Digitalisierung von Audio

Albert-Ludwigs-Universität Freiburg
Praxis-Seminar Telekommunikation WS 05/06
Lehrstuhl für Kommunikationssysteme
Prof. Dr. Gerhard Schneider
Betreuer: Dirk von Suchodoletz

Sanja Jahnke
30.03.2006

Inhaltsverzeichnis

1. Einführung ... 2
2. Digitalisierung ... 2
2.1 Signale ... 2
2.2 A/D-Wandler ... 3
2.3 Das Abtasttheorem von Nyquist ... 4
2.4 Aliasing ... 5
2.5 Auflösung ... 5
2.6 Der Klirrfaktor ... 5
2.7 Datenmenge und Speicherbedarf ... 6
3. Datenkompression ... 6
3.1 Redundanz ... 7
3.2 Verlustfreie Kompression ... 8
3.3 Verlustbehaftete Kompression ... 8
4. Codecs ... 9
4.1 Sprachkodierungsmethoden ... 9
4.1.1 Waveform Codecs ... 9
4.1.2 Source Codecs ... 10
4.1.3 Hybrid Codecs ... 11
4.1.3.1 Linear Prediction Coding (LPC) ... 12
4.1.3.2 Long-Term Prediction (LTP) ... 12
4.1.3.3 Multi-Pulse Excited (MPE) und Regular-Pulse Excited (RPE) ... 12
4.1.3.4 Code Excited Linear Prediction (CELP) ... 12
4.2 ITU Standards für Sprache ... 13
4.2.1 Der G.711-Standard ... 13
4.2.2 Der G.722-Standard ... 13
4.2.3 Der G.723.1-Standard ... 13
4.2.4 Der G.728-Standard ... 13
4.2.5 Der G.729 und G.729A-Standard ... 13
4.3 ISDN Standard ... 14
4.4 GSM Standards ... 14
4.4.1 Die GSM-Sprachübertragung ... 14
4.4.2 Der Full-Rate (FR) Codec ... 15
4.4.3 Der Enhanced Full-Rate (EFR) Codec ... 16
4.4.4 Der Half-Rate (HR) Codec ... 16
4.4.5 Der Adaptive Multi-Rate (AMR) Codec ... 16
4.5 UMTS Standard ... 17
5. Fazit ... 18
Quellenangabe ... 19

1. Einführung

Die reale Welt, wie wir sie hören und sehen, strahlt kontinuierliche Wellen aus – Ton und Licht –, die in unseren Ohren und Augen zu dem werden, was wir wahrnehmen. Doch diese kontinuierlichen analogen Signale können nicht direkt auf digitalen Systemen wie Computern und Mobiltelefonen gespeichert bzw. bearbeitet werden. Man muss sie zuerst in digitale Werte umwandeln. Dies geschieht im A/D-Wandler und wird in *Abschnitt 2: Digitalisierung* erläutert.

Die so gewonnenen digitalen Daten sind aber meistens zu umfangreich, um sie auf dem PC zu speichern oder über Internet oder Mobilfunknetze zu übertragen. Die Lösung dafür bieten Komprimierungsverfahren, wobei zwischen verlustfreien und Verlust behafteten Verfahren unterschieden wird. Sie werden in *Abschnitt 3: Datenkomprimierung* behandelt.

Für die verschiedenen Ansprüche an die Ton-/ Sprachqualität und die unterschiedlichen zur Verfügung stehenden Bandbreiten wurden zahlreiche Sprachkodierungsmethoden zur Komprimierung entwickelt. Diese werden von den in Mobiltelefonen integrierten CODECs (COder/DECoder) angewendet, welche die Sprache vom Mikrophon kodieren und nach der Übermittlung wieder dekodieren und über den Lautsprecher ausgeben *(Abschnitt 4: Codecs)*.

2. Digitalisierung

Medien, wie Töne aus dem Mikrophon oder der heimischen Stereoanlage, Bilder und Video, existieren in der realen Welt als analoge kontinuierliche Schwingungen. Damit diese Medien computerisiert bearbeitet und über Internet oder Mobilfunknetze übertragen werden können, müssen die analogen Spannungswerte digitalisiert werden. Dabei müssen einige Regeln beachtet werden, die dieses Kapitel behandelt.

2.1 Signale

Signale sind physikalische Träger zur Vermittlung von Daten.[1] Das Maximum eines Signals heißt Amplitude, und die Frequenz ist die Anzahl der Perioden pro Sekunde, die zumeist in Hz gemessen werden.

Signale können nach Wert und Zeit klassifiziert werden:

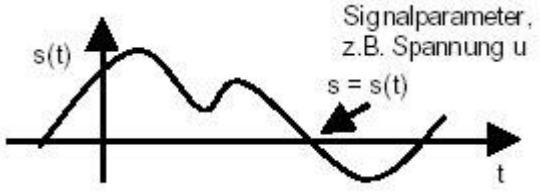


Abbildung 2.1: Wert- und zeitkontinuierliches Signal. [1]

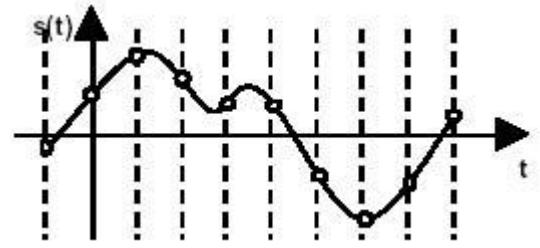


Abbildung 2.2: Wertkontinuierliches und zeitdiskretes Signal. [1]

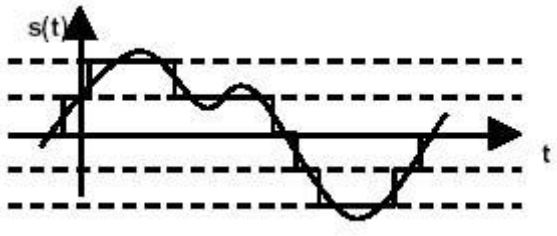


Abbildung 2.3: Wertdiskretes und zeitkontinuierliches Signal. [1]

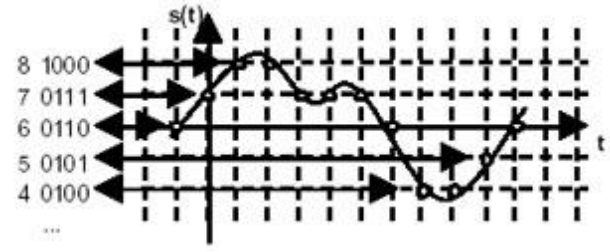


Abbildung 2.4: Wert- und zeitdiskretes Signal. [1]

[1] nach DIN 44 300

Ein rein analoges Signal ist wert- und zeitkontinuierlich, d.h. es bildet einen kontinuierlichen Vorgang – Ton oder Bild – kontinuierlich ab und benutzt dafür eine kontinuierliche zeitliche oder räumliche Domäne [siehe Abbildung 2.1].

Ein Abtastsignal ist wertkontinuierlich und zeitdiskret. Man erhält es, indem man ein analoges Signal an diskreten Zeitpunkten abtastet, d.h. dass man an gleichmäßig verteilten Abtastpunkten (Samples) die Signalwerte misst, und so die zeitkontinuierliche Spannung in zeitdiskrete Werte umwandelt [siehe Abb.2.3]. Abtastsignale werden auch Pulsamplitudenmodulation (PAM) genannt.

Ein quantisiertes Signal ist zeitkontinuierlich und wertdiskret, da der Wertebereich des Signals in eine endliche Anzahl diskreter Intervalle unterteilt und jedes Intervall jeweils durch einen Wert repräsentiert wird [siehe Abb.2.2].

Ein rein digitales Signal ist wert- und zeitdiskret [Abb.2.4]. Die Digitalisierung eines analogen Signals geschieht in drei Schritten: zuerst wird es abgetastet, dann quantifiziert, und anschließend die Abtastrate binär verschlüsselt [siehe Abb.2.5]. Dabei werden unterschiedliche Abtastraten (auch Samplefrequenzen genannt) und Quantisierungsbereiche benutzt, welche von den Tabellen 2.1 und 2.2 aufgelistet werden.

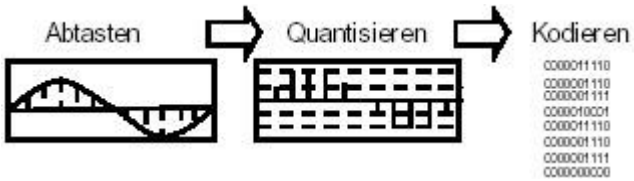


Abbildung 2.1: Umwandlung analoger Signale in digitale Werte [1]

Tabelle 2.1: Typische Abtastraten und Quantisierungsbereiche [1]

Audio	Abtastrate	Quantisierungsbereich
Telephon	8 kHz	8 Bit
CD Audio	44,1 kHz	16 Bit

Tabelle 2.2: Typische Abtastraten [9]

Hz	System
8000	SUN, digitales Telefon
8013	NeXT
8195	Atari Falcon
11127	Mac
12517	Atari STE/TT/Falcon
18900	CD-ROM/XA-Standard
22000	PC
22050	0.5 mal CD-Standard
22254	Mac
24000	Mac
25033	Atari STE/TT/Falcon
33880	Atari Falcon
44100	CD
48000	DAT, Mac
49170	Atari Falcon

2.2 A/D-Wandler

Ein Analog/Digital-Wandler (oder kurz A/D-Wandler) übernimmt die Aufgabe analoge Signale abzutasten, zu quantisieren und zu kodieren, damit sie digital verarbeitbar sind. Früher musste er extern am Computer angeschlossen werden, doch heutzutage ist die nötige Hardware in Form der Soundkarte bereits eingebaut. Soundkarten beinhalten ebenfalls das Gegenstück zum A/D-Wandler, den D/A-Wandler. Dieser macht aus den digitalen Daten wieder analoge Spannungen, die dann – über Lautsprecher ausgegeben – die Geräuschkulisse von Computerspielen oder die Klänge von Stereoanlagen ergeben.

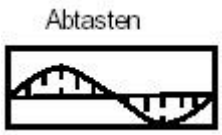
Vor den A/D-Wandler wird eine *Eingangsstufe* geschaltet, welche die unterschiedlichen Eingangssignale an die Anforderungen des Wandlers anpasst. Sie besteht aus verschiedenen Vorverstärkern, die die Signalpegel der unterschiedlichen Audioquellen (Mikrofon, CD, Tonbandgerät) anpassen, und Filtern, die unerwünschte Frequenzen aus dem Signal herausfiltern (siehe Abschnitt 2.4: Aliasing).

Der A/D-Wandler selbst enthält außer dem eigentlichen Wandler auch die so genannte *Sample-and-Hold-Stufe*, die den Wandler in der Abtastphase unterstützt. Sie nimmt zu festgelegten Zeitpunkten Proben aus dem Analogsignal und speichert sie, da die Änderungen in den Audiosignalen teilweise schneller sind als der Wandler, so dass sich der zu messende Wert während der Messung ändern würde.

Für die Quantisierungsphase unterteilt der A/D-Wandler seinen Messbereich in eine Anzahl von gleichgroßen Abschnitten bzw. Spannungsstufen. Dann vergleicht er den Spannungswert des Eingangssignals mit den durch seine Unterteilung definierten Spannungswerten der Stufen und kodiert ihn dementsprechend. Dies wird realisiert durch *Komparatoren* (Vergleicher), die gleichmäßig über den Messbereich verteilt sind und Signal geben (durch Spannung am Ausgang), sobald ihre Schaltschwelle überschritten wird. Der höchste Signal gebende Komparator entspricht dann dem gerundeten Wert des Eingangssignals, das dann nur noch entsprechend kodiert werden muss. Je nach Anzahl der Schaltstufen (siehe Abschnitt 2.5: Auflösung) des A/D-Wandlers werden 256, 65536 oder mehr Komparatoren verwendet. Das ist selbst heutzutage noch ein sehr großer Aufwand, da alle Komparatoren aufeinander abgestimmt sein müssen.

Da die Wandlung analoger Spannungen in digitale Werte nicht ohne Zeitverlust möglich ist, sind kontinuierliche Messungen nicht möglich. Deshalb behilft man sich damit, das Signal in regelmäßigen Abständen abzutasten. Dieser Umstand führt zu einer Reihe von Problemen, bei deren Erkennung und Vermeidung die Wahl der richtigen Abtastrate, oder auch Sample-Frequenz, von entscheidender Bedeutung ist. Die wichtigsten dieser Probleme werden im Folgenden noch angesprochen.

2.3 Das Abtasttheorem von Nyquist



Bei der Wahl der Sample-Frequenz (Abtastrate) ist das Abtasttheorem von entscheidender Bedeutung. Es besagt: Wenn man ein Signal mit maximal auftretender Frequenz $\mathbf{f_{max}}$ digitalisieren will, muss die Abtastrate mindestens $\mathbf{2f_{max}}$ sein, damit das Signal wieder angenähert rekonstruiert werden kann. Für die Abtastfrequenz $\mathbf{f_a}$ muss folglich gelten:

$$\mathbf{f_a \geq 2 \cdot f_{max}}.$$

Und wenn die Sample-Frequenz richtig gewählt wurde, entspricht die gedachte Verbindungslinie durch die Messpunkte dem Verlauf des ursprünglichen Signals.

Dementsprechend benötigt man für ein analoges Signal mit 4 kHz eine Sample-Frequenz von mindestens 8 kHz, und CDs werden in der Regel mit 44,1 kHz digitalisiert, um einen Frequenzbereich von bis zu 22 kHz zu gewährleisten [siehe auch Tab.2.1 & 2.2].

Die Grundlage des Theorems erkennt man leicht, indem man testet was geschieht, wenn man es nicht befolgt. Wird ein Signal mit einer zu niedrigen Frequenz (d.h. zu langsam) abgetastet, geht zuviel Information über die Originalfrequenz verloren, und bei der Rekonstruktion entstehen falsche Frequenzteile [Abb.2.6].

Auch die Wahl einer Abtastrate von genau $2f_{max}$ ist nicht ganz problemfrei, denn dann müssen die Abtastpunkte genau bei den Minima und Maxima liegen. Falls sie das nicht tun, werden zu wenige Signalwerte erfasst, um eine adäquate Rekonstruktion zu ermöglichen, wie Abbildung 2.7 zeigt.

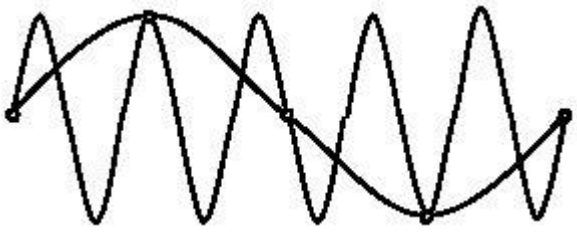


Abbildung 2.6: Abtasten unterhalb der Nyquist-Rate. Das hochfrequente Originalsignal wird zu langsam abgetastet (hier: an den fünf Punkten). So entsteht bei dem Rekonstruktionsversuch – dem Verbinden der Punkte – eine falsche Frequenz. [1]

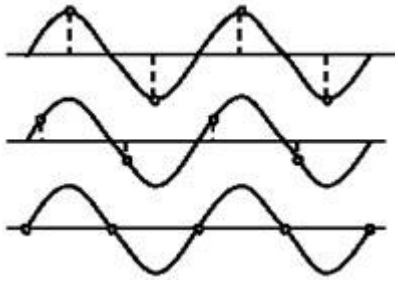


Abbildung 2.7: Abtasten mit Nyquist-Rate. Wenn die Abtastpunkte nicht genau bei den Minima und Maxima liegen, wird zu wenig Information erfasst. [1]

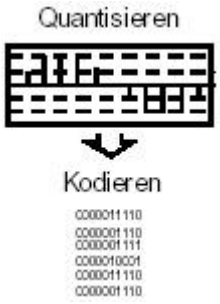
2.4 Aliasing

Wenn ein Signal mit einer Sample-Frequenz von weniger als $\mathbf{2f_{max}}$ abgetastet wird, entstehen Frequenzen, die im Originalsignal nicht vorhanden sind, so genannte *Spiegelfrequenzen*. Es gehen folglich nicht nur Frequenzen des Originals verloren, sondern es werden auch neue – falsche – hinzugefügt, welche die Klangqualität erheblich stören können. Dieser Effekt heißt *Aliasing*.

Jedoch beinhalten praktisch alle natürlichen Tonsignale Frequenzen, die sogar über den hörbaren Bereich von 20kHz hinausgehen, weshalb nur eine unendlich hohe Abtastrate das Aliasing verhindern könnte. Um die Abtastrate begrenzen zu können, wird ein *Anti-Aliasing-Filter* eingesetzt, der nur die Frequenzen bis zur Hälfte der verwendeten Sample-Frequenz durchlässt und die zu hohen Frequenzen entfernt. D.h. für verschiedene Sample-Frequenzen müssen verschiedene Filter benutzt werden.

Bei kommerziell erhältlichen Samplern ist das üblicherweise mehr oder weniger gut berücksichtigt. Z.B. werden bei PC-Soundkarten häufig Wandler verwendet, in denen die entsprechenden Filter in den Wandlungsprozess von analog nach digital integriert sind. Falls solche Filter fehlen, entstehen höhere Frequenzen, die sich als eine Art Pfeifen oder Klirren bemerkbar machen.

2.5 Auflösung



Die Auflösung bezeichnet die Anzahl der Stellen der binären Zahlen, die beim Digitalisierungsprozess zur Repräsentation der Messwerte des analogen Signals benutzt werden.

Der Messbereich wird in viele gleichgroße Spannungs-Stufen (Quantisierungsstufen) aufgeteilt, wobei die Anzahl der Stufen durch die Anzahl der unterschiedlichen Zustände gegeben ist, die durch die gewählte binäre Darstellung repräsentiert werden können. Z.B. sind mit einer 8 Bit Auflösung $2^8 = 256$ Quantisierungsstufen kodierbar. Je feiner wir messen wollen, desto größer muss die Auflösung gewählt werden.

Bis auf ganz wenige Ausnahmen werden in der aktuellen Audio-Digitaltechnik Auflösungen verwendet, deren Anzahl der Stellen ein Vielfaches von 8 ist, da dies der Größe eines Bytes entspricht.

In der Regel kann man für die analogen Signale einen Spannungsbereich von -2,5 V bis +2,5 V annehmen, denn der Eingangsbereich eines gebräuchlichen A/D-Wandlers beträgt ca. 5 V. Für eine Auflösung von 8 Bit (also 256 Stufen) bedeutet dies, dass pro Stufe ein Bereich von ungefähr 19,53 mV abgedeckt wird, und bei einer 16 Bit Auflösung (65536 Stufen) ca. 0,07 mV/Stufe.

2.6 Der Klirrfaktor

Bei der Umwandlung der analogen Signale in digitale Werte ist die Auflösung von entscheidender Bedeutung, denn die Quantisierung der Werte führt bei der Wiedergabe zu einem Klirren bzw. Rauschen, dem so genannten Quantisierungsrauschen. Dieses entsteht dadurch, dass alle Messwerte eines Signals, die zwischen den Werten liegen, die den einzelnen Quantisierungs-stufen entsprechen, jeweils auf bzw. abgerundet werden müssen. Je höher die Auflösung ist, desto kleiner werden die Fehler und das Klirren nimmt ab.

Als Maß für das Klirren wird in der Audiotechnik der Klirrfaktor benutzt. Den minimalen Klirrfaktor kann man direkt aus der Anzahl der Quantisierungsstufen q berechnen:

$$\text{Klirrfaktor } k = \frac{1}{\sqrt{q^2 - 1}}$$

Dazu kommen noch die Fehler des Analogkreises, und da das Signal nur selten voll ausgesteuert wird, werden weniger Stufen benutzt, wodurch k ansteigt.

Der Signalgeräuschabstand oder Fremdspannungsabstand hängt ebenfalls direkt von der Anzahl der Quantisierungsstufen ab:

$$\text{Signalrauschabstand } s = 10 \log\left(\frac{1}{k^2}\right) \text{ [dB]}$$

Doch viele Computeranwender können sich ihre Auflösung nicht aussuchen, weil ihre Sampler nur 8 Bit zulassen. Sprachsignale können damit aber schon gut aufgezeichnet werden, da hierbei erst ein Klirrfaktor von über 5% als störend angesehen wird. Für Hifi-Anwendungen dagegen sind 16 Bit gerade gut genug, denn Verstärker haben (analog bedingte) Klirrfaktoren von ca. 0.008%. Auch CD-Player liegen im Bereich von 0.004%, und besser als 0.0015% geht es theoretisch nicht. Die typischen Auflösungen sind in Tabelle 2.3 aufgelistet.

Tabelle 2.3: Auflösung q, Klirrfaktor k und Signalrauschabstand s für typische Datenformate. [2]

Bits	**1**	**4**	**8**	**12**	**16**
q	2	16	256	4096	65536
k	58%	6,26%	0,39%	0,024%	0,0015%
s	4,8	24	48	72	96

2.7 Datenmenge und Speicherbedarf

Bei der Arbeit mit Samples entstehen erhebliche Datenmengen. Wer digitale Audiosignale speichert oder gar per Mail-Anhang versenden möchte bekommt das recht eindrucksvoll zu spüren. Nimmt man beispielsweise auf dem PC Musik in CD-Qualität und Stereo auf, fallen 200 KByte/s an. So belegt eine Single relativ schnell 30 - 50 MByte. Die Größe der Datenmenge hängt von Sample-Frequenz und Auflösung ab, je differenzierter und originalgetreuer wir unsere Aufnahmen machen wollen, desto mehr Daten müssen gespeichert werden [s. Tab. 2.4 & 2.5].

Tabelle 2.4 & 2.5: Speicherbedarf pro Minute in Abhängigkeit von Abtastrate und Auflösung.

Tabelle 2.4: Speicherbedarf in MiByte pro Minute. [2]

Abtastrate	**8 Bit Mono**	**8 Bit Stereo**	**16 Bit Mono**	**16 Bit Stereo**
8000	0,45	0,91	0,91	1,83
11050	0,62	1,26	1,26	2,52
22050	1,26	2,52	2,52	5,05
44100	2,52	5,05	5,05	10,09

Tabelle 2.5: Speicherbedarf in MByte pro Minute. [9]

Abtastrate	**8 Bit Stereo**	**16 Bit Stereo**	**24 Bit Stereo**	**32 Bit Stereo**
22050	2,58	5,16	7,75	10,34
44100	5,16	10,34	15,50	20,67
48000	5,63	11,25	16,87	22,50
96000	11,25	22,50	33,75	45,00

3. Datenkompression

Beim Digitalisieren analoger Signale entstehen je nach Qualität mehr oder weniger große Sample-Dateien. Ein unkomprimiertes Stereo-Audiosignal in CD-Qualität – abgetastet mit 44,1 kHz, quantisiert mit 16 Bit – benötigt 10,09 MByte/min; ein Video mit einer Framerate von 25 Bildern/s und einer Auflösung von 640 · 480 Pixeln kommt leicht auf 22 MByte/s.

Frühere Rechnergenerationen konnten diesen enormen Speicherbedarf nicht decken, und auch heutzutage ist es z.B. für Internetanwendungen und Mobiltelefonie vorteilhaft und notwendig, wenn die Audio- oder Videodateien kleiner sind. Deshalb gibt es eine Reihe von Methoden diese Daten zu komprimieren, wobei die Klangqualität teilweise auch erheblich abnimmt. Denn im Gegensatz zu herkömmlichen Packprogrammen, die lediglich die Redundanzen entfernen und somit verlustfreie Datenkompression anwenden, wird bei Komprimierung von Audio und Video die Information wirklich reduziert. Das bedeutet, dass ein begrenzter Datenverlust in Kauf genommen wird. Im Folgenden werde ich mich (zur Datenreduktion) auf die Komprimierung von Audio beschränken.

3.1 Redundanz

Der Begriff Redundanz (v. lat. *redundare* – im Überfluss vorhanden sein) bezeichnet allgemein das mehrfache Vorhandensein funktions-, inhalts- oder wesensgleicher Objekte. Ein Teil einer Nachricht ist dann redundant, wenn es ohne Informationsverlust weggelassen werden kann, z.B. weil er implizit oder explizit schon vorher in der Nachricht gegeben wurde.

Bei (geschriebener) Sprache gibt es zwei Arten von Redundanzen: *Verteilungsredundanz* bezeichnet das unterschiedlich wahrscheinliche Auftreten der einzelnen Zeichen des Alphabets. Und die *Bindungsredundanz* liegt darin, dass nach bestimmten Zeichen ein bestimmtes anderes Zeichen mit besonders hoher Wahrscheinlichkeit auftritt. Z.B. folgt in einem deutschen Text auf ein q fast immer ein u. 73% der deutschen Sprache ist redundant, und nur 27% trägt Information, doch gerade diese Redundanz ist durchaus erwünscht, da dadurch bei gestörten Nachrichten eine Fehlerkorrektur und -erkennung ermöglicht wird.

Gesprochene Sprache enthält ebenfalls Redundanzen, wobei zwischen kurz- und langfristigen unterschieden wird. Sie entstehen aufgrund der Methode, mit der Menschen Sprache erzeugen: Der Ton entsteht durch die Vibration der Stimmbänder, zwischen denen Luft hindurchströmt. Bei Männern entstehen auf diese Art typischerweise Frequenzen von 80- 160 Hz und bei Frauen, die zumeist eine höhere Stimme haben, 180- 320 Hz. Der Klang des Tons wird durch Veränderung des Stimmraumes kontrolliert. Diese Veränderung wird z.B. durch Zungenbewegungen erreicht; jeder Laut entsteht aufgrund einer spezifischen Bewegung oder Position der Zunge, beispielsweise liegt die Zungenspitze bei ‘l’ immer am oberen Gaumenrand. [Den Sprachraum modellieren einige Codecs als Filter; s. Abschnitt 4.1.2]

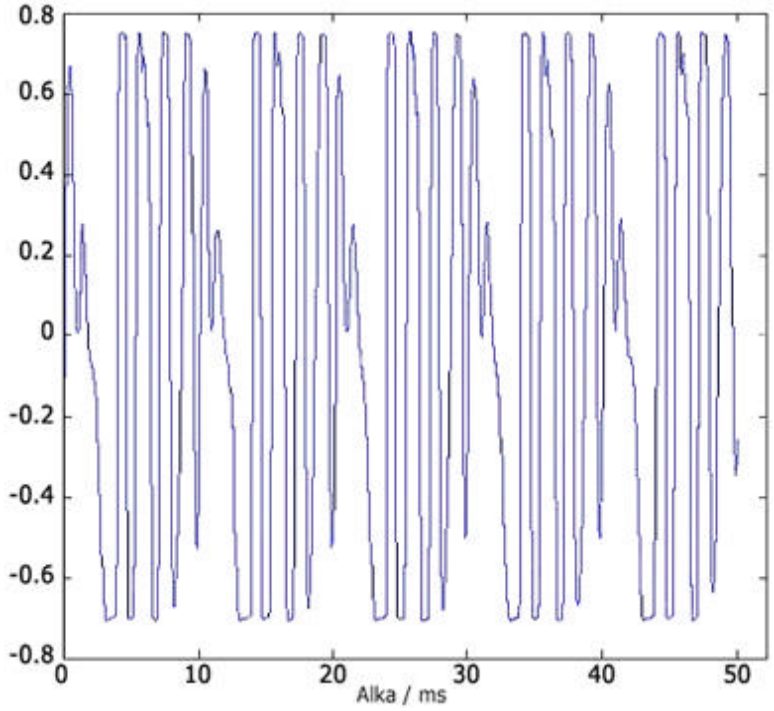


Abbildung 3.1 Sprachbeispiel eines Mannes: ‘aaa…’. [3]

Die *kurzfristige Korrelation* oder Redundanz entsteht wegen der Struktur des Stimmraumes (Stimmbänder,…), die *langfristige*, weil das Sprachsignal von Natur aus periodisch ist. Abbildung 3.1 zeigt diese Periodizität: in diesem Beispiel wiederholt sich das Signal ungefähr alle 10 ms.

3.2 Verlustfreie Kompression

Bei verlustfreier Kompression werden hauptsächlich Redundanzen entfernt, wodurch eine Kompressionsrate von 2:1 bis 50:1 erreicht werden kann. Sie wird beispielsweise bei Faxübertragungen angewendet und ist Grundlage von Unix compress und winzip.

Universelle verlustfreie Kompressionsverfahren arbeiten ohne Kenntnis der Daten, wie z.B. die Lauflängenkodierung, die jede Serie von sich wiederholenden Zeichen durch einen Zähler und das sich wiederholende Zeichen ersetzt (aus AAAACCCBBBBB wird 4A3C5B).

Statistische Verfahren verwenden für häufiger vorkommende Zeichenfolgen kürzere Kodierungen; Beispiele dafür sind die Huffman-Kodierung und die Arithmetische Kodierung.

Bei Audiosignalen ist die durch Redundanzreduktion erreichbare Komprimierungsrate gering und stark von den statistischen Eigenschaften des Signals abhängig. Hier besteht ein Weg zur verlustfreien Kompression darin, alle nicht genutzten Bits wegzulassen. Das führt bei leisen Passagen zu einem höheren Kompressionsgrad als bei lauten. Eine andere Möglichkeit ist – ausgehend von einem Startwert – nur noch die Differenzen zwischen den einzelnen Abtastwerten zu speichern. Auch die Anpassung von mathematischen Gleichungen an den Signalverlauf (Vorhersage unter Ausnutzung der statistischen Redundanz) eignet sich zur Kompression von Audiodaten. Je nach Art der Daten und der Kompression lassen sich Kompressionsfaktoren im Bereich von 1 bis 5 erreichen.

3.3 Verlustbehaftete Kompression

Die Verlustbehaftete Kompression kommt hauptsächlich bei Audio- und Video-Dateien zum Einsatz und erreicht Kompressionsraten von über 100:1, typischerweise 50:1. Dabei gilt aber immer die Forderung, dass die Reduzierung der Datenmenge ohne einen wahrnehmbaren Informationsverlust erfolgen soll, da sonst der Qualitätsverlust des Audiosignals zu groß wäre. Um dies zu erreichen werden die Schwächen der menschlichen Wahrnehmung ausgenutzt, indem man aus dem Audiosignal die Informationen entfernt, die man ohnehin nicht hören würde. Dazu gehören insbesondere Frequenzen von unter 16 Hz oder über 20.000 Hz, die das menschliche Gehör nicht wahrnehmen kann. Die ZAIK/RRZK Multimedia Gruppe [2] nennt zum Thema der menschlichen Wahrnehmung die folgenden Begriffe:

Begriffe der Psychoakustik:

Ruhehörschwelle: Nur Töne oberhalb dieser Schwelle werden vom menschlichen Ohr wahrgenommen.

Mithörschwelle: leise Töne, die sich im Frequenzbereich in der Nähe von lauten Tönen befinden, werden ebenfalls vom menschlichen Ohr nicht wahrgenommen.

Verdeckungseffekt: Leise Töne werden durch laute, zeitlich vor- oder nacheilende Töne verdeckt.

Frequenzabhängige Lautstärkenempfindung: Das Ohr hat eine unterschiedliche Lautstärkenempfindung bei hohen und tiefen Frequenzen.

Für die Komprimierung von Audio bedeutet dies, dass z.B. das Quantisierungsrauschen toleriert werden kann, solange es unter der Mithörschwelle liegt. Ebenso kann der Verdeckungseffekt genutzt werden, indem man jeweils nur die lautesten Frequenzanteile benutzt und die leisen, verdeckten Töne entfernt. Dadurch kann die Anzahl der Quantisierungsstufen und somit die Auflösung reduziert werden. Das Gehör besitzt außerdem nur ein zeitlich begrenztes Wahrnehmungsvermögen für Lautstärkeschwankungen, das vom Schallpegel und der Frequenz abhängig ist. Dies bedeutet wiederum eine mögliche Reduzierung der Auflösung entsprechend der wahrnehmbaren Lautstärke- Änderung.

Diese Audio-Kompressionsmethode kommt auch im Telephoniebereich zur Anwendung, in den Codecs, welche im folgenden Abschnitt erläutert werden.

4. Codecs

Codecs kommen im Telephoniebereich – bei Internettelephonie ebenso wie bei Handys – und überall, wo Audiodaten in andere Formate umgewandelt werden müssen, zum Einsatz. Ein CODEC (COder/DECoder) komprimiert die Sprache, die der A/D-Wandler digitalisiert hat, um sie nach der Übermittlung auf der anderen Seite wieder zu dekomprimieren.

Da die verschiedenen Audiosignale, Sprachqualitätswünsche und Übertragungsbandbreiten unterschiedliche Anforderungen an die Kompression stellen, wurde eine Vielzahl von Codecs entwickelt, die entsprechend auch in Kombination mit unterschiedlichen A/D-Wandlern (betreffend Abtastfrequenz und Auflösung) benutzt werden. Codecs sind – als Hardware oder Software realisiert – teilweise für sehr eng begrenzte Einsatzbereiche optimiert. So gibt es Codecs die speziell für den Einsatz bei Sprachübertragungen abgestimmt sind und deshalb bei Musik zu vollkommen unbrauchbaren Ergebnissen führen und umgekehrt.

4.1 Sprachkodierungsmethoden

Die menschliche Sprache ist ebenso wie Tonaufnahmen auf Band oder CD ein analoges Signal, das mithilfe der Stimmbänder und des Stimmraums erzeugt wird [s. Abschnitt 3.1].

Sprachkodierung kann dementsprechend als Repräsentation eines analogen Signals durch eine Sequenz binärer Zahlen definiert werden. Die Grundidee dahinter ist das Ausnutzen spezieller Eigenschaften des menschlichen Sprachsystems: der statistischen Redundanz – kurz- und langfristig – und der eingeschränkten Fähigkeiten Töne wahrzunehmen.

Abbildung 4.1 illustriert die Grundidee der Sprachkodierung: Irrelevanz wird durch Quantisierung minimiert, und Redundanz durch Vorhersage entfernt. Dabei geht durch die Quantisierung Information verloren – auch wenn diese nur irrelevant ist –, wohingegen Vorhersage normalerweise alle Informationen des Signals erhält.

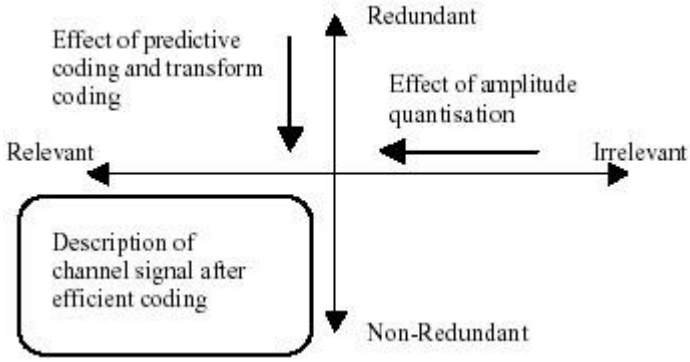


Abbildung 4.1: Grundidee hinter der Sprachkodierung. Quantisierung reduziert die Irrelevanz und Vorhersage die Redundanz. [3]

Es gibt drei verschiedene Sprachkodierungsmethoden, welche diese verschiedenen Eigenschaften auf unterschiedliche Weise nutzen: Waveform coding, Source coding und Hybrid coding.

4.1.1 Waveform Codecs

Waveform Codecs versuchen ein digitales Signal zu erzeugen, dass dem analogen Originalsignal so ähnlich wie nur möglich ist.

Pulse Code Modulation (**PCM**) ist der einfachste und reinste Waveform Codec und benutzt ausschließlich Sampling und Quantisierung zur Digitalisierung des Eingangssignals. Er benutzt eine Abtastrate von 8 kHz, da die Bandbreite in öffentlichen Telefonnetzen auf 4 kHz begrenzt ist (Nyquist-Regel). Bei linearer Quantisierung wäre eine Auflösung von 12 Bits pro Sample notwendig, um eine gute Sprachqualität zu erreichen. Dies würde eine Bitrate (Übertragungsrate) von 8 Bit · 12 kHz = 96 KBit/s bedeuten. Doch die Auflösung kann durch die Verwendung einer uneinheitlichen Quantisierung nach *A-law* auf 8 Bit reduziert werden (d.h. es wird nur eine Bitrate von 8 Bit · 8 kHz = 64 KBit/s benötigt), ohne dass die Qualität darunter leidet: Der PCM-Codec erzeugt aus einem 16-Bit-Abtastwert über eine logarithmische Kennlinie einen 8-Bit-Wert. Dieser wird nach dem Übersenden wieder in einen 16-Bit-Wert umgewandelt, dessen Qualität etwa der eines 14-Bit-Samples entspricht.

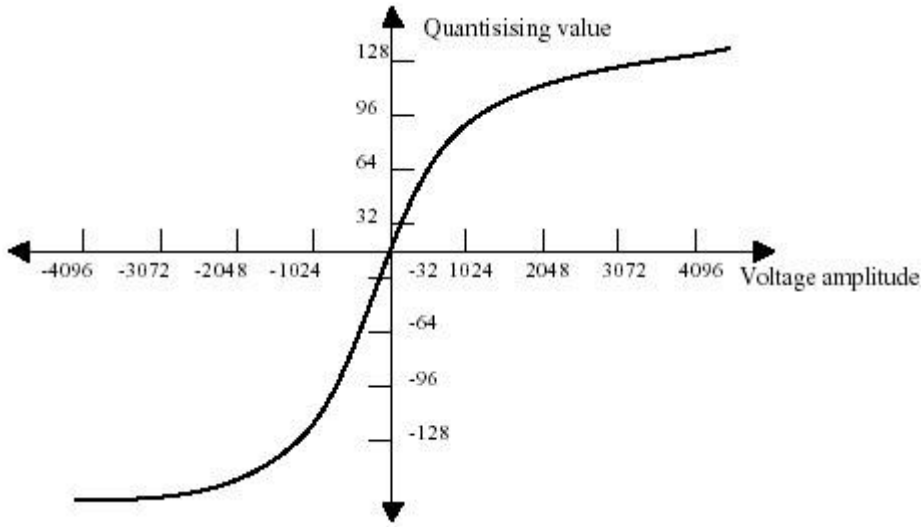


Abbildung 4.2: Eine Skizze der nicht-linearen Quantisierung nach A-law (Achsenwerte nicht maßstabsgetreu). [3]

Wegen dieser logarithmischen Komprimierung [s. Abb. 4.2] werden die Änderungen in leisen Passagen genauer wiedergegeben als in lauten. Das ist sinnvoll, da beispielsweise ein leises 16-Bit-Signal, das im Bereich von -100 bis +100 liegt bei einer Änderung der Amplitude um 1000 ungefähr doppelt so laut wird, aber die gleiche Änderung bei einem lauten Signal von ca. -10000 bis +10000 nur eine Zunahme der Lautstärke um ca. 5% bewirkt.

Eine Verbesserung des PCM-Codecs ist *Differential Pulse Code Modulation* (**DPCM**). Hierbei wird zusätzlich versucht aus den bisherigen Samplewerten den Wert des nächsten Samples vorherzusagen. Solche Vorhersage-Methoden sind in der Sprachkodierung sehr verbreitet und aufgrund der in Sprachsignalen vorhandenen Korrelation (Redundanz) möglich. Mit ihnen kann anstelle des aktuellen Signals ein Errorsignal r(n), das den Unterschied zwischen dem vorhergesagten Signal und dem aktuellen Signal enthält, benutzt werden: $\boldsymbol{r(n) = s(n) - s'(n)}$, wobei s(n) das Originalsignal und s'(n) das vorhergesagte ist.
Wenn die Vorhersage effektiv ist, hat das Errorsignal eine niedrigere Varianz als die ursprünglichen Sprachsamples. Gemessen wird dies in der Signal-to-Quantisation-Noise Ratio: $\boldsymbol{SNR = 10 \log (\sigma_s^2/\sigma_r^2)}$, mit σ_s^2 = Signalvarianz, σ_r^2 = Rekonstruktionsfehler-Varianz.
Eine niedrige SNR bedeutet, dass es möglich ist das Errorsignal mit weniger Bits zu quantisieren als das Originalsignal.

Die Effizienz des Vorhersage-Kodierungs-Schemas kann verbessert werden, wenn Vorhersage- und Quantisierungsfunktion sich über die Zeit ändern und so an die charakteristischen Eigenschaften des Sprachsignals angepasst werden. Dies führt zu *Adaptive Differential PCM* (**ADPCM**). ADPCM wird unter anderem dort in der Telefontechnik verwendet, wo µ-law noch zu große Datenmengen produziert, z.B. bei Voicemailing per Modem.

4.1.2 Source Codecs

Der grundlegende Unterschied zwischen Waveform und Source Codecs ist der, dass Waveform Codecs kein Wissen über die Quelle des Signals benutzen, sondern ein digitales Signal erzeugen, dass dem analogen Originalsignal so ähnlich wie nur möglich ist, wohingegen Source Codecs versuchen ein digitales Signal zu erzeugen, das die Quelle des Codecs modelliert. D.h., dass Source Codecs ein Modell benutzen, das darstellt wie die Quelle generiert wurde, und versuchen aus dem kodierten Signal die Parameter des Modells zu extrahieren. Diese Modellparameter werden dann zum Decoder übersandt.

Source Codecs für Stimmanwendungen werden **Vocoder** genannt und funktionieren folgendermaßen [s. Abb.4.3]: Das Gespräch wird in mehrere kleine Segmente unterteilt, wobei zwischen stillen (unvoiced) und stimmhaften (voiced) Segmenten unterschieden wird.

In den stimmhaften Segmenten werden von der Stimme ausgelöste Impulse zum Stimmraum (vocal tract) übermittelt, und während den stillen Segmenten wird weißes Rauschen, das während den Schweigemomenten das Knistern von Telefonleitungen imitiert, erzeugt. Der Stimmraum dient als Zeit- variierender Filter und sendet die notwendige Information weiter an den Decoder: die Filterspezifikation, ob es ein stilles oder ein stimmhaftes Segment ist, die notwendige Varianz des Signals und für stimmhafte Segmente die Stimmhöhe. Entsprechend der Natur von gewöhnlicher Sprache gibt es alle 10-20 ms es ein Update.

Die Parameter des Modells können vom Encoder auf unterschiedliche Weise bestimmt werden, mit Techniken, die entweder eine Zeit- oder eine Frequenz-Domäne benutzen. Und die Information kann für die Transformation auf viele verschiedene Arten kodiert werden.

Vocoders wurden hauptsachlich in militärischen Anwendungen benutzt, bei denen der natürliche Klang der Stimme nicht so wichtig war wie die sehr geringe Bitrate, die starken Schutz und gute Verschlüsselung erlaubt.

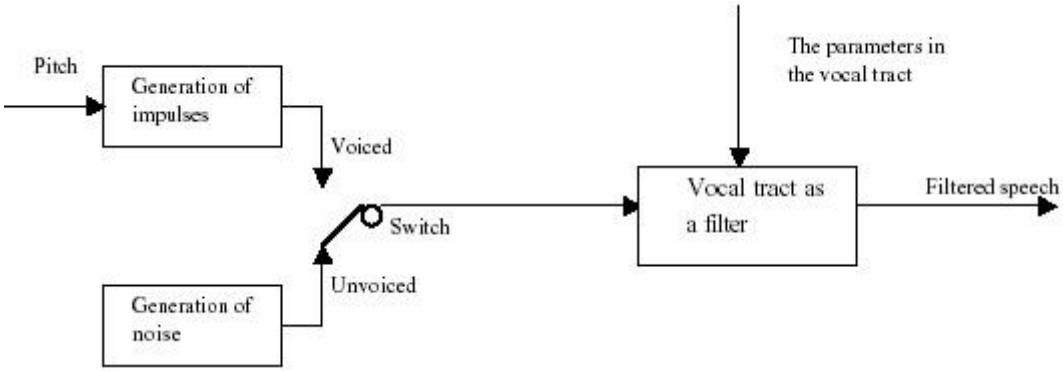


Abbildung 4.3: Der Sprach-Kreations-Prozess benutzt bei Source Coding. [3]

4.1.3 Hybrid Codecs

Hybrid Codecs versuchen die Lücke zwischen Waveform und Source Codecs zu füllen. Waveform Codecs liefern mit Bitraten von ungefähr 16 KBit/s eine gute Sprachqualität, aber bei niedrigen Bitraten sind sie nur von begrenztem Nutzen. Source Codecs hingegen können bei Bitraten von 2.4 KBit/s (und weniger) verständliche Sprache erzeugen, aber bei keiner Bitrate eine natürlich klingende Sprache liefern. Hybride Codecs kombinieren Techniken von Waveform und Source Codecs, wodurch bei mittleren Bitraten eine gute Qualität erzeugt wird. Die Qualität der verschiedenen Codecs ist auf Abbildung 4.4 dargestellt.

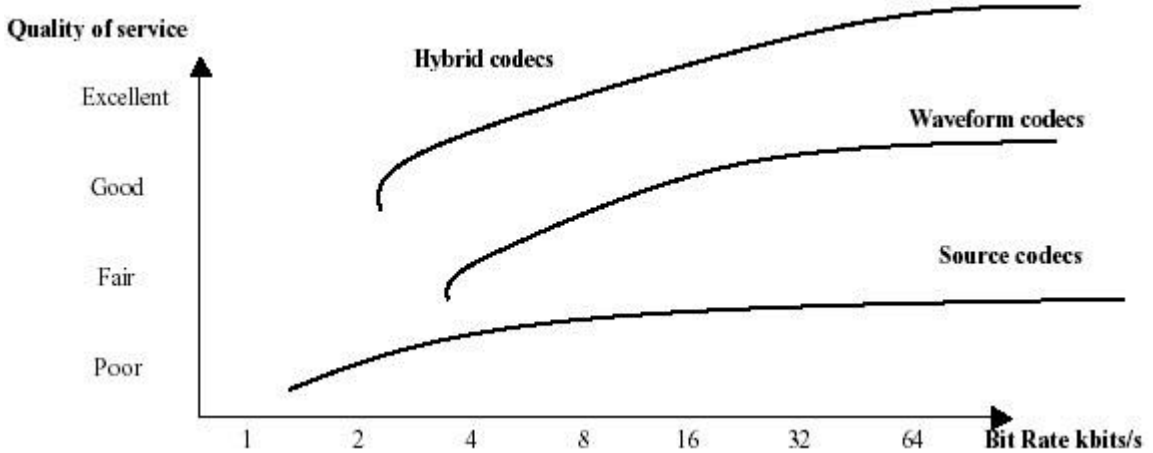


Abbildung 4.4: Sprachqualität als Funktion der Bitrate für die drei verschiedenen Sprachkodierungsmethoden. Durch Kombination der Techniken von Waveform und Source Codecs erzeugen Hybrid Codecs eine bessere Sprachqualität. [3]

4.1.3.1 Linear Prediction Coding (LPC)

Der erfolgreichste und meist benutzte Hybrid Codec-Typ ist *Analysis-by-Synthesis* (**AbS**), auch *Linear Prediction Coding* (**LPC**) genannt. Diese Codecs benutzen das gleiche lineare Vorhersage-Filter-Modell des Stimmraumes wie Source Codecs. Doch anstelle der Anwendung des einfachen 2-Zustandssystems (still/stimmhaft) wird ein Signal generiert, das die Wellenform des Originalsignals so gut wie möglich imitiert.

LPC-Codecs teilen die Inputsprache, die kodiert werden soll, in Frames von üblicherweise 20 ms. Für jeden Frame werden die Parameter für den Synthesefilter bestimmt. Um den Fehler zwischen Inputsprache und rekonstruierter Sprache möglichst gering zu halten, synthetisiert der Encoder viele verschiedene Approximationen der Inputsprache und analysiert sie dabei (daher der Name Analysis-by-Synthesis). Die Idee dahinter ist, dass jedes Sprachsample durch eine lineare Kombination der vorhergegangen Samples approximiert werden kann. Der Synthesefilter hat die Form: $\boldsymbol{H(z) = 1/A(z)}$, mit $\boldsymbol{A(z) = 1 + \Sigma a_i z^{-i}}$, i = 1,…,p. Die a_i sind Koeffizienten, die *Linear Prediction Coeffizient* genannt werden. Sie werden durch Minimierung des Unterschiedes zwischen dem aktuellen und dem vorhergesagten Signal bestimmt. Die Variable p gibt die Ordnung des Filters an. Dieser Filter ist für die Modellierung der kurzzeitigen Korrelationen der Sprache gedacht, d.h. für die Korrelationen zwischen Samples, die weniger als 16 Samples auseinander liegen.

4.1.3.2 Long-Term Prediction (LTP)

Um Codecs noch effizienter zu machen, muss die quasi-periodische Natur der menschlichen Sprache – die langzeitige Korrelation – ausgenutzt werden. Bei dieser *Long-Term Prediction* (**LTP**) werden die Korrelationen zwischen Samples, die 20-120 Samples weit auseinander liegen, ausgewertet. Die Transferfunktion ist: $\boldsymbol{P(z) = 1 + bz^{-N}}$, wobei N der Zeitraum der Basisfrequenz (der Stimmhöhe) ist, und b der *Linear Predictive Coeffizient*. N wird so gewählt, dass die Korrelation zwischen dem gesampleten Signal x[n] und dem Signal x[n+N] maximal ist.

4.1.3.3 Multi-Pulse Excited (MPE) und Regular-Pulse Excited (RPE)

Multi-Pulse Excited (**MPE**) und *Regular-Pulse Excited* (**RPE**) Codecs sind mit dem LPC-Codec verwandt. Doch bei MPE und RPE besteht das dem Filter übermittelte Signal für jeden Sprachframe aus einer festen Anzahl von Impulsen, die nicht Null sind. Anders als bei MPE werden bei RPE die Impulse gleichmäßig auf ein festgelegtes Intervall verteilt. Das bedeutet, dass der Encoder nur die Position des ersten Impulses und die Amplitude aller Impulse bestimmen muss, während bei MPE die Positionen und Amplituden aller Impulse festgelegt werden müssen. Folglich muss der RPE-Codec weniger Information an den Decoder übermitteln, was bei Mobilfunknetzen wie GSM, bei denen die Bandweite besonders knapp ist, von großer Bedeutung ist.

Obwohl MPE und RPE Codecs bei Raten von 10KBit/s und mehr eine gute Sprachqualität liefern, sind sie für niedrigere Raten nicht anwendbar. Das liegt an der großen Menge an Information über Impuls-Positionen und Amplituden, die übermittelt werden muss.

4.1.3.4 Code Excited Linear Prediction (CELP)

Der meistbenutzte Algorithmus für gute Sprachqualität bei Raten unter 10KBit/s ist *Code Excited Linear Prediction* (**CELP**). CELP-Codecs benutzten ein Codebuch, dessen Einträge die Parameter der menschlichen Stimme sind, die zur synthetischen Generierung eines Sprachsignals benötigt werden. Das an den Filter übermittelte Signal besteht dementsprechend aus einem Eintrag des großen *Quantiser Codebooks* und einem Term, der dessen Kraft (Power) kontrolliert.

4.2 ITU Standards für Sprache

Die ITU (Internationale Telekommunikations- Union) hat zur Schaffung eines Rahmenwerkes für multimediale Anwendungen existierende Standards zur ITU H.23X-Familie zusammengefasst. Zu den darin aufgenommenen Standards gehören unter anderem auch die G.7XX-Codecs zur Komprimierung der Sprache. Für die Sprachkodierung gibt es eine Reihe verschiedener Standards, die unterschiedliche Komprimierungsverfahren verwenden und daher auch unterschiedliche Bandbreiten auf der Übertragungsstrecke benötigen und unterschiedliche Einsatzempfehlungen haben.

Die folgenden Audio-Codecs gehören zum Standard H.323, der ebenfalls von der ITU festgelegt wurde.

4.2.1 Der G.711-Standard

Der G.711 Standard unterstützt Verbindungen von 56 und 64 KBit/s. Er hat eine Audiobandbreite von 3.1 kHz (nämlich 0,3 – 3,4 kHz), die mittels PCM komprimiert wird. [siehe auch Abschnitt *4.1.1: Waveform Codec* und *4.3: ISDN*].

Dieser Standard ist in der Telephonieumgebung weit verbreitet und ermöglicht Interoperabilität mit Microsoft's NetMeeting PC Conferencing Package, und somit unternehmensweite Kommunikation über große Entfernungen (weltweit).

4.2.2 Der G.722-Standard

Der G.722 Standard ist zur Übertragung von 7 kHz auf Verbindungen mit 48, 56 und 64 KBit/s geeignet. Durch die Anwendung der ADPCM (Adaptive Differential PCM) kann für dieses 7-kHz-Audiosignal eine Steigerung der Sprachqualität erzielt werden (Qualitätstelephonie). Folglich liefert er eine höhere Sprachqualität als der G.711 Standard.

Auch für Videokonferenzsysteme, die eine Übertragungsrate von mehr als 384 KBit/s haben sollen, wird häufig auf Lösungen zurückgegriffen, die auf Basis der G.722 Spezifikation arbeiten.

4.2.3 Der G.723.1-Standard

Dieser Audio Codec liefert bei einer Übertragung 3.1 kHz auf Verbindungen mit 5,3 und 6,3 KBit/s eine hinreichend gute Sprachqualität (Multimedia-Kommunikation). Als Komprimierungsverfahren wird ACELP (5,3 KBit/s) oder MP-MLQ (6,3 KBit/s) angewendet.

Da die erforderliche Übertragungsrate hier so niedrig ist, wird diese Spezifikation als Basiskodierung für VoIP- Anwendungen des VoIP- Forums verwendet.

4.2.4 Der G.728-Standard

Der G.728-Standard ist für Videokonferenzsysteme weit verbreitet. Unter Verwendung des rechenintensiven Kompressionsverfahren LD-CELP (Low Delay CELP) wird die Sprachqualität des G.711-Standards erreicht, obwohl nur eine Datenrate von 16 KBit/s erzeugt wird.

4.2.5 Der G.729 und G.729A-Standard

Dieser 1996 verabschiedete Standard zeichnet sich gegenüber dem G.723.1-Standard durch niedrigere Verzögerungen aus. Die Sprachdaten werden hier mittels CS-CELP bis auf 8 KBit/s komprimiert. Deshalb eignet er sich für Videotelephonie über analoge Verbindungen und neuere VoIP- Anwendungen.

4.3 ISDN Standard

Die ISDN-Telefone haben eine Abtastrate von 8 kHz und quantisieren mit 12 Bit. Danach erfolgt eine Komprimierung nach G.711, d.h. es wird die PCM-Sprachkompression verwendet. Da in Europa und Amerika für Sprachübertragung über ISDN-Verbindungen unterschiedliche Bandbreiten zur Verfügung stehen, werden auch unterschiedliche PCM-Verfahren angewendet: In Europa wird die *A-law*-Methode (**PCMA**) benutzt [Abb. 4.2], die bei einer Kompression auf 8 Bits pro Sample eine ausreichend gute Sprachqualität liefert, und so den zur Verfügung stehende Kanal von 64 KBit/s voll ausnutzt (8 Bit · 8 kHz = 64 KBit/s). In Amerika hingegen gibt es für die ISDN-Übertragung nur Kanäle mit einer Bandbreite von 56 KBit/s. Deshalb ist dort *μ-law* (**PCMU**) mit einer Kompression auf 7 Bits Standard (7 Bit · 8 kHz = 56 KBit/s).

Die Frequenzbandbreite beträgt die in Abschnitt *4.2.1: G.711* bereits angesprochenen 3,1 kHz, es gibt jedoch auch einen 7 kHz-Telefondienst im ISDN.

4.4 GSM Standards

Die Übertragung von Sprache über das GSM Mobiltelefonnetzwerk ist komplex, nicht nur wegen der Infrastruktur des Netzwerkes und dem Management, das notwendig ist um eine Verbindung herzustellen, sondern auch wegen dem Codec-Schema, welches zur Encodierung der zu übertragenden Stimme benutzt wird. Im GSM-Netz gibt es zwei verschiedene Stellen, die Sprachkompression anwenden müssen: Einerseits das Handy, das die Sprache vom Mikrophon umwandelt, und anderseits die TRAU (Transcoder and Rate Adaptation Unit), die vom MSC (Mobile Switching Center) umkomprimierte Sprache aus anderen Netzen erhält.

Der GSM-Standard unterstützt vier verschiedene aber ähnliche Kompressionstechnologien, um Sprache zu analysieren und zu kodieren. Diese sind: Full-Rate, Enhanced Full-Rate, Half-Rate und Adaptive Multe-Rate. Obwohl sie Verlust behaftet sind (bei der Kompression können Daten verloren gehen), wurden diese Codecs optimiert Sprache am Ende einer drahtlosen Verbindung akkurat wiederzugeben und so eine gute Sprachqualität zu liefern.

4.4.1 Die GSM-Sprachübertragung

Wenn man in das Mikrophon eines GSM-Telefons spricht, wird das Gesprochene in ein digitales Signal mit 13 Bit Auflösung und 8 kHz Samplerate konvertiert und anschließend komprimiert. Die Outputrate des GSM-Codecs reicht von 4.75 KBit/s bis 13 KBit/s, abhängig von seinem Typ. Tabelle 4.1 zeigt die verschiedenen Bitraten, Kompressionsgrade, die benutzten Komprimierungsmethoden und die Zeit, die zum Enkodieren und Dekodieren eines zufälligen sprachähnlichen Datenstroms gebraucht wurde.

Tabelle 4.1: Die Kodierungsraten und die allgemeine Enkodierungs- und Dekodierungskomplexität. [4]

Codec	Bitrate (KBit/s)	Kompres-sion	Codec Typ	Enkodieren (ms)	Dekodieren (ms)
FR	13	8	RTE-LTP	0,41	0,16
EFR	12,2	8,5	ACELP	9,02	0,86
HR	5,6	18,4	VSELP	8,31	1,30
AMR 12.2	12,2	8	ACELP	8,99	1,11
AMR 10.2	10,2	10,2	ACELP	8,31	1,12
AMR 7.95	7,95	13,1	ACELP	8,70	1,07
AMR 7.4	7,4	14,1	ACELP	8,10	1,07
AMR 6.7	6,7	15,5	ACELP	8,53	1,08
AMR 5.9	5,9	17,6	ACELP	7,19	1,44
AMR 5.15	5,15	20,2	ACELP	6,40	1,38
AMR 4.75	4,75	21,9	ACELP	7,71	1,08

4.4.2 Der Full-Rate (FR) Codec

Der FR-Codec benutzt das LPC-LTP-RPE Sprachkodierungsverfahren [s. Abschn. 4.1.3] und verarbeitet Sprachblöcke von 20 ms, die jeweils 260 Bits enthalten (260 Bit / 20 ms = 13000 Bit/s = 13 KBit/s). Die genaue Aufteilung der Bits zeigt Tabelle 4.2.

Die drei wichtigsten Teile des Encoder sind folglich: LPC (*Linear Prediction*, Short-Term Analysis), LTP (*Long-Term Prediction*) und RPE (*Excitation Analysis*) [Abb. 4.5].

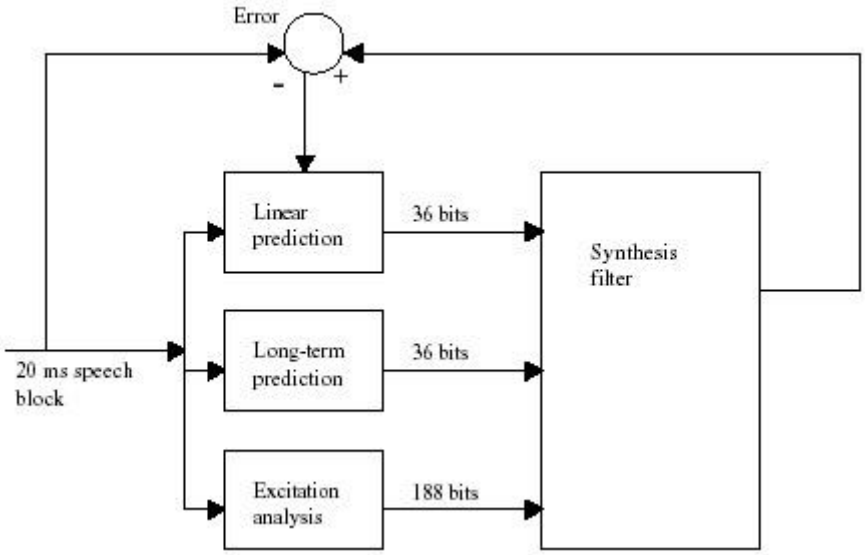


Abbildung 4.5: Diagramm des GSM Full-Rate LPC-LTP-RPE Codecs. [3]

Die LPC ist für die Analyse der kurzfristige Korrelation des Inputsignals (20 ms-Block) zuständig und bestimmt daraus die ‘Reflexions-’ Koeffizienten des Filters. Diese Reflexions-Koeffizienten werden in *log area ratios* (LARs) transformiert und dann als acht Parameter mit insgesamt 36 Bit Information über die Luft übertragen. Danach werden sie benutzt, um die kurzfristige Filterung des Inputsignals durchzuführen, wodurch ein Signal von nur noch 160 Samples entsteht. Diese werden dann in vier Sub-Frames von jeweils 40 Samples geteilt.

Die LTP berechnet anschließend die Parameter der langfristigen Korrelation, die dann vom langfristigen Filter angewendet werden, um das Ergebnis der kurzfristigen Filterung zu schätzen. Der Fehler zwischen der Schätzung und dem tatsächlichen Ergebnis wird dann an die RPE- Analyse geschickt, welche die Datenkomprimierung vornimmt.

Die RPE reduziert die 40 Samples, die nach der langfristigen Filterung übrig geblieben sind, zu vier Mengen von 13 Bit Sub-Sequenzen. Davon wird die optimale Sub-Sequenz mit dem geringsten Fehler bestimmt, und per ADPCM in 45 Bits kodiert.

Das daraus entstehende Signal wird zurück durch den RPE-Decoder geschickt und mit der Schätzung des kurzfristig gefilterten Signals vermischt, um es als Quelle der langfristigen Analyse für den nächsten Frame zu verwenden. Dies beendet die Feedback-Schleife [Tab.4.2].

Tabelle 4.2: Die Aufteilung der Bits beim GSM Full-Rate Codec. [3]

		Bits per 5 ms Block	**Bits per 20 ms Block**
LPC filter	8 parameters		36
LTP filter	Delay parameter	7	28
	Gain parameter	2	8
Excitation signal	Subsampling phase	2	8
	Maximum amplitude	6	24
	13 samples	38	156
Total			260

4.4.3 Der Enhanced Full-Rate (EFR) Codec

Mit der Verbesserung der Prozessoren konnten komplexere Codecs für bessere Sprachqualität entwickelt werden. Zu diesen gehört auch der EFR-Codec, der im Vergleich zum FR-Codec eine detailliertere Sprache übermitteln kann, obwohl seine Bitrate niedriger ist.

Der EFR-Codec benutzt ein ähnliches LPC-LTP-RPE Kodierungsverfahren wie der FR-Codec. Doch um die Verbesserungen zu erreichen, wird zur kurzfristigen Analyse ein LPC-Filter 10. Ordnung benutzt und pro Frame zweimal angewendet. Und zur langfristigen Vorhersage wird ein *Algebraic CELP* (**ACELP**) Codec verwendet, der mit dem CELP-Codec [s. Abschn. 4.1.3.4] verwandt ist, aber zusätzlich ein adaptives Codebuch, dessen Parameter sich während des Kodierens auf das Sprachsignal einstellen, besitzt.

Tabelle 4.1 verdeutlicht, wie sich die bessere Sprachqualität bei niedrigerer Bitrate auf die Kodierungszeiten – und damit auch die Berechnungskomplexität – niederschlägt: der Enhanced Full-Rate Codec braucht ungefähr die 22fache Enkodierungszeit und die 5fache Dekodierungszeit des FR-Codecs.

4.4.4 Der Half-Rate (HR) Codec

Es gibt auch eine Half-Rate Version des GSM-Codecs, die in den mittleren neunziger Jahren eingeführt wurde. Implementiert wurde dieser HR-Codec als lineares Sprach-kodierungsverfahren der *Vector Sum Excitation Linear Prediction* (**VSELP**) mit einer Bitrate von 5.6 KBit/s. VSELP ist verwandt mit dem CELP- Verfahren aus Abschnitt 4.1.3.4, benutzt aber mehr als ein Codebuch.

Die geringe Bitrate ist der große Vorteil des HR-Codecs, denn die Luftübertragungs-spezifikation für GSM erlaubt das Aufspalten eines Sprachkanals in zwei Unterkanäle, die unterschiedliche Anrufe aufrechterhalten können. Auf diese Weise kann mit einem Sprach-Codec, der nur die Hälfte der Kanalkapazität verwendet, die Kapazität einer Zelle verdoppelt werden.

Doch die allgemeine Vorstellung der Sprachqualität des HR-Codec war so schlecht, dass er heute im Allgemeinen nicht verwendet wird.

4.4.5 Der Adaptive Multi-Rate (AMR) Codec

Das Prinzip des AMR-Codec ist es, einen Satz von Codecs mit sehr ähnlicher Berechnung zu verwenden, um Output von unterschiedlichen Raten zu erzeugen. In GSM wird die Qualität des empfangenen Luft-Schnittstellensignals überwacht und die Kodierungsrate der Sprache kann geändert werden. So wird für niedrigere Signalbereiche mehr Schutz verwendet, indem man die Kodierungsrate verringert und die Redundanz erhöht, und in den Bereichen guter Signalqualität wird die Qualität der Sprache verbessert.

Für alle AMR-Typen wird das ACELP- Kodierungsverfahren verwendet, und so ist der AMR-Codec mit 12,2 KBit/s rechnerisch derselbe wie der EFR-Codec. Für niedrigere Raten wird die kurzfristige Analyse nur einmal pro Rahmen durchgeführt. Als Resultat der verringerten Berechnungen bei den niedrigeren Output-Bitraten entstehen weniger zu übertragende Parameter, und es werden weniger Bits benutzt, um sie darzustellen.

Indem man den Output auf die sechs niedrigsten Kodierungsraten (4,75 bis 7.95 KBit/s) begrenzt, kann der Benutzer den Qualitätsnutzen der anpassungsfähigen Sprachkodierung erfahren, während der Netzwerk-Operator (ähnlich wie bei HR-Codecs) von der erhöhten Kapazität profitiert.

Tabelle 4.1 stellt die verschiedenen AMR-Typen, ihre Bitraten, Kompressionsgrade und die benötigten Enkodierungs- und Dekodierungszeiten dar.

4.5 UMTS Standard

Wie bei GSM gibt es auch im UMTS-Netz die zwei Stellen – Handy und TRAU –, an denen Sprachkodierung eingesetzt wird.

Das Handy tastet die Sprache mit 8 kHz ab und quantisiert die daraus entstehenden Werte mit 13 Bit. Dann fügt es noch 3 Füllbits hinzu, damit jeder Abtastwert aus 16 Bit besteht und vom Codec reduziert werden kann.

Die TRAU erhält Sprachdaten mit einer Datenrate von 64 KBit/s, was 8000 Abtastwerten pro Sekunde mit jeweils 8 Bit Auflösung entspricht. Aber obwohl hier die Auflösung nur halb so groß ist wie beim Handy, erzeugt der verwendete Codec bei beiden die gleichen Datenraten.

UMTS benutzt einen AMR-Codec, der ähnlich wie bei GSM verschiedene Betriebszustände zur Verfügung stellt und sich den Übertragungsbedingungen anpassen kann. Er beruht ebenfalls auf den ACELP- Kodierungsverfahren und stellt acht verschiedene Übertragungsmodi mit unterschiedlichen Bitraten zur Verfügung, sowie weitere Modi für Sprachpausen [s. Tab.4.3]. D.h., dass auch hier Kurz- und Langzeitanalysen bezüglich der Sprachredundanz vorgenommen und ein fixes und ein adaptives Codebuch benutzt werden. Und genau wie bei GSM zerlegt der AMR-Kodierer die Sprachdaten in 20 ms lange Pakete á 160 Abtastwerte. Für UMTS bedeutet das, dass jeweils zwei Funkzeitrahmen (á 10ms) notwendig sind, um einen Sprachrahmen zu übertragen.

Ist ein Sprachrahmen fertig übertragen besteht prinzipiell die Möglichkeit die Sprachdatenrate zu verändern, wodurch die Verkehrslast der Funkschnittstelle berücksichtigt werden kann. Aus diesem Grund spricht man auch von *Multi Rate ACELP* (**MR-ACELP**).

Tabelle 4.3: Die UMTS Übertragungsmodi des AMR-Codecs [8].

Rahmen-Typ	**Codec-Typ**	**Bits pro 20 ms Sprachrahmen**	**Rahmengröße in Bit**
0	AMR 4,75	95	114
1	AMR 5,15	103	122
2	AMR 5,90	118	137
3	AMR 6,70	134	153
4	AMR 7,40	148	167
5	AMR 7,95	159	178
6	AMR 10,2	204	223
7	AMR 12,2	144	263
8	AMR SID		58
9	GSM-EFR SID		62
10	TDMA-EFR SID		57
11	PDC-EFR SID		56
12	reserviert		-
13	reserviert		-
14	reserviert		-
15	Keine Daten		-

Die 8 verschiedenen AMR-Sprachmodi beinhalten aus Kompatibilitätsgründen auch drei bereits verwendete Sprachmodi: Der Rahmentypus ***7****: AMR 12,2 KBit/s* entspricht EFR-Sprachübertragung bei GSM. Der Rahmentypus ***4****: AMR 7,4 KBit/s* entspricht der Sprachkompression des amerikanischen IS-641 Mobilfunkstandards. Und der Rahmentypus ***3****: AMR 6,4 KBit/s* entspricht der EFR-Sprachübertragung des japanischen PDC-Mobilfunkstandards.

5. Fazit

Digitalisierung von Audio spielt nicht nur bei Tonaufnahmen, -speicherungen und -weiterverarbeitungen auf digitalen Datenträgern eine große Rolle, sondern auch im Bereich der Telephonie. Die Umwandlung der analogen, kontinuierlichen Signale in binären Zahlencode – sprich: die Abtastung, Quantisierung und Kodierung – wird vom A/D-Wandler durchgeführt. Anschließend komprimieren Codecs die dadurch entstehenden Daten, indem sie Redundanzen und – für das menschliche Ohr – nicht wahrnehmbare Frequenzen entfernen.

Da die verschiedenen Telefon-/ Mobilfunknetze über unterschiedliche Bandbreiten und Übertragungsmethoden verfügen, und es unterschiedliche Anforderungen an die Sprachqualität gibt, wurden über die Zeit viele verschiedene Sprachkodierungsmethoden und Codecs entwickelt und Standards festgelegt. Einen Vergleich zwischen einigen häufig verwendeten Codecs zeigt Tabelle 5.1, die auch den Zusammenhang zwischen Datenrate und Soundqualität hervorhebt.

Tabelle 5.1: Verschiedene Codecs und ihre Bewertungen: 1 = bad, 2 = poor, 3 = clear, 4 = good, 5 = excellent. [10; 11]

Codec	Datenrate (KBit/s)	Datenrate/Minute (KB)	Soundqualität
16-Bit PCM	128	960	5
G.711	64	480	4 / 5
G.726	32	240	4
ADPCM	32	240	3 / 4
GSM 6.10	13.2	98	2 / 3
G.729A	8	60	2 / 3

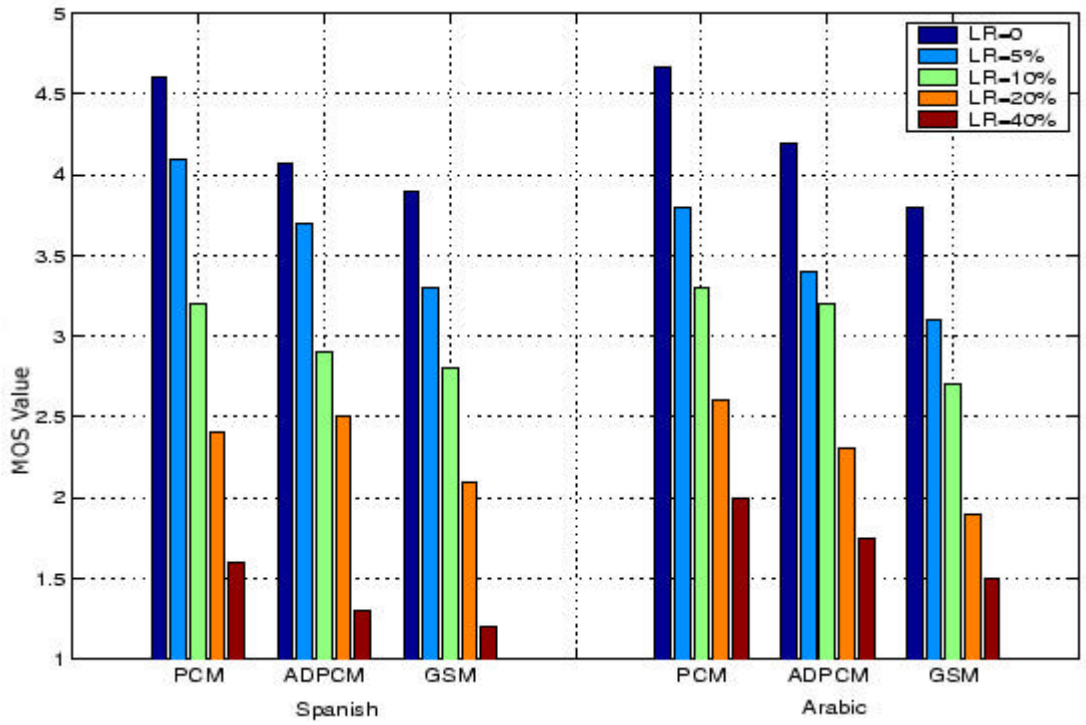


Abbildung 5.1: Die Qualitätsunterschiede – Mean Opinion Score (MOS) – als Funktion über 'Loss Rate' (LR) und angewendetem Codec. LR gibt den Prozentsatz an verlorenen Paketen an. [12]

Zur Verbesserung der Sprachqualität und besseren Ausnutzung der Bandbreite werden Codecs weiterhin weiterentwickelt. Beispielsweise hat die ITU Anfang dieses Jahres eine neue Empfehlung für die Übertragung von Sprachsignalen verabschiedet: **G.722.2**. Diese wird es ermöglichen, die Sprachqualität sowohl im ISDN- als auch im Mobilfunknetz deutlich zu verbessern. Das neue Verfahren verwendet 16000 16-Bit-Messwerte. Dies ist ein großer Fortschritt im Vergleich zum bisherigen Standard, denn der Frequenzbereich wird nach oben um etwas mehr als eine Oktave (von 3400 auf 7000 Hz) und nach unten um zwei Oktaven

(von 200 auf 50 Hz) erweitert. Die Erweiterung nach unten macht die Sprache am Telefon natürlicher und den Gesprächspartner präsenter. Die Erweiterung nach oben trägt hingegen zum Sprachverstehen bei. Die höhere Bitrate und der erweiterte Frequenzumfang bewirken auch, dass das Hintergrundgeschehen (Musik, Nebengeräusche) genauer übertragen wird als bisher.

Um die größeren Datenmengen übertragen zu können, müssten vier herkömmliche ISDN-Kanäle parallel geschaltet werden. Um kompatibel zu den bisherigen 64-KBit-Kanälen zu sein, wird die Sprache mit einem *Adaptive Multi-Rate Wideband* (**AMR-WB**) Codec komprimiert. Die stärkste Kompression reduziert die ursprünglichen 256KBit/s zu 6,6KBit/s und erzeugt dadurch einen der Klang nicht viel besser ist als eine gewöhnliche Telefonverbindung. Bei der geringsten Kompression hingegen werden 23,85KBit/s übertragen, wodurch die oben erwähnten Verbesserungen bei der Bandbreite voll zur Geltung kommen. Doch um die bessere Sprachqualität nutzen zu können, werden neue Endgeräte, die AMR-WB unterstützen, notwendig. So führt die Weiterentwicklung der Codecs zwangsweise auch zu Veränderungen der Hardware.

6. Quellenangabe

[1] Prof. Dr. ANDRE, Elisabeth: *Signale und Kodierung*. Universität Augsburg. Online im Internet: URL [besucht 11.12.2005]: http://mm-werkstatt.informatik.uni-augsburg.de/files/teaching_content/944_Kompression6up.pdf

[2] ZAIK/RRZK Multimedia Gruppe. *Digitale Audiobearbeitung*. Universität Köln. Online im Internet: URL [besucht 19.12.2005]: http://www.uni-koeln.de/rrzk/multimedia/dokumentation/audio/

[3] LEHTONEN, Kristo: *Digital Signal Processing and Filtering.* Universität Saarland. Online im Internet: URL [besucht 11.12.2005]: http://www.rz.uni-saarland.de/projekte/VoIP/sip/glossar.htm

[4] MESTON, Richard: *Sorting Through GSM Codecs: A Tutorial*. CommsDesign.com. Online im Internet: URL [besucht 20.12.2005]: http://www.commsdesign.com/design_corner/OEG20030711S0010

[5] KÖHLER, Rolf-Dieter (2002): *Voice over IP*. Bonn: mitp-Verlag

[6] Wikipedia. Online im Internet: URL [besucht 30.01.2006]: http://de.wikipedia.org/wiki/Redundanz

[7] Teltarif.de. Online im Internet: URL [besucht 06.02.06]: http://www.teltarif.de/arch/2002/kw15/s7661.html

[8] UMTSlink. Online im Internet: URL [besucht 06.02.06]: http://umtslink.at/cgi-bin/reframer.cgi?../UMTS/amr.htm

[9] Moviecollege. Online im Internet: URL [besucht 06.02.06]: http://www.movie-college.de/filmschule/ton/digital.htm

[10] Sericyb.com. Online im Internet: URL [besucht 10.02.06]: http://sericyb.com.au/sc/audio.html

[11] Cisco.com. Online im Internet: URL [besucht 10.02.06]: http://www.cisco.com/univercd/cc/td/doc/product/voice/c_unity/whitpapr/codecs.htm

[12] Irisa.fr. Online im Internet: URL [besucht 20.03.06]: http://www.irisa.fr/armor/lesmembres/Mohamed/Thesis/node190.html