Diplomarbeit

Thema:

Kanalkompensation in der Spracherkennung

von

Rainer Baumgärtner

Bearbeitungszeitraum:

1. Mai 1996 - 31. Oktober 1996

Institut für Logik, Komplexität und Deduktionssysteme

Betreuer:
Prof. Alexander Waibel
Dipl.-Ing. Martin Westphal

Hiermit erkläre ich, die vorliegende Arbeit selbständig erstellt und keine anderen als die angegebenen Quellen verwendet zu haben.

Karlsruhe, 31. Oktober 1996

Rainer Baumgärtner

Inhaltsverzeichnis

Kapitel 1

Einführung

Die in den letzten Jahren entwickelten Systeme zur automatischen Spracherkennung sind heute in der Lage kontinuierliche Sprache sprecherunabhängig zu verarbeiten. Dabei wird zwar immer noch häufig mit eingeschränkten Vokabularien und Grammatiken gearbeitet, dies ist jedoch zunächst für viele Anwendungen ausreichend. Reale Anwendungen erfordern, daß die Leistung eines Spracherkenners unabhängig von den Aufnahmebedingungen ist. Dies ist bei den meisten heute existierenden Systemen nicht der Fall. Die aufgenommene Sprache wird durch unterschiedliche Mikrofone und wechselnde Aufnahmeumgebungen stark verändert. Zum Erreichen der Sprecherunabhängigkeit werden Aufnahmen von vielen unterschiedlichen Sprechern zum Trainieren des Spracherkenners benutzt. Versucht man nun im Training zusätzlich verschiedene Aufnahmeumgebungen zu verwenden, dann fällt die Erkennungsrate deutlich ab. Außerdem wird damit nur für die in der Trainingsmenge enthaltenen Aufnahmeumgebungen eine Verbesserung erzielt. Die Generalisierung unterschiedlicher Aufnahmeumgebungen ist eine Aufgabe, die nur schwer durch den Spracherkenner selbst geleistet werden kann.

Deshalb sollte es möglich sein Aufnahmen aus unterschiedlichen Aufnahmeszenarien so zu normalisieren, daß einheitliche Muster für den Erkenner zur Verfügung stehen. Dieser Normalisierungsschritt wird **Kanalkompensation** genannt. Dabei wird der Weg vom Mund des Sprechers bis zur digitalen Repräsentation des Sprachsignals im Rechner als Kanal bezeichnet (siehe Abbildung 1.1) .

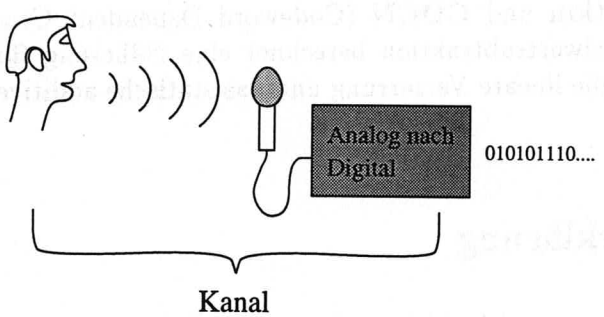


Abbildung 1.1: Die Veränderung des Sprachsignals vom Stimmband bis zur Analogdigitalwandlung wird als Kanal bezeichnet.

Der Kanal kann durch eine Vielzahl von Faktoren beschrieben werden. Viele davon beeinflussen die Erkennungsrate des Spracherkenners. Additives Rauschen von Geräten und

Umwelt, Schallreflektion an Oberflächen, Übersprechen von anderen Sprechern, lineare Verzerrung durch Mikrofon und Analogelektronik und die Struktur des Vokaltraktes unterschiedlicher Sprecher sind nur einige Möglichkeiten (siehe Abbildung 1.2) .

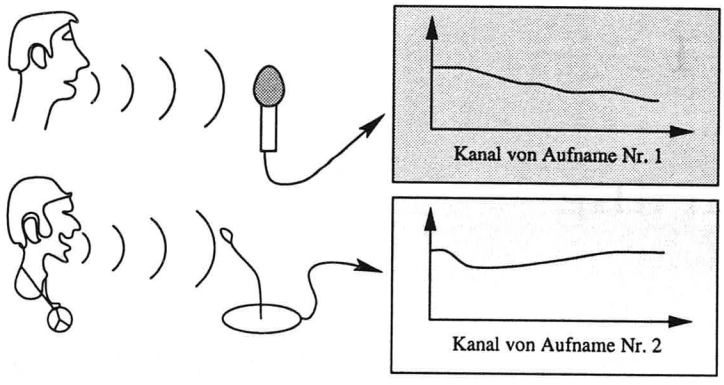


Abbildung 1.2: Unterschiedliche Aufnahmebedingungen führen zu unterschiedlichen Signalformen. Für Spracherkenner sind in diesem Zusammenhang die unterschiedlichen Formen des Amplitudenspektrums störend.

Dabei gibt es zwei Klassen von Kanalfaktoren die mathematisch gut erfaßbar sind, das statische additive Rauschen und die lineare Verzerrung. Alle Hintergrundgeräusche werden zum additiven Rauschen zusammengefaßt. Davon ausgenommen sind alle Kurzzeitereignisse wie Tastaturklappern, Türgeräusche, Stuhlrücken und ähnliches. Die Auswirkungen der linearen Verzerrung können als Durchgang der unverfälschten Sprache durch ein lineares Filter modelliert werden. Ursachen für die lineare Verzerrung sind Vokaltraktunterschiede zwischen verschiedenen Sprechern, Sprachstile (schnell, sanft, gehaucht, geschrieen u.s.w.), die Raumakustik, und die verwendete Aufnahmeelektronik. Sowohl das statische additive Rauschen als auch die lineare Verzerrung sind a priori unbekannt und können nur durch Näherungsverfahren geschätzt werden.

Im Laufe der letzten Jahre wurde eine Reihe von Verfahren zur Kanalkompensation entwickelt. Praktische Erfordernisse wie Rechenzeitbedarf und Reaktionszeit zwingen dabei zur Vereinfachung des Kanalmodells. In dieser Diplomarbeit werden die beiden Verfahren **Mittelwertsubtraktion** und **CDCN** (Codeword Dependent Cepstral Normalization) untersucht. Die Mittelwertsubtraktion berechnet eine Näherung für die lineare Verzerrung. CDCN schätzt die lineare Verzerrung und das statische additive Rauschen durch ein iteratives Verfahren.

1.1 Begriffserklärung

Der Bergriff **Sprachsegment** bezeichnet eine kurze Äußerung (siehe Abbildung 1.3). Gemeint ist damit der gerade vom Spracherkenner bearbeitete Abschnitt des Sprachsignals. Ein Sprachsegment kann aus einem Buchstaben, einem Wort, einem Satz oder einer beliebigen Ansammlung von Worten bestehen. Oft ist der Umfang eines Sprachsegments willkürlich festgelegt. In der englischsprachigen Literatur wird der Begriff "*utterance*" als Synonym für Sprachsegment benutzt.

Beim Übergang in die Spektraldarstellung wird das Sprachsegment in äquidistante Blöcke von Abtastwerten unterteilt. Im folgenden wird dafür der Begriff **Sprachrahmen** oder Rahmen (englisch: "*frame*".) verwendet.

Für den Begriff "logarithmierter Spektralbereich" wird abkürzend Log-Spektralbereich benutzt.

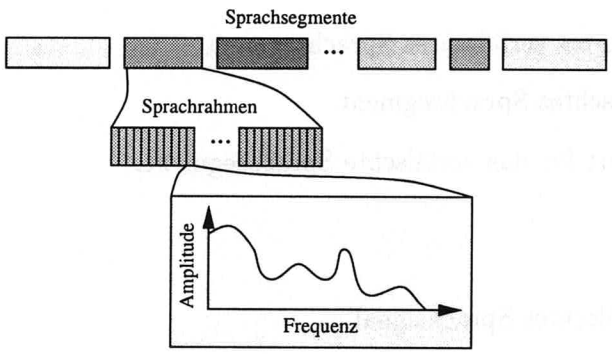


Abbildung 1.3: Kurze Ausschnitte aus einer Sprachaufnahme werden Sprachsegmente bzw. "*utterance*" genannt. Jedes Sprachsegment unterteilt sich in viele Sprachrahmen. Jeder Sprachrahmen repräsentiert ein Kurzzeitspektrum.

1.2 Formelzeichen

Es folgt eine Aufstellung der im weiteren Verlauf benutzten Formelzeichen.

Allgemein

- Z = Beobachtetes verfälschtes Sprachsegment.
- X = Unverfälschtes Sprachsegment.
- Y = Schätzwert für das verfälschte Sprachsegment.

Zeitberreich

- $x[t]$ = Unverfälschtes Sprachsignal .
- $y[t]$ = Schätzwert für das verfälschtes Sprachsignal.
- $h[t]$ = lineare Verzerrung des Kanals.
- $n[t]$ = additives Rauschen des Kanals.

Spektralberreich

- $X(\omega)$ = Leistungsspektrum des unverfälschten Sprachsignals.
- $Y(\omega)$ = Leistungsspektrum des verfälschten Sprachsignals.
- $N(\omega)$ = Leistungsspektrum des additiven Rauschanteils.
- $|H(\omega)|^2$ = Betragsquadrat der Übertragungsfunktion der linearen Verzerrung.

Log-Spektralbereich

- z = Beobachtetes verfälschtes Sprachsignal.
- x = Unverfälschtes Sprachsignal.
- y = Schätzwert für das verfälschte Sprachsignal.
- q = lineare Verzerrung des Kanals.
- n = additives Rauschen.
- $r(x, n, q)$ = Korrekturvektor.
- $s(x, n, q)$ = Korrekturvektor.
- z_i = Sprachrahmen $\quad i = 0, 1, ..., N-1.$
- N = Anzahl der Sprachrahmen z_i im Sprachsegment Z.

- $\lambda = \{(c[0], C[0], P[0]), ..., (c[K-1], C[K-1], P[K-1])\}$ = Kodebuch.
- $c[k]$ = Kodebuchvektor $\quad k = 0, 1, ..., K-1$.
- $C[k]$ = Kovarianzmatrix zum Kodebuchvektor k.
- $P[k]$ = a priori Wahrscheinlichkeit mit der der Kodebuchvektor k auftritt.
- K = Anzahl der Kodebuchvektoren $c[0], ..., c[K-1]$.
- K_{Sil} = Anzahl der Kodebuchvektoren, die Stille repräsentieren $c[0], ..., c[K_{Sil}-1]$.

Kapitel 2

Motivation

Im folgenden soll kurz darauf eingegangen werden was die Gründe für den Einfluß des Kanals auf die Erkennungsleistung eines Spracherkenners sind.

In Abbildung 2.1 sind die ersten zwei Verarbeitungsebenen eines Spracherkenners dargestellt. Das Zeitsignal der zu erkennenden Sprache wird durch die Signalvorverarbeitung einigen grundlegenden Transformationen unterworfen. Häufig sind digitale Filter, die Fouriertransformation, die Logarithmierung und die Mittelwertsubtraktion Bestandteile der Signalvorverarbeitung.

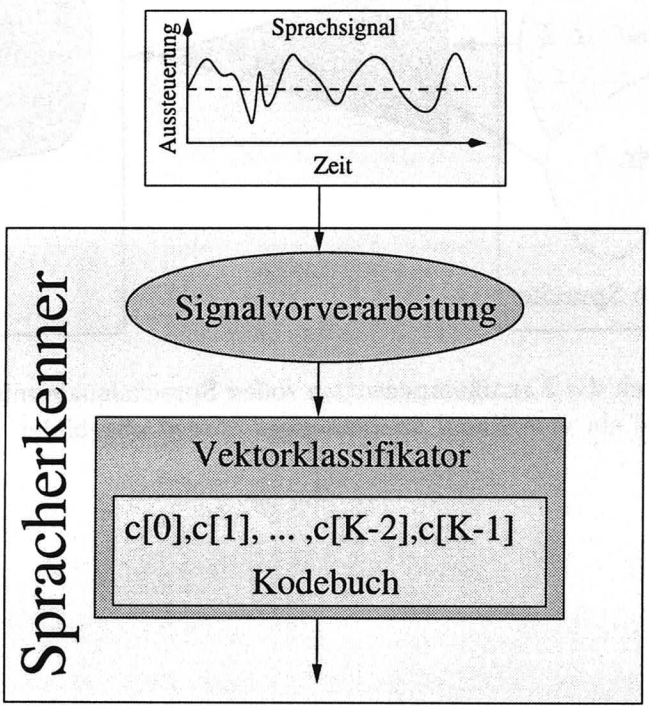


Abbildung 2.1: Die beiden ersten Ebenen eines Spracherkenners. Die Signalvorverarbeitung führt zuerst einige elementare Transformationen aus, anschließend klassifiziert ein Vektorklassifikator die vorverarbeiteten Daten.

Danach werden die vorverarbeiteten Daten einem Vektorklassifikator zugeführt. Basis des Klassifikators bilden eine Anzahl von Merkmalsvektoren $\{c[0], c[1], ..., c[k-1]\}$ die unter

dem Begriff Kodebuch zusammengefaßt sind. Die Merkmalsvektoren werden vorab anhand von Beispielen festgelegt. Man bezeichnet das Festlegen der Merkmalsvektoren als Training.

Weichen die Aufnahmebedingungen von den beim Training der Merkmalsvektoren herrschenden Aufnahmebedingungen ab, dann wird die eindeutige Klassifikation erschwert. Unterschiedliche Kanaleigenschaften führen dabei zu unterschiedlichen Signalrepräsentationen. Die vorverarbeiteten Sprachdaten unterscheiden sich von den Merkmalsvektoren des Kodebuchs. Dadurch wird die Qualität der Klassifikation verschlechtert.

Die durch den Kanal verursachte Signalveränderung trägt nichts zum Informationsgehalt von Sprache bei. Deshalb ist es nicht notwendig den Kanaleinfluß in den Merkmalsvektoren des Klassifikators zu repräsentieren. Vielmehr soll beim Training des Kodebuchs und im späteren Betrieb des Spracherkenners der Einfluß des Kanals vorab eliminiert werden.

Dies wird durch die **Kanalkompensation** erreicht. Dabei werden Sprachsignale mit unterschiedlichen Kanaleigenschaften auf ein vom Kanal unabhängiges Modell abgebildet (siehe Abbildung 2.2).

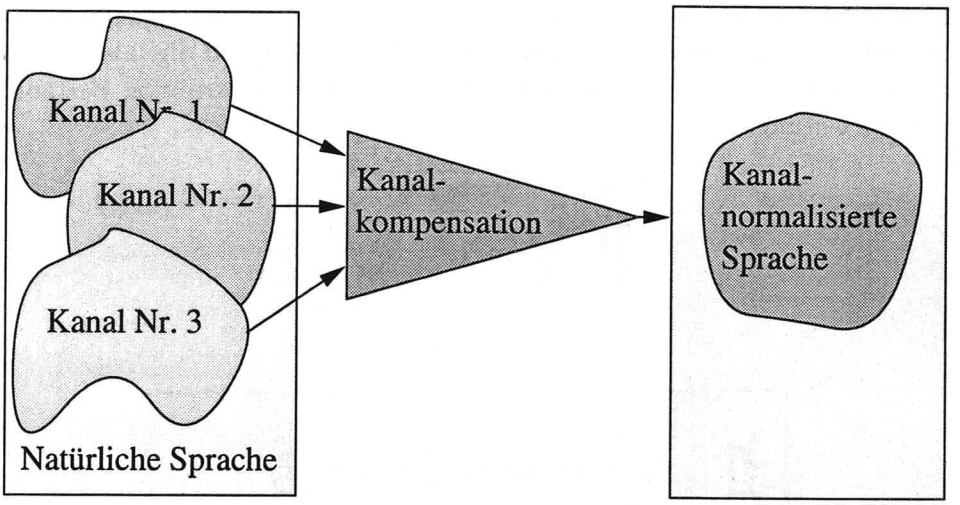


Abbildung 2.2: Durch die Kanalkompensation sollen Sprachsignale mit verschiedenen Kanaleigenschaften auf ein vom Kanal unabhängiges Modell abgebildet werden.

Kapitel 3

Modellierung des Übertragungskanals

Basis der Arbeit ist ein vereinfachtes Modell für den Übertragungskanal (siehe Abbildung 3.1). Man beschränkt sich dabei auf die zwei Haupteinflüsse **lineare Verzerrung** $h[t]$ und **additives Rauschen** $n[t]$.

3.1 Zeitbereich

Die lineare Verzerrung kann durch ein lineares Filter modelliert werden. Das unverfälschte Eingangssignal $x[t]$ wird zuerst mit der Impulsantwort der linearen Verzerrung $h[t]$ gefaltet und das Ergebnis mit einem, dazu unkorrelierten Rauschsignal $n[t]$ addiert.

Die Aufgabe besteht darin das verfälschte Signal $y[t]$ so zu verändern, daß die Veränderungen durch den Kanal wieder rückgängig gemacht werden. Es hat sich gezeigt, daß Spektralbereich und logarithmierter Spektralbereich dafür besonders geeignet sind.

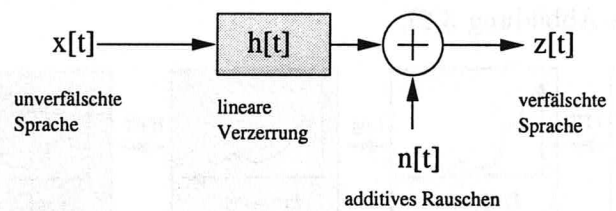


Abbildung 3.1: Die Modellierung des Übertragungskanals besteht aus der linearen Verzerrung und dem additiven Rauschen.

3.2 Spektralbereich

Im Spektralbereich wird die Faltung mit $h[t]$ zur Multiplikation mit der Übertragungsfunktion $|H(\omega)|^2$. Beim Übergang in den logarithmierten Spektralbereich ergibt sich ein additiver Zusammenhang zur linearen Verzerrung $\ln(|H(\omega)|^2)$ (siehe Formel 3.1).

Vorab einige Definitionen zur verwendeten Notation.

- $X(\omega)$ = Leistungsspektrum des unverfälschten Eingangssignals.
- $Y(\omega)$ = Leistungsspektrum des modellierten verfälschten Ausgangssignals.
- $N(\omega)$ = Leistungsspektrum des additiven Rauschanteils.
- $|H(\omega)|^2$ = Quadrat der Übertragungsfunktion der linearen Verzerrung.

Formel (3.1) zeigt das Kanalalmodell im Spektralbereich. Das durch den Übertragungskanal veränderte Ausgangssignal $Y(\omega)$ wird durch Multiplikation des Eingangssignals $X(\omega)$ mit der linearen Verzerrung $|H(\omega)|^2$ und anschließendem Addieren des Rauschanteils $N(\omega)$ modelliert.

$$Y(\omega) = X(\omega) \cdot |H(\omega)|^2 + N(\omega) \tag{3.1}$$

Das additive Rauschen kann auch vor der linearen Verzerrung zugefügt werden. Es kann gezeigt werden, daß auch dieser Fall durch Formel (3.1) modelliert wird. Unter der Voraussetzung, daß $|H(\omega)|^2$ an keiner Stelle Null wird spielt die Reihenfolge der beiden Operationen keine Rolle (3.2). In der Praxis kann von dieser Annahme ausgegangen werden.

$$X(\omega) \cdot |H(\omega)|^2 + N(\omega) = \left(X(\omega) + \frac{N(\omega)}{|H(\omega)|^2}\right) \cdot |H(\omega)|^2 \quad \forall\omega : |H(\omega)|^2 \neq 0 \tag{3.2}$$

3.3 Übergang zur logarithmierten Spektraldarstellung

Durch den Übergang zur logarithmierten Spektraldarstellung ergeben sich einige mathematische Vereinfachungen. Außerdem wird die Lautstärkeauflösung des menschlichen Gehörs besser modelliert [8]. Der am Institut verwendete Spracherkenner **Janus** und viele andere Spracherkenner arbeiten deshalb im Log-Spektralbereich oder im Cepstralbereich. In den Cepstralbereich gelangt man durch inverse Fouriertransformation des logarithmierten Spektrums (siehe Abbildung 3.2).

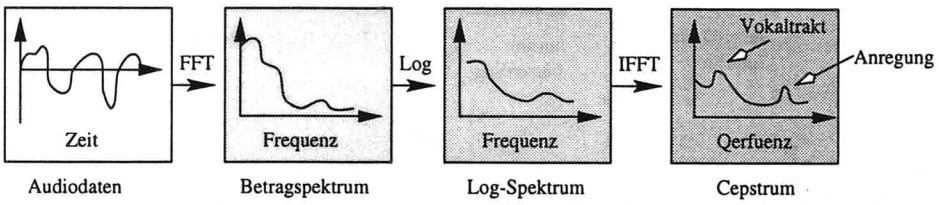


Abbildung 3.2: Der Zusammenhang zwischen Spektrum, Log-Spektrum und Cepstrum. Durch die Fouriertransformation der Audiodaten erfolgt der Übergang vom Zeitbereich in den Frequenzbereich. Um das logarithmierte Spektrum zu erhalten wird das Betragsspektrum logarithmiert. Das Cepstrum erhält man durch die inverse Fouriertransformation des Log-Spektrums. Im Cepstralbereich ist die Trennung von Anregung und Vokaltrakt besonders einfach.

Beim Übergang zur logarithmierten Spektraldarstellung erhalten wir folgende Kanaldarstellung:

$$\ln(Y(\omega)) = \ln(X(\omega) \cdot |H(\omega)|^2 + N(\omega)) \tag{3.3}$$

daraus folgt

$$\ln(Y(\omega)) = \ln\left(X(\omega)\cdot|H(\omega)|^2\cdot\left(1+\frac{N(\omega)}{X(\omega)\cdot|H(\omega)|^2}\right)\right) \tag{3.4}$$

oder

$$\ln(Y(\omega)) = \ln\left(N(\omega)\cdot\left(1+\frac{X(\omega)\cdot|H(\omega)|^2}{N(\omega)}\right)\right) \tag{3.5}$$

Um übersichtlichere Ausdrücke zu erhalten wendet man folgende Substitutionen an.

$$x = \ln(X(\omega)) \tag{3.6}$$
$$y = \ln(Y(\omega)) \tag{3.7}$$
$$n = \ln(N(\omega)) \tag{3.8}$$
$$q = \ln(|H(\omega)|^2) \tag{3.9}$$
$$\tag{3.10}$$

$$r(x,n,q) = \ln\left(1+\frac{N(\omega)}{X(\omega)\cdot|H(\omega)|^2}\right) \tag{3.11}$$

$$s(x,n,q) = \ln\left(1+\frac{X(\omega)\cdot|H(\omega)|^2}{N(\omega)}\right) \tag{3.12}$$

Dann folgt aus (3.4):

$$\boxed{\mathrm{y = x + q + r(x,n,q)}} \tag{3.13}$$

oder alternativ aus (3.5) folgt:

$$\boxed{\mathrm{y = n + s(x,n,q)}} \tag{3.14}$$

Beim Übergang in die logarithmierte Darstellung wird der Einfluß des Kanals in erster Näherung additiv modelliert (siehe Formel 3.13). **r** und **s** werden Korrekturvektoren genannt. Durch weiteres Umformen erhält man **r** und **s** in einer Darstellung die nur von Variablen im Log-Spektralbereich abhängt.

$$r(x,n,q) = \ln\left(1+\epsilon^{\ln\left(\frac{N(\omega)}{X(\omega)\cdot|H(\omega)|^2}\right)}\right) \tag{3.15}$$

$$r(x,n,q) = \ln\left(1+\epsilon^{\ln\left(N(\omega)\right)-\left(\ln(X(\omega))+\ln(|H(\omega)|^2)\right)}\right) \tag{3.16}$$

$$\boxed{r(x,n,q) = \ln\left(1+\epsilon^{n-x-q}\right)} \tag{3.17}$$

$$s(x,n,q) = \ln\left(1 + \epsilon^{\ln\left(\frac{X(\omega)\cdot|H(\omega)|^2}{N(\omega)}\right)}\right) \tag{3.18}$$

$$\boxed{s(x,n,q) = \ln\left(1+\epsilon^{x+q-n}\right)} \tag{3.19}$$

3.4 Eigenschaften der Korrekturvektoren

Sprache besteht aus Sprachpassagen und Pausen. Bei großem Signalrauschabstand gelten die folgenden Abschätzungen.

In Sprachpassagen gilt:

$$\boxed{r(x,n,q) \ll x+q} \tag{3.20}$$

In Pausen gilt:

$$\boxed{s(x,n,q) \ll n} \tag{3.21}$$

Dies kann als Motivation für die gewählte Kanaldarstellung durch zwei alternative Formeln (3.13) und (3.14) betrachtet werden. Durch den unterschiedlichen Einfluß der Korrekturvektoren in Sprachpassagen und Pausen ergeben sich die folgenden Näherungen.

In Sprachpassagen:

$$y \approx x+q \tag{3.22}$$

In Pausen:

$$y \approx n \tag{3.23}$$

Kapitel 4

Mittelwertsubtraktion im Log-Spektralbereich

Zur Kanalkompensation in der Sprachverarbeitung wurden im Laufe der Zeit viele Verfahren entwickelt [1]. Beim Übergang in den Log-Spektralbereich existiert ein elegantes und robustes Verfahren, die Mittelwertsubtraktion. Obwohl dieses Verfahren empirisch entwickelt wurde, hat es ausgezeichnete Leistungsmerkmale.

4.1 Analyse des Verfahrens

Im folgenden wird angenommen, daß die lineare Verzerrung innerhalb eines Sprachsegments konstant ist, also $q = \overline{q}$.

Zuerst erfolgen einige Definitionen zur verwendeten Notation.

- z_i = Log-Spektralvektor des Sprachrahmens Nummer i . Das Sprachsegment setzt sich aus den Sprachrahmen z_i zusammen. $i = 0, 1, ..., N-1$
- x_i = Log-Spektralvektor des zu z_i gehörenden unverfälschten Sprachrahmen.
- n = Log-Spektralvektor für das statischen additive Rauschsignal .
- q = Log-Spektralvektor der zeitinvarianten linearen Verzerrung .

Betrachtet man den Mittelwert $\overline{z}$ über alle Rahmen des Sprachsegments dann folgt mit (3.13):

$$\overline{z} = \frac{1}{N} \cdot \sum_{l=0}^{N-1} z_l = \frac{1}{N} \cdot \sum_{l=0}^{N-1} (x_l + q + r(x_l, n, q)) \tag{4.1}$$

q wird als zeitinvariant vorrausgesetzt

$$\overline{z} = q + \frac{1}{N} \cdot \sum_{l=0}^{N-1} x_l + \frac{1}{N} \cdot \sum_{l=0}^{N-1} r(x_l, n, q) \tag{4.2}$$

$$q = \overline{z} - \overline{x} - \overline{r(x, n, q)} \tag{4.3}$$

Durch Einsetzen von (4.3) in (3.13) ergibt sich:

$$z_i = x_i + r(x_i, n, q) + \overline{z} - \overline{x} - \overline{r(x, n, q)} \tag{4.4}$$

$$z_i - \overline{z} = x_i - \overline{x} + r(x_i, n, q) - \overline{r(x, n, q)} \tag{4.5}$$

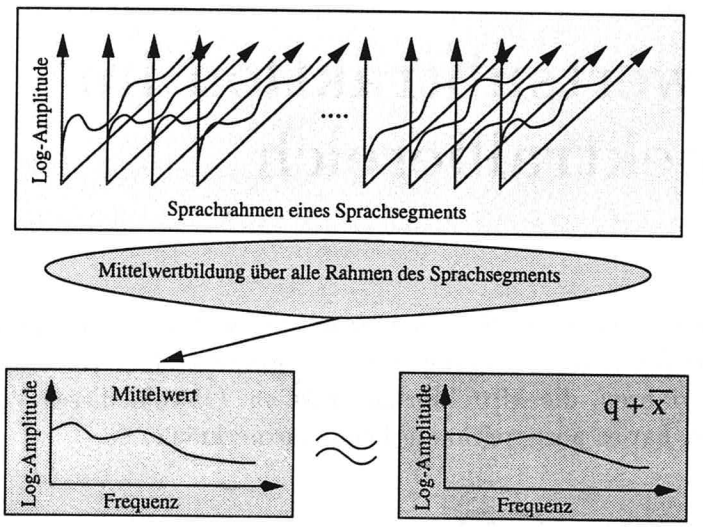


Abbildung 4.1: Der Mittelwert über alle Rahmen eines Sprachsegments entspricht in erster Näherung der linearen Verzerrung q plus dem Mittelwert der unverfälschten Sprachrahmen $\overline{x}$.

Formel (4.5) stellt den Vorgang der Mittelwertsubtraktion im Log-Spektralbereich dar. Bemerkenswert ist, daß die lineare Verzerrung q komplett entfernt wird. Allerdings wird dabei die Summe der Mittelwerte von x_i und $r(x_i, n, q)$ zusätzlich abgezogen. Der Mittelwert $\overline{x}$ sollte aus dem Sprachsignal entfernt werden, da er wenig zum Informationsgehalt beiträgt und vom Sprecher abhängt. Wenn der Signalrauschabstand groß ist, dann ist der Mittelwert der Korrekturvektoren $\overline{r(x, n, q)}$ klein. Wenn also ausreichend große Sprachsegmente mit großem Signalrauschabstand zur Mittelwertbildung herangezogen werden, dann entspricht der Mittelwert $\overline{z}$ in erster Näherung der lineare Verzerrung q plus dem Mittelwert der unverfälschten Sprachrahmen $\overline{x}$ (siehe Abbildung 4.1).

$$\overline{z} \approx q + \overline{x} \tag{4.6}$$

4.1.1 Pausenberücksichtigung

Beim Training und im späteren Betrieb sollte der Wert von $\overline{x}$ vergleichbar sein. Eine Voraussetzung dafür ist, daß die Anzahl der Rahmen über die gemittelt wird groß ist. Außerdem sollte das Verhältnis der Pausenrahmen zu den Sprachrahmen beim Training und beim Test gleich sein. Es ist offensichtlich, daß die zweite Bedingung in der Realität nicht erfüllt ist. Ein Ausweg stellt die getrennte Ermittlung der Mittelwerte $\overline{z_{Pause}}$ für Pausenrahmen und $\overline{z_{Sprache}}$ für Sprachrahmen dar.

Die in [5] gemachten Untersuchungen weisen darauf hin, daß die Berechnung von $\overline{z_{Sprache}}$ ausreicht. Die Subtraktion von $\overline{z_{Pause}}$ in Pausen ergab sogar durchweg höhere Fehlerraten als die Subtraktion von $\overline{z_{Sprache}}$. Probleme an den Wortkanten, wo Pausen in Sprache übergehen, sind vermutlich eine Ursache.

Da der Signalrauschabstand in Pausen klein ist, hat der Mittelwert der Korrekturvektoren $\overline{r(x,n,q)}$ einen großen Einfluß auf den Mittelwert $\overline{z_{Pause}}$ (Abschnitt 3.4). Er ist deshalb eine schlechtere Näherung für $q+\overline{x}$ als $\overline{z_{Sprache}}$ (4.7).

$$\boxed{\overline{z_{Sprache}} \approx q + \overline{x}} \tag{4.7}$$

4.2 Erkennungsversuche

Um das zuvor Gesagte zu belegen, wurden einige Versuche ausgeführt. Die Mittelwertsubtraktion wird bereits standardmäßig in vielen Systemen eingesetzt.

Zwei Verfahren wurden erprobt:

- Mittelwertsubtraktion ohne Pausenberücksichtigung ($z_i - \overline{z}$).
- Mittelwertsubtraktion mit Pausenberücksichtigung ($z_i - \overline{z_{Sprache}}$).

Die Versuche wurden mit einem Erkenner für kontinuierliche Sprache durchgeführt. Zur Pausenerkennung wurde ein einfaches, energiebasiertes Verfahren verwendet. Die Energie der Sprachrahmen wird ermittelt, logarithmiert, normalisiert, gefiltert und anschließend einer Schwellwerterkennung zugeführt.

Zum Training wurden mehr als 10000 Sprachsegmente verwendet. Der Test erfolgte mit 269 Sprachsegmenten aus der Standardtestmenge "*VM-Devtest 96*" [1].

	Mittelwertsubtraktion ohne Pausenberücksichtigung	Mittelwertsubtraktion mit Pausenberücksichtigung
Korrekt	84.8 %	85.0 %

Tabelle 4.1: Versuchsergebnisse der Mittelwertsubtraktion mit Pausenberücksichtigung.

[1] Die Development-Testmenge für die innerhalb des Verbundprojekts "*Verbmobil*" durchgeführten Evaluation 1996.

Kapitel 5

CDCN

Da wir uns im logarithmierten Spektralbereich bewegen, müßte es eigentlich CDLSN (Code word Dependent Log-Spectral Normalization) heißen. Die Unterschiede zum ursprünglichen CDCN (Codeword Dependent Cepstral Normalization) [1] sind jedoch so gering, daß im weiteren der Ausdruck CDCN verwendet wird.

Wie der Name schon sagt soll mit der Hilfe der Kodebuchvektoren das Eingangssignal normalisiert werden. Das Eingangssignal soll dabei so verändert werden, daß nur noch Informationen die kodebuchkonform sind erhalten bleiben. Speziell das additive Rauschen und die lineare Verzerrungen sollen eliminiert werden. Da hierfür kein explizites Lösungsverfahren existiert, wird in mehreren Iteration $\mathbf{r}$, $\mathbf{n}$, $\mathbf{q}$ und schließlich $\mathbf{x}$ geschätzt.

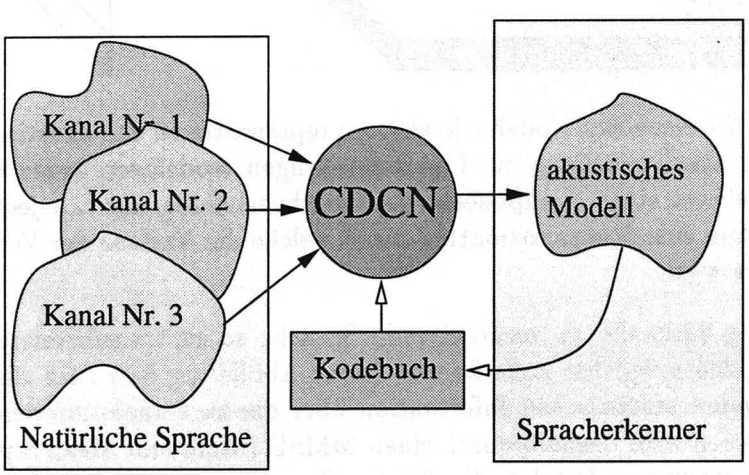


Abbildung 5.1: Durch das Verfahren wird Sprache auf das vom Kodebuch aufgespannte akustische Modell abgebildet. Dabei werden die Einflüsse unterschiedlicher Kanäle entfernt.

Die Vektoren $\mathbf{n}$ und $\mathbf{q}$ werden so geschätzt, daß die korrigierten Sprachrahmen näher an einem universellen akustischen Sprachmodell liegen. Dieses Modell wird durch ein Kodebuch repräsentiert. Das Sprachmodell wird dabei durch eine Mischverteilung aus Gaußverteilungen modelliert (siehe Formel 5.1). Die Verteilungskomponenten $N_x(c[k], C[k])$ werden während des Trainings des Spracherkenners durch die Trainingsdaten festgelegt. Jeder Kodebuchvektor $c[k]$ stellt den Mittelwert einer Komponente der Mischverteilung dar.

Jede Komponente besitzt eine Kovarianzmatrix $C[k]$, durch welche die Varianz der Verteilungskomponente modelliert wird (siehe Abbildung 5.2). Die Linearkombination der Verteilungskomponenten, mit den a priori Wahrscheinlichkeiten $P[k]$, repräsentiert das zum Kodebuch $\lambda = \{(c[0], C[0], P[0]), ..., (c[K-1], C[K-1], P[K-1])\}$ gehörende akustische Modell (siehe Formel 5.1). Die a priori Wahrscheinlichkeiten $P[k]$ ordnen dabei jedem Kodebuchvektor $c[k]$ eine beim Training berechnete Wahrscheinlichkeit zu.

$$p(x) = \sum_{k=0}^{K-1} P[k] \cdot N_x(c[k], C[k]) \tag{5.1}$$

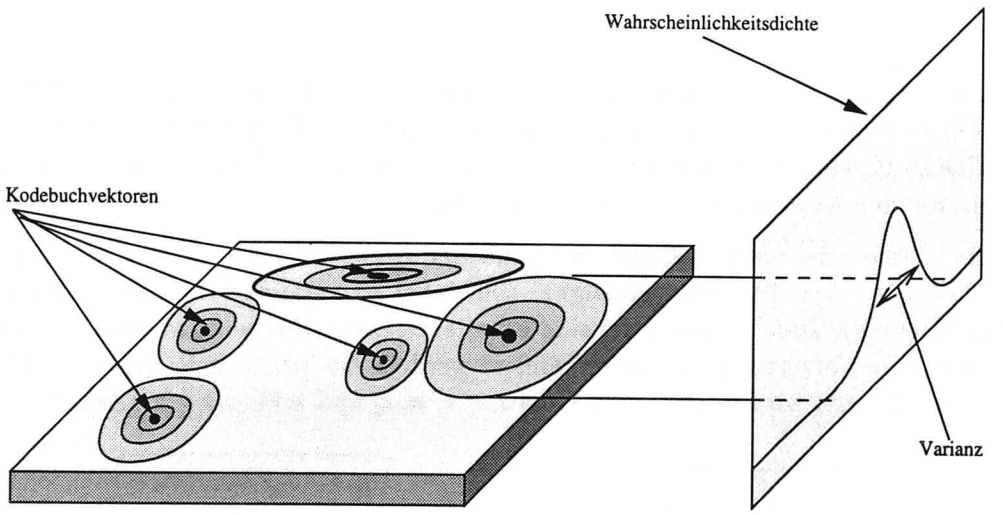


Abbildung 5.2: Die einzelnen Kodebuchvektoren repräsentieren das akustische Modell. Dies wird durch eine Mischverteilung aus Gaußverteilungen modelliert. Jeder Kodebuchvektor stellt den Mittelwert einer Komponente der Mischverteilung dar. Zu jeder Komponente existiert außerdem eine Kovarianzmatrix durch welche die Varianz der Verteilungskomponente modelliert wird.

Das Ziel besteht darin die zu verarbeitende Sprache so zu transformieren, daß sie dem universellen Modell möglichst nahe kommt (siehe Abbildung 5.1) . Im allgemeinen liegen keine hinreichenden statistischen Information über die zu transformierende Sprache vor. Die Transformation wird deshalb durch einen MMSE (Minimum Mean Square Error Estimation) Ansatz gewonnen. Die dazu benötigten Parameter $\mathbf{n}$, $\mathbf{q}$ und die Korrekturvektoren $\mathbf{r}(\mathbf{x}, \mathbf{n}, \mathbf{q})$ werden durch Schätzung gemäß ML (Maximum Likelihood) bestimmt. Als Basis der Schätzung dient der beobachtete Sprachrahmen und die im Kodebuch enthaltene a priori Information über die statistische Zusammensetzung von Sprache. Der Hauptvorteil von CDCN (Codeword Dependent Cepstral Normalization) liegt darin, daß die lineare Verzerrung $\mathbf{q}$ passend zum Kodebuch des Spracherkenners ermittelt wird. Zusätzlich wird das additive Rauschen durch die Korrekturvektoren in die Transformation eingebunden. Speziell in Situationen mit niedrigem Signal-Rauschabstand hat das additive Rauschen einen großen Einfluß auf das Eingangsignal. Ein weiterer Vorteil besteht darin, daß keine Langzeitinformationen benötigt werden. Alle Schätzungen basieren ausschließlich auf dem gerade beobachteten Sprachsegment. In der praktischen Anwendung ist dieser Umstand

erwünscht, um schnell auf sich verändernde Aufnahmebedingungen und Sprecher reagieren zu können (siehe Abbildung 5.3).

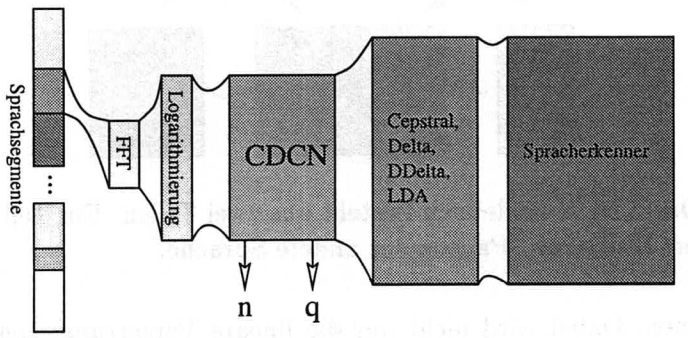


Abbildung 5.3: CDCN paßt nahtlos in die Vorverarbeitungsschritte des Spracherkenners. Zur Schätzung der lineare Verzerrung **q** und des additiven Rauschens werden ausschließlich Informationen aus dem aktuellen Sprachsegment benutzt.

Der Gebrauch von a priori Wissen über Sprachvektoren wurde gleichzeitig von [1] und [7] vorgeschlagen. Dabei wurde im Cepstralbereich bzw. im logarithmierten Spektralbereich gearbeitet.

Die hier beschriebene Version basiert hauptsächlich auf [1]. Es wurden jedoch eine Anzahl von Veränderungen vorgenommen. Es wird im Log-Spektralbereich gearbeitet, außerdem wird ein erweitertes Kodebuch verwendet. Der Log-Spektralbereich erlaubt eine einfache Überprüfung des Schätzwerts für die lineare Verzerrung $\hat{\mathbf{q}}$. Aus der Untersuchung der Mittelwertsubtraktion wissen wir, daß $\overline{z} \approx q + \overline{x}$ ist (4.6). Wenn die Kodebuchvektoren Mittelwertfrei sind kann $q^* = q + \overline{x} \approx \overline{z}$ zur Überprüfung des Schätzwerts verwendet werden. Das eigene Kodebuch macht das Verfahren unabhängig vom verwendeten Spracherkenner. Im Gegensatz zu [1] wurde die Anzahl der Kodebuchvektoren für Stille erhöht und der Algorithmus entsprechend angepaßt.

Da ein eigenes Kodebuch für CDCN (Codeword Dependent Cepstral Normalization) trainiert werden mußte, stellte sich die Frage wie die Signalvorverarbeitung dafür realisiert werden sollte. Das Ziel war reine, unverfälschte Sprache zum Training heranzuziehen. Nach den Erfahrungen mit der Mittelwertsubtraktion (Kapitel 4) und offensichtlichen Parallelen zwischen beiden Verfahren wurde die Mittelwertsubtraktion als Vorverabeitungsschritt beim Training verwendet. Die Wahl eines Startwertes für **n** und **q** beeinflußt in hohem Maße die Konvergenzgeschwindigkeit des iterativen ML (Maximum Likelihood) Verfahrens. Eine Anzahl von Versuchen zu diesem Thema führten zu einem Verfahren, das sehr gute Startwerte liefert.

5.1 Analyse des Verfahrens

Ein Spracherkenner sollte weitgehend unabhängig von der linearen Verzerrung und dem additiven Rauschen sein. Viele Verfahren erreichen dies durch Langzeitanalysen oder Mehrkanalaufnahmen.

CDCN (Codeword Dependent Cepstral Normalization) ermöglicht die Adaption auf eine neue Aufnahmesituation innerhalb eines Sprachsegments ohne Zusatzinformationen von

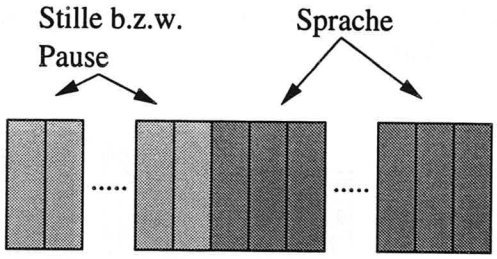


Abbildung 5.4: Das CDCN Kodebuch besteht aus zwei Teilen. Ein Teil der Kodebuchvektoren repräsentiert Stille bzw. Pausen der andere Sprache.

anderen Mikrofonen. Dabei wird nicht nur die lineare Verzerrung, sondern auch das additive Rauschen berücksichtigt. Alle Veränderungen finden ohne weiterführendes Wissen über Sprachsyntax und Semantik statt.

Zwei Hauptaufgaben sind dabei zu bewältigen:

1. Zu einem gegebenem Sprachsegment $\mathbf{Z}$ müssen die Parameter $\hat{\mathbf{q}}$ und $\hat{\mathbf{n}}$ so gewählt werden, daß $\hat{y}_i = x_i + \hat{q} + r(x_i, \hat{n}, \hat{q})$ im Mittel den beobachteten Sprachrahmen z_i annähert. Ein ML (Maximum Likelihood) Schätzer wird hierfür verwendet.

2. Nach der Schätzung von $\mathbf{n}$ und $\mathbf{q}$ muß für jeden Sprachrahmen z_i der unverfälschte Sprachvektor x_i bestimmt werden. Die Berechnung des Korrekturvektors $\mathbf{r}$ steht dabei im Mittelpunkt. Durch ein MMSE (Minimum Mean Square Error Estimation) Verfahren wird $\mathbf{r}$ so bestimmt, daß der quadratische Fehler minimiert wird.

5.1.1 Das MMSE Verfahren zur Bestimmung des unverfälschten Sprachsignals

Ohne explizite Informationen über den statistischen Aufbau der beobachteten Spracherahmen $\mathbf{z}$ muß Information durch passende Modellierung implizit zur Schätzung von $\mathbf{x}$ herangezogen werden. Die Dichtefunktion für $\mathbf{x}$ wird als Linearkombination von K gaußverteilten Dichten mit Mittelwert $c[k]$ Kovarianzmatrix $C[k]$ und Gewicht $P[k]$ modelliert (vergleiche (5.1)):

$$p(x) = \sum_{k=0}^{K-1} P[k] \cdot p(x|k) = \sum_{k=0}^{K-1} P[k] \cdot N_x(c[k], C[k]) \tag{5.2}$$

Weiterhin kann die Dichte der Wahrscheinlichkeit für unverfälschte Sprache $\mathbf{x}$ unter der Beobachtung $\mathbf{z}$ und der Annahme $\hat{\mathbf{n}}, \hat{\mathbf{q}}$ wie folgt ausgedrückt werden:

$$p(x|z, \hat{n}, \hat{q}) = \frac{\sum_{k=0}^{K-1} P[k] \cdot p(z|x, \hat{n}, \hat{q}, k) \cdot p(x|k)}{\sum_{k=0}^{K-1} P[k] \cdot \int p(z|x, \hat{n}, \hat{q}, k) \cdot p(x|k) \quad dx} \tag{5.3}$$

Diese a posteriori Wahrscheinlichkeit liefert alle Informationen die wir für einen MMSE (Minimum Mean Square Error Estimation) Schätzer benötigen. $p(x|k)$ ist die Wahrscheinlichkeitsdichte der k-ten Komponente der Linearkombination. $p(z|x,\hat{n},\hat{q},k)$ ist die Dichte der Verteilung $p(z|y,k)$. Der MMSE (Minimum Mean Square Error Estimation) Schätzer ergibt sich dann als:

$$x_{MMSE} = E[x|z,\hat{n},\hat{q}] = \frac{\sum_{k=0}^{K-1} P[k] \cdot \int x \cdot p(z|x,\hat{n},\hat{q},k) \cdot p(x|k) \quad dx}{\sum_{k=0}^{K-1} P[k] \cdot \int p(z|x,\hat{n},\hat{q},k) \cdot p(x|k) \quad dx} \tag{5.4}$$

Der wahre MMSE (Minimum Mean Square Error Estimation) Schätzwert für $\mathbf{x}$ kann wegen Nichtlinearitäten nicht berechnet werden. Da der Signalrauschabstand in Sprachpassagen groß und in Sprachpausen klein ist kann man folgende Näherung verwenden:

- Wir benutzen die Kodebuchvektoren für Pausen zur Bestimmung des additiven Rauschens $\hat{\mathbf{n}}$.
- Alle anderen Kodebuchvektoren werden zur Bestimmung von $\hat{\mathbf{q}}$ verwendet.

Für die Bestimmung des unverfälschten Sprachvektors $\hat{\mathbf{x}}_\mathbf{i}$ erhält man nach längerer Umformung [1] den folgenden Ausdruck:

$$\boxed{\hat{x}_i = \sum_{k=Sil}^{K-1} f[i,k] \cdot (z_i - \hat{q} - r(c[k],\hat{n},\hat{q}))} \tag{5.5}$$

Die a posteriori Wahrscheinlichkeit $f[i,k]$ wird durch folgende Gleichungen berechnet:

$$f[i,k] = \frac{\frac{P[k]}{|C[k]|^{1/2}} \cdot \epsilon^{-\frac{d[i,k]}{2}}}{\sum_{l=0}^{K-1} \frac{P[l]}{|C[l]|^{1/2}} \cdot \epsilon^{-\frac{d[i,l]}{2}}} \tag{5.6}$$

$$d[i,k] = \epsilon^T[i,k] \cdot C^{-1}[k] \cdot \epsilon[i,k] \tag{5.7}$$

$$\epsilon[i,k] = z_i - \hat{q} - r(c[k],\hat{n},\hat{q}) - c[k] \tag{5.8}$$

5.1.2 Das ML Verfahren zur Bestimmung von n und q

Ausgehend von einer Anzahl $\mathbf{N}$ von Sprachrahmen z_i werden nun das additive Rauschen $\hat{\mathbf{n}}$ und die lineare Verzerrung $\hat{\mathbf{q}}$ bestimmt. Die einzigen zur Verfügung stehenden Informationen sind das im Kodebuch $\lambda = \{(c[0],C[0],P[0]),...,(c[K-1],C[K-1],P[K-1])\}$ repräsentierte Sprachmodell und das aktuell bearbeitete Sprachsegment $\mathbf{Z}$.

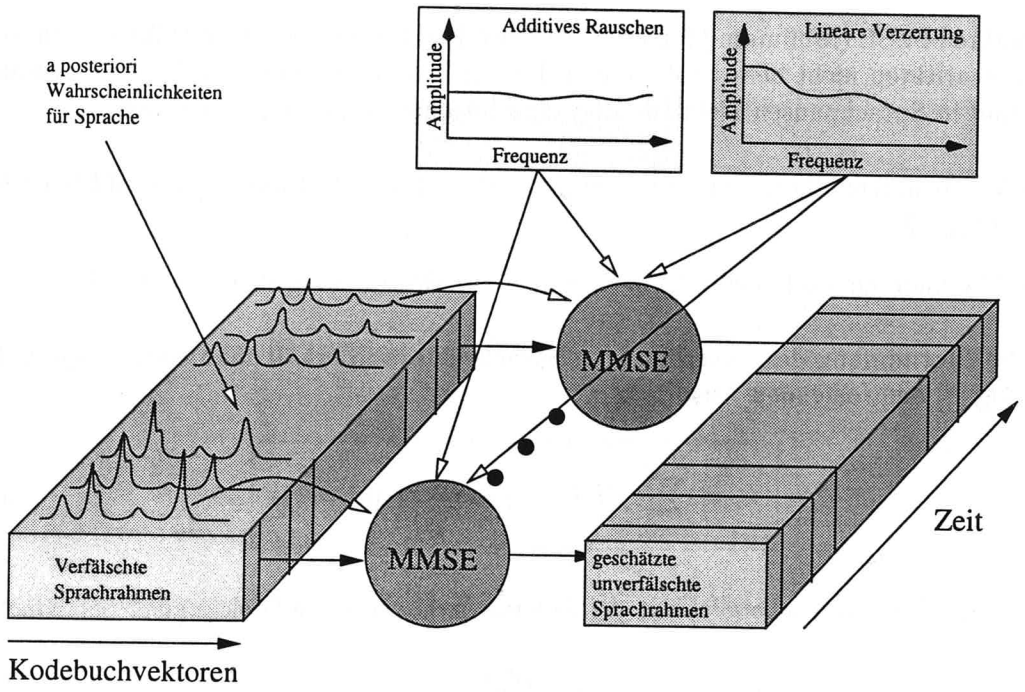


Abbildung 5.5: Bei gegebenem $\hat{\mathbf{n}}$ und $\hat{\mathbf{q}}$ werden die Sprachrahmen für unverfälschte Sprache mit Hilfe der a posteriori Wahrscheinlichkeiten des Kodebuches für Sprache geschätzt. Von jedem verfälschten Sprachrahmen wird dabei die lineare Verzerrung $\hat{\mathbf{q}}$ und die Linearkombination der Korrekturvektoren $r[k] = r(c[k], \hat{n}, \hat{q})$ abgezogen.

Da keine a priori Information über die Gestalt von $\mathbf{n}$ und $\mathbf{q}$ vorhanden ist, wird ein ML (Maximum Likelihood) Verfahren angewandt. Die Parameter $\hat{\mathbf{n}}$ und $\hat{\mathbf{q}}$ werden dabei so gewählt, daß die Wahrscheinlichkeit $p(Z|\hat{q}, \hat{n}, \lambda)$ maximal ist.

Durch die Annahme, daß unterschiedliche Sprachrahmen z_i unabhängig sind, kann man folgenden Ausdruck benutzen:

$$\ln(p(Z|n,q)) = \sum_{i=0}^{N-1} \ln(p(z_i|n,q)) \tag{5.9}$$

Die Maximierung für das additive Rauschen führt zu folgendem Ausdruck:

$$\frac{\partial \ln(p(Z|n,q))}{\partial n} = \sum_{i=0}^{N-1} \left(\frac{1}{p(z_i|n,q)} \cdot \frac{\partial p(z_i|n,q)}{\partial n} \right) = 0 \tag{5.10}$$

Ein ähnlicher Ausdruck existiert für die lineare Verzerrung $\mathbf{q}$. Nun werden die in Abschnitt 3.4 eingeführten Näherungen verwendet:

$$p(z_i|n,q) = \alpha \cdot \left(\sum_{k=0}^{K_{Sil}} \frac{P[k]}{|C[k]|^{1/2}} \cdot \epsilon^{-\frac{d_{Silence}[i,k]}{2}} + \sum_{k=K_{Sil}}^{K-1} \frac{P[k]}{|C[k]|^{1/2}} \cdot \epsilon^{-\frac{d_{Speech}[i,k]}{2}} \right) \tag{5.11}$$

$$d_{Speech}[i,k] = \epsilon^T_{Speech}[i,k] \cdot C^{-1}[k] \cdot \epsilon_{Speech}[i,k] \tag{5.12}$$

$$\epsilon_{Speech}[i,k] = z_i - \underbrace{q - r[k] - c[k]}_{y} \quad : r \ll q + x \quad \text{(in Sprachpassagen)} \tag{5.13}$$

$$d_{Silence}[i,k] = \epsilon^T_{Silence}[i,k] \cdot C^{-1}[k] \cdot \epsilon_{Silence}[i,k] \tag{5.14}$$

$$\epsilon_{Silence}[i,k] = z_i - \underbrace{n - s[k]}_{y} \quad : s \ll n \quad \text{(in Pausen)} \tag{5.15}$$

Der erste Summand in (5.11) hängt in erster Linie von $\mathbf{n}$ ab, weil die Kodebuchvektoren für Pausen bzw. Stille unempfindlich gegen $\mathbf{q}$ sind. Der zweite hängt nur von $\mathbf{q}$ ab weil die Kodebuchvektoren für Sprache unempfindlich gegen $\mathbf{n}$ sind. Die ersten K_{Sil} Kodebuchvektoren sind hierbei für Pausen bzw. Stille zuständig die restlichen $K - K_{Sil}$ für Sprache.

Weil (5.10) zu einer hochgradig nichtlinearen Gleichung führt, wird ein Iterationsverfahren zur Approximation von $\mathbf{n}$ und $\mathbf{q}$ verwendet. Der EM (Estimate Maximize) Algorithmus ist ein iteratives Verfahren zur Lösung von ML (Maximum Likelihood) Problemen mit unvollständigen Daten.

1. Wähle Startwerte für $\hat{n}_{(0)}$ und $\hat{q}_{(0)}$. Im einfachsten Fall 0. Es ist jedoch von Vorteil einen besseren Startwert zu wählen um die Konvergenz des Verfahrens zu beschleunigen.

2. Berechne die Korrekturvektoren $r_{(j)}[k]$ und $s_{(j)}[k]$ aus $\hat{n}_{(j-1)}$ und $\hat{q}_{(j-1)}$ und $x = c[k]$.

3. Maximiere die logarithmierte Wahrscheinlichkeit von $\ln(p(Z|\hat{n}, \hat{q}))$. Die neuen Schätzwerte für $\hat{n}_{(j)}$ und $\hat{q}_{(j)}$ sind:

$$\hat{n}_{(j)} = \frac{\sum_{i=0}^{N-1} \sum_{k=0}^{K_{Sil}-1} f_{(j)}[i,k] \cdot (z_i - s_{(j)}[k])}{\sum_{i=0}^{N-1} \sum_{k=0}^{K_{Sil}-1} f_{(j)}[i,k]} \tag{5.16}$$

$$\hat{q}_{(j)} = \frac{\sum_{i=0}^{N-1} \sum_{k=K_{Sil}}^{K-1} f_{(j)}[i,k] \cdot (z_i - c[k] - r_{(j)}[k])}{\sum_{i=0}^{N-1} \sum_{k=K_{Sil}}^{K-1} f_{(j)}[i,k]} \tag{5.17}$$

4. Ende wenn Konvergenz erreicht wurde. Sonst weiter mit 2.

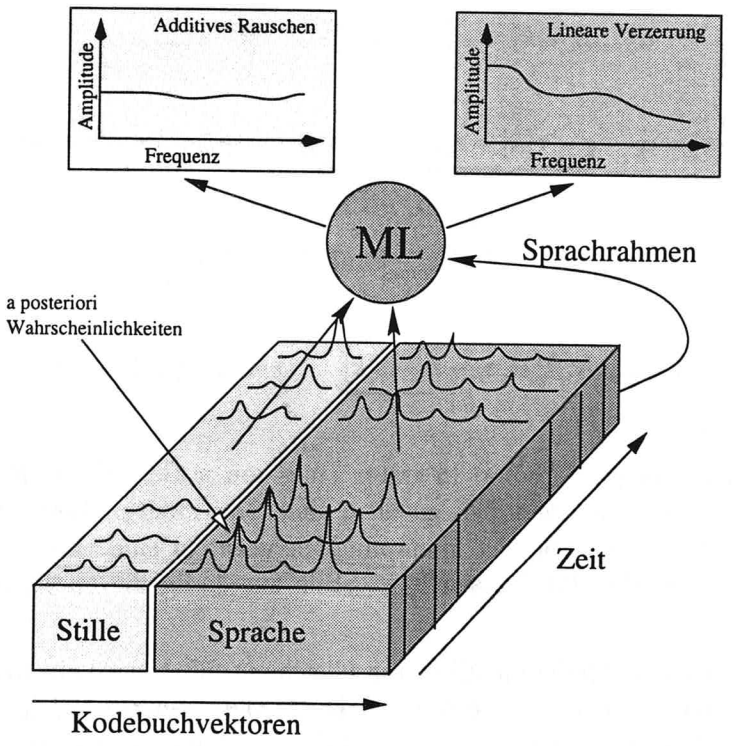


Abbildung 5.6: CDCN schätzt mit den a posteriori Wahrscheinlichkeiten für Sprache die lineare Verzerrung **q**. Das additive Rauschen **n** wird mit den a posteriori Wahrscheinlichkeiten für Stille geschätzt. Die a posteriori Wahrscheinlichkeiten werden für jeden Sprachrahmen eines Sprachsegmentes anhand der beiden Kodebücher berechnet.

5.2 Der CDCN-Algorithmus

Zuerst einige Definitionen zur verwendeten Notation.

- N = Anzahl der Sprachrahmen z_i im Sprachsegment Z.
- K = Anzahl der Kodebuchvektoren $c[0], ..., c[K-1]$.
- K_{Sil} = Anzahl der Kodebuchvektoren, die Stille repräsentieren $c[0], ..., c[K_{Sil}-1]$.
- z_i = Log-Spektralvektor des Sprachrahmens Nr.i $\quad i = 0, 1, ..., N-1$.
- $c[k]$ = Kodebuchvektor $\quad k = 0, 1, ..., K-1$.
- $\hat{n}_{(j)}$ = Log-Spektralvektor für das statische additiven Rauschsignal (wird als konstant innerhalb eines Sprachsegments angenommen).
- $\hat{q}_{(j)}$ = Log-Spektralvektor der zeitinvarianten linearen Verzerrung.
- $r_{(j)}[k]$ = Korrekturvektor für die spätere Linearkombination $\quad k = 0, 1, ..., K-1$.
- $f_{(j)}[i,k]$ = a posteriori Wahrscheinlichkeit der k-ten Gaußverteilung. $\quad k = 0, 1, ..., K-1$ und $i = 0, 1, ..., N-1$.
- $P[k]$ = a priori Wahrscheinlichkeit mit der der Kodebuchvektor k auftritt (wird während des Trainings ermittelt).
- $C[k]$ = Kovarianzmatrix zum Kodebuchvektor k.
- Außerdem bezeichnet der Buchstabe **j** die Iteration also $\hat{q}_{(j)}$ bezeichnet $\hat{q}$ in der Iteration Nr.j.

5.2.1 Schätzung der Parameter n und q mittels ML (Maximum Likelihood) Ansatz

1. Wähle Startwerte für $\hat{n}_{(0)}$ und $\hat{q}_{(0)}$. Eine geeignete Wahl ist $\hat{n}_{(0)} = 0$ und $\hat{q}_{(0)} = \overline{z} = \frac{1}{N} \cdot \sum_{i=0}^{N-1} z_i$.

2. Zuerst werden die Korrekturvektoren **s** bzw. **r** bestimmt (3.17).

$$r_{(j)}[k] = \ln\left(1 + e^{\left(\hat{n}_{(j-1)} - \hat{q}_{(j-1)} - c[k]\right)}\right) \tag{5.18}$$

$$s_{(j)}[k] = \ln\left(1 + e^{\left(c[k] + \hat{q}_{(j-1)} - \hat{n}_{(j-1)}\right)}\right) \tag{5.19}$$

3. Nun werden die a posteriori Wahrscheinlichkeiten berechnet. $f[i,k]$ entspricht der Wahrscheinlichkeit $p(k|z_i, \hat{q}_{(j-1)}, \hat{n}_{(j-1)})$, daß der Rahmen z_i den Kodebuchvektor

$c[k]$ als unverfälschten Sprachrahmen x_i enthält.

$$f_{(j)}[i,k] = \frac{\frac{P[k]}{|C[k]|^{1/2}} \cdot \epsilon^{-\frac{d_{(j)}[i,k]}{2}}}{\sum_{l=0}^{K-1} \frac{P[l]}{|C[l]|^{1/2}} \cdot \epsilon^{-\frac{d_{(j)}[i,l]}{2}}} \tag{5.20}$$

$$d_{(j)}[i,k] = \epsilon_{(j)}^T[i,k] \cdot C^{-1}[k] \cdot \epsilon_{(j)}[i,k] \tag{5.21}$$

$$\epsilon_{(j)}[i,k] = z_i - \hat{q}_{(j-1)} - r_{(j)}[k] - c[k] \tag{5.22}$$

4. Das additive Rauschen $\mathbf{n}$ wird neu geschätzt. Dabei hilft das a priori Wissen, daß der Signalrauschabstand in Passagen mit Stille klein ist. Deshalb wird $\hat{\mathbf{n}}$ durch die a posteriori Wahrscheinlichkeiten des Kodebuchs für Stille neu berechnet. Zusätzlich wird die Annahme gemacht, daß bei Stille $s[k] \approx 0$ ist (3.19). Mit (3.14) folgt dann $n \approx y_i$:

$$\hat{n}_{(j)} = \frac{\sum_{i=0}^{N-1} \sum_{k=0}^{K_{Sil}-1} f_{(j)}[i,k] \cdot (z_i)}{\sum_{i=0}^{N-1} \sum_{k=0}^{K_{Sil}-1} f_{(j)}[i,k]} \tag{5.23}$$

5. Die Berechnung der linearen Verzerrung $\hat{\mathbf{q}}$ basiert auf den a posteriori Wahrscheinlichkeiten für Rahmen in Sprachpassagen. In Sprachpassagen ist der Signalrauschabstand groß und der Schätzwert für $r[k]$ ist gut.

$$\hat{q}_{(j)} = \frac{\sum_{i=0}^{N-1} \sum_{k=K_{Sil}}^{K-1} f_{(j)}[i,k] \cdot (z_i - c[k] - r_{(j)}[k])}{\sum_{i=0}^{N-1} \sum_{k=K_{Sil}}^{K-1} f_{(j)}[i,k]} \tag{5.24}$$

6. Ende wenn $\hat{\mathbf{n}}$ und $\hat{\mathbf{q}}$ konvergieren. Sonst weiter mit 2.

5.2.2 Berechnung des unverfälschten Log-Spektralvektor $\hat{x}_i$ durch MMSE

Nach Abschnitt 5.1.1:

$$\hat{x}_i = z_i - \hat{q} - \sum_{k=K_{Sil}}^{K-1} f[i,k] \cdot r[k] \qquad i = 0, 1, ..., N-1 \tag{5.25}$$

Kapitel 6

Implementierung

Nun folgen einige Überlegungen zur Wahl der Startwerte für $\hat{n}$ und $\hat{q}$. Die Verringerung der von CDCN benötigten Rechenzeit steht dabei im Mittelpunkt. Im Anschluß wird die Implementierung von CDCN (Codeword Dependent Cepstral Normalization) unter **Janus** gezeigt. Den Abschluß des Kapitels bilden einige Versuche, die erste Einblicke in die Arbeitsweise des Verfahrens zeigen sollen.

6.1 Bestimmung der Startwerte für $\hat{n}$ und $\hat{q}$

Da die Konvergenz des Verfahrens garantiert ist [1], können die Startwerte für $\hat{n}$ und $\hat{q}$ beliebig gewählt werden. Es hat sich jedoch gezeigt, daß die Wahl der Startwerte einen großen Einfluß auf die Konvergenzgeschwindigkeit hat. Weiterhin hängt die Konvergenzgeschwindigkeit stark von den verwendeten Sprachsegmenten ab. Für Sprachsegment aus der Kodebuchtrainingsmenge konvergiert das Verfahren schneller als für Sprachsegmente, die auf eine andere Art gewonnen wurden. Weiterhin beeinflußt der Signalrauschabstand die Konvergenzgeschwindigkeit. Ein niedriger Signalrauschabstand verringert die Konvergenzgeschwindigkeit. Bei ungünstiger Wahl der Startwerte kann die Anzahl der nötigen Iterationen sehr groß werden.

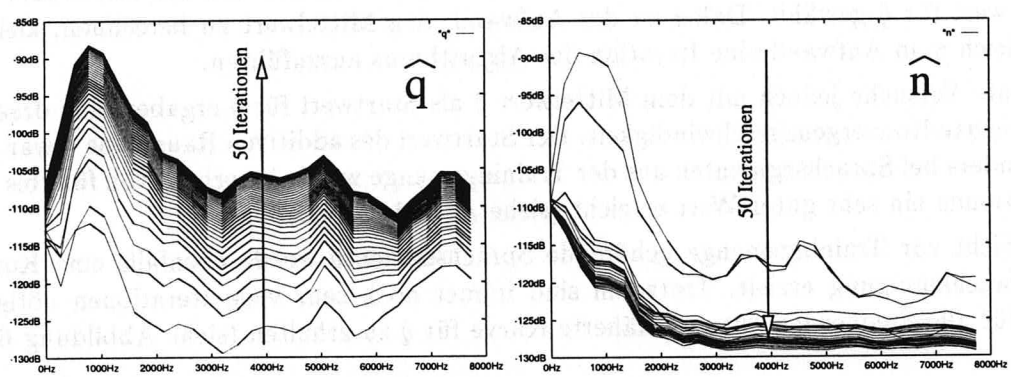


Abbildung 6.1: Zwischenwerte während 50 Iterationen mit einem Sprachsegment aus der Trainingsmenge. Die linken Kurven zeigen die lineare Verzerrung $\hat{q}$ die rechten Kurven zeigen das additive Rauschen $\hat{n}$.

Abbildung 6.1 zeigt $\hat{q}$ und $\hat{n}$ für ein Sprachsegment aus der Trainingsmenge. Es wurden 50 Iteration ausgeführt. Nach zuerst sehr rascher Annäherung werden immer kleinere Schritte gemacht. Ein brauchbarer Wert für $\hat{q}$ wird erst nach mehr als 30 Iterationen erreicht. Als Startwert wurde für $\hat{n}$ und $\hat{q}$ Null gewählt. In Abbildung 6.2 werden die Ergebnisse für ein Sprachsegment, welches nicht aus der Kodebuchtrainingsmenge stammt, gezeigt. Nach ungefähr 30 bis 40 Iterationen ist ein verwertbarer Endzustand erreicht.

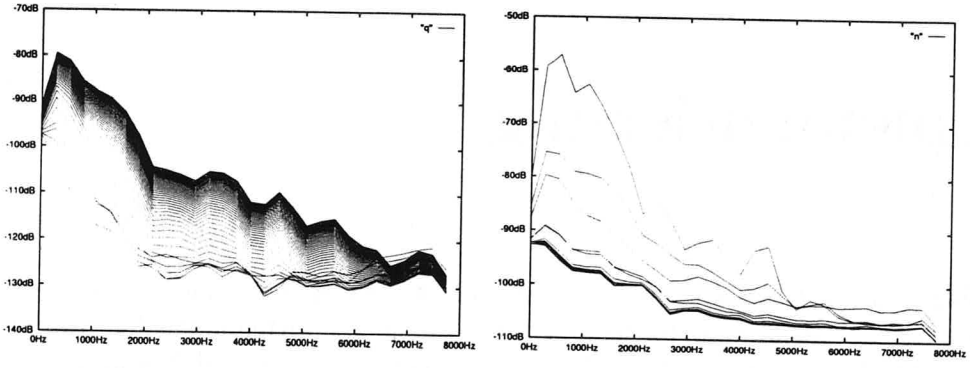


Abbildung 6.2: Ein Sprachsegment welches nicht aus der Trainingsmenge stammte. Erst nach über 30 Iterationen erreicht $\hat{q}$ einen akzeptablen Wert.

6.1.1 Der Mittelwert als Startwert für $\hat{q}$

Die Wahl der Startwerte sollte so erfolgen, daß mit möglichst wenigen Iterationen ein brauchbarer Wert für $\hat{n}$ und $\hat{q}$ erreicht wird. Brauchbarer Wert bedeutet in diesem Zusammenhang, daß der Abstand zum tatsächlichen Grenzwert der Iteration möglichst klein ist.

In [1] wurde $\hat{n}$ durch eine Vorberechnung bestimmt und $\hat{q}$ zu Null gesetzt. Wir werden hier jedoch ein anderes Verfahren benutzen. Die Ergebnisse aus dem Abschnitt zur Mittelwertsubtraktion bilden dazu die Grundlage. Zuerst beschränken wir uns deshalb auf die Wahl der linearen Verzerrung. Wir wählen $\hat{n} = \vec{0}$. Zur Bestimmung von $\hat{q}$ ziehen wir die Ergebnisse aus der Mittelwertsubtraktion heran. Dort wurde gezeigt, daß $\overline{z}$ als erste Näherung für $q + \overline{x}$ betrachtet werden kann (siehe Formel 4.6). Deshalb wird der Mittelwert $\overline{z}$ als Startwert für $\hat{q}$ gewählt. Dabei ist der Aufwand, den Mittelwert zu berechnen, klein im Vergleich zum Aufwand eine Iteration des Algorithmus auszuführen.

Erneute Versuche jedoch mit dem Mittelwert $\overline{z}$ als Startwert für $\hat{q}$ ergaben eine drastisch gesteigerte Konvergenzgeschwindigkeit. Der Startwert des additiven Rauschens $\hat{n}$ war Null. Besonders bei Sprachsegmenten aus der Trainingsmenge wurde innerhalb von fünf bis zehn Iterationen ein sehr guter Wert erreicht (siehe Abbildung 6.3).

Für nicht zur Trainingsmenge gehörende Sprachsegmente wurde ebenfalls eine Konvergenzbeschleunigung erzielt. Trotzdem sind immer noch sehr viele Iterationen nötig um eine für alle Frequenzen gut angenäherte Kurve für $\hat{q}$ zu erhalten (siehe Abbildung 6.4).

6.1.2 Zweistufenberechnung des Startwertes für $\hat{n}$

Aus den vorangegangenen Versuchen steht nun der Mittelwert $\overline{z}$ des Sprachsegments als effizienter Startwert für die lineare Verzerrung $\hat{q}$ zur Verfügung. Seine Berechnung ist

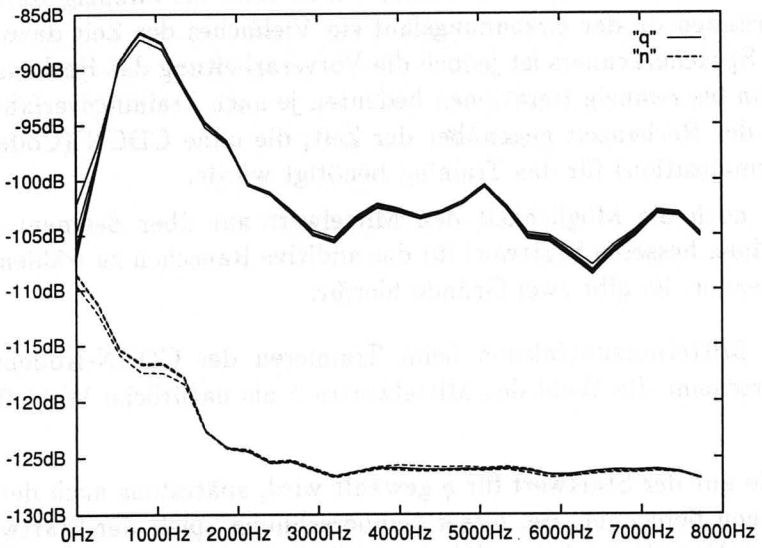


Abbildung 6.3: Durch die Wahl des Mittelwertes $\overline{z}$ als Startwert für $\hat{q}$ wurde eine erhebliche Konvergenzbeschleunigung erreicht. Die Kurven wurden wiederum für ein Sprachsegment der Trainingsmenge erstellt. Der Startwert von $\hat{n}$ war Null und es werden 50 Iterationen dargestellt.

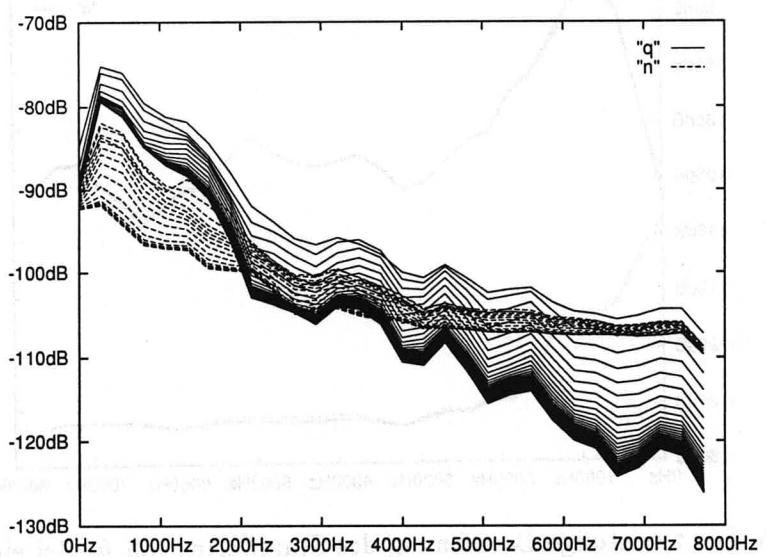


Abbildung 6.4: Ein Sprachsegment welches nicht aus der Trainingsmenge stammte. Nach zehn Iterationen werden schon sehr gute Werte erreicht. Es werden wiederum 50 Iterationen gezeigt, wobei der Startwert für $\hat{n}$ Null und für $\hat{q}$ der Mittelwert $\overline{z}$ war.

einfach und die erreichte Verbesserung der Konvergenzgeschwindigkeit signifikant.

Beim späteren Einsatz im Spracherkenner wären zehn bis zwanzig Iterationen durchaus noch zu begründen da der Erkennungslauf ein Vielfaches der Zeit davon benötigt. Beim Training des Spracherkenners ist jedoch die Vorverarbeitung das Rechenzeit bestimmende Moment. Zehn bis zwanzig Iterationen bedeuten je nach Trainingsverfahren fast eine Verzehnfachung der Rechenzeit gegenüber der Zeit, die ohne CDCN (Codeword Dependent Cepstral Normalization) für das Training benötigt würde.

Es bestünde noch die Möglichkeit den Mittelwert nur über Segmente mit Sprache zu berechnen. Einen besseren Startwert für das additive Rauschen zu wählen erscheint jedoch vielversprechender. Es gibt zwei Gründe hierfür:

1. Da die Mittelwersubtraktion beim Trainieren der CDCN-Kodebücher verwendet wird, erscheint die Wahl des Mittelwertes $\overline{z}$ als natürliche Wahl für den Startwert von $\hat{q}$.

2. Egal wie gut der Startwert für $\hat{q}$ gewählt wird, spätestens nach der ersten Iteration wird $\hat{q}$ neu berechnet. Bei dieser Neuberechnung spielt der Startwert von $\hat{n}$ in der Form des Korrekturvektors $r[k]$ eine große Rolle.

Um das System nicht negativ zu beeinflussen, wäre eine Wahl von $\hat{n}$ geeignet, bei der die im Algorithmus benutzten Verfahrensschritte benutzt werden. Der geradlinigste Weg wäre $\hat{n}$ einfach über die a posteriori Wahrscheinlichkeiten $f[i,k]$ mit (5.23) zu bestimmen. Das Berechnen der a posteriori Wahrscheinlichkeiten benötigt den Hauptteil der Rechenzeit im Algorithmus. Deshalb wurde ein zweistufiges Verfahren entwickelt.

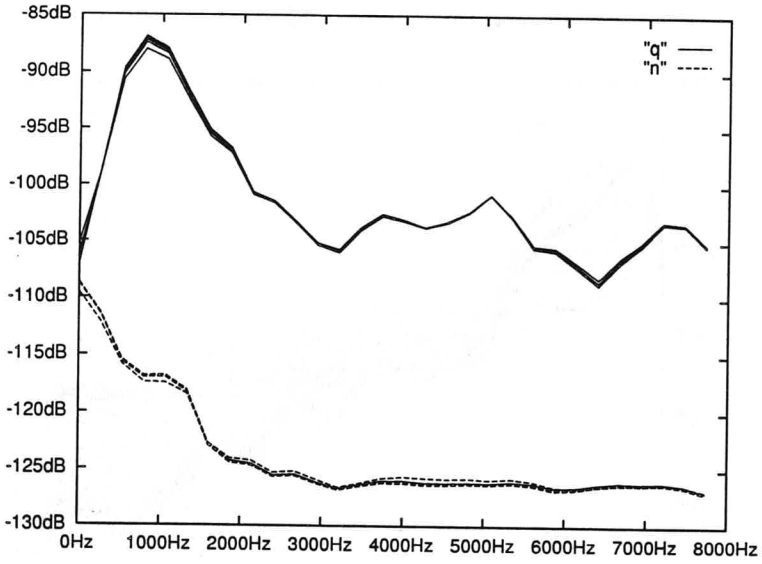


Abbildung 6.5: Die zweistufige Berechnung des Startwertes von $\hat{n}$. Bei einem Sprachsegment aus der Trainingsmenge konvergiert das Verfahren schon nach drei bis vier Schritten.

1. Der Startwert für $\hat{\mathbf{q}}$ ist der Mittelwert $\overline{z}$, der Startwert für $\hat{\mathbf{n}}$ ist Null.

2. - Die Korrekturvektoren $r[k]$, die a posteriori Wahrscheinlichkeiten $f[i,k]$ und $\hat{\mathbf{n}}$ werden berechnet.

- Abweichend vom Algorithmus werden in der ersten Iteration an dieser Stelle die Korrekturvektoren $r[k]$ neu berechnet. Dies geschieht vor der Berechnung von $\hat{\mathbf{q}}$. Damit fließt der neue Wert von $\hat{\mathbf{n}}$ direkt in die Berechnung von $\hat{\mathbf{q}}$ ein.
- Alle weiteren Iteration werden ohne Änderung ausgeführt.

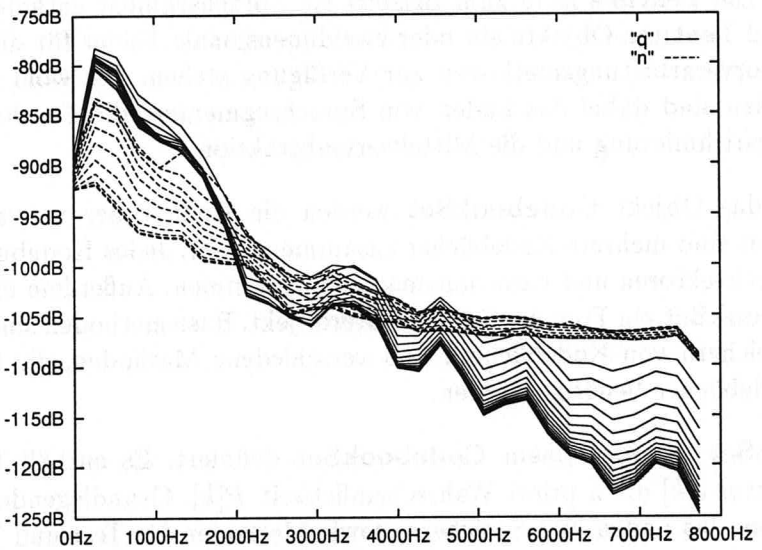


Abbildung 6.6: Fünfzehn Iterationen mit einem Sprachsegment welches nicht zur Trainingsmenge gehörte. Erneut ist eine Verbesserung in der Kovergezgeschwindigkeit zu sehen.

Durch das Neuberechnen der Korrekturvektoren $r[k]$ in der ersten Iteration fließt der neue Schätzwert von $\hat{\mathbf{n}}$ sofort in die Berechnung von $\hat{\mathbf{q}}$ ein. Die garantierte Konvergenz des Algorithmus wird durch diese Änderung nicht beeinflußt, da sie nur in der ersten Iteration stattfindet.

Bei Sprachsegmenten aus der Trainingsmenge erhält man dadurch nach drei bis vier Iterationen fast optimale Werte (siehe Abbildung 6.5). Die Konvergenzgeschwindigkeit bei nicht zur Trainingsmenge gehörenden Sprachsegmenten ist ebenfalls angestiegen (siehe Abbildung 6.6) .

Es besteht nun die Möglichkeit beim Training des Spracherkenners mit sechs bis zehn Iterationen zu arbeiten. Voraussetzung ist, daß ausschließlich Sprachsegmente aus der Trainingsmenge des CDCN-Kodebuchs benutzt werden. Im späteren Betrieb kann dann mit zehn bis dreißig Iteration gearbeitet werden, was eine fast zu vernachlässigende Verlangsamung zwischen ein und fünf Prozent bedeutet.

6.2 Realisierung von CDCN unter Janus

Der Algorithmus wurde in C programmiert. Weiterhin wurden einige Skripten zur Initialisierung in **TCL/TK** erstellt.

Janus besitzt eine Objektstruktur in die CDCN (Codeword Dependent Cepstral Normalization) als Methode eingebettet wurde. Das ausgewählte Mutterobjekt war durch die

benötigte Funktionalität fast vollständig bestimmt. Vorab eine kurze Aufzählung der in diesem Zusammenhang benutzten Objekte:

- **FeatureSet** ist das Basisobjekt zur Sprachsegmentverarbeitung. Wie der Name schon sagt sind in diesem Objekt eine Menge von **Feature**-Objekten zusammengefaßt. Ein **Feature** kann zum Beispiel ein Sprachsegment enthalten. Im allgemeinen sind **Feature**-Objekte ein oder zweidimensionale Felder für die eine Reihe von Signalvorverarbeitungsmethoden zur Verfügung stehen. Die wohl gebräuchlichsten Methoden sind dabei das Laden von Sprachsegmenten, die Fouriertransformation, die Logarithmierung und die Mittelwertsubtraktion.

- Durch das Objekt **CodebookSet** werden die Kodebücher verwaltet. Im **CodebookSet** sind mehrere Kodebücher zusammengefaßt. Jedes Kodebuch setzt sich aus Kodebuchvektoren und Kovarianzmatrizen zusammen. Außerdem existiert zu jedem **CodebookSet** ein **FeatureSet** als Unterobjekt. Basismethoden sind das Laden und das Speichern von Kodebüchern und verschiedene Methoden, die für das Training der Kodebücher benötigt werden.

- **DistribSet** ist über einem **CodebookSet** definiert. Es enthält für jeden Kodebuchvektor $c[k]$ die a priori Wahrscheinlichkeit $P[k]$. Grundlegende Methoden sind wiederum das Laden und Speichern, sowie Methoden für Test und Training.

Zwei Mutterobjekte standen zur Wahl. Als Methode die **Feature**-Objekte verarbeitet wäre **FeatureSet** als Objekt geeignet gewesen. Das benötigte Kodebuch und seine Funktionalität hätten dann jedoch umständlich realisiert werden müssen. Deshalb kam nur noch **DistribSet** als Objekt in Frage (siehe Abbildung 6.7). Darin sind alle Informationen über a priori Wahrscheinlichkeiten und Kodebücher als Unterobjekte enthalten. Es mag zwar auf den ersten Blick ungewohnt erscheinen, ein so komplexes Objekt in der Signalvorverarbeitung einzusetzen, jedoch hätte jede andere Realisierung das explizite und nichgekapselte Mitführen der Kodebücher bedeutet. Daß Informationen aus höheren Schichten mit einfließen ist ja gerade eine herausragenden Eigenschaft von CDCN. Als Methode von **DistribSet** konnte dank der eigenen Kodebücher eine kompakte und gut gekapselte Systemerweiterung geschaffen werden.

6.2.1 Datenstrukturen

DistribSet ist das Objekt in **Janus** , welches zentral die Kodebücher und die dazugehörigen Verteilungen beinhaltet. Dadurch paßte es gerade ideal zu unserem Verfahren.

Folgende Datenstrukturen wurden darin benutzt:

- Kodebuch für Sprache.
- Kodebuch für Stille
- Verteilung für Sprache (a priori Wahrscheinlichkeiten $P[k]$) .
- Verteilung für Stille.

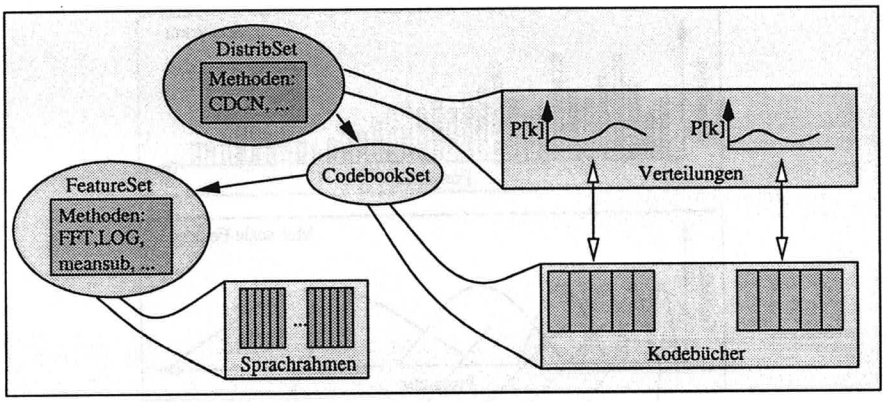


Abbildung 6.7: Die Einbindung von CDCN in die Objektstruktur von **Janus** . CDCN ist als Methode von **DistribSet** implementiert. Alle nötigen Komponenten wie Kodebücher und Verteilungen sind in diesem Objekt enthalten.

Die Kodebücher sind durch Kodebuchvektoren mit zugehörigen Kovarianzmatrizen realisiert. Die Kovarianzmatrizen haben dabei Diagonalgestalt, was keine notwendige Voraussetzung ist, jedoch die Zeit zum Berechnen der a posteriori Wahrscheinlichkeiten verringert. Zu jedem Kodebuchvektor existiert ein Verteilungskoeffizient $P[k]$, der die a priori Wahrscheinlichkeit für das Auftreten des Vektors repräsentiert. Die Kodebücher, Kovarianzmatrizen und Verteilungen wurden in einem Trainingslauf berechnet. Als Trainingsdaten wurden die Trainingsdaten des Spracherkenners benutzt. Dies gewährleistet, daß die im CDCN-Kodebuch repräsentierte Sprache konsistent mit der des Spracherkenners ist. Die Mittelwertsubtraktion wurde als Vorverarbeitungsschritt beim Training der CDCN-Kodebücher gewählt.

Zusätzlich wurden verschiedene Ausgabefunktionen integriert, die die Fehlersuche und die Leistungsabschätzung ermöglichen sollen. So können die geschätzten Vektoren $\hat{\mathbf{n}}$ und $\hat{\mathbf{q}}$ abgefragt werden, außerdem besteht die Möglichkeit die a posteriori Wahrscheinlichkeiten für das bearbeitete Sprachsegment auszugeben.

6.2.2 An welcher Position soll CDCN eingesetzt werden?

Es gibt mehrere Punkte in der Signalvorverarbeitung eines Spracherkenners an denen man CDCN einbinden kann. Die Möglichkeit direkt auf die logarithmierten Kurzzeitspektren (10 Millisekunden) aufzusetzen erscheint zunächst sehr unbedenklich. Sie hat jedoch mehrere gravierende Nachteile. Direkte Spektren enthalten viele sprecherspezifische Komponenten, die nicht im Kodebuch repräsentiert werden sollten. Das Kurzzeitspektrum enthält die Anregung, die sich zum Beispiel bei stimmhaften Lauten durch eine Grundfrequenz (F_0) und zugehörige Oberwellen bemerkbar macht. F_0 ist sprecherabhängig und ist für die Erkennung nicht erforderlich [1]. Außerdem muß durch die hohe Dimension des Kodebuchvektorraums das Kodebuch eine große Anzahl von Vektoren enthalten um gut modellieren zu können. Ein hochdimensionales Kodebuch mit einer großen Anzahl von Vektoren hat einen hohen Rechenzeitbedarf.

[1] Dies gilt zumindest für europäische Sprachen wie Deutsch und Englisch.

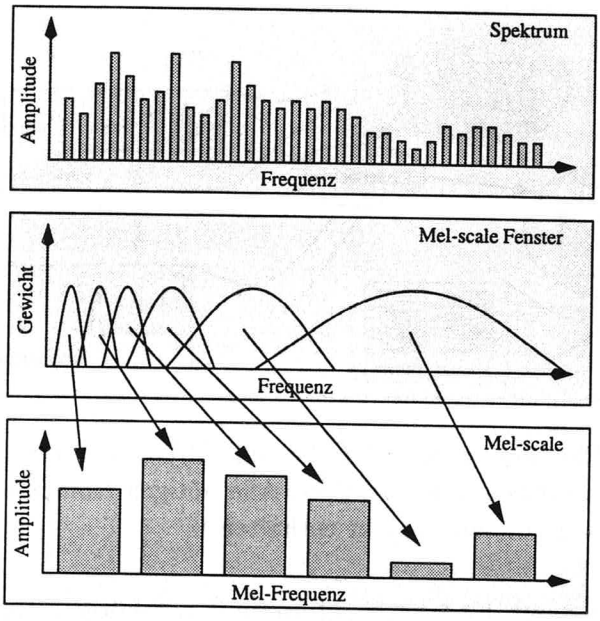


Abbildung 6.8: Die Melscale-Bänder werden als die gewichtete Summe der Spektralkomponenten mit der zugehörigen Fensterfunktion berechnet. Die Frequenzachse der Melsscaledarstellung ist nicht mehr linear sie wird als Mel-Frequenz bezeichnet.

Genau dies sind auch die Gründe warum viele Spracherkenner auf Filterbänken basieren. Diese haben den Vorteil das Problem der unterschiedlichen spektralen Repräsentation verschiedener Stimmlagen zu entschärfen. Es erfolgt der Übergang auf das geglättete Spektrum (Einhüllende). Dabei wird das Spektrum durch Fensterfunktionen zu wenigen geeigneten Bändern zusammengefaßt (siehe Abbildung 6.8) . In unserem Fall wird von 129 Spekralkomponenten zu 30 Bändern übergegangen. Die Fensterfunktionen sind auf die Frequenzauflösung des menschlichen Gehörs abgestimmt. Die Breite der einzelnen Bandfenster wächst näherungsweise exponentiell mit der Frequenz. Dadurch werden niedrige Frequenzanteile mit einer hohen Auflösung dargestellt, hohe Frequenzen mit niedriger Auflösung. Untersuchungen haben gezeigt, daß der Bereich oberhalb von 4000 Hertz wenig zum Sprachverständnis beiträgt. Darum ist es sinnvoll die niedrigen Frequenzen besser aufzulösen als die hohen. Allein die Reduzierung auf 30 Komponenten ergibt eine Verringerung der Rechenzeit um den Faktor vier. Durch die niedrigere Dimensionalität der Kodebuchvektoren werden weniger Kodebuchvektoren gebraucht. Deshalb kann man von einer weiteren Halbierung der Rechenzeit gegenüber CDCN (Codeword Dependent Cepstral Normalization) mit direkten Spektralvektoren ausgehen.

Es stellt sich die Frage, ob dann nicht gleich im Cepstralbereich (siehe Abschnitt 3.3) gearbeitet werden soll. Prinzipiell spricht nichts dagegen. Die Reduzierung auf 13 Cepstralkomponenten würde eine weitere Beschleunigung bringen. Die dafür verwendete Fouriertransformation ist eine lineare Operation, so daß CDCN sowohl im Cepstral als auch im Log-Spektralbereich anwendbar ist. Jedoch müßte zur Berechnung der Korrekturvektoren $\mathbf{r}$ ständig zwischen Cepstral und Log-Spektralbereich hin und her gerechnet werden.

Deshalb wird hier nicht in die Cepstraldarstellung übergegangen, sondern direkt im Log-Spektralbereich gearbeitet.

6.3 Versuche mit CDCN

Nun folgen einige Versuche zur Arbeitsweise von CDCN. Dabei wird die Mittelwertsubtraktion als Referenzsystem benutzt. Es soll untersucht werden was CDCN (Codeword Dependent Cepstral Normalization) anders macht. Die Fähigkeit unterschiedliche Aufnahmeumgebungen zu normalisieren steht dabei im Vordergrund. Als Ausgangsdaten diente dabei ein willkürlich ausgewähltes Sprachsegment, welches das gesprochenene Wort "gut" enthielt. Es werden fünf Mikrofone betrachtet: das Tischmikrofon AT 9750, das Standmikrofon AT 9820X, das Ansteckmikrofon, das Headset HME 1410K und das Headset HMD 410.

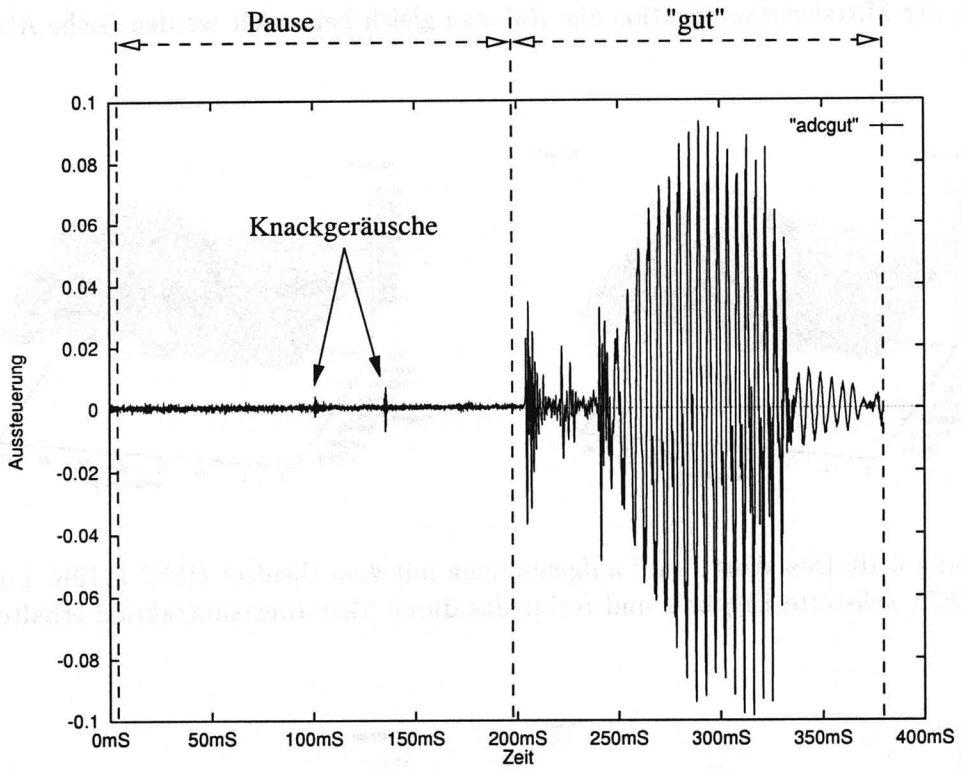


Abbildung 6.9: Das Zeitsignal des Wortes "gut" aufgenommen mit dem Headset HME 1410K.

Der Sprecher des Sprachsegments trug das Headset HME 1410K, so daß die zugehörige Aufnahme den größten Signalrauschabstand hat. Es wird deshalb als Referenz benutzt. Das zweite Headset HMD 410 wurde vom zweiten Sprecher getragen und hat deshalb den niedrigsten Signalrauschabstand im hier verwendeten Sprachsegment. Auf dem Tisch im Aufnahmeraum befanden sich das AT 9750 und das Standmikrofon AT 9820X in etwa 90 Zentimeter Abstand vom Sprecher. An der Brust des Sprechers war das Ansteckmikrofon befestigt.

6.3.1 Gegenüberstellung der Kurzzeitspektren

Die folgenden Abbildungen enthalten die Gegenüberstellung der Kurzzeitspektren der Resultate von CDCN und der Mittelwertsubtraktion. Dabei wurde jeweils das gesamte Sprachsegment bearbeitet. Aus Platzgründen werden jedoch nur die 38 zum Wort "gut" gehörenden Sprachrahmen gezeigt. Abbildung 6.9 zeigt das Zeitsignal des gewählten Ausschnitts.

Deutlich ist zu sehen, daß es große Unterschiede zwischen den beiden Verfahren gibt. CDCN (Codeword Dependent Cepstral Normalization) verändert durch die Korrekturvektoren $r[k]$ jeden Sprachrahmen separat. Auffällig ist dies bei den Sprachrahmen fünf und vierzehn wo CDCN offenbar einige Sprachrahmen vermutete (siehe Abbildung 6.11). Tatsächlich befinden sich an diesen Positionen Knackgeräusche (siehe Abbildung 6.9). Bei der Aufnahme mit dem schlechtesten Signalrauschabstand hat CDCN Vorteile, weil nicht wie bei der Mittelwertsubtraktion alle Rahmen gleich behandelt werden (siehe Abbildung 6.14) .

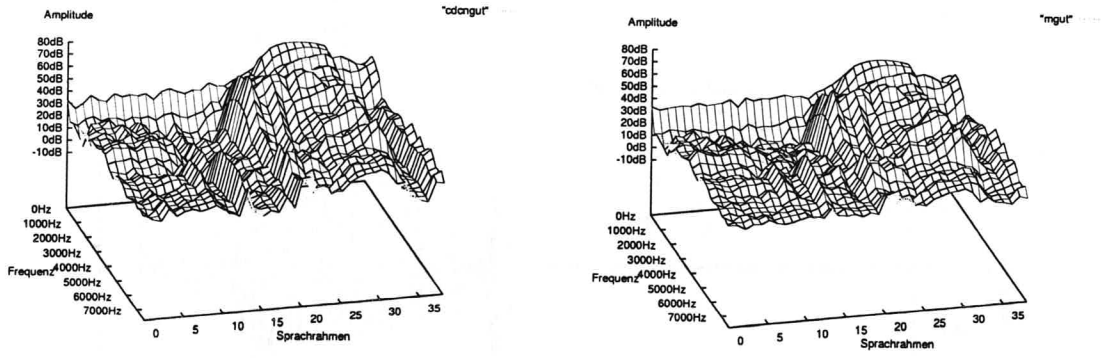


Abbildung 6.10: Das Wort "gut" aufgenommen mit dem Headset HME 1410K. Links das von CDCN gelieferte Ergebnis und rechts das durch Mittelwertsubtraktion erhaltene.

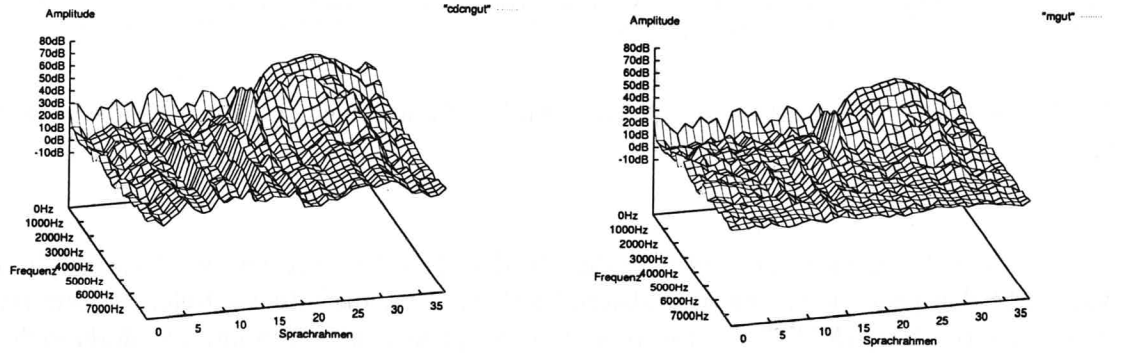


Abbildung 6.11: Das Wort "gut" aufgenommen mit dem Tischmikrofon AT 9750. Links CDCN rechts Mittelwertsubtraktion.

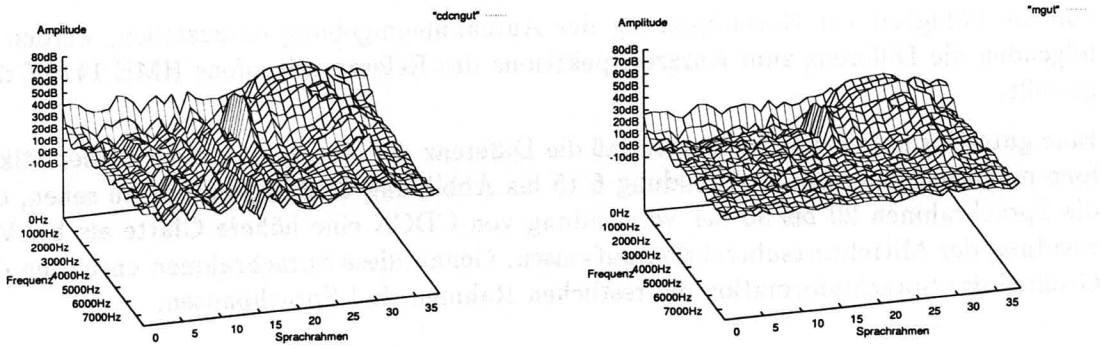


Abbildung 6.12: Das Wort "gut" aufgenommen mit dem Standmikrofon AT 9820X. Links CDCN rechts Mittelwertsubtraktion.

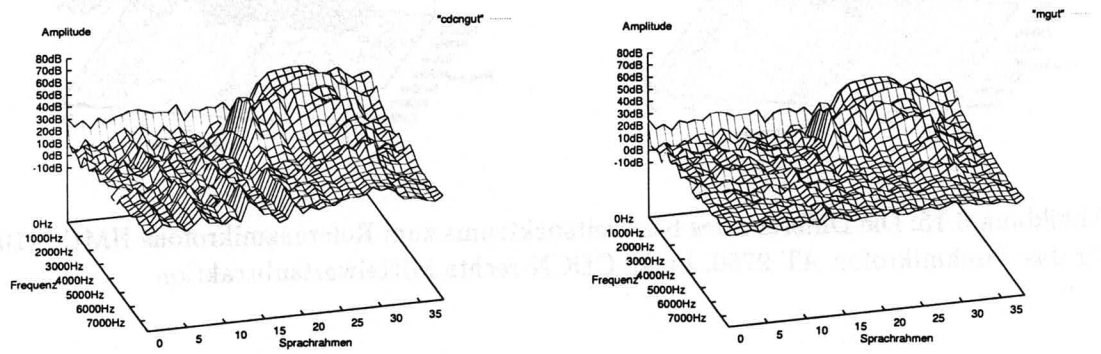


Abbildung 6.13: Das Wort "gut" aufgenommen mit dem Ansteckmikrofon. Links CDCN, rechts Mittelwertsubtraktion.

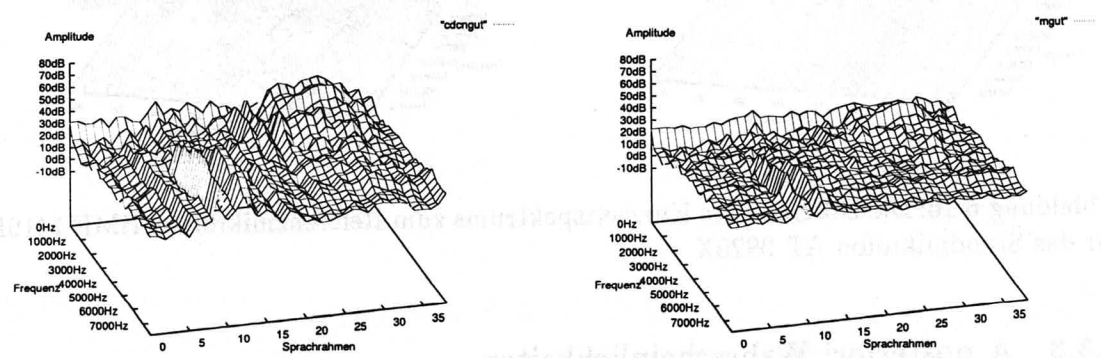


Abbildung 6.14: Das Wort "gut" aufgenommen mit dem Headset HMD 410. Links CDCN, rechts Mittelwertsubtraktion.

6.3.2 Vergleich mit dem Referenzmikrofon HME 1410K

Um die Fähigkeit zur Normalisierung der Aufnahmeumgebung darzustellen, werden im folgenden die Differenz zum Kurzzeitspektrums des Referenzmikrofons HME 1410K dargestellt.

Eine gute Normalisierung bedeutet, daß die Differenz der Kurzzeitspektren zweier Mikrofone möglichst glatt ist. In Abbildung 6.15 bis Abbildung 6.18 ist deutlich zu sehen, daß die Sprachrahmen 20 bis 30 bei Verwendung von CDCN eine höhere Glätte als bei Verwendung der Mittelwertsubtraktion aufweisen. Genau diese Sprachrahmen enthalten den Großteil der Sprachinformation die restlichen Rahmen sind Sprachpausen.

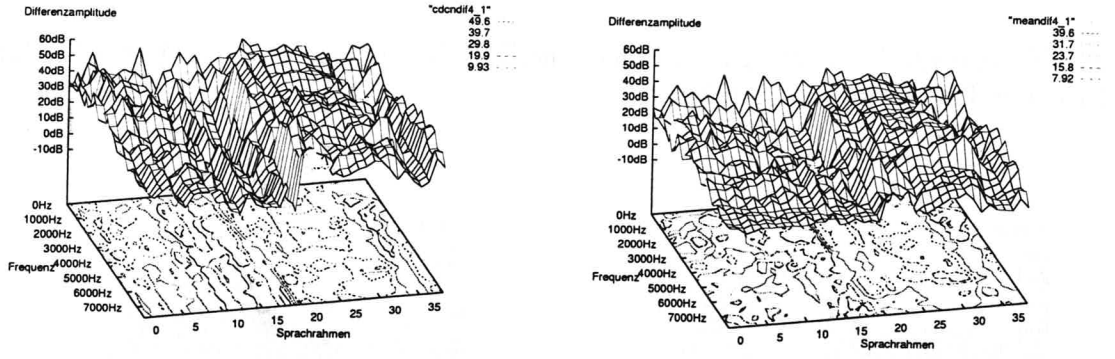


Abbildung 6.15: Die Differenz des Kurzzeitspektrums zum Referenzmikrofons HME 1410K für das Tischmikrofon AT 9750. Links CDCN rechts Mittelwertsubtraktion.

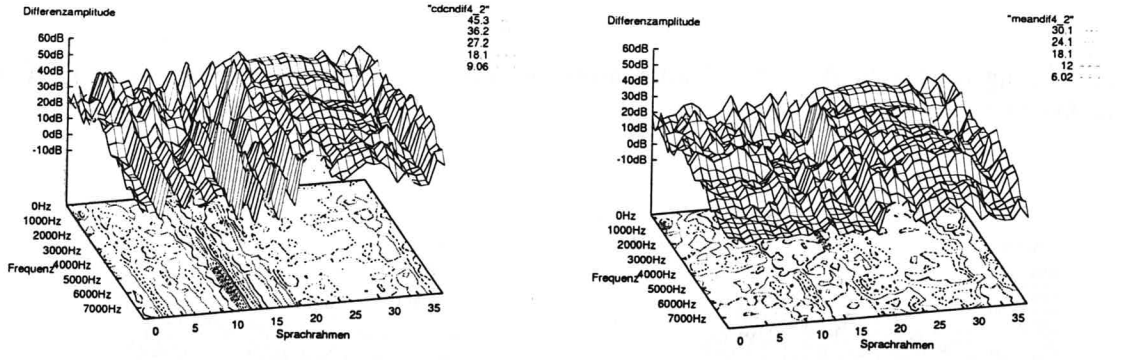


Abbildung 6.16: Die Differenz des Kurzzeitspektrums zum Referenzmikrofons HME 1410K für das Standmikrofon AT 9820X

6.3.3 A posteriori Wahrscheinlichkeiten

Einen tieferen Einblick in die Arbeitsweise von CDCN (Codeword Dependent Cepstral Normalization) erhält man durch die Betrachtung der berechneten a posteriori Wahrscheinlichkeiten. Zu den in den vorangegangenen Abschnitten gemachten Versuchen wer-

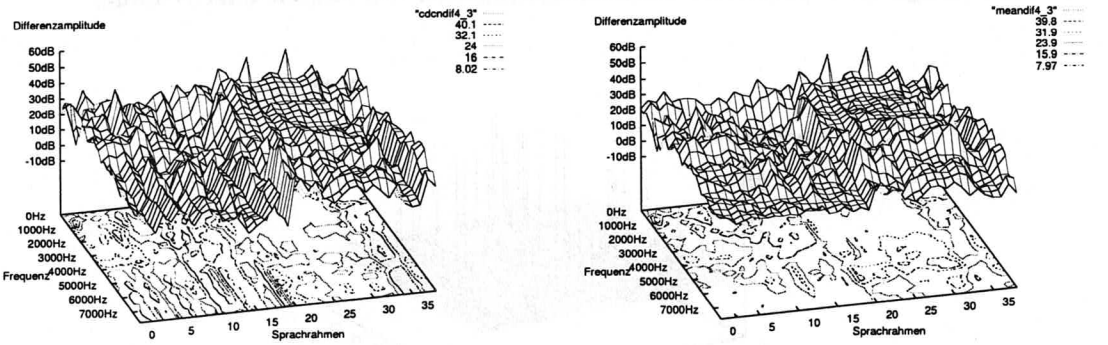


Abbildung 6.17: Die Differenz des Kurzzeitspektrums zum Referenzmikrofons HME 1410K für das Ansteckmikrofon.

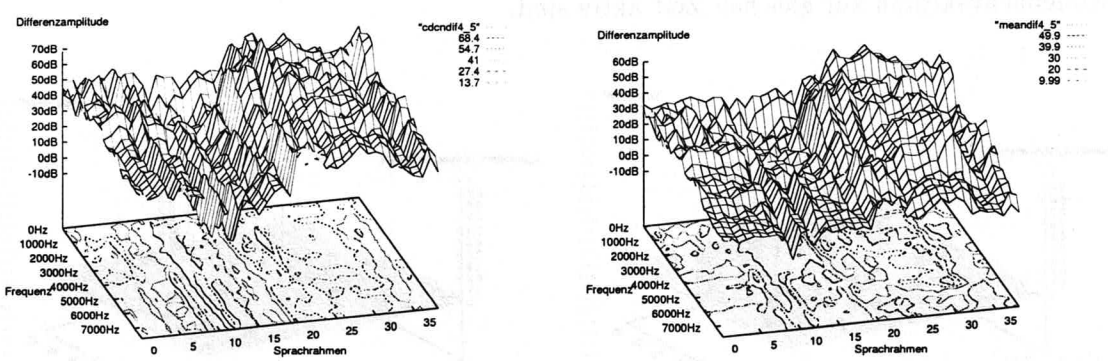


Abbildung 6.18: Die Differenz des Kurzzeitspektrums zum Referenzmikrofons HME 1410K für das Headset HMD 410.

den nun die zugehörigen a posteriori Wahrscheinlichkeiten gezeigt (Abschnitt 6.3.1). Vertikal ist die berechnete Wahrscheinlichkeit aufgetragen, horizontal die Kodebuchvektoren. Die Vektoren von Null bis fünfzig repräsentieren dabei Stille oder Pausen die restlichen zweihundert repräsentieren Sprache. Rechts sind die Nummern der Sprachrahmen aufgetragen. Wiederum werden 38 Sprachrahmen des Wortes "gut" dargestellt. Der dargestellte Ausschnitt zeigt die a posteriori Wahrscheinlichkeiten des letzten CDCN-Laufs.

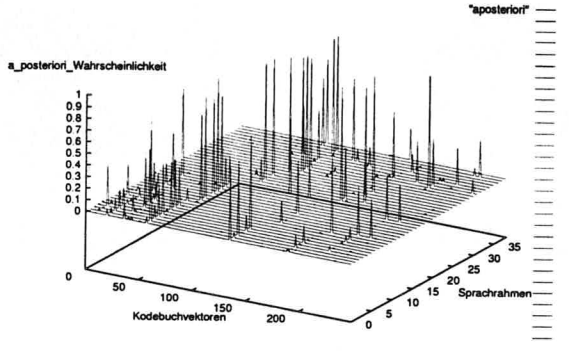


Abbildung 6.19: Das Referenzmikrofons HME 1410K zeigt deutlich wie gut das Kodebuch Sprache repräsentiert. Zu fast jedem Sprachrahmen paßt genau ein Kodebuchvektor.

Abbildung 6.19 zeigt die a posteriori Wahrscheinlichkeiten für das Referenzmikrofons HME 1410K. Bemerkenswert ist die fast eindeutige Zuordnung eines Kodebuchvektors für Sprache zu einem Sprachsegment. Der Grund dafür ist der große Signalrauschabstand und die gute Repräsentation von Sprache im Kodebuch. In Abbildung 6.20 und Abbildung 6.21 sieht man, daß die Pausenwahrscheinlichkeiten für die Sprachsegmente 15 bis 38 fast Null sind. Die Unterscheidung zwischen Sprache und Pausen funktioniert also. Selbst für das Headset HMD 410 ist noch eine deutliche Unterscheidung möglich, obwohl sehr viele Kodebuchvektoren zur gleichen Zeit aktiv sind.

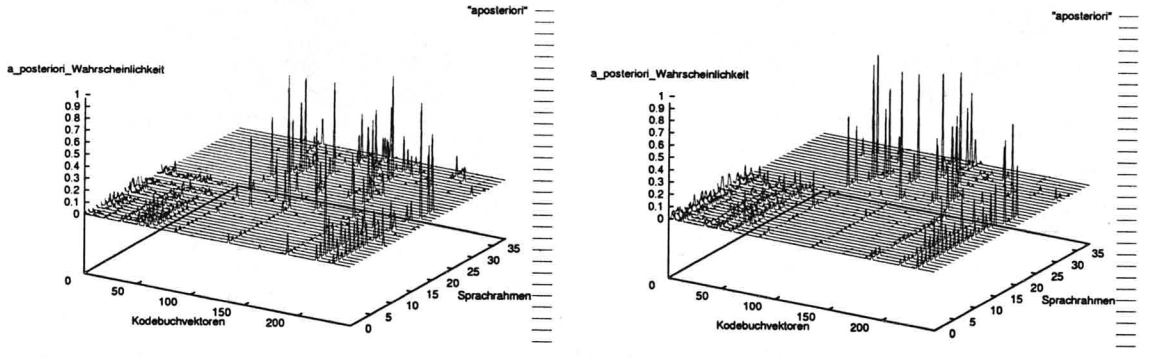


Abbildung 6.20: Das linke Bild zeigt die a posteriori Wahrscheinlichkeiten für das Tischmikrofon AT 9750 das rechte Bild für das Standmikrofon AT 9820X.

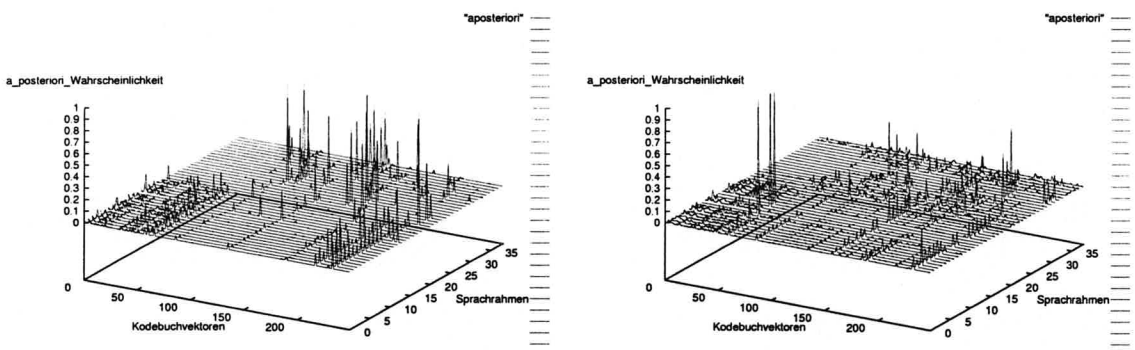


Abbildung 6.21: Links sieht man die a posteriori Wahrscheinlichkeiten für das Ansteckmikrofon und rechts für das Headset HMD 410.

Kapitel 7

Erstellung der Testdaten

Zum Test der Fähigkeit von CDCN mußten Aufnahmen mit unterschiedlichen Mikrofonen erstellt werden. Als besonderer Aspekt galt dabei die Vergleichbarkeit der Daten. Wenn ein Dialog simultan auf mehreren unterschiedlichen Mikrofonen aufgenommen wird, ist diese Forderung erfüllt. Die Verwendung der institutseigenen Aufnahmegeräte (Aufnahmebox direkt an eine Workstation angeschlossen) zusammen mit einem gewöhnlichen Dat-Recorder schied von vorneherein aus, da es Probleme bei der synchronen Aufnahme gab. Außerdem unterscheiden sich die beiden Verfahren so stark, daß Fragen nach der Vergleichbarkeit erst geklärt werden müßten. Als nächste Möglichkeit wurde das Verwenden von zwei simultanen baugleichen Dat-Rekordern erwogen. Dabei besteht ebenfalls das Problem des Synchronisierens der beiden Recorder. Zusätzlich ensteht wie beim vorherigen Verfahren ein nicht unerheblicher Bedienungsaufwand der beiden Geräte. Schließlich fiel die Wahl auf einen Achtkanal-Dat-Recorder. Er kann acht Kanäle auf einmal aufnehmen. Für Mehrkanalaufnahmen ist dieses Gerät gut geeignet da alle acht Kanäle exakt gleich ausgelegt sind. Der Nachteil an diesem System ist, daß die aufgenommen Daten später in einem sehr zeitaufwendigen Verfahren auf einen Rechner überspielt werden müssen.

7.1 Die Aufnahme der Audiodaten

Zur Aufnahme wurde der 8-Kanal Dat-Recorder Sony PC-208 werwendet. Mit dem Sony PC-208 können 8 Kanäle gleichzeitig bei 20 kHz Abtastrate aufgenommen werden. Es besteht dabei die Möglichkeit die aufgenommen Daten später digital auf einen PC zu überspielen. Weiterhin wurde ein 8-Kanal Vorverstärker verwendet. Dieser bietet für jeden Kanal ein separates Bandpaß-Filter und eine separate Verstärkungseinstellung.

Ein als Teil der Diplomarbeit angefertigter Signalgeber mit zwei Sinussignalen (1000 Hz und 2000 Hz) wurde zum Markieren der Aufnahmen verwendet (siehe Abbildung 7.1 und 7.3).

Jedes Sinussignal dauert mindestens 200 Millisekunden um die spätere Verarbeitung zu erleichtern. Außerdem wurde in der Elektronik des Signalgebers der zeitliche Ablauf des der Signale vorgegeben. So muß zuerst ein 1000 Hz Ton und dann eine gerade Anzahl von 2000 Hz Tönen erfolgen, bevor wieder ein 1000 Hz erlaubt wird. Das 1000 Hz Signal dient zum Aufteilen in einzelne Aufgaben (Sessions). Durch das 2000 Hz Signal werden die Sprachsegmente (Utterances) eingerahmt (siehe Abbildung 7.2) .

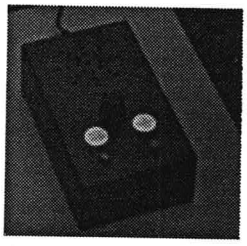


Abbildung 7.1: Der im Rahmen der Diplomarbeit gebaute Signalgenerator zum akustischen Markieren der Bandaufnahmen. Die grüne Taste (links) erzeugt einen 1000 Hz Ton die rote Taste (rechts) einen 2000 Hz Ton.

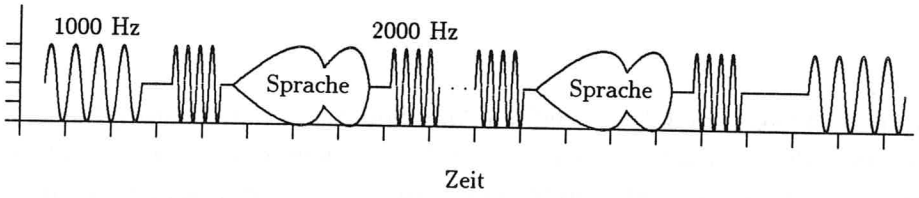


Abbildung 7.2: Die Markierung der Sprachsegmente geschieht durch zwei Sinustöne mit 1000 Hz bzw 2000 Hz.

Durch die zusätzlich aufgezeichneten Sinussignale können die gesammelten Daten automatisch in Sprachsegmente zerlegt werden. Durch das 1000 Hz Signal erfolgt eine einfache Zuordnung der Sprachsegmente zu den Aufgaben und den beteiligten Sprechern. Weiterhin wurde während des Aufnahmebetriebs Buch über die Aufgaben und die Sprecher geführt.

Es wurden fünf Mikrofone verwendet. Davon waren drei separate Mikrofone und die restlichen Bestandteil von zwei Headsets [1] (siehe Abbildung 7.4):

1. Audio Technica AT 9750  .
2. Audio Technica AT 9820X  .
3. Audio Technica Ansteckmikro  .
4. Headset Sennheiser HME 1410K (Sprecher 1)  .
5. Headset Sennheiser HMD 410 (Sprecher 2)  .

7.2 Das Aufnahmeszenario

Als Szenario für die Aufnahmen wurde das am Institut und von anderen Spracherkennungsgruppen häufig verwendete "Reise spontan" benutzt. Dabei müssen jeweils zwei Sprecher

[1] Kopfhörer mit Mikrofon.

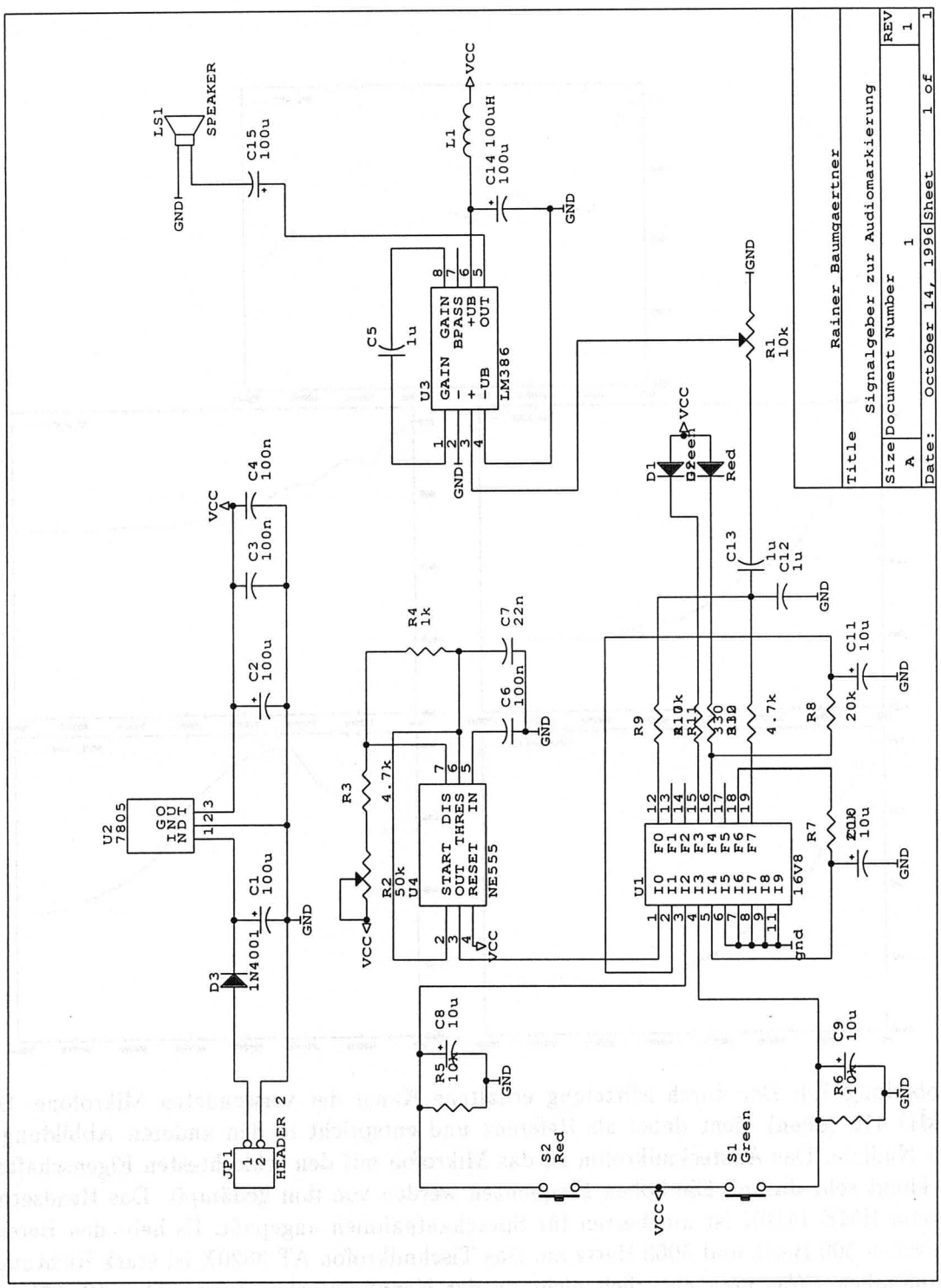


Abbildung 7.3: Schaltplan des Signalgenerators. Die gesamte Steuerung wird von einem programmierbaren Logikbaustein (GAL16V8) übernommen. Der NE555 dient zur Erzeugung eines Rechtecksignals mit 4000 Hertz. Ein integrierter Leistungsverstärker (LM386) verstärkt die Signale damit sie mit dem Lautsprecher hörbar gemacht werden können.

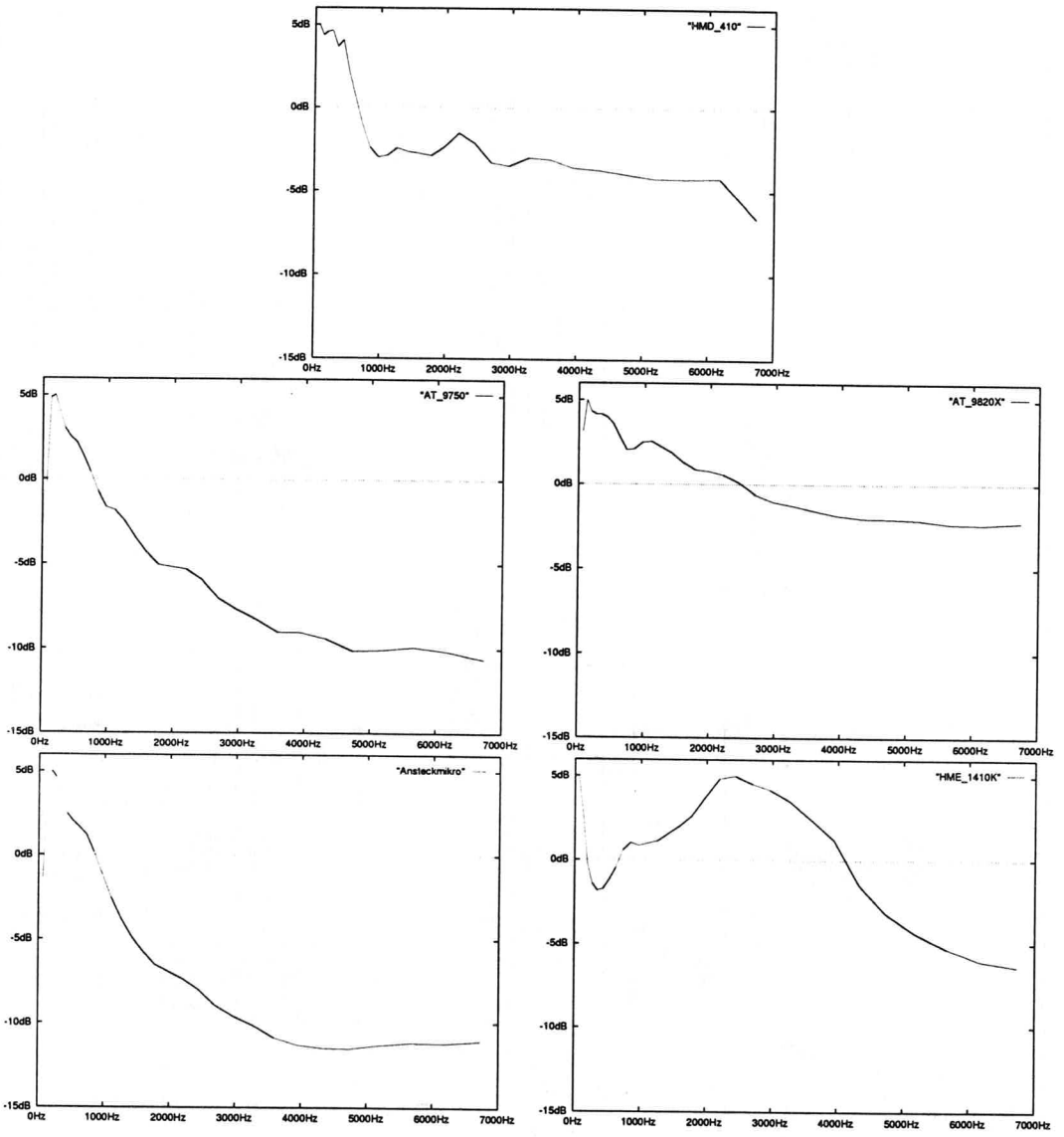


Abbildung 7.4: Der durch Mittelung erhaltene Kanal der verwendeten Mikrofone. Das HMD 410 (oben) dient dabei als Referenz und entspricht in den anderen Abbildungen der Nullinie. Das Ansteckmikrofon ist das Mikrofon mit den schlechtesten Eigenschaften, es klingt sehr dumpf. Die hohen Frequenzen werden von ihm gedämpft. Das Headsetmikrofon HME 1410K ist am besten für Sprachaufnahmen angepaßt. Es hebt den Bereich zwischen 500 Hertz und 5000 Hertz an. Das Tischmikrofon AT 9820X ist stark Richtungsempfindlich (Was man natürlich nicht an der Kennlinie sehen kann). Das AT 9750 ist ein Allroundmikrofon mit annehmbarer Übertragungskennlinie. Es unterscheidet sich vom Ansteckmikrofon durch einen größeren Signalrauschabstand.

eine Terminabsprache für ein Geschäftstreffen tätigen. Zeitangaben, Terminkonflikte und Anreise werden dabei eingeflochten. Jeder der Sprecher erhält einen Zeitplan und eine Aufgabe, die beschreibt wo das Treffen stattfinden soll. Der Dialog erfolgt frei in natürlicher Sprache.

Die Aufnahmen wurden bei geschlossener Tür in einem Seminarraum durchgeführt. Da realistische Daten gewonnen werden sollten, wurden keine weiteren Maßnahmen zur Verbesserung der Aufnahmebedingungen gemacht. Sprecher waren ausschließlich Studenten.

Nach zwangloser Begrüßung wurde den beiden Teilnehmern kurz das Szenario erklärt. Diese Phase wurde so kurz wie möglich gehalten, um spontane Gespräche zu begünstigen. Als nächstes wurde dann ein kurzer Test ohne Aufnahme gemacht. Danach begann die Aufnahme. Dabei waren die beiden Sprecher gezwungen zu improvisieren. Bis auf wenige Ausnahmen klappte dies auf Anhieb.

Zu jeder Aufnahme wurde ein Aufnahmeprotokoll ausgefüllt, das die Namen der beiden Sprecher, die Uhrzeit, die bearbeitete Aufgabe, die genaue Bandposition am Beginn und Sprechereigenheiten wie Lispeln, Dialekt oder Akzent enthielt (Anhang C).

Kleinere Fehler wie Versprecher und Papierrascheln, wurden toleriert. Sie sind Bestandteile von Spontansprache. Nur für größere Zwischenfragen wurde die Aufnahme neu gestartet.

7.3 Segmentierung der Audiodaten

Die zusätzlich aufgenommenen Sinustöne wurden zum halbautomatischen Schneiden verwendet. Dabei wurde der erste Kanal zuerst fouriertransformiert, dann das Ergebnis logarithmiert und eine Schwellwertfunktion angewendet. Die so erhaltenen Rohschnittpunkte wurden zum manuellen Anfahren der Sprachsegmente beim Transkribieren benutzt. Es hat sich gezeigt, daß durch die Sinustöne ein sicheres Auffinden der Sprachsegmente gewährleistet ist. Der Umstand, daß im Aufnahmeraum viele Schallreflektionen vorhanden waren wirkte sich ungünstig auf das genaue Schneiden aus. Wenn ein Sprecher sofort nach dem Signalton zu sprechen begann, konnte es zu Überlagerungen mit dem Echo des Signaltons kommen. Auch war durch das Echo das Auffinden des genauen Endes des Signaltons nur schwer zu automatisieren.

Als Lösung dieses Problems bietet sich an den Signalton auf einem getrennten Kanal aufzunehmen. Oft ist dies jedoch nicht möglich weil es nicht einfach ist zusätzliche synchrone Kanäle bereitzustellen. Deshalb ist es ebenfalls ausreichend die Testpersonen zu kleinen Pausen vor und nach einem Signalton aufzufordern. Als dritte Möglichkeit besteht die Eliminierung der stark bandbegrenzten Echos durch einen Filteralgorithmus. Dabei muß jedoch darauf geachtet werden, daß das Sprachsignal so wenig wie möglich verfälscht wird.

Kapitel 8

Erkennungsergebnisse

Hier soll nun die Leistungsfähigkeit von CDCN durch einige Versuche gezeigt werden. Dabei geht es im besonderen Maße um die Fähigkeit zur Anpassung an andere Aufnahmebegebenheiten.

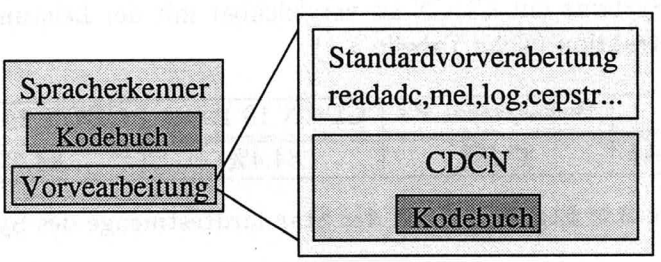


Abbildung 8.1: CDCN (Codeword Dependent Cepstral Normalization) stellt einen Teil der Vorverarbeitung des Spracherkenners dar. Durch das eigene Kodebuch wird eine weitgehende Unabhängigkeit vom darüberliegenden Erkenner erreicht. Dabei ist das Kodebuch nur von den Trainingsdaten und der Pausendetektion abhängig.

Als Basissystem diente ein Vorläufer des Evaluationssystems, das mit deutlichem Vorsprung die VM-Evaluation September 96 gewonnen hat. Dieses System besteht aus 10000 Kodebuchverteilungen, die über 2500 Kodebüchern definiert sind. Die Erkennungsrate liegt bei diesem System bei ungefähr 85%. Das System ist sprecherunabhängig und verarbeitet kontinuierliche Sprache. Der Wortschatz und die verwendeten Grammatiken sind für Terminabsprachen ausgelegt. Von vorneherein sollten alle Änderungen so gehalten werden, daß ein leichtes Übertragen auf andere Spracherkenner unter Janus möglich ist. Wenn man die Größe des Systems (2500 Kodebücher) betrachtet wird klar, daß eine Vorverarbeitung mit den systemeigenen Kodebüchern nur mit sehr viel Rechenaufwand möglich ist. Außerdem basieren die Kodebuchverteilungen auf dem verwendeten **HMM**, so daß Kodebuchvektoren nicht ohne weiteres für CDCN verwendet werden können. Deshalb wurde CDCN als eigenständiges System konzipiert. Dabei wird lediglich in die Signalvorverarbeitung und nicht in den eigentlichen Erkenner eingegriffen. Es wird ein eigenes relativ kleines Kodebuch verwendet, daß nur von den Trainingsdaten und der Pausendetektion und nicht vom Erkenner abhängt (siehe Abbildung 8.1) . So kann durch minimale Änderung des Systems und anschließendes Neutrainieren das Verfahren eingebaut werden.

8.1 Test des Basissystems

Um zu zeigen welchen Einfluß CDCN auf die Leistung des Spracherkenners hat, wird nun ein für CDCN angepaßtes System mit dem Basissystem verglichen. Die dabei verwendeten Testdaten unterscheiden sich nur wenig in ihren Kanaleigenschaften von den benutzten Trainingsdaten.

Das Basissystem benutzt die Mittelwertsubtraktion mit Pausenberücksichtigung zur Kanalkompensation. Zum Training wurden mehr als 10000 Sprachsegmente verwendet.

Als Testdaten wurde "VM-Devtest 96" [1] verwendet. Die Testdaten bestehen aus 268 Sprachsegmenten, die nicht zum Training benutzt wurden. Dabei wurden für die Aufnahme der Testdaten andere Sprecher verwendet als für die Aufnahme der Trainingsdaten. Es sind darin Spachsegmente von mehreren Sprechern beiderlei Geschlecht enthalten, wobei die Testdaten unter den selben Bedingungen aufgenommen wurden wie die Trainingsdaten. Die beiden getesteten Spracherkenner unterscheiden sich lediglich in der Signalvorverarbeitung. Die gesamte Mittelwertsubtraktion des Basissystems wurde entfernt und CDCN dafür eingefügt.

Die Leistung des Systems mit CDCN ist vergleichbar mit der Leistung des Basissystems mit Mittelwertsubtraktion (siehe Tabelle 8.1).

	Basissystem P3	CDCN 10 Iterat	CDCN 30 Iterat
Korrekt	85.0%	84.4%	84.5%

Tabelle 8.1: Ergebnisse mit der Standardtestmenge des Systems.

8.2 Test mit Mehrkanalaufnahmen

Hier soll untersucht werden in wie weit CDCN (Codeword Dependent Cepstral Normalization) in der Lage, ist Aufnahmen mit unterschiedlichen Kanaleigenschaften zu normalisieren.

Basis der Tests sind die im Rahmen der Diplomarbeit erstellten Mehrkanalaufnahmen (Kapitel 7). Diese Daten unterscheiden sich vollständig von den zum Training verwendeten Daten. Die Trainingsdaten wurden direkt an einem Rechner mit Mikrofonanschluß aufgenommen, die Mehrkanalaufnahmen wurden mit einem DAT-Recorder gemacht. Die bei den Mehrkanalaufnahmen verwendeten Headsets HME 1410K und HMD 410 sind vergleichbar mit den Headsets der Trainingsdaten. Jedoch unterscheidet sich die Eingangselektronik des DAT-Recorders und der zur Aufnahme verwendeten Rechner so stark, daß die Aufnahmen sehr verschieden sind. Weiterhin wurden die Aufnahmen mit mehreren Mikrofonen in einem anderen Raum gemacht. Das Tischmikrofon AT 9750, das Standmikrofon AT 9820X und das Ansteckmikrofon haben komplett andere Kanaleigenschaften als die bei der Aufnahme der Trainingsdaten verwendeten Headsets. Dadurch, daß diese drei Mikrofone einen großen Abstand zum Mund des Sprechers haben, enthalten die mit

[1] Die Development-Testmenge für die innerhalb des Verbundprojekts "*Verbmobil*" durchgeführten Evaluation 1996.

ihnen gemachten Aufnahmen mehr Störgeräusche. Ihr Signalrauschabstand ist geringer als der der Headsets.

Die Ergebnisse in Tabelle 8.2 belegen diesen Sachverhalt. Die Erkennungsrate des Basissystems ist um mehr als zwanzig Prozent gesunken. Dafür verantwortlich ist der niedrigere Signalrauschabstand, aber auch die ungenügende Kanalkompensation durch die Mittelwersubtraktion. CDCN erreicht um sieben bis zwölf Prozent bessere Ergebnisse. Dadurch ist die Fähigkeit von CDCN zur Kanalnormalisierung belegt. Deutlich ist zu sehen, daß durch eine höhere Anzahl von Iterationen eine weitere Leistungssteigerung möglich ist. Die dabei erreichbare Leistungssteigerung hängt von der verwendeten Aufnahmeumgebung ab und ist begrenzt. Dennoch können in manchen Fällen beachtliche Steigerungen erzielt werden. Beim Standmikrofon AT 9820X ist dieser Zusammenhang zu sehen. Durch zusätzliche zwanzig Iterationen wurde eine Steigerung von fast zehn Prozent erreicht.

Korrekt	AT 9750	AT 9820X	Ansteckmikro	HME 1410K	HMD 410
Basissystem	50.7 %	32.1%	43.8%	72.6%	75.5%
CDCN 10 Iterat.	61.2%	41.0%	56.0%	74.6%	77.9%
CDCN 30 Iterat.	63.1%	50.0%	59.8%	74.1%	78.0%

Tabelle 8.2: Tests auf den mit fünf Mikrofonen aufgenommenen Daten.

Kapitel 9

Vorschlag für weitere Arbeiten

Obwohl die in dieser Arbeit gezeigten Ergebnisse sehr ermutigend sind, steht die Erkennungsrate heutiger Spracherkenner noch weit hinter den Fähigkeiten des menschlichen Gehörs zurück. Speziell unter fremden Aufnahmebedingungen werden selbst mit CDCN als Kanalkompensationsverfahren vergleichsweise geringe Erkennungsraten erreicht.

Dabei stellt sich die Frage wo die Probleme bei der Kanalkompensation liegen. Die folgenden Punkte fallen dabei auf:

1. Die Kanalmodellierung könnte zu grob sein.
2. Informationen aus höheren Schichten des Spracherkenners könnten die Kanalkompensation wirkungsvoll unterstützen.

Da die Kanalmodellierung mit sehr großer Sorgfalt ausgewählt wurde (Kapitel 3) und außerdem Rechenzeiteffizienz eine Rolle spielt wird der erste Punkt nur schwer zu verbessern sein. Informationen aus höheren Schichten zur Kanalkompensation zu benutzen ist sicherlich ein vielversprechender Ansatz jedoch muß wegen der hohen Komplexität auf die Rechenzeit geachtet werden.

Bei den Arbeiten zu CDCN (Codeword Dependent Cepstral Normalization) trat die fehlende Vokaltraktnormalisierung häufiger als Randproblem in Erscheinung. Wenn keine Vokaltraktnormalisierung durchgeführt wird, dann sind die a posteriori Wahrscheinlichkeiten die in CDCN benutzt werden unzuverlässig. Es wäre also wünschenswert vor der Bearbeitung durch CDCN eine Vokaltraktnormalisierung auszuführen. Der Umstand, daß im Kodebuch von CDCN eine Repräsentation der Sprache die beim Training verwendet wurde existiert führt zu folgender Überlegung.

Die im Kodebuch vorhandenen Vokaltraktinformationen müßten zur Vokaltraktnormalisierung herangezogen werden. Nebenbei sei bemerkt das CDCN dies nicht leisten kann da CDCN rein additiv arbeitet (5.25). Zur Vokaltraktnormalisierung benötigt man eine Transformation, die im allgemeinen nicht rein additiv ist. Dennoch muß es möglich sein, die Informationen des Kodebuchs zur Normalisierung zu benutzen.

9.1 Kodebuchabhängige Vokaltraktnormalisierung

Hier wird nun die Idee für ein solche System skizziert. Als Name wäre CDFW (Codebook Dependent Fequency Warping) geeignet.

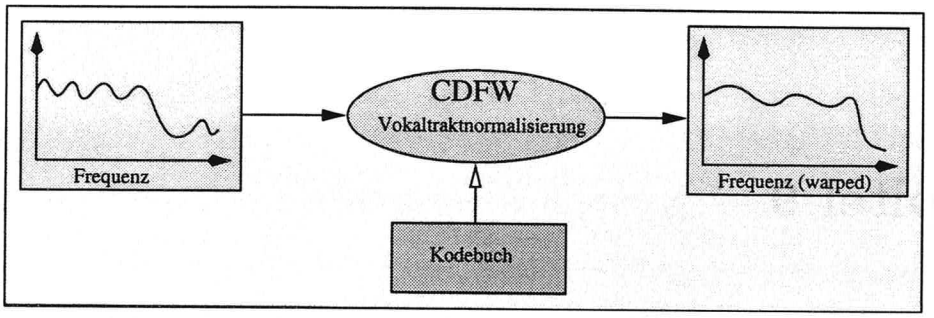


Abbildung 9.1: Vorschlag für ein Verfahren welches die Informationen des Kodebuchs zur Vokaltraktnormalisierung nutzt.

Genau wie schon CDCN soll das Verfahren ausschließlich auf der Sprachrepräsentation in einem Kodebuch $\lambda = \{(c[0], C[0], P[0]), ..., (c[K-1], C[K-1], P[K-1])\}$ basieren. Dieses Kodebuch muß nicht notwendigerweise das von CDCN benutzte sein. Ziel ist es eine Vokaltrakttransformation W_λ zu finden die die Ausgangsrahmen z_i so transformiert, daß das Ergebnis w_i besser zum Kodebuch paßt (siehe Formel 9.1).

$$w_i = W_\lambda(z_i) \tag{9.1}$$

9.2 ML Verfahren zur Bestimmung von W_λ

Zur Bestimmung von W_λ kann ein ML (Maximum Likelihood) Ansatz verwendet werden. Durch die Annahme, daß unterschiedliche Sprachrahmen z_i unabhängig sind, kann man folgenden Ausdruck benutzen:

$$\ln(p(Z|W_\lambda)) = \sum_{i=0}^{N-1} \ln(p(z_i|W_\lambda)) \tag{9.2}$$

Dabei wird $p(z_i|W_\lambda)$ durch eine Mischverteilung aus Gaußverteilungen modelliert (siehe Abschnitt 5.1.2):

$$p(z_i|W_\lambda) = \alpha \cdot \sum_{k=0}^{K-1} \frac{P[k]}{|C[k]|^{1/2}} \cdot \epsilon^{-\frac{\epsilon^T[i,k] \cdot C^{-1}[k] \cdot \epsilon[i,k]}{2}} \tag{9.3}$$

$$\epsilon[i,k] = W_\lambda(z_i) - c[k] \tag{9.4}$$

Die Maximierung von (9.2) ist in dieser allgemeinen Form nicht einfach. Es gibt jedoch einige a priori Informationen, die eine Lösung ermöglichen sollten. Die Transformation W_λ ist nicht beliebig. Nur der Einfluß des Vokaltrakts soll durch W_λ ausgeglichen werden. Als Transformation könnte man zum Beispiel eine Matrix als lineare Transformation wählen. Gängige Verfahren zur Vokaltraktnormalisierung benutzen zum Beispiel Transformationen, die nur von einem Parameter abhängen. Dies kann zu einer einfachen Matrixdarstellung führen.

Kapitel 10

Zusammenfassung

In Kapitel 3 wurde als Basis für die weiteren Arbeiten ein vereinfachtes Modell des Übertragungskanals vorgestellt. Besonderes Augenmerk wurde dabei auf die spätere Anwendung in Spracherkennern gelegt. Durch diese Vorüberlegungen konnte in Kapitel 4 eine Analyse der Mittelwertsubtraktion im logarithmierten Spektralbereich durchgeführt werden. Dabei wurde der Zusammenhang zwischen der linearen Verzerrung und dem Mittelwert eines Sprachsegments näher betrachtet. Es konnte gezeigt werden, daß der Mittelwert eine erste Näherung der lineareren Verzerrung darstellt. Außerdem konnte gezeigt werden, daß die Pausenberücksichtigung bei der Mittelwertberechnung eine bessere Näherung für die lineare Verzerrung liefert.

Die theoretische Analyse von CDCN (Codeword Dependent Cepstral Normalization) in Kapitel 5 leitet den Hauptteil dieser Diplomarbeit ein. CDCN transformiert Sprachdaten aus unterschiedlichen Aufnahmeumgebungen in ein durch die Trainingsdaten des Spracherkenners festgelegtes akustisches Modell. Als Basis dient dabei ein Kodebuch, welches dieses Modell repräsentiert. Ein ML (Maximum Likelihood) Schätzverfahren bestimmt dabei die linearen Verzerrung und das additiven Rauschen. Anschließend wird ein Schätzwert für unverfälschte Sprache durch einen MMSE (Minimum Mean Square Error Estimation) Ansatz gewonnen.

Die Untersuchungen zur Wahl der Startwerte in Abschnitt 6.1 ermöglichten einen ersten Einblick in die Arbeitsweise von CDCN. Das Konvergenzverhalten sollte dabei verbessert werden. Mit den Erfahrungen aus der Mittelwertsubtraktion wurde schrittweise ein Verfahren zur Bestimmung der Startwerte für die lineare Verzerrung und das additive Rauschen entwickelt. Der Abschnitt 6.2 zeigte die Einbindung von CDCN in **Janus** sowie einige Implementierungsdetails.

Eine Reihe von Versuchen an einem willkürlich herausgegriffenen Sprachsegment stellten in Abschnitt 6.3 die beiden Kanalkompensationsverfahren Mittelwertsubtraktion und CDCN gegenüber. Die Fähigkeit von CDCN Sprachrahmen zu normalisieren wurde dabei im direkten Vergleich mit der Mittelwertsubtraktion bestätigt.

Auf die Erstellung der Aufnahmen mit fünf Mikrofonen wurde in Kapitel 7 eingegangen. Einer kurzen Begründung des Aufnahmeverfahrens folgte die Beschreibung der genauen Vorgehensweise beim Aufnehmen. Außerdem wurden einige Probleme bei der Ausführung der Aufnahmen angesprochen.

Die erstellten Aufnahmen bildeten die Grundlage der Tests, die in Kapitel 8.2 durchgeführt wurden. Ein Spracherkenner für kontinuierliche Sprache wurde dort als Testsystem eingesetzt. Durch die Veränderung der Signalvorverarbeitung des Erkenners standen zwei Systeme für Tests zur Verfügung. Erstens das Basissystem selbst, welches auf der Mittelwertsubtraktion mit Pausenberücksichtigung basierte, und zweitens ein modifiziertes System, welches CDCN (Codeword Dependent Cepstral Normalization) benutzte. Die damit erhaltenen Resultate zeigten deutlich, daß CDCN eine effiziente Kanalkompensation leistet. Bei den Tests mit den erstellten Mehrkanalaufnahmen schnitt CDCN immer deutlich besser als die Mittelwertsubtraktion ab.

Den Abschluß der Diplomarbeit bildete ein Vorschlag zur Vokaltraktnormalisierung. Darin wurde eine ähnliche Vorgehensweise wie bei CDCN vorgeschlagen. Jedoch sind nicht die lineare Verzerrung und das additive Rauschen das Ziel des Verfahrens, sondern eine Funktion die den Vokaltrakt unterschiedlicher Sprachsegmente in eine normalisierte Darstellung überführt.

Im Verlauf dieser Diplomarbeit hat sich gezeigt, daß CDCN ein elegantes und leistungsfähiges System zur Kanalkompensation ist. CDCN hatte in beinahe allen Tests einen positiven Einfluß auf die Leistung im Vergleich zum Basissystem. Dadurch, daß keine Informationen aus dem eigentlichen Spracherkenner benutzt werden, ist CDCN ein kompaktes und leicht auf andere Spracherkenner übertragbares Verfahren. Dabei ist der Rechenaufwand beim Test klein im Vergleich zum Rechenaufwand des Erkenners.

Ein Nachteil ist der hohe Rechenaufwand beim Training des Spracherkenners. Durch einmaliges Abspeichern der Trainingsdaten nach der ersten Berechnung durch CDCN und anschließendes Neuladen kann der Rechenaufwand jedoch reduziert werden. Ein weiterer Nachteil ist das zusätzliche Kodebuch. Seine Größe ist zwar vernachlässigbar im Vergleich zum Kodebuch des verwendeten Spracherkenners, dennoch muß seine Größe und die Art und Weise wie es trainiert wird sorgfältig gewählt werden.

Literaturverzeichnis

[1] Acero Alejandro. *Acoustical and Environmental Robustness in Automatic Speech Recognition*, Department of Electrical and Computer Engineering Carnegie Mellon University Pittsburgh Pennsylvania 15231 13.9.1990

[2] Acero Alejandro, Richard Stern. *Enviromental Robustness in Automatic Speech Recognition*, Department of Electrical and Computer Engineering Carnegie Mellon University Pittsburgh Pennsylvania 15231 ICASSP 1990

[3] Fu-Hua Liu, Acero Alejandro, Richard Stern. *Efficient Joint Compensation of Speech for the Effects of additive Noise and linear Filtering*, Department of Electrical and Computer Engineering Carnegie Mellon University Pittsburgh Pennsylvania 15231 ICASSP 1992

[4] Uday Jain, Matthew Siegler, Sam-Joo Doh. *Recognition of Continious Broadcast News whith muliple unknown Speakers and Enviroments*, Department of Electrical and Computer Engineering Carnegie Mellon University Pittsburgh Pennsylvania 15231 1995

[5] W. Wittmann et. al. *Online Channel Compensation for robust Speech Recognition*, Simens AG Central Research an Development 1993

[6] M.J.Gales S.J.Young. *PMC for Speech Recognition in additive and convolutional Noise*, Cambridge University Engineering Department England 1993

[7] A.Erell M. Weintraub. *Spectral Estimation for Noise Robust Speech Recognition*, Proc. Speech and Natural Language Workshop , Cape Cod, MA 1989

[8] L.R. Rabiner R.W. Schafer. *Digital Processing of Speech Signals*, Prentice-Hall 1978

[9] Robert McDonough, Anthony Whalen. *Detection of Signals in Noise*, Academic Press Inc. 1995

[10] H.Stark John W.Woods. *Probability, Random Processes, and Estimation Theory for Engineers*, Pretice Hall 1994

[11] Prof. Karl Bosch. *Statistik Taschenbuch*, Oldenburg Verlag GmbH München 1992

Anhang A

Schnittstellenbeschreibung

Die Benutzerschnittstelle von CDCN (Codeword Dependent Cepstral Normalization) besteht im wesentlichen aus einer Funktion zum Initialisieren des Basisobjekts und der eigentlichen Arbeitsfunktion. Die folgende Beschreibung bezieht sich auf die **TCL**-Schnittstelle von **Janus** .

Die Benutzung ist deshalb denkbar einfach. Das Aufrufen der Funktion `cdcninit` während der Erkennerinitialisierung initialisiert alle benötigten Datenstrukturen. Durch eine Reihe von Parametern ist eine einfache Konfiguration möglich. Während der Initialisierungsphase werden die Verteilungen und die Kodebücher geladen und anschließend für CDCN modifiziert.

Der Aufrufsyntax von `cdcninit` ist:

`cdcnInit <SystemID> [optionale Parameter]`

Als optionale Parameter stehen zur Verfügung:

- `-dssdesc <DistibSet>` lädt die Verteilungsbeschreibung.
- `-dssparam <Distibutionweights>` lädt die Verteilungsgewichte.
- `-cbsdesc <CodebookSet>` lädt die Kodebuchbeschreibung.
- `-cbsparam <Codebooks>` lädt die Kodebücher.

Ein Beispiel:

Die Verteilungsbeschreibung wird aus `cdcn_desc/cdcnDistribSet` die Kodebuchbeschreibung aus `cdcn_desc/cdcnCodebookSetMel` geladen. Anschließend werden die Verteilungsgewichte aus `cdcn_create_melv/3i.dss.gz` geladen und die Kodebücher aus `cdcn_create_melv/3i.cbs.gz`.

```
# ------------------------------------------------------------------
# Initialisierung von  CDCN:  Basisobjekt = CDCNdss
# ------------------------------------------------------------------
# Zuerst werden die ben"otigten Skripten geladen
source ../cdcn_desc/cdcn.tcl
```

```
#Dies ist der eigentliche Initialisierungsaufruf
cdcnInit        P3   -dssdesc  cdcn_desc/cdcnDistribSet \
                     -dssparam cdcn_create_melv/3i.dss.gz \
                     -cbsdesc  cdcn_desc/cdcnCodebookSetMel \
                     -cbsparam cdcn_create_melv/3i.cbs.gz
# ------------------------------------------------------------------
```

Damit ist die Initialisierung abgeschlossen. Im Betrieb wird CDCN hauptsächlich in "Featuredescription" Dateien eingesetzt. Dies sind separate **TCL**-Skripten die die gesamte Signalvorverarbeitung definieren. Die Arbeitsfunktion heißt `cdcn` und ist eine Methode des Objekts `CDCNdss` vom Typ **DistribSet** welches bei der Initialisierung angelegt wurde.

Der Aufrufsyntax von `cdcn` ist:

`<DistribSet> cdcn <CDCNKodebuch> <Pausenkodebuch> <Segment> [Parameter]`

Als optionale Parameter stehen zur Verfügung:

- `-itcout <Zahl>` legt die Anzahl der Iterationen fest.
- `-init <Zahl>` legt die Art und Weise der Initialisierung der Startwerte fest.
  - *init* = 10 Der Startwert für q ist der Mittelwert des Sprachsegments. Der Startwert von n wird durch das zweistufige Verfahren bestimmt.
  - *init* = 3 Der Startwert für q und n ist Null.
  - *init* = 2 Der Startwert für q ist der Mittelwert des Sprachsegments. Der Startwert von n ist Null.
- `-n <Feature>` gibt den Schätzwert für n zurück.
- `-q <Feature>` gibt den Schätzwert für q zurück.
- `-f <Feature>` gibt die a posteriori Wahrscheinlichkeiten zurück.

Ein Beispiel für den Aufruf von `cdcn`. Die Objekte `CDCNdss`, `CDCN` und `SIL(|)` wurden bei der Initialisierung angelegt. Im Beispiel werden 10 Iterationen ausgeführt. Der Startwert von q ist der Mittelwert des Sprachsegments. Der Startwert von n wird durch das zweistufige Verfahren bestimmt. Das Sprachsegment `MEL` wird dabei bearbeitet. Das Ergebnis wird in das in der Kodebuchbeschreibung festgelegte Ziel geschrieben.

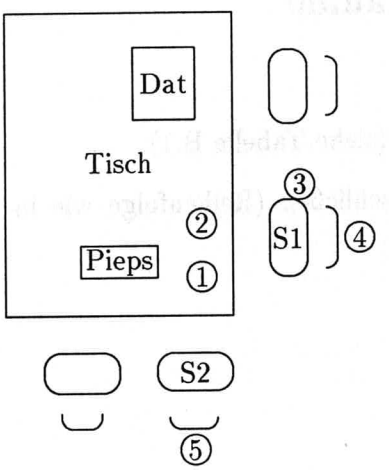
```
CDCNdss cdcn     CDCN    SIL(|)   MEL -init 10  -itcount 10
```

Anhang B

Checkliste für Mehrkanalaufnahmen

Um vergleichbare Aufnahmebedingungen bei mehreren Aufnahmesitzungen zu bekommen wurde folgende Checkliste erstellt. Bei jedem erneuten Aufstellen der Aufnahmegeräte wurde anhand dieser Liste vorgegangen.

B.1 Checkliste für die Multi-Mikro Aufnahmen



S1 Position von Sprecher 1.

S2 Position von Sprecher 2.

1 Mikrofon Nr. 1 Audio Technica AT 9750  .

2 Mikrofon Nr. 2 Audio Technica AT 9820X  .

3 Mikrofon Nr. 3 Audio Technica Ansteckmikro  .

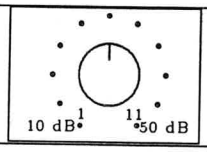
	Audio Technica AT 9750 	Audio Technica AT 9820X 	Audio Technica Ansteckmikro 	Headset Sennheiser HME 1410K (Sprecher 1) 	Headset Sennheiser HMD 410 (Sprecher 2) 
Mikrofon Nr.	1	2	3	4	5
	8	9	8	8	9

Tabelle B.1: Einstellung der Verstärkung am Vorverstärker.

4 Mikrofon Nr. 4 Headset Sennheiser HME 1410K (Sprecher 1)  .

5 Mikrofon Nr. 5 Headset Sennheiser HMD 410 (Sprecher 2)  .

B.2 Aufstellen und Inbetriebnahme

1. Dat-Recorder auf den Tisch stellen.
2. Verstärkung am Vorverstärker einstellen (siehe Tabelle B.1).
3. Mikrofone 1 bis 5 am Vorverstärker anschließen (Reihenfolge wie in Tabelle B.1 dargestellt).
4. Bei Mikrofon 1 muß der Schalter auf der Rückseite auf OMNI stehen. 
5. Bei Mikrofon 2 muß der Schalter auf der Oberseite auf ON stehen. 
6. Stromversorgung an den Dat-Recorder anschließen.
7. Stromversorgung für Piepser (Steckernetzteil) anschließen.
8. Dat-Recorder einschalten.
9. Dat-Band einlegen.
10. Vorverstärker einschalten.
11. Der Dat-Recorder muß mit doppelter Geschwindigkeit laufen. (so oft auf Speed drücken bis Lampe für 2-fache Geschwindigkeit leuchtet).

12. Die zuletzt geschriebene Id anfahren: [Search] ..... [Id]
    [Up] oder [Down] oder [Ch/Next] ..... [Id].
13. Warten bis Lampe neben [Search] nicht mehr blinkt.
14. ID auf den angefahrenen Wert setzen: [Id] [Up] oder [Down] oder [Ch/Next] [Id].
15. Recorder für die Aufnahme vorbereiten: [Rec] .... [Pause].
16. Die Lautstärke des Piepsers einstellen: (grün) gedrückt halten und Lautstärkeregler am Piepser solange nach rechts drehen bis ein deutlicher Ausschlag auf Kanal 1 des Dat-Recorders sichtbar ist.
17. Wenn die grüne Diode des Piepsers leuchtet dann einmal auf die grüne Taste (grün) drücken.

B.3 Aufnehmen

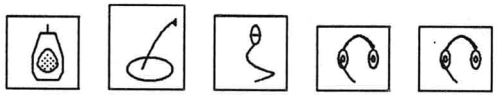
Für jede Aufgabe (Session) muß dieser Block wiederholt werden.

- Id Notieren.
- Session Start: [Pause] .... (grün) .... (die grüne Diode des Piepsers leuchtet).
- Der Sprecher muß die rote Taste (rot) solange er spricht gedrückt halten.
- Sprecher 1 beginnt darauf folgt Sprecher 2 u.s.w. . Für jedes aufzunehmende Utterance wird Folgender Vorgang wiederholt: [(rot) drücken .... sprechen .... (rot) loslassen]
- Session Ende: (grün) (die grüne Diode ist erloschen) .... [Id] .... [Pause].

B.4 Aufnahme beenden

1. Id notieren.
2. [Pause] .... 15 Sekunden warten .... [Stop].

B.5 Ausschalten

1. Vorverstärker ausschalten.
2. Dat-Band mit [Eject] auswerfen.
3. Dat-Recorder ausschalten.
4. Stromversorgungsstecker ziehen.
5. Mikrofone aus dem Vorverstärker ziehen  .
6. Recorder und alles andere wieder aufräumen.

Anhang C

Formular für die Aufnahmen

Zu jedem Sprecherpaar wurde das folgende Formular ausgefüllt.

Formular für die Multi-Mikro Aufnahmen

Sammler: _____

Dat-Band: _____ Datum: _____ Zeit: _____

A Sprecher 1: _____ Frau/Mann (nichtzutreffendes streichen)

B Sprecher 2: _____ Frau/Mann (nichtzutreffendes streichen)

ID

2. Dialog: _____

3. Dialog: _____

4. Dialog: _____

(Für Dialog Nr. 4 bitte Monat Januar verwenden!)