

**H O C H S C H U L E
R A P P E R S W I L
H S R**

**1. Studienarbeit
Digitale Signalverarbeitung (DS)
WS-99-02**

Sprachverständlichkeitsindex



Speech Intelligibility Index (SII)

Dokumentation

Verfasser : Stefan Huber
Markus Hurschler

Betreuer : A. Schaub

Inhaltsverzeichnis

Inhaltsverzeichnis.....	1
1 Abstract.....	3
2 Aufgabenstellung.....	4
2.1 Zeitplan.....	5
3 Einleitung.....	6
3.1 Verständlichkeit von Sprache.....	6
3.2 Sprachverständlichkeitsindex SII.....	6
3.3 Methoden zur Berechnung des SII.....	6
3.3.1 Methode mit kritischen Teilbändern.....	6
3.3.2 Methode mit gleichmässiger Berücksichtigung der kritischen Teilbänder.....	7
3.3.3 Oktavbänder-Methode:.....	7
3.3.4 Drittel-Oktavbänder-Methode.....	7
4 Grundkenntnisse der digitalen Signalverarbeitung.....	8
4.1 Abtasten, Quantisieren, Quantisierungsrauschen.....	8
4.2 FIR-Filter.....	9
4.3 Änderung der Abtastrate.....	13
4.3.1 Halbierung der Abtastrate.....	13
4.3.2 Verdoppelung der Abtastrate.....	13
5 Digitale Verarbeitung der Eingangssignale für SII.....	15
5.1 Hardware.....	15
5.1.1 Abtastrate, Quantisierung.....	15
5.1.2 Eichung der Aufnahmelautstärke.....	15
5.2 Software.....	16
5.2.1 Filterbank mit FIR-Filtern.....	16
5.2.2 Eichung der Teilbandsignalpegel auf die normierten Pegel des SII.....	18
5.2.3 Erzeugung eines Störsignals für Testzwecke.....	19
5.2.4 Rekonstruktion eines Signals anhand seiner Teilbänder.....	20
5.2.5 Rückwandlung der digitalen Signale in analoge.....	30
6 Untersuchungen mit der Sprachverständlichkeit.....	31
6.1 Analyse des SII-Algorithmus.....	31
6.2 Test zur Erfassung des subjektiven Verständlichkeitsempfindens.....	34
6.3 Vergleich SII mit subjektivem Verständlichkeitsempfinden.....	37
7 Schlussgedanken zu dieser Studienarbeit.....	38
8 Abkürzungen und Symbole.....	39

9

Beilagen.....	40
9.1 SII	40
9.1.1 Algorithmus zur Berechnung des SII.....	40
9.1.2 Tabelle: Drittel-Oktavbänder-Methode.....	42
9.2 Matlab Funktionen.....	42
9.2.1 SII-Algorithmus.....	42
9.2.2 Koeffizienten für FIR-Tiefpassfilter.....	44
9.2.3 Verdoppelung der Abtastrate.....	44
9.2.4 SII-Filterbank.....	45
9.2.5 Anpassung der Signallänge an eine durch 32 teilbare Zahl.....	46
9.2.6 Eichung der Teilbandsignalpegel auf die normierten Pegel des SII.....	47
9.2.7 Erzeugung eines Störsignals für Testzwecke.....	47
9.2.8 Rekonstruktionsbank.....	49
9.3 Matlab M-Scripts.....	50
9.3.1 Test zur Erfassung des subjektiven Verständlichkeitsempfindens.....	50
9.4 Formulare.....	52
9.4.1 Test zur Erfassung des subjektiven Verständlichkeitsempfindens.....	52

10

Literatur.....	53
-----------------------	-----------

1 Abstract

Bei der Entwicklung und Verbesserung von Sprachübertragungssystemen benötigt man eine Messgrösse, welche etwas über die Qualität des vorhandenen Systems aussagt. Eine solche Messgrösse ist der Sprachverständlichkeitsindex (Speech Intelligibility Index, SII). Er wird ausgedrückt durch eine Zahl, welche sich im Bereich zwischen null und eins bewegt. Je näher dieser Wert bei eins liegt, desto mehr Informationen gelangen zur Auswertung ins menschliche Gehirn. Der Index berücksichtigt zudem nicht nur ideale akustische Verhältnisse, sondern ist vor allem dazu gedacht, Sprachübertragungssysteme unter realen Bedingungen zu testen.

Wesentliche Beiträge für die Entwicklung des SII wurden vor allem in den 50er Jahren geleistet, aber seine Ursprünge reichen bis an den Anfang des 20. Jahrhunderts zurück. Die Berechnungsmethoden erfuhren aber auch danach eine fortwährende Weiterentwicklung; so wurde erst 1997 eine neue, verbesserte Version eines ANSI Standards herausgegeben. Wegen der grundsätzlichen Schwierigkeit, die relevanten Phänomene im Zusammenhang mit dem Verstehen von Sprache treffend zu modellieren, sind auch in Zukunft noch weitere Verbesserungen zu erwarten.

Das menschliche Gehirn analysiert auftreffende Sprachsignale auch im Frequenzbereich. Diese Gegebenheit wird durch den SII berücksichtigt, denn die wesentlichen Einflussgrössen für dessen Berechnung sind die Spektren des Sprachsignals und des Störgeräusches. Die Einflussgrössen müssen vorerst in Teilbänder, z.B. in Oktavbänder, zerlegt werden, bevor daraus der SII berechnet werden kann. Für die Berechnung selbst spielen dann noch weitere audiologische Gegebenheiten wie Maskierungsphänomene und die menschliche Hörschwelle eine wichtige Rolle.

In unserer Arbeit nehmen wir den Sprachverständlichkeitsindex gemäss ANSI Standard etwas genauer unter die Lupe.

2 Aufgabenstellung

Die Hauptaufgabe bei dieser Studienarbeit besteht darin, Methoden zur Berechnung des SII nach dem neuen verbesserten ANSI-Standard aus dem Jahr 1997 [1] kennenzulernen und anschliessend die Methode, welche auf Drittel-Oktavbändern beruht, zu implementieren. In einem zweiten Schritt ist anhand von verschiedenen Sprachsignalen zu vergleichen, wie weit der jeweilige Indexwert mit dem subjektiven Verständlichkeitsempfinden von Personen mit normalem Gehör übereinstimmt.

Damit wir die einzelnen Teilaufgaben erfolgreich ausführen können, müssen wir uns zudem weitere Grundkenntnisse der digitalen Signalverarbeitung aneignen:

- Filterung von Signalen mit FIR-Filtern,
- Reduktion der Abtastrate und
- Erhöhung der Abtastrate.

Diese Grundkenntnisse erlauben es uns dann, Programme zu implementieren, welche folgende Funktionalitäten aufweisen:

- Aufteilen und Zusammenfügen von Signalen in bzw. aus Drittel-Oktavbandsignalen mittels FIR-Filtern mit perfekter Rekonstruktion und aliasfreiem Down- und Up-Sampling,
- Aufteilen von Sprachsignalen und weissem Rauschen in Teilbandsignale,
- Messen und Einstellen der Teilbandpegel und anschliessendes Zusammenfügen zu Gesamtsignalen sowie
- Addieren von aufbereiteten Sprach- und Rauschsignalen mit anschliessender Bestimmung der resultierenden Indexwerte.

Ein weiterer Punkt ist die ausführliche Dokumentation der gesamten Studienarbeit. Dabei soll das Schwergewicht vor allem auf die selbst geleisteten Arbeiten gelegt werden.

Literatur

- [1] ANSI S3.5-1997: Methods for Calculation of the Speech Intelligibility Index, American National Standard Institute, 1997
- [2] A. Schaub: Skript zur Einführung in die digitale Signalverarbeitung für Informatiker, HSR, 1998

Bericht

Über die Arbeit ist ein Bericht zu verfassen. Dabei sind die Richtlinien der Abteilung Elektrotechnik für die Erstellung von Berichten einzuhalten. Alle verwendeten Quellen sind im Literaturverzeichnis des Berichtes anzugeben; ein Exemplar bleibt im Labor für digitale Signalverarbeitung der HSR, das Doppel erhalten die Verfasser nach der Korrektur und der Bewertung zurück. Die erstellten Programme und der Text des Berichtes sind auf Diskette(n) (mit Nummer der Studienarbeit) beizulegen.

Termine, Bedingungen

Gemäss Vorgaben und Terminplan des Vorstandes der Abteilung für Elektrotechnik.

- Ausgabe der Aufgabenstellung : 2.11.1998
- Vorbereitungsteil : 2.11. - 18.12.1998
- Abgabe Zwischenbericht : 18.12.1998
- Praktischer Teil : 4.1. - 26.2.1999
- Arbeitsplatz : DS-Labor 1219 (zuständig: P. Roffler, Laborassistent)
- Abgabe des Berichts, Ende der Arbeit : 26.2.1999

Assistenz

Th. Hugentobler

Kontaktadresse

Bernafon AG
 A. Schaub
 Eichtalstr. 55
 8634 Hombrechtikon
 Tel. 055 - 264 13 51
 Fax. 055 - 264 13 54

2.1 Zeitplan

Die Studienarbeit gliedert sich in zwei Phasen. In einer ersten Phase geht es vor allem darum, sich in die wesentlichen theoretischen Aspekte der Arbeit einzuarbeiten.

Die zweite Phase beginnt mit dem Bezug der Laborplätze, wo dann die theoretischen Erkenntnisse angewendet werden.

Wochennummer	46	47	48	49	50	51	52	53	1	2	3	4	5	6	7	8
Erstellen des Programms für SII-Berechnung																

1. Testphase																	
Einarbeiten in die Theorie der FIR-Filter und Erstellen von Filterbänken																	
Erstellen von Programmen zum Aufteilen und Zusammenfügen von Signalen																	
Dokumentation																	
Anwendung der Programme auf Sprachsignale																	
Testreihe und Auswertung																	

18.12.98◆

26.2.99◆

Aufteilung der Arbeiten

Die Einarbeitung in die theoretischen Grundlagen während der ersten Phase werden von beiden Studenten parallel durchgeführt. In der zweiten, praktischen Phase sowie bei der Dokumentation werden die anfallenden Arbeiten soweit wie möglich aufgeteilt.

3 Einleitung

Für das Verständnis der gesamten Arbeit wollen wir uns in diesem Abschnitt erst einmal damit befassen, wann wir Sprache als verständlich empfinden. Im weiteren werden die Methoden zur Bestimmung des SII gemäss ANSI Standard kurz erläutert.

3.1 Verständlichkeit von Sprache

Das menschliche Gehirn kann Sprachsignale dann optimal auswerten, wenn das Sprachspektrum im Mittel 15 dB höher ist als alle andern ebenfalls vorhandenen Signalspektren wie z.B. Störgeräusch oder Hörschwelle. Dies hat zur Folge, dass wir in einer Umgebung mit einem hohen Geräuschpegel lauter sprechen müssen, um die gleiche Verständlichkeit zu erreichen, als in einer Umgebung mit einem geringen Geräuschpegel. Das Überdecken eines Sprachsignals durch ein Störgeräusch bezeichnet man auch als Maskierung. Ist der Sprachstörersignalabstand dabei im Mittel geringer als 15 dB, dann bezeichnet man das dabei auftretende Phänomen der Verständlichkeitsverschlechterung als Maskierungsphänomen. Ist nun aber das Sprachspektrum im Mittel höher als 15 dB über den andern vorkommenden Spektren, so ist bereits die volle Verständlichkeit erreicht. Wenn das Sprachspektrum im Mittel deutlich mehr als 15 dB darüber ist, so verringert sich die Verständlichkeit durch zunehmende Verzerrungen.

3.2 Sprachverständlichkeitsindex SII

Der SII wird mit einem Wert zwischen null und eins ausgedrückt. Dabei bedeutet null, das Sprachsignal ist nicht verständlich, während eins für volle Verständlichkeit steht. Aber auch bei einem Wert von 0.5 ist die Sprache noch durchaus verständlich.

Die zentralen Grössen zur Berechnung des SII sind die Spektren des Sprachsignals und des Störgeräusches. Im ANSI Standard gibt es mehrere Berechnungsmethoden, wobei der Algorithmus zur Berechnung des SII für alle Methoden dasselbe Konzept verfolgt. Die einzelnen Methoden unterscheiden sich nur in den Teilschritten.

Für die Auswertung von berechneten SII-Werten, ist zu berücksichtigen, dass diese sich im Mittel immer auf mehrere Sprecher und Hörer beider Geschlechter beziehen. Bei den Hörern geht man von einem normal funktionierenden Gehör aus. Nicht berücksichtigt werden Hörer, welche die Sprache nicht oder nur teilweise verstehen oder Hörer, die irgendwelche anderen Defizite im Verstehen von Sprache aufweisen.

3.3 Methoden zur Berechnung des SII

Der ANSI Standard aus dem Jahre 1997 [1] beinhaltet vier Methoden zur Bestimmung des SII. Im folgenden werden diese Methoden kurz erläutert. Die Drittel-Oktavbänder-Methode ist dabei für diesen Bericht die wichtigste, weil wir gemäss Aufgabenstellung nur Programme für diese Methode zu realisieren hatten.

3.3.1 Methode mit kritischen Teilbändern

(Critical band SII procedure)

Diese Methode berücksichtigt 21 Teilbänder im Bereich von 150-8500 Hz. Die Bandwichtigkeit I_i (band importance) beträgt für das 4. - 17. Teilband genau gleichviel nämlich:

$$I_i = 0.0577 \quad i = 4, 5, \dots, 17$$

Gegen unten und oben nimmt die Bandwichtigkeit ab. Die Bandwichtigkeit gibt darüber Auskunft, wie stark ein einzelnes Teilband für die Berechnung des SII gewichtet wird. Die Summe der Bandwichtigkeiten über alle Teilbänder ergibt bei allen Methoden stets eins.

$$\sum_{i=1}^N I_i = 1 \quad (3.1)$$

N: Anzahl Teilbänder

3.3.2 Methode mit gleichmässiger Berücksichtigung der kritischen Teilbänder

(Equally-contributing (17 band) critical-band SII procedure)

Die Eingangssignale werden hier in 17 Teilbänder im Bereich von 350-5800 Hz aufgeteilt. Im Gegensatz zu den andern Methoden hat hier jedes Teilband die gleiche Bandwichtigkeit. Diese beträgt:

$$I_i = 0.0588 \quad i = 1, 2, \dots, 17$$

3.3.3 Oktavbänder-Methode

(Octave band SII procedure)

Bei dieser Methode erfolgt eine Aufteilung in sechs Oktaven, welche nicht mehr weiter in Teilbänder aufgeteilt werden. Die Mittenfrequenz der untersten Oktave beträgt 250 Hz, die der obersten 8000 Hz. Die Bandwichtigkeit ist bei der 3. - 5. Oktave am höchsten.

3.3.4 Drittel-Oktavbänder-Methode

(One-third octave band SII procedure)

Bei der Drittel-Oktavbänder-Methode gibt es achtzehn Teilbänder im Bereich von 160-8000 Hz. Innerhalb dieses Bereichs gibt es sechs Oktaven, welche alle in drei Teilbänder mit einem Abstand von jeweils einer Dritteloktave aufgeteilt werden. Dabei sind die Bänder mit Mittenfrequenzen von 1000-4000 Hz am wichtigsten, während die Bänder am Rande nur eine kleine Rolle spielen. Im wesentlichen ermittelt nun der Algorithmus zur Berechnung des SII aus dem Sprachsignal und aus dem Störgeräusch einen Hörbarkeitsfaktor A_i (band audibility) pro Teilband mit Mittenfrequenz f_i (center frequency of an SII band). Mit diesen Grössen wird der Verständlichkeitsindex wie folgt berechnet:

$$SII = \sum_{i=1}^{18} A_i \cdot I_i \quad (3.2)$$

Eine genauere Analyse des SII-Algorithmus für diese Methode wird in Abschnitt 6.1 durchgeführt. Unter der Beilage 9.2.1 ist die Matlab-Funktion zur Berechnung des SII zu finden.

4 Grundkenntnisse der digitalen Signalverarbeitung

Bevor die Spektren von Sprach- und Störsignalen dem SII-Algorithmus zugeführt werden können, müssen die Sprach- und Störsignale zuerst digital verarbeitet werden. Dieser Abschnitt behandelt deshalb jene Grundkenntnisse der digitalen Signalverarbeitung, die für diese Studienarbeit notwendig sind. Die Theorie dazu stammt aus den im Literaturverzeichnis angegebenen Lehrmitteln [2], [3], [4] und [5].

4.1 Abtasten, Quantisieren, Quantisierungsrauschen

In der Welt der digitalen Signalverarbeitung gibt es nur diskrete Signale, d.h. Signale deren Verlauf nicht kontinuierlich erfasst wird, sondern nur mit regelmässig gemessenen Stützwerten. Dies im Gegensatz zu den in der analogen Welt vorkommenden kontinuierlichen Signalen.

Will man nun ein analoges Signal (z.B. ein Sprachsignal) digital verarbeiten, so muss dieses zuerst in digitale Form umgewandelt werden. Dies geschieht in einem A/D Wandler, welcher das analoge Signal mit einer bestimmten Frequenz, der Abtastfrequenz f_A , abtastet. Dem Abtasttheorem entsprechend muss die Abtastfrequenz mindestens doppelt so gross sein wie die im Signal maximal vorkommende Frequenz (Nyquistfrequenz f_{Ny}).

Jeder Stützwert wird nach der Erfassung gehalten (sample and hold) und anschliessend quantisiert. Unter Quantisieren versteht man das Codieren eines Messwertes mit einer binären Zahl.

Für die Codierung dieses Messwertes stehen n Binärzeichen (Bits) zur Verfügung. Das bedeutet, dass dem Nachrichtensignal N Quantisierungsstufen zugeordnet werden können.

$$N = 2^n \quad (4.1)$$

Weil der Übergang von einer Quantisierungsstufe in die nächst höhere oder tiefere nicht kontinuierlich, sondern in Stufen erfolgt, gibt es nicht für jeden Messwert auch eine genau gleichwertige Quantisierungsstufe. Liegt ein Messwert genau zwischen zwei Stufen, so wird er in der Regel in die tiefere der beiden umgewandelt. Dabei entsteht ein Fehler, der sogenannte Quantisierungsfehler. Die

Leistung des dabei entstehenden Quantierungsrauschens P_Q setzt man ins Verhältnis zur Leistung des nach der Quantisierung zurückgewonnenen Nachrichtensignals P_E und erhält so das Signal zu Rausch-Verhältnis SNR (signal to noise ratio).

$$SNR = 10 \cdot \lg \left(\frac{P_E}{P_Q} \right) \quad (4.2)$$

Gemäss [5] lässt sich der SNR auch direkt aus der Anzahl Quantisierungsbits n bestimmen:

$$SNR = 6 \cdot n \quad (4.3)$$

4.2 FIR-Filter

FIR-Filter sind diskrete Filter mit einer endlichen Impulsantwort (Finite Impulse Response). Das Ausgangssignal eines FIR-Filters wird mit folgender Differenzgleichung berechnet:

$$y[n] = \sum_{k=0}^N b_k \cdot x[n - k] \quad (4.4)$$

Das in Abbildung 1 angeführte Blockschema zeigt auf bildliche Weise, wie ein FIR-Filter zu implementieren ist.

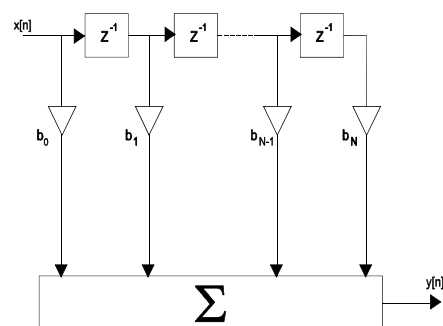


Abbildung 1: Blockschema FIR-Filter

Die im Schema vorkommenden Variablen haben folgende Bedeutung:

b_k : Filterkoeffizienten $x[n]$: Eingangssignal $y[n]$: Ausgangssignal
 z : übliche Schreibweise für $e^{j\omega}$

Bei den Blöcken, welche mit z^{-1} bezeichnet sind, handelt es sich um Verzögerungsglieder. Dabei werden die Werte der Eingangssignalfolge jeweils um ein Abtastintervall verzögert.

Die Filterkoeffizienten b_k geben gerade den Wert der Impulsantwort an der Stelle k an. Für deren Berechnung kann man von der Impulsantwort eines idealen Filters ausgehen. Weil diese aber akausal und zeitlich unbegrenzt ist, sind zwei Schritte notwendig, um ein implementierbares Filter zu erhalten:

1. Die Impulsantwort h wird zeitlich beschränkt; d.h. man setzt $h_i = 0$ für $|i| > M$.
2. Die zeitlich begrenzte Impulsantwort wird um M Abtastintervalle verschoben, damit sie kausal wird.

Auf diese Weise erhält man ein FIR-Filter mit $N = 2M+1$ Filterkoeffizienten. Die zeitliche Beschränkung der Impulsantwort führt dazu, dass ein ideales Filter mit einem FIR-Filter nicht exakt nachgebildet werden kann, denn infolge dieser Begrenzung erhält die resultierende Übertragungsfunktion einen unerwünschten wellenförmigen Verlauf. Dieser kann jedoch durch den Einsatz einer sanften Zeitfensterfunktion, wie z.B. eines Hammingfensters, verringert werden.

Ein Hammingfenster berechnet sich gemäss [4] wie folgt:

$$W_k = \begin{cases} 0.54 - 0.46 \cdot \cos\left(\frac{2\pi k}{N-1}\right), & \text{für } 0 \leq k \leq N-1 \\ 0, & \text{sonst} \end{cases} \quad (4.5)$$

N : Anzahl Filterkoeffizienten

W_k : k -ter Koeffizient für Hammingfenster

Für die Realisierung eines FIR-Tiefpassfilters betrachten wir zunächst folgende Übertragungsfunktion eines idealen Tiefpassfilters mit der Grenzfrequenz f_g :

$$H(f) = \begin{cases} 1 & \text{für } |f| < f_g \\ 0 & \text{sonst} \end{cases} \quad (4.6)$$

Die Impulsantwort dieses idealen Tiefpassfilters sieht damit folgendermassen aus:

$$h(t) = \frac{2\pi f_g}{\pi} \cdot \frac{\sin(2\pi f_g t)}{2\pi f_g t} = 2f_g \cdot \frac{\sin(2\pi f_g t)}{2\pi f_g t} \quad (4.7)$$

Neben den oben genannten Schritten sind bei diskreten Signalen noch zwei weitere formale Punkte zu beachten:

- Die diskreten Werte eines Signals, werden nicht in Funktion der Zeit ausgedrückt, sondern in Funktion der Position des Wertes im Signal.

$$s = s[n] \quad n = 0, 1, 2, 3, \dots \quad (4.8)$$

- Die Frequenzachse wird normiert, so dass die Abtastfrequenz f_A bei 1 liegt. Die im Signalspektrum maximal erlaubte Frequenz f_{Ny} (Nyquistfrequenz) liegt somit bei 0.5.

Die Formel für die einzelnen Werte einer zeitlich begrenzten, diskreten Impulsantwort h_i ohne Berücksichtigung der Verschiebung um M Abtastintervalle sieht somit folgendermassen aus:

$$h_i = \frac{f_g}{f_{Ny}} \cdot \frac{\sin\left(\pi \frac{f_g}{f_{Ny}} i\right)}{\pi \frac{f_g}{f_{Ny}} i} \quad i = -M \dots +M \quad (4.9)$$

Die Filterkoeffizienten b_k erhalten wir nun durch Verschieben von h_i um M Abtastintervalle:

$$b_k = h_i \quad k = 0 \dots N, i = -M \dots +M \quad (4.10)$$

Durch diese Verschiebung, erfährt jedes Eingangssignal eine Verzögerung von M Werten am Ausgang.

Bei der Verwendung eines Hammingfensters wird jeder Filterkoeffizient b_k mit dem entsprechenden Hammingkoeffizienten W_k gewichtet:

$$b'_k = b_k \cdot W_k \quad (4.11)$$

Abbildung 2 zeigt die Impulsantwort eines FIR-Tiefpassfilters mit 129 Koeffizienten und $f_g = f_{Ny}/2$ ohne Hammingfenster.

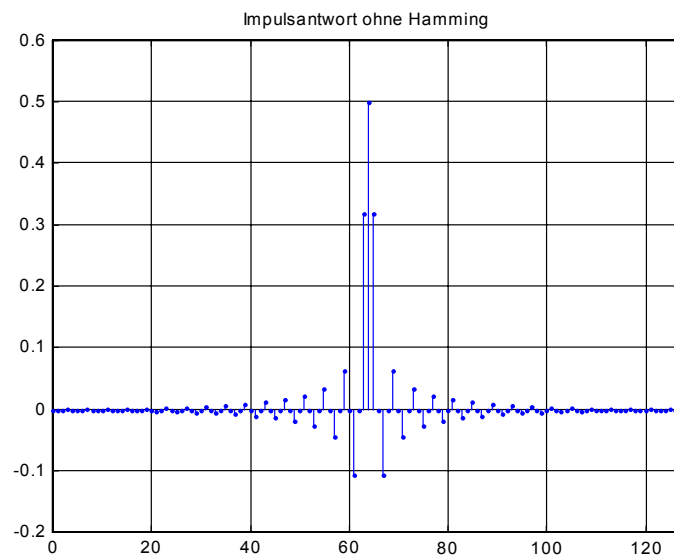


Abbildung 2: Impulsantwort FIR-Filter mit 129 Koeffizienten

In der Beilage befindet sich unter 9.2.2 die Matlab-Funktion zur Berechnung der Filterkoeffizienten eines solchen FIR-Tiefpassfilters.

In Abbildung 3 wird der Amplitudengang dieses FIR-Tiefpassfilters gezeigt; einmal mit und einmal ohne Hammingfenster.

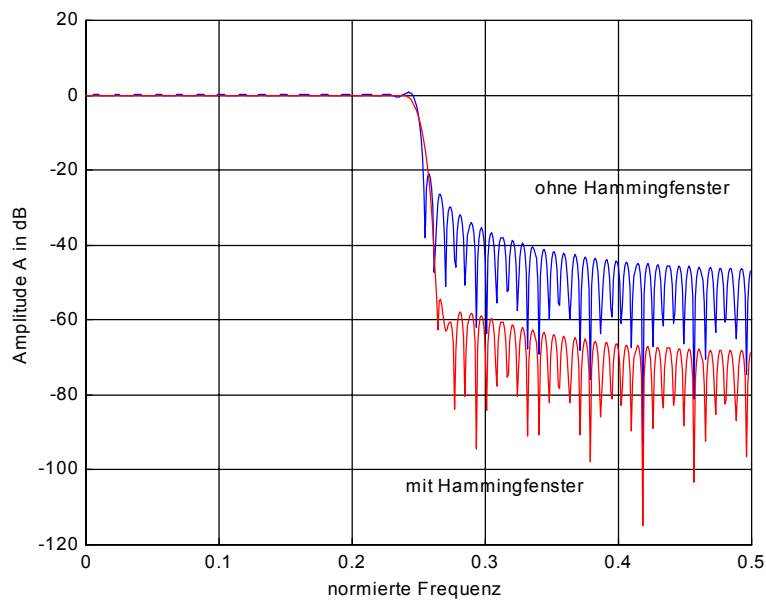


Abbildung 3: Amplitudengang FIR-Tiefpassfilter

Ein grosser Vorteil von FIR-Filtern ist deren linearer Phasengang. Ein solcher Phasengang hat zur Folge, dass alle Frequenzanteile eines Signals beim Durchlaufen des Filters dieselbe Laufzeit haben. Dadurch gibt es im gefilterten Signal keine Phasenverzerrungen. Abbildung 4 verdeutlicht den linearen Phasengang eines FIR-Filters. Es handelt sich dabei um das selbe Filter wie in Abbildung 2 und 3.

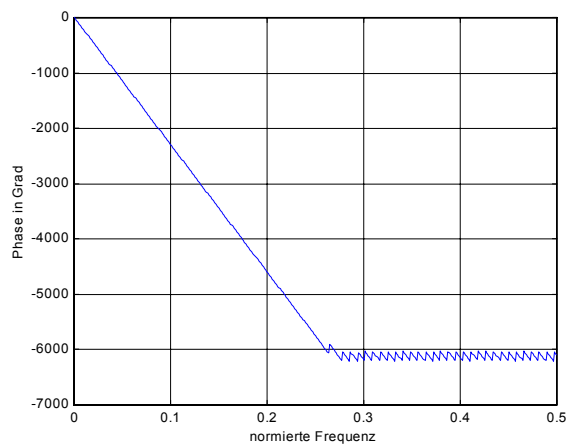


Abbildung 4: Phasengang FIR-Tiefpassfilter

Verzögerungsglieder mit FIR-Filtern

Mit FIR-Filtern lassen sich auch reine Verzögerungsglieder realisieren. Dazu sind die Filterkoeffizienten wie folgt zu wählen:

$$b_0 \dots b_{M-1} = 0, \quad b_M = 1 \quad M = 2, 3, 4, \dots \quad (4.12)$$

Eine solche Koeffizientenwahl hat zur Folge, dass das Eingangssignal um M Werte verzögert wird. Die Abbildung 5 zeigt ein Rechtecksignal $r[n]$ mit Periode $T = 32$ und $r[0] = 1$, welches mit einem FIR-Verzögerungsglied um $M = 10$ Werte verzögert wurde. Damit man den Verschiebungseffekt besser sieht, wurde das verzögerte Signal anschliessend noch mit dem Faktor 1.5 multipliziert.

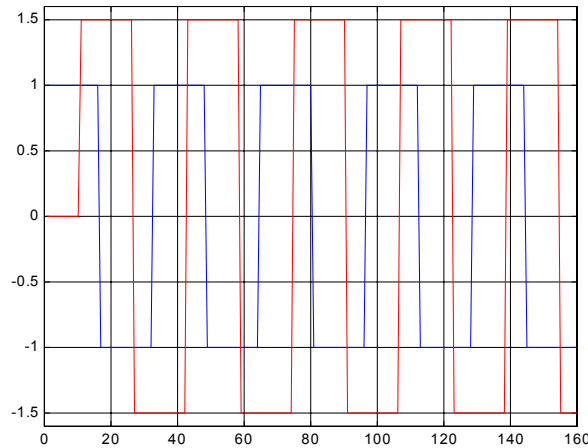


Abbildung 5: Rechtecksignal vor und nach Verzögerungsglied

4.3 Änderung der Abtastrate

Die Verarbeitung der Eingangssignale für den SII kann durch nachträgliche Änderung der Abtastrate an einem diskreten Signal stark optimiert werden. Dabei sind für diese Arbeit folgende zwei Methoden relevant:

- Halbierung der Abtastrate
- Verdoppelung der Abtastrate

4.3.1 Halbierung der Abtastrate

Bei einem Signal, welches nur Frequenzanteile mit $f < f_{Ny}/2$ hat, kann die Abtastrate auf die Hälfte reduziert werden. Bei diskreten Signalen wird dies durch Entfernen jedes zweiten Abtastwertes erreicht. Es spielt dabei keine Rolle, ob alle geraden Werte oder alle ungeraden entfernt werden. In beiden Fällen liegt danach eine Zahlenfolge vor, die man beim Abtasten mit halber Abtastrate auch hätte erreichen können. Abbildung 6 zeigt das Symbol für eine Halbierung der Abtastrate.

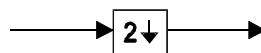


Abbildung 6: Blockschaltbild für Halbierung der Abtastrate

Für die Halbierung der Abtastrate eignet sich die Matlab-Funktion `reshape()`, welche zum Standardpaket von Matlab gehört. Dabei ist wichtig, dass die Eingangszahlenfolge von `reshape()` eine durch zwei teilbare Anzahl Werte aufweist.

4.3.2 Verdoppelung der Abtastrate

Die Verdoppelung der Abtastrate ist die klassische Interpolationsaufgabe der digitalen Signalverarbeitung. Zuerst wird zwischen je zwei Abtastwerten ein neuer Abtastwert eingefügt, und zwar eine Null. Anschliessend erfolgt eine Filterung mit einem Halbband-FIR-Tiefpassfilter, wie er in 4.2 beschrieben wurde. Im weiteren muss dann noch zusätzlich eine Verstärkung um den Faktor zwei erfolgen. Dieses Vorgehen funktioniert, weil das mit Nullen ergänzte Signal folgendes Spektrum aufweist:

- Das gewünschte Signalspektrum im Frequenzintervall $[0; 0.25]$, aber um den Faktor 0.5 abgeschwächt (normierte Nyquistfrequenz $f_{Ny} = 0.5$) und
- das Spiegelbild dazu im Frequenzintervall $[0.25; 0.5]$.

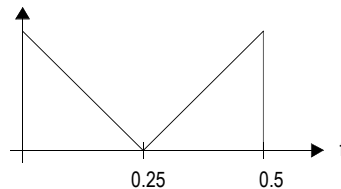


Abbildung 7: Frequenzspektrum nach Verdoppelung der Abtastrate durch Einfügen von Nullen

Abbildung 8 zeigt das Blockschaema für eine Verdoppelung der Abtastrate:

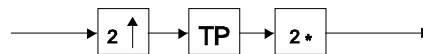


Abbildung 8: Blockschaema für Verdoppelung der Abtastrate

Die Matlab-Funktion, welche eine Verdoppelung der Abtastrate durchführt, heisst Upsample2() und befindet sich in der Beilage unter 9.2.3.

5 Digitale Verarbeitung der Eingangssignale für SII

In diesem Abschnitt werden die Hard- und Softwareeinrichtungen, welche für die Ermittlung des SII und für die Durchführung von Testprogrammen notwendig sind, erläutert.

5.1 Hardware

Bei der Berechnung des SII gibt es in unserer Arbeit keine Echtzeitanforderungen. Es ist somit kein digitaler Signalprozessor (DSP) erforderlich. Es reicht ein einfacher PC, ausgerüstet mit Soundkarte, Mikrophon und Kopfhörer.

5.1.1 Abtastrate, Quantisierung

Das von uns zu testende Sprachsignal wird als erstes mit dem Windows 95 Audiorecorder aufgenommen. Mit diesem Audiorecorder ist es nicht möglich mit einer Abtastrate von 16000 Hz abzutasten. Diese Abtastrate ist mindestens nötig, weil der SII Frequenzen bis 8000 Hz berücksichtigt. In unserem Fall erfolgt das Abtasten mit 22050 Hz sowie mit $n = 16$ Quantisierungsstufen. Da eine so hohe Abtastrate nichts einbringt, nutzen wir die Matlab-Funktion `resample()`, um die Abtastrate auf 16000 Hz zu reduzieren. Das durch die Quantisierung entstehende Rauschen beeinflusst natürlich unser zu testendes Sprachsignal. Um den so entstandenen Fehler zu beurteilen, soll das Signal zu Rauschverhältnis (SNR) gemäss Gleichung (4.3) berechnet werden:

$$SNR = 6 \cdot 16 = 96dB$$

Da dieser Unterschied so gross ist, fällt das Quantisierungsrauschen für die SII Berechnung nicht ins Gewicht und wird deshalb für die weiterführenden Berechnungen nicht berücksichtigt.

5.1.2 Eichung der Aufnahmelautstärke

Vor der Aufnahme von Sprachsignalen muss als erstes die Aufnahmelautstärke auf einen geeigneten Wert geeicht werden, damit der A/D- Wandler nicht übersteuert wird. Dazu wird irgend ein Text mit normaler Lautstärke ins Mikrophon gesprochen. Der Abstand des Sprechers zum Mikrophon soll dabei so gewählt werden, wie er bei den Aufnahmen der Sprachsignale für die Tests sein wird; d.h. etwa 50 cm. Die Aufnahmelautstärke ist dann optimal gewählt, wenn von den 16 Quantisierungsbits etwa 14 ausgenutzt werden. So ist es möglich, dass der A/D-Wandler auch beim Sprechen mit erhobener Lautstärke noch nicht übersteuert wird. Mit dem Programm Real-Player kann auf einfache Weise ermittelt werden, welche Quantisierungsbits ausgenutzt werden.

5.2 Software

Alle erforderlichen Programme zur digitalen Signalverarbeitung sind in Matlab, Version 5.2, implementiert. Nachfolgend wird deren Konzept beschrieben. Eine detaillierte Beschreibung der dabei verwendeten Standardfunktionen von Matlab kann man aus [6] entnehmen.

5.2.1 Filterbank mit FIR-Filtern

Die Aufteilung der Eingangssignale für die Berechnung des SII erfolgt über eine Filterbank, welche mit FIR-Tiefpässen realisiert wird. Die Abtastrate von 16 kHz erlaubt es leider nicht, das oberste Band des Sprachsignals um 8 kHz zur Berechnung des SII zu berücksichtigen. Im Vergleich zu den wichtigsten Bändern, welche eine Bandwichtigkeit von 0.08 und mehr aufweisen, ist dies aber nicht weiter relevant, denn die Bandwichtigkeit des Bandes um 8 kHz beträgt bloss 0.0185, ist also ca. 4.5 mal kleiner.

Abbildung 1 zeigt das Blockschema der Filterbank. Auf der rechten Seite sieht man die entsprechenden Teilbänder, angegeben durch deren Mittenfrequenz f_i . Die obere und untere Grenze des jeweiligen Bandes berechnet sich mit folgenden Formeln:

$$f_u = f_i \cdot 2^{-\frac{1}{6}} \quad (5.1)$$

untere Grenzfrequenz

$$f_o = f_i \cdot 2^{\frac{1}{6}} \quad (5.2)$$

obere Grenzfrequenz

Die Grenzfrequenz des Eingangstiefpasses berechnet sich zu:

$$f_g = 6300 \cdot 2^{\frac{1}{6}} = 7072 \text{ Hz} \quad (5.3)$$

Die Grenzfrequenz aller folgenden Tiefpassfilter ist gleich der unteren Grenzfrequenz des jeweiligen Teilbandes.

Sobald die obere Grenzfrequenz eines Bandes eine Oktave tiefer ist als das Eingangssignal, kann die Abtastrate auf die Hälfte reduziert werden. Bei der vorgegebenen Aufteilung in sechs Oktaven, ist das somit fünfmal möglich. Diese Reduktion erlaubt es, für tiefere Frequenzen FIR-Filter mit derselben Anzahl Koeffizienten zu verwenden wie bei höheren Frequenzen.

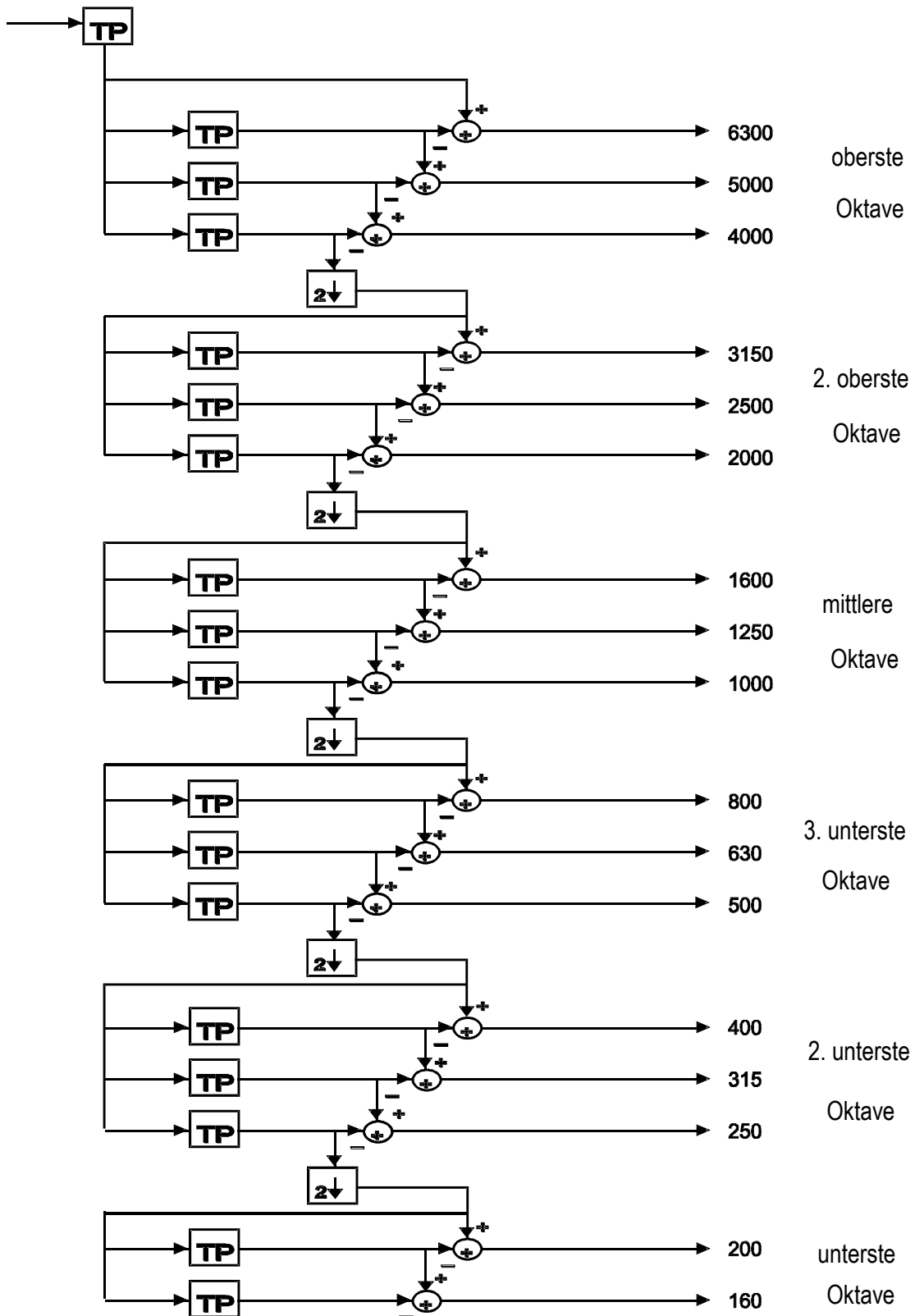


Abbildung 9: Blockschema Filterbank

Abbildung 2 zeigt den Amplitudengang der verwendeten Filterbank für die oberste Oktave. Die darin enthaltenen FIR-Filter besitzen 129 Koeffizienten.

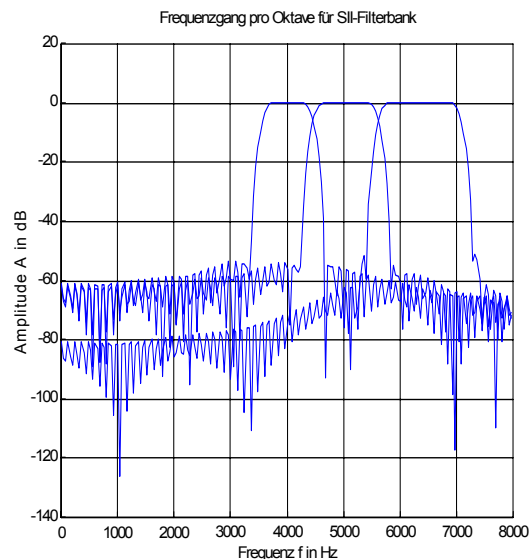


Abbildung 10: Amplitudengang oberste Oktave

Infolge der Reduktion der Abtastrate um die Hälfte nach jeder Oktave, sieht der Amplitudengang der Filterbank für die tiefer liegenden Oktaven genau gleich aus. Die Skalierung der Frequenzachse muss dabei einfach für jede tiefere Oktave halbiert werden.

Die Matlab-Funktion zur SII-Filterbank befindet sich in der Beilage unter 9.2.4.

Die Anzahl Werte der Eingangssignale für die Filterbank muss infolge der Reduktionen der Abtastraten mindestens fünfmal durch zwei geteilt werden können. Dies ist in der Regel nicht gegeben. Zwar ist anzunehmen, dass die Anzahl Werte einiges grösser ist als $2^5 = 32$, aber sie wird im allgemeinen kein Vielfaches von 32 sein. Deshalb haben wir eine Funktion geschrieben, welche einem diskreten Signal noch so viele Nullen anfügt, bis die Anzahl Werte durch 32 teilbar ist. Die Funktion findet man in der Beilage unter 9.2.5.

5.2.2 Eichung der Teilbandsignalpegel auf die normierten Pegel des SII

Mit der Matlab-Standardfunktion `wavread()` kann ein vom Audiorecorder aufgenommenes und unter `*.wav` abgespeichertes Audiosignal in einen Signalvektor umgewandelt werden. `Wavread()` normiert dieses Signal auf den Bereich `[-1;1]`. Die Pegel der Eingangssignale für den SII müssen aber auf ein genau normiertes Spektrum bezogen sein. Bei diesem Spektrum handelt es sich um das Spektrum eines Sprachsignals mit normaler Lautstärke, welches folgenden Gesamtpegel (overall speech level, SPL) aufweist:

$$SPL = 62.35dB \quad (5.4)$$

Deshalb wird eine softwaremässige Eichung benötigt. Dazu wird als erstes ein Sprachsignal mit normaler Lautstärke (wie in 5.1.2) aufgenommen.

Von diesem Sprachsignal wird die Leistung wie folgt berechnet:

$$P = \frac{1}{N} \cdot \sum_{n=0}^{N-1} (s[n])^2 \quad (5.5)$$

N : Anzahl Signalwerte

s[n] : Sprachsignal

Diese Signalleistung wird mit dem SPL gemäss (5.4) verglichen und daraus ein Eichungsfaktor A_f gebildet.

$$A_f = \frac{10^{\frac{SPL}{10}}}{P} \quad (5.6)$$

Die Teilbandleistungen $P(f_i)$ jedes von nun an aufgenommenen Sprachsignals werden mit diesem Anpassungsfaktor A_f multipliziert und in dB dargestellt.

$$E_B(f_i) = 10 \cdot \lg(A_f \cdot P(f_i)) \quad (5.7)$$

$E_B(f_i)$: Pegel des ganzen Teilbandes mit Mittenfrequenz f_i , in dB

Jetzt beziehen sich die Teilbandpegel auf das normierte Spektrum. Gemäss ANSI werden jedoch nur die Pegel plus minus 1 Hz um die Mittenfrequenzen benötigt, deshalb müssen wir noch eine von ANSI angegebene Korrektur anbringen.

$$E_T(f_i) = E_B(f_i) - \Delta_i \quad (5.8)$$

Δ_i : Differenz zur Berechnung des Pegels ± 1 Hz um eine Mittenfrequenz, wenn nur der Pegel des gesamten Teilbandes zur Verfügung steht (band with adj), in dB

Die erhaltenen Teilbandpegel $E_T(f_i)$ entsprechen nun dem Format, welches für die Eingangsgrössen des SII benötigt wird.

Die Matlab-Funktionen zu dieser Eichung befinden sich in der Beilage unter 9.2.6.

5.2.3 Erzeugung eines Störsignals für Testzwecke

Für unsere Tests ist ein Rauschsignal erforderlich, dessen Teilbänder dieselben Leistungen aufweisen wie die Teilbänder des zu vergleichenden Sprachsignals. Mit der Matlab-Funktion `randn()` wird zuerst ein weisses Rauschen erzeugt. Dieses wird anschliessend mit der SII-Filterbank in die entsprechenden Teilbänder zerlegt. Ebenfalls wird das Sprachsignal, mit welchem das Störsignal verglichen werden soll, in seine Teilbänder zerlegt. Nun werden die Leistungen pro Teilband für das Sprachsignal P_{Spr_i} und das Störgeräusch $P_{Stö_i}$ ermittelt. Pro Teilband kann nun festgestellt werden, um wieviel die Sprachsignalleistung höher ist als die Rauschsignalleistung.

$$G_i = \frac{P_{Spr_i}}{P_{Stö_i}} \quad (5.9)$$

G_i : Verhältnis der Teilbandleistungen des Sprachsignals zum Störsignal

Nun sind die Teilbandsignale des Störgeräusches nur noch mit der Wurzel des entsprechenden Faktors G_i zu multiplizieren und können durch eine Rekonstruktion zum erforderlichen Störsignal zusammengesetzt werden.

$$s_{stöt_i}[n] = \sqrt{G_i} \cdot s_{stö_i}[n] \quad (5.10)$$

$s_{stö_i}[n]$: Störsignal des Teilbandes i

$s_{stöt_i}[n]$: Störsignal für Testzwecke des Teilbandes i

Die Matlab-Funktion dazu befindet sich in der Beilage unter 9.2.7.

5.2.4 Rekonstruktion eines Signals anhand seiner Teilbänder

Neben der Zerlegung eines Signals in Teilbänder, ist es in unserer Arbeit auch erforderlich, ein Signal nach der Zerlegung anhand seiner Teilbandsignale zu rekonstruieren.

Da die einzelnen Teilbandsignale mit unterschiedlichen Abtastraten erzeugt wurden, kann die Rekonstruktion nicht durch bloße Addition erfolgen, d.h. vor der Addition müssen alle beteiligten Teilbänder dieselbe Abtastrate aufweisen. Ähnlich wie bei der Zerlegung in Teilbänder erfolgt auch das Zusammenfügen zum ursprünglichen Signal durch schrittweises Vorgehen:

Zuerst werden die beiden Teilbänder der untersten Oktave zusammengefügt und anschliessend wird die Abtastrate verdoppelt, damit man sie zu den Teilbändern der 2. untersten Oktave addieren kann. Bei diesem Summensignal erfolgt wiederum eine Verdoppelung der Abtastrate und eine anschliessende Addition zur nächsthöheren Oktave. Dieses Vorgehen wird noch dreimal wiederholt, bis man bei der obersten Oktave angelangt ist. Am Ausgang liegt dann das rekonstruierte Signal vor.

Abbildung 3 illustriert dieses Vorgehen.

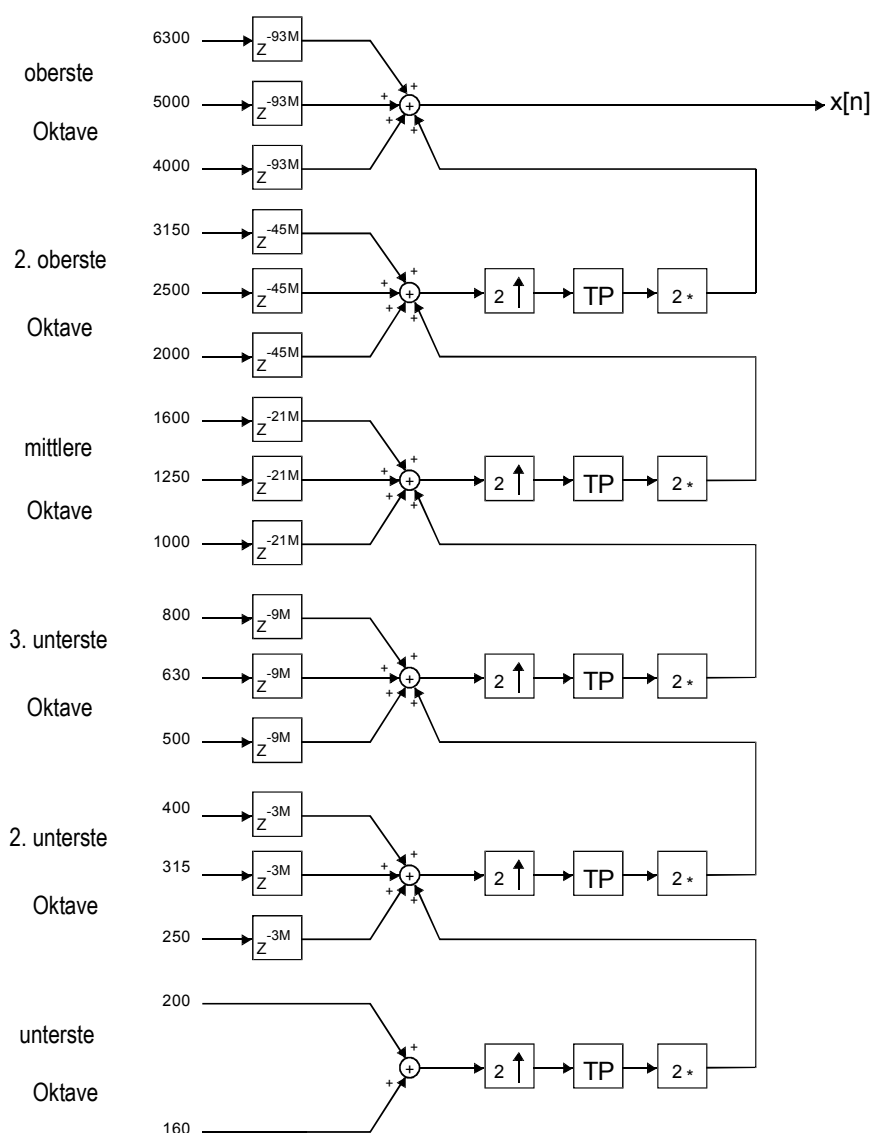


Abbildung 11: Rekonstruktionsbank

Die Matlabfunktion für diese Rekonstruktionsbank befindet sich in der Beilage unter 9.2.8.

Verzögerungsglieder für Teilbänder

In der Gleichung (4.10) auf Seite 10 wurde erläutert, dass jedes FIR-Filter eine Signalverzögerung um M Werte einführt. Diese Begebenheit ist bei der Rekonstruktion zu berücksichtigen, indem für jedes Teilband Verzögerungsglieder eingebaut werden. Dabei ist nicht so einfach ersichtlich, wie gross die einzelnen Verzögerungen jeweils sein müssen. Die Signale der einzelnen Oktaven durchlaufen nämlich sowohl bei der Zerlegung als auch bei der Rekonstruktion eine verschiedene Anzahl FIR-Filter und verschieben sich somit gegenseitig. Dabei haben alle FIR-Filter sowohl der Filterbank als auch der Rekonstruktionsbank dieselbe Anzahl Filterkoeffizienten. Abbildung 4 illustriert den Signalweg, welcher ein Signal dabei durchläuft.

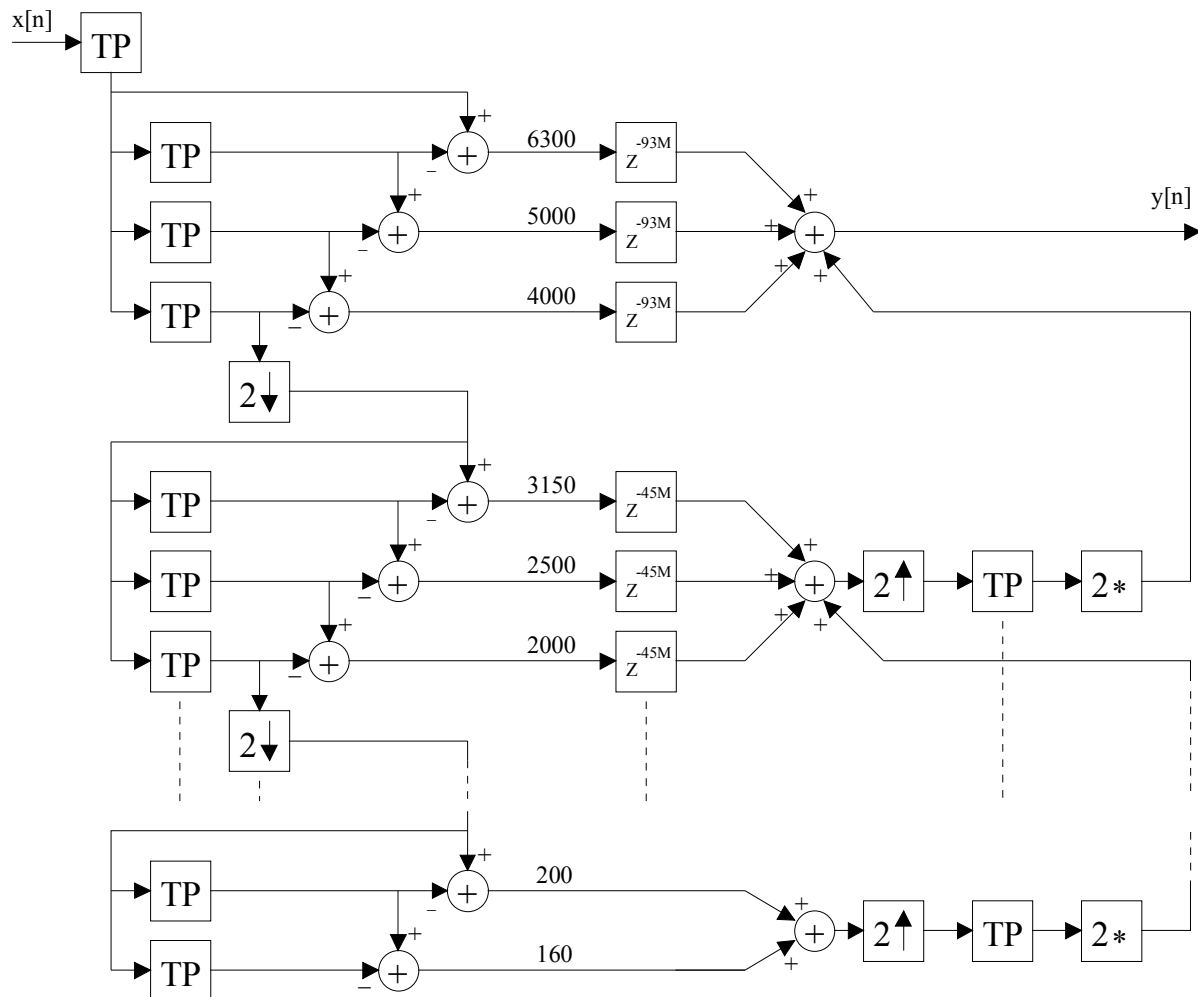


Abbildung 12: Signalweg für Zerlegung in Teilbänder und anschliessende Rekonstruktion

Für die Ermittlung der einzelnen Verzögerungen kann man auf zwei Arten vorgehen:

- Wahl eines geeigneten Signals, welches einen Verlauf aufweist, das man die Verzögerungen anhand der Positionen von markanten Stellen ermitteln kann, wenn man das Signal an unterschiedlichen Orten des Signalweges graphisch betrachtet.
- Ermittlung der jeweiligen Verzögerungen durch rein theoretische Überlegungen.

Als erstes wollen wir uns mit der graphischen Ermittlung der einzelnen Verzögerungen befassen. Dazu wird ein Impuls, wie es Abbildung 5 zeigt, verwendet.

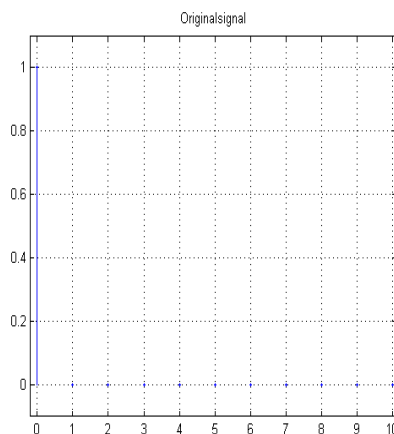


Abbildung 13: Testimpuls

Dieser Impuls wird nun durch die Filterbank in Teilbänder zerlegt. Anschliessend erfolgt eine Rekonstruktion mit der Rekonstruktionsbank ohne Verzögerungsglieder an den Eingängen. Die erste markante Stelle für die Ermittlung von Verzögerungen ist der Addierer bei der 2. untersten Oktave, denn bei der untersten Oktave müssen keine Verzögerungen eingeführt werden. Abbildung 6 zeigt einerseits das Summensignal der Teilbänder der 2. untersten Oktave und andererseits das Signal der untersten Oktave nach dem Upsampling.

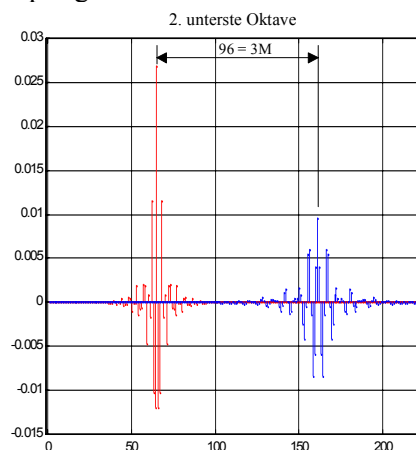


Abbildung 14: Signale beim Addierer der 2. untersten Oktave

Selbstverständlich haben die Signale nach der Filterbank nicht mehr exakt den Verlauf des Testimpulses. Sie weisen aber einen Verlauf mit einem Peak auf, der links- und rechtsseitig abklingt. Das Signal der 2. untersten Oktave hat den Peak zuerst. Das Signal der untersten Oktave nach dem Upsampling hat den Peak später. Die Position der Peaks ist für die Ermittlung der notwendigen Verzögerungen von zentraler Bedeutung, denn die Differenz der Werte zwischen den beiden Peaks entspricht exakt der Anzahl Verzögerungen, die die Signale der untersten Oktave mehr erfahren haben als die Signale der 2. untersten. Diese Differenz beträgt hier 96 Werte. Damit also das Summensignal nach dem Addierer keine zeitlichen Verschiebungen unter den Teilbandsignalen der dabei beteiligten Oktaven aufweist, sind die Signale der 2. untersten Oktave um 96 Werte zu verzögern.

Es ist im weiteren erforderlich, dass diese Werteanzahl allgemein in Funktion der Verzögerung M eines FIR-Filters angegeben werden kann. Im vorliegenden Fall haben alle FIR-Filter sowohl der Filterbank als auch der Rekonstruktionsbank $N = 65$ Filterkoeffizienten. Somit fügt jedes FIR-Filter eine Verzögerung $M = 32$ Werte ein. Die Verzögerung V_5 für die 2. unterste Oktave kann nun wie folgt allgemein durch M ausgedrückt werden.

$$V_5 = \frac{96}{32} \cdot M = 3M \quad (5.11)$$

Als nächstes betrachten wir nun die Signale beim Addierer der 3. untersten Oktave, um die Anzahl Verzögerungswerte der dort anzubringenden Verzögerungsglieder zu ermitteln. Abbildung 7 zeigt die dazugehörige Graphik:

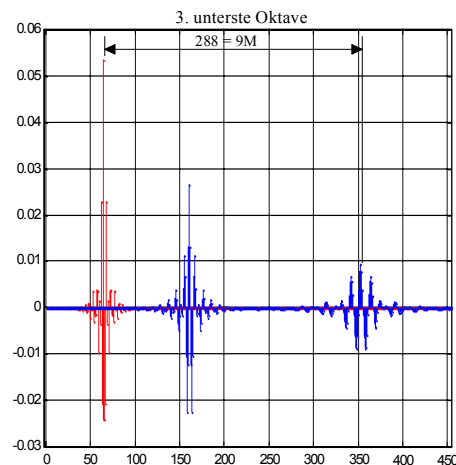


Abbildung 15: Signale beim Addierer der 3. untersten Oktave

Beim Signal, welches den Peak zeitlich zuerst hat, handelt es sich um das Summensignal der 3. untersten Oktave. Das Signal mit dem Peak an zweiter Stelle, ist das Summensignal der 2. untersten Oktave. Das Summensignal der untersten Oktave hat ihren Peak an letzter Stelle. Die Signale der untersten und der 2. untersten Oktave sind hier noch zeitlich verschoben, weil die notwendigen Verzögerungsglieder noch nicht angebracht wurden. Die Anzahl einzuführender Verzögerungswerte für die 3. unterste Oktave erhält man nun aus der Wertedifferenz des Peaks der 3. untersten Oktave und des Peaks der untersten Oktave. Diese beträgt 288 Werte. Auch diese Verzögerung V_4 kann allgemein durch M ausgedrückt werden:

$$V_4 = \frac{288}{32} \cdot M = 9M \quad (5.12)$$

Für die Ermittlung der erforderlichen Verzögerungen der weiteren Oktaven bleibt das Prinzip dasselbe. Der zeitlich erste Peak in der Graphik wird erzeugt durch die Summensignale der neu dazukommenden Oktave, alle andern Peaks stammen von den unteren Oktaven. Die Differenz zum zeitlich letzten Peak ist massgebend für die erforderliche Verzögerung. Abbildungen 8-10 zeigen die Signale bei den entsprechenden Addierern:

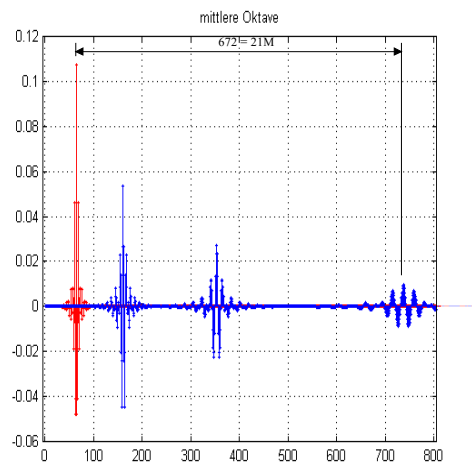


Abbildung 16: Signale beim Addierer der mittleren Oktave

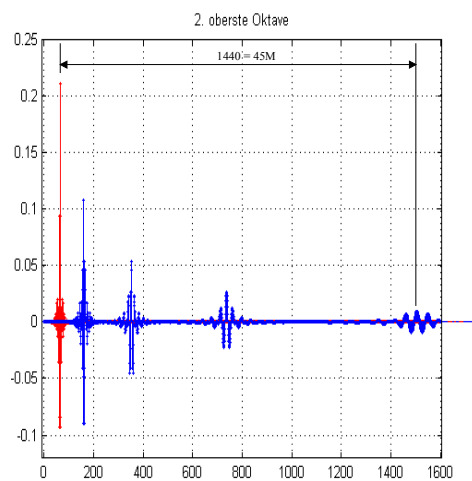


Abbildung 17: Signale beim Addierer der 2. obersten Oktave

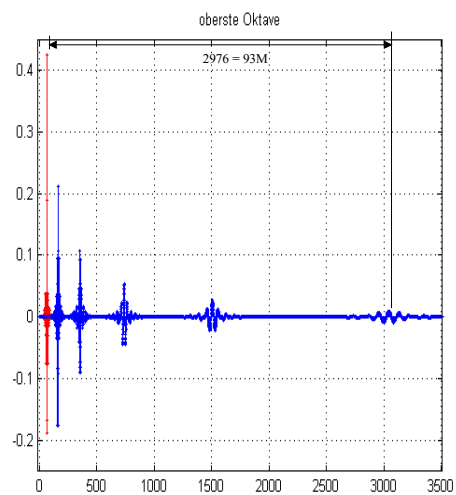


Abbildung 18: Signale beim Addierer der obersten Oktave

Die noch nicht ermittelten Verzögerungen berechnen sich somit wie folgt:

$$\text{mittlere Oktave:} \quad V_3 = \frac{672}{32} \cdot M = 21M \quad (5.13)$$

$$\text{2. oberste Oktave:} \quad V_2 = \frac{1440}{32} \cdot M = 45M \quad (5.14)$$

$$\text{oberste Oktave:} \quad V_1 = \frac{2976}{32} \cdot M = 93M \quad (5.15)$$

In einem weiteren Schritt sollen die notwendigen Verzögerungen durch rein theoretische Überlegungen ermittelt werden. Dazu werden als erstes drei Voraussetzungen festgehalten:

1. Alle verwendeten FIR-Filter haben dieselbe Anzahl Filterkoeffizienten N und führen somit überall dieselbe Verzögerung M ein, wobei:

$$M = \frac{N-1}{2} \quad (5.16)$$

2. Damit vor der Halbierung der Abtastrate immer noch eine gerade Anzahl Verzögerungswerte vor dem eigentlichen Signal anliegt, soll M folgende Bedingung erfüllen:

$$M = 2^k \quad k \geq 5 \quad (5.17)$$

3. Bei der Reduktion der Abtastrate werden die geraden Werte entfernt.

Die Voraussetzungen 2 und 3 dienen dazu, dass der jeweils erste Wert eines Signals in den nun folgenden Betrachtungen immer vorhanden ist, d.h. durch die Halbierung der Abtastrate nie wegfallen kann.

Nun werden zuerst jene Verzögerungen betrachtet, welche durch die Filterbank eingeführt werden:

Die Verzögerungen durch den Eingangstiefpass und die Tiefpässe der obersten Oktave muss nicht berücksichtigt werden, weil das gesamte Signal dort verzögert wird, und sich somit die einzelnen Teilbänder noch nicht gegenseitig verschieben können. Am Eingang des obersten Downsamplers liegt nun das Signal x_1 an. Dieses besteht aus den diskreten Werten $x_1 = [X_1, X_2, X_3, \dots, X_n]$. Der Index n soll dabei den Wert

$$n = 2^m \quad m > 5 \quad (5.18)$$

haben.

Der Grund für $m > 5$ ist, dass anschliessend noch fünfmal eine Halbierung der Abtastrate durchgeführt wird, und somit das Signal in den untersten Teilbändern noch die Werteanzahl Z aufweist. Diese Werteanzahl soll immer grösser als eins sein.

$$Z = \frac{2^m}{2^5} = 2^{m-5} > 1 \quad (5.19)$$

Die folgende Zusammenstellung zeigt nun Schritt für Schritt, welche Verzögerungen durch die Filterbank eingeführt werden. Dabei bedeutet der Wert, der zwischen zwei geschweiften Klammern {} steht, die Länge der eingeführten Verzögerung.

Signal vor 1. Downsampling	: $x_1 = [X_1, X_2, X_3, \dots, X_n]$
Signal nach erstem Downsampling	: $x_2 = [X_1, X_3, X_5, \dots, X_{n-1}]$
Ausgangssignal der 2. obersten Oktave	: $x_3 = [\{M\}, X_1, X_3, X_5, \dots, X_{n-1}]$
Signal nach zweitem Downsampling	: $x_4 = [\{M/2\}, X_1, X_5, \dots, X_{n-3}]$
Ausgangssignal der mittleren Oktave	: $x_5 = [\{M+M/2\}, X_1, X_5, \dots, X_{n-3}]$
Signal nach drittem Downsampling	: $x_6 = [\{3M/4\}, X_1, X_9, \dots, X_{n-7}]$
Ausgangssignal der 3. untersten Oktave	: $x_7 = [\{M+3M/4\}, X_1, X_9, \dots, X_{n-7}]$
Signal nach viertem Downsampling	: $x_8 = [\{7M/8\}, X_1, X_{17}, \dots, X_{n-23}]$
Ausgangssignal der 2. untersten Oktave	: $x_9 = [\{M+7M/8\}, X_1, X_{17}, \dots, X_{n-23}]$
Signal nach fünftem Downsampling	: $x_{10} = [\{15M/16\}, X_1, X_{33}, \dots, X_{n-55}]$
Ausgangssignal der untersten Oktave	: $x_{11} = [\{31M/16\}, X_1, X_{33}, \dots, X_{n-55}]$

Jedes FIR-Filter fügt eine Verzögerung um M Werte ein. Durch die Halbierung der Abtastrate wird die gesamte Verzögerung auf die Hälfte reduziert. Der erste eigentliche Signalwert X_1 kann in Folge der oben gemachten Voraussetzungen nie wegfallen.

Für die Rekonstruktionsbank wird angenommen, dass nur an der untersten Oktave ein Signal anliegt, und zwar das Signal $y_{11} = x_{11}$ aus der vorangegangenen Zusammenstellung. Für die oberen Oktaven heisst dies, dass keine neuen Werte ausser null hinzukommen. Diese Vereinfachung kann man machen, weil es ja nur um die Ermittlung der Verzögerungen geht. Auch in diesem Fall wird eine Zusammenstellung wie oben durchgeführt:

Signal vor erstem Upsampling	: $y_{11} = [\{31M/16\}, X_1, X_{33}, \dots, X_{n-55}]$
Signal nach erstem Upsampling	: $y_{10} = [\{31M/8\}, X_1, 0, X_{33}, 0, \dots, X_{n-55}]$
Signal bei Addierer 2. unterste Oktave	: $y_9 = [\{M+31M/8\}, X_1, 0, X_{33}, 0, \dots, X_{n-55}, 0]$
Signal nach zweitem Upsampling	: $y_8 = [\{39M/4\}, X_1, 0, 0, 0, X_{33}, \dots]$
Signal bei Addierer 3. unterste Oktave	: $y_7 = [\{M+39M/4\}, X_1, 0, 0, 0, X_{33}, \dots]$
Signal nach drittem Upsampling	: $y_6 = [\{43M/2\}, X_1, 0, \dots]$
Signal bei Addierer mittlere Oktave	: $y_5 = [\{M+43M/2\}, X_1, 0, \dots]$
Signal nach viertem Upsampling	: $y_4 = [\{45M\}, X_1, 0, \dots]$
Signal bei Addierer 2. oberste Oktave	: $y_3 = [\{M+45M\}, X_1, 0, \dots]$
Signal nach fünftem Upsampling	: $y_2 = [\{92M\}, X_1, 0, \dots]$
Signal bei Addierer oberste Oktave	: $y_1 = [\{M+92M\}, X_1, 0, \dots]$

Jedes Verdoppeln der Abtastrate verdoppelt auch die Anzahl Verzögerungswerte. Zusätzlich erfolgt bei jedem Tiefpassfilter wiederum eine Verzögerung um M Werte.

Für jede Oktave beginnt das eigentliche Summensignal beim Wert X_1 , welcher zahlenmässig nicht immer derselbe sein muss. Die entsprechenden Verzögerungen V_i lassen sich nun berechnen, indem man die Differenz der Anzahl Verzögerungswerte zwischen dem Signal beim jeweiligen Addierer $V(y_i)$ und dem Ausgangssignal der entsprechenden Oktave $V(x_j)$ bestimmt.

$$V_i = V(y_i) - V(x_j) \quad i = 1, 2, 3, 4, 5, 6; \quad j = 1, 3, 5, 7, 9, 11 \quad (5.20)$$

Auf diese Weise erhält man die genau gleichen Resultate wie bei der graphischen Ermittlung.

Funktionstest für Rekonstruktionsbank

Damit die Rekonstruktionsbank auf ihre richtige Funktionsweise getestet werden kann, wird ein Impuls der Höhe eins, wie er in Abbildung 11 dargestellt ist, als Testsignal verwendet. Weil die Filterbank im Eingang eine Tiefpassfilterung durchführt, wird der Impuls zunächst über ein FIR-Tiefpassfilter mit der Grenzfrequenz $f_g = 7072 \text{ Hz}$ geführt. Abbildung 12 zeigt das Ausgangssignal nach dem Tiefpassfilter. Dieses wird im folgenden als Originalsignal bezeichnet. Anschliessend wird das Signal über die Filterbank in Teilbänder zerlegt, welche danach der Rekonstruktionsbank zugeführt werden. Die Funktionsweise der Rekonstruktionsbank ist dann exakt, wenn an deren Ausgang dasselbe Signal wie am Ausgang des Tiefpassfilters anliegt, einfach um die entsprechende Verzögerung verschoben.

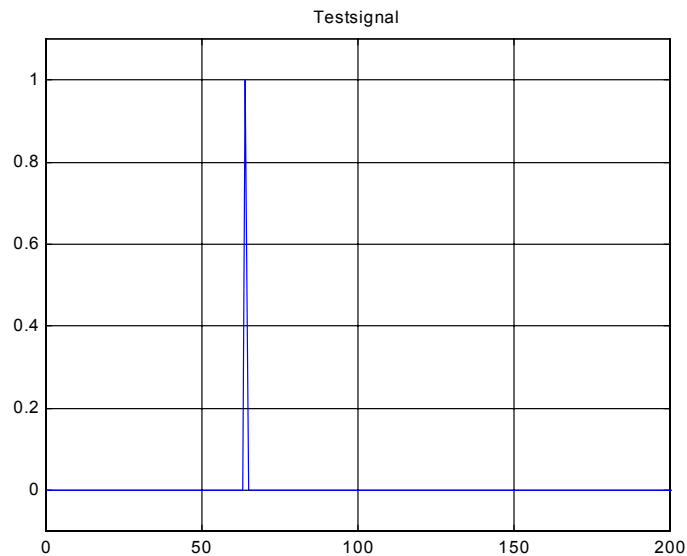


Abbildung 19: Testimpuls

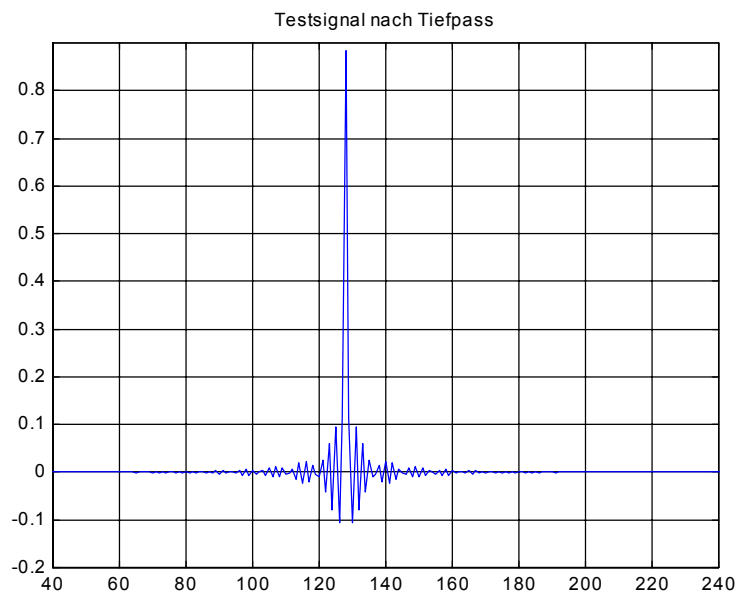


Abbildung 20: Ausgangssignal Tiefpassfilter (Originalsignal)

Dieser Test wird nun mit der Rekonstruktionsbank durchgeführt. Alle vorkommenden Tiefpassfilter haben dabei $N = 129$ Koeffizienten. Als Ausgangssignal erscheint das Signal gemäss Abbildung 13.

Dieses erfüllt die oben genannten Bedingungen.

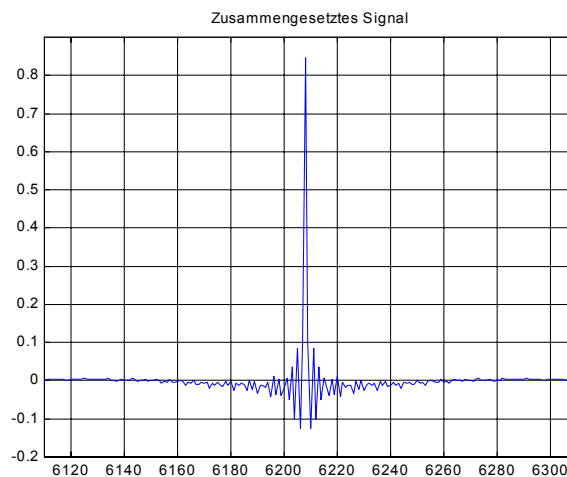


Abbildung 21: Ausgangssignal nach Rekonstruktion

Um die richtige Funktion der Rekonstruktionsbank noch etwas besser zu verdeutlichen wird nun das Originalsignal um so viele Werte nach rechts verschoben wie die Verzögerungen, welche durch die Filterbank und die Rekonstruktionsbank eingefügt wurden. Diese Verzögerung V beträgt:

$$V = 93M + 2M = 95M = 95 \cdot 64 = 6080 \quad (5.21)$$

Die Verzögerung V setzt sich dabei zusammen aus der Verzögerung von $93M$ zwischen der obersten Oktave der Filterbank und dem Ausgangssignal der Rekonstruktionsbank plus der Verzögerung von $2M$, welche die beiden Tiefpassfilter am Eingang der Filterbank einfügen.

In der Abbildung 14 wird nun das verzögerte Originalsignal und das rekonstruierte Signal übereinandergelegt. Die beiden Signale überdecken sich beinahe, was für eine richtige Funktionsweise der Rekonstruktionsbank spricht.

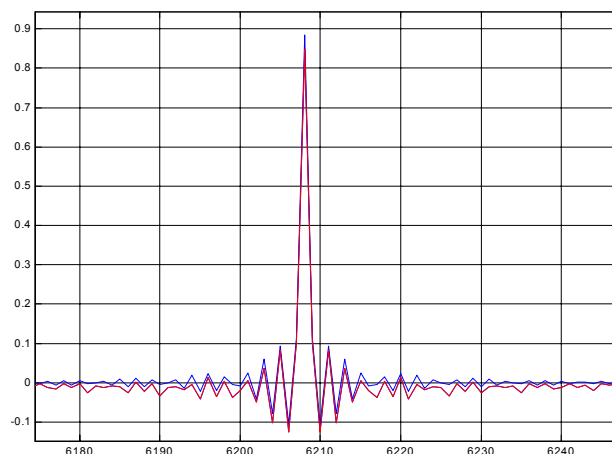
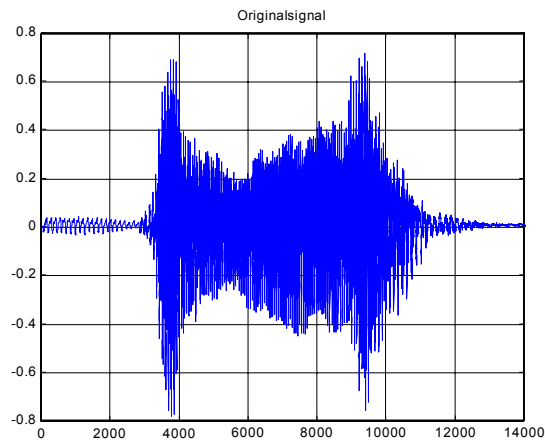
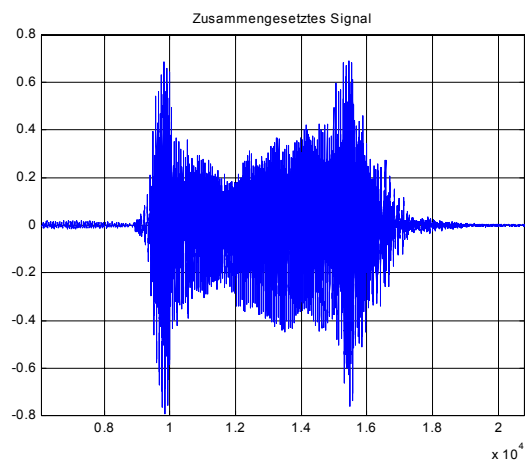
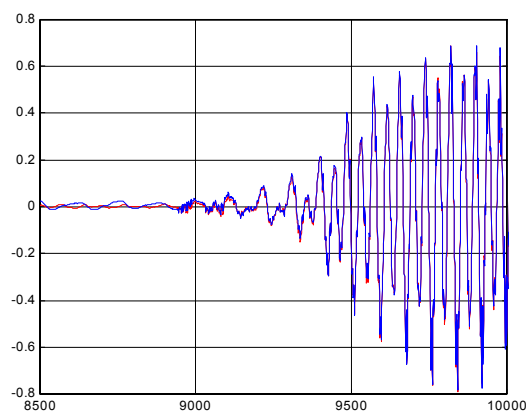


Abbildung 22: verzögertes Originalsignal und rekonstruiertes Signal übereinandergelegt

Damit man die Wirkungsweise der Rekonstruktionsbank auch anhand eines Sprachsignals sehen kann, wird nun im folgenden mit einem Sprachsignal derselbe Test durchgeführt. Beim Sprachsignal handelt es sich um den Buchstaben "B" gesprochen von Markus Hurschler. Es wurden wiederum 129 Filterkoeffizienten verwendet. Abbildungen 15 bis 17 zeigen die entsprechenden Resultate:

**Abbildung 23: Originalsprachsignal****Abbildung 24: rekonstruiertes Sprachsignal****Abbildung 25: verzögertes Originalsprachsignal und rekonstruiertes Sprachsignal übereinandergelegt (Ausschnitt)**

Das verzögerte Originalsprachsignal und das rekonstruierte Signal überdecken sich nicht ganz. Abbildung 18 illustriert deren Differenz.

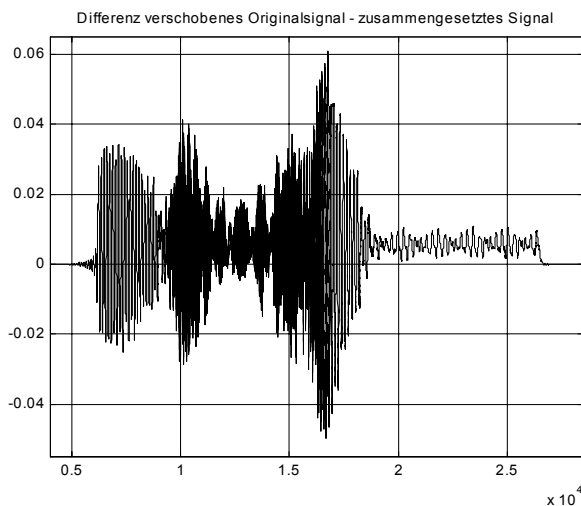


Abbildung 26: Differenz zwischen verschobenem Originalsignal und rekonstruiertem Signal

Dieses Differenzsignal wirkt sich als nicht gewollte Störung aus. Um abzuschätzen, wie stark sie das Originalsprachsignal beeinflusst, wird nun das Signal zu Rauschverhältnis berechnet. Dieses beträgt bei $N = 129$ Filterkoeffizienten:

$$SNR_{N=129} = 19.5dB$$

Als Vergleich dazu soll das SNR bei der Verwendung von FIR-Filtern mit $N = 257$ Koeffizienten bestimmt werden:

$$SNR_{N=257} = 19.9dB$$

Diese Resultate sind unbefriedigend für Sprachübertragung, denn dort ist mindestens ein SNR von 30 dB erforderlich. Für unsere Arbeit, wo es ja nur darum geht, ein Rauschsignal aus seinen Teilbändern zu rekonstruieren, genügt der SNR bei $N=129$ Filterkoeffizienten vollauf, weshalb wir auf eine Optimierung der Rekonstruktionsbank verzichten.

5.2.5 Rückwandlung der digitalen Signale in analoge

Um ein diskretes Signal in ein analoges umzuwandeln und anschliessend auf den Kopfhörer zu geben, gibt es in Matlab den Befehl `sound()`.

6 Untersuchungen mit der Sprachverständlichkeit

Ein wesentlicher Punkt unserer Arbeit ist der Vergleich zwischen den vom SII-Algorithmus berechneten Werten und dem subjektiven Verständlichkeitsempfinden von Personen mit normalem Gehör. Dafür wird nun als erstes der SII-Algorithmus gemäss der Drittel-Oktavbänder-Methode noch etwas genauer analysiert, damit dann anschliessend ein geeignetes Testverfahren entwickelt und durchgeführt werden kann.

6.1 Analyse des SII-Algorithmus

Bei dieser Analyse wird von einem mit normaler Lautstärke gesprochenem Eingangssprachsignal ausgegangen. Als Störsignal wird ein Rauschen gewählt, dessen Spektrum (spectrum level of equivalent noise, N'_i in dB) dieselben Teilbandpegel aufweist wie das Spektrum des Sprachsignals (equivalent speech spectrum level, E' in dB). Die Pegel für die Hörschwelle (equivalent hearing threshold level, T'_i in dB) liegen bei Personen mit normalem Gehör für jedes Teilband bei 0 dB.

Als erstes wird nun das Selbstmaskierungsspektrum für das Sprachsignal (self-speech masking spectrum level, V_i in dB) ermittelt:

$$V_i = E'_i - 24 \quad (6.1)$$

Der grössere Wert zwischen V_i und N'_i bildet die Zwischengrösse B_i . Da in unseren Betrachtungen N'_i immer grösser als V_i ist, wird B_i immer durch N'_i bestimmt.

Als nächstes wird die Steigung pro Oktave der Maskierung (slope per octave of the spread of masking, C_i in dB) berechnet:

$$C_i = -80 + 0.6 \cdot [B_i + 10 \lg(f_i) - 6.353] \quad (6.2)$$

N'_i , B_i , C_i sowie die Mittenfrequenzen (center frequency of an SII band, f_i in Hz) bilden die Ausgangsgrössen für die Berechnung des Leistungsspektrums für die Maskierung (equivalent masking spectrum level, Z_i in dB):

$$Z_i = 10 \cdot \lg \left\{ 10^{0.1 \cdot N'_i} + \sum_{k=1}^{i-1} 10^{0.1 \cdot [B_k + 3.32 \cdot C_k \cdot \log\left(\frac{0.89 \cdot f_i}{f_k}\right)]} \right\} \quad (6.3)$$

Die Auswertung der Gleichung (6.3) mit Matlab ergibt für unsere Eingangsgrössen Werte von Z_i , welche sich nicht wesentlich von N'_i unterscheiden. Deshalb gilt:

$$Z_i \approx N'_i \quad (6.3a)$$

Als weiterer Schritt wird das Leistungsspektrum des gehörinternen Rauschens (equivalent internal noise spectrum level, X'_i in dB) aus dem Spektrum der Hörschwelle T'_i und aus dem Referenzspektrum für das gehörinterne Rauschen (reference internal noise spectrum level, X_i in dB) bestimmt:

$$X'_i = X_i + T'_i \quad (6.4)$$

Weil in unserem Fall die Hörschwelle bei 0 dB liegt, gilt für X'_i :

$$X'_i = X_i \quad (6.4a)$$

Der grössere Wert aus Z_i und X'_i bestimmt nun das Leistungsspektrum des gesamten Störsignals (equivalent disturbance spectrum level, D_i in dB). Für unseren Fall ist Z_i immer grösser als X'_i , somit gilt für D_i :

$$D_i = Z_i \quad (6.5)$$

Aus Gleichung (6.3a) folgt für die gewählten Eingangsgrössen sofort:

$$D_i \approx N'_i \quad (6.5a)$$

Bereits in Abschnitt 3.3.4 wurde der Hörbarkeitsfaktor (band audibility, A_i) eingeführt. Er besteht aus einer Zahl zwischen null und eins und ist ein Mass für den qualitativen Beitrag des jeweiligen Teilbandes an die Sprachverständlichkeit. Pro Teilband berechnet sich der Hörbarkeitsfaktor wie folgt:

$$A_i = L_i \cdot K_i \quad (6.6)$$

Bei L_i handelt es sich um den Verzerrungsfaktor pro Teilband (speech level distortion factor). Er berechnet sich zu:

$$L_i = \min \begin{cases} 1 - \frac{E'_i - U_i - 10}{160} \\ 1 \end{cases} \quad (6.7)$$

U_i : normierter Sprachpegel des i-ten Teilbandes bei mit normaler Lautstärke gesprochener Sprache (spectrum level of standard speech for normal effort), in dB

Bei Sprachsignalen mit normaler Lautstärke, weist der Verzerrungsfaktor stets einen Wert nahe eins auf. Somit wird das Verhalten des Hörbarkeitsfaktors vor allem durch die Hilfsgrösse K_i bestimmt, welche man wie folgt erhält:

$$K_i = \frac{E'_i - D_i + 15}{30} \quad (6.8)$$

Wobei K_i nur für Werte im Intervall $[0;1]$ definiert ist. Ergibt die Berechnung für K_i einen Wert ausserhalb dieses Intervalls, so ist bei negativen Werten $K_i = 0$ und bei Werten grösser als 1 $K_i = 1$ zu setzen.

Da $D_i \approx N'_i$ folgt:

$$K_i \approx \frac{E'_i - N'_i + 15}{30} \quad (6.8a)$$

Durch die Wahl der Eingangssignale mit $E'_i = N'_i$ gilt für K_i weiter:

$$K_i \approx \frac{15}{30} = 0.5 \quad (6.8b)$$

Wird das Störsignal gegenüber dem Sprachsignal um x dB gesenkt, so bestimmt dieses x das Signal zu Rauschverhältnis (SNR) und K_i kann in guter Näherung wie folgt berechnet werden:

$$K_i \approx \frac{x + 15}{30} \quad (6.8c)$$

Da sich K_i nur im Bereich von 0 bis 1 bewegen kann, ist schnell ersichtlich, dass x nur im Bereich von -15 bis 15 dB andere Resultate als 0 bzw. 1 liefert. Deshalb macht es Sinn, praktische Versuche auf diesen Bereich zu beschränken. Der lineare Zusammenhang von x und K_i ist ersichtlich. Die Frage, ob sich die Linearität auch im SII widerspiegelt ist schnell beantwortet.

Mit $A_i \approx K_i$ ergibt sich für den SII aus Gleichung (3.2):

$$SII \approx \begin{cases} 0 & , x < -15 \\ \frac{x+15}{30} \cdot \sum_{i=1}^{18} I_i & , -15 \leq x \leq 15 \\ 1 & , x > 15 \end{cases} \quad (6.9)$$

Wie schon in Gleichung (3.1) erklärt wurde, ergibt die Bandwichtigkeit I summiert über alle Teilbänder eins. Somit wird die Linearität des SII offensichtlich.

$$SII \approx \begin{cases} 0 & , x < -15 \\ \frac{x+15}{30} & , -15 \leq x \leq 15 \\ 1 & , x > 15 \end{cases} \quad (6.9a)$$

Abbildung 1 illustriert diesen linearen Zusammenhang:

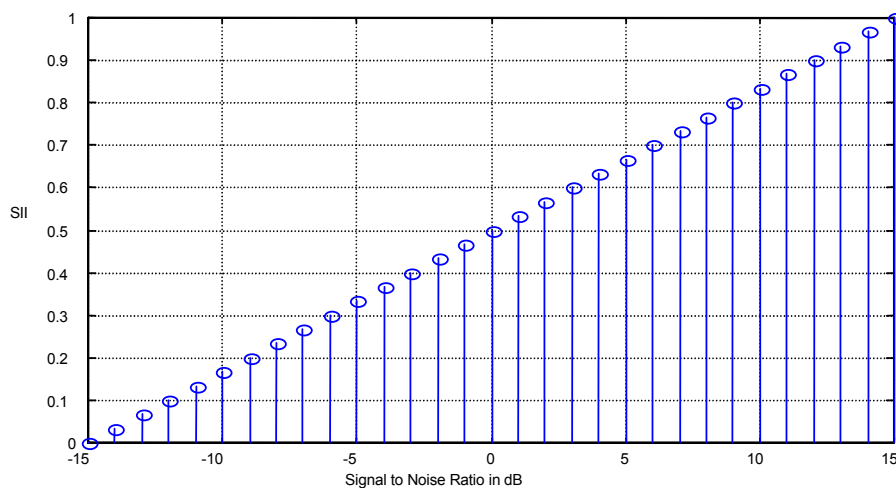


Abbildung 27: Linearer Zusammenhang für SII

Zusammenfassung

Durch die Angleichung der Teilbandpegel des Störsignals an das Sprachsignal, wird ein linearer Zusammenhang für den SII erreicht. Aus Abbildung 1 ist ersichtlich, dass der Inhalt eines Sprachsignals nicht mehr verständlich ist, wenn der SNR im Mittel -15 dB beträgt. Wenn hingegen der SNR im Mittel 15 dB oder mehr beträgt, so ist die volle Verständlichkeit erreicht. Im Bereich zwischen diesen zwei Extremen nimmt der SII bei zunehmendem SNR linear zu.

6.2 Test zur Erfassung des subjektiven Verständlichkeitsempfindens

Mit diesem Test untersuchten wir, inwiefern sich der vorhin beschriebene Zusammenhang mit dem subjektiven Verständlichkeitsempfinden von Personen mit normalem Gehör deckt. Dazu mussten wir zuerst ein geeignetes Testverfahren finden, welches so auf Testpersonen angewendet werden kann, dass diese dabei nicht mit Informationen überhäuft und somit überfordert werden. Ebenfalls soll es aber den Testpersonen auch nicht möglich sein, das Testergebnis zu verfälschen, indem sie nicht oder nur teilweise verstandene Informationen zufällig richtig erraten.

Es zeigte sich, dass ein Test mit Logatomen diese Bedingungen am besten erfüllt. Als Logatome bezeichnet man gemäss [7] sinnleere, in der Regel einsilbige Pseudowörter, deren Lautzusammen-

setzung den phonetischen Gesetzen der jeweiligen Sprache genügt. Für den Test wählten wir nun vier solche Logatome aus:

plol, chit, schlef und trunkst.

Diese Wörter wurden von Stefan Huber auf Band gesprochen und anschliessend mit einem wie in 5.2.3 beschriebenen Rauschsignal überlagert, wobei das Rauschsignal gegenüber dem Sprachsignal verschieden stark gewichtet wurde:

1. Rauschsignal 15 dB höher als Sprachsignal,
2. Rauschsignal 10 dB über dem Sprachsignal,
3. Rauschsignal 5 dB über dem Sprachsignal,
4. Rauschsignal gleich stark wie Sprachsignal,
5. Rauschsignal 5 dB tiefer als Sprachsignal,
6. Rauschsignal 10 dB unter dem Sprachsignal und
7. Rauschsignal 15 dB unter dem Sprachsignal.

Die Testpersonen hatten nun in einer Hörprobe die Aufgabe, das aufzuschreiben, was sie gehört hatten. Im weiteren mussten sie qualitative Aussagen machen darüber:

- wie belästigend sie das Hintergrundrauschen fanden,
- wie stark sie sich anstrengen mussten, um überhaupt etwas zu hören und
- welche allgemeinen Qualitätsaussagen sie zu dieser Hörprobe sonst noch machen können.

Die Hörprobe bestand aus insgesamt sieben Sequenzen, wobei für jede Sequenz das Rauschsignal gegenüber dem Sprachsignal anders gewichtet wurde. Zuerst wurde das Rauschen 15 dB stärker gewichtet als das Sprachsignal. Bei jeder späteren Sequenz reduzierten wir dann die Stärke des Rauschens gemäss dem oben aufgeführten Schema. Damit war es den Testpersonen unmöglich zu schummeln, denn sie konnten das Sprachsignal erst bei den Sequenzen richtig erkennen, bei welchen das Rauschsignal gleich stark oder schwächer war als das Sprachsignal. Die Sequenzen hatten dabei folgenden zeitlichen Ablauf:

- Zuerst war kurzzeitig nur Rauschen zu hören, damit sich das Ohr daran gewöhnen konnte,
- anschliessend wurde das Rauschen ergänzt durch das erste Logatom (plol),
- nach dem Logatom war noch für ganz kurze Zeit das Rauschen alleine zu hören, und
- dann folgte eine etwa fünfzehn Sekunden dauernde Pause.

Der Vorgang wiederholte sich nun bis die Testperson das vierte Logatom gehört hatte. Dann musste sie die entsprechenden qualitativen Aussagen notieren, wobei sie sich jeweils wie folgt entscheiden musste:

Aussagen über das Hintergrundrauschen

Unhörbar (=1), hörbar aber nicht belästigend (=2), leicht belästigend (=3), belästigend (=4) oder sehr belästigend (=5).

Höranstrengung

Völlig entspannt (=1), aufmerksam, aber ohne Anstrengung (=2), mässige Anstrengung (=3), beträchtliche Anstrengung (=4) oder unverständlich trotz grösster Anstrengung (=5).

Allgemeine Qualitätsaussagen

Ausgezeichnet (=1), gut (=2), ordentlich (=3), schlecht (=4) oder sehr schlecht (=5).

Anschliessend wurde das Rauschen auf die nächst tiefere Stufe reduziert und die Sequenz wiederholt. Abbildung 2 zeigt den zeitlichen Ablauf einer solchen Sequenz auch noch auf bildliche Art und Weise.

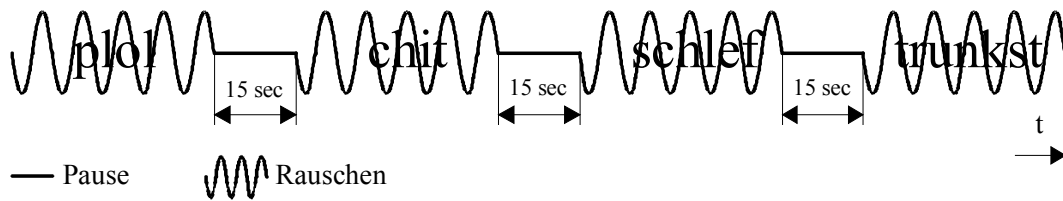


Abbildung 28: zeitlicher Ablauf der Hörprobe für jeweils eine Sequenz

In der Beilage 9.4.1 findet man das entsprechende Formular, welches die Versuchspersonen bei dieser Hörprobe auszufüllen hatten. Die Beilage 9.3.1 enthält das Matlab M-Script, welches für diesen Test geschrieben wurde.

Durchführung und Auswertung

Die Hörprobe führten wir nun mit zehn vorwiegend männlichen Testpersonen im Alter zwischen zwanzig und dreissig Jahren durch. Damit sie sich möglichst gut auf die entsprechenden Signale konzentrieren konnten, wurde die Hörprobe mit Kopfhörer durchgeführt.

Für die quantitative Auswertung des Testes sind wir wie folgt vorgegangen:

- Pro richtig verstanden Laut gab es einen Punkt; das gibt pro Sequenz maximal achtzehn Punkte, die erreicht werden konnten.
- Keine Punkte gab es für falsch oder nicht verstandene Laute.
- Wenn man bei "plol" "b" statt "p" verstanden hatte, gab es einen halben Punkt,
- bei "trunkst" hingegen war das erste "t" undeutlich gesprochen, weshalb es auch für ein "d" einen Punkt gab.
- Ebenfalls wurde kein Punktabzug gemacht, wenn jemand Laute mit Doppelbuchstaben geschrieben hatte; z.B. "ploll" statt "plol".
- "Ch" und "sch" gelten als ein Laut.

Pro Sequenz bildeten wir dann die durchschnittlich erreichte Punktzahl und deren Standardabweichung. Dies ergab folgende Resultate:

	Rauschen +15 dB	Rauschen +10 dB	Rauschen +5 dB	Rauschen 0 dB	Rauschen -5 dB	Rauschen -10 dB	Rauschen -15 dB
Durchschnittlich erreichte Punktzahl (max.18)	0.00	0.00	4.89	12.72	15.00	16.17	16.28

Standardabweichung	0.00	0.00	3.02	3.11	2.30	1.20	1.00
Durchschnittliche Verständlichkeit in Prozent	0.00 %	0.00 %	27.16 %	70.68 %	83.33 %	89.81 %	90.43 %

Tabelle 1: quantitative Auswertung der Hörproben

Tabelle 2 zeigt die Ergebnisse der qualitativen Auswertung:

		Rauschen +15 dB	Rauschen +10 dB	Rauschen +5 dB	Rauschen 0 dB	Rauschen -5 dB	Rauschen -10 dB	Rauschen -15 dB
Aussagen über das Hintergrundrauschen	Durchschnitt	5.00	4.78	4.56	3.78	3.22	1.94	1.22
	Standardabweichung	0.00	0.44	0.53	0.67	0.44	0.52	0.44
	korrespondierende Aussage	sehr belästigend	sehr belästigend	sehr belästigend	belästigend	leicht belästigend	hörbar, aber nicht belästigend	unhörbar
Höranstrengung	Durchschnitt	5.00	4.89	4.44	3.56	3.00	2.22	1.89
	Standardabweichung	0.00	0.33	0.53	0.53	0.87	0.67	0.60
	korrespondierende Aussage	unverständlich trotz grösster Anstrengung	unverständlich trotz grösster Anstrengung	beträchtliche Anstrengung	beträchtliche Anstrengung	mässige Anstrengung	aufmerksam, aber ohne Anstrengung	aufmerksam, aber ohne Anstrengung
Allgemeine Qualitätsaussagen	Durchschnitt	5.00	4.89	4.56	4.00	3.28	2.44	1.78
	Standardabweichung	0.00	0.33	0.53	0.50	0.67	0.88	0.83
	korrespondierende Aussage	sehr schlecht	sehr schlecht	sehr schlecht	schlecht	ordentlich	gut	gut

Tabelle 2: Auswertung der qualitativen Aussagen

Abbildung 3 illustriert den Zusammenhang zwischen dem subjektiven Verständlichkeitsempfinden und der Gewichtung des Rauschsignals. Im Gegensatz zu den obigen Tabellen wird diese Gewichtung indirekt durch die Gewichtung des Sprachsignals gegenüber dem Rauschsignal ausgedrückt, was dem Signal zu Rauschverhältnis (SNR) entspricht.

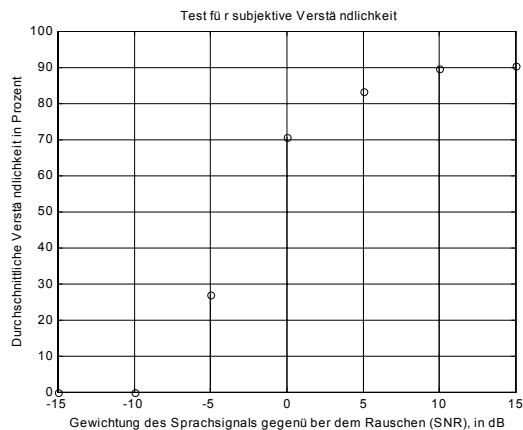


Abbildung 29: Quantitative Ergebnisse der Hörprobe

6.3 Vergleich SII mit subjektivem Verständlichkeitsempfinden

Ein erster Vergleich zwischen Abbildung 1 und Abbildung 3 deutet nicht unbedingt auf einen Zusammenhang zwischen SII und subjektivem Verständlichkeitsempfinden hin. Der Begriff Sprachverständlichkeitsindex ist sowieso etwas verfänglich, denn der SII ist ein Mass für die Anzahl Informationen aus dem Sprachsignal, welche zur Auswertung ins Gehirn gelangen, während das subjektive Verständlichkeitsempfinden auch die Auswertung durch das Gehirn umfasst. Ein genauerer Vergleich der beiden Grössen zeigt hingegen einen eindeutigen Zusammenhang:

- Bei einem SII nahe null gelangen keine Informationen zur Auswertung ins Gehirn und damit ist auch keine subjektive Verständlichkeit möglich.
- Je grösser das Signal zu Rauschverhältnis wird, desto mehr Informationen gelangen zur Auswertung ins Gehirn, womit auch eine Erhöhung der subjektiven Verständlichkeit stattfindet.
- Ab einem SNR von etwa 15 dB gelangen alle Informationen zur Auswertung ins Gehirn, weshalb auch die subjektive Verständlichkeit maximal wird.

Weshalb bei einem SNR von 15 dB aber nicht alle Laute richtig erkannt werden konnten, lässt sich wie folgt erklären:

1. Die Aussprache von Stefan Huber war den Testpersonen nicht sehr stark vertraut, und deshalb kannten sie deren Eigenheiten nicht.
2. Das Verwenden von Logatomen liess es nicht zu, dass die Testpersonen ein Wort, von dem sie nur einen Teil akustisch aufgenommen hatten, durch logisches Denken hätten richtig ergänzen können.
3. Ebenfalls führte das Verwenden von Logatomen unter den Versuchspersonen zu einer gewissen Verunsicherung, ob sie die ihnen unbekanntes Wörter wirklich richtig verstanden hatten.

Im weiteren ist zu beachten, dass die subjektive Verständlichkeit stark vom Inhalt des Sprachsignals abhängt und vor allem davon, wie vertraut dieser Inhalt dem jeweiligen Hörer ist.

7 Schlussgedanken zu dieser Studienarbeit

Mit dieser Semesterarbeit bekamen wir einen Einblick in einige Verfahren der digitalen Signalverarbeitung. Da für uns vieles neu war, haben wir die theoretischen Grundkenntnisse der digitalen Signalverarbeitung ausführlich dokumentiert. Die Neuerung, einen Zwischenbericht zu verfassen, zwang uns schon von Anfang an, unsere Erkenntnisse zu dokumentieren. Dies hat sich

eindeutig bewährt, denn so mussten wir uns schon früh mit der Thematik, wie man kurze und verständliche Berichte schreibt, beschäftigen. Da wir uns zuerst die theoretischen Kenntnisse aneigneten, war eine Aufgabenteilung von Anfang an nicht gut möglich. Auch der Umstand, dass das Labor erst nach Weihnachten bezugsbereit war, hatte uns gehindert, die für die Versuche benötigten Programme frühzeitig zu erstellen. So war es uns nicht mehr möglich noch weitere Tests als den mit den Logatomen durchzuführen. Dennoch reichte dieser Test aus, um die wesentlichsten Erkenntnisse aufzuzeigen.

Wir danken A. Schaub für seine Betreuung, und dass er stets mit hilfreichen Anregungen und Bemerkungen zur Seite stand.

Rapperswil, 25. Februar 1999

Markus Hurschler

Stefan Huber

8 Abkürzungen und Symbole

ANSI	:	American National Standard Institute
DSP	:	digitaler Signalprozessor
SII	:	Sprachverständlichkeitsindex (speech intelligibility index)
A_i	:	Hörbarkeitsfaktor pro Teilband (band audibility)
A_f	:	Anpassungsfaktor für Teilbandleistungen
b_k	:	k-ter Filterkoeffizient für FIR-Filter
b'_k	:	k-ter Filterkoeffizient für FIR-Filter mit Hammingfenster
C_i	:	Steigung pro Oktave der Maskierung (slope per octave of the spread of masking,) in dB
D_i	:	Pegel des gesamten Störsignals pro Teilband (spectrum level of equivalent speech), in dB
E'_i	:	Sprachpegel des i-ten Teilbandes (spectrum level of equivalent speech), in dB
$E_B(f_i)$:	Pegel über das ganze Teilband mit Mittenfrequenz f_i , in dB
$E_T(f_i)$:	Pegel ± 1 Hz um die Mittenfrequenz f_i , in dB
f_A	:	Abtastfrequenz, in Hz
f_i	:	Mittenfrequenz eines Teilbandes (center frequency of an SII band), in Hz
f_g	:	Grenzfrequenz eines Tiefpassfilters, in Hz
f_{Ny}	:	Nyquistfrequenz, in Hz
f_o	:	obere Grenzfrequenz eines Teilbandes, in Hz
f_u	:	untere Grenzfrequenz eines Teilbandes, in Hz
$H(f)$:	Übertragungsfunktion eines LTI-Systems
$h(t)$:	Impulsantwort eines LTI-Systems
h_i	:	diskrete Impulsantwort eines LTI-Systems
I_i	:	Bandwichtigkeit (band importance)
G_i	:	Verhältnis der Teilbandleistungen des Sprachsignals zum Störsignal
K_i	:	Hilfsgrösse zur Berechnung des SII
L_i	:	Verzerrungsfaktor pro Teilband i (speech level distortion factor)
M	:	Anzahl Verzögerungswerte, die ein FIR-Filter einfügt
N'_i	:	Spektrum des Eingangsstörsignals (equivalent noise spectrum level), in dB
P_E	:	Leistung des zurückgewonnenen quantisierten Signals, in W
P_Q	:	Leistung des Quantisierungsrauschens, in W
$r[n]$:	Rechtecksignal

SNR	:	Signal zu Rauschverhältnis (signal to noise ratio), in dB
SPL	:	normierter Gesamtpegel eines Sprachsignals normaler Lautstärke (overall speech level), in dB
s[n]	:	allgemeine Bezeichnung für diskretes Signal
T _i	:	Pegel pro Teilband für die Hörschwelle (equivalent hearing threshold level), in dB
U _i	:	normierter Sprachpegel des i-ten Teilbandes bei mit normaler Lautstärke gesprochener Sprache (spectrum level of standard speech for normal effort), in dB
V	:	Verzögerung
W _k	:	k-ter Koeffizient für Hammingfenster
X _i	:	Spektrum des gehörinternen Rauschens (equivalent noise spectrum level), in dB
X _i	:	Referenzspektrum für das gehörinterne Rauschen (reference internal noise spectrum level, in dB)
x[n]	:	diskretes Eingangssignal
y[n]	:	diskretes Ausgangssignal
Z _i	:	Spektrum für Maskierung (equivalent masking spectrum level), in dB
z	:	übliche Schreibweise für e ^{jω}
Δ _i	:	Differenz zur Berechnung des Teilbandpegels ± 1 Hz um eine Mittenfrequenz, wenn nur der Pegel des gesamten Teilbandes zur Verfügung steht (band with adj), in dB

9 Beilagen

9.1 SII

9.1.1 Algorithmus zur Berechnung des SII

(übersetzt aus [1])

Schritt 1 – Wahl der Berechnungsmethode

Drittel-Oktavbänder-Methode gemäss Aufgabenstellung

Schritt 2 – Leistungsspektren der Eingangssignale bei der Mittenfrequenzen

- 1) Leistungsspektrum des Sprachsignales (equivalent speech spectrum level), E' in dB
- 2) Leistungsspektrum des Störsignales (equivalent noise spectrum level), N' in dB
- 3) Leistungsspektrum der Hörschwelle (equivalent hearing threshold level), T' in dB

Bemerkung:

Wenn das Leistungsspektrum für 2) fehlt, ist per Default -50 dB einzusetzen.

Für 3) wird per Default 0 dB eingesetzt.

Schritt 3 – Leistungsspektrum für Maskierung

- 1) Leistungsspektrum für Selbstmaskierung des Sprachspektrums (self-speech masking spectrum level), V_i in dB

$$V_i = E'_i - 24$$

- 2) Hilfsgrösse B_i, wobei B_i gleich dem grösseren Wert zwischen V_i und N'_i entspricht.
- 3) Steigung pro Oktave der Maskierung (slope per octave of the spread of masking), C_i in dB

$$C_i = -80 + 0.6 \cdot [B_i + 10 \lg(f_i) - 6.353]$$

- 4) Leistungsspektrum für Maskierung (equivalent masking spectrum level), Z_i in dB

$$Z_i = 10 \cdot \lg \left\{ 10^{0.1 \cdot N'_i} + \sum_{k=1}^{i-1} 10^{0.1 \left[B_k + 3.32 \cdot C_k \cdot \log \left(\frac{0.89 \cdot f_i}{f_k} \right) \right]} \right\}$$

Schritt 4 – Leistungsspektrum des gehörinternen Rauschens, X'_i in dB
(equivalent internal noise spectrum level)

$$X'_i = X_i + T'_i$$

Schritt 5 – Leistungsspektrum des gesamten Störsignals, D_i in dB
(equivalent disturbance spectrum level)

D_i ist gleich dem grösseren Wert zwischen Z_i und X'_i .

Schritt 6 – Verzerrungsfaktor, L_i
(level distortion factor)

$$L_i = \begin{cases} 1 - \frac{E'_i - U_i - 10}{160} & , L_i \leq 1 \\ 1 & , \text{sonst} \end{cases}$$

Schritt 7 – Hörbarkeitsfaktor

1) Hilfsgrösse K_i

$$K_i = \begin{cases} 0 & , K_i < 0 \\ \frac{E'_i - D_i + 15}{30} & , 0 \leq K_i \leq 1 \\ 1 & , K_i > 1 \end{cases}$$

2) Hörbarkeitsfaktor (Band audibility), A_i

$$A_i = L_i \cdot K_i$$

Schritt 8 – Sprachverständlichkeitsindex, SII
(speech intelligibility index)

$$SII = \sum_{i=1}^{18} I_i \cdot A_i$$

9.1.2 Tabelle: Drittel-Oktavbänder-Methode

(aus [1] Table 3 Page 5)

Band no.	Frequency band			Standard speech spectrum level for stated vocal effort				Reference internal noise spectrum level	Free-field to eardrum transfer function
	Nominal midband freq	Band width adj.	Band importance	Normal	Raised	Loud	Shout		
i	f_i , Hz	Δ_i , dB	I_i	U_i , dB	dB	dB	dB	X_i , dB	H_i , dB
1	160	15.65	0.0083	32.41	33.81	35.29	30.77	0.60	0.00
2	200	16.65	0.0095	34.48	33.92	37.76	36.65	-1.70	0.50
3	250	17.65	0.0150	34.75	38.98	41.55	42.50	-3.90	1.00
4	315	18.65	0.0289	33.98	38.57	43.78	46.51	-6.10	1.40
5	400	19.65	0.0440	34.59	39.11	43.30	47.40	-8.20	1.50
6	500	20.65	0.0578	34.27	40.15	44.85	49.24	-9.70	1.80
7	630	21.65	0.0653	32.06	38.78	45.55	51.21	-10.80	2.40
8	800	22.65	0.0711	28.30	36.37	44.05	51.44	-11.90	3.10
9	1000	23.65	0.0818	25.01	33.86	42.16	51.31	-12.50	2.60
10	1250	24.65	0.0844	23.00	31.89	40.53	49.63	-13.50	3.00
11	1600	25.65	0.0882	20.15	28.58	37.70	47.65	-15.40	6.10
12	2000	26.65	0.0898	17.32	25.32	34.39	44.32	-17.70	12.00
13	2500	27.65	0.0868	13.18	22.35	30.98	40.80	-21.20	16.80
14	3150	28.65	0.0844	11.55	20.15	28.21	38.13	-24.20	15.00
15	4000	29.65	0.0771	9.33	16.78	25.41	34.41	-25.90	14.30
16	5000	30.65	0.0527	5.31	11.47	18.35	28.24	-23.60	10.70
17	6300	31.65	0.0364	2.59	7.67	13.87	23.45	-15.80	6.40
18	8000	32.65	0.0185	1.13	5.07	11.39	20.72	-7.10	1.80
Overall SPL, dB				62.35	68.34	74.85	82.30		

9.2 Matlab Funktionen

9.2.1 SII-Algorithmus

```

function [S,M]=SII(E,N,T)
%Syntax: [S,M]=SII(E,N,T)
%Die Funktion SII dient zur Berechnung des Sprachverständlichkeitsindexes
%SII (Speech Intelligibility Index)nach der Methode für Drittel-Oktavbaender
%gemaess ANSI Standard S3.5-1997
%
%Die Übergabeparameter sind Vektoren der Laenge 18 mit folgender Bedeutung:
%E Spectrum level of equivalent speech, in dB
%N Spectrum level of equivalent noise, in dB
%T Equivalent hearing threshold level, in dB
%
%Rückgabewerte:
%M : 13*18 Matrix mit folgendem Inhalt:
%Zeile 1: A Band audibility function
%Zeile 2: B Larger of the spectrum levels for equivqlent noise and self-speech masking, in dB
%Zeile 3: C Slope per octave (doubling of frequency) of the upward spread of masking, dB/octave
%Zeile 4: D Spectrum level for equivalent disturbance, in dB
%Zeile 5: F Center frequency of an SII band, in Hz
%Zeile 6: I Band importance function
%Zeile 7: K Temporary variable used in the calculation of the audibility function
%Zeile 8: L Speech level distortion factor, in dB
%Zeile 9: U Spectrum level of standard speech for normal vocal effort, in dB
%Zeile 10: V Spectrum level for self-speech masking, in dB
%Zeile 11: X Spectrum level of internal noise, in dB
%Zeile 12: X1 Spectrum level of equivalent internal noise, in dB
%Zeile 13: Z Spectrum level of equivqlent masking, in dB
%S ist der SII

%Feste Groessen genaess Tabelle 3 ANSI S3.5-1997
F=[160 200 250 315 400 500 630 800 1000 1250 1600 2000 2500 3150 4000 5000 6300 8000];
I=[0.0083 0.0095 0.015 0.0289 0.044 0.0578 0.0653 0.0711 0.0818 0.0844 0.0882 0.0898 0.0868 0.0844 0.0771 0.0527 0.0364 0.0185];
U=[32.41 34.48 34.75 33.98 34.59 34.27 32.06 28.3 25.01 23.00 20.15 17.32 13.18 11.55 9.33 5.31 2.59 1.13];
X=[0.6 -1.7 -3.9 -6.1 -8.2 -9.7 -10.80 -11.90 -12.5 -13.5 -15.4 -17.7 -21.20 -24.2 -25.9 -23.6 -15.8 -7.10];

%STEP 3
V=E-24;
B=Largerof(N,V);
C=-80+0.6.*(B+10.*log10(F)-6.353);
Z=B(1,1);
for i=2: length(E)
    Summe=0;
    for k=1 : i-1
        Summe=Summe+10^(0.1*(B(1,k)+3.32*C(1,k)*log10(0.89*F(1,i)/F(1,k))));
    end
    Z=[Z, 10*log10(10^(0.1*N(1,i))+Summe)];
end

%STEP 4
X1=X+T;

%STEP 5
D=Largerof(Z,X1);

%STEP 6
L=1-(E-U-10)./160;
L=Smallerof(L,1);

%STEP 7
K=(E-D+15)./30;
K=Smallerof(K,1);
Zero=[zeros(1,18)];
K=Largerof(K,Zero);
A=L.*K;

%STEP 8
S=sum(I.*A);

%Rueckgabematrix
M=[A;B;C;D;F;I;K;L;U;V;X;X1;Z];

```

Hilfsfunktion Smallerof()

```

function R=Smallerof(First, Second)
%Syntax: R=Smallerof(First, Second)
%Smallerof ermittelt zwischen einem Vektor (First)und einer Zahl (Second) welches

```

```
%i-te Element jeweils kleiner als die Zahl ist. Ist es kleiner, so
%wird es als i-tes Element in den Rückgabvektor R eingefügt, andernfalls wird
%die Zahl als i-tes Element von R eingefügt.
```

```
if length(Second)~=1
    disp('Achtung: Die Vergleichsgroesse ist ein Vektor und keine Zahl.');
```

```
Result=0;
else
    if First(1,1)>=Second
        Result=Second;

        else
            Result=First(1,1);
        end
    for i=2:length(First)
        if First(1,i)>=Second
            Smaller=Second;

            else
                Smaller=First(1,i);
            end
        Result=[Result, Smaller];
    end
end
R=Result;
```

Hilfsfunktion Largerof()

```
function R=Largerof(First, Second)
%Syntax: R=Largerof(First, Second)
%Largerof ermittelt zwischen zwei Vektoren (First, Second), welches
%i-te Element jeweils groesser ist. Das groessere Element
%wird dann als i-tes Element in den Rückgabvektor R abgespitzt.
%First und Second muessen zwei gleichlange Vektoren sein.
```

```
%Sind Eingangsvektoren gleichlang
```

```
if length(First)~=length(Second)
    disp('Achtung: Die Eingangsgrössen haben nicht dieselbe Länge.')
```

```
Result=0;
else
    if First(1,1)<=Second(1,1)
        Result=Second(1,1);
    else
        Result=First(1,1);
    end

    for i=2:length(First)
        if First(1,i)<=Second(1,i)
            Larger=Second(1,i);
        else
            Larger=First(1,i);
        end
        Result=[Result, Larger];
    end
end
R=Result;
```

9.2.2 Koeffizienten für FIR-Tiefpassfilter

```
function b=FIRTPkoeff(fg,fmax,N)
%Syntax: b=FIRTPkoeff(fg,fmax,N)
%FIRTPkoeff berechnet die Koeffizienten eines FIR-Tiefpasses
%mit Grenzfrequenz fg und Nyquistfrequenz fmax
%Übergabeparameter:
%
%fg : Grenzfrequenz des Tiefpasses
%fmax : Nyquistfrequenz
%N : Anzahl Koeffizienten im FIR-Tiefpass

%Rückgabewert:
%b : Koeffizienten b0..bk

omegag=fg/fmax;
b=omegag; %Koeffizient bei k=N/2

for k=1:N/2 %Koeffizienten von k>N/2
```

```

    b=[b,sin(k*omegag*pi)/k/pi];
end;

for k=1:N/2
    %Koeffizienten von k<N/2
    b=[sin(-k*omegag*pi)/-k/pi,b];
end;

%Einbau eines Hammingfensters
b=b.*hamming(length(b))';

```

9.2.3 Verdoppelung der Abtastrate

```

function y=Upsample2(x,N)
%Syntax: y=Upsample2(x,N)
%Funktion zur Verdoppelung der Abtastrate eines digitalen Eingangssignals
%Übergabeparameter:
%x: Eingangssignal zur Verdoppelung der Abtastrate
%N: Anzahl Koeffizienten für notwendiges Tiefpassfilter
%Rückgabewert:
%y: Ausgangssignal mit doppelter Abtastrate

x(2,:)=zeros(1,length(x));
x=reshape(x,1,2*length(x));
b=FIRTPKoeff(0.5,1,N);
x=filter(b,1,x);
y=2*x;

```

9.2.4 SII-Filterbank

```

function [Y1,Y2,Y3,Y4,Y5,Y6,B]=SIIFilterbank(x,N)
%Syntax: [Y1,Y2,Y3,Y4,Y5,Y6,B]=SIIFilterbank(x,N)
%SIIFilterbank berechnet die für die Berechnung des SII-Indexes
%gemäß der Drittelsoktavenbändern-Methode nötigen Teilbänder
%
%Übergabeparameter:
%x      : Eingangssignal
%N      : Anzahl Filterkoeffizienten FIR-Tiefpass
%
%Rückgabewerte:
%Y1     : Matrix mit den Signalen der obersten Oktave
%Y1(1,:) : Teilband um 6300 Hz
%Y1(2,:) : Teilband um 5000 Hz
%Y1(3,:) : Teilband um 4000 Hz
%Y2     : Matrix mit den Signalen der 2. obersten Oktave
%Y2(1,:) : Teilband um 3150 Hz
%Y2(2,:) : Teilband um 2500 Hz
%Y2(3,:) : Teilband um 2000 Hz
%Y3     : Matrix mit den Signalen der mittleren Oktave
%Y3(1,:) : Teilband um 1600 Hz
%Y3(2,:) : Teilband um 1250 Hz
%Y3(3,:) : Teilband um 1000 Hz
%Y4     : Matrix mit den Signalen der 3. untersten Oktave
%Y4(1,:) : Teilband um 800 Hz
%Y4(2,:) : Teilband um 630 Hz
%Y4(3,:) : Teilband um 500 Hz
%Y5     : Matrix mit den Signalen der 2. untersten Oktave
%Y5(1,:) : Teilband um 400 Hz
%Y5(2,:) : Teilband um 315 Hz
%Y5(3,:) : Teilband um 250 Hz
%Y6     : Matrix mit den Signalen der untersten Oktave
%Y6(1,:) : Teilband um 200 Hz
%Y6(2,:) : Teilband um 160 Hz
%B      : Matrix mit den Filterkoeffizienten für die Teilbänder
%B(1,:) : Teilband um 6300 Hz
%B(2,:) : Teilband um 5000 Hz
%B(3,:) : Teilband um 4000 Hz
%B(4,:) : Teilband um 3150 Hz
%B(5,:) : Teilband um 2500 Hz
%B(6,:) : Teilband um 2000 Hz
%B(7,:) : Teilband um 1600 Hz
%B(8,:) : Teilband um 1250 Hz
%B(9,:) : Teilband um 1000 Hz
%B(10,:) : Teilband um 800 Hz
%B(11,:) : Teilband um 630 Hz
%B(12,:) : Teilband um 500 Hz
%B(13,:) : Teilband um 400 Hz
%B(14,:) : Teilband um 315 Hz
%B(15,:) : Teilband um 250 Hz
%B(16,:) : Teilband um 200 Hz
%B(17,:) : Teilband um 160 Hz

%Eingangstiefpass

```

```

b=FIRTPkoeff(6300*2^(1/6),8000,N);
x=filter(b,1,x);
%oberste Oktave
A=DritOktBandFiltkoeff(4000,5000,6300,8000,N);
B=[b-A(1,:);A(1,:)-A(2,:);A(2,:)-A(3,:)];
for i=1:3
    Y1(i,:)=filter(B(i,:),1,x);
end
%Y1=[x-Y1(1,:);Y1(1,:)-Y1(2,:);Y1(2,:)-Y1(3,:)];
x=filter(A(3,:),1,x);

%Reduktion der Abtastrate
y=reshape(x,2,length(x)/2);
x=y(1,:);

%2. oberste Oktave
b=FIRTPkoeff(3150*2^(1/6),4000,N);
A=DritOktBandFiltkoeff(2000,2500,3150,4000,N);
B=[B;b-A(1,:);A(1,:)-A(2,:);A(2,:)-A(3,:)];
for i=1:3
    Y2(i,:)=filter(B(i+3,:),1,x);
end
%y=x;
%x=Y2(3,:);
%Y2=[y-Y2(1,:);Y2(1,:)-Y2(2,:);Y2(2,:)-Y2(3,:)];
x=filter(A(3,:),1,x);
%Reduktion der Abtastrate
y=reshape(x,2,length(x)/2);
x=y(1,:);

%mittlere Oktave
b=FIRTPkoeff(1600*2^(1/6),2000,N);
A=DritOktBandFiltkoeff(1000,1250,1600,2000,N);
B=[B;b-A(1,:);A(1,:)-A(2,:);A(2,:)-A(3,:)];
for i=1:3
    Y3(i,:)=filter(B(i+6,:),1,x);
end
x=filter(A(3,:),1,x);
%Reduktion der Abtastrate
y=reshape(x,2,length(x)/2);
x=y(1,:);

%3. unterste Oktave
b=FIRTPkoeff(800*2^(1/6),1000,N);
A=DritOktBandFiltkoeff(500,630,800,1000,N);
B=[B;b-A(1,:);A(1,:)-A(2,:);A(2,:)-A(3,:)];
for i=1:3
    Y4(i,:)=filter(B(i+9,:),1,x);
end
x=filter(A(3,:),1,x);
%Reduktion der Abtastrate
y=reshape(x,2,length(x)/2);
x=y(1,:);

%2. unterste Oktave
b=FIRTPkoeff(400*2^(1/6),500,N);
A=DritOktBandFiltkoeff(250,315,400,500,N);
B=[B;b-A(1,:);A(1,:)-A(2,:);A(2,:)-A(3,:)];
for i=1:3
    Y5(i,:)=filter(B(i+12,:),1,x);
end
x=filter(A(3,:),1,x);
%Reduktion der Abtastrate
y=reshape(x,2,length(x)/2);
x=y(1,:);

%unterste Oktave
b=FIRTPkoeff(200*2^(1/6),250,N);
A=DritOktBandFiltkoeff(125,160,200,250,N);
B=[B;b-A(1,:);A(1,:)-A(2,:);A(2,:)-A(3,:)];
for i=1:3
    Y6(i,:)=filter(B(i+15,:),1,x);
end

```

Hilfsfunktion Filterkoeffizienten pro Oktave

```

function B=DritOktBandFiltkoeff(fm3, fm2, fm1, Ny, N)
%Syntax: B=DritOktBandFiltkoeff(fm3, fm2, fm1, Ny, N)
%DritOktBandkoeff berechnet Filterkoeffizienten für Tiefpässe.
%
%Übergabeparameter:
%fm1      : Frequenz
%fm2      : Frequenz

```

```

%fm3      : Frequenz
%Ny       : Nyquistfrequenz
%N        : Anzahl Koeffizienten für FIR-Filter
%
%Rückgabewerte:
%B        : Matrix mit folgenden Zeileninhalten:
%B(1,:)   : Filterkoeffizienten für Tiefpässe mit fg=fm1*2^(-1/6)
%B(2,:)   : Filterkoeffizienten für Tiefpässe mit fg=fm2*2^(-1/6)
%B(3,:)   : Filterkoeffizienten für Tiefpässe mit fg=fm3*2^(-1/6)

b1=FIRTPkoeff(fm1*2^(-1/6),Ny,N);
b2=FIRTPkoeff(fm2*2^(-1/6),Ny,N);
b3=FIRTPkoeff(fm3*2^(-1/6),Ny,N);

%Rückgabewerte
B(1,:)=b1;
B(2,:)=b2;
B(3,:)=b3;

```

9.2.5 Anpassung der Signallänge an eine durch 32 teilbare Zahl

```

function neueSiglänge=neueSignallänge(Signal)
%Setzt einem Signal Nullen an, bis die neue Signallänge durch
%32 teilbar ist
length (Signal);

while rem(length(Signal),32)~=0
    Signal=[Signal,0];
end
length (Signal);

neueSiglänge=Signal;

```

9.2.6 Eichung der Teilbandsignalpegel auf die normierten Pegel des SII

```

function [dBPower,Power,SPL,faktor]=Signalleistungsanpassung(normierteLeistung)
%Syntax : [dBPower,Power]=Signalleistungsanpassung(normierteLeistung)
%Signalleistungsanpassung passt die normierte Leistung an die von SII vor-
%geschlagene Overall SPL, in dB für normal gesprochenen Sprache 61.22dB
%Eingabe      : die Leistungen pro 1/30ktaventeilband (Ausgabe der SigLeistungproBand )
%Rückgabewerte :
%Power        : gibt die Leistung pro Teilband als Zeilenvektor pro Teilband aus
%dBPower      : gibt die Leistung in dB (10*log10(Power)als Zeilenvektor pro Teilband aus von
160-6300

%Summe der normierteLestungen pro 1/30ktaventeilband
Summe=sum(normierteLeistung);

%faktor von SII vorgeschlagenen Leistung und der normierten Leistung
faktor=1717744.0212/Summe;

%Anpassung und Ausgabe der Leistung (Power)
Power=faktor.*normierteLeistung;

%Ausgabe von dBPower
for i=1:17
    dBPowertemp(i,:)=10*log10(Power(1,i));
end
dBPowerwith=dBPowertemp';

SPL=10*log10(sum(Power));

%Spectrum level of equivalent speech dB
%E*(f) = EB(f) - (Bandwith adj(f)) ANSI S*.5-1997 Formel(3)

Bandwith=[31.65 30.65 29.65 28.65 27.65 26.65 25.65 24.65 23.65 22.65 21.65 20.65 19.65 18.65
17.65 16.65 15.65];
dBPowertemp=dBPowertemp-Bandwith;
dBPowert=fliplr(dBPowertemp);
dBPower=[dBPowert,0];

```

9.2.7 Erzeugung eines Störsignals für Testzwecke

```

function [y,stoer]=Stoeranpassung(x,N);

```

```

%Syntax : [y,stoer]=Stoeranpassung(x);
%
%Stoeranpass passt die einzelnen Teilbandleistungen eines weissen Rausschen
%den Leistungen der Teilbänder des Sprachsignales x an
%Übergabeparameter:
%x      : Sprachsignal
%N      : Anzahl Koeffizienten für FIR-Tiefpassfilter
%
%Rückgabewerte:
%y      : Sprachsignal wie Eingang
%stoer  : Störsignal mit Teilbandleistungen wie Sprachsignal

x=neueSignallänge(x);
%Zerlegung in Teilbänder
[X1,X2,X3,X4,X5,X6]=SIIFilterbank(x,N);
[dBPowerX,PowerX]=SigLeistungproBand(X1,X2,X3,X4,X5,X6);

y=x;

%weisses Rauschen
stoer1=randn(1,length(x));

%Zerlegung in Teilbänder
[S1,S2,S3,S4,S5,S6]=SIIFilterbank(stoer1,N);
[dBPowerS,PowerS]=SigLeistungproBand(S1,S2,S3,S4,S5,S6);

%Verstärkungsfaktor pro Band

%Anpassung pro Teilband
V= sqrt(PowerX./PowerS);
V=[V, 0];

for i=1:3
    S1(i,:)=S1(i,:)*V(1,i);
    S2(i,:)=S2(i,:)*V(1,i+3);
    S3(i,:)=S3(i,:)*V(1,i+6);
    S4(i,:)=S4(i,:)*V(1,i+9);
    S5(i,:)=S5(i,:)*V(1,i+12);
    S6(i,:)=S6(i,:)*V(1,i+15);
end
[dBPowerX,PowerX]=SigLeistungproBand(X1,X2,X3,X4,X5,X6);
[dBPowerS,PowerS]=SigLeistungproBand(S1,S2,S3,S4,S5,S6);
stoer=Rekonstbank(S1,S2,S3,S4,S5,S6,N);

```

Hilfsfunktion Signalleistung()

```

function [dBPower,Power]=Signalleistung(Signal)
%Syntax : [dBPower,Power]=Signalleistung(Signal)
%Signalleistung berechnet aus einem Signal die Leistung
%Eingabe      : Eingangssignal
%
%Rückgabewerte :
%Power       : gibt die normierte Leistung als nackte Zahl aus
%dBPower     : gibt die normierte Leistung in dB (10*log10(Power) aus

Signaltemp=Signal;
Power=(Signal*Signaltemp')/length(Signal);
dBPower=10*log10(Power);

```

Hilfsfunktion SigleistungproBand()

```

function [dBPowerB,PowerB]=SigLeistungproBand(Band1,Band2,Band3,Band4,Band5,Band6)
%Syntax : [dBPower,Power]=SigLeistungproBand(Band1,Band2,Band3,Band4,Band5,Band6)
%SigLeistungproBand berechnet aus einem Signal die Leistung für jedes 1/3 Oktavenband aus
%und gibt die 2 verschiedenen Resultate in je einem Zeilenvektor aus
%
%Eingabe      : Band1 = 1/3Band6300-4000
%              Band2 = 1/3Band3150-2000
%              Band3 = 1/3Band1600-1000
%              Band4 = 1/3Band 800-500
%              Band5 = 1/3Band 400-250
%              Band6 = 1/3Band 200-160
%Rückgabewerte :
%Power       : gibt die normierte Leistung für jedes 1/3 Oktavenband als Zeilenvektor aus
%dBPower     : gibt die normierte Leistung für jedes 1/3 Oktavenband in dB (10*log10(Power)
%als Zeilenvektor aus

%Berechnung für Band1
for i=1:3
    [dBPower(i,:),Power(i,.)]=Signalleistung(Band1(i,:));

```

```

end

dBPowerBtemp=dBPower';
PowerBtemp=Power';
%Berechnung für Band2
for i=1:3
    [dBPower(i,:),Power(i,.)]=Signalleistung(Band2(i,:));
end

dBPowerBtemp=[dBPowerBtemp,dBPower'];
PowerBtemp=[PowerBtemp,Power'];

%Berechnung für Band3
for i=1:3
    [dBPower(i,:),Power(i,.)]=Signalleistung(Band3(i,:));
end

dBPowerBtemp=[dBPowerBtemp,dBPower'];
PowerBtemp=[PowerBtemp,Power'];

%Berechnung für Band4
for i=1:3
    [dBPower(i,:),Power(i,.)]=Signalleistung(Band4(i,:));
end

dBPowerBtemp=[dBPowerBtemp,dBPower'];
PowerBtemp=[PowerBtemp,Power'];

%Berechnung für Band5
for i=1:3
    [dBPower(i,:),Power(i,.)]=Signalleistung(Band5(i,:));
end

dBPowerBtemp=[dBPowerBtemp,dBPower'];
PowerBtemp=[PowerBtemp,Power'];

%Berechnung für Band6
for i=1:2
    [dBPowerl(i,:),Powerl(i,.)]=Signalleistung(Band6(i,:));
end

dBPowerBtemp=[dBPowerBtemp,dBPowerl'];
PowerBtemp=[PowerBtemp,Powerl'];

%Rückgabe
dBPowerB=dBPowerBtemp;
PowerB=PowerBtemp;

```

9.2.8 Rekonstruktionsbank

```

function y=Rekonstbank(Y1,Y2,Y3,Y4,Y5,Y6,N)
%Syntax: y=Rekonstbank(Y1,Y2,Y3,Y4,Y5,Y6,N)
%Rekonstbank rekonstruiert ein Signal anhand von Teilbandsignalen
%gemäß der Drittelsoktavenbänder-Methode für die Berechnung des SII
%
%Übergabeparameter:
%
%Y1      : Matrix mit den Signalen der obersten Oktave
%Y1(1,:) : Teilband um 6300 Hz
%Y1(2,:) : Teilband um 5000 Hz
%Y1(3,:) : Teilband um 4000 Hz
%Y2      : Matrix mit den Signalen der 2. obersten Oktave
%Y2(1,:) : Teilband um 3150 Hz
%Y2(2,:) : Teilband um 2500 Hz
%Y2(3,:) : Teilband um 2000 Hz
%Y3      : Matrix mit den Signalen der mittleren Oktave
%Y3(1,:) : Teilband um 1600 Hz
%Y3(2,:) : Teilband um 1250 Hz
%Y3(3,:) : Teilband um 1000 Hz
%Y4      : Matrix mit den Signalen der 3. untersten Oktave
%Y4(1,:) : Teilband um 800 Hz
%Y4(2,:) : Teilband um 630 Hz
%Y4(3,:) : Teilband um 500 Hz
%Y5      : Matrix mit den Signalen der 2. untersten Oktave
%Y5(1,:) : Teilband um 400 Hz
%Y5(2,:) : Teilband um 315 Hz
%Y5(3,:) : Teilband um 250 Hz
%Y6      : Matrix mit den Signalen der untersten Oktave
%Y6(1,:) : Teilband um 200 Hz
%Y6(2,:) : Teilband um 160 Hz
%
%Rückgabewerte
%y       : rekonstruiertes Ausgangssignal

```

```

%unterste Oktave
y=Y6(1,:)+Y6(2,:);
y=Upsample2(y,N);

%2. unterste Oktave
bv=[zeros(1,3*(N-1)/2),1];           %Verzögerungsglied
v=Y5(1,:)+Y5(2,:)+Y5(3,:);
v=filter(bv,1,v);
y=y+v;
y=Upsample2(y,N);

%3. unterste Oktave
bv=[zeros(1,3*(N-1)/2),1];           %Verzögerungsglied
v=Y4(1,:)+Y4(2,:)+Y4(3,:);
v=filter(bv,1,v);
y=y+v;
y=Upsample2(y,N);

%mittlere Oktave
bv=[zeros(1,21*(N-1)/2),1];          %Verzögerungsglied
v=Y3(1,:)+Y3(2,:)+Y3(3,:);
v=filter(bv,1,v);
y=y+v;
y=Upsample2(y,N);

%2. oberste Oktave
bv=[zeros(1,45*(N-1)/2),1];          %Verzögerungsglied
v=Y2(1,:)+Y2(2,:)+Y2(3,:);
v=filter(bv,1,v);
y=y+v;
y=Upsample2(y,N);

%oberste Oktave
bv=[zeros(1,93*(N-1)/2),1];          %Verzögerungsglied
v=Y1(1,:)+Y1(2,:)+Y1(3,:);
v=filter(bv,1,v);
y=y+v;

```

9.3 Matlab M-Scripts

9.3.1 Test zur Erfassung des subjektiven Verständlichkeitsempfindens

```

n=input('Geben Sie die Anzahl der Filterkoeffizienten ein : ');

%Einzug der Testsignale und Verarbeitung

Signal1temp=wavread('plol.wav');
Signal1=resample(Signal1temp',16000,22050);
neueSiglänge=neueSignallänge(Signal1);
[Y1,Y2,Y3,Y4,Y5,Y6]=SIIFilterbank(neueSiglänge,n);
[dBPowerB,PowerBSignal1]=SigLeistungproBand(Y1,Y2,Y3,Y4,Y5,Y6);
[dBPowerSignal1,Power,SPL,faktor1]=Signalleistungsanpassung(PowerBSignal1);

Signal2temp=wavread('chit.wav');
Signal2=resample(Signal2temp',16000,22050);
neueSiglänge=neueSignallänge(Signal2);
[Y1,Y2,Y3,Y4,Y5,Y6]=SIIFilterbank(neueSiglänge,n);
[dBPowerB,PowerBSignal2]=SigLeistungproBand(Y1,Y2,Y3,Y4,Y5,Y6);
[dBPowerSignal2,Power,SPL,faktor2]=Signalleistungsanpassung(PowerBSignal2);

Signal3temp=wavread('schlef.wav');
Signal3=resample(Signal3temp',16000,22050);
neueSiglänge=neueSignallänge(Signal3);
[Y1,Y2,Y3,Y4,Y5,Y6]=SIIFilterbank(neueSiglänge,n);
[dBPowerB,PowerBSignal3]=SigLeistungproBand(Y1,Y2,Y3,Y4,Y5,Y6);
[dBPowerSignal3,Power,SPL,faktor3]=Signalleistungsanpassung(PowerBSignal3);

Signal4temp=wavread('trunkst.wav');
Signal4=resample(Signal4temp',16000,22050);
neueSiglänge=neueSignallänge(Signal4);
[Y1,Y2,Y3,Y4,Y5,Y6]=SIIFilterbank(neueSiglänge,n);
[dBPowerB,PowerBSignal4]=SigLeistungproBand(Y1,Y2,Y3,Y4,Y5,Y6);
[dBPowerSignal4,Power,SPL,faktor4]=Signalleistungsanpassung(PowerBSignal4);

%Generieren von Rauschen
TestSignal1temp=wavread('Beispieltext.wav');

```

```

BezugsSprachsignal=resample(TestSignal1temp',16000,22050);
[Eingang,Rauschen]=Stoeranpassung(BezugsSprachsignal,n);

%Test mit den vier Testsignalen

%Maskierung des Testsprachsignals mit dem Rauschsignal
differenz=input('Um wieviel dB soll das Rauschsignal angehoben oder gesenkt werden bei Eingabe
0.07 Versuch Ende: ');

%zeros(1,70000) entspricht einer Ruhepause von etwa 5 Sekunden
Testsignal1=[zeros(1,50000),Signal1,zeros(1,2000)];
Testsignal2=[zeros(1,50000),Signal2,zeros(1,2000)];
Testsignal3=[zeros(1,50000),Signal3,zeros(1,2000)];
Testsignal4=[zeros(1,50000),Signal4,zeros(1,2000)];

Z=[zeros(1,18)];

%BezugsSprachsignal
neueSiglänge=neueSignallänge(BezugsSprachsignal);
[Y1,Y2,Y3,Y4,Y5,Y6]=SIIFilterbank(neueSiglänge,n);
[dB,PowerBand]=SigLeistungproBand(Y1,Y2,Y3,Y4,Y5,Y6);

%Rauschen in dB bezogen auf Signal1
[NoisedBtemp1,Power,SPL]=Rauschleistungsanpassung(PowerBand,faktor1);
[NoisedBtemp2,Power,SPL]=Rauschleistungsanpassung(PowerBand,faktor2);
[NoisedBtemp3,Power,SPL]=Rauschleistungsanpassung(PowerBand,faktor3);
[NoisedBtemp4,Power,SPL]=Rauschleistungsanpassung(PowerBand,faktor4);

while differenz ~= 0.07

    disp('-----')

    %Beeinflussung analoges Signal
    Noise=10^(differenz/10).*Rauschen;

    %Beeinflussung dB Noise
    NoisedB1=differenz+NoisedBtemp1;
    NoisedB2=differenz+NoisedBtemp2;
    NoisedB3=differenz+NoisedBtemp3;
    NoisedB4=differenz+NoisedBtemp4;

    Signal1Noise=Testsignal1+Noise(1,1:length(Testsignal1));
    Signal2Noise=Testsignal2+Noise(1,1:length(Testsignal2));
    Signal3Noise=Testsignal3+Noise(1,1:length(Testsignal3));
    Signal4Noise=Testsignal4+Noise(1,1:length(Testsignal4));

    sound(Signal1Noise,16000);
    pause(15)
    %disp('pol')
    %SIIIndex=SII(dBPowerSignal1,NoisedB1,Z)

    disp('-----')

    sound(Signal2Noise,16000);
    pause(15)
    %disp('chit')
    %SIIIndex=SII(dBPowerSignal2,NoisedB2,Z)

    disp('-----')

    sound(Signal3Noise,16000);
    pause(15)
    %disp('schlef')
    %SIIIndex=SII(dBPowerSignal3,NoisedB3,Z)

    disp('-----')

    sound(Signal4Noise,16000);
    pause(15)
    %disp('trunkst')
    %SIIIndex=SII(dBPowerSignal4,NoisedB4,Z)

    disp('-----')

    differenz =input('Um wieviel dB soll das Rauschsignal angehoben oder gesenkt werden bei
Eingabe 0.07 Versuch Ende: ');
end

```

9.4 Formulare

9.4.1 Test zur Erfassung des subjektiven Verständlichkeitsempfindens

Subjektive Verständlichkeit Datum: _____ Name: _____ Vorname: _____

Maskierung [dB]	Rauschen	Verstandene Wörter			
+15					
+10					
+5					
0					
-5					
-10					
-15					

Aussagen über das Hintergrundrauschen

dB	unhörbar	hörbar, aber nicht belästigend	leicht belästigend	belästigend	sehr belästigend
+15					
+10					
+5					
0					
-5					
-10					
-15					

Höranstrengung

dB	völlig entspannt	Aufmerksam, aber ohne Anstrengung	mässige Anstrengung	beträchtliche Anstrengung	unverständlich trotz grösster Anstrengung
+15					
+10					
+5					
0					
-5					
-10					
-15					

Allgemeine Qualitätsaussagen

dB	ausgezeichnet	gut	ordentlich	schlecht	sehr schlecht
+15					
+10					
+5					
0					
-5					
-10					
-15					

X : Trifft für alle vier Wörter zu

10 Literatur

- [1] ANSI S.3.5-1997: Methods for Calculation of the Speech Intelligibility Index, American National Standard Institute, 1997.
- [2] A. Schaub: Skript zur Einführung in die digitale Signalverarbeitung für Informatiker, HSR, 1998.
- [3] A. Schüeli: Skript für Vertiefungsfach digitale Signalverarbeitung, HSR Abteilung E, 1998 .
- [4] W.M. van den Enden, A.M. Verhoeckx: Digitale Siganlverarbeitung, Vieweg, 1990.
- [5] Josef Schilter: Skript für Fach Nachrichtentechnik im zweiten Studienjahr, HSR Abteilung E, 1998
- [6] Matlab Reference Guide, The MathWorks Inc, Reprint, April 1995.
- [7] Peter Vary, Ulrich Heute und Wolfgang Hess: Digitale Sprachsignalverarbeitung, Teubner, 1998.

