18

Towards Robust Distant-Talking Automatic Speech Recognition in Reverberant Environments

Armin Sehr and Walter Kellermann

Multimedia Communications and Signal Processing
University of Erlangen-Nuremberg, Germany

In distant-talking scenarios, automatic speech recognition (ASR) is hampered by background noise, competing speakers and room reverberation. Unlike background noise and competing speakers, reverberation cannot be captured by an additive or multiplicative term in the feature domain because reverberation has a dispersive effect on the speech feature sequences. Therefore, traditional acoustic modeling techniques and conventional methods to increase robustness to additive distortions provide only limited performance in reverberant environments.

Based on a thorough analysis of the effect of room reverberation on speech feature sequences, this contribution gives a concise overview of the state of the art in reverberant speech recognition. The methods for achieving robustness are classified into three groups: Signal dereverberation and beamforming as preprocessing, robust feature extraction, and adjustment of the acoustic models to reverberation. Finally, a novel concept called reverberation modeling for speech recognition, which combines advantages of all three classes, is described.

18.1 Introduction

Even for difficult tasks, current state-of-the-art ASR systems achieve impressive recognition rates if a clean speech signal recorded by a close-talking microphone is used as input [51, 52]. In many applications however, using a close-talking microphone is either impossible or unacceptable for the user.

As an example, for the automatic transcription of meetings or lectures [2, 11], equipping each speaker with a close-talking microphone would be very inconvenient. Instead, distant microphones, e. g., placed at the meeting table, are used. Voice control of medical systems allows a surgeon to work with both hands while controlling diagnostic instruments or assistance devices.

Telephone-based speech dialogue systems for information retrieval or transactions, like telephone-based flight information desks or telephone banking systems, need to cope with users calling from hands-free telephones. Further applications of distant-talking ASR are dictation systems, information terminals, and voice-control of consumer electronics, like television sets or set-top boxes.

In all these scenarios, the distance between speaker and microphone is in the range of one to several meters. Therefore, the microphone does not only pick up the desired signal, but also additive distortions like background noise or competing speakers, and reverberation of the desired signal. While significant progress has been achieved over the last decades in improving the robustness of ASR to additive noise and interferences, the research on reverberation-robust ASR is still in its infancy. This contribution focuses on robust ASR in reverberant environments.

The chapter is structured as follows: The distant-talking ASR scenario is discussed in Sec. 18.2 and the different properties of additive distortions and reverberation are emphasized. Sec. 18.3 outlines how the measures for increasing robustness to reverberation are embedded into ASR systems and explains the basics of ASR which will be needed for describing these measures. The effect of reverberation on speech feature sequences is investigated in Sec. 18.4. The known approaches to achieve robust ASR in reverberant environments are classified into three groups:

- first, signal dereverberation and beamforming as preprocessing (Sec. 18.5),
- second, usage of robust features which are insensitive to reverberation or feature-domain compensation of reverberation (Sec. 18.6),
- third, adjustment of the acoustic models of the recognizer to reverberation by training or adaptation (Sec. 18.7).

A novel approach called reverberation modeling for speech recognition, which combines advantages of all three classes, is discussed in Sec. 18.8. It uses a statistical reverberation model to perform feature-domain dereverberation within the recognizer. Sec. 18.9 summarizes and concludes this contribution.

18.2 The Distant-Talking ASR Scenario

Fig. 18.1 shows a typical *distant-talking* ASR scenario. Compared to the close-talking scenario, the gain of the microphone amplifier has to be increased because of the greater distance between the desired speaker and the microphone. Therefore, the microphone does not only pick up the desired signal but also background noise, interfering speakers and the reverberation of the desired signal. The reverberation results from the fact that the desired signal does not only travel along the direct path from the speaker to the microphone, but is also reflected by walls and other obstacles in the enclosure. Therefore, the microphone picks up many delayed and attenuated copies of the desired signal

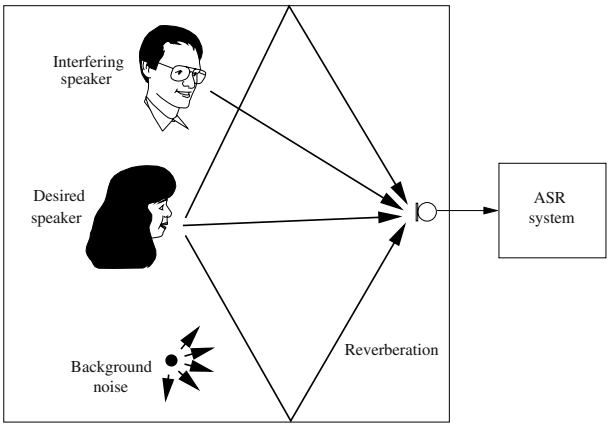

Fig. 18.1. Distant-talking ASR scenario.

which are perceived as reverberation. In the time domain, reverberation can be very well modeled by convolving the signal $s(n)$ of the desired speaker with the impulse response $h(n)$ describing the acoustic path between speaker and microphone [37]. Additive distortions, like background noise and interfering speakers, are modeled by the signal $b(n)$ so that the microphone signal $y(n)$ is given as

$$y(n) = h(n) * s(n) + b(n) \; . \tag{18.1}$$

The corresponding block diagram for the distant-talking signal capture is depicted in Fig. 18.2.

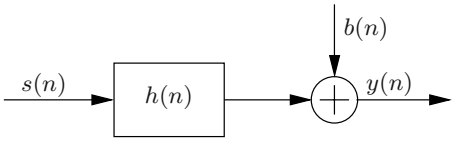

Fig. 18.2. Block diagram of distant-talking signal capture.

As the additive distortions $b(n)$ and the desired speech signal $s(n)$ result from different sources, they can be modeled as statistically independent random processes. Therefore, very effective methods for reducing additive distortions in the microphone signal, and for adjusting speech recognizers to additive distortions have been developed in the last decades. See [7, 8, 24, 26] and [33, 35] for overviews.

In contrast to that, the reverberation is strongly correlated to the desired signal and cannot be described by an additive term. Therefore, the approaches developed for additive distortions are not appropriate to increase robustness against reverberation.

As we will focus on reverberation in this chapter, we neglect the signal $b(n)$ in the following treatment so that the microphone signal is given as

$$y(n) = h(n) * s(n) \; . \tag{18.2}$$

Fig. 18.3 shows a typical *room impulse response* (RIR) measured in a lecture room. After an initial delay of approximately 12.5 ms, which is caused by the time the sound waves need to travel from the speaker to the microphone (here roughly 4 m), the first peak in the RIR is caused by the direct sound. Following the direct sound, several distinct peaks corresponding to prominent reflections can be observed. With increasing delay, more and more reflections overlap so that no distinct peaks but rather an exponentially decaying envelope characterizes the last part of the RIR.

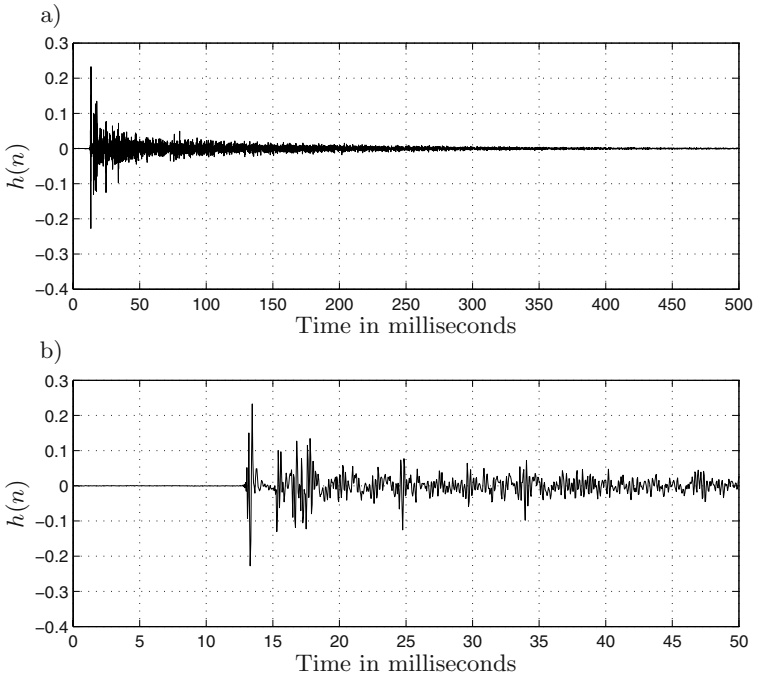

Fig. 18.3. RIR $h(n)$ of a lecture room in the time domain, a) complete RIR, b) first section of RIR.

The time needed for a 60 dB decay in sound energy is called the *reverberation time* T_{60}. Typical reverberation times are in the range of 20-100 ms in cars and 200-800 ms in offices or living rooms. In large lecture rooms, concert halls or churches, the reverberation time is often significantly longer than 1 s.

The *signal-to-reverberation ratio* (SRR) compares the energy of the direct sound to the energy of the reverberation and is defined as

$$SRR = 10\log_{10}\left\{\frac{\sum_{n=0}^{N_d-1} h^2(n)}{\sum_{n=N_d}^{N_h-1} h^2(n)}\right\},$$

where the first part of the RIR from $0 \ldots N_d - 1$ is considered as direct sound and the second part of the RIR from $N_d \ldots N_h - 1$ is considered as reverberation. The RIR is strongly time-variant. Already small changes in the position of the speaker or the microphone, movements of other objects, like doors, windows or persons, or variations in temperature change the details of the RIR significantly. However, its overall characteristics, like the reverberation time, the SRR and even the envelope of the time-frequency pattern corresponding to the RIR, are hardly affected by such changes.

Please note the distinction between reverberation and acoustic echoes. In this contribution, reverberation is used for multiple delayed copies of the desired signal, while in acoustic echo cancellation [9], the term echo is used to describe multiple delayed copies of interfering signals originating from loudspeakers.

18.3 How to Deal with Reverberation in ASR Systems?

This section discusses how the different approaches to increase robustness against reverberation can be embedded into an ASR system. For this purpose, the general task of ASR is formulated first, and the options for increasing robustness to reverberation are explained using a generic ASR block diagram.

The task of a speech recognizer can be formulated as finding the best estimate $\hat{W}$ of the true word sequence W_t corresponding to a certain utterance, given the respective speech signal $s(n)$. Usually, the recognizer does not use the speech signal itself but rather speech feature vectors $\boldsymbol{s}(k)$ derived from the speech signal. Denoting the sequence of all observed speech feature vectors $\boldsymbol{s}(1) \ldots \boldsymbol{s}(K)$ as $\boldsymbol{S}$, where K is the length of the sequence, the recognition problem can be expressed as maximizing the posterior probability $P(W|\boldsymbol{S})$ over all possible word sequences W

$$\hat{W} = \underset{W}{\operatorname{argmax}} \left\{P(W|\boldsymbol{S})\right\}. \tag{18.3}$$

Equivalently, the product of the likelihood and the prior probability can be maximized

$$\hat{W} = \underset{W}{\operatorname{argmax}} \left\{P(\boldsymbol{S}|W) \cdot P(W)\right\}. \tag{18.4}$$

The exact determination of both $P(\boldsymbol{S}|W)$ and $P(W)$ is very difficult in real-world systems. Therefore, the likelihood $P(\boldsymbol{S}|W)$ of observing the feature sequence $\boldsymbol{S}$ given the word sequence W is approximated by some acoustic score $A(\boldsymbol{S}|W)$, which is modeled by the *acoustic model*. The prior probability

$P(W)$ of the word sequence is approximated by some language score $L(W)$ and is modeled by a *language model* so that the recognition problem can be expressed as

$$\hat{W} = \underset{W}{\operatorname{argmax}} \left\{ A(\boldsymbol{S}|W) \cdot L(W) \right\} . \tag{18.5}$$

In a distant-talking scenario, the clean-speech feature sequence $\boldsymbol{S}$ is not available. Instead, the feature sequence $\boldsymbol{Y}$ derived from the reverberant microphone signal $y(n)$ has to be used so that the distant-talking recognition problem is given as

$$\hat{W} = \underset{W}{\operatorname{argmax}} \left\{ A(\boldsymbol{Y}|W) \cdot L(W) \right\} . \tag{18.6}$$

Robustness to reverberation is achieved, if the solution to the problem described in Eq. 18.6 is approaching the solution to the problem of Eq. 18.5. That is, the accuracy of the transcription determined from the reverberant sequence $\boldsymbol{Y}$ approaches the accuracy determined from the clean-speech sequence $\boldsymbol{S}$.

Fig. 18.4 shows a generic block diagram of an ASR system which is used to solve the problem described in Eq. 18.5. The speech signal is preprocessed in order to reduce distortions and then transformed into speech feature vectors. Before the recognizer can be used to determine the word sequence or transcription of unknown utterances, both its acoustic model and its language model have to be trained using training data with known transcriptions.

The function blocks where measures to increase robustness against reverberation can be embedded into the ASR system so that it can effectively solve problem 18.6 are marked by areas shaded in dark gray in Fig. 18.4. The attached labels point out the section where these measures are discussed.

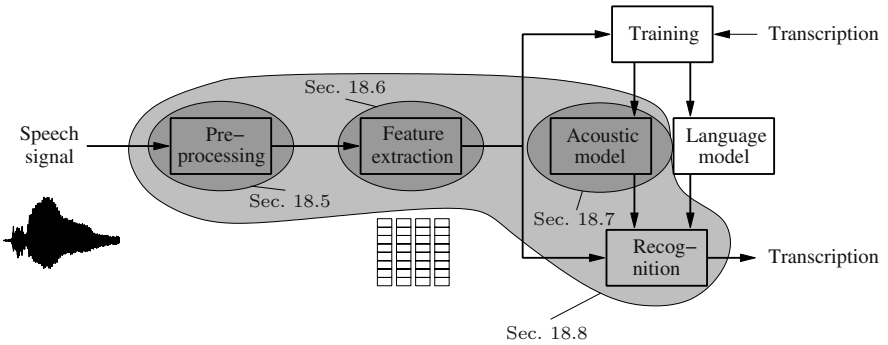

Fig. 18.4. Block diagram of a speech recognition system.

The novel concept of reverberation modeling for speech recognition implements the idea of preprocessing directly in the feature domain using an improved acoustic model to dereverberate the speech features during recognition (Sec. 18.8). In this way, robustness is achieved by utilizing four main

function blocks instead of only one as indicated by the area shaded in light gray. Therefore, the advantages of all three classes of approaches are utilized by the novel concept.

In the following, the blocks of the speech recognition system according to Fig. 18.4 are explained in more detail. The goal of the feature extraction is to reduce the dimension of the input data roughly by one order of magnitude (e. g., from 256 samples to 25 features). The features should concentrate all information of the speech signal which is necessary for the classification of different phones and words and all information irrelevant for speech recognition should be removed.

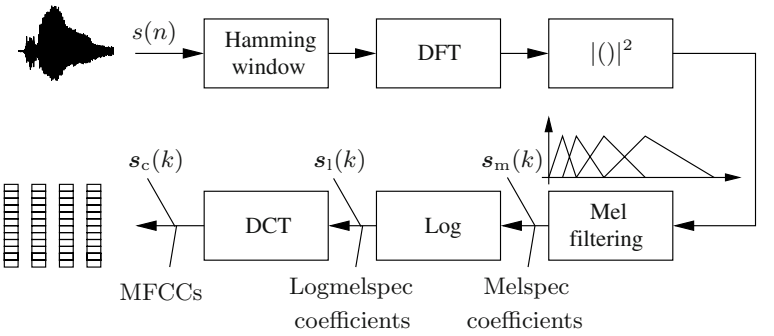

Fig. 18.5. Block diagram of the feature extraction for MFCCs.

Currently, the most popular speech features are the so-called *mel-frequency cepstral coefficients* (MFCCs) [12]. Their calculation is illustrated in Fig. 18.5. In the first step of the feature extraction, a short-time spectrum analysis is performed by windowing overlapping frames of the speech signal with a Hamming window $w(n)$ and applying an F-point discrete Fourier transform (DFT)

$$S(f,k) = \sum_{n=0}^{F-1} w(n)\, s(kN+n)\, e^{-j\frac{2\pi}{F}\,n\,f}\ , \tag{18.7}$$

where f is the index of the DFT bin, k is the frame index and $N \leq F$ is the frame shift. The magnitude square of the DFT coefficients $S(f,k)$ is filtered by a mel filter bank $C(l,f)$ to obtain the mel-spectral (*melspec*) coefficients

$$s_{\mathrm{m}}(l,k) = \sum_{f=0}^{F/2} C(l,f)\big|S(f,k)\big|^2\ , \tag{18.8}$$

where the subscript m denotes "melspec domain" and l is the index of the mel channels. Due to the symmetry of the DFT, it is sufficient to calculate the sum over $f = 0 \ldots F/2$. Like the human auditory system, the mel filter bank has a better frequency resolution for low frequencies than for high frequencies.

This is commonly realized by triangular weighting functions $C(l, f)$ for the mel channels as depicted in Fig. 18.6. The widths of these weighting functions increase with the channel number [33] and approximate a logarithmic spectral resolution similar to the human hearing. The feature vector $\boldsymbol{s}_{\mathrm{m}}(k)$ holds all melspec coefficients of frame k

$$\boldsymbol{s}_{\mathrm{m}}(k) = \left[s_{\mathrm{m}}(1,k), \ldots, s_{\mathrm{m}}(L,k)\right]^{\mathrm{T}} ,$$

where T denotes matrix transpose and L is the number of mel channels.

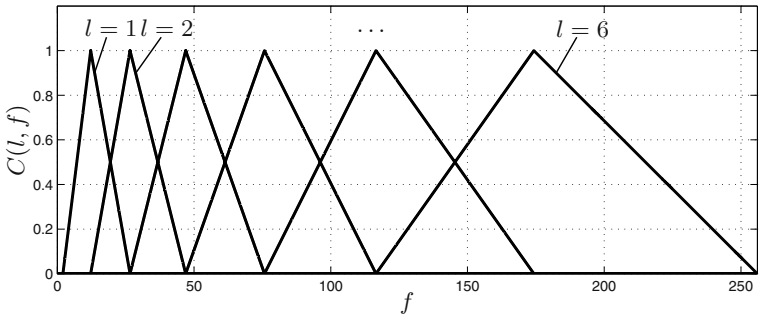

Fig. 18.6. Triangular weighting functions $C(l, f)$ of the mel filter bank for $F = 512$, $L = 6$.

Calculating the logarithm of the melspec coefficients, the logarithmic melspec (*logmelspec*) coefficients are obtained

$$\boldsymbol{s}_{\mathrm{l}}(k) = \log\left\{\boldsymbol{s}_{\mathrm{m}}(k)\right\} , \tag{18.9}$$

where the logarithm is performed element-wise and the subscript l denotes "logmelspec domain".

Due to the spectral overlap of the channels in the mel filter bank, both melspec and logmelspec coefficients are strongly correlated. Performing a discrete cosine transform (DCT) on the logmelspec features, the elements of the feature vectors are largely decorrelated and the MFCCs are obtained. For speech recognition, only the first $I \leq L$ MFCCs are important, and we have

$$\boldsymbol{s}_{\mathrm{c}}(k) = \boldsymbol{B}\,\boldsymbol{A}\,\boldsymbol{s}_{\mathrm{l}}(k) , \tag{18.10}$$

where the subscript c denotes "cepstral domain" (MFCC),

$$\boldsymbol{A} = \{a_{il}\} \text{ with } a_{il} = \sqrt{2/L} \cdot \cos\left(\pi/L \cdot i \cdot (l + 0.5)\right)$$

is the $L \times L$ DCT matrix, and the $I \times L$ selection matrix $\boldsymbol{B} = [\mathbf{1}_{I\times I} \;\; \mathbf{0}_{I\times(L-I)}]$ selects the first I elements of a $L \times 1$ vector by left multiplication. $\mathbf{1}_{I\times I}$ is the $I \times I$ identity matrix, and $\mathbf{0}_{I\times(L-I)}$ is an $I \times (L - I)$ matrix of zeros.

Note that $\boldsymbol{s}(k)$ is used in the following to denote the current clean-speech vector for relationships that hold regardless of the feature kind. Whenever we want to describe relations which are only valid for a certain feature kind, $\boldsymbol{s}(k)$ is replaced by $\boldsymbol{s}_{\mathrm{m}}(k)$, $\boldsymbol{s}_{\mathrm{l}}(k)$, or $\boldsymbol{s}_{\mathrm{c}}(k)$. The corresponding reverberant feature vector $\boldsymbol{y}(k)$ is derived in the same way using the reverberant speech signal $y(n)$.

Most state-of-the-art recognizers use *hidden Markov models* (HMMs) to describe the acoustic score $A(\boldsymbol{S}|W)$. The reasons for the prevalent use of HMMs are the efficient training and recognition algorithms available for HMMs and their ability to model both temporal and spectral variations. See [33, 34, 55] for comprehensive introductions to HMMs.

HMMs can be considered as finite state machines controlled by two random experiments. Fig. 18.7 shows a typical HMM topology used in speech recognition consisting of five states. The first random experiment controls the transition from the previous state $q(k-1)$ to the current state $q(k)$ according to the *state transition probabilities*

$$a_{ij} = P\big(q(k) = j | q(k-1) = i\big) \ .$$

In this way, different phoneme durations can be modeled. Only transitions from left to right are allowed, and we assume that the HMM starts in state 1 at frame 1 and ends in the last state J at the final frame K. The second random experiment determines the output feature vector according to the *output density* $f_\lambda(q(k), \boldsymbol{s}(k))$ of the current HMM state $q(k)$ so that spectral variations in the pronunciation can be captured. In summary, an HMM λ is defined by its transition probabilities a_{ij}, its output densities $f_\lambda(q(k), \boldsymbol{s}(k))$ and the initial state probabilities. Since we always assume that the HMM starts in state 1 at frame 1, the initial state probabilities will be neglected in the following.

The HMM is based on two fundamental assumptions [33]:

- The (first-order) *Markov assumption* implies that the current state $q(k)$ depends only on the previous state $q(k-1)$.
- The *conditional independence assumption* implies that the current output feature vector $\boldsymbol{s}(k)$ depends only on the current state $q(k)$ and not on previous states or previous output feature vectors.

Based on these two assumptions, very effective algorithms for training and recognition could be derived [4, 5].

The clean-speech feature sequence $\boldsymbol{s}(k)$ can be considered as a realization of the vector-valued random process $\mathcal{S}(k)$. The HMM λ models this random process as a non-stationary random process. Because of the conditional independence assumption, two statistically independent random vectors $\mathcal{S}_{k_1}$ and $\mathcal{S}_{k_2}$ are obtained if $\mathcal{S}(k)$ is observed at different frames $k_1 \neq k_2$. Since in the real world, neighboring feature vectors of a clean-speech feature sequence exhibit some statistical dependence, the conditional independence assumption

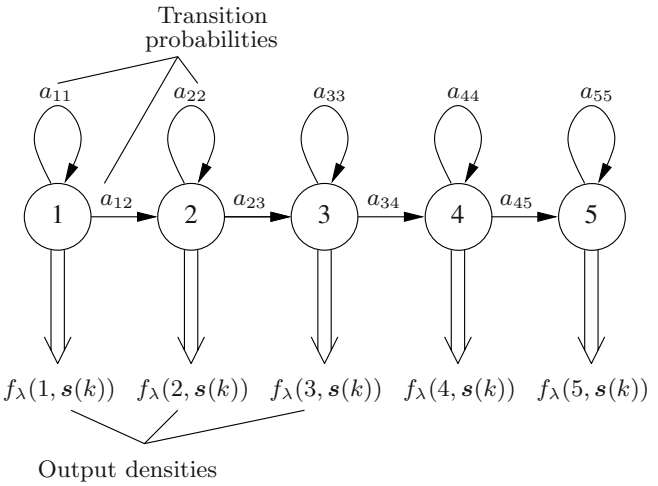

Fig. 18.7. Typical HMM topology used in ASR.

is already a simplification if the HMM is used to model clean-speech feature sequences. In practice however, it has turned out that HMM-based recognizers achieve remarkable recognition rates for clean speech despite this simplification. If HMMs are used to model reverberant feature sequences with much stronger statistical dependencies between frames (see Sec. 18.4), the conditional independence assumption becomes a severe limitation of the model's capability to describe $A(\boldsymbol{Y}|W)$.

The output density $f_\lambda(q(k), \boldsymbol{s}(k))$ of the current HMM state describes the conditional density of the random process $\mathcal{S}(k)$ given the current state $q(k)$

$$f_\lambda\big(q(k), \boldsymbol{s}(k)\big) = f_{\mathcal{S}(k)|q(k)}\big(\boldsymbol{s}(k)\big) \; . \tag{18.11}$$

The task of determining the parameters of an HMM given a set of utterances with known transcription is called training. The parameters of the HMM are chosen so that the probability of observing the feature sequences corresponding to the training utterances is maximized. Usually the Baum-Welch algorithm is used to solve this maximization problem (see e. g. [33, 55]).

To describe the probability $P(\boldsymbol{S}|W)$ or the acoustic score $A(\boldsymbol{S}|W)$ by HMMs, the sequence of words W has to be split into smaller units. For each of these units an HMM is trained. The complexity of this acoustic-phonetic modeling depends on the vocabulary size of the recognition task. For small vocabularies, like e. g., in digit recognition, it is possible to model each word with its own word-level HMM. For large vocabularies, it is more efficient to model subword units like phonemes with HMMs. Due to coarticulation phenomena, the pronunciation of subword units strongly depends on their contexts. Therefore, training different HMMs for the same phoneme with different contexts increases the accuracy of the acoustic-phonetic modeling. Triphones

which consider both the previous and the following phoneme are often used in large-vocabulary recognizers (see e. g. [33]).

For continuous speech recognition, HMM networks are constructed incorporating the grammar of the recognition task (see Fig. 18.8) and the pronunciation dictionaries, specifying the subword units which make up a word (see e. g. [78]).

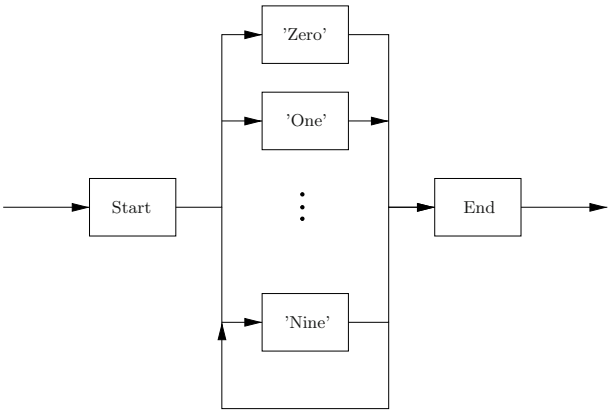

Fig. 18.8. Task grammar for connected digit recognition, 'start' and 'end' is associated with start and end of the utterance.

The information given by a language model can also be included. These recognition networks can be considered as large HMMs. Fig. 18.9 shows a very simple HMM network $\mathcal{N}_\lambda$ for connected digit recognition which can be considered as one of the simplest examples of continuous speech recognition.

Given the feature sequence of the utterance to be recognized, the recognizer searches for the most likely path through the recognition network and records the words along this path so that the most likely transcription can be determined. The *Viterbi algorithm* can be used to find the most likely path through the HMM network.

As we will focus on the determination of the acoustic score, we use a notation similar to [49] to separate the acoustic score and the language score. A large number of search algorithms exists for solving the resulting search problem 18.5, see [33, 49] for overviews. To simplify the search, the acoustic score $A(\boldsymbol{S}|W)$ approximates the probability $P(\boldsymbol{S}|W)$ by only considering the most likely state sequence through the HMM sequence Λ describing W. If, for example, the recognition task is connected digit recognition based on word-level HMMs and the word sequence W is "three, five, nine", the HMM sequence Λ corresponding to W is the concatenation of the word-level HMMs $\lambda_{\text{'three'}}$, $\lambda_{\text{'five'}}$, and $\lambda_{\text{'nine'}}$. Then, the acoustic score $A(\boldsymbol{S}|W)$ can be expressed as

$$A(\boldsymbol{S}|W) = \max_{Q} \left\{ P(\boldsymbol{S}, Q|\Lambda) \right\} , \tag{18.12}$$

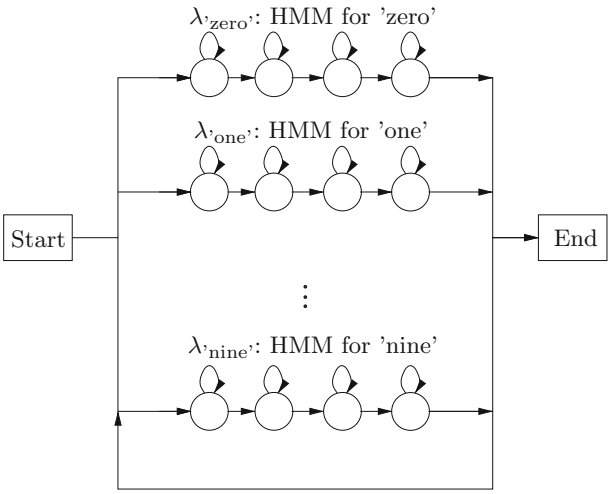

Fig. 18.9. HMM network $\mathcal{N}_\lambda$ for connected digit recognition.

where the maximization is performed over all allowed state sequences Q through Λ.

To calculate the acoustic score $A(\boldsymbol{S}|W) = A(\boldsymbol{S}|\Lambda)$, the Viterbi algorithm, defined by the following equations, is commonly used. Note that it is assumed that the HMM starts in state 1 and ends in the last state J at the final frame K of the sequence $\boldsymbol{S}$.

Initialization:

$$\begin{aligned} \gamma_1(1) &= f_\Lambda\big(1, \boldsymbol{s}(1)\big) \,, \\ \gamma_j(1) &= 0 \quad \forall j = 2 \ldots J \,, \\ \psi_j(1) &= 0 \quad \forall j = 1 \ldots J \,. \end{aligned}$$

Recursion:

$$\begin{aligned} \gamma_j(k) &= \max_i \{\gamma_i(k-1) \cdot a_{ij}\} \cdot f_\Lambda(j, \boldsymbol{s}(k)) \,, \\ \psi_j(k) &= \operatorname*{argmax}_i \{\gamma_i(k-1) \cdot a_{ij}\} \,. \end{aligned} \tag{18.13}$$

Termination:

$$A(\boldsymbol{S}|W) = \gamma_J(K), \quad q(K) = J \,.$$

Backtracking:

$$q(k) = \psi_{q(k+1)}(k+1) \quad \forall k = K-1, \ \ldots, \ 1 \,.$$

Here, i indexes all considered previous states leading to the current state j, $\gamma_j(k)$ is the Viterbi metric for state j at frame k. The greater the Viterbi

metric $\gamma_j(k)$, the more likely is the corresponding partial sequence up to frame k ending in state j. $f_\Lambda(j, \boldsymbol{s}(k))$ is the output density of state j of the HMM sequence Λ describing W evaluated for the clean-speech vector $\boldsymbol{s}(k)$. The backtracking pointer $\psi_j(k)$ refers to the previous state and allows backtracking of the most likely state sequence.

The Viterbi algorithm can be illustrated by a trellis diagram as depicted in Fig. 18.10 for the HMM of Fig. 18.7. The vertical axis represents the states (from 1 to 5 in the given example) and the horizontal axis represents the frames. Each dot in the diagram corresponds to the Viterbi metric $\gamma_j(k)$ of state j for frame k and each arc between the dots illustrates the non-zero transition probability between the respective states. The Viterbi scores are calculated from left to right by multiplying the score of the possible predecessor states with the corresponding transition probability, selecting the maximum over all predecessors, and then multiplying the output density of the current state for the current feature vector as given in Eq. 18.13.

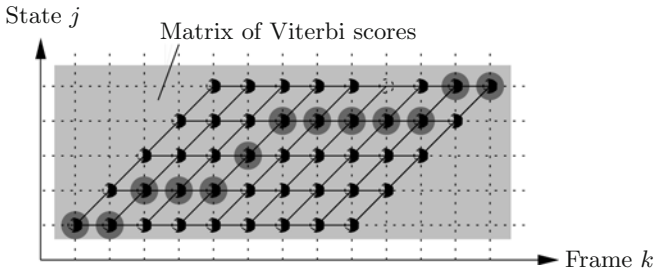

Fig. 18.10. Trellis diagram for the HMM of Fig. 18.7.

In this way, the Viterbi algorithm fills the matrix of Viterbi scores (see Fig. 18.11) with the elements $\gamma_j(k)$. At the same time, the backtracking matrix is filled with elements $\psi_j(k)$. As we assume that the HMM ends in the last state J, the final acoustic score is obtained by reading the Viterbi score $\gamma_J(K)$ of the dot in the upper right corner of the trellis diagram. Using the backtracking matrix, the most likely path through the HMM as indicated by the large dots in Fig. 18.10 for the current utterance is reconstructed.

18.4 Effect of Reverberation in the Feature Domain

This section investigates the effect of reverberation on the speech feature sequences used in ASR. Based on the exact description in the time domain (see Sec. 18.2), an approximative relationship between clean and reverberant feature vectors is derived.

The time-domain convolution of Eq. 18.2 is transformed to a multiplication in the frequency domain if the discrete-time Fourier transform is employed

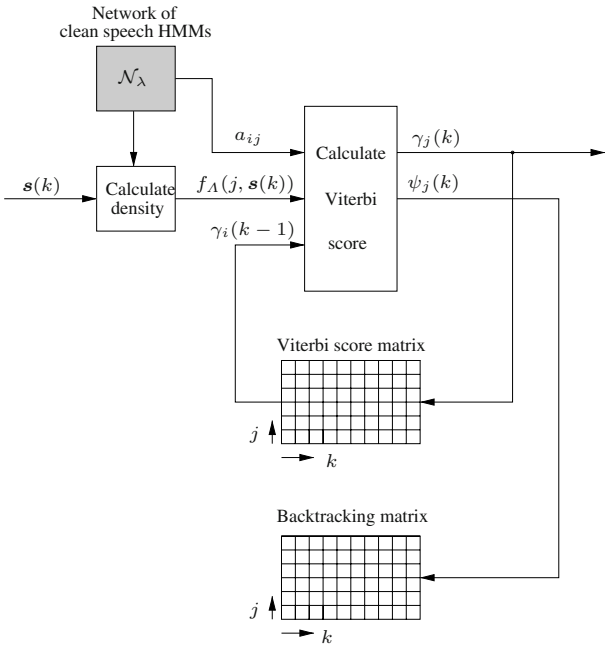

Fig. 18.11. Illustration of the Viterbi algorithm.

$$Y\left(e^{j\Omega}\right) = H\left(e^{j\Omega}\right) \cdot S\left(e^{j\Omega}\right) \ , \tag{18.14}$$

were $Y\left(e^{j\Omega}\right)$, $H\left(e^{j\Omega}\right)$ and $S\left(e^{j\Omega}\right)$ are the discrete-time Fourier transforms of the complete sequences $y(n)$, $h(n)$, and $s(n)$, respectively. However, common feature extraction schemes as described in Sec. 18.3 use short-time spectral analysis, like the DFT, performing the transform on short windows of the time-domain signal. If these time windows are shorter than the sequences to be convolved, the time-domain linear convolution cannot be expressed as a multiplication in the frequency domain anymore. The overlap-save or overlap-add methods [53] can be used to perform the linear convolution in the short-time spectral domain, if the DFT length is larger than the length of the impulse response.

In most environments, the length of the impulse response (200-800 ms in offices or living-rooms) is significantly longer than the DFT length used for feature extraction (typically 10-40 ms). In this case, partitioned convolution methods can be used. These methods were first introduced in [67] and are successfully used for the implementation of long adaptive filters (e. g. [63, 64]) and the efficient convolution of very long sequences [71]. We use the partitioned overlap-save method to describe the effect of reverberation on speech features.

For feature calculation, the reverberant speech signal $y(n)$ is split into overlapping frames which are weighted with a suitable window function $w(n)$. By

calculating an F-point DFT, the short-time frequency-domain representation

$$Y(f,k) = \sum_{n=0}^{F-1} w(n)\, y(kN+n)\, e^{-j\frac{2\pi}{F}\, n\, f} \tag{18.15}$$

of the reverberant speech signal is obtained. Note that for the following analysis, the frame shift N between neighboring frames needs to fulfill $N \leq F/2$. The RIR $h(n)$ is partitioned into M non-overlapping partitions of length N. These partitions are zero-padded to length F so that an F-point DFT yields the short-time frequency-domain representation

$$H(f,k) = \sum_{n=0}^{F-1} w_h(n)\, h(kN+n)\, e^{-j\frac{2\pi}{F}\, n\, f} \tag{18.16}$$

of the RIR, where $w_h(n) = 1\ \forall\ 0 \leq n < N;\ w_h(n) = 0\ \forall\ N \leq n < F$ is the window function used for the impulse response. Using the short-time frequency-domain representation $S(f,k)$ of the clean speech signal $s(n)$ (see Eq. 18.7), we obtain

$$Y(f,k) = \sum_{m=0}^{M-1} \text{constraint}\big\{H(f,m) \cdot S(f,k-m)\big\}\ , \tag{18.17}$$

where the constraint-operation removes the time-aliasing effects due to the circular convolution performed by the multiplication of two DFT sequences (see e. g. [53]). A common constraint operation foresees an inverse DFT, setting the first $F - N$ points in the time domain to zero, and performing a DFT [65].

If the constraint operation is neglected, the relationship between $S(f,k)$ and $Y(f,k)$ is given as

$$Y(f,k) \approx \sum_{m=0}^{M-1} H(f,m) \cdot S(f,k-m)\ . \tag{18.18}$$

Applying the mel filter bank to the magnitude square of $Y(f,k)$, we obtain the melspec representation

$$y_{\text{m}}(l,k) = \sum_{f=0}^{F/2} C(l,f)\big|Y(f,k)\big|^2 \tag{18.19}$$

$$\approx \sum_{f=0}^{F/2} C(l,f)\left|\sum_{m=0}^{M-1} H(f,m) \cdot S(f,k-m)\right|^2 \tag{18.20}$$

of the reverberant microphone signal. A simpler approximation is obtained if we exchange the order of the mel-filtering operation and the convolution

$$y_{\mathrm{m}}(l,k) \approx \sum_{m=0}^{M-1} \left(\sum_{f=0}^{F/2} C(l,f) \left| H(f,m) \right|^2 \right) \cdot \left(\sum_{f=0}^{F/2} C(l,f) \left| S(f,k-m) \right|^2 \right) \quad (18.21)$$

$$= \sum_{m=0}^{M-1} h_{\mathrm{m}}(l,m) \cdot s_{\mathrm{m}}(l,k-m) \; . \quad (18.22)$$

Note that the squared magnitude of the sum in Eq. 18.20 is replaced by the sum of squared magnitudes in Eq. 18.21. In vector notation, Eq. 18.22 reads

$$\boldsymbol{y}_{\mathrm{m}}(k) \approx \sum_{m=0}^{M-1} \boldsymbol{h}_{\mathrm{m}}(m) \odot \boldsymbol{s}_{\mathrm{m}}(k-m) \; , \quad (18.23)$$

where $\odot$ denotes element-wise multiplication. The *melspec convolution* (as described in Eq. 18.23) will be used throughout this contribution to describe the relationship between the clean feature sequence $\boldsymbol{s}(k)$ and the reverberant feature sequence $\boldsymbol{y}(k)$. The approximations included in Eq. 18.23 compared to the exact relationship according to Eq. 18.17 can be summarized as follows:

- The constraint which had to be applied to realize an exact linear convolution by the overlap-save method [53] is neglected.
- Due to the squared magnitude operation in the feature extraction, the phase is ignored.
- Because of the mel-filtering, the frequency resolution is reduced.
- Since the order of convolution and feature extraction is reversed, the squared magnitude of a sum is replaced by a sum of squared magnitudes.

Fig. 18.12 shows that Eq. 18.23 is nevertheless a good approximation of Eq. 18.17. The figure compares three different melspec feature sequences corresponding to the utterance "four, two, seven". The clean sequence (subfigure a)), exhibits a short period of silence before the plosive /t/ in "two" (around frame 52) and a region of low energy for the lower frequencies at the fricative /s/ in "seven" (around frame 78). These are filled with energy from the preceding frames in the reverberant case (subfigure b)). This illustrates that the reverberation has a dispersive effect on the feature sequences: the features are smeared along the time axis so that the current feature vector depends strongly on the previous feature vectors. We believe that this contradiction to the conditional independence assumption of HMMs (compare Sec. 18.3), namely that the current feature vector depends only on the current state, implies a major performance limitation of HMM-based recognizers in reverberant environments.

Comparing the true reverberant feature sequence in subfigure b) and the approximated reverberant feature sequence according to Eq. 18.23 in subfigure c) reveals that the approximation does not capture the exact texture of the time-frequency pattern (time-mel-channel pattern) of the original sequence. However, the envelope of the time-frequency pattern is very well approximated.

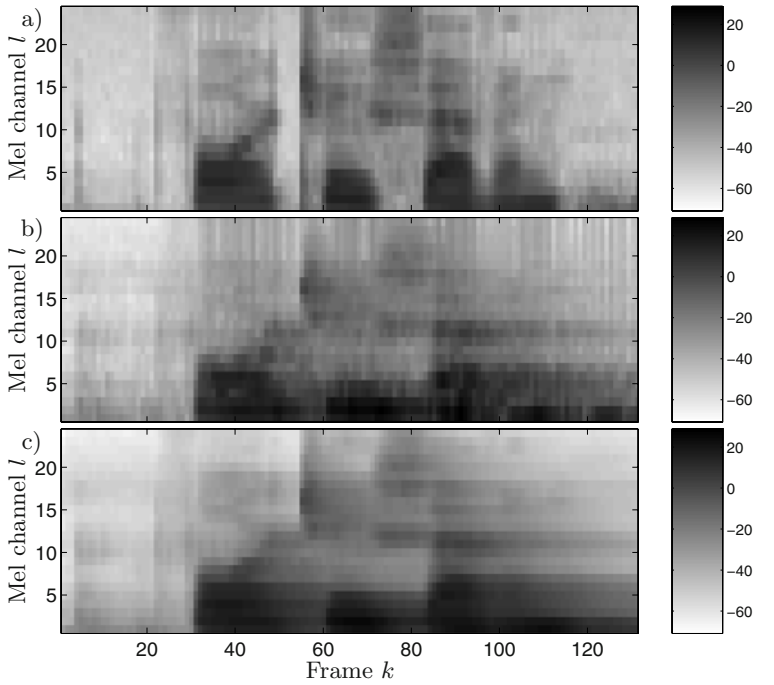

Fig. 18.12. Melspec feature sequences of the utterance "four, two, seven" in dB gray scale a) clean utterance, recorded by a close-talking microphone, b) reverberant utterance, recorded by a microphone four meters away from the speaker, c) approximation of the reverberant utterance by melspec convolution.

Fig. 18.13 b) illustrates the melspec representation of the RIR (frame shift 10 ms) for a very short RIR with a length of only 100 ms and the relationship to its time-domain representation. This picture underlines that even a short RIR extends over several frames in the feature domain. Therefore, the effect of reverberation cannot be modeled by a simple multiplication or addition in the feature domain. A much more accurate approximation is obtained by the melspec convolution of Eq. 18.23.

18.5 Signal Dereverberation and Beamforming

Robust distant-talking ASR can be achieved by dereverberating the speech signal before the feature vectors are calculated. For dereverberation, the convolution of the clean speech signal with the RIR has to be undone by inverse filtering. Since RIRs are in general non minimum-phase, an exact causal inverse filter is not stable [48]. Therefore, only approximations of inverse filters can be determined. As many zeros of the RIR are located close to the unit circle, the inverse of the RIR is usually even longer than the RIR itself so that

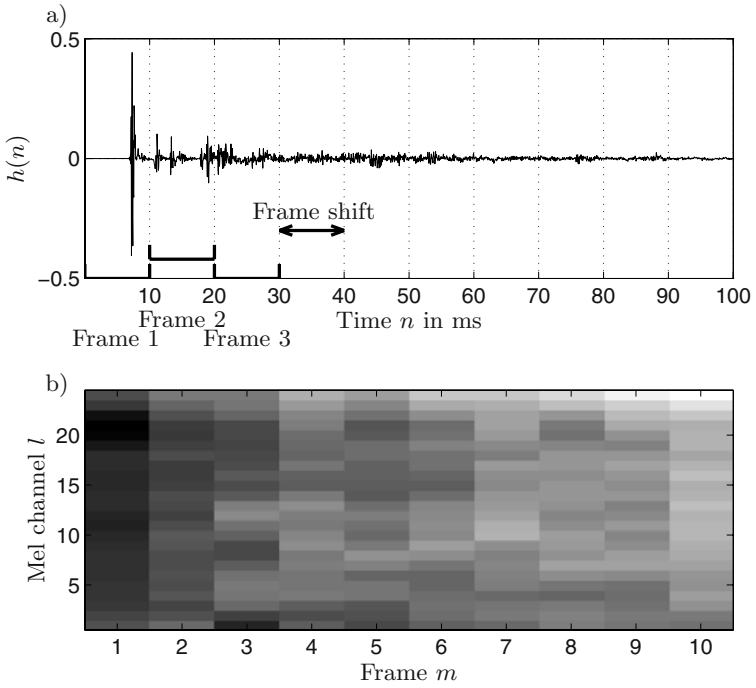

Fig. 18.13. RIR a) $h(n)$ in the time domain b) $\boldsymbol{h}_{\mathrm{m}}(k)$ in the melspec domain using dB gray scale.

an extremely large number of coefficients is necessary to model the inverse by an FIR filter.

Miyoshi and Kaneda show in [42] that multi-channel recordings allow for an exact realization of the inverse filter if the RIRs of all channels are known and do not exhibit common zeros (multiple input/output inverse theorem, MINT). The inverse filters are obtained by inverting the multi-channel convolution matrix which describes the single-input multiple-output system between speaker and microphones. The lengths of the resulting inverse filters are smaller than those of the RIRs. If the RIRs can only be estimated, small deviations from the true RIRs lead to large deviations from the optimum solution [54] so that, in practice, it is still very difficult to implement robust dereverberation algorithms based on MINT.

In [14], Furuya et al. suggest to use the inverse of an estimated correlation matrix of the reverberant speech signal for the calculation of the inverse filters. This approach is equivalent to MINT if

a) it is known which of the microphones is closest to the speaker,
b) the estimation of the correlation matrix is sufficiently accurate, and
c) the source signal is white.

Therefore, whitening filters are applied to the microphone signals to remove the correlation introduced by the speech production from the correlation matrix. A recursive time-averaging is suggested in [15] for the estimation of the correlation matrix in order to track changes of the RIRs between speaker and microphones.

Eigenvector-based multi-channel blind system identification [6, 22] to estimate the RIRs and subsequent inversion based on the MINT theorem is used in [29]. A major problem of this approach is that the order of the RIRs is usually not known. Therefore, an appropriate size of the correlation matrix can hardly be determined, and the accuracy of the blind system identification is significantly reduced. Hikichi et al. suggest in [29] to overestimate the lengths of the RIRs and to employ a post-processing scheme to compensate for the common part which is introduced into the RIRs because of the overestimation.

An alternative approach to estimating the RIRs by blind system identification and subsequent inversion by MINT is to estimate the inverse filters directly. In [10], a versatile framework for multi-channel blind signal processing is proposed, which can be used for blind dereverberation. In a second-order version of the approach, multi-channel filters are adapted to obtain a desired correlation matrix, where the entries along the main diagonal and close to the main diagonal are unchanged while all other elements are minimized. In this way, the clean speech signal is hardly distorted, since the correlation caused by the vocal tract is concentrated around the main diagonal. In contrast, the correlation due to room reverberation extends across the entire autocorrelation matrix. In this way, partial dereverberation can be achieved.

Single-channel approximate dereverberation of the microphone signal can be accomplished by modifying the linear prediction residual. Yegnanarayana et al. show in [75] that the residual of clean speech exhibits one distinct phonation impulse per pitch period in voiced segments, while the residual of reverberant speech exhibits many impulses. By attenuation of the impulses due to reflections compared to the phonation impulse, a dereverberation effect is achieved. In [75], a weighting factor for the residual based on the entropy and the short-time energy contour is suggested. In [76], the approach is extended to multi-channel recordings by coherent summation over the residuals of the individual microphone channels. Further approaches aiming at speech dereverberation by enhancement of the prediction residual are described in [17,18,20].

Nakatani et al. propose to use the short-term harmonicity of voiced speech segments for dereverberation (Harmonicity-based dEReverBeration, HERB) [45]. Based on an estimate of the pitch period, the harmonic part of speech is extracted by adaptive filtering and used as initial estimate of the dereverberated speech signal. Averaging the quotient of the Fourier transforms of the harmonic part and the reverberant part, respectively, over numerous training utterances, a dereverberation filter is determined which reduces reverberation both in voiced and unvoiced speech segments.

An implementation of HERB for the dereverberation of single-word utterances achieves a significant reduction of reverberation [47] as indicated by the

resulting reverberation curves. Using HERB as preprocessing for a speaker-dependent isolated word recognition system which employs HMMs trained on clean speech, a decisive increase in word accuracy is achieved. However, the recognition rate is still significantly lower than the clean-speech performance because of changes in the spectral shapes of the dereverberated signals [47]. Using HMMs trained on utterances dereverberated by HERB to recognize dereverberated speech, recognition rates which are very close to the clean-speech performance are achieved even for strongly reverberant speech signals ($T_{60} = 1\,\mathrm{s}$).

The main problems for the implementation of the approach are the large DFT length required (10.9 seconds in [47]) and the averaging operation necessary for the calculation of the inverse filter. However, the number of utterances needed for the averaging operation has been decreased considerably by several improvements of the approach [36, 46].

Beamforming methods, which use microphone arrays to achieve spatial selectivity, are also potential candidates for signal dereverberation. Steering the main lobe of the beamformer towards the direct sound of the desired source and attenuating reflections arriving from different directions, a dereverberating effect can be achieved. However, several aspects limit the dereverberation capability of beamformers.

By compensating for the delays due to different sound propagation times, the delay-and-sum-beamformer achieves a coherent addition of the signals arriving from the desired direction and an incoherent addition of the signals arriving from other directions. In this way, a relative attenuation of the undesired signals in relation to the desired signals is achieved. The delay-and-sum-beamformer is very robust, but due to the limited spatial selectivity of the apertures of typical microphone arrays, only limited dereverberation can be achieved. Nevertheless, slight improvements of the recognition rates are reported in [50] when delay-and-sum-beamformers are used as preprocessing unit for ASR systems.

Adaptive beamformers [73] are established as powerful approaches for attenuating distortions which are uncorrelated to the desired signal. An adaptive filter is used for each sensor signal and is adapted according to some optimization criterion. For example, the variance of the output signal can be minimized subject to the constraint that the signal arriving from the desired direction passes the filter undistorted (minimum variance distortionless response (MVDR) beamformer). For implementing adaptive beamformers, the structure of the *generalized sidelobe canceler* (GSC) [21] has turned out to be very advantageous. To achieve a robust GSC implementation for broad-band speech signals, restrictions of the filter coefficients have to be enforced [31]. The performance of the GSC for speech signals can be further improved by controlling the adaptation in individual DFT bins instead of using a single broad-band control [26, 27].

A remaining problem for the use of adaptive beamformers in signal dereverberation is the correlation between the desired signal and its reverberation.

Therefore, it is very difficult to completely avoid cancellation of the desired signal and the gain of adaptive beamformers compared to a fixed delay-and-sum-beamformer is reduced.

In [61], Seltzer proposes to integrate beamforming and speech recognition into one unit. The coefficients of a filter-and-sum-beamformer are adapted in order to maximize the probability of the correct transcription, which is estimated by an initial recognition iteration (unsupervised version) or is known for an initial training utterance (supervised version). The probability is determined based on the HMMs of the speech recognizer.

Using the speech features and the acoustic model of the ASR system for the adaptation of the filter coefficients ensures that those speech properties are emphasized which are crucial for recognition. In [61], a noticeable increase of the recognition performance compared to using only a single microphone or using a delay-and-sum-beamformer is reported for both additive noise and moderate reverberation. For strong reverberation, a subband version of the approach [62] is more suitable. The performance of the supervised version is limited if the RIRs change significantly between calibration and test. The performance of the unsupervised version is limited by the accuracy of the initial transcription estimate. In strongly reverberant environments, this initial estimate can be very inaccurate so that, then, hardly any gain can be achieved with the approach.

The adaptation of the filter coefficients is very challenging. Because of the nonlinear relationship between the filter coefficients and the cost function, in general, the error surface exhibits local minima so that the convergence to a satisfying solution is not assured. The reduced data rate of the speech features compared to the speech signal samples (see Sec. 18.3) implies that a large number of filter coefficients has to be adapted with only little training data. Thus, for a given duration of the utterance used for adaptation, the number of adjustable filter coefficients is limited or the optimum coefficients cannot be identified.

18.6 Robust Features

A simple way to alleviate the limitations of the conditional independence assumption (see Sec. 18.3) is to extend the speech feature vector by so-called *dynamic features* like Δ and $\Delta\Delta$ coefficients [13, 25]. These features can be considered as the first (Δ) and second ($\Delta\Delta$) derivative of the static features and are usually approximated by simple differences or by linear regression calculations. In this way, the dynamic features capture the temporal changes in the spectra across several frames (2 to 10 frames) and thus enlarge the temporal coverage of each feature vector. Nevertheless, for strongly reverberant environments, the limited reach of the Δ and $\Delta\Delta$ features is not sufficient to cover the dispersive character of the reverberation so that the recognition performance is still limited.

RASTA (RelAtiveSpecTrA)-based speech features [28] are largely insensitive to a convolution with a short time-invariant impulse response. The key steps in calculating RASTA-based features are the following: The speech signal is divided into sub-bands (e. g., similar to the critical bands) and a nonlinear compressing transform is performed on each sub-band signal. Each compressed sub-band signal is then filtered by a bandpass filter (passband from 0.26 Hz to 12.8 Hz) which removes the very low and high modulation frequencies.

If a logarithmic transform is used, the convolution in the signal domain, which approximately corresponds to a multiplication in the sub-band domain, is transformed into an addition in the compressed sub-band domain so that the impulse response is represented by an additive constant in each sub-band, which is removed by the respective bandpass filters. Therefore, a convolution of the time-domain signal with a short time-invariant impulse response has hardly any influence on RASTA-based speech features.

In virtually all reverberant environments, the RIR is significantly longer than the frames used for feature calculation. Therefore, the time-domain convolution (Eq. 18.2) cannot be represented by a multiplication in the sub-band domain, but rather by a convolution in each sub-band as discussed in Sec. 18.4. Consequently, the time-domain convolution cannot be represented by additive constants in the compressed sub-band domain and will not be removed by the bandpass filters. Therefore, the RASTA-based features are not insensitive to long reverberation.

Cepstral mean subtraction (CMS) (see, e. g., [3] and [33], Sec. 10.6.4) is another way of alleviating convolutional distortions. A convolution with an impulse response in the time domain is transformed into an addition of the cepstral representation of the impulse response in the cepstral domain, if the frame length of the analysis window is long compared to the length of the impulse response. Thus, convolutive effects characterized by a short impulse response result in an addition of the cepstral representation of the impulse response. This representation of the impulse response can be estimated by calculation of the linear mean across the utterance and can be removed by subtraction. If the utterances are long enough so that the cepstral representation of the impulse response can be estimated reliably, the robustness of the recognizer to short convolutional effects can be significantly increased. For long reverberation with a typical duration from 200 to 800 ms in offices and living-rooms compared to the cepstral analysis window length of typically 10 to 40 ms, CMS yields only limited gains.

18.7 Model Training and Adaptation

ASR systems perform best if the acoustical conditions of the environment where the training data have been recorded match the acoustical conditions of the environment where the recognizer is applied. Therefore, using training data recorded in the application environment results in models which are well

suited for the application environment. However, recording a complete set of training data for each application environment requires tremendous effort and is therefore unattractive for most real-world applications.

Giuliani et al. [19] generate reverberant training data by convolving clean-speech training utterances with RIRs measured in the application environment. In this way, the data collection effort is considerably reduced. In [66], Stahl et al. show that the performance of HMMs trained on artificially reverberated training data is significantly degraded relative to that of HMMs trained on data recorded in the application environment. On the other hand, the recognition performance based on artificial reverberation is significantly improved compared to models trained on clean speech. However, training the recognizer for each application environment still implies a huge computational load and is quite inflexible.

Therefore, Haderlein et al. [23] use RIRs recorded at different loudspeaker and microphone positions in the application environment to generate artificially reverberated data which allow the training of HMMs suitable for different speaker and microphone positions in the application environment. These HMMs show good performance also in different rooms with similar reverberation characteristics.

To reduce the effort required by a complete training with reverberant data, well-trained clean-speech models can be used as starting point for *model adaptation*. Using only a few utterances recorded in the target environment, the clean-speech models are adapted to the acoustic conditions in the application environment. Numerous adaptation schemes have been proposed for adaptation to additive distortions (e. g. background noise) and channel effects characterized by impulse responses shorter than the frame length of the feature extraction analysis window (e. g. to compensate for different frequency responses of the microphones used for training and test): Maximum a posteriori estimation [38, 39], parallel model combination [16], vector Taylor series (VTS) [43], and HMM composition [44, 68, 69].

These approaches rely on the assumption that the observed features result from the addition of the clean features and a noise term or a channel distortion term, respectively. In the case of room reverberation, the relation between clean-speech features and the feature-domain representation of the RIR is not additive as shown in Sec. 18.4. Therefore, the above-mentioned model adaptation approaches are not appropriate for reverberant environments.

In [56] and [30], two model adaptation techniques tailored to long reverberation are proposed. Based on information of the reverberation characteristics of the application environment, the means of the output density of the current state are adapted taking into account the means of the previous states. This adaptation is performed for all states in the HMM network before recognition. Therefore, the average time of remaining in each state has to be considered. In this way, a performance approaching that of reverberant training is achieved for isolated digit recognition in [56] and for connected digit recognition in [30].

However, both the model adaptation approaches [30, 56] and the training with reverberant data [19, 23, 66] suffer from the conditional independence assumption which limits the capability of the HMMs to accurately model reverberant feature vector sequences.

In [70] a frame-by-frame adaptation method is suggested which overcomes the limitation of the conditional independence assumption. The reverberation of the previous feature vectors is modeled by a first-order linear prediction and is added to the means of the clean-speech HMM at decoding time. This implies an approximation of the reverberation by a strictly exponentially decaying function and achieves slightly lower recognition rates than matched reverberant training [70].

18.8 Reverberation Modeling for Speech Recognition

A novel concept for robust distant-talking ASR in reverberant environments, called **RE**verberation **MO**deling for **S**peech recognition (REMOS), is discussed in this section. The concept was first introduced in [57] and has been extended in [58–60]. The acoustic model of a REMOS-based recognizer is a combination of a clean-speech HMM network $\mathcal{N}_\lambda$ and a statistical *reverberation model* η as depicted in Fig. 18.14. This combined acoustic model allows for very accurate and flexible modeling of reverberant feature sequences without the limitations of the conditional independence assumption.

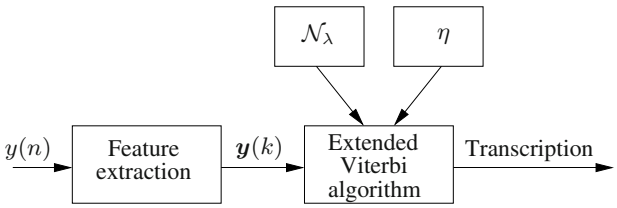

Fig. 18.14. Block diagram of the REMOS concept.

During the recognition process, the improved acoustic model is used to estimate the most likely clean-speech feature sequence directly in the feature domain. This kind of dereverberation follows the idea of preprocessing. Performing the dereverberation directly in the feature domain makes the approach less sensitive to variations of the spectro-temporal details of the acoustic path between speaker and microphone and allows for a more efficient implementation. The calculation of the acoustic score is based on this clean-speech feature estimate. In this way, the REMOS concept combines advantages of all three previously described classes of robust approaches: signal preprocessing, feature compensation and improved acoustical modeling (see Fig. 18.4).

We introduce the REMOS concept from the perspective of feature production. In particular, we show how the combination of the clean-speech HMM network and the statistical reverberation model describes the reverberant feature sequence. For the actual speech recognition, however, the combined model will be employed to find the most likely transcription for a given reverberant input feature sequence. Before deriving a solution for the decoding of the combined model, a detailed description of the reverberation model and its training is given.

18.8.1 Feature Production Model

The idea of modeling reverberation directly in the feature domain is based on the following observation: While the spectro-temporal details of the acoustic path between speaker and microphone are very sensitive to changes like small movements of the speaker, the spectro-temporal envelopes are hardly affected by such changes (see also Sec. 18.4). As the speech features used for ASR only capture the envelopes, a good feature-domain model for describing reverberation in a certain room can be obtained without detailed information on speaker and microphone positions.

We assume that the sequence of reverberant speech feature vectors $\boldsymbol{y}(k)$ is produced by a combination of a network $\mathcal{N}_\lambda$ of word-level HMMs λ describing the clean-speech and a reverberation model η as illustrated in Fig. 18.15. The word-level HMMs λ may be composed of subword HMMs. The task grammar and the language model can be embedded into the network of HMMs to represent the actual recognition task.

The reverberation model is completely independent of the recognition task and describes the reverberation of the room where the recognizer will be used. The strict separation of the task information incorporated into the network of HMMs and the information about the acoustic environment reflected by the reverberation model yields a high degree of flexibility when the recognition system has to be adapted to new tasks or new acoustic environments.

The REMOS concept can be applied to any kind of speech features which allow the formulation of an appropriate relation between the sequence $\boldsymbol{s}(k)$ of output feature vectors of the HMM network, the sequence $\boldsymbol{H}(k)$ of the reverberation model output matrices (see Sec. 18.8.2) and the sequence $\boldsymbol{y}(k)$ of reverberant speech feature vectors.

Based on the melspec convolution described in Eq. 18.23, the feature-dependent *combination operator* in Fig. 18.15 is given in generic form and then for melspec features $\boldsymbol{y}_\mathrm{m}(k)$, logmelspec features $\boldsymbol{y}_\mathrm{l}(k)$, and MFCC features $\boldsymbol{y}_\mathrm{c}(k)$ in the following:

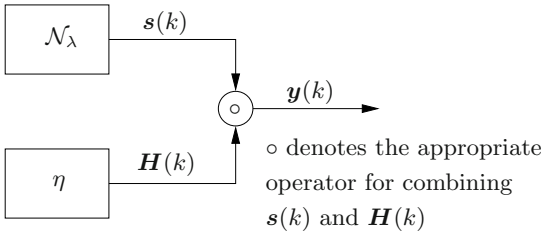

Fig. 18.15. Feature production model of the REMOS concept.

$$\boldsymbol{y}(k) = \boldsymbol{H}(k) \circ \boldsymbol{s}(k) \quad \forall\, k = 1 \ldots K + M - 1 \,, \tag{18.24}$$

$$\boldsymbol{y}_{\mathrm{m}}(k) = \sum_{m=0}^{M-1} \boldsymbol{h}_{\mathrm{m}}(m,k) \odot \boldsymbol{s}_{\mathrm{m}}(k-m) \,, \tag{18.25}$$

$$\boldsymbol{y}_{\mathrm{l}}(k) = \log\left(\sum_{m=0}^{M-1} \exp\left(\boldsymbol{h}_{\mathrm{l}}(m,k)\right) \odot \exp\left(\boldsymbol{s}_{\mathrm{l}}(k-m)\right) \right) \,, \tag{18.26}$$

$$\boldsymbol{y}_{\mathrm{c}}(k) = \boldsymbol{B}\,\boldsymbol{A} \cdot \log\left(\sum_{m=0}^{M-1} \exp\left(\boldsymbol{A}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{h}_{\mathrm{c}}(m,k)\right) \odot \exp\left(\boldsymbol{A}^{-1}\boldsymbol{B}^{\mathrm{T}}\boldsymbol{s}_{\mathrm{c}}(k-m)\right) \right) . \tag{18.27}$$

Here, $\odot$ denotes element-wise multiplication, the vector $\boldsymbol{h}(m,k)$ is a realization of the reverberation model for frame delay m and frame k, while M and K are the lengths of the reverberation model and the clean utterance, respectively. The matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ were introduced in Eq. 18.10. The logarithm and the exponential function are applied element-wise. The dependency of $\boldsymbol{h}(m,k)$ on the current frame k results from fact that the reverberation model allows the feature-domain representation of the RIR to change each frame (see Sec. 18.8.2). Note that all combination operators (Eqs. 18.25, 18.26, and 18.27) are equivalent, since they use the same model of reverberation, namely the feature vector convolution in the melspec domain (Eq. 18.23).

The reverberant features sequence $\boldsymbol{y}(k)$ can be considered as a realization of the vector-valued random process $\mathcal{Y}(k)$. The combined acoustic model according to Fig. 18.15 describes $\mathcal{Y}(k)$ as a non-stationary random process with statistical dependencies between neighboring frames characterized by the reverberation model η.

18.8.2 Reverberation Model

The reverberation model represents the acoustic path between speaker and microphone in the feature domain. As the acoustic path can be modeled sufficiently well by an RIR, the reverberation model basically represents the RIR

in the feature domain. As shown in Fig. 18.13 b), the feature-domain representation of the RIR can be considered as a matrix, where each column corresponds to a certain frame and each row corresponds to a certain mel channel. Each matrix element in Fig. 18.13 b) has a fixed value as illustrated by the gray level in the image.

Since the exact RIR is usually not known and since the combination operation provides only an approximation of the exact relationship between the clean and the reverberant feature sequences (see Sec. 18.4), we do not use a fixed feature-domain representation of a single RIR as the reverberation model. Instead, we use a statistical model where each matrix element is modeled by an independent identically distributed (IID) random process. For simplification, each element of the matrix is assumed to be statistically independent from all other elements and is modeled by a shift-invariant Gaussian density. Therefore, the reverberation model is completely described by the matrices of the means and variances of the Gaussian distributions. Fig. 18.16 illustrates the reverberation model.

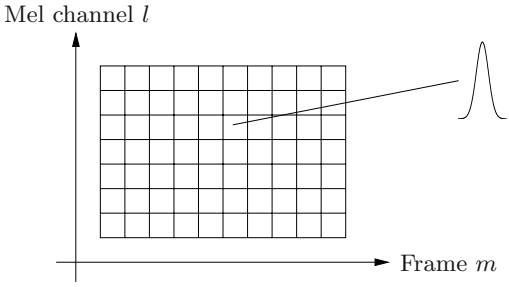

Fig. 18.16. Reverberation model η.

In summary, the reverberation model describes an IID matrix-valued Gaussian random process $\mathcal{H}(k)$. The sequence of the feature-domain RIR representations $\boldsymbol{H}(k)$ is a realization of this random process as illustrated in Fig. 18.17. The IID property of the random process implies that all elements of the random process at frame k_1 are statistically independent from all elements of the random process at frame k_2 as long as $k_1 \neq k_2$. Because the random process $\mathcal{H}(k)$ is strict-sense stationary, its probability density is not time-dependent and is denoted as $f_{\mathcal{H}(k)}(\boldsymbol{H}(k)) = f_{\eta}(\boldsymbol{H}(k))$.

18.8.3 Training of the Reverberation Model

The training of the reverberation model is based on a number of measured or hypothesized feature-domain RIR representations $\hat{\boldsymbol{H}}(k)$. Using these RIR representations, the mean matrix μ_η and the variance matrix σ_η^2 of the reverberation model η are estimated

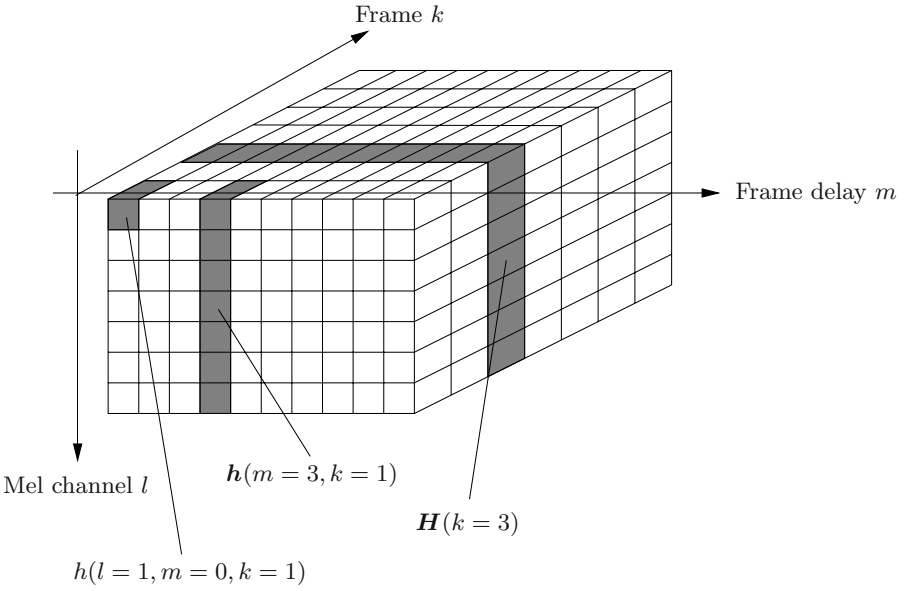

Fig. 18.17. Sequence of feature-domain RIR representations $\boldsymbol{H}(k)$ as a realization of the random process $\mathcal{H}(k)$ described by the reverberation model η.

$$\mu_\eta = \frac{1}{P} \sum_{k=1}^{P} \hat{\boldsymbol{H}}(k) \,, \tag{18.28}$$

$$\sigma_\eta^2 = \frac{1}{P-1} \sum_{k=1}^{P} \left(\hat{\boldsymbol{H}}(k) - \mu_\eta \right)^2 \,, \tag{18.29}$$

where P is the number of RIR representations $\hat{\boldsymbol{H}}(k)$.

There are two ways of obtaining a set of RIR representations $\hat{\boldsymbol{H}}(k)$: Either time-domain RIRs are transformed to the feature domain, or $\hat{\boldsymbol{H}}(k)$ is estimated directly in the feature domain.

Training of the Reverberation Model using Time-Domain RIRs

A set of time-domain RIRs for different microphone and loudspeaker positions of the room where the ASR system will be applied can be used for calculating a set of realizations $\hat{\boldsymbol{H}}(k)$. These RIRs can either be measured before using the recognizer, estimated by blind system identification approaches or modeled, e. g., using the image method as described in [1]. To train the reverberation model, the RIRs are time-aligned so that the direct path of all RIRs appears at the same delay. Calculation of the features yields a set of realizations $\hat{\boldsymbol{H}}(k)$ which are used to estimate the means and the variances of the reverberation model according to Eq. 18.28 and Eq. 18.29. A block diagram of the training based on time-domain RIRs is given in Fig. 18.18.

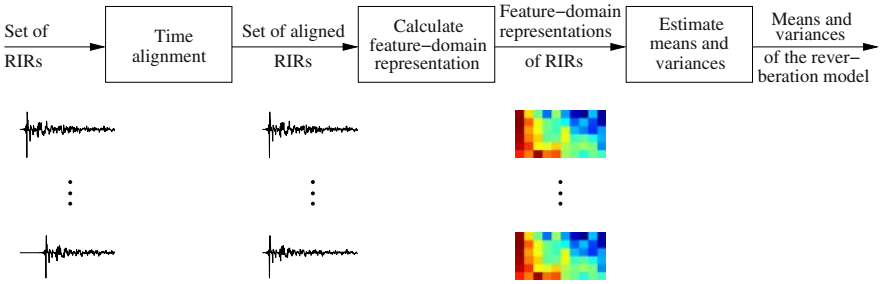

Fig. 18.18. Training of the reverberation model using time-domain RIRs.

Estimation in the Feature Domain

The realizations $\hat{\boldsymbol{H}}(k)$ can also be obtained directly in the feature domain. For example, maximum likelihood (ML) estimation based on a few training utterances with known transcription as depicted in Fig. 18.19 can be employed. Using the reverberant feature sequence $\boldsymbol{y}(k)$, a set of clean-speech HMMs, and the correct transcription of the training utterance corresponding to $\boldsymbol{y}(k)$, the optimum state sequence through the HMM representing the correct transcription is obtained by forced alignment [78]. Using this state sequence and the clean-speech HMMs, a joint density of the clean-speech feature sequence $f_{\mathcal{S}}(\boldsymbol{S})$ is estimated.

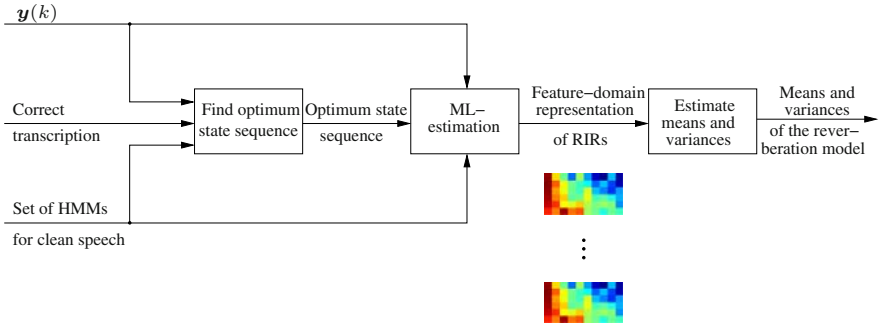

Fig. 18.19. Block diagram for the feature-domain training of the reverberation model based on maximum likelihood estimation.

To obtain the corresponding conditional Gaussian density $f_{\mathcal{Y}|\boldsymbol{H}(k)}(\boldsymbol{Y})$ of the reverberant feature sequence given $\boldsymbol{H}(k)$, the means $\mu_{\mathcal{S}(k)}$ are combined with $\boldsymbol{H}(k)$ to get the means $\mu_{\mathcal{Y}(k)|\boldsymbol{H}(k)}$

$$\mu_{\mathcal{Y}(k)|\boldsymbol{H}(k)} = \boldsymbol{H}(k) \circ \mu_{\mathcal{S}(k)} \ .$$

For simplification, the variances $\sigma^2_{\mathcal{Y}(k)|\boldsymbol{H}(k)}$ are assumed to be equal to the clean-speech variances $\sigma^2_{\mathcal{S}(k)}$ as suggested in [56]. The ML estimate $\hat{\boldsymbol{H}}_{\mathrm{ML}}(k)$ is obtained by maximizing the conditional density of the reverberant feature sequence with respect to $\boldsymbol{H}(k)$

$$\hat{\boldsymbol{H}}_{\mathrm{ML}}(k) = \underset{\boldsymbol{H}(k)}{\operatorname{argmax}} \left\{ f_{\mathcal{Y}|\boldsymbol{H}(k)}(\boldsymbol{Y}) \right\} .$$

A more detailed description of this approach including the derivation of the ML estimate in the melspec domain and corresponding experimental results can be found in [60].

18.8.4 Decoding

So far, we introduced the REMOS concept from the perspective of feature production, describing how reverberant speech features are generated given the model. For speech recognition, however, the opposite task has to be solved. Given a reverberant utterance, a recognition network of clean-speech HMMs and a reverberation model, the task of the recognizer is to find the path through the network yielding the highest probability for the given feature sequence in connection with the reverberation model.

Independently of the acoustic-phonetic modeling, the distant-talking speech recognition search problem has been formulated in Sec. 18.3 as finding the word sequence $\hat{W}$ maximizing the product of the acoustic score $A(\boldsymbol{Y}|W)$ of $\boldsymbol{Y}$ given the word sequence W and the language score $L(W)$

$$\hat{W} = \underset{W}{\operatorname{argmax}} \left\{ A(\boldsymbol{Y}|W) \cdot L(W) \right\} . \tag{18.30}$$

For conventional HMMs, the acoustic score based on the most likely state sequence is given as (see Sec. 18.3)

$$A(\boldsymbol{Y}|W) = \max_{Q} \left\{ P(\boldsymbol{Y}, Q|\Lambda) \right\} .$$

For the combined acoustic model consisting of a clean-speech HMM network and the reverberation model according to Fig. 18.15, the acoustic score is given as

$$\begin{aligned} A(\boldsymbol{Y}|W) &= \max_{Q,\boldsymbol{S},\boldsymbol{H}} \left\{ P(Q, \boldsymbol{S}, \boldsymbol{H}|\Lambda, \eta) \right\} \quad \text{subject to Eq. 18.24} \\ &= \max_{Q} \left\{ P(Q|\Lambda) \cdot \max_{\boldsymbol{S},\boldsymbol{H}} \left\{ P(\boldsymbol{S}, \boldsymbol{H}|\Lambda, \eta, Q) \right\} \right\} \\ &\quad \text{subject to Eq. 18.24} . \end{aligned}$$

As only the calculation of the acoustic score is different in the REMOS concept compared to conventional HMMs, the same search algorithms as for conventional HMMs can be used to solve the problem described in Eq. 18.30 by the

REMOS concept if a few extensions are added which account for the modified acoustic score calculations. These extensions will be derived in the following.

In the proposed approach, the acoustic score $A(\boldsymbol{Y}|W)$ can be calculated iteratively by an extended version of the Viterbi algorithm, where we assume that the HMM starts in state 1 and ends in state J.

Initialization:

$$\begin{aligned} \gamma_1(1) &= \max_{\boldsymbol{s}(1),\boldsymbol{h}(0,1)} \left\{ f_\Lambda\big(1, \boldsymbol{s}(1)\big) \cdot f_\eta\big(\boldsymbol{h}(0,1)\big) \right\} , \\ \text{subject to} \quad \boldsymbol{x}(1) &= \boldsymbol{s}(1) \circ \boldsymbol{h}(0,1) \\ \gamma_j(1) &= 0 \quad \forall j = 2 \dots J \, , \\ \psi_j(1) &= 0 \quad \forall j = 1 \dots J \, . \end{aligned}$$

Recursion:

$$\gamma_j(k) = \max_i \left\{ \gamma_i(k-1) \cdot a_{ij} \cdot O_{ij}(k) \right\} , \tag{18.31}$$

$$\psi_j(k) = \operatorname*{argmax}_i \left\{ \gamma_i(k-1) \cdot a_{ij} \cdot O_{ij}(k) \right\} ,$$

$$O_{ij}(k) = \max_{\boldsymbol{s}(k),\boldsymbol{H}(k)} \left\{ f_\Lambda\big(j, \boldsymbol{s}(k)\big) \cdot f_\eta\big(\boldsymbol{H}(k)\big) \right\} , \tag{18.32}$$

$$\text{subject to} \quad \boldsymbol{y}(k) = \boldsymbol{H}(m,k) \circ \boldsymbol{s}(k) \, , \tag{18.33}$$

$$\forall j = 1 \dots J, \quad k = 2 \dots K + M - 1, \, .$$

Termination:

$$A(\boldsymbol{Y}|W) = \gamma_J(K+M-1), \quad q(K+M-1) = J \, .$$

Backtracking:

$$q(k) = \psi_{q(k+1)}(k+1) \quad \forall k = K+M-2, \, \dots, \, 1 \, .$$

As in the conventional Viterbi algorithm, i indexes all considered previous states leading to the current state j, $\gamma_j(k)$ is the Viterbi metric for state j at frame k. $f_\Lambda(j, \boldsymbol{s}(k))$ is the output density of state j of the HMM sequence Λ describing W evaluated for the clean-speech vector $\boldsymbol{s}(k)$. $f_\eta(\boldsymbol{H}(k))$ is the probability density of the reverberation model η evaluated for the feature-domain representation $\boldsymbol{H}(k)$ of the RIR (see Sec. 18.8.2). The backtracking pointer $\psi_j(k)$ refers to the previous state and allows backtracking of the most likely state sequence.

The result $O_{ij}(k)$ of the optimization in Eq. 18.32, which is referred to as *inner optimization*, is obtained by varying the vector of the current clean-speech frame $\boldsymbol{s}(k)$ and the matrix of the current feature-domain RIR representation $\boldsymbol{H}(k)$ in order to maximize the product of their probability densities subject to the constraint described in Eq. 18.33. That is, the combination of $\boldsymbol{H}(k)$ and $\boldsymbol{s}(k)$ needs to be equal to the current reverberant feature vector $\boldsymbol{y}(k)$. The subscript ij in $O_{ij}(k)$ indicates that this term is based on the optimum partial state sequence $\hat{Q}_{ij}(k)$ from frame $k - M + 1$ to frame k with current state j and previous state i (see Fig. 18.21) given by

$$\hat{Q}_{ij}(k) = \hat{q}_{ij}(k-M+1),\ \ldots,\ \hat{q}_{ij}(k-2),\ \ \hat{q}_{ij}(k-1) = i,\ \ \hat{q}_{ij}(k) = j\ .$$

Comparing the update equation 18.13 of the conventional Viterbi algorithm to the update equation 18.31 of the extended Viterbi algorithm, we observe two differences. The first difference is that the output density $f_\Lambda(j, \boldsymbol{s}(k))$ of the current HMM state in (18.13) is replaced by the term $O_{ij}(k)$ in Eq. 18.31. This term can be considered as the output density of the combined model according to Fig. 18.15 and is calculated by solving the inner optimization problem (Eq. 18.32) subject to Eq. 18.33. The second difference is that $O_{ij}(k)$ is included in the maximization over all possible state sequences Q in Eq. 18.31 while $f_\Lambda(j, \boldsymbol{s}(k))$ is not included in the corresponding maximization in Eq. 18.13. Therefore, the inner optimization has to be performed for each frame k, each state j and each possible predecessor state i. The inner optimization is the main extension compared to the conventional Viterbi algorithm and will be discussed in more detail in the following section.

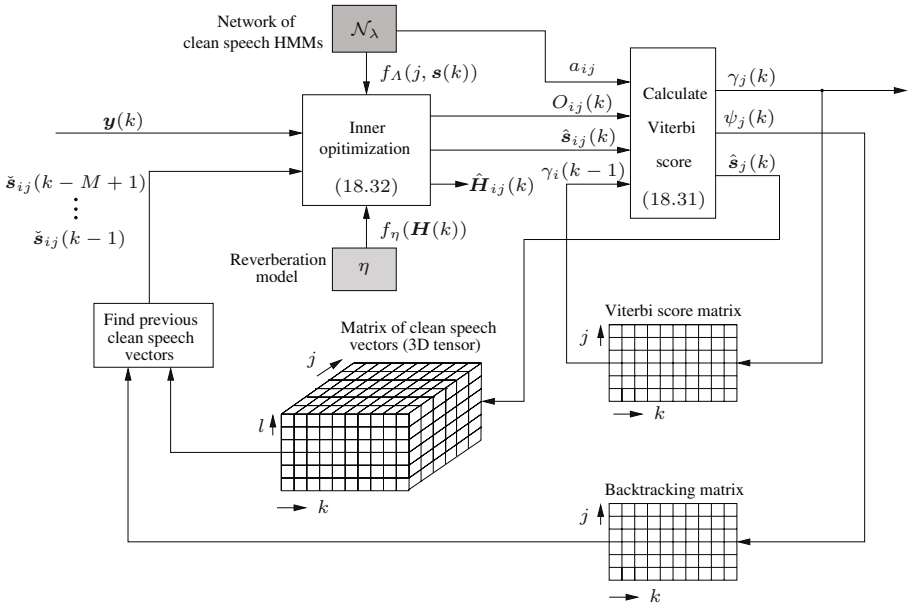

Fig. 18.20. Illustration of the extended Viterbi algorithm.

Fig. 18.20 illustrates the extended Viterbi algorithm. To calculate the current Viterbi score $\gamma_j(k)$, the previous Viterbi score $\gamma_i(k-1)$, the transition probability a_{ij} and the output density $O_{ij}(k)$ of the combined model have to be maximized according to Eq. 18.31. In order to obtain $O_{ij}(k)$, the inner optimization according to Eq. 18.32 has to be solved. Therefore, the optimum contributions $\hat{\boldsymbol{s}}_{ij}(k)$ and $\hat{\boldsymbol{H}}_{ij}(k)$ of the current HMM state and the reverberation model to the current reverberant observation vector $\boldsymbol{y}(k)$ are estimated

by maximizing the product of the HMM output density $f_\Lambda(j, \boldsymbol{s}(k))$ and the reverberation model output density $f_\eta(\boldsymbol{H}(k))$ subject to the constraint that the combination of $\boldsymbol{s}(k)$ and $\boldsymbol{H}(k)$ yields $\boldsymbol{y}(k)$. In this way, $O_{ij}(k)$, $\hat{\boldsymbol{s}}_{ij}(k)$, and $\hat{\boldsymbol{H}}_{ij}(k)$ are obtained.

To solve the inner optimization based on one of the combination operators described in Eqs. 18.25, 18.26, or 18.27, all clean-speech feature vectors $\boldsymbol{s}(k-M+1)\ldots\boldsymbol{s}(k-1)$ are necessary. These true clean-speech feature vectors are replaced by estimates determined in previous iterations of the extended Viterbi algorithm for the frames $k' < k$ and the states j'. The clean-speech feature vector estimates are calculated as follows.

The inner optimization for frame k', state j', and each possible predecessor state i' yields a clean-speech feature estimate $\hat{\boldsymbol{s}}_{i'j'}(k')$ for each i'. By maximizing over i' in the Viterbi recursion (Eq. 18.31), the most likely predecessor state

$$\hat{i}' = \underset{i'}{\operatorname{argmax}} \left\{ \gamma_{i'}(k'-1) \cdot a_{i'j'} \cdot O_{i'j'}(k') \right\} \tag{18.34}$$

is determined. Using, $\hat{i}'$, the most likely clean-speech feature estimate among all estimates $\hat{\boldsymbol{s}}_{i'j'}(k')$ is selected according to

$$\hat{\boldsymbol{s}}_{j'}(k') = \hat{\boldsymbol{s}}_{\hat{i}'j'}(k') \; . \tag{18.35}$$

For each frame k' and each state j', the most likely clean-speech feature estimate $\hat{\boldsymbol{s}}_{j'}(k')$ is stored in a matrix of clean-speech vectors (3D tensor) as depicted in Fig. 18.20.

Since the matrix of clean-speech vectors is filled up to column $k-1$ by the previous iterations, before the recursions for frame k start, the estimated clean-speech vectors can be obtained from this matrix using the optimum partial path $\hat{Q}_{ij}(k)$. The states corresponding to $\hat{Q}_{ij}(k)$ are determined by tracing back the path from frame $k-1$ and state i using the backtracking pointers ψ as follows

$$\hat{q}_{ij}(k) = j \; , \tag{18.36}$$

$$\hat{q}_{ij}(k-1) = i \; , \tag{18.37}$$

$$\hat{q}_{ij}(\kappa) = \psi_{\hat{q}_{ij}(\kappa+1)}(\kappa+1) \quad \forall\, \kappa = k-2,\, \ldots,\, k-M+1 \; . \tag{18.38}$$

Fig. 18.21 illustrates the two optimum partial paths $\hat{Q}_{i_1 j}(k)$ and $\hat{Q}_{i_2 j}(k)$ for frame k, state j and the two possible predecessor states i_1 and i_2 for the HMM topology according to Fig. 18.7.

Now the clean-speech feature estimates corresponding to $\hat{Q}_{ij}(k)$ are determined by selecting the corresponding vectors from the matrix of clean-speech vectors as follows

$$\check{\boldsymbol{s}}_{ij}(\kappa) = \hat{\boldsymbol{s}}_{\hat{q}_{ij}(\kappa)}(\kappa) \quad \forall\, \kappa = k-1,\, \ldots,\, k-M+1 \; . \tag{18.39}$$

Note that the clean-speech estimate $\check{\boldsymbol{s}}_{ij}(\kappa)$ extracted from the matrix of clean-speech vectors is in general different from the initial clean-speech estimate

$\hat{\boldsymbol{s}}_{ij}(\kappa)$ obtained from the inner optimization. Now, all input data required for the inner optimization are available, and Eq. 18.32 can be solved.

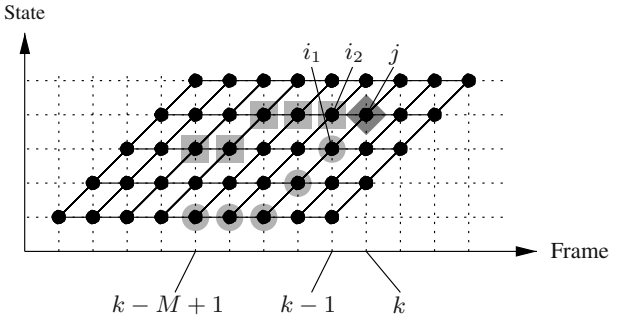

Fig. 18.21. Illustration of two optimum partial paths $\hat{Q}_{i_1 j}(k)$ (indicated by the large dots) and $\hat{Q}_{i_2 j}(k)$ (indicated by the squares) corresponding to the two possible predecessor states i_1 and i_2 in the trellis diagram of the HMM according to Fig. 18.7 for $k = 10$, $j = 4$, $i_1 = 3$, $i_2 = 4$, $M = 6$.

After each iteration, the Viterbi score $\gamma_j(k)$ and the backtracking pointer $\psi_j(k)$ are stored in the corresponding matrices. After all iterations are finished, these two matrices are used to determine the final acoustic score and to find the optimum path through the HMM network which enables the reconstruction of the most likely word sequence W corresponding to the feature sequence $\boldsymbol{Y}$.

Note that the decoding of the combined acoustic model described above exhibits some similarities to the HMM decomposition approach proposed in [72] for additive noise. Indeed, REMOS can be considered as a generalization of the HMM decomposition approach to a convolutive combination of the model outputs if the reverberation model is considered as a one-state HMM with matrix-valued output. However, there is a significant difference in the evaluation of the output density of the combined model. The HMM decomposition approach proposes to integrate over all possible combinations of the outputs of the individual models to calculate the output probability of the combined feature vector. We propose to search for the most likely combination to calculate the probability of the reverberant feature vector. While both approaches are feasible for simple combinations like addition, the method proposed here provides significant computational savings for more complex combinations like convolution.

18.8.5 Inner Optimization

To find the best combination of the HMM output and the reverberation model output, the extended Viterbi algorithm performs an inner optimization in each iteration. In this inner optimization, the joint density of the current HMM

state and the reverberation model has to be maximized subject to the constraint that the combination of $\boldsymbol{s}(k)$ and $\boldsymbol{H}(k)$ yields the current reverberant feature vector $\boldsymbol{y}(k)$ as described by Eq. 18.32 and Eq. 18.33.

Instead of maximizing the objective function

$$f_{\Lambda\eta} = f_{\Lambda}\big(j, \boldsymbol{s}(k)\big) \cdot f_{\eta}\big(\boldsymbol{H}(k)\big)$$

directly, equivalently, the logarithm of the objective function $\log\left(f_{\Lambda\eta}\right)$ can be maximized, since the logarithm is a monotone function. Therefore, the inner optimization problem can be expressed as

$$\tilde{O}_{ij}(k) = \max_{\boldsymbol{s}(k),\boldsymbol{H}(k)} \big\{ \log\{f_{\Lambda\eta}\}\big\} \quad \text{subject to Eq. 18.33}\,. \tag{18.40}$$

The objective function $f_{\Lambda\eta}$ of the inner optimization problem depends on the output density of the current HMM state $f_{\Lambda}(j, \boldsymbol{s}(k))$ and the output density of the reverberation model $f_{\eta}(\boldsymbol{H}(k))$. If single Gaussian densities are used both in the HMM and in the reverberation model, $\log\{f_{\Lambda\eta}\}$ is a quadratic function with a single global maximum. If mixtures of Gaussians are used in the HMMs and/or in the reverberation model, $\log\{(f_{\Lambda\eta}\}$ is a sum of weighted quadratic functions and, in general, exhibits several local maxima.

The constraint of the inner optimization problem depends on the kind of features used, since the combination operation of the HMM output features and the reverberation model output features is feature-dependent and is given for melspec features, logmelspec features, and MFCCs in Eqs. 18.25, 18.26, and 18.27. For all three kinds of features, the constraint is a nonlinear function. Note that the independent variables to be optimized are $\boldsymbol{s}(k-0), \boldsymbol{h}(0,k), \ldots, \boldsymbol{h}(M-1,k)$. The terms $\boldsymbol{s}(k-M+1), \ldots, \boldsymbol{s}(k-1)$ are known from previous iterations, since they are given by the clean-speech feature estimates $\check{\boldsymbol{s}}_{ij}(k-M+1)$, …, $\check{\boldsymbol{s}}_{ij}(k-1)$.

The discussion above shows that the complexity of the inner optimization problem depends both on the output densities of the HMM and the reverberation model, and the kind of features used in the recognizer. In general, numerical optimization methods have to be employed for the solution of the inner optimization problem.

If single Gaussian densities are used and the constraint is linearized, a closed-form solution of the inner optimization problem can be found in the melspec domain. This solution is derived in the following section as an example of how to solve the inner optimization problem.

18.8.6 Solution of the Inner Optimization Problem in the Melspec Domain for Single Gaussian Densities

In the melspec domain, the inner optimization problem can be expressed as

$$\tilde{O}_{ij}(k) = \max_{\boldsymbol{s}_{\mathrm{m}}(k),\boldsymbol{H}_{\mathrm{m}}(k)} \left\{ \log\{f_{\Lambda\eta}\} \right\} \tag{18.41}$$

$$\text{subject to} \quad \boldsymbol{y}_{\mathrm{m}}(k) = \boldsymbol{h}_{\mathrm{m}}(0,k) \odot \boldsymbol{s}_{\mathrm{m}}(k) + \sum_{m=1}^{M-1} \boldsymbol{h}_{\mathrm{m}}(m,k) \odot \check{\boldsymbol{s}}_{\mathrm{m},ij}(k-m) \; . \tag{18.42}$$

Using single Gaussian densities both in the HMMs and the reverberation model, the objective function $\log\{f_{\Lambda\eta}\}$ becomes a multivariate quadratic function. If the constraint in the melspec domain (Eq. 18.42) is linearized, an optimization problem with a quadratic cost function and a linear constraint is obtained, which exhibits a single unique solution. The determination of this solution using the method of Lagrange multipliers is described in the following.

We introduce a simplified notation which neglects the subscript m indicating "melspec domain", the dependencies on the frame index k and the partial state sequence $Q_{ij}(k)$ as follows: $\boldsymbol{s}_{\mathrm{m}}(k-0) := \boldsymbol{s}_0$, $\check{\boldsymbol{s}}_{\mathrm{m},ij}(k-m) := \check{\boldsymbol{s}}_m$, $\boldsymbol{y}_{\mathrm{m}}(k) := \boldsymbol{y}$, $\boldsymbol{h}_{\mathrm{m}}(m,k) := \boldsymbol{h}_m$. That is, $\boldsymbol{y}$ is the current reverberant feature vector, $\boldsymbol{s}_0$ is the current clean-speech feature vector and $\check{\boldsymbol{s}}_m$ is the estimated clean-speech vector for frame $k-m$. $\boldsymbol{h}_m$ is the m-th column of the current melspec RIR representation (see Fig. 18.13 for illustration).

With this simplified notation, the constraint of Eq. 18.42 can be written as

$$\boldsymbol{y} = \underline{\boldsymbol{h}_0} \odot \underline{\boldsymbol{s}_0} + \sum_{m=1}^{M-1} \underline{\boldsymbol{h}_m} \odot \overline{\check{\boldsymbol{s}}_m} \; , \tag{18.43}$$

where the *underlined vectors* are unknown realizations of multivariate Gaussian random vectors with diagonal covariance matrix and the *overlined vectors* are known from previous iterations.

To linearize the constraint, we approximate the generally non-Gaussian random vector $\tilde{\mathcal{Y}}_0 = \mathcal{H}_0 \odot \mathcal{S}_0$ describing the realizations $\tilde{\boldsymbol{y}}_0 = \boldsymbol{h}_0 \odot \boldsymbol{s}_0$ by a Gaussian random vector $\mathcal{Y}_0$ with the same mean and variance as $\tilde{\mathcal{Y}}_0$. The realizations of $\mathcal{Y}_0$ are denoted $\boldsymbol{y}_0$. Thus we obtain the following linear constraint

$$\boldsymbol{y} = \underline{\boldsymbol{y}_0} + \sum_{m=1}^{M-1} \underline{\boldsymbol{h}_m} \odot \overline{\check{\boldsymbol{s}}_m} \; . \tag{18.44}$$

Based on this constraint, a two-step closed-form solution of the inner optimization problem can be derived as follows:

First step: Find $\boldsymbol{y}_0$ and $\boldsymbol{h}_{m'}$.

We apply the method of Lagrange multipliers (see e. g. [41], appendix B.2) to

$$\max_{\boldsymbol{y}_0,\boldsymbol{h}_1,\ldots,\boldsymbol{h}_{M-1}} \left\{ f_{\mathcal{Y}_0}(\boldsymbol{y}_0) \cdot f_\eta(\boldsymbol{h}_1) \cdot \ldots \cdot f_\eta(\boldsymbol{h}_{M-1}) \right\} \quad \text{subject to Eq. 18.44} \; , \tag{18.45}$$

where $f_{\mathcal{Y}_0}(\boldsymbol{y}_0)$ is the probability density of $\mathcal{Y}_0$ evaluated at $\boldsymbol{y}_0$, $f_\eta(\boldsymbol{h}_m)$ is the probability density of the m-th column of the reverberation model evaluated at $\boldsymbol{h}_m$. Since the columns of the reverberation model are assumed to be statistically independent as described in Sec. 18.8.2,

$$f_\eta(\boldsymbol{h}_0) \cdot \ldots \cdot f_\eta(\boldsymbol{h}_{M-1}) = f_\eta\big(\boldsymbol{H}(k)\big) \; .$$

Using the negative logarithm of the densities to be maximized and neglecting irrelevant constants, the Lagrangian function $\mathcal{L}_1$ is obtained as

$$\begin{aligned} \mathcal{L}_1 = \frac{\big(\boldsymbol{y}_0 - \mu_{\boldsymbol{y}_0}\big)^2}{2\,\sigma^2_{\boldsymbol{y}_0}} + \sum_{m=1}^{M-1} \frac{\big(\boldsymbol{h}_m - \mu_{\boldsymbol{h}_m}\big)^2}{2\,\sigma^2_{\boldsymbol{h}_m}} \\ + \nu_1 \cdot \left(\boldsymbol{y} - \boldsymbol{y}_0 - \sum_{m=1}^{M-1} \boldsymbol{h}_m \odot \check{\boldsymbol{s}}_m \right) \; , \end{aligned} \tag{18.46}$$

where the squaring and the division operations are performed element-wise (as for the remainder of this section), ν_1 is the Lagrange multiplier, and $\mu_{\boldsymbol{h}_m}$ and $\sigma^2_{\boldsymbol{h}_m}$ denote the mean and the variance vector of $\boldsymbol{h}_m$, respectively, and likewise for the other variables.

Setting the derivatives of the Lagrangian $\mathcal{L}_1$ with respect to $\boldsymbol{y}_0$, $\boldsymbol{h}_1$, …, $\boldsymbol{h}_{M-1}$, and ν_1 to zero and solving the resulting system of equations, we obtain $\hat{\boldsymbol{y}}_0$ and $\hat{\boldsymbol{h}}_{m'}$, for $m' = 1, \ldots, M-1$, as solutions

$$\begin{aligned} \hat{\boldsymbol{y}}_0 = \frac{\sum_{m=1}^{M-1} \check{\boldsymbol{s}}_m^2 \odot \sigma^2_{\boldsymbol{h}_m}}{\sigma^2_{\boldsymbol{y}_0} + \sum_{m=1}^{M-1} \check{\boldsymbol{s}}_m^2 \odot \sigma^2_{\boldsymbol{h}_m}} \odot \mu_{\boldsymbol{y}_0} \\ + \frac{\sigma^2_{\boldsymbol{y}_0}}{\sigma^2_{\boldsymbol{y}_0} + \sum_{m=1}^{M-1} \check{\boldsymbol{s}}_m^2 \odot \sigma^2_{\boldsymbol{h}_m}} \odot \left(\boldsymbol{y} - \sum_{m=1}^{M-1} \check{\boldsymbol{s}}_m \odot \mu_{\boldsymbol{h}_m} \right) , \end{aligned} \tag{18.47}$$

$$\begin{aligned} \hat{\boldsymbol{h}}_{m'} = \frac{\sigma^2_{\boldsymbol{y}_0} + \sum_{\substack{m=1 \\ m \neq m'}}^{M-1} \check{\boldsymbol{s}}_m^2 \odot \sigma^2_{\boldsymbol{h}_m}}{\sigma^2_{\boldsymbol{y}_0} + \sum_{m=1}^{M-1} \check{\boldsymbol{s}}_m^2 \odot \sigma^2_{\boldsymbol{h}_m}} \odot \mu_{\boldsymbol{h}_{m'}} \\ + \frac{\check{\boldsymbol{s}}^2_{m'} \odot \sigma^2_{\boldsymbol{h}_{m'}}}{\sigma^2_{\boldsymbol{y}_0} + \sum_{m=1}^{M-1} \check{\boldsymbol{s}}_m^2 \odot \sigma^2_{\boldsymbol{h}_m}} \odot \frac{1}{\check{\boldsymbol{s}}_{m'}} \odot \left(\boldsymbol{y} - \mu_{\boldsymbol{y}_0} - \sum_{\substack{m=1 \\ m \neq m'}}^{M-1} \check{\boldsymbol{s}}_m \odot \mu_{\boldsymbol{h}_m} \right) . \end{aligned} \tag{18.48}$$

Second step: Find $\boldsymbol{h}_0$ and $\boldsymbol{s}_0$ given $\hat{\boldsymbol{y}}_0$.

Applying the method of Lagrange multipliers to

$$\max_{\boldsymbol{s}_0,\boldsymbol{h}_0} \left\{ f_\Lambda(j,\boldsymbol{s}_0) \cdot f_\eta(\boldsymbol{h}_0) \right\} \quad \text{subject to} \quad \overline{\hat{\boldsymbol{y}}_0} = \underline{\boldsymbol{h}_0} \odot \underline{\boldsymbol{s}_0} \;, \tag{18.49}$$

replacing the densities with their negative logarithm, and neglecting irrelevant constants, we obtain the following Lagrangian function

$$\mathcal{L}_2 = \frac{(\boldsymbol{s}_0 - \mu_{\boldsymbol{s}_0})^2}{2\,\sigma^2_{\boldsymbol{s}_0}} + \frac{(\boldsymbol{h}_0 - \mu_{\boldsymbol{h}_0})^2}{2\,\sigma^2_{\boldsymbol{h}_0}} + \nu_2 \cdot (\hat{\boldsymbol{y}}_0 - \boldsymbol{h}_0 \odot \boldsymbol{s}_0) \;. \tag{18.50}$$

Setting the derivatives of the Lagrangian $\mathcal{L}_2$ with respect to $\boldsymbol{h}_0$, $\boldsymbol{s}_0$, and ν_2 to zero and solving the resulting system of equations, we obtain the following fourth-order equation to be fulfilled by the desired vector $\boldsymbol{h}_0$

$$\sigma^2_{\boldsymbol{s}_0} \odot \boldsymbol{h}_0^4 - \mu_{\boldsymbol{h}_0} \odot \sigma^2_{\boldsymbol{s}_0} \odot \boldsymbol{h}_0^3 + \mu_{\boldsymbol{s}_0} \odot \sigma^2_{\boldsymbol{h}_0} \odot \hat{\boldsymbol{y}}_0 \odot \boldsymbol{h}_0 - \hat{\boldsymbol{y}}_0^2 \odot \sigma^2_{\boldsymbol{h}_0} = 0 \;, \tag{18.51}$$

where the exponents denote element-wise powers. It can be shown that this equation has a pair of complex conjugate solutions, one real-valued positive and one real-valued negative solution. As only the real-valued positive solution achieves the maximization of the desired probability, we obtain exactly one vector $\hat{\boldsymbol{h}}_0$ and thus exactly one vector $\hat{\boldsymbol{s}}_0$

$$\hat{\boldsymbol{s}}_0 = \frac{\hat{\boldsymbol{y}}_0}{\hat{\boldsymbol{h}}_0} \;. \tag{18.52}$$

In this way, $\hat{\boldsymbol{s}}_{ij}(k)$ and $\hat{\boldsymbol{H}}_{ij}(k)$ are obtained so that $O_{ij}(k)$ can be calculated as

$$O_{ij}(k) = f_\Lambda\big(j, \hat{\boldsymbol{s}}_{ij}(k)\big) \cdot f_\eta\big(\hat{\boldsymbol{H}}_{ij}(k)\big) \;.$$

18.8.7 Simulations

To investigate the effectiveness of the REMOS concept, simulations of a connected digit recognition (CDR) task using melspec features and single Gaussian densities are performed. The performance of the proposed approach is compared to that of conventional HMM-based recognizers trained on clean and reverberant speech, respectively.

The REMOS concept is implemented by extending the functionality of HTK [32] with the inner optimization as described in Sec. 18.8.6. HTK employs Viterbi beam search implemented by the so-called *token passing paradigm* as continuous speech recognition search algorithm [77].

The CDR task is chosen for evaluation, since it can be considered as one of the easiest examples of continuous speech recognition. Furthermore, the probability of the current digit can be assumed to be independent of the preceding digits so that a language model is not required. Therefore, the recognition rate is solely determined by the quality of the acoustic model,

making the CDR task well suited for the evaluation of the REMOS concept, which aims at improving the acoustic model.

The simulations are performed using RIRs measured in three different rooms. Room A is a lab environment, room B a studio environment and room C a lecture room. The details of the room characteristics are summarized in Tab. 18.1. Note that room A is a moderately reverberant environment while room B and room C are highly reverberant environments. A set of RIRs is measured for different loudspeaker and microphone positions in each room. In room C, three RIR sets with different loudspeaker/microphone-distances are measured which are denoted C1, C2 and C4, where the number corresponds to the distance in meter. Each set of RIRs is split into two disjoint sets, one used for training and the other used for test (see Tab. 18.1 for detailed numbers). In this way, a strict separation of test and training data is achieved.

Table 18.1. Summary of room characteristics: T_{60} is the reverberation time, d the distance between speaker and microphone and SRR is the signal-to-reverberation-ratio.

	Room A	Room B	Room C1	Room C2	Room C4
Type	Lab	Studio	Lecture rooms		
T_{60}	300 ms	700 ms	900 ms	900 ms	900 ms
d	2.0 m	4.1 m	1.0 m	2.0 m	4.0 m
SRR	4.0 dB	−4.5 dB	7.4 dB	2.9 dB	−1.5 dB
Number of training RIRs	36	18	36	72	44
Number of test RIRs	18	6	18	36	22
Length of rev. model M	20	50	70	70	70

The used feature vectors are calculated in the following way: The speech signal, sampled at 20 kHz, is decomposed into overlapping frames of length 25 ms with a frame shift of 10 ms. After applying a first-order pre-emphasis (coefficient 0.97) and a Hamming window, a 512-point DFT is computed. From the DFT representation, 24 melspec coefficients are calculated. Only static features and no Δ and $\Delta\Delta$ coefficients are used.

A 16-state left-to-right model without skips over states is trained for each of the 11 digits ('0'-'9' and 'oh'). Additionally, a three-state silence model with a backward skip from state 3 to state 1 is trained. The output densities are single Gaussians with diagonal covariance matrices. All HMMs are trained according to the following procedure: First, single Gaussian MFCC-based HMMs are trained by 10 iterations of Baum-Welch re-estimation [33].

Then the melspec HMMs are obtained from the MFCC HMMs by single-pass retraining [74]. In this way, more reliable models are obtained than by training melspec models from scratch.

For the training, 4579 connected digit utterances corresponding to 1.5 hours of speech from the TI digits [40] training data are used. For the training with reverberant speech, the clean training data are convolved with measured RIRs randomly selected from the training set of the corresponding room. A uniform distribution is employed for the random selection so that a balanced use of all RIRs is ensured. The HMMs trained on clean data are denoted λ_{clean}, the HMMs trained on data convolved with RIRs from room A are denoted λ_A and so on. The corresponding HMM networks are denoted $\mathcal{N}_{\lambda_{\text{clean}}}$, $\mathcal{N}_{\lambda_A}$ and so on. For the conventional HMM-based clean recognizer and for the REMOS-based recognizer, identical HMM networks are used. The HMM network of the conventional reverberant recognizers for each room shares the same structural parameters and the same training procedure but differs with respect to the training data.

For the recognition, a silence model is added in the beginning and at the end of the HMM network consisting of the 11 digit-HMMs connected in a loop similar to Fig. 18.9. As test data, 512 test utterances randomly selected from the TI digits test set are used. To obtain the reverberant test data, the clean test data are convolved with RIRs randomly selected from the test set of the corresponding rooms.

To train the reverberation models for each room, the measured RIRs from the corresponding training set are used according to the procedure described in Sec. 18.8.3. The reverberation models are denoted according to the rooms where the RIRs have been measured. E. g., the reverberation model of room A is denoted η_A. In addition to the reverberation models η_{C1}, η_{C2}, η_{C4}, a universal model for room C is trained using all training RIRs measured in room C. This model is denoted η_{C124}.

In a first test series, the performance of REMOS is compared to conventional HMM-based recognizers. Tab. 18.2 shows the word accuracies achieved with conventional HMM-based recognizers and with the REMOS concept for the connected digit recognition task in the rooms described above. The relatively low accuracy of 82 % achieved by applying the conventional HMM-based recognizer using clean HMMs to the clean test data (clean-speech performance) results from the fact that melspec features cannot be modeled very accurately by single Gaussian densities. With increasing reverberation, the accuracy decreases significantly if HMMs trained on clean speech are used in the conventional HMM-based recognizers. The accuracy is improved to some extent if HMMs trained with reverberant data from the corresponding rooms are used.

The lower recognition rate in room B compared to room C4 for the clean HMM-based recognizer can be explained by the strong low-pass characteristic of the transfer functions corresponding to the RIRs measured in room B. Therefore, the mismatch between the clean training data and the reverberant

test data is larger in room B than in room C4. As the low-pass characteristic can be modeled very well by the reverberant training, the performance increase between clean and reverberant training is higher in room B than in room C4.

The word accuracy achieved by the REMOS concept is significantly higher than that of the reverberant HMM-based recognizers in all three rooms. In room A, the recognition rate of REMOS even approaches the clean-speech performance. The performance gain compared to the reverberant training increases with growing reverberation from 10.8 % absolute in room A to 21.6 % absolute in room C4. These results confirm that the REMOS concept is much more robust to reverberation than conventional HMM-based recognizers, even if the latter use HMMs trained on reverberant data.

Table 18.2. Comparison of word accuracies of a conventional HMM-based recognizer and of the proposed REMOS concept in the melspec domain using single Gaussian densities.

Test data	Recognizer						
	Conventional HMM-based				REMOS concept		
	Clean training		Reverberant training				
	HMM	Acc.	HMM	Acc.	HMM	Rev. model	Acc.
Clean	$\mathcal{N}_{\lambda_{\text{clean}}}$	82.0 %	-	-	-	-	-
Room A	$\mathcal{N}_{\lambda_{\text{clean}}}$	51.5 %	$\mathcal{N}_{\lambda_A}$	66.8 %	$\mathcal{N}_{\lambda_{\text{clean}}}$	η_A	77.6 %
Room B	$\mathcal{N}_{\lambda_{\text{clean}}}$	13.4 %	$\mathcal{N}_{\lambda_B}$	54.6 %	$\mathcal{N}_{\lambda_{\text{clean}}}$	η_B	71.6 %
Room C4	$\mathcal{N}_{\lambda_{\text{clean}}}$	25.9 %	$\mathcal{N}_{\lambda_{C4}}$	46. 0%	$\mathcal{N}_{\lambda_{\text{clean}}}$	η_{C4}	67.6 %

In a second test series, the sensitivity of the REMOS concept to a mismatch between the set-up in the target environment and the reverberation model is investigated. Therefore, the reverberation models η_{C1}, η_{C2}, η_{C4}, η_{C124} are applied to the test data of the scenarios C1, C2 and C4. The word accuracies for all possible combinations are summarized in Tab. 18.3. The results for scenario C1 are similar for all reverberation models, while significant differences between different reverberation models are observed for the set-ups C2 and C4. For all of the tested loudspeaker/microphone-distances, the matched model (e. g., η_{C2} for scenario C2) achieves the best results among all models or is at least close to the best result. Using a reverberation model with higher SRR than the test conditions (e. g., η_{C1} for scenario C2), decreases the recognition rate much more than using a reverberation model with lower SRR (e. g., η_{C4} for scenario C2).

The reverberation model η_{C124} trained on RIRs with different loudspeaker/microphone-distances performs very well for all scenarios C1, C2 and

C4. For the test data with a loudspeaker/microphone-distance of 4 m (scenario C4) it even outperforms the matched model. In summary, we can conclude that using RIRs measured at various loudspeaker and microphone positions with various distances in the target environment enables the training of a reverberation model which achieves a good performance in the target environment regardless of the loudspeaker/microphone-distance.

Table 18.3. Word accuracy of the REMOS concept for test data with different loudspeaker/microphone-distances in room C and different reverberation models.

Test data	Reverberation model			
	η_{C1}	η_{C2}	η_{C4}	η_{C124}
Room C1	73.9 %	74.5 %	73.0 %	73.2 %
Room C2	58.8 %	71.7 %	68.0 %	71.4 %
Room C4	45.6 %	46.9 %	67.6 %	70.2 %

The performance of the REMOS concept as a function of the reverberation model length M is investigated in a third test series in room C4. Therefore, the model η_{C4} with an original length of $M = 70$, covering a reverberation time of 700 ms is truncated to the lengths given in Tab. 18.4. For all tests in this series, the test data of scenario C4 are used.

Table 18.4. Word accuracy of the REMOS concept for room C4 and different lengths of the reverberation model η_{C4}.

Length M of rev. model	1	2	3	4	6	8	10
Accuracy	21.3 %	27.5 %	31.4 %	36.5 %	39.8 %	43.7 %	45.8 %
Length M of rev. model	15	20	30	40	50	60	70
Accuracy	48.3 %	51.1 %	58.7 %	62.5 %	66.4 %	67.5 %	67.5 %

Tab. 18.4 and Fig. 18.22 show that the word accuracy increases monotonically with increasing length M of the reverberation model. At the first glance, it might be surprising that for $M = 1$, the recognition rate of the REMOS concept is slightly lower than that of the clean HMM-based recognizer. Even with a one-frame reverberation model, REMOS can compensate for differences in the transfer function of training and test data. However, the energy of the reverberation model is reduced by the truncation so that the

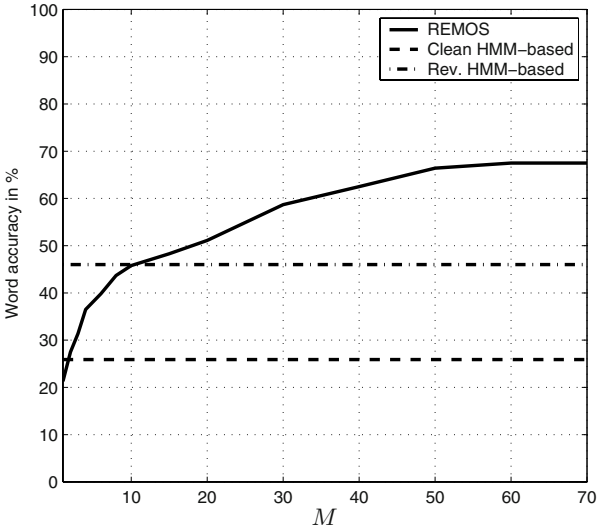

Fig. 18.22. Word accuracy of REMOS in room C4 as a function of the length M of the reverberation model η_{C4}.

resulting mismatch in the signal energy between the test sequence and the model causes the slight decrease in recognition rate.

Already with a length of $M = 10$, the REMOS concept achieves the same recognition rate as the conventional HMM-based recognizer trained on reverberant data. A further increase in the reverberation model length M leads to further significant gains in the recognition rate until a saturation can be observed for lengths larger than $M = 60$. This curve confirms that by modeling the effect of reverberation not simply by a multiplication in the feature domain but rather by a feature-domain convolution, REMOS has the capability to significantly outperform HMM-based recognizers, even if they are trained on reverberant data. If context-dependent sub-word HMMs (e. g. triphones) are used instead of word HMMs, the context of the HMMs is reduced and the gain of REMOS compared to reverberantly trained HMM-based recognizers is expected to increase further.

18.9 Summary and Conclusions

In this contribution, the progress towards robust distant-talking speech recognition in reverberant environments has been reviewed and a novel concept has been described. Since the length of the RIR describing the acoustic path between speaker and microphone is significantly larger than the frame length used for short-time spectrum analysis in the ASR feature extraction, the RIR extends over several frames. Therefore, reverberation has a dispersive effect

on the feature vector sequences used for ASR so that the current feature vector strongly depends on the previous feature vectors. This contradiction to the conditional independence assumption of HMMs, which are state-of-the-art in acoustic-phonetic modeling, has been identified as the main performance limitation of HMM-based recognizers in reverberant environments.

The numerous approaches to improve the ASR performance in reverberant environments have been classified into three groups according to the function block of the ASR system they are applied to. Preprocessing algorithms like blind dereverberation and beamforming aim at removing or at least reducing the reverberation of the input signal before the feature vectors are calculated. Robust speech features and feature-domain compensation techniques try to remove the effect of reverberation at the feature level. Alternatively, the acoustic model of the ASR system can be adjusted to reverberation. This can be performed either by training the HMMs with reverberant data or by adapting well-trained clean-speech HMMs using a few calibration utterances recorded in the target environment.

Finally, a novel concept based on reverberation modeling for speech recognition (REMOS) has been discussed. A combination of an HMM network and a feature-domain reverberation model is used to determine the acoustic score. During recognition, an optimization problem is solved in each iteration of the extended Viterbi algorithm to find the most likely contribution of the HMM network and the reverberation model to the current reverberant observation vector. The complexity of this inner optimization depends both on the kind of features and the output densities used in the HMM and the reverberation model. In general, it has to be solved by numerical optimization algorithms.

For melspec features and single Gaussian densities a closed form solution is possible. Based on this solution, simulations of a connected digit recognition task have been performed in three different rooms. These simulations confirm that the REMOS concept, which explicitely models the dispersive character of reverberation, achieves significantly better recognition rates than conventional HMM-based algorithms, even if the latter are trained on reverberant data. Future work on the REMOS concept includes incorporation of more powerful speech features, like MFCCs, and more accurate output densities, like mixtures of Gaussians, as well as the application of REMOS to more complex tasks, such as large-vocabulary continuous speech recognition for more natural human/machine speech interfaces.

References

1. J. B. Allen, D. A. Berkley: Image method for efficiently simulating small-room acoustics, *JASA*, **65**(4), 943–950, April 1979.
2. AMI project: "Webpage of the AMI project," http://corpus.amiproject.org.
3. B. Atal: Effectiveness of linear prediction characteristics of the speech wave for automatic speaker identification and verification, *JASA*, **55**(6), 1304–1312, 1974.

4. L. E. Baum, J. A. Eagon: An inequality with applications to statistical estimation for probabilistic functions of Markov processes and to a model for ecology, *Bulletin of American Mathematical Society*, **73**, 360–363, 1967.
5. L. E. Baum, et al.: A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains, *Annals of Mathematical Statistics*, **41**, 164–171, 1970.
6. J. Benesty: Adaptive eigenvalue decomposition algorithm for passive acoustic source localization, *Journal of the Acoustical Society of America*, **107**(1), 384–391, Jan. 2000.
7. J. Benesty, S. Makino, J. Chen (eds.): *Speech Enhancement*, Berlin, Germany: Springer, 2005.
8. M. Brandstein, D. Ward (eds.): *Microphone Arrays*, Berlin, Germany: Springer, 2001.
9. C. Breining, P. Dreiseitel, E. Hänsler, A. Mader, B. Nitsch, H. Puder, T. Schertler, G. Schmidt, J. Tilp: Acoustic echo control. An application of very-high-order adaptive filters, *IEEE Signal Process. Mag.*, **16**(4), 42–69, 1999.
10. H. Buchner, R. Aichner, W. Kellermann: TRINICON: A versatile framework for multichannel blind signal processing, *Proc. ICASSP '04*, **3**, 889–892, Montreal, Canada, 2004.
11. CHIL project: "Webpage of the CHIL project," http://chil.server.de.
12. S. Davis, P. Mermelstein: Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences, *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-28**(4), 357–366, 1980.
13. S. Furui: On the role of spectral transition for speech perception, *JASA*, **80**(4), 1016–1025, 1986.
14. K. Furuya, S. Sakauchi, A. Kataoka: Speech dereverberation by combining MINT-based blind deconvolution and modified spectral subtraction, *Proc. ICASSP '06*, **1**, 813–816, Toulouse, France, 2006.
15. K. Furuya, A. Kataoka: Robust speech dereverberation using multichannel blind deconvolution with spectral subtraction, *IEEE Trans. Audio Speech Language Process.*, **T-ASLP-15**(5), 1579–1591, 2007.
16. M. J. F. Gales, S. J. Young: Robust continuous speech recognition using parallel model combination, *IEEE Trans. Speech Audio Process.*, **T-SAP-4**(5), 352–359, 1996.
17. N. D. Gaubitch, P. A. Naylor, D. B. Ward: On the use of linear prediction for dereverberation of speech, *Proc. IWAENC '03*, 99–102, Kyoto, Japan, 2003.
18. B. W. Gillespie, L. E. Atlas: Strategies for improving audible quality and speech recognition accuracy of reverberant speech, *Proc. ICASSP '03*, **1**, 676–679, Hong Kong, 2003.
19. D. Giuliani, M. Matassoni, M. Omologo, P. Svaizer: Training of HMM with filtered speech material for hands-free recognition, *Proc. ICASSP '99*, **1**, 449–452, Phoenix, AZ, USA, 1999.
20. S. M. Griebel, M. S. Brandstein: Microphone array speech dereverberation using coarse channel modeling, *Proc. ICASSP '01*, **1**, 201–204, Salt Lake City, UT, USA, 2001.
21. L. Griffiths, C. Jim: An alternative approach to linearly constrained adaptive beamforming, *IEEE Trans. on Antennas and Propagation.*, **30**(1), 27–34, 1982.
22. M. I. Gürelli, C. L. Nikias: EVAM: an eigenvector-based algorithm for multichannel blind deconvolution of input colored signals, *IEEE Trans. on Signal Processing*, **T-SP-43**(1), 134–149, 1995.

23. T. Haderlein, E. Nöth, W. Herbordt, W. Kellermann, H. Niemann: Using Artificially Reverberated Training Data in Distant Talking ASR, in *Proc. TSD '05*, V. Matoušek, P. Mautner, T. Pavelka (eds.), 226–233, Berlin, Germany: Springer, 2005.
24. E. Hänsler, G. Schmidt (eds.): *Topics in Acoustic Echo and Noise Control: Selected Methods for the Cancellation of Acoustical Echoes, the Reduction of Background Noise, and Speech Processing*, Berlin, Germany: Springer, 2006.
25. B. Hanson, T. Applebaum: Robust speaker-independent word recognition using static, dynamic and acceleration features: Experiments with lombard and noisy speech, *Proc. ICASSP '90*, **2**, 857–860, Albuquerque, NM, USA, 1990.
26. W. Herbordt: *Sound Capture for Human/Machine Interfaces – Practical Aspects of Microphone Array Signal Processing*, Heidelberg, Germany: Springer, 2005.
27. W. Herbordt, H. Buchner, S. Nakamura, W. Kellermann: Multichannel bin-wise robust frequency-domain adaptive filtering and its application to adaptive beamforming, *Trans. Audio Speech Language Process.*, **T-ASLP-15**(4), 1340–1351, 2007.
28. H. Hermansky, N. Morgan: RASTA processing of speech, *IEEE Trans. Speech Audio Process.*, **T-SAP-2**(4), 578–589, 1994.
29. T. Hikichi, M. Delcroix, M. Miyoshi: Blind dereverberation based on estimates of signal transmission channels without precise information of channel order, *Proc. ICASSP '05*, **1**, 1069–1072, Philadelphia, PA, USA, 2005.
30. H.-G. Hirsch, H. Finster: A new HMM adaptation approach for the case of a hands-free speech input in reverberant rooms, *Proc. INTERSPEECH '06*, 781–783, Pittsburgh, PA, USA, 2006.
31. O. Hoshuyama, A. Sugiyama, A. Hirano: A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters, *IEEE Trans. Signal Process.*, **T-SP-47**(10), 2677–2684, 1999.
32. HTK: "HTK webpage," http://htk.eng.cam.ac.uk.
33. X. Huang, A. Acero, H.-W. Hon: *Spoken Language Processing: A Guide to Theory, Algorithm, and System Development*, Upper Saddle River, NJ, USA: Prentice Hall, 2001.
34. F. Jelinek: *Statistical Methods for Speech Recognition*, Cambridge, MA, USA: MIT Press, 1998.
35. J.-C. Junqua: *Robustness in Automatic Speech Recognition*, Boston, MA: Kluwer Academic Publishers, 1996.
36. K. Kinoshita, T. Nakatani, M. Miyoshi: Fast estimation of a precise dereverberation filter based on speech harmonicity, *Proc. ICASSP '05*, **1**, 1073–1076, Philadelphia, PA, USA, 2005.
37. H. Kuttruff: *Room Acoustics*, 4th ed., London, UK: Spon Press, 2000.
38. C.-H. Lee, C.-H. Lin, B.-H. Juang: A study of speaker adaptation of continuous density HMM parameters, *Proc. ICASSP '90*, **1**, 145–148, Albuquerque, NM, USA, 1990.
39. C. J. Leggetter, P. C. Woodland: Speaker adaptation of continuous density HMMs using multivariate linear regression, *Proc. ICSLP '94*, **2**, 451–454, Yokohama, Japan, 1994.
40. R. G. Leonard: A database for speaker-independent digit recognition, *Proc. ICASSP '84*, 42.11.1–42.11.4, San Diego, CA, USA, 1984.

41. D. G. Manolakis, V. K. Ingle, S. M. Kogon: *Statistical and Adaptive Signal Processing: Spectral Estimation, Signal Modeling, Adaptive Filtering and Array Processing*, Boston, MA: McGraw-Hill, 2000.
42. M. Miyoshi, Y. Kaneda: Inverse filtering of room acoustics, *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-36**(2), 145–152, February 1988.
43. P. J. Moreno, B. Raj, R. M. Stern: A vector taylor series approach for environment independent speech recognition, *Proc. ICASSP '96*, **2**, 733–736, Atlanta, GA, USA, 1996.
44. S. Nakamura, T. Takiguchi, K. Shikano: Noise and room acoustics distorted speech reognition by HMM composition, *Proc. ICASSP '96*, **1**, 69–72, Atlanta, GA, USA, 1996.
45. T. Nakatani, M. Miyoshi: Blind dereverberation of single channel speech signal based on harmonic structure, *Proc. ICASSP '03*, **1**, 92–95, Hong Kong, 2003.
46. T. Nakatani B.-H. Juang, K. Kinoshita, M. Miyoshi: Speech dereverberation based on probabilistic models of source and room acoustics, *Proc. ICASSP '06*, **1**, 821–824, Toulouse, France, 2006.
47. T. Nakatani, K. Kinoshita, M. Miyoshi: Harmonicity-based blind dereverberation for single-channel speech signals, *IEEE Trans. Audio Speech Language Process.*, **T-ASLP-15**(1) 80–95, Jan. 2007.
48. S. Neely, J. Allen: Invertibility of a room impulse response, *JASA*, **66**(1), 165–169, July 1979.
49. H. Ney, S. Orthmanns: Dynamic programming search for continuous speech recognition, *IEEE Signal Process. Mag.*, **16**(5), 64–63, 1999.
50. M. Omologo, M. Matassoni, P. Svaizer, D. Giuliani: Microphone array based speech recognition with different talker-array positions, *Proc. ICASSP '97*, **1**, 227–230, Munich, Germany, 1997.
51. D. S. Pallett, J. G. Fiscus, W. M. Fisher, J. S. Garofolo, B. S. Lund, A. Martin, M. A. Przybocki: The 1994 benchmark tests for the ARPA spoken language program, *Proc. Spoken Language Technology Workshop*, 5–38, Austin, TX, USA, 1995.
52. D. S. Pallett: A look at NIST's benchmark ASR tests: past, present, and future, *Proc. ASRU '03*, 483–488, St. Thomas, Virgin Islands, 2003.
53. J. G. Proakis, D. G. Manolakis: *Digital Signal Processing: Principles, Algorithms, and Applications*, Upper Saddle River, NJ, USA: Prentice Hall, 1996.
54. W. Putnam, D. Rocchesso, J. Smith: A numerical investigation of the invertibility of room transfer functions, *Proc. WASPAA '95*, 249–252, Mohonk, NY, USA, 1995.
55. L. R. Rabiner: A tutorial on hidden markov models and selected applications in speech recognition, *Proc. IEEE*, **77**(2), 257–286, 1989.
56. C. K. Raut, T. Nishimoto, S. Sagayama:Model adaptation for long convolutional distortion by maximum likelihood based state filtering approach, *Proc. ICASSP '06*, **1**, 1133–1136, Toulouse, France, 2006.
57. A. Sehr, M. Zeller, W. Kellermann: Hands-free speech recognition using a reverberation model in the feature domain, *Proc. EUSIPCO '06*, Florence, Italy, 2006.
58. A. Sehr, M. Zeller, W. Kellermann: Distant-talking continuous speech recognition based on a novel reverberation model in the feature domain, *Proc. INTERSPEECH '06*, 769 – 772, Pittsburgh, PA, USA, 2006.

59. A. Sehr, W. Kellermann: A new concept for feature-domain dereverberation for robust distant-talking ASR, *Proc. ICASSP '07*, **4**, 369–372, Honolulu, Hawaii, 2007.
60. A. Sehr, Y. Zheng, E. Nöth, W. Kellermann: Maximum likelihood estimation of a reverberation model for robust distant-talking speech recognition, *Proc. EUSIPCO '07*, 1299-1303, Poznan, Poland, 2007.
61. M. L. Seltzer, B. Raj, R. M. Stern: Likelihood-maximizing beamforming for robust hands-free speech recognition, *IEEE Trans. Speech Audio Process.*, **T-SAP-12**(5), 489–498, 2004.
62. M. L. Seltzer, R. M. Stern: Subband likelihood-maximizing beamforming for speech recognition in reverberant environments, *Trans. Audio Speech Language Process.*, **T-ASLP-14**(6), 2109–2121, 2006.
63. P. C. W. Sommen: Partitioned frequency domain adaptive filters, *Proc. 23rd Asilomar Conference on Signals Systems and Computers*, 676–681, Pacific Grove, CA, USA, 1989.
64. J. S. Soo, K. K. Pang: Multidelay block frequency domain adaptive filter, *IEEE Trans. Acoust. Speech Signal Process.*, **ASSP-38**(2), 373–376, 1990.
65. J. S. Soo, K. K. Pang: A multistep size (MSS) frequency domain adaptive filter, *IEEE Trans. Signal Process.*, **T-SP-39**(1), 115–121, 1991.
66. V. Stahl, A. Fischer, R. Bippus: Acoustic synthesis of training data for speech recognition in living-room environments, *Proc. ICASSP '01*, **1**, 285–288, Salt Lake City, UT, USA, 2001.
67. T. G. Stockham: High-speed convolution and correlation, *Proc. AFIPS '66*, **28**, 229–233, 1966.
68. T. Takiguchi, S. Nakamura, Q. Huo, K. Shikano: Model adaption based on HMM decomposition for reverberant speech recognition, *Proc. ICASSP '97*, **2**, 827–830, Munich, Germany, 1997.
69. T. Takiguchi, S. Nakamura, K. Shikano: HMM-separation-based speech reognition for a distant moving speaker, *IEEE Trans. Speech Audio Process.*, **T-SAP-9**(2), 127–140, 2001.
70. T. Takiguchi, M. Nishimura, Y. Ariki: Acoustic model adaptation using first-order linear prediction for reverberant speech, *IEICE Trans. Information and Systems*, **E89-D**(3), 908–914, 2006.
71. A. Torger, A. Farina: Real-time partitioned convolution for ambiophonics surround sound, *Proc. WASPAA '01*, 195–198, Mohonk, NY, 2001.
72. A. P. Varga, R. K. Moore: Hidden Markov model decomposition of speech and noise, *Proc. ICASSP '90*, **2**, 845–848, Albuquerque, NM, USA, 1990.
73. B. van Veen, K. Buckley: Beamforming: A versatile approach to spatial filtering, *IEEE ASSP Magazine*, **5**(2), 4–24, 1988.
74. P. C. Woodland, M. J. F. Gales, D. Pye: Improving environmental robustness in large vocabulary speech recognition, *Proc. ICASSP '96*, **1**, 65–68, Atlanta, GA, USA, 1996.
75. B. Yegnanarayana, P. Satyanarayana Murthy: Enhancement of reverberant speech using LP residual signal, *IEEE Trans. Speech Audio Process.*, **T-SAP-8**(3), 267–281, 2000.
76. B. Yegnanarayana, S. R. Mathadeva Prasanna, K Sreenivasa Rao: Speech enhancement using excitation source information, *Proc. ICASSP '02*, **1**, 541–544, Orlando, FL, USA, 2002.

77. S. J. Young, N. H. Russel, J. H. S. Thornton: Token passing: a simple conceptual model for connected speech recognition systems, CUED technical report, Cambridge University Engineering Department, 1989.
78. S. Young, G. Evermann, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, P. Woodland: *The HTK Book (for HTK Version 3.2)*, Cambridge, UK: Cambridge University Engineering Department, 2002.