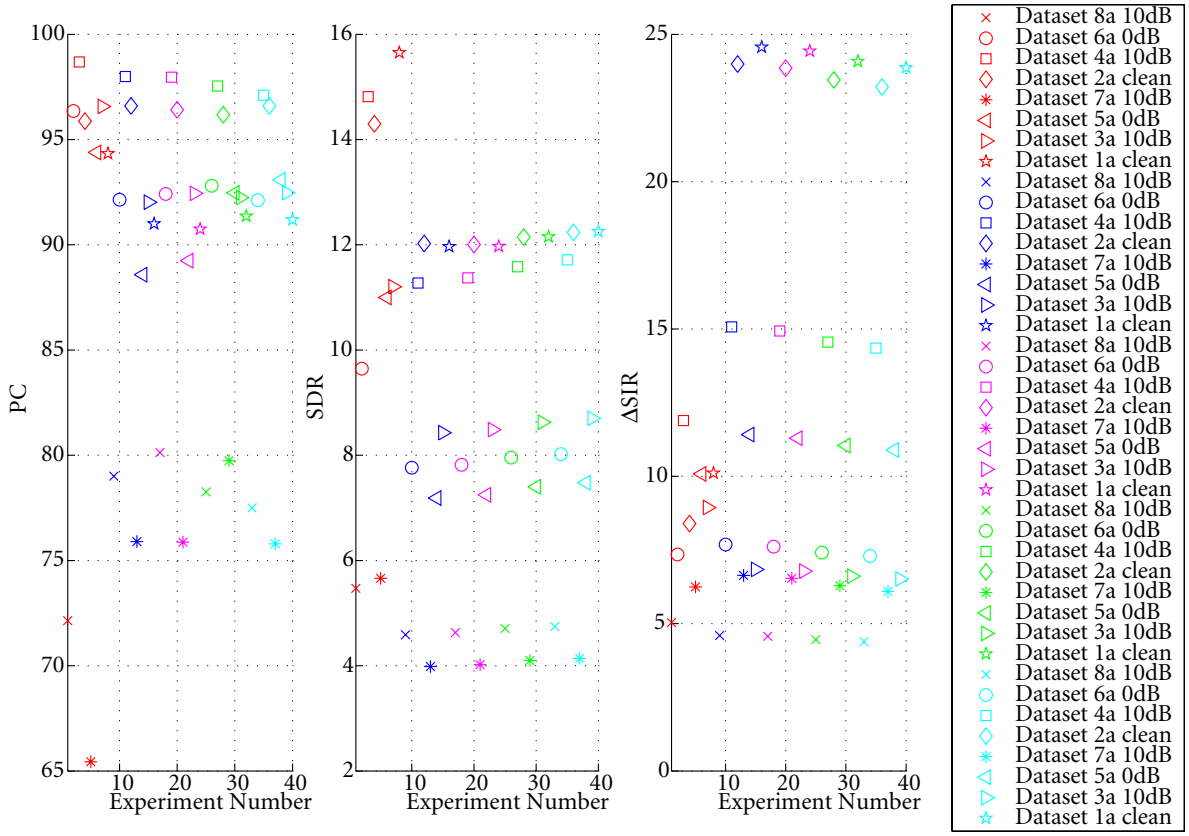

Figure 7.10: Effect of phase-based mask on results of JADE with new EM-based permutation correction algorithm, shown for all artificial datasets. Again, the red symbols show performance when no mask is applied. All other results were obtained with phase-based postmasking, with blue symbols indicating a likelihood ratio threshold Thr=0.9, purple symbols correspond to a likelihood ratio threshold of 1.0, for green symbols, the threshold was set to 1.3 and the turquoise symbols correspond to Thr=1.5. As before, equal symbols denote equal datasets.

Table 7.29: Effect of phase-based mask on results of JADE with new EM-based permutation correction algorithm, shown for all in-car datasets.

Method	Dataset	Percent Accuracy	SDR	SIR
Mask OFF	1c	69.0	8.3	4.4
Mask OFF	2c	76.0	11.3	8.4
Mask OFF	3c	60.7	7.1	4.4
Mask OFF	4c	53.9	4.4	2.6
Mask OFF	**mean**	**64.9**	**7.8**	**5.0**
Mask ON Likelihood Ratio 1.5	1c	74.6	7.3	8.5
Mask ON Likelihood Ratio 1.5	2c	78.9	3.2	9.8
Mask ON Likelihood Ratio 1.5	3c	60.9	2.9	5.5
Mask ON Likelihood Ratio 1.5	4c	52.6	2.2	2.8
Mask ON, Thresh = 1.5	**mean**	**66.7**	**3.9**	**6.6**
Mask ON Likelihood Ratio 1.3	1c	73.7	7.2	8.5
Mask ON Likelihood Ratio 1.3	2c	78.3	3.2	9.9
Mask ON Likelihood Ratio 1.3	3c	60.8	2.6	5.3
Mask ON Likelihood Ratio 1.3	4c	52.6	2.1	2.9
Mask ON, Thresh = 1.3	**mean**	**66.3**	**3.8**	**6.7**
Mask ON Likelihood Ratio 1.0	1c	76.4	7.2	8.7
Mask ON Likelihood Ratio 1.0	2c	80.0	3.1	10.1
Mask ON Likelihood Ratio 1.0	3c	57.3	2.3	5.0
Mask ON Likelihood Ratio 1.0	4c	51.6	1.8	2.9
Mask ON, Thresh = 1.0	**mean**	**66.3**	**3.6**	**6.7**
Mask ON Likelihood Ratio 0.9	1c	76.7	7.1	8.7
Mask ON Likelihood Ratio 0.9	2c	79.5	3.1	10.1
Mask ON Likelihood Ratio 0.9	3c	57.0	2.3	4.8
Mask ON Likelihood Ratio 0.9	4c	50.8	1.7	2.9
Mask ON, Thresh = 0.9	**mean**	**66.0**	**3.5**	**6.6**

Table 7.30: Effect of phase-based mask on results of JADE with new EM-based permutation correction algorithm, shown for the reverberant datasets.

Method	Dataset	Percent Accuracy	SDR	SIR
Mask OFF	1r	50.0	7.7	5.8
Mask OFF	2r	42.3	7.3	4.8
Mask OFF	**mean**	**46.1**	**7.5**	**5.3**
Mask ON, Likelihood Ratio 1.5	1r	64.0	5.6	8.8
Mask ON, Likelihood Ratio 1.5	2r	53.4	5.3	7.0
Mask ON, Thresh = 1.5	**mean**	**58.7**	**5.4**	**7.0**
Mask ON, Likelihood Ratio 1.3	1r	59.5	5.5	9.0
Mask ON, Likelihood Ratio 1.3	2r	48.9	5.2	7.1
Mask ON, Thresh = 1.3	**mean**	**54.2**	**5.4**	**8.1**
Mask ON, Likelihood Ratio 1.0	1r	59.2	5.4	9.2
Mask ON, Likelihood Ratio 1.0	2r	48.9	5.1	7.4
Mask ON, Thresh = 1.0	**mean**	**55.1**	**5.2**	**8.3**
Mask ON, Likelihood Ratio 0.9	1r	61.6	5.3	9.3
Mask ON, Likelihood Ratio 0.9	2r	54.2	5.0	7.5
Mask ON, Thresh = 0.9	**mean**	**57.9**	**5.2**	**8.4**

7.3.4 Discussion

Time-frequency masks can be obtained based on the results of ICA and they consistently improve separation performance as measured by Signal-to-Interference-Ratio (SIR).

Depending on the scenario, the best results are attained with different ICA algorithms. When considering the detailed results in the preceding evaluation, the following appear to be the most significant distinguishing characteristics of a mixing situation.

- **Non-reverberant mixtures** When the mixing is non-reverberant, as in Tables 7.19, 7.22, 7.25 and 7.28, the best results are attained by JADE with flatness permutation correction. This is also a reasonable result, since non-reverberant mixtures have very short corresponding impulse responses, leading to very flat room transfer functions, just as utilized by flatness based permutation correction.

- **Non-noisy reverberant mixtures** In fairly clean recordings with reverberation, namely in the car at standstill and in the room recordings, JADE with EM permutation correction and phase mask appears to be the best choice.

- **Noisy reverberant mixtures** In contrast, when real room recordings are made under noisy conditions, the phase ratio of the signals appears to deteriorate more than the amplitude ratio. This could also be expected and it leads to JADE with EM permutation correction plus *amplitude* mask clearly outperforming the other algorithms, see Tables 7.20, 7.23, 7.26 and 7.29 for the detailed performance.

Overall, thus, the appropriate choice of the ICA algorithm is the most important point in ascertaining the success of postmasking.

Once this decision has been made, postmasking can be applied successfully to all considered scenarios. In each scenario, excepting in a few cases of artificial mixtures at 0dB or -10dB, the signal to interference ratio is improved by all of the amplitude based postmasks. In contrast, phase based postmasking can occasionally converge only to the dominant target speaker, leading to a deterioration of separation for non-dominant targets, but among all experiments, this behavior was only observed in one isolated case, namely the artificial dataset 3a.

Otherwise, all presented algorithms perform robustly over all considered cases and with all tested ICA algorithms. However, as a disadvantage, there is also a consistent loss in Signal-to-Distortion-Ratio (SDR), thus, time-frequency masking will always lead to a trade-off between interferer suppression and target signal quality. This trade-off could be improved upon, either by employing a more sophisticated mask calculation, or by replacing deleted time-frequency bins with appropriate estimates. Nonwithstanding such improvements, though, while SIR is improved, the signal is simultaneously distorted to some degree. Thus, it depends on the chosen application area, whether or not the method is attractive for the specific case at hand.

For speech recognition purposes, this decision is not a clear one when only time-frequency masking is applied without any sort of compensation or uncertainty evaluation. However, as it is shown in the next section, with appropriate missing-feature methods, time-frequency masking can be used as a consistent improvement for speech recognition in all considered scenarios.

7.4 Evaluation of combined ICA and Missing Feature Recognition

7.4.1 Speech Recognition Setup

7.4.1.1 Hidden Markov Model Toolkit (HTK)

HTK is a toolkit for building speech recognition models based on HMMs. The toolkit provides routines for building and training HMMs and for carrying out the recognition by Viterbi decoding. The toolkit was used in this thesis to obtain speech models of two different granularities: one at the word level, and one at the phoneme level. These models were trained for the typical speech parameterizations of 13 mel frequency cepstral coefficients with delta and acceleration parameters, where a first order preemphasis filter was used to emphasize higher frequencies, and the energy was normalized in order to be tolerant to changes in speech level.

The HTK parameters used for parameterization are given below, the syntax of this parameter file is defined in [You2002].

```
# coding parameters 'configMFCC'
TARGETKIND = MFCC_0_D_A
TARGETRATE = 100000.0
SAVECOMPRESSED = T
SAVEWITHCRC = T
SOURCEFORMAT = NIST
WINDOWSIZE = 250000.0
USEHAMMING = T
PREEMCOEF = 0.97
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
ENORMALISE = T
```

To train the recognition model, a speaker independent seed model was trained on all phonemes from the TIDigits database. From the phoneme models, word models were built and they were adapted to the clean data recorded in the reverberant lab room. This explains why in contrast to DaimlerChrysler's recognizer,

which is designed for and trained on car data, the results for the HTK models are best on the reverberant room and artificial data.

7.4.1.2 Matlab Implementation

A speech recognizer was implemented in Matlab due to a number of limitations of HTK:

- HTK does not allow easy access to many internal variables, where especially framewise log likelihoods and would have been needed.
- Matlab allows easier integration of preprocessing and speech recognizer, e.g. for feeding feature likelihoods to the recognizer.

Thus, an HMM recognizer was implemented in the course of two diploma theses, [Huy2004, Ast2005]. This recognition system consists of two main parts needed here, which are on the one hand responsible for speech analysis (*getspeech*) and on the other hand for speech recognition (*recognize*), of an own Baum-Welch reestimation module for training and a couple of interface modules, as shown in Figure 7.11.

Since the recognizer training runs independently of the later recognition, HTK was used for all recognizer training, therefore, the Matlab system is also equipped with an interface for reading HTK HMMs.

7.4.2 Results Missing Feature Recognition

In order to make time-frequency masking a reliable option for speech recognition purposes also, missing feature recognition as described in Chapter 6 was implemented for the Matlab-based recognizer described above and tested on all datasets, both using Monte Carlo simulation and the unscented transform. Recognition was carried out by modified imputation as detailed in Section 6.4.

Experimentally, it was found that for the masking function defined in (6.2), three different parameters were useful for different cases. These sets of parameters are given in Table 7.31.

All other parameters were set as discussed above in 7.3.1.

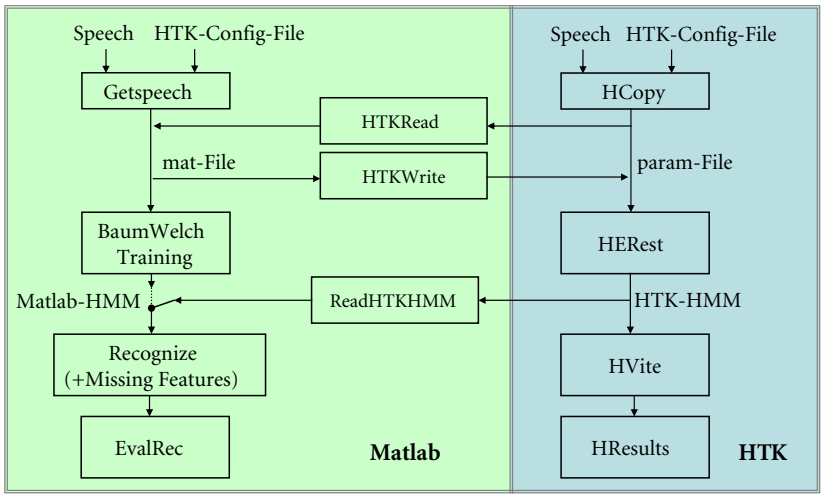

Figure 7.11: Structure and interfaces of Matlab and HTK speech recognizers.

Table 7.31: Parameters used for uncertain recognition.

Situation	**Label**	**Mask Damping d_m**	**Mask Threshold θ_{dB}**	**Error Probability p_e**
Strong interference from noise or other speaker	L	0	-2dB	0.08
Occasional interference	M	0.4	-1dB	0.08
Clean separation by ICA	H	0.9	-2dB	0.08

7.4.2.1 In-Car-Datasets

For the car datasets, generally speaking, good separation was already achieved by ICA alone at standstill, so the parameter set ”H” was usually used in that case, only the single female speaker (code *II*) makes an exception, since her voice is not as present in the recordings as that of the interfering male speaker. Regarding the quiet in-car data, it can be seen that missing feature techniques con-

tribute only slightly to improved recognition. On average, the results change from 49.1% correctness and 34.9% accuracy without masking to 43.5% correctness and 38.4% accuracy with masking, corresponding to a notable drop in understandability coupled with an about equal increase in interferer suppression. Using missing data techniques gives a final result of 51.5% correctness and 39.0% in accuracy, thus improving both figures, when compared to either purely linear ICA or ICA with time-frequency masking.

On the other hand, for noisy data, the improvement is significant, though it takes place at a low level. However, it can be expected that much better performance would result, if a recognizer were trained on noisy data at -10dB, which was not possible here due to a sparsity of recordings.

But at this low initial level, ICA alone leads to a recognition rate of 22.0% and 18.0% correctness respectively accuracy. By time-frequency masking, accuracy is actually improved to 25.3% and correctness also improves slightly to 19.0%. After application of missing data techniques, both figures improve noticably to 34.8% respectively 23.4%.

All detailed results can be seen in Tables 7.32 through 7.35.

Table 7.32: Results after missing feature recognition for in-car recording of speaker GA, results are given as PC/PA.

Dataset	**Parameter Label**	**ICA Out No Mask**	**Masked, No Missing Data**	**Monte Carlo**	**Unscented Transform**
Clean 1c	H	45.7/31.0	36.5/33.6	47.4/33.7	47.6/33.6
Noisy 3c	L	27.2/19.0	30.2/19.6	34.5/21.1	34.0/20.3

Table 7.33: Results after missing feature recognition for in-car recording of speaker AR, results are given as PC/PA.

Dataset	Mask	ICA Out No Mask	Masked, No Missing Data	Monte Carlo	Unscented Transform
Clean 1c	H	47.7/37.6	42.8/37.0	46.6/39.1	46.0/38.8
Noisy 3c	L	19.9/18.4	24.0/22.8	26.7/22.8	29.2/23.0

Table 7.34: Results after missing feature recognition for in-car recording of speaker FM, results are given as PC/PA.

Dataset	Mask	ICA Out No Mask	Masked, No Missing Data	Monte Carlo	Unscented Transform
Clean 2c	H	44.6/31.2	37.2/32.7	47.5/34.5	47.5/34.5
Noisy 4c	L	21.3/18.3	27.5/15.6	31.6/22.3	32.4/21.9

Table 7.35: Results after missing feature recognition for in-car recording of speaker II, results are given as PC/PA.

Dataset	Mask	ICA Out No Mask	Masked, No Missing Data	Monte Carlo	Unscented Transform
Clean 2c	M	58.4/39.8	57.4/50.5	65.3/49.0	65.4/48.7
Noisy 4c	L	19.6/16.3	19.3/17.9	40.6/28.5	43.7/28.4

7.4.2.2 Lab-Room-Datasets

In the reverberant room datasets, separation was of intermediate quality without postmasking, where one speaker was clearly better separated. Thus, for the better speaker with code *AR*, to be seen in Table 7.37, the medium quality parameters were chosen, and the parameters for speaker *GA* were set for low quality separation. With these settings, the average performance for configuration "1r", the one, which is easier to separate due to greater distance between the loudspeakers, changes from 83.4% respectively 58.8% without masking to 75.0% and 68.4% with

masking, which is the usual improvement in accuracy at the cost of correctnes, due to distortions in the speech signal. With missing feature techniques, here, the funal performane is 86.8% correctness, so all problems caused by masking are compensated, and the accuracy rises to 79.8%, which is a 49% relative error rate reduction, when compared to its value before masking.

For the second configuration, separation is more difficult, but here, the same tendency is visible. The original error rates after ICA are 76.2% and 55.8%, masking reduces the correctness to 72.9% and increases the accuracy to 66.7%. After application of missing data techniques, the loss in correctness is compensated to a final value of 86.7% and the correctness also improves to 70.9%.

For the reverberant room recording, in one dataset, Monte Carlo simulation gave an improved recognition rate compared to the unscented transform, however, since it is a technique where random changes can occur in the output occasionally, which can by chance help to improve recognition in some cases, this is most likely rather a random effect, especially since it was not observed in any of the other datasets.

Table 7.36: Results missing feature recognition of speaker GA in lab-room, results are given as PC/PA.

Dataset	**Mask**	**ICA Out No Mask**	**Masked, No Missing Data**	**Monte Carlo**	**Unscented Transform**
1r	L	86.8/57.1	69.7/66.3	93.6/93.6	89.4/89.4
2r	L	76.6/50.9	70.1/62.1	80.4/66.3	84.5/67.1

Table 7.37: Results missing feature recognition of speaker AR in lab-room, results are given as PC/PA.

Dataset	**Mask**	**ICA Out No Mask**	**Masked, No Missing Data**	**Monte Carlo**	**Unscented Transform**
1r	M	80.1/60.5	80.3/70.6	84.3/70.3	84.3/70.3
2r	M	75.7/60.7	75.7/71.3	88.8/74.8	88.8/74.8

7.4.2.3 Artificial Datasets

At higher SNRs, the source separation quality was very good with linear ICA alone, as seen in Tables 7.13 and 7.14, thus there is no need for time-frequency masking. However, at the lower SNRs of 0dB and -10dB, the artificial mixtures can also profit from time-frequency masking. The tables below show the results on these datasets, where dataset 5a and 6a were at 0dB and 7a and 8a at -10dB SNR. Again, the only female speaker (code *II*) is less well separated that the three male speakers, so the parameterization "L" was chosen for missing feature recognition in that case in Table 7.40, whereas all other datasets were recognized with the setting "H" for well separated data.

On average, the recognition rates change from 64.5% and 51.5% for the cleaner datasets and 37.3% respectively 32.7% for the noisy ones to 51.1% / 42.3 % for 0dB and 30.9% / 26.5% for -10dB after time-frequency masking. With missing features, 68.6% correctness results for the 0dB dataset and the accuracy is 55.1% which is a relative improvement of about 8% compared to purely linear ICA. For the noisy dataset 47.6% is the correctness, up from 37.3% for linear ICA, and the final accuracy after ICA, time-frequency masking and missing features at -10dB is 37.5% compared to originally 32.7%, so the relative correctness improvement in this case is about 28%.

Table 7.38: Noisy Artificial Mixtures Speaker GA, results are given as PC/PA.

Dataset	**Mask**	**ICA Out No Mask**	**Masked, No Missing Data**	**Monte Carlo**	**Unscented Transform**
5a	H	53.8/43.3	43.3/30.2	57.3/45.5	57.3/45.5
7a	H	34.0/24.8	31.5/23.7	42.1/29.4	41.3/29.2

Table 7.39: Noisy Artificial Mixtures Speaker AR, results are given as PC/PA.

Dataset	Mask	ICA Out No Mask	Masked, No Missing Data	Monte Carlo	Unscented Transform
5a	H	84.5/65.8	66.5/56.5	85.5/66.2	85.1/65.8
7a	H	52.6/43.2	32.1/30.6	54.4/47.6	58.3/50.0

Table 7.40: Noisy Artificial Mixtures Speaker II, results are given as PC/PA.

Dataset	Mask	ICA Out No Mask	Masked, No Missing Data	Monte Carlo	Unscented Transform
6a	L	61.5/48.5	53.3/44.4	68.4/57.2	67.8/56.8
8a	L	44.1/33.0	34.6/31.0	48.1/33.3	45.8/33.6

Table 7.41: Noisy Artificial Mixtures Speaker FM, results are given as PC/PA.

Dataset	Mask	ICA Out No Mask	Masked, No Missing Data	Monte Carlo	Unscented Transform
6a	H	58.2/48.3	41.4/38.3	64.0/52.1	64.0/52.1
8a	H	35.2/29.6	25.5/20.8	44.1/36.2	44.9/37.0

7.4.3 Discussion

Overall, it can be seen that the proposed architecture of ICA, followed by time-frequency masking and missing feature recognition, can improve the recognition performance on all datasets that were tested. Thus, uncertainty propagation and uncertain recognition by means of modified imputation can provide the necessary link between spectrum domain uncertainties and robust recognition in a multitude of other domains.

There is an added computational effort, however, which consists of two aspects:

- Transformation of uncertain features by *linear* stages is performed by two rather than one matrix multiplications for each time frame. The *nonlinear* transformations by means of the unscented transform require their respective nonlinearity to be evaluated $2n$ times, with $n = \dim(o)$, instead of once.

- For recognition, for each time frame and each HMM state, an evaluation of two additional matrix inversions and one more matrix multiplication become necessary. This roughly triples the evaluation time for each state.

On the positive side, though, the new approach opens up the opportunity of using uncertain recognition in other than the spectrum domain, even when uncertainties are estimated in the spectrum, as is often the case. Also, the new approach of uncertain recognition offers a smooth interpolation between completely uncertain features, which would result in imputation or marginalization, and completely certain features, as used in standard recognition. This allows for continuous-valued uncertainties and thus leads to smoother and more flexible approach.

Regarding the parameter settings, it is still necessary to select the masking parameters in accordance with the mixing situation, where experiments have shown that three parameter sets are sufficient, one for reliably separable mixtures[4], one for noisy and highly reverberant mixtures[5] and one for low-quality or low-amplitude speech signals in quiet scenarios. However, an automatic selection of the appropriate time-frequency masking parameters should also be possible, for example according to confidence measures output by the recognizer or the DOA-model.

With such appropriately set parameters, then, the performance in terms of recognition accuracy is always increased on all tested datasets, where it can be seen that time-frequency masking generally entails a loss in performance, but the subsequent missing-feature recognition can reliably compensate this loss.

Overall, best performance improvements are attained in the reverberant and most noisy scenarios, and for speakers with low amplitudes, where ICA alone is problematic. In those cases, missing features bring the biggest gains when compared to only ICA, as it is visible e.g. in Tables 7.35,7.36 or 7.41.

[4]This comprises all artificial and quiet in-car datasets with a sufficient speaker amplitude, i.e. excluding the one low-amplitude female speaker with speaker code II.

[5]These parameters are optimal for all noisy in-car recordings.

But the performance improvement is also clear in general. This can be seen from Table 7.42, which summarizes the results for all considered mixtures.

Table 7.42: Overview of all results in terms of PC/PA.

Datasets	**ICA Out No Mask**	**Masked, No Missing Data**	**Monte Carlo**	**Unscented Transform**
Quiet In-Car	49.1/34.9	43.5/38.5	51.7/39.1	51.6/38.9
Noisy In-Car	22.0/18.0	25.3/19.0	33.4/23.7	34.8/23.4
Reverberant Room	79.8/57.3	74.0/67.6	86.8/76.3	86.8/75.4
Noisy Artificial	53.0/42.1	41.0/34.4	58.1/45.9	58.1/46.3

Here, for the reverberant room as well as for the quiet in-car recordings, it is clearly visible how time-frequency masking results in a significant improvement of accuracy entailing a loss in correctness, and how missing feature recognition allows to increase correctness back to the original value, or even to a notably higher value, while retaining or improving on the accuracy achieved by time-frequency masking. In the noisy in-car-recordings, time-frequency masking on its own is already successful at improving recognition performance, albeit at very low values which are due to the insufficient SNR of -10dB. In this case, also, significantly better values can be obtained by missing feature recognition. Finally, as seen from the performance on artificial mixtures, at times time-frequency masking under noisy conditions can actually result in a decrease of performance. Here, missing feature techniques can again be used to improve results, and while the mask calculation is not very reliable in such noisy conditions, the entire setup of masking followed by missing feature recognition still improves performance in this difficult scenario.

As a target for future research, it will be interesting to combine the described setup with noise reduction techniques. Since ICA can actually worsen the SNR due to the white noise gain of the unmixing system, this may be a necessity when encountering data such as those observed in car-environments. The proposed architecture also allows seamless integration of noise reduction techniques, though, and when it is of interest, the noise estimate can be used to estimate also the uncertainty of features, which has already been successful in several tests and will be investigated in more detail in the future.

Chapter 8

Conclusions and Outlook

In this thesis, the use of independent component analysis for automatic speech recognition has been investigated. The main contribution of this work consists of three aspects.

Firstly, the problem of permutation correction was solved in a new, mathematically rigorous manner, which applies probabilistic models of the source directions. These models are estimated via the EM-algorithm, which has been proved to converge at least to a local optimum. Thus, a mathematically sound solution is applied for a problem which has so far often been solved by ad hoc methods.

Secondly, the use of time-frequency masking for ICA output signals has been suggested for the first time. While time-frequency masking for speech source separation has recently gained attention, the masking function was customarily estimated from beamforming-related or speech specific characteristics. Here, ICA has been successfully used as a basis for mask estimation, which brings with it the advantage of being inherently noise robust and furthermore lends itself to the use in reverberant environments, as it does not require the direction of arrival of source and interference to be fixed over frequency.

However, while time-frequency masked ICA leads to high gains in SIR, speech recognizer performance does not increase significantly and in some scenarios it even deteriorates. More precisely speaking, when time-frequency masking is applied after ICA, the interferer suppression increases, thus the accuracy figure generally increases significantly. But in many cases, this comes at the cost of reduced understandability of the target speaker, usually leading to a significantly reduced percentage of correctness, which is likely due to the distortion of recognition relevant features.

Therefore, as the final aspect of this thesis, a novel method is suggested to pass preprocessing information on to the recognition engine. This allows to inform

the recognizer of the confidence of the preprocessing stage in the estimated features and therefore lends itself easily to the integration of time-frequency masks in the recognition process. The approach is based on the unscented transform which makes it efficient computationally and flexible with respect to the parametrization of the speech recognizer. Also, a method has been described, which uses feature confidence information for recognition in a modified imputation approach, which smoothly varies the weight of the observed features based on the ratio of feature reliability and speech model variance. This approach does not require a threshold to distinguish ”reliable features” from ”unreliable” ones, and it compensates incurred feature distortions, giving better performance both in terms of correctness and accuracy than either ICA alone or ICA with time-frequency masking.

Thus, overall, two improvements of convolutive ICA were suggested which are specific to the speech processing scenario, and a new approach for integrating the so enhanced ICA algorithm was presented, which is also easily applicable to recognizing speech preprocessed by other speech enhancement algorithms.

Future investigations can proceed in a number of directions.

Regarding the mask, integrative mask computation can include other criteria in addition to those derived from ICA. For example, Bayesian approaches can flexibly handle an arbitrary number of information sources and integrate them into one optimum decision. This would allow a joint evaluation of, for example, ICA, beamforming and CASA criteria to arrive at one overall optimum masking function, which can be either binary or soft.

Concerning the uncertain recognition, in contrast to other techniques of uncertainty propagation, other features than the mel-cepstrum coefficients are also suitable for use within this framework. For this purpose, it is only necessary to separate them into their linear and nonlinear stages, then, output uncertainties can be obtained from input uncertainties also for currently much discussed features such as wavelet coefficients or RASTA-PLP features.

And finally, going beyond ICA, it will be interesting to see how well the proposed approach of uncertainty transformation and missing feature recognition integrates with other speech processing methods such as the Ephraim-Malah filter, with beamforming methods and with deconvolution techniques. It is expected that many approaches which make use of hard or soft time-frequency masks can in this way be better integrated with existing speech recognition systems.

Appendices

Appendix A

From the Theory of Random Variables and Statistics

A.1 Random Variables

In conducting random experiments, often, the interest of the observer lies not so much in the outcome itself, but rather in some function of this outcome. For example, when two dice are thrown, the observer might be interested not in the values of the individual dice but rather in its sum. This function, defined over the sample space, the space of all possible outcomes of a random experiment, is called a random variable. Formally speaking, given a random experiment and an associated sample space Ω, every function X that maps Ω to $\mathbb{R}$ is called a random variable.

A.1.1 Moment Generating Functions

The moment generating function is an alternative description of a random variable with a one-to-one correspondence between the distribution function and the moment generating function. The moment generating function of a random variable X is defined for all t by

$$\phi(t) = E(e^{Xt}). \tag{A.1}$$

For a continuous random variable x, this means

$$\phi(t) = \int_{-\infty}^{\infty} p_x(x) e^{xt} dx. \tag{A.2}$$

Successively differentiating ϕ at $t = 0$ yields all moments of x, for example

$$\begin{aligned}
\phi(t)^{'} &= \frac{d}{dt}\int_{-\infty}^{\infty} p_x(x)e^{xt}dx \\
&= \int_{-\infty}^{\infty} \frac{d}{dt}p_x(x)e^{xt}dx \\
&= \int_{-\infty}^{\infty} p_x(x)xe^{xt}dx \\
&= E(xe^{xt}) \qquad \text{(A.3)}
\end{aligned}$$

so that

$$\phi(0)^{'} = E(x), \qquad \text{(A.4)}$$

and

$$\phi(0)^{''} = E(x^2), \qquad \text{(A.5)}$$

and the n^{th} moment

$$\phi(0)^{n} = E(x^n) \qquad \text{(A.6)}$$

can be derived similarly.

A.2 Matrix Multiplication for Random Variables

Given a vector of random variables $\mathbf{s} = [s_1 s_2 \ldots s_n]$, and a linear transform

$$\mathbf{x} = \mathbf{A}\mathbf{s} \qquad \text{(A.7)}$$

and also knowing the means and variances μ_s and $\mathbf{\Sigma}_s$, what is needed are the means and variances of the measurement vector $\mathbf{x}$. To start out, matrix multiplication is decomposed into its constituent operations. Each entry of the output vector is the weighted sum of n random variables,

$$x_i = \sum_{j=1}^{n} a_{ij} \cdot s_j. \qquad \text{(A.8)}$$

Regarding the mean of one random variable,

$$\begin{aligned}
\mu_{a_{ij} \cdot s_j} &= \mathrm{E}(a_{ij} \cdot s_j) \\
&= \int_{-\infty}^{\infty} a_{ij} \cdot s_j p_{s_j}(s_j) ds_j \\
&= a_{ij} \int_{-\infty}^{\infty} s_j p_{s_j}(s_j) ds_j \\
&= a_{ij} \mu_{s_j} \qquad \text{(A.9)}
\end{aligned}$$

holds, so the mean will just behave linearly according to

$$\begin{aligned}
\mathrm{E}(\mathbf{x}) &= \mathrm{E}(A\mathbf{s}) \\
&= \mathrm{E}\left(\sum_j a_{ij} \cdot s_j \right) \\
&= \mathrm{E}\left(\sum_j a_{ij} \cdot s_j \right) \\
&= \sum_j \mathrm{E}(a_{ij} \cdot s_j) \\
&= \sum_j a_{ij} \cdot \mathrm{E}(s_j) \\
&= A \cdot \mathrm{E}(\mathbf{s}). \qquad \text{(A.10)}
\end{aligned}$$

Concerning the covariance,

$$\begin{aligned}
\Sigma_x &= E(\mathbf{x}\mathbf{x}^T) \\
&= E(A\mathbf{s}(A\mathbf{s})^T) \\
&= E(A\mathbf{s}\mathbf{s}^T A^T) \\
&= AE(\mathbf{s}\mathbf{s}^T)A^T \\
&= A\Sigma_s A^T \qquad \text{(A.11)}
\end{aligned}$$

can be used to propagate variances through linear transformations.

A.3 Unscented Transform

For the unscented transform, it is desired to generate a set of samples and weights, such that the weighted mean and average of the samples is equal to the

mean $\overline{\mathbf{x}}$ and covariance Σ_{xx} of a random variable $\mathbf{x}$. The points are generated by

$$\begin{aligned}
\mathcal{X}_0 &= \overline{\mathbf{x}} & \mathcal{W}_0 &= \tfrac{\kappa}{n+\kappa} & i &= 0 \\
\mathcal{X}_i &= \overline{\mathbf{x}} + (P_{xx})_{\cdot,i} & \mathcal{W}_i &= \tfrac{1}{2(n+\kappa)} & i &= 1,\ldots,n \\
\mathcal{X}_i &= \overline{\mathbf{x}} - (P_{xx})_{\cdot,i-n} & \mathcal{W}_i &= \tfrac{1}{2(n+\kappa)} & i &= n+1,\ldots,2n.
\end{aligned}$$

where P_{xx} is defined via

$$P_{xx} = \sqrt{(n+\kappa)\Sigma_{xx}}. \tag{A.12}$$

The weighted mean of the samples is given by

$$\overline{\mathcal{X}_i} = \sum_{i=0}^{2n} \mathcal{W}_i \mathcal{X}_i \tag{A.13}$$

which is equal to

$$\begin{aligned}
\overline{\mathcal{X}_i} &= \overline{\mathbf{x}}\,\mathcal{W}_0 + \sum_{i=1}^{n} \mathcal{W}_i \left(\overline{\mathbf{x}} + (P_{xx})_{\cdot,i} + \overline{\mathbf{x}} - (P_{xx})_{\cdot,i}\right) && \text{(A.14)} \\
&= \overline{\mathbf{x}}\left(\mathcal{W}_i + \sum_{i=1}^{n} 2\mathcal{W}_i\right) && \text{(A.15)} \\
&= \overline{\mathbf{x}}\left(\frac{\kappa}{n+\kappa} + \frac{2n}{2(n+\kappa)}\right) = \overline{\mathbf{x}} && \text{(A.16)}
\end{aligned}$$

as desired.

The weighted covariance is

$$\Sigma_{\mathcal{X}_i\mathcal{X}_i} = \sum_{i=0}^{2n} \mathcal{W}_i \left(\mathcal{X}_i - \overline{\mathbf{x}}\right)\left(\mathcal{X}_i - \overline{\mathbf{x}}\right)^T, \tag{A.17}$$

which leads to

$$\begin{aligned}
\Sigma_{\mathcal{X}_i\mathcal{X}_i} &= \sum_{i=1}^{n} \mathcal{W}_i \left(P_{xx}\right)_{\cdot,i} \left(P_{xx}\right)_{\cdot,i}^T + \sum_{i=n+1}^{2n} \mathcal{W}_i \left(P_{xx}\right)_{\cdot,(i-n)} \left(P_{xx}\right)_{\cdot,i-n}^T && \text{(A.18)} \\
&= 2\sum_{i=1}^{n} \mathcal{W}_i \left(P_{xx}\right)_{\cdot,i} \left(P_{xx}\right)_{\cdot,i}^T && \text{(A.19)} \\
&= \frac{1}{(n+\kappa)} \sum_{i=1}^{n} \left(P_{xx}\right)_{\cdot,i} \left(P_{xx}\right)_{\cdot,i}^T. && \text{(A.20)}
\end{aligned}$$

Because Σ_{xx} is a symmetric matrix, its matrix square root is also symmetric, so that the transpose of its i'th column $(P_{xx})^T_{\cdot,i}$ is equal to its i'th row $(P_{xx})_{i,\cdot}$ and therefore

$$\Sigma_{\mathcal{X}_i\mathcal{X}_i} = \frac{1}{(n+\kappa)} \sum_{i=1}^{n} (P_{xx})_{\cdot,i} (P_{xx})_{i,\cdot} . \tag{A.21}$$

Since the summation in (A.21) describes a matrix multiplication of P_{xx} with itself, one can also write

$$\Sigma_{\mathcal{X}_i\mathcal{X}_i} = \frac{1}{(n+\kappa)} P_{xx} P_{xx} \tag{A.22}$$

and due to the definition of P_{xx} in (A.12),

$$\Sigma_{\mathcal{X}_i\mathcal{X}_i} = \frac{1}{(n+\kappa)} \sqrt{(n+\kappa)\Sigma_{xx}} \sqrt{(n+\kappa)\Sigma_{xx}} \tag{A.23}$$

$$= \Sigma_{xx}, \tag{A.24}$$

which is the intended result.

Appendix B

Independent Component Analysis

Independent Component Analysis (ICA) has emerged as a mathematically well founded tool for recovering original signals from mixtures of the general form

$$\mathbf{x} = f(\mathbf{s}) \tag{B.1}$$

on the basis of statistical independence of the sources. Since the general formulation of the problem is very much ill-posed, various restrictions on the mixing process have been investigated to obtain practical solutions to the problem in specific cases. The most often applied simplification is the restriction to the linear mixing case, which is described in this appendix. However, in the case discussed in this thesis, convolutive mixing of sources is encountered. In order to solve this problem, linear ICA can be applied in the frequency domain, solving one ICA problem for each frequency bin of the mixtures. As discussed in the main section, care must be taken when the frequency bins are re-assembled to find a correct assignment of the outputs to the sources and to avoid arbitrary scaling of different frequency bins, but as long as these scaling and permutation problems are dealt with appropriately, the linear methods discussed here are applicable for general, convolutive mixing. The only further requirement, which stems from the application in the frequency domain, is that the methods must be suitable for complex valued signals and mixing systems, but this is the case for all algorithms discussed in the following.

Linear ICA

In linear ICA, the sensor signals are supposed to consist of linear combinations of the sources, as it is shown in Figure B.1. Then, the mixing process is described mathematically by

$$\mathbf{x} = A\mathbf{s}. \tag{B.2}$$

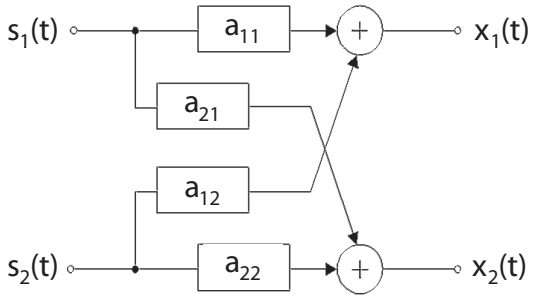

Figure B.1: Linear Mixing Model

The aim of ICA is to identify and subsequently invert this mixing process, so that the output signals $\mathbf{y}$ will be reconstructions of the source signals $\mathbf{s}$. Ideal separation would be achieved, if the mixing matrix A could be estimated perfectly and was also invertible, for in that case, one would have

$$\mathbf{y} = \hat{A}^{-1}\mathbf{x} = A^{-1}A\mathbf{s} = \mathbf{s}, \tag{B.3}$$

where $\hat{A}$ is the estimated mixing matrix. However, in general, insufficient information is available to obtain back the original sources. Thus, in linear ICA, the achievable, slightly lesser goal, consists in finding an unmixing matrix $W = \hat{A}^{-1}$, which will reconstruct the source signals up to scaling and permutation. This can be written as

$$\mathbf{y} = \hat{A}^{-1}\mathbf{x} = WA\mathbf{s} = PD\mathbf{s}, \tag{B.4}$$

where D is a diagonal matrix of scaling factors for the individual sources and P is a permutation matrix[1].

This matrix is found by searching for an unmixing matrix which will produce independent outputs $y_1 \ldots y_n$, as is shown in Figure B.2 for the two-by-two case. Statistical independence is a strong requirement. Two random variables are only independent, when their joint pdf factorizes. The requirement can thus be stated as follows:

$$p_{y_1,y_2}(y_1, y_2) = p_{y_1}(y_1)\, p_{y_2}(y_2)\,. \tag{B.5}$$

[1] A permutation matrix has exactly one entry equal to one in each row and column, while all other entries are zero.

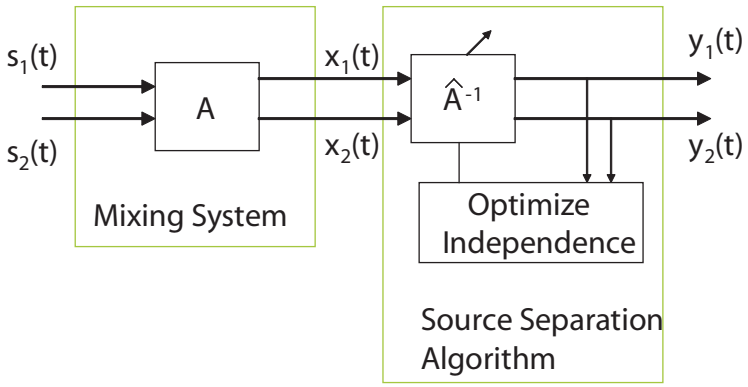

Figure B.2: Unmixing by Statistical Independence.

However, in order to obtain separated sources, the mere requirement of uncorrelatedness, or of

$$E(y_1 y_2) = 0 \tag{B.6}$$

is insufficient to recover sources from mixed signals, as can be seen from the following consideration.

Firstly, two random vectors are called uncorrelated, when all pairs of vector elements are uncorrelated according to (B.6), so

$$\mathrm{E}\left(\left(\mathbf{y} - \mu_y\right)\left(\mathbf{y} - \mu_y\right)^T\right) = \Delta, \tag{B.7}$$

where Δ is an arbitrary diagonal matrix. Without loss of generality, the following considerations are based on assuming zero-mean and unit-variance vectors $\mathbf{y}$, such that

$$C_{yy} = \mathrm{E}\left(\mathbf{y}\,\mathbf{y}^H\right) = I. \tag{B.8}$$

Then, a rotation of the vector $\mathbf{y}$ can be performed by pre-multiplying with a unitary matrix U via $\mathbf{z} = U\mathbf{y}$.[2] The effect of this multiplication is seen from

$$\begin{aligned} &C_{zz} = E(U\mathbf{y}\,\mathbf{y}^H U^H) = UIU^H = I \\ &\text{if} \quad UU^H = I. \end{aligned} \tag{B.9}$$

[2]A matrix U is called unitary, iff $UU^H = I$, with I as the unit matrix. A real unitary matrix is also an orthogonal matrix with the defining property $UU^T = I$.

Therefore, if $\mathbf{y}$ is uncorrelated, an arbitrarily rotated version will also be uncorrelated. Thus, making output signals uncorrelated is necessary for obtaining source separation but it is not sufficient and more information must be taken into consideration in order to achieve blind source separation.

This is the major task of independent component analysis, which aims to find output signals that are independent rather than merely uncorrelated.

Still, the most easily computed necessary condition for independence is uncorrelatedness of the data. Therefore, as a first processing step, the data are usually orthogonalized by Principle Components Analysis (PCA), such that their autocorrelation matrix $E_{\mathbf{y}\,\mathbf{y}}$ becomes the unit matrix I. Then, the actual ICA algorithm itself will only have to find an orthogonal rotation matrix, because this is the only part of the unmixing system that is still unknown. Figure B.3 illustrates this approach.

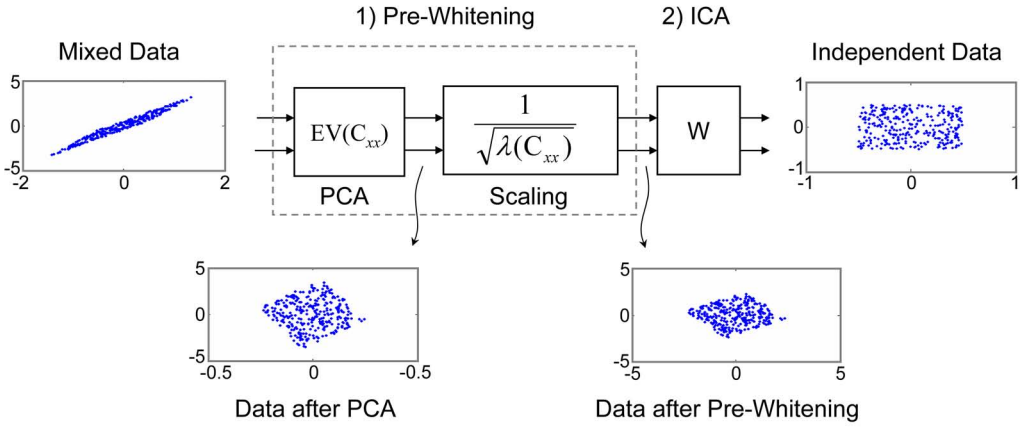

Figure B.3: Demixing is usually a two-stage procedure composed of pre-whitening followed by the actual ICA.

PCA

The first step, PCA, consists in finding that transformation matrix T which will orthonormalize the data, so

$$\mathrm{E}\left(T\mathbf{x}\,(T\mathbf{x})^{H}\right) = I. \tag{B.10}$$

If one separates this process into two steps, the first one of finding an unitary basis[3] and the second one of rescaling orthogonal data to make it orthonormal, this leads to

$$\begin{aligned} \mathrm{E}(U\mathbf{x}\,(U\mathbf{x})^H) &\stackrel{!}{=} \Delta \\ \Leftrightarrow U\mathrm{E}(\mathbf{x}\,\mathbf{x}^H)U^H &= \Delta \\ \Leftrightarrow \mathrm{E}(\mathbf{x}\,\mathbf{x}^H)U^H &= U^{-1}\Delta \end{aligned} \tag{B.11}$$

for the first step. Since U is assumed to be unitary, $U^H = U^{-1}$, so

$$\mathrm{E}(\mathbf{x}\mathbf{x}^H)U^H = U^H\Delta. \tag{B.12}$$

This expression is recognized as an eigenvector-problem of the data's autocorrelation matrix $R_{xx} = \mathrm{E}(\mathbf{x}\mathbf{x}^H)$, so the measurement vector must be projected onto the eigenvectors of the autocorrelation matrix first.[4] Secondly, from (B.11) it is seen that these decorrelated measurements will still be scaled by the eigenvalues of R_{xx}, which are contained in the diagonal matrix Δ. This can be compensated by multiplying with the inverse square root $\sqrt{\Delta^{-1}}$ of this matrix of eigenvalues. On the whole, then, the whitened data w can be obtained from the measurement values $\mathbf{x}$ by projecting onto the eigenvectors of R_{xx} first, and by secondly multiplying with the associated inverse roots of eigenvalues $\sqrt{\Delta^{-1}}$, which always exist because autocorrelation matrices are positive definite and thus have only positive eigenvalues.

This yields

$$T = \sqrt{\Delta^{-1}}U \tag{B.13}$$

as the entire whitening matrix, with U^H and Δ as the eigenvectors and eigenvalues of $\mathrm{E}(\mathbf{x}\,\mathbf{x})$,[5] so that

$$\mathrm{E}\left(T\mathbf{x}\,(T\mathbf{x})^H\right) = \sqrt{\Delta^{-1}}U\mathrm{E}(\mathbf{x}\,\mathbf{x}^H)U^H\sqrt{\Delta^{-1}} \tag{B.14}$$

$$= \sqrt{\Delta^{-1}}UU^H\Delta UU^H\sqrt{\Delta^{-1}} \tag{B.15}$$

$$= I.$$

[3] A set of vectors forms a unitary basis, if each of the vectors has the length one, and the Hermitian inner product of each pair of vectors is zero. A matrix composed of such a unitary basis as rows is itself unitary ($UU^H = I$) due to these properties of its constituent row vectors.

[4] The eigenvectors of a symmetric matrix are unitary [Hyv2001], so a solution with the assumed property $U^H = U^{-1}$ can always be found.

[5] If U^H are the eigenvectors and Δ contains the eigenvalues of $\mathrm{E}(\mathbf{x}\,\mathbf{x})$, this means that the corresponding eigenvector decomposition is $\mathrm{E}(\mathbf{x}\,\mathbf{x}) = U^H\Delta U$.

ICA

After PCA, the data already is orthogonal and since it must remain so in order to be independent, the actual search process of ICA is reduced to the search for a unitary transform which makes the output signals as independent as possible. For this purpose, some practical measure of statistical independence, which in itself is not easily estimated, needs to be found. Independence, as stated above in (B.5), implies that the joint pdf of the output vectors must factorize. Therefore, it is also necessary that

$$\mathrm{E}[g(y_i)h(y_j)] = \mathrm{E}[g(y_i)]\mathrm{E}[h(y_j)] \tag{B.16}$$

holds for any set of nonlinear functions $g(\circ), h(\circ)$ and for each pair of output signals y_i, y_j. This condition does not lend itself to direct verification, so many approximations to it have been investigated considering various types of nonlinear functions. Most popular are the choice of sigmoid non-linearities, which is motivated from the neural network perspective of source separation, and higher order polynomials. Since the computational effort and the sensitivity to measurement errors both increase with increased order, and since second order statistics alone are not sufficient for source separation (unless additional information regarding temporal or spectral source characteristics can be used), polynomials of third and fourth order have an inherent advantage. However, third order statistics are zero for all symmetric distributions. This consideration led to the use of fourth order statistics as the criterion for ICA in the course of this thesis.

The j'th order moment of a random variable x is defined as

$$m_j(x) = \mathrm{E}((x - \mu_x)^j). \tag{B.17}$$

It can also be obtained from the variable's first characteristic function

$$\varphi(x) = \mathrm{E}(e^{j\omega x}) = \mathfrak{F}(p_x(x)) \tag{B.18}$$

as the n'th coefficient of its Taylor series expansion.

While the moments of a random variable are thus based on its first characteristic function, a mathematically attractive alternative lies in the use of the second characteristic function $\phi(x)$ defined via:

$$\phi(x) = \ln(\varphi(x)). \tag{B.19}$$

The Taylor coefficients of this second characteristic function are called *cumulants* and are very well suited as an ICA criterion due to three useful properties:

$$\text{cum}(x + y) = \text{cum}(x) + \text{cum}(y) \tag{B.20}$$

$$\text{cum}(x_{gauss}) = 0 \qquad \text{for all orders} \geq 3 \tag{B.21}$$

$$\text{cum}(\beta x) = \beta^4 \text{cum}(x) \qquad \text{for fourth order cumulants.} \tag{B.22}$$

These are properties of cumulants of a single random variable, whereas cross cumulants are the cross statistics of vector valued random variables, defined equivalently as the expansion coefficients of (B.19), only for the multivariate case, which gives

$$\begin{aligned} \text{cum}(x_1, x_2, x_3, x_4) = \quad & \text{E}(x_1, x_2, x_3, x_4) - \text{E}(x_1, x_2)\text{E}(x_3, x_4) - \\ & \text{E}(x_1, x_3)\text{E}(x_2, x_4) - \text{E}(x_1, x_4)\text{E}(x_2, x_3) \end{aligned} \tag{B.23}$$

for fourth oder cumulants in the case of zero mean variables [Hyv2001]. For these fourth order cross cumulants, the equivalent of equation (B.22) is the useful multilinearity property

$$\text{cum}(\alpha x_1, \beta x_2, \gamma x_3, \delta x_4) = \alpha\beta\gamma\delta \, \text{cum}(x_1, x_2, x_3, x_4). \tag{B.24}$$

Because of these properties, joint diagonalization algorithms can be applied to the fourth order cross cumulant tensor in order to obtain maximally independent outputs $\mathbf{y}$.

The main assumptions, which are required for successful application of fourth order diagonalization algorithms are

- There is at most one Gaussian source signal,
- all sources are statistically independent, and
- if there is additive noise, it is normally distributed, none of the sources is Gaussian and the noise is independent from all sources.

Since these assumptions are all at least approximately fulfilled for speech signals in realistic environments, diagonalization of the fourth order cumulant tensor can be considered appropriate for the task of speech separation.

The JADE Algorithm

The algorithm which was used in this work to obtain fourth-order independence is the JADE-algorithm [Car1993]. It was chosen because it works on the basis of fourth order statistics only, so that it is not as sensitive to outliers as methods may be which consider even higher orders of source statistics, and because it does not consider third order statistics, which would be of no value for symmetric distributions such as those of speech. Also, among methods exploiting fourth order statistics it offers a good trade-off between accuracy and computational effort, because it makes efficient use of the eigenstructure of the fourth order cumulant tensor.

JADE is always applied on pre-whitened signals $\mathbf{z}$, where the whitening matrix T is a result of PCA as described in Section B. The mixing model after pre-whitening can be written as

$$\mathbf{z} = TA\mathbf{s} = W^H\mathbf{s} \tag{B.25}$$

where W^H is the whitened mixing matrix [6].

Since pre-whitening is carried out before ICA, the search can be constrained to unitary unmixing matrices W. JADE finds such a matrix by diagonalizing the fourth order cross cumulant tensor, whose entries are defined via

$$c_{ijkl} = \mathrm{cum}(x_i, x_j^*, x_k, x_l^*) \tag{B.26}$$

for complex valued signals $x_1 \dots x_n$. Because a cross cumulant is zero as soon as two of its argument variables are independent, this tensor needs to be diagonal in the sense that

$$c_{ijkl} \neq 0 \;\Rightarrow\; i = j = k = l. \tag{B.27}$$

This condition can also be achieved by "piecewise" diagonalization of the cumulant tensor. For this purpose, a *cumulant matrix* Q_x is defined entry wise as a function of an arbitrary $n \times n$ matrix M via

$$q_{ij} = \sum_{k=1}^{n}\sum_{l=1}^{n} \mathrm{cum}(x_i, x_j^*, x_k, x_l^*) M_{l,k}. \tag{B.28}$$

[6] W is the desired unmixing matrix, and the mixing matrix is $W^{-1} = W^H$, since W is unitary after pre-whitening.

If the matrix M is composed of $M = \mathbf{b_l}\mathbf{b_k^T}$, where $\mathbf{b_k}$ is an $n \times 1$ vector with a 1 in the k'th position and zeros everywhere else, the resulting cumulant matrix will consist of the cumulant tensor entries with fixed third and fourth index, thus they are referred to as *parallel cumulant slices*. All these parallel slices can be subsumed in one set $\mathcal{N}^p$ by defining

$$\mathcal{N}^p \stackrel{def}{=} \left\{ Q_x(\mathbf{b_l}\mathbf{b_k^T} | 1 \leq k, l \leq n) \right\}. \tag{B.29}$$

If the cumulant tensor is diagonal, so is every one of its n^2 parallel slices, thus, diagonalization of the n^2 matrices would offer a possibility of attaining independence. However, the computational effort of this approach would be considerable and it can be significantly reduced by concentrating on a reduced set of matrices. For this purpose, it is possible to utilize the fact that computation of the cumulant matrix Q_x as a function of the matrix M is a linear operator $Q(M)$. Since this operator is also symmetric, it has an eigenvalue decomposition of the form $Q(M) = \lambda M$, where λ is an associated eigenvalue and the matrix M is said to be an *eigenmatrix* of the cumulant tensor [Hyv2001].

To find those eigenmatrices, it is useful to look at the columns of the whitened mixing matrix W^H, which is $\left(W^H\right)_{\cdot,m} = \left(W^*_{m,\cdot}\right)^T$ for column m. The outer product[7] of this vector is

$$M_m = \left(W^*_{m,\cdot}\right)^T W_{m,\cdot}. \tag{B.30}$$

These matrices M_m are eigenmatrices of the cumulant tensor. This can be seen from the properties of cumulants (B.20) through (B.24) together with (B.28):

$$\begin{aligned} Q(M)_{ij} &= Q\left(\left(W^*_{m,\cdot}\right)^T W_{m,\cdot}\right) = \sum_{k=1}^{n}\sum_{l=1}^{n} \mathrm{cum}(z_i, z_j^*, z_k, z_l^*) M_{l,k} && \text{(B.31)} \\ &= \sum_{k=1}^{n}\sum_{l=1}^{n} \mathrm{cum}(z_i, z_j^*, z_k, z_l^*) W^*_{m,l} W_{m,k}. && \text{(B.32)} \end{aligned}$$

Since

$$z_i = (W^H)_{i,\cdot}\mathbf{s} = \sum_{q=1}^{n} (W^H)_{i,q} s_q \tag{B.33}$$

[7] The outer product of a complex column vector $\mathbf{v}$ is $\mathbf{v}\mathbf{v}^H$.

due to the mixing model in (B.25), this is

$$\begin{aligned} Q(M)_{ij} &= \sum_{k,l=1}^{n} \operatorname{cum}\left(\sum_{q=1}^{n} W_{q,i}^{*} s_q, \left(\sum_{q'=1}^{n} W_{q',j}^{*} s_{q'} \right)^{*}, \sum_{r=1}^{n} W_{r,k}^{*} s_r, \left(\sum_{r'=1}^{n} W_{r',l}^{*} s_{r'} \right)^{*} \right) W_{m,l}^{*} W_{m,k} \\ &= \sum_{k,l,q,q',r,r'=1}^{n} W_{m,l}^{*} W_{m,k} W_{q,i}^{*} W_{q',j} W_{r,k}^{*} W_{r',l} \operatorname{cum}(s_q, s_{q'}^{*}, s_r, s_{r'}^{*}). \end{aligned} \tag{B.34}$$

which follows from (B.24) and (B.20). Since the sources are independent, only the terms with $q = q' = r = r'$ are not equal to zero. Thus

$$Q(M)_{ij} = \sum_{k,l,q}^{n} W_{m,l}^{*} W_{m,k} W_{q,i}^{*} W_{q,j} W_{q,k}^{*} W_{q,l} \operatorname{cum}(s_q, s_q^{*}, s_q, s_q^{*}). \tag{B.35}$$

Because the matrix W is unitary, $\sum_k W_{m,k} W_{q,k}^{*} = \delta_{mq}$ and $\sum_l W_{m,l}^{*} W_{q,l} = \delta_{mq}$, so

$$\begin{aligned} Q(M)_{ij} &= \sum_{l,q=1}^{n} W_{m,l}^{*} W_{q,i}^{*} W_{q,j} W_{q,l} \delta_{mq} \operatorname{cum}(s_q, s_q^{*}, s_q, s_q^{*}) && \text{(B.36)} \\ &= \sum_{q=1}^{n} W_{q,i}^{*} W_{q,j} \delta_{mq} \operatorname{cum}(s_q, s_q^{*}, s_q, s_q^{*}) && \text{(B.37)} \\ &= W_{m,i}^{*} W_{m,j} \operatorname{cum}(s_m, s_m^{*}, s_m, s_m^{*}) && \text{(B.38)} \\ &= (M_m)_{i,j} \operatorname{cum}(s_m, s_m^{*}, s_m, s_m^{*}). && \text{(B.39)} \end{aligned}$$

Thus, matrices produced by on outer product of rows of the mixing matrix are eigenmatrices of the cumulant tensor and the eigenvalues corresponding to them are the fourth order cumulants of the corresponding source signals.

The entire set of eigenmatrices $\mathcal{N}^{em}$ of Q actually consists of n^2 matrices $M_1 \ldots M_{n^2}$ of the form

$$\mathcal{N}^{em} = \left\{ \left(W_{m,\cdot}^{*} \right)^{T} \cdot W_{k,\cdot} \mid 1 \le m, k \le n \right\}, \tag{B.40}$$

which can be shown in a proof similar to the one given above, and it is an orthonormal basis for the space of $n \times n$ matrices [Car1993]. Thus, any matrix M can be composed of eigenmatrices of Q with

$$M = \sum_{i=1}^{n^2} \kappa_i M_i \tag{B.41}$$

and therefore

$$Q(M) = Q\left(\sum_{i=1}^{n^2} \kappa_i M_i\right) = \sum_{i=1}^{n^2} \kappa_i \lambda_i M_i. \tag{B.42}$$

Thus, in order to diagonalize an arbitrary cumulant slice $Q(M)$, it is sufficient to diagonalize those eigenmatrices of the cumulant tensor, whose eigenvalues are greater than zero. As shown in [Car1993], at most n cumulant matrices have non-zero eigenvalues, and these are exactly the matrices described in (B.30), for which $m = k$ in (B.40). This is one of two points which make the JADE-algorithm computationally efficient: Since only n of the n^2 eigenvalues of the cumulant tensor are not zero, the n most significant eigenmatrices of Q are first determined and only these need to be diagonalized.

To examine the properties of the resulting mixing + unmixing system, the effect of the estimated unmixing matrix $\hat{W}$ on the estimated output signal $\hat{\mathbf{s}}$ is of interest. As described above, $\hat{W}$ is chosen so as to guarantee

$$\hat{W} W_{m,\cdot}^H W_{m,\cdot} \hat{W}^H = \Delta_m, \tag{B.43}$$

for the entire set of significant eigenmatrices $M_m, m = 1 \ldots n$, where Δ_m is a diagonal matrix. This is equivalent to

$$\Delta_m = \hat{W} \left(W_{m,\cdot}^*\right)^T W_{m,\cdot} \hat{W}^H = \hat{W} M_m \hat{W}^H \propto \hat{W} Q(M_m) \hat{W}^H. \tag{B.44}$$

When this unmixing matrix $\hat{W}$ is found, the cumulant tensor of unmixed signals $\hat{\mathbf{s}} = \hat{W}\mathbf{z}$ will own as eigenmatrices M_m the matrices composed of the n rows $m = 1 \ldots n$ of $\hat{W}W^H$ according to (B.30), so that the cumulant matrix of an arbitrary matrix M will have the form

$$\begin{aligned}
Q(M) &= Q(\sum_{m=1}^{n^2} \kappa_m M_m) && \text{(B.45)}\\
&= \sum_{m=1}^{n^2} \kappa_m \lambda_m M_m && \text{(B.46)}\\
&= \sum_{m=1}^{n} \kappa_m \lambda_m M_m && \text{(B.47)}\\
&= \sum_{m=1}^{n} \kappa_m ((\hat{W}W^H)_{\cdot,m})(\hat{W}W^H)_{\cdot,m}^H \mathrm{cum}(s_m, s_m^*, s_m, s_m^*), && \text{(B.48)}
\end{aligned}$$

which is zero in all off-diagonal elements if and only if the matrix $\hat{W}W^H$ can be described as $\hat{W}W^H = P\Delta$, with P being a permutation matrix and Δ an arbitrary diagonal matrix. Therefore, the diagonalization of significant cumulant eigenmatrices leads to a signal estimate $\hat{\mathbf{s}}$, which is a permuted, scaled version of the original signals:

$$\hat{\mathbf{s}} = P\Delta\mathbf{s}, \tag{B.49}$$

which is the expected, fundamental indeterminacy in independent component analysis.

In order to find such a matrix $\hat{W}$, diagonalization of all eigenmatrices of the cumulant tensor is thus necessary. This is achieved by means of successive rotations of the cumulant eigenmatrices M according to

$$M'_m = V_{i,j}^H M_m V_{i,j} \tag{B.50}$$

for each eigenmatrix $m = 1 \ldots n$, until all M'_m are sufficiently diagonal. For this purpose, Givens rotation matrices are used for $V_{i,j}$, which are of the form

$$V_{i,j} = \begin{pmatrix} 1 & & & & & & \\ & \ldots & & & & & \\ & & \cos(\theta) & \ldots & -\exp(j\phi)\sin(\theta) & & \\ & & \vdots & 1 & \vdots & & \\ & & \exp(-j\phi)\sin(\theta) & \ldots & \cos(\theta) & & \\ & & & & & \ldots & \\ & & & & & & 1 \end{pmatrix}.$$

All off-diagonal elements are zero, except for the elements $u_{i,j} = -\exp(j\phi)\sin(\theta)$ and $u_{j,i} = \exp(-j\phi)\sin(\theta)$, and all diagonal elements are equal to one, with the exception of $u_{i,i} = u_{j,j} = \cos(\theta)$. With this elementary rotation matrix, each step of the JADE algorithm has the goal of determining the optimum rotation angles (θ, ϕ), which minimize the (i, j)'th entry in all cumulant eigenmatrices M'_m simultaneously.[8]

[8]This is a second point which makes JADE an efficient fourth order algorithm: Because diagonalization is carried out jointly for all eigenmatrices, computational effort scales roughly linear with the dimension n and it can be further reduced by appropriate initialization of V [Car1993].

Thus, the rotation angles θ, ϕ of the matrices $V_{i,j}$ are chosen in each step so as to minimize a cost function

$$\theta, \phi = \arg\min_{\theta', \phi'} R = \arg\min_{\theta', \phi'} \sum_{m=1}^{n} \text{off}\left(V_{i,j}(\theta', \phi')^H M_m V_{i,j}(\theta', \phi')\right), \tag{B.51}$$

which describes the squared sum of off-diagonal-elements of all M'_m. This sum is measured for each matrix by $\text{off}(M_m)$, which is defined by

$$\text{off}(A) \overset{def}{=} \sum_{1 \leq i \neq j \leq n} |a_{ij}|^2. \tag{B.52}$$

Since a Givens rotation according to (B.50) minimizes the elements (i, j), an outer loop is necessary which carries out the optimization according to (B.50) and (B.51) for all matrix elements $1 \leq i \neq j \leq n$.

In summation, the entire JADE algorithm consists of the following steps:

- Orthogonalize the data via $\mathbf{z} = T\mathbf{x}$, so that $\mathbf{z}$ corresponds to the mixing model $\mathbf{z} = TA\mathbf{s} = W^H\mathbf{s}$.
- Compute the cumulant tensor of the whitened data, $Q_{\mathbf{z}}$.
- Determine the n most significant eigenmatrices $M_m, m = 1 \ldots n$ of the cumulant tensor.
- For all elements (i, j) find the rotation matrix $V_{i,j}$ which is the minimizer of $R = \sum_{m=1}^{n} \text{off}\left(V_{i,j}^H M_m V_{i,j}\right)$.
- The unitary part of the unmixing matrix is then given by $\hat{W} = \prod_{i,j} V_{i,j}^H$.
- Thus, the entire unmixing matrix $\hat{A}^{-1}$ is composed of the unmixing matrix V^H and the whitening matrix T, and the sources can be estimated from the complete mixing model $\hat{\mathbf{s}} = V^H TA\mathbf{s} = \hat{A}^{-1}A\mathbf{s}$.

Appendix C

Measurement of Room Impulse Responses

The so-called *time stretched pulse signal* was used to obtain the impulse response from source to sensor.

This principle of measuring impulse responses is described in detail in [Suz1995]. The basic idea is to use a chirp signal, defined in the discrete frequency domain by

$$H(\Omega) = \begin{cases} G(\Omega)\exp(j4\pi m\Omega^2/N^2) & \text{for } 0 \leq \Omega \leq N/2, \\ H(N-\Omega)^* & \text{for } N/2 < \Omega < N; \end{cases} \tag{C.1}$$

with $G(\Omega)$ defining the signals power spectrum and m the stretch of the pulse.

To get a better understanding of this signal, it is useful to consider it as the output of a filter whose input is $G(\Omega)$ and whose transfer function is given by $H(\Omega) = \exp(j4\pi m\Omega^2/N^2)$. The application of $H(\Omega)$ only changes the phase of an input signal but not the magnitude, and the change of the phase can be seen from computing the time-delay on each frequency resulting from filtering with $H(\Omega)$

$$\delta t = \frac{\mathrm{d}\angle(H(\Omega))}{\mathrm{d}\Omega} = \frac{8\pi m}{N^2}\Omega, \tag{C.2}$$

so the delay time is proportional to the frequency, and effectively $H(\Omega)$ expands its input signal in time by shifting constituent frequencies by different amounts of time [Aos1981].

Then, if $G(\Omega)$ is just a constant over frequency, it has a Dirac impulse as a time domain correspondence. After application of $H(\Omega)$, all constituent frequencies of the delta impulse are spread out over time, which is the reason, why the signal defined in Equation (C.1) is called a "time stretched pulse" (TSP).

When this signal is transformed to the time domain by an IDFT, the resulting signal $h_{tsp}(k)$ is real due to the definition of H as conjugate symmetric in Equation (C.1).

The inverse filter for this TSP with $G(\Omega) = 1$ can be expressed in the discrete frequency domain as follows:

$$H^{-1}(\Omega) = \begin{cases} \exp(-j4\pi m\Omega^2/N^2) & \text{for } 0 \leq \Omega \leq N/2, \\ H^{-1}(N-\Omega) & \text{for } N/2 < \Omega < N. \end{cases} \tag{C.3}$$

As the inverse TSP $h_{tsp}^{-1}(k)$ is also a real sequence in the discrete time domain, a noteworthy characteristic of the TSP $h_{tsp}(k)$ is that its convolution with $h_{tsp}^{-1}(k)$ is an almost perfect Dirac delta function, i.e.,

$$h_{tsp}(k) * h_{tsp}^{-1}(k) = \delta(k). \tag{C.4}$$

This is loosely speaking due to the fact, that, since $H^{-1}(\Omega)$ (the inverse of $H(\Omega)$) is just the conjugate complex of $H(\Omega)$, so

$$\frac{\mathrm{d}\angle(H^{-1}(\Omega))}{\mathrm{d}\Omega} = -\frac{\mathrm{d}\angle(H(\Omega))}{\mathrm{d}\Omega} \tag{C.5}$$

and wherever $H(\Omega)$ was shifting a constituent frequency of the Dirac pulse $G(\Omega)$ forward in time by a certain amount, applying $H^{-1}(\Omega)$ leads to an equivalent backward shift, so that the time stretched pulse is *compressed* again to give back the original Dirac function.

Therefore, when the TSP $h_{tsp}(k)$ is played back, transmitted, recorded, and filtered by inverse TSP $h_{tsp}^{-1}(k)$, the result is the impulse response of the whole system including the sound playback system, the loudspeaker, the transmission channel, the microphone, and the recording system.

This can be seen from

$$\begin{aligned} x(k) = h_{tsp}(k) * h_{ir}(k) * h_{tsp}^{-1}(k) &= \\ h_{tsp}(k) * h_{tsp}^{-1}(k) * h_{ir}(k) &= \\ \delta(k) * h_{ir}(k) &= h_{ir}(k). \end{aligned} \tag{C.6}$$

This time stretched pulse signal was played back in the car and the lab room from artificial mouths respectively loudspeakers 50 times each, with a 1 second

pause between TSP signals. Subsequently, the recording was split into 1 second segments and added synchronously, so that all TSP signals were also summed synchronously, which is important to suppress noise in the recordings by averaging. Finally, the synchronous sum of TSPs was inverse-filtered with $h_{tsp}^{-1}(k)$ to obtain the impulse response.

List of Symbols

When a signal is defined in the time domain as x, its frequency domain representation is the upper case version X and the mel spectrum, log mel spectrum and cepstrum are always denoted as X_{mel}, X_{log} and x_{cep}, respectively.
To differentiate between different microphones, a subscript is used for indicating the sensor number. Thus, the observed signal at sensor i in the frequency domain would be denoted by X_i.
Finally, when the true value of a signal or parameter is, e.g. s or A, its estimated value is denoted as $\hat{s}$ and $\hat{A}$, respectively.

Symbol	Use in this thesis
$\mathbf{A}$	mixing matrix
a_i	initial probability of HMM state i
$a_{i,j}$	HMM transition probability from state i to j
b_i	output probability distribution of HMM state i
c	speed of sound
$\mathbf{C}$	DCT-Matrix
d	signal of directed disturbance (=interfering speaker)
δ	Dirac delta distribution
δ	delay or delay vector
γ	required confidence for permutation detection
γ_m	mixture weight of m'th Gaussian mixture
h_s^i	room impulse response desired speaker to microphone i
H_s^i	room transfer function desired speaker to microphone i
h_d^i	room impulse response interfering speaker to microphone i
H_d^i	room transfer function interfering speaker to microphone i
h_n^i	room impulse response directed noise to microphone i
H_n^i	room transfer function directed noise to microphone i
h_{ir}	room impulse response
h_{tsp}	TSP sequence
$\mathbf{h_c}$	room impulse response in the cepstrum domain
k	discrete time frame index
$\mathbf{k}$	wavenumber vector
$\mathbf{\Lambda}$	set of HMM model parameters

Symbol	Use in this thesis
$L_j(t)$	likelihood of being in HMM-State j at time t
LF	control parameter for cepstral liftering
$\mathbf{M}$	mel-filterbank parameters in matrix notation
M	number of sensors
μ	mean
N	number of sources
n_a^i	ambient noise at sensor i
n_s^i	sensor noise of microphone i
$\mathcal{N}(x, \mu, \Sigma)$	$p_x(x)$ is Gaussian with parameters μ and Σ
$NFFT$	number of frequency bands
$\mathbf{O}$	sequence of observed feature vectors
$\mathbf{o}_t$	observed feature vector at time t
$\mathbf{\Omega}$	frequency bin number
ω	continuous angular frequency
ω_k	angular frequency at center of k'th frequency bin
φ_s	direction of desired signal
φ_d	direction of interfering signal
Q	state sequence of HMM
q_t	state of HMM at time t
$\hat{q}$	state estimate
$\mathbf{r_i}$	SPLICE correction vector for noise type i
$\mathbf{r_{xx}}$	autocorrelation vector for signal x
$\mathbf{R_{xx}}$	autocorrelation matrix for signal x
$\hat{s}$	estimated signal
s	source signal of desired speaker
σ	variance
Σ	covariance matrix
t	continuous time, time delay index of cepstrum
$\mathbf{W}$	unmixing matrix
x	observed signal
y^i	i^{th} output of unmixing system

List of Abbreviations

Abbreviation	Use in this thesis
BSS	Blind Source Separation
CASA	Computational Auditory Scene Analysis
cdf	cumulative distribution function
CMS	Cepstral Mean Subtraction
DAT	Digital Audio Tape
DCSR	DaimlerChrysler speech recognition engine
DCT	Discrete Cosine Transform
DOA	Direction of Arrival
HMM	Hidden Markov Model
EM	Expectation Maximization
ETSI	European Telecommunications Standards Institute
ICA	Independent Component Analysis
MAP	Maximum a Posteriori
MCE	Minimum Classification Error
MFCC	Mel Frequency Cepstral Coefficient
MMSE	Minimum Mean Squared Error
MOG	Mixture of Gaussians
MVDR	Minimum Variance Distortionless Response
PCM	pulse code modulation
pdf	probability density function
PMC	Parallel Model Combination
SNR	Signal to Noise Ratio
SIR	Signal to Interference Ratio
TSP	Time Stretched Pulse
VAD	Voice Activity Detection

Bibliography

[Aal2002] Aalburg S., Beaugeant C., Stan S., Fingscheidt T. and Rosca J. "Single- and Two-Channel Noise Reduction for Robust Speech Recognition", *ITRW on Multi-Modal Dialogue in Mobile Environments*, Kloster Irsee, Germany, 2002.

[Abr1972] Abramowitz M. and Stegun I.A. Modified Bessel Functions I and K, §9.6 in "Handbook of Mathematical Functions with Formulas, Graphs and Mathematical Tables,", S. 374-377, Dover Publications, New York, 1972.

[Als1972] Alspach D. and Sorenson H. "Nonlinear Bayesian Estimation using Gaussian Sum Approximations," IEEE Transactions on Automatic Control, Vol. AC-17, No. 4, pp. 439-448, 1972.

[Ane2001] Anemüller J. "Across-Frequency Processing in Convolutive Blind Source Separation", Ph.D. Thesis, Universität Oldenburg, Juli 2001.

[Aos1981] Aoshima N. "Computer-generated pulse signal applied for sound measurement," J. Acoust. Soc. Am. **69**(5), pp. 1484-1488, 1981.

[Ara2001] Araki S., Makino S., Mukai R., Nishikawa T. and Saruwatari H. "Fundamental Limitation of Frequency Domain Blind Source Separation for Convolved Mixture of Speech," *Proc. ICA 2001*, pp. 132-137, 2001.

[Asa2003] Asano F., Ikeda S., Ogawa M., Asoh H. and Kitawaki N. "Combined approach of array processing and independent component analysis for blind separation of acoustic signals," IEE Trans. Speech and Audio Processing **11**(3), pp. 204-215, 2003.

[Ast2005] Fernandez Astudillo R. "Implementierung eines Tied-Mixture-Models unter Matlab," Diploma Thesis, TU Berlin, 2005.

[Ata1982] Atal B. "Predictive Coding of Speech at Low Bit Rates," IEE Trans. Communications **30**(4), pp. 600-614, 1982.

[Ath2004] Athineos M., Hermansky H. and Ellis D. "PLP^2: Autoregressive modelling of auditory-like 2-D spectro-temporal patterns," *Proc. ISCA Tutorial Workshop Statistical and Perceptual Audio Processing SAPA-04*, pp. 37-42, October 2004.

[Att1998] Attias, H. "Independent Factor Analysis," Neural Computation **11**, pp. 803-851, November 1998.

[Att2001] Attias, H. "ICA, graphical models, and variational methods," in "Independent Component Analysis," pp. 95-112, Cambridge University Press, Cambridge, UK, 2001.

[Bal2001] Balan R., Rickard S. and Rosca J. "Robustness of Parametric Source Demixing in Echoic Environments," *Proc. ICA 2001*, pp. 144-149, 2001.

[Bal2002] Balan R. and Rosca J. "Microphone Array Speech Enhancement by Bayesian Estimation of Spectral Amplitude and Phase," *Proc. IEEE Sensor Array and Multichannel Processing Workshop SAM2002*, August 2002.

[Ban2003] Banno H., Shinde T., Takeda K. and Itakura F. "In-Car Speech Recognition using Distributed Microphones - Adapting to Automatically Detected Driving Conditions," *Proc. ICASSP 2003*, pp. 324-327, Vol. 1, April 2003.

[Bark2001] Barker J., Green P. and Cooke M. "Linking Auditory Scene Analysis and Robust ASR by Missing Data Techniques," *Proc. WISP 01, Stratfort, UK*, pp. 295-307, April 2001

[Bau2001] Baumann W., Köhler B.-U., Kolossa D. and Orglmeister R. "Real-Time Separation of Convolutive Mixtures," *Proc. ICA 2001*, pp. 65-69, December 2001.

[Bau2003] Baumann W., Kolossa D. and Orglmeister R. “Beamforming-Based Convolutive Source Separation,”*Proc. ICASSP 2003*, pp. 357-360, Vol. 5, April 2003.

[Bau2005] Baumann W. “Optimierung frequenzvarianter Nullbeamformer für akustische Signale mittels Statistik höherer Ordnung - Anwendungen im Kfz und in Büroräumen,” Ph.D. Thesis, TU Berlin, December 2005.

[Bea1991] Beattie V. and Young S. “Noisy Speech Recognition using Hidden Markov Model State-Based Filtering,” *Proc. ICASSP 1991*, pp. 917-920, 1991.

[Bel2000] Bell K., Ephraim Y. and Van Trees H. “A Bayesian Approach to Robust Adaptive Beamforming,” IEE Trans. Signal Processing **48**(2), pp. 386-398, 2000.

[Ben2003] Benaroya L. and Bimbot F. “Wiener Based Source Separation with HMM/GMM using a Single Sensor,” *Proc. ICA 2003*, pp. 957-961, 2003.

[Bit1999] Bitzer J., Simmer K.U. and Kammeyer K. “Multimicrophone noise reduction techniques for handsfree speech recognition - a comparative study,” *Proc. Robust Methods for Speech Recognition in Adverse Conditions ROBUST-99*, pp. 171-174, Tampere, Finland, May 1999.

[Bog1963] Bogert B., Healy M. and Tukey J. “The Quefrency Alanysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum, and Saphe Cracking,” *Proc. of the Symposium on Time Series Analysis*, Ch. 15, pp. 209-243, Wiley, New York, 1963.

[Bol1979] Boll S. “Suppression of Noise in Speech using Spectral Subtraction,” IEE Trans. Audio, Speech and Signal Processing **27**, pp. 113-120, 1979.

[Bra1983] Brandwood D.H. “A complex gradient operator and its application to adaptive array theory,” IEE Proc. pts. F and H **130**(1), pp. 11-16, February 1983.

[Bre1999] Bregman A. “Auditory Scene Analysis,” MIT Press, Cambridge, MA, USA, 1999.

[Bro2001] Bronstein I., Semendjajew K., Musiol G. and Mühlig H.“Taschenbuch der Mathematik, fünfte Ausgabe,” Verlag Harry Deutsch, Germany, 2001.

[Buc2003] Buchner H., Aichner R. and Kellermann W. “Blind Source Separation for Convolutive Mixtures Exploiting Nongaussianity Nonwhiteness and Nonstationarity,” *Proc. IWAENC 2003*, pp. 275-278, Kyoto, Japan, September 2003.

[Cap1995] Capdevielle V., Serviere Ch. and Lacoume J.L. “Blind separation of wide-band sources in the frequency domain,” *Proc. ICASSP 1995*, pp. 2080-2083, 1995.

[Car1993] Cardoso J.-F. and Souloumniac A. “Blind Beamforming for Non-Gaussian Signals,” IEE Proceedings F **140**(6), pp. 362-370, December 1993.

[Car1999] Cardoso J.-F. “High order contrasts for independent component analysis”, Neural Computation **11**, pp. 157-192, 1999.

[Car2001] Cardoso J.-F. “The three easy routes to independent component analysis,” *Proc. ICASSP 2001*, San Diego, USA, 2001.

[Cap1967] Capon J., Greenfield, R.J. and Kolker R.J. “Multidimensional maximum-likelihood processing of a large aperture seismic array,” Proceedings of the IEEE *55*(2), pp. 192-211, 1967.

[Cap1979] Capon J. “Maximum-likelihood spectral estimation,” in: S. Haykin, editor, *Nonlinear methods of spectral analysis*, pp. 155-179, Springer, New York, 1979.

[Che2004] Chen B., Zhu Q. and Morgan N. “Learning long term temporal features in LVCSR using neural networks,” *Proc. ICSLP 2004*, pp. 612-615, 2004.

[Cla1993] Class F., Kaltenmeier A. and Regel-Brietzmann P. “Evaluation of an HMM Speech Recognizer with various Continuous Speech Databases,” *Proc. Eurospeech 2003*, Vol. 3, pp. 1587-1590, 2003.

[Coh2001] Cohen I. “On Speech Enhancement under Signal Presence Uncertainty,” *Proc. ICASSP 2001*, Vol. 1, pp. 661-664, 2001.

[Coh2002] Cohen I. and Berdugo B. “Noise Estimation by Minima Controlled Recursive Averaging for Robust Speech Enhancement,” Signal Processing Letters **9**(1), pp. 12-15, 2002.

[Cou2002] Coulter G. and Vogt G. “The effects of space flight on the human vestibular system,” report number EB-2002-09-011-KSC, National Aeronautics and Space Administration, Life Sciences Division, Washington, DC, 2002.

[Cou2000] Couvreur, C. and Van Hamme, H. “Model Based Feature Enhancement for Noisy Speech Recognition,” *Proc. ICASSP 2000*, pp. 1719-1722, 2000.

[Coo2001] Cooke M., Green P., Josifovski L. and Vizinho A. “Robust automatic speech recognition with missing and unreliable acoustic data,” Speech Communication, pp. 267-285, 2001.

[Dav1980] Davis S. and Mermelstein P. “Comparison of parametric representationsof monosyllabic word recognition in continuously spoken sentences”, IEEE Trans. Acoustics, Speech and Signal Processing **28**, pp. 357-366, 1980.

[Del1993] Deller J., Proakis J. and Hansen J. “Discrete-Time Processing of Speech Signals,” Prentice Hall, Upper Saddle River, New Jersey, USA, 1993.

[Dem1977] Dempster A., Laird N. and Rubin D. “Maximum Likelihood from Incomplete Data via the EM Algorithm”, Journal of the Royal Statistical Society **39**, pp. 1-38, 1977.

[Den2001] Deng L. et al “High-Performance Robust Speech Recognition Using Stereo Training Data,” *Proc. ICASSP 2001*, pp. 301-304, 2001.

[Den2002] Deng L., Droppo J. and Acero A. "Exploiting Variances in Robust Feature Extraction Based on a Parametric Model of Speech Distortion," *Proc. ICSLP 2002*, pp. 2449-2452, 2002.

[Den2005] Deng L., Droppo J. and Acero A. "Dynamic Compensation of HMM Variances using the Feature Enhancement Uncertainty computed from a Parametric Model of Speech Distortion," IEEE Trans. Speech and Audio Processing **13**(3), pp. 412-421, 2005.

[Den2004] Deng Y., Chakrabartty S. and Cauwenberghs G. "Analog Auditory Perception Model for Robust Speech Recognition," *Proc. IEEE Conf. Neural Networks*, Vol. 3, pp. 1705-1709, 2004.

[Div2005] Divenyi P. (Ed.) "Speech Separation by Humans and Machines," Kluwer Academic Publishers, Dordrecht, Netherlands, 2005.

[Dro2002] Droppo J., Acero A. and Deng L. "Uncertainty Decoding with SPLICE for Noise Robust Speech Recognition," *Proc. ICASSP 2002*, pp. 57-60, 2002.

[Ehl1997] Ehlers F. and Schuster H.G. "Blind Separation of Convolutive Mixtures and an Application in Automatic Speech Recognition in Noisy Environment," IEEE Trans. Signal Processing **45**(10), pp. 2608-2612, 1997.

[Ell2006] Ellis D. "Model-Based Scene Analysis,", in "Computational Auditory Scene Analysis: Principles, Algorithms, and Applications," Wiley/IEEE Press, to appear.

[Eph1984] Ephraim Y. and Malah D. "Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator," IEEE Trans. Acoustics, Speech, and Signal Processing **32**(6), pp. 1109-1121, 1984.

[Eph1985] Ephraim Y. and Malah D. "Speech enhancement using a minimum-mean square error log-spectral amplitude estimator," IEEE Trans. Acoustics, Speech, and Signal Processing **33**(2), pp. 443-445, 1985.

[Eph1995] Ephraim Y. and Van Trees H. “A signal subspace approach for speech enhancement,” IEEE Transactions on Speech and Audio Processing **34**, pp. 251-266, 1995.

[ETS2003A] ETSI Standard ES 201 108 “Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Front End feature extraction algorithm; Compression algorithms,” September 2003.

[ETS2003B] ETSI Standard ES 202 050 “Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Advanced front End feature extraction algorithm; Compression algorithms,” November 2003.

[Fel1984] Fellbaum K. “Sprachverarbeitung und Sprachübertragung,” Springer Verlag, Berlin, Heidelberg, New York, Tokyo, 1984.

[Fro1972] Frost O.L. “An algorithm for linearly constrained adaptive array processing”, Proc. IEEE **60**, pp. 926-935, August 1972.

[Gal1995] Gales M. “Model-Based Techniques for Noise Robust Speech Recognition”, Ph.D. Thesis, University of Cambridge, September 1995.

[Gal1995b] Gales M. and Young S. “A fast and flexible implementation of parallel model combination,” *Proc. ICASSP 1995*, pp. 133-136, 1995.

[Gan2001] Gannot S., Burshtein D. and Weinstein E. “Signal enhancement using beamforming and nonstationarity with applications to speech,” IEEE Trans. Signal Processing **49**(8), pp. 1614-1626, August 2001.

[Gar2005] Garau G., Renals S. and Hain T. “Applying Vocal Tract Length Normalization to Meeting Recordings”, *Proc. Interspeech 2005*, September 2005.

[Gau1994] Gauvain J.-L. and Lee C.-H. “Maximum A Posteriori Estimation for Multivariate Gaussian Mixture Observations of Markov Chains,” IEEE Trans. Speech and Audio Processing **2**, pp. 291-298, April 1994.

[Gri1969] Griffiths L.J. “A simple adaptive algorithm for real-time processing in antenna arrays”, Proceedings of the IEEE **57**(10), pp. 1696-1704, October 1969.

[Gri1982] Griffiths L.J. and Jim C.W. “An alternative approach to linearly constrained adaptive beamforming”, IEEE Trans. Ant. Prop. **AP-30**, pp. 27-34, January 1982.

[Hae1994] Haeb-Umbach R., Geller D. and Ney H. “Improvements in connected digit recognition using linear discriminant analysis and mixture densities,” *Proc. ICASSP 1994*, Vol. 2, pp. 239-242, 1994.

[Hai2001] Haiber F. “Beschreibung der Configurationsparameter (.CNF-file) für SUN- und PC-Erkenner”, DaimlerChrysler internal documentation file, 2001.

[He2003] He X. and Chou W. “Minimum Classification Error Linear Regression for Acoustic Model Adaptation of Continuous Density HMMs,” *Proc. ICASSP 2003*, Vol. 1, pp. 556-559, 2003.

[Hei2001] Heisterkamp P. “Linguatronic: Product-Level Speech System for Mercedes-Benz Car,” *Proc.* 1^{st} *Int. Conf. on Human Language Technology Research*, J. Allan, ed., Morgan Kaufmann, San Francisco, USA, 2001.

[Hem2004] Hemmert W., Holmberg M. and Gelbart D. “Auditory-based Automatic Speech Recognition,” *Proc. ISCA Tutorial Workshop Statistical and Perceptual Audio Processing SAPA-04*, October 2004.

[Her1994] Hermansky H. and Morgan N. “RASTA Processing of Speech,” IEEE Trans. Speech and Audio Processing **2**, pp. 578-589, 1994.

[Her1990] Hermansky H. “Perceptual Linear Prediction (PLP) Analysis of Speech,” J. Acoust. Soc. Am. **87**(4), pp. 1738-1752, 1990.

[Hir2002] Hirsch G. “Experimental Framework for the Performance Evaluation of Speech Recognition Front Ends in a large Vocabulary Task,” ETSI STQ-Aurora DSR Working Group, December 2002.

[Hua2001] Huang F., Acero A. and Hon H.-W. "Spoken Language Processing," Prentice Hall, Upper Saddle River, NJ, USA, 2001.

[Hub1965] Huber P.J. "A robust version of the probability ratio test," Ann. Math. Statist. **32**(4), pp. 1753-1758, 1965.

[Hun1989] Hunt M. and Lefebvre C. "A comparison of several acoustic representations for speech recognition with degraded and undegraded speech," *Proc. ICASSP 1989*, pp. 262-265, 1989.

[Hue2002] Hüning H., Breton A., Haiber U. and Class F. "Speech Recognition Methods and their Potential for Dialogue Systems in Mobile Environments", *ITRW on Multi-Modal Dialogue in Mobile Environments*, Kloster Irsee, Germany, 2002.

[Huo2000] Huo Q. and Lee C.-H. "A Bayesian Predictive Classification Approach to robust speech recognition," IEEE Trans. Audio Processing **8**, pp. 200-204, 2000.

[Huy2004] Huynh D. "Implementierung von Hidden Markov-Modellen zur Spracherkennung unter Matlab," Diploma Thesis, TU Berlin, 2004.

[Hyv2001] Hyvärinen A., Karhunen J. and Oja E. "Independent Component Analysis," John Wiley and Sons, New York, USA, 2001.

[Jia2003] Jia C., Ding P. and Xu B. "Sequential MAP Estimation based Speech Feature Enhancement for Noise Robust Speech Recognition," *Proc. ICASSP 2003*, Vol. 1, pp. 412-415, 2003.

[Jia1999] Jiang H., Hirose K. and Huo Q. "Robust Speech Recognition based on a Bayesian Prediction Approach," IEEE Trans. Speech and Audio Processing **7**, pp. 426-440, 1999.

[Joh1993] Johnson D. and Dudgeon D. "Array Signal Processing: Concepts and Techniques," Prentice Hall, Englewood Cliffs, NJ, USA, 1993.

[Jou2000] Jourjine A., Rickard S. and Yilmaz Ö. "Blind Separation of Disjoint Orthogonal Signals: Demixing N Sources from 2 Mixtures," *Proc. ICASSP 2000*, pp. 2985-2988, 2000.

[Jul1996] Julier S.J. and Uhlmann J.K. "A General Method for Approximating Nonlinear Transformations of Probability Distributions," Technical Report, Dept. of Engineering Science, University of Oxford, UK, 1996.

[Jun1993] Junqua J.C. "The Lombard Reflex and its Role in Human Listeners and Automatic Speech Recognition," Journal of the Acoustical Society of America **93**(1), pp. 510-524, 1993.

[Kai2000] Kailath T., Sayed A. and Hassibi B. "Linear Estimation," Prentice Hall, Upper Saddle River, NJ, USA, 2000.

[Kim2006] Kim T., Eltoft T. and Lee T.-W "Independent Vector Analysis: An Extension of ICA to Multivariate Components," *Proc. ICA 2006*, pp. 165-172, 2006.

[Ken1998] Kennedy R., Abhayapala T. and Ward D. "Broadband Nearfield Beamforming using a Radial Beampattern Transformation," IEEE Trans. Signal Processing **46**(8), pp. 2147-2156, 1998.

[Kle2002] Kleinschmidt M. and Gelbart D. "Improving word accuracy with Gabor feature extraction," *Proc. ICSLP2002*, pp. 25-28, September 2002.

[Kna2003] Knaack M. "Rekonstruktion gestörter Maschinengeräusche durch mehrkanalige Signalverarbeitung", Ph.D. Thesis, TU Berlin, October 2003.

[Koe2005] Koehler B.-U. "Konzepte der statistischen Signalverarbeitung," Springer Verlag, Berlin, Germany, 2005.

[Kol2004] Kolossa D. and Orglmeister R. "Nonlinear Postprocessing for Blind Speech Separation," *Proc. ICA 2004*, in Lecture Notes in Computer Science, Vol. 3195, pp. 832-839, Springer, Berlin, 2004.

[Kol2005] Kolossa D., Klimas A. and Orglmeister R. "Separation and Robust Recognition of Noisy, Convolutive Speech Mixtures using Time-Frequency Masking and Missing Data Techniques," *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics 2005*, October 16-19, New Paltz, NY, 2005.

[Kou2002] Koutras A. and Dermatas E. "Robust Speech Recognition in a High Interference Real Room Environment using Blind Speech Extraction," *Proc. 14th International Conference on Digital Signal Processing (DSP 2002)*, pp. 167-171, 2002.

[Kri2001] Kristjansson T., Frey B., Deng L. and Acero A. "Towards non-stationary model-based adaptation for large vocabulary speech recognition," *Proc. ICASSP 2001*, pp. 337-340, 2001.

[Kur2000] Kurita S., Saruwatari H., Kajita S., Takeda K. and Itakura F. "Evaluation of blind signal separation method using directivity pattern under reverberant conditions," *Proc. ICASSP 2000* Vol. 5, pp. 3140-3143, 2000.

[Lee2006] Lee I., Kim T. and Lee T.-W "Complex FastIVA: A Robust Maximum Likelihood Approach of MICA for Convolutive BSS," *Proc. ICA 2006*, pp. 625-632, 2006.

[Lee2003] Lee S.-Y., Kim C.-M., Won Y.-G. and Park H.-M. "Auditory Pathway Model and its VLSI Implementation for Robust Speech Recognition in Real-World Noisy Environment," *Proc. IEEE Int. Conf. Neural Networks and Signal Processing*, pp. 1728-1733, Nanjing, China, December 2003.

[Lee2005] Lee T.-W. "Blind Source Separation using Graphical Models," in "Speech Separation by Humans and Machines," pp. 55-64, Kluwer Academic Publishers, Dordrecht, Netherlands, 2005.

[Leg1994] Leggetter C.J. and Woodland P.C. "Speaker Adaptation of Continuous Density HMMs using Linear Regression," *Proc. ICSLP 1994*, pp. 451-454, 1994.

[Leg1995] Leggetter C.J. and Woodland P.C. "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," Computer Speech and Language **9**, pp. 171-185, 1995.

[LDC1993] Linguistic Data Consortium "TIDigits Speech Database: Studio Quality Speaker-Independent Connected-Digit Corpus," URL: `http://morph.ldc.upenn.edu/Catalog/LDC93S10.html`, 1993.

[Lun1999] Lungwitz T. "Untersuchungen zur mehrkanaligen adaptiven Geräuschreduktion für die Spracherkennung im Kraftfahrzeug", Ph.D. Thesis, Christian-Albrechts-Universität Kiel, Shaker Verlag, Aachen, 1999.

[Mac2005] Macho D., Padrell J., Abad A., Nadeu C., Hernando J., McDonough J., Wolfel M., Klee U., Omologo M., Brutti A., Svaizer P., Potamianos G. and Chu S.M. "Automatic Speech Activity Detection, Source Localization, and Speech Recognition on the Chil Seminar Corpus," *Proc. ICME 2005 IEEE International Conference on Multimedia and Expo*, pp. 876-879, 2005.

[Mac2003] MacKay D.J.C. "Information Theory, Inference and Learning Algorithms," Cambridge University Press, Cambridge, UK, 2003.

[Mag2005] Maganti H.K., Vepa J. and Bourland H. "Continuous Microphone Array Speech Recognition on Wall Street Journal Corpus," Technical Report No. IDIAP-RR 05-47, Martigny, Switzerland, 2005.

[McA1980] McAulay R. and Malpass M. "Speech enhancement using a soft-decision noise suppression filter," IEEE Trans. Acoustics, Speech and Signal Processing **28**(2), pp. 137-145, 1980.

[McC2000] McCowan I., Marro C. and Mauuary L. "Robust Speech Recognition using Near-Field Superdirective Beamforming with Post-Filtering," *Proc. ICASSP 2000*, Vol. 3, pp. 1723-1726, 2000.

[Mey1997] Meyer J. and Simmer K.U. "Multi-Channel Speech Enhancement in a Car Environment using Wiener Filtering and Spectral Subtraction," *Proc. ICASSP 1997*, pp. 1167-1170, 1997.

[Mak2004] Mak B., Tam Y.-C. and Li P. "Discrimitave Auditory-Based Features for Robust Speech Recognition," IEEE Trans. Speech and Audio Processing **12**(1), pp. 27-36, January 2004.

[Mar1974] Markel, J. and Gray, A. "A linear prediction vocoder simulation based upon the autocorrelation method," IEEE Trans. Acoustics, Speech, and Signal Processing **22**(2), pp. 124-134, 1974.

[Mar1994] Martin R. "Spectral Subtraction Based on Minimum Statistics," *Proc. EUSIPCO 1994*, pp. 1182-1185, 1994.

[Mer1993] Merhav N. and Lee C.-H. "A minimax classification approach with application to robust speech recognition," IEEE Trans. Speech and Audio Processing **1**, pp. 90-100, 1993.

[Mer2000] van der Merwe R., Doucet A., de Freitas N. and Wan E. "The Unscented Particle Filter," CUED Technical Report No. CUED-F-INFENG/TR 380, Cambridge University Engineering Department, Cambridge, 2000.

[Moo2003] Moore D. and McCowan I. "Microphone Array Speech Recognition: Experiments on Overlapping Speech in Meetings," *Proc. ICASSP 2003*, Vol. 5, pp. 497-500, 2003.

[Mor1996] Moreno P.J. "Speech Recognition in Noisy Environments," Ph.D. Thesis, Carnegie Mellon University, Pittsburgh, PA, USA, April 1996.

[Mor1995] Morgan N. and Bourland H. "Continuous Speech Recognition," IEEE Signal Processing Magazine, pp. 25-42, 1995.

[Mor2005] Morgan N. et. al. "Pushing the envelope - Aside," IEEE Signal Processing Magazine **22**(5), pp. 81-88, September 2005.

[Muk2003] Mukai R., Sawada H., Araki S. and Makino S. "Robust Real-Time Blind Source Separation for Moving Speakers in a Room," *Proc. ICASSP 2003*, pp. 469-472, 2003.

[Muk2004] Mukai R., Sawada H., Araki S. and Makino S. "Near-Field Frequency Domain Blind Source Separation for Convolutive Mixtures," *Proc. ICASSP 2004*, Vol. 4, pp. 49-52, 2004.

[Mye1981] Myers C. and Rabiner L. "A comparative study of several dynamic time-warping algorithms for connected word recognition," The Bell System Technical Journal **60**(7), pp. 1389-1409, September 1981.

[Nis2003] Nishiura T., Nakayama M. and Nakamura S. "An evaluation of Adaptive Beamformer based on Average Speech Spectrum for Noisy Speech Recognition," *Proc. ICASSP 2003*, Vol. 1, pp. 668-671, 2003.

[Nol1993] Nolazco-Flores J. and Young S. "Adapting a HMM-Based Recogniser for Noisy Speech enhanced by Spectral Subtraction," *Proc. Eurospeech 1993*, pp. 829-832, 1993.

[Nol1994] Nolazco-Flores J. and Young S. "Continuous Speech Recognition in Noise using Spectral Subtraction and HMM-Adaptation," *Proc. ICASSP 1994*, Vol. 1, pp. 409-412, 1994.

[Omo1997] Omologo M., Matassoni M., Svaizer P. and Giuliani D. "Microphone Array Based Speech Recognition with Different Talker-Array Positions," *Proc. ICASSP 1997*, Vol. 1, pp. 227-230, 1997.

[Opp2004] Oppenheim A. and Schafer R. "From Frequency to Quefrency: A History of the Cepstrum," IEEE Signal Processing Magazine, September 2004.

[Par1999] Park H.-M., Jung H.-Y., Lee S.-Y. and Lee T.-W. "On subband-based blind separation for noisy speech recognition" *Proc. ICONIP '99*, Vol. 1, pp. 204-209, 1999.

[Par2000] Parra L. and Spence C. "Convolutive Blind Separation of Non-Stationary Sources," IEEE Trans. Speech and Audio Processing **8**, pp. 320-327, 2000. Implementation in Matlab by Stefan Harmeling, downloaded from `http://homepages.inf.ed.ac.uk/sharmeli/#sw`.

[Rab1978] Rabiner L. and Schafer R. "Digital Processing of Speech Signals," Prentice Hall, Englewood Cliffs, NJ, USA, 1978.

[Rab1989] Rabiner L. "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition," Proc. IEEE, pp. 257-286, 1989.

[Rab1993] Rabiner L. and Juang B.-H. "Fundamentals of Speech Recognition," Prentice Hall, Englewood Cliffs, NJ, USA, 1993.

[Raj2004] Raj B., Seltzer M. and Stern R. “Reconstruction of Missing Features for Robust Speech Recognition”, Speech Communication **43**, pp. 275-296, 2004.

[Raj2005] Raj B. and Stern R. “Missing-Feature Approaches in Speech Recognition,” IEEE Signal Processing Magazine **22**(5), pp. 101-116, 2005.

[Ren2000] Renevey P. and Drygajlo A. “Statistical Estimation of Unreliable Features for Robust Speech Recognition,” *Proc. ICASSP 2000*, Vol. 3, pp. 1731-1734, 2000.

[Rey2003] Reyes-Gomez J., Raj B. and Ellis D.P.W. “Multi-Channel Source Separation by Factorial HMMs” *Proc. ICASSP 2003*, Vol. 1, pp. 664-667, 2003.

[Ric2001] Rickard S., Balan R. and Rosca J. “Real-Time Time-Frequency Based Blind Source Separation,” *Proc. ICA 2001*, pp. 651-656, 2001.

[Rob2001] Roberts S. and Everson R. “Independent Component Analysis, Principles and Practice,” Cambridge University Press, Cambridge, UK, 2001.

[Rom2003] Roman N., Wang D. and Brown G. “Speech Segregation based on Sound Localization,” J. Acoust. Soc. Am. **114**(4), pp. 2236-2252, October 2003.

[Ros1997] Ross S. “Introduction to Probability Models,” Academic Press, San Diego, CA, USA, 1997.

[Row2000] Roweis S.T. “One Microphone Source Separation,” *Proc. Neural Information Processing Systems (NIPS) 2000*, pp. 793-799, 2000.

[Rut2001] Rutkowski T., Cichocki A. and Barros A. “Speech Enhancement from Interfering Sounds using CASA Techniques and Blind Source Separation,” *Proc. ICA 2001*, pp. 728-733, 2001.

[Sar1999] Saruwatari H., Kajita S., Takeda K. and Itakura F. “Speech Enhancement using Nonlinear Microphone Array Based on Complementary Beamforming,” IEICE Trans. Fundamentals **E 82-A**(8), pp. 1501-1510, August 1999.

[Saw2004] Sawada H., Mukai R., Araki S. and Makino S. "A Robust and Precise Method for Solving the Permutation Problem of Frequency-Domain Blind Source Separation," IEEE Trans. Speech and Audio Processing **12**(5), pp. 530-538, 2004.

[Saw2006] Sawada H., Araki S., Mukai R. and Makino, S. "Blind Extraction of Dominant Target Sources Using ICA and Time-Frequency Masking," IEEE Trans. Audio, Speech and Language Processing **14**(6), pp. 2165-2173, 2006.

[Seg2002] Segura J.C., Benitez M.C., de la Torre A. and Rubio A.J. "Feature Extraction Combining Spectral Noise Reduction and Cepstral Histogram Equalization for Robust ASR," *Proc. ICSLP 2002*, pp. 225-228, 2002.

[Sel2003] Seltzer M. and Stern R. "Subband Parameter Optimization of Microphone Arrays for Speech Recognition in Reverberant Environments," *Proc. ICASSP 2003*, Vol. 1, pp. 408-411, 2003.

[Sen1984] Seneff S. "Pitch and spectral estimation of speech based on auditory synchrony model," *Proc. ICASSP 1984*, Vol. 9, pp. 45-48, 1984.

[Ser2003] Serviere C. "Separation of Speech Signals with Segmentation of the Impulse Responses under Reverberant Conditions," *Proc. ICA 2003*, pp. 511-516, Nara, Japan, April 2003.

[Sko2004] Skowronski M. "Biologically inspired noise-robust speech recognition for both man and machine", Ph.D. Thesis, University of Florida, May 2004.

[Sla1993] Slaney M. and Lyon R. "On the importance of time - A temporal representation of sound," in "Visual Representations of Speech Signals," John Wiley, Sussex, England, 1993.

[Sma1997] Smaragdis P. "Information Theoretic Approaches to Source Separation," MaSc Thesis, MIT, 1997.

[Sri2004] Srinivasan S., Roman N. and Wang D. "On binary and ratio Time-Frequency Masks for Robust Speech Recognition," *Proc. ICSLP 2004*, pp. 2541-2544, 2004.

[Ste1992] Stern R., Liu F.-H., Ohshima Y., Sullivan T. and Acero A. "Multiple Approaches to Robust Speech Recognition," *Proc. Speech and Natural Language 1992*, pp. 274-279, Harriman, NY, February 1992.

[Suz1995] Suzuki Y., Asano F., Kim H.-Y. and Sone T. "An optimum computer-generated pulse signal suitable for the measurement of very long impulse responses", J. Acoust. Soc. Am. **97**(2), February 1995.

[Syr1986] Syrdal A. and Gopal H.S. "A perceptual model of vowel recognition based on the auditory representation of American English vowels," J. Acoust. Soc. Am. **79**(4), pp. 1086-1100, April 1986.

[Tch1999] Tchorz J. and Kollmeier B. "A psychoacoustical model of the auditory periphery as front end for ASR," Proc. ASA/EAA/DEGA Joint Meeting on Acoustics, March 1999, Berlin, Germany.

[Tre2003] Trentin E. and Gori M. "Robust Combination of Neural Networks and Hidden Markov Models for Speech Recognition," IEEE Trans. Neural Networks **14**(6), pp. 1519-1531, November 2003.

[Van1990] Van Compernolle D., Ma W., Xie F. and Van Diest M. "Speech Recognition in Noisy Environments with the aid of Microphone Arrays," Speech Communication **9**(5-6), pp. 433-442, 1990.

[Van2004] Van Hamme H. "Robust Speech Recognition using Cepstral Domain Missing Data Techniques and Noisy Masks," *Proc. ICASSP 2004*, Vol. 1, pp. 213-216, 2004.

[Vas2001] Vaseghi S. "Advanced Signal Processing and Digital Noise Reduction," John Wiley and Sons, New York, USA, 2001.

[Vis2003] Visser E., Otsuka M. and Lee T.-W. "A spatio-temporal speech enhancement scheme for robust speech recognition in noisy environments," Speech Communication **41**, pp. 393-407, 2003.

[Vis2003b] Visser E. and Lee T.-W. "Speech enhancement using Blind Source Separation and Two-Channel Energy Based Speaker Detection," *Proc. ICASSP 2003*, Vol. 1, pp. 836-839, 2003.

[Vit1967] Viterbi A. "Error bounds for convolutional codes and an asymptotically optimum decoding algorithm," IEEE Trans. Information Theory **13**(2), pp. 260-269, April 1967.

[Viz1999] Vizinho A., Green P., Cooke M. and Josifovski L. "Missing data theory, spectral subtraction and Signal-to-noise Estimation for robust ASR: An Integrated Study," *Proc. Eurospeech 1999*, September 1999.

[Wai1989] Waibel A., Hanazawa T., Hinton G., Shikano K. and Lang K. "Phoneme Recognition using Time Delay Neural Networks," IEEE Trans. Acoustics Speech and Signal Processing **37**(3), pp. 328-339, March 1989.

[Wan2001] Wan W. and Au O. "Robust Speech Recognition based on the second-order Difference Cochlear Model," *Proc. 2001 International Symposium on Intelligent Multimedia, Video and Speech Processing*, pp. 543-546, Hong Kong, May 2004.

[War1996] Ward D.B., Kennedy R.A. and Williamson R.C. "FIR filter design for frequency invariant beamformers," Signal Processing Letters **3**(3), pp. 69-71, March 1996.

[Wu2002] Wu J. and Huo Q. "Supervised Adaptation of MCE-Trained CD-HMMs using Minimum Classification Error Linear Regression," *Proc. ICASSP 2002*, Vol. 1, pp. 605-608, 2002.

[Yam1996] Yamada T., Nakamura S. and Shikano K. "Robust Speech Recognition with Speaker Localization by a Microphone Array," *Proc. ICSLP 1996*, pp. 1317-1320, 1996.

[Yap2003] Yapanel U. and Hansen J. "A New Perspective on Feature Extraction for Robust In-Vehicle Speech Recognition," *Proc. Eurospeech 2003*, pp. 1281-1284, 2003.

[Yil2004] Yilmaz Ö. and Rickard, S. “Blind Separation of Speech Mixtures via Time-Frequency Masking,” IEEE Trans. Signal Processing **52**(7), July 2004.

[You2002] Young S. et. al. “The HTK Book,” for HTK-Version 3.2.1., Cambridge University Engineering Department, Cambridge, 2002.

[Zha1997] Zhan P. and Waibel A. “Vocal tract length normalization for large vocabulary continuous speech recognition”, CMU Technical Report No. CMU-CS-97-148, Pittsburgh, USA, May 1997.

[Zwi1999] Zwicker E. and Fastl H. “Psychoacoustics, Facts and Models” 2nd Edition, Springer Verlag, Berlin, 1999.