TECHNISCHE UNIVERSITÄT
CHEMNITZ

Erhöhung der Sprachqualität von CASA separierten Sprachsignalen

Bachelorarbeit

zur
Erlangung des akademischen Grades
B.Sc.

Fakultät für Informatik
Professur Technische Informatik

Eingereicht von: Jonas van der Smissen
Matrikel Nr.: 436768
Einreichungsdatum: 8. November 2021

Betreuer: René Schmidt
Prof. Dr. W. Hardt

Abstract

Ziel dieser Arbeit war es unter Einsatz einer Auswahl an Algorithmen die Sprachqualität von CASA separierten Sprachsignalen zu verbessern. Der Ursprung dieses Problems liegt in der von CASA verwendeten Zeit-Frequenz-Maskierung. Diese kann zu einer fehlerbehafteten Ausgabe führen, wenn zur gleichen Zeit zwei Sprecher den selben Frequenzbereich mit ihren Stimmen abdecken. Der beschriebene Bereich kann folglich nur als Ganzes einem Sprecher zugeordnet werden, sodass fehlerbehaftete Signale entstehen. Zur Verbesserung dieser wurde der Einsatz eines LPC-Vocoders, die Reduktion der Abtastrate der Signale sowie eine Kombination beider als Reparaturverfahren eingesetzt und überprüft.

Keywords: CASA, LPC, Abtastratenkonvertierung, Sprachqualität

Inhaltsverzeichnis

Abbildungsverzeichnis

Tabellenverzeichnis

Abkürzungsverzeichnis

ASA auditory scene analysis

BCH Bose-Chaudhuri-Hocquenghem

BSS blind signal separation

CASA computational asa

FOSS free and open source software

FFT fast fourier transformation

IFFT inverse fast fourier transformation

LPC linear predictive coding

RMSE root mean square error

RS Reed-Solomon

1. Einleitung

Bei der Untersuchung von Audioaufnahmen wird deutlich, dass in den meisten Fällen neben dem eigentlich aufgenommenen Signal weitere Störsignale aufgenommen wurden. Die Verknüpfung aller aufgenommenen Signale beschreibt eine auditive Szene. Betrachtet man beispielsweise eine Party auf welcher man selbst ein Gespräch führt. Trotz der Vielzahl an Störgeräuschen durch die Gespräche anderer Personen ist es im Normalfall möglich den Gesprächspartner zu verstehen. Dieser Cocktailparty-Effekt beschreibt die menschliche Fähigkeit des selektiven Hörens, welche es uns ermöglicht ein gezieltes Geräusch besser wahrzunehmen.

Die Realisierung dieses Verhaltens unter Einsatz eines Computers bildet die Grundlage der computergestützen auditiven Szenenanalyse, kurz CASA. Ziel eines CASA-Systems ist es ohne Vorwissen über die Entstehung einer auditiven Szene mit multiplen Sprechern, diese in Einzelsignale zu zerlegen, welche jeweils nur einen Sprecher beinhalten. Das Aufteilen der auditiven Szenen in einem CASA-System geschieht in mehreren Schritten. Das Auftrennen der Signale geschieht hierbei durch eine Zeit-Frequenz-Maskierung, diese wählt Signalabschnitte basierend auf Frequenz- und Zeitabschnitten aus. Treten jedoch Überlappungen der einzelnen Sprecher im Frequenzspektrum auf, so ist eine fehlerfreie Trennung mit diesem Verfahren nicht möglich. Zur Reparatur der beschädigten Audiosignale werden im Kontext dieser Arbeit bereits existierende Verfahren auf ihre Eignung als Reparaturverfahren geprüft.

Hierzu wird die Telefonie als praktisches Einsatzgebiet für das Verarbeiten und Übermitteln von Sprachdaten genauer betrachtet. In den Anfängen der Telefonie wurden die übermittelten Sprachdaten aufgrund der geringeren Kanalkapazität stark komprimiert. Die hierfür eingesetzten Komprimierungsalgorithmen kodierten das Signal, die Grundlage für viele der Kodierungsverfahren mit geringer Bitrate bildet Linear Predictive Coding (kurz LPC). Basierend auf diesem Verfahren soll zur Reparatur der beschädigten Audiosignale ein Kodierer eingesetzt werden, welcher das Signal zunächst komprimiert, sodass dieses anschließenden Schritt vollständig neu synthetisiert werden kann. Neben den Komprimierungsverfahren werden in der Telefonie weitere Algorithmen eingesetzt, welche die fehlerfreie Übertragung der komprimierten Sprachsignale ermöglichen sollen. Die Algorithmen der Kategorie der Kanalkodierung eröffnen ein weiteres umfangreiches Feld an Verfahren, welche potentiell für die Reparatur der beschädigten Signale eingesetzt werden können.

Neben der Auswahl möglicher Algorithmen und deren Implementierung wird in folgenden Kapiteln zudem eine Auswertung stattfinden. Diese dient zum Klären der Frage, ob die gewählten Algorithmen die Sprachqualität der beschädigten Signale verbessern und somit als Reparaturverfahren eingestuft werden können.

2. Stand der Technik

2.1. auditive Szenenanalyse

Die Basis der auditive Szenenanalyse (ASA) bildet hierbei das 1994 veröffentlichte gleichnamige Werk des Psychologen Albert S. Bregman. Dieses thematisiert die menschliche Fähigkeit der Unterscheidung und Gruppierung von Audiosignalen, so können einzelne Geräusche als ganzes oder separiert wahrgenommen werden. Diese Eigenschaft kann beispielsweise beim Spielen eines Akkordes auf einem Instrument beobachtet werden, hierbei kann dieser entweder als der Akkord selbst oder als die einzelnen Noten wahrgenommen werden. Während sich ein Akkord aus einer geringen Menge Frequenzen zusammensetzt, enthalten natürliche Geräusche ein vergleichsweise großes Frequenzspektrum. Das Entflechten von überlappenden Geräuschen dieser Kategorie stellt ein großes Problem dar, da das Auftrennen und falsche Gruppieren der Signale zum Hören von Geräuschen führt, welche nicht existent sind. Basierend darauf unterteilte Bregman das Themengebiet daher in die Aspekte der Segmentierung, Integration und Segregation. Segregation im Bezug auf Audiosignale bezieht sich hierbei auf die Fähigkeit des selektiven Hörens und wird auch als Cocktailparty-Effekt bezeichnet. Dieser Name ergibt sich aus der beispielhaften Szene einer Cocktailparty auf welcher es, trotz vieler anderer Gespräche und Hintergrundgeräusche, möglich ist seinen Gesprächspartner zu verstehen. [4]

2.2. computergestütze auditive Szenenanalyse

Die computergestütze auditive Szenenanalyse (CASA) umfasst das gesamte Themengebiet der auditive Szenenanalyse unter Einsatz computergestützter digitaler Datenverarbeitung. CASA-Systeme arbeiten mit dem Ziel der Segregation gemischter Audiosignale nach dem selben Verfahren wie es ein Mensch tun würde. Dazu wird ein Verfahren das der blind-signal-separation (BSS), bei welchem das Ziel der Segregation ohne Wissen über die Ausgangssignale oder den Vermischungsprozess erreicht werden soll, ähnlich ist eingesetzt. CASA-Systeme wurden auf Basis des menschlichen Hörvorgangs entwickelt, dadurch entsteht ein Unterschied zur herkömmlichen BSS, da somit maximal zwei Mikrofone zur Tonaufnahme verwendet werden.

1985 entstand unter Weintraub das erste CASA-System, dessen Aufgabe die Unterscheidung männlicher und weiblicher Stimmen war, dabei wurden die Perioden der einzelnen Frequenzkanäle durch Autokorrelation ermittelt. Ein häufig verwendeter Ansatz ist das „Bregman at face value“-System, welches 1992 von Brown beschrieben wurde. Es basiert auf der Forschung Bregmans und kann unterteilt werden die

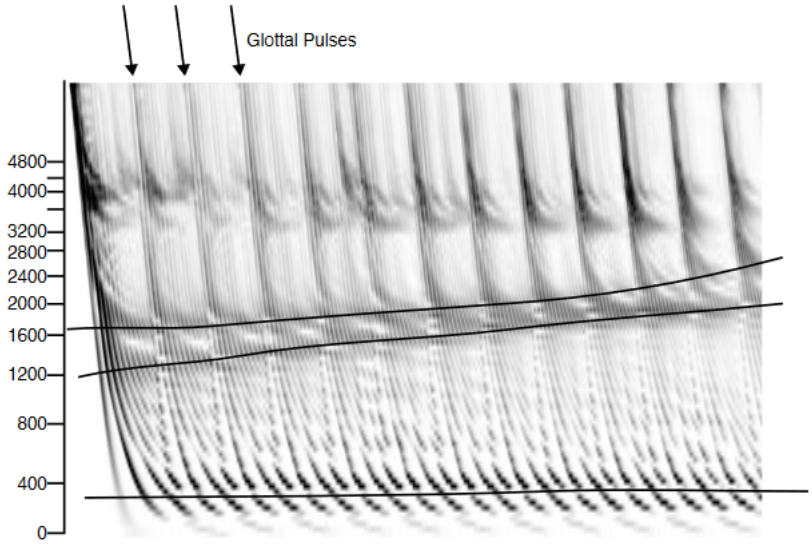

Abbildung 2.1.: Cochleagramm der amerikanischen Aussprache der Silbe „rea“. Die vertikalen Linien stellen die Stimmritzenpulse (engl. glottal pulses) dar, welche aufgrund der natürlichen Verzögerung der Cochlea geneigt sind.[28]

Segmentierung des Eingangssignals sowie anschließender Gruppierung der erhaltenen Einzelobjekte nach strikten Regeln. Mit diesem System ist es möglich gesprochene Worte herauszufiltern, jedoch wird hierzu ein periodisches Signal vorausgesetzt. Im Laufe der folgenden Jahre entstanden weitere CASA-System, welche entweder auf Basis des Störsignals oder auf einem durch Hypothesen veränderten Eingangssignales arbeiten. Ein modernes CASA-System kann in fünf Modelle, welche teilweise Aspekte des Hörvorgangs simulieren, unterteilt werden: das Cochlea-, Korrelo- und Kreuzkorrelogramm sowie die Zeit-Frequenz-Masken und Resynthese.[33]

2.2.1. Cochleagramm

Das Cochleagramm wurde nach der Cochlea, im deutschen Hörschnecke, benannt. Diese ist Bestandteil des Innenohrs und bildet das Rezeptorfeld der auditiven Wahrnehmung. Umgeben von Knochenmaterial besitzt jede Cochlea eine Anzahl an Windungen, welche direkt proportional mit dem Hörvermögen verknüpft ist. Beispielsweise besitzt die gesunde menschliche Cochlea zweieinhalb Windungen, bei angeborener Schwerhörigkeit ist diese Zahl auf bis eineinhalb Windungen reduziert. Die Umwandlung von Schall zu einem Nervenimpuls geschieht durch vier Reihen Haarzellen, welche in äußere und innere Haarzellen unterteilt werden können. Die äußeren drei Reihen beinhalten die äußeren Haarzellen und agieren als Sensor und Motor wodurch die registrierten Schallsignale verstärkt werden. In Zusammenarbeit mit der Basilarmembran, welche ebenfalls Teil der Cochlea ist, ergibt sich eine große Frequenzselektivität, welche für die Unterscheidung einzelner Geräusche nötig ist. Die

durch die Basilarmembran entstandenen mechanischen Schwingungen werden durch die vierte innere Reihe Haarzellen in Nervenimpulse umgewandelt und an das Gehirn weitergeleitet.

Für die Erstellung eines Cochleagrammes ist es notwendig all diese Aspekte zu simulieren. Die Frequenzselektivität der Basilarmembran wird durch eine Filterbank realisiert, welche durch eine Reihe von Bandfiltern realisiert wird. Jeder Filter stellt hierbei einen bestimmten Punkt der Membran dar und selektiert eine bestimmte Frequenz. Für die Realisierung der, im realen durch die Haarzellen entstehenden, Spitzen ist es möglich eine Impulsantwortfunktion, zum Beispiel in Form eines Gammatonfilters einzusetzen. In den meisten Fällen wird darauf jedoch verzichtet und der Fokus auf die Übertragung der, durch die inneren Haarzellen angeregten, Nerven gelegt. Dabei wird für jede Einzelfrequenz nur die erste Hälfte der Welle betrachtet und die hintere genullt. Durch anschließendes ziehen der Wurzel entsteht eine dem Verschiebungsmodell der Haarzellen ähnelnde Wellenform. Ein weiteres Modell ist das Meddis-IHC-Modell, welches die inneren Haarzellen auf Grundlage der durch die Basilarmembran freigegebenen Neurotransmitter simuliert.

Werden die erstellten Einzelfrequenzen im Anschluss in einem Zeit-Frequenz-Diagramm zusammengeführt, ergibt sich das Cochleagramm. Das so erhaltene Diagramme stellt hierbei eine vollständige Repräsentation des aufgenommen Geräusches dar. Ein Beispiel für ein solches Diagramm ist in Abbildung 2.1 für die Aussprache der Silbe „rea“ zu finden.[33]

2.2.2. Korrelogramm

Erstmals 1951 von James Licklider als Modell für die auditive Wahrnehmung vorgeschlagen ist das Korrelogramm ein wichtiger Bestandteil jedes CASA-Systems. Das Diagramm entsteht durch die graphische Darstellung der Abhängigkeiten statistischer Daten über einen bestimmten Zeitraum. Unter Berechnung diverser statistischer Werte ist es außerdem möglich festzustellen, ob vorliegende Daten zufälliger Natur sind oder beispielsweise auf einer Sinusfunktion basieren. Weiterhin bietet das Diagramm eine anschaulichere Darstellung als das Cochleagramm, da die in diesem enthaltene Redundanz reduziert wird. Diese Redundanz ist im Beispiel in Abbildung 2.1 gut sichtbar und in Form der sich durch die Stimmritzenpulse getrennten wiederholenden Segmente im Diagramm zu erkennen.[33]

Im Allgemeinen findet für die Erstellung des Korrelogrammes eine Autokorrelation der simulierten Nervensignale mit den Daten der einzelnen Filterkanäle der Filterbank statt. Durch Gruppieren dieser Korrelationen nach deren Frequenzen kann ein weiteres Diagramm erstellt werden, bei welchem die entstehenden Spitzen den wahrgenommenen Tonhöhen entsprechen.[22] 1990 wurde zudem von Assman und weiteren Wissenschaftlern belegt, dass das Korrelogramm ebenfalls hilfreich bei der Modellierung von gleichzeitig gesprochenen Vokalen sein kann.

2.2.3. Kreuzkorrelogramm

Das Kreuzkorrelogramm ist die Darstellung der Kreuzkorrelationsfunktion, welche die Korrelation zweier Eingangssignale mit unterschiedlicher zeitlicher Verschiebung beschreibt. Diese Funktion findet häufig Anwendung in den Gebieten der Mustererkennung oder Kryptoanalyse, wobei sich der Einsatz allgemein auf die Suche eines kurzen bekannten Signales innerhalb eines längeren Signales beschränken lässt. Ein binaurales CASA-System besteht aus zwei räumlich getrennten Mikrofonen, welche die Ohren simulieren. Durch eben diese räumliche Trennung erreicht ein Audiosignal beide Mikrofone zu unterschiedlichen Zeiten. Aus der aufgenommenen Verzögerung ist es zunächst möglich die Ursprungsposition des Signals zu berechnen. Trotz dieser zeitlichen Verzögerung ist es durch Kreuzkorrelation der beiden Mikrofonkanäle möglich, das ursprünglichen Eingangssignal wiederherzustellen. Es tritt in Folge der Kreuzkorrelation in Form der übereinstimmenden Spitzen der erhaltenen Funktion auf.[33, 14]

2.2.4. Zeit-Frequenz-Maskierung

Der Einsatz einer Maskierung des Cochleagrammes in Abhängigkeit von Zeit und Frequenz entsteht in Szenarien, bei welchen ein gesuchtes Audiosignal durch ein lauteres Geräusch überdeckt wird. In Folge dessen ist das gesuchte Signal unverständlich und muss durch Filtern der überdeckenden Geräusche wiederhergestellt werden. Allgemein geschieht dies durch Wichtung des gesuchten Signals sowie Unterdrückung des Restsignals. Lange Zeit wurden zu diesem Zweck statistische Filter, wie beispielsweise der Wiener-Filter, eingesetzt. Durch Messung der minimalen mittleren quadratischen Abweichung ist es mit diesem möglich Rauschen optimal zu Unterdrücken. Es wird angenommen, dass das gesuchte Signal $s(t)$ durch ein additives Rauschen $n(t)$ gestört wird. Durch Faltung des so erhaltenen Eingangssignales ist es dann möglich den Erwartungswert des quadratischen Fehlers zu berechnen, unter der Voraussetzung das für $s(t)$ und $n(t)$ die Autokorrelationen sowie Kreuzkorrelation bekannt sind. Diese Voraussetzung ist jedoch immer erfüllt, da die benötigten Korrelationen in den vorherigen Schritten bei der Erstellung des Korrelo- und Kreuzkorrelogrammes berechnet wurden.[33]

Aktuelle Forschungen entfernen sich zunehmend von der statistischen Filterung und arbeiten an einer Filterung durch den Einsatz künstlicher Intelligenz. Dies ergab sich durch das Problem herkömmlicher Algorithmen, welche für starkes Rauschen schlechtere Ergebnisse produzieren, da durch das strikte Filtern ebenfalls Segmente des gesuchten Signals verloren gehen. Unter Verwendung eines Deep Learning Algorithmus ist es 2019 gelungen selbst bei sehr starkem Rauschen die gesprochenen Texte zu filtern und die entstandenen Lücken zu füllen. [15]

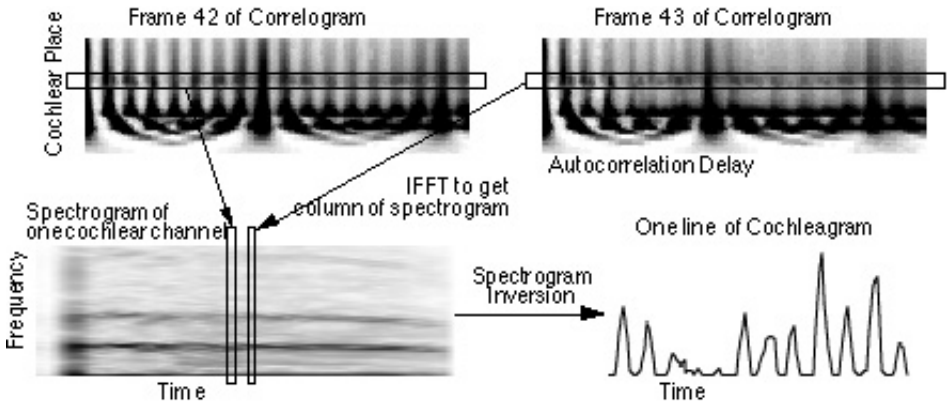

Abbildung 2.2.: Invertierung des Korrelogrammes durch anwenden der inversen schnellen Fourier-Transformation (IFFT) für jede Reihe sowie anschließende Invertierung der entstehenden Spektrogramme.[29]

2.2.5. Resynthese

Die Resynthese bildet den abschließenden Schritt, hierbei werden die aufgetrennten Geräusche so aufbereitet, das diese für den Menschen hörbar werden. Die Resynthese kann in zwei Teilschritte unterteilt werden, zunächst muss das Korrelogramm und im Anschluss das Cochleagramm invertiert werden.

Die im ersten Schritt geplante Invertierung des Korrelogrammes soll das ursprüngliche Cochleagramm bereitstellen, welches im Anschluss invertiert wird. Jede Reihe des Korrelogrammes repräsentiert eine zeitabhängige Autokorrelation, welche wie in Abbildung 2.2 gezeigt mittels einer inversen schnellen Fourier-Transformation in ein einzelnes Spektrogramm umgewandelt werden kann. Die so aus allen Reihen erhaltenen Spektrogramme werden durch iteratives Prüfen, welche Signalwelle mit dem Spektrogramm übereinstimmt, invertiert. Die so bestimmten Signale werden aufeinander abgestimmt, wodurch das ursprüngliche Geräusch in Form des Cochleagrammes aufgebaut werden kann. Die hier beschriebene Invertierung des Korrelogrammes erzeugt ein dem ursprünglichen Geräusch sehr ähnliches Ergebnisse, da das Korrelogramm ein vollständiges Informationsmodell des Geräusches darstellt.

Für die Invertierung des Cochleagrammes muss zunächst die Komprimierung der ursprünglichen Lautstärke rückgängig gemacht werden. Im Anschluss werden die Werte der simulierten Nervensignale rückläufig durch die jeweiligen Filterkanäle gesendet und anschließend aufsummiert. Dieser Vorgang wird bis das Diagramm vollständig abgearbeitet ist wiederholt. Handelt es sich um ein ideales Cochleagramm so sind die Unterschiede zwischen Ursprungssignal und synthetisiertem Signal nicht wahrnehmbar.[33]

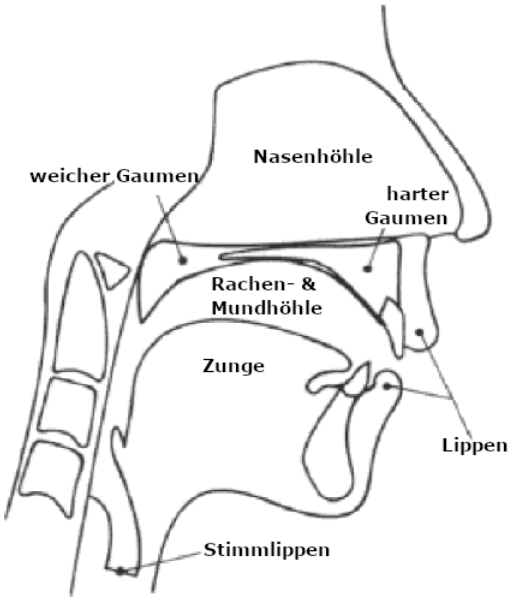

Abbildung 2.3.: Querschnittskizze des für die Sprachbildung relevanten Teils des Sprachapparates vergl. [19].

2.3. Linear Predictive Coding

Linear Predictive Coding (LPC) ist ein mathematisches Modell zur Verarbeitung von Sprachsignalen, welches häufig zur Analyse und Synthese von Sprache eingesetzt wird. Das grundlegende Modell der linearen kleinsten Quadrate wurde bereits 1795 von Carl Friedrich Gauss beschrieben, jedoch fand das Verfahren erstmals 1966 im japanischen Telekommunikationsunternehmen Nippon Telegraph and Telephone Anwendung in der Sprachverarbeitung. Unabhängig von den dort forschenden Wissenschaftlern Itakura und Saito wurde das Verfahren 1967 von Atal und Schroeder in der amerikanischen Forschungseinrichtung Bell Labs ebenfalls zur Verarbeitung von Sprachsignalen eingesetzt. In den folgenden Jahren wurde das Verfahren ständig weiterentwickelt, sodass 1971 eine erste Hardwarerealisierung zur Echtzeitverarbeitung von Signalen sowie 1978 der erste LPC-Algorithmus mit variable Bitrate entstand. LPC bildet bis heute die Grundlage für den Mobilfunkstandard GSM und findet, teils in veränderter Form, Einsatz als verlustbehaftetes Kompressionsverfahren für Sprachsignale, beispielsweise für das Audiodateiformat mp3.[9]

2.3.1. Physikalische Grundlagen

Die Grundlage des LPC-Modells bildet der Sprachapparat des Menschen (siehe 2.3), welcher die aus der Lunge strömende Luft so in Schwingung versetzt, dass aus dem Mund Laute bzw. gesprochene Sprache kommt. Hierbei verändern verschiedene Teile des Sprachapparates unterschiedliche Aspekte der entstehenden Sprache. Die Men-

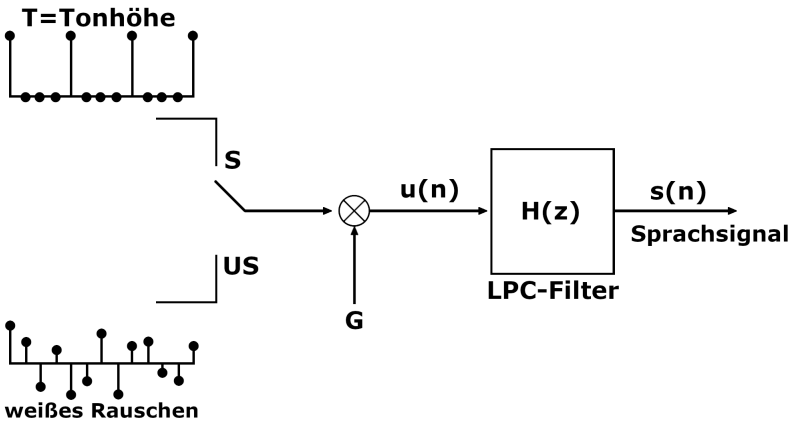

Abbildung 2.4.: LPC-Modell zur Synthese von Sprachsignalen vergl. [26].

ge der aus der Lunge strömenden Luft bestimmt die Lautstärke. Die ausströmende Luft wird durch das Öffnen und Schließen der Stimmfalten, auch Stimmbänder genannt, in Schwingung versetzt. Für stimmhafte Töne vibrieren diese und erzeugen je nach Frequenz unterschiedliche Tonhöhen. Frequenz und Tonhöhe sind hierbei proportional voneinander abhängig, sodass hohe Frequenzen zu hohen Tönen und niedrige Frequenzen zu tiefen Tönen führen. Für Reib- oder Explosivlaute bleiben die Stimmfalten konstant geöffnet bzw. geschlossen. Die nun Schwingung versetzte Luft wird durch die Form des aus Rachen-, Mund- und Nasenhöhle bestehenden Vokaltraktes weiter moduliert. Während die Nasenhöhle ein festes Volumen ausweist, können Rachen- und Mundhöhle durch Bewegungen der Zunge variiert werden. Dieser variable Resonanzraum erzeugt nun die finalen Töne. Während des Sprechens findet die Veränderung des Vokaltraktes alle 10 bis 100 Millisekunden statt. [25, 8]

2.3.2. LPC Analyse und Synthese

Alle zuvor genannten Aspekte, welche Einfluss auf das bilden von Sprach haben, finden sich auch im LPC-Modell (siehe Abbildung 2.4) wieder. Das Ausgangssignal wird hierbei durch die Art des Signals das synthetisiert werden soll bestimmt. Für stimmhafte Töne S wird ein Sinussignal verwendet, dessen Frequenz T mit der Schwingungsfrequenz der Stimmbänder korreliert und somit direkt die Tonhöhe beeinflusst. Für unstimmhafte Töne wie Reib- und Explosivlaute wird als Ausgangssignal ein Signal mit vielen Frequenzen gleicher Intensität, sogenanntes weißes Rauschen verwendet. Die im physikalischen Modell durch die Luftmenge bestimmte Lautstärke wird im Modell durch einen Faktor G umgesetzt, welcher die Amplituden des Signals verstärkt oder vermindert. Der im Anschluss angewandte LPC-Filter modelliert den Vokaltrakt des Menschen und verändert das Eingabesignal. Der LPC-Filter ist gegeben durch die Gleichung

$$H(z) = \frac{1}{1 + a_1 z^{-1} + a_2 z^{-2} + \ldots + a_p z^{-p}}$$

. Es ergibt sich folglich

$$s(n)+\sum_{i=1}^{p} a_i s(n-i) = u(n) \tag{2.1}$$

für die Relation zwischen Eingabe und Ausgabe des Filters. Das LPC-Modell kann somit für jedes Frame als Vektor der Form

$$A = (a_1, a_2, \ldots, a_p, G, S|US, T)$$

dargestellt werden. Dieser Wert von A ändert sich nach einem festen Zeitintervall, dieses wird durch die für das Modell verwendete Framegröße bestimmt. Für eine Framegröße von 20ms und einer Abtastrate von 8000Hz ergibt sich

$$\frac{1\,Frame}{20ms} = 50\frac{Frames}{s}$$

$$\frac{8000Hz}{50\frac{Frames}{s}} = \frac{8000\frac{Samples}{s}}{50\frac{Frames}{s}} = 160\frac{Samples}{Frame}$$

. Folglich können jeweils 160 Samples durch einen Vektor A repräsentiert werden. Für die Synthese des Sprachsignales wird, unter Verwendung der Filtergleichung 2.1, aus einem gegebenen Vektor A die entsprechende Anzahl an Samples generiert.

Die als LPC Analyse bezeichnete Umkehrung dieser Synthese berechnet aus einer gegebenen Anzahl Samples den bestmöglichen Vektor A. Aus den gegebenen Samples $S = (s(0), s(1), ..., s(N))$ wird unter Verwendung von Gleichung 2.1 das Grundsignal $u(n)$ berechnet. Für das Aufstellen eines linearen Gleichungssystems setzt man

$$\begin{aligned} f_{a_1} &= \tfrac{\partial f}{\partial a_1} = 0 \\ f_{a_2} &= \tfrac{\partial f}{\partial a_2} = 0 \\ &\ldots \\ f_{a_p} &= \tfrac{\partial f}{\partial a_p} = 0 \end{aligned}$$

mit

$$f = \sum_{i=0}^{N} u^2(i)$$

. Unter Verwendung der Autokorelation von $s(n)$

$$R(k) = \sum_{n=0}^{N-k} (s(n) * s(n+k)) \tag{2.2}$$

ergibt sich eine symmetrische Matrix im aufgestellten Gleichungssystem

$$\begin{bmatrix} R(0) & R(1) & R(2) & \cdots & R(p-1) \\ R(1) & R(0) & R(1) & \cdots & R(p-2) \\ R(2) & R(1) & R(0) & \cdots & R(p-3) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ R(p-1) & R(p-2) & R(p-3) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_p \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ R(3) \\ \vdots \\ R(p) \end{bmatrix}$$

. Durch anschließendes Lösen der Matrixgleichung mit einem beliebigen Algorithmus werden direkt die Parameter $a_1, ..., a_p$ bestimmt. Durch das Betrachten der Autokorelation von $u(n)$ wird nun die Entscheidung zwischen stimmhaften und unstimmhaften Grundsignal getroffen sowie die Werte der Lautstärke G und Tonhöhe T festgelegt. [8, 25]

Der Aufbau eines simplen LPC Vocoders setzt sich zusammen aus Analyse gefolgt von anschließender Synthese des Signals. Dies ermöglicht es das durch die Analyse komprimierte Signal über einen Kanal zu übertragen. Die dabei benötigte Übertragungsrate ist abhängig von der gewählten Framegröße und der Anzahl der Filterkoeffizienten $a_1, ..., a_p$. Für die historisch eingesetzte Konstellation mit 10 Koeffizienten und einer Framegröße von 20ms ergibt sich eine Übertragungsrate von 2,4 kb/s. Diese ist im Vergleich zur, für das Original mit einer Abtastrate von 8000Hz, benötigten Übertragungsrate von 64 kb/s sehr gering. Folglich können unter Verwendung eines Vocoders für einen gegebenen Kanal mit fester Kanalkapazität gleichzeitig mehr Sprachsignale übertragen werden als ohne, wodurch dieses Modell großen Einsatz in der Telefonie fand.[5]

2.4. Kanalkodierung

Die Geschichte der Kanalkodierung beginnt 1948 mit der damals revolutionären Pionierarbeit des US-amerikanischen Mathematikers und Elektrotechnikers Claude E. Shannon zur Informationstheorie. Die Veröffentlichung seiner Arbeit gilt als Geburtsstunde der Kanalkodierung und bewies auf theoretischer Basis das Vorhandensein zuverlässiger Alternativen zu den damals verwendeten Wiederholungscodes. Diese bilden die einfachsten fehlerkorrigierende Kanalcodes und kodieren jede zu übertragende Nachricht durch n-malige Wiederholung jedes einzelnen Symbols. Die Dekodierung der so entstandenen Nachrichten basiert auf einer Mehrheitsentscheidung für jedes empfangene Symbol. Liegt der Fehler unter 50% können die empfangenen Symbole korrekt dekodiert werden. Dieser Kanalcode hat keinen Codegewinn und ist für praktische Zwecke nur unter Verwendung weiterer Kodierungen sinnvoll. In Shannons Arbeit bewies er, dass jeder Kanal eine Kanalkapazität besitzt, welche die zuverlässige Informationsrate limitiert. Als zuverlässig wurden hierbei alle Informationen definiert, welche einen festgelegten Symbolfehlerwert nicht überschritten. Im von ihm formulierten Kanalkodierungstheorem beschrieb er außerdem das Kanalcodes existieren, welche beliebig nahe an die Kanalkapazität heran kommen. Die von Shannon beschriebenen Kanalcodes sind unendlich und zufällig und sind somit für eine praktische Verwendung ungeeignet. Weiterhin beinhaltet seine rein theoretische Arbeit keine Erklärung für die Konstruktion oder der effizienten Dekodierung der dort beschriebenen Kanalcodes.[27]

Erst in den folgenden Jahren wurden, auf Basis dieser theoretischen Grundlage, praktisch verwendbare Kanalcodes entwickelt. Hierbei sind sowohl der 1949 von Marcel J. E. Golay entwickelte Golay-Code als auch der 1950 von Richard W. Hamming entwickelte Hamming-Code bedeutend. Unter dem Oberbegriff Golay-Code werden

der binäre sowie der ternäre Golay-Code zusammengefasst. Der binäre Golay-Code ist als quadratischer Rest-Code der Länge 23 definiert und kann bis zu drei Fehler korrigieren. Im Vergleich dazu ist der ternäre Code definiert mit einer quadratischen Restlänge von 11 und kann somit nur zwei Fehler korrigieren. Beide Golay-Codes zählen zur kleinen Menge der perfekten fehlerkorrigierenden Codes. Dies bedeutet, dass jedem empfangenen Wort durch finden des geringsten Hamming-Abstandes eindeutig ein Codewort zugeordnet werden kann. Folglich sind perfekte Codes eindeutig dekodierbar.[7, 18]

Weitere perfekte fehlerkorrigierende Codes sind der ungerade binäre Wiederholungscode sowie der Hamming-Code für endliche Körper. Der von Richard W. Hamming entwickelte Code entstand 1950 aus dem Problem der Fehlerbehaftung des Relay-Computers seiner Arbeitsstelle. Durch häufiges Einlesen der Lochkarten wurde diese nach einiger Zeit nicht mehr richtig erkannt und die dadurch auftretenden Fehler mussten per Hand nachkorrigiert werden. Um diesem Mehraufwand ein Ende zu setzen entstand der Hamming-Code, ein System das unter Verwendung von Paritätsbits einen Fehler pro Wort korrigieren kann. Die Werte der Paritätsbits entstehen hierbei durch Verketten der Nutzdatenbits nach einem festen Schema und sind somit redundant. Diese Kontrollbits werden im Codewort an den Positionen platziert, welche einer zweier Potenz entsprechen. Die entstehenden Lücken zwischen den Kontrollbits werden in geordneter Reihenfolge mit den Nutzdatenbits gefüllt. Das so entstehende Codewort ist in Abbildung 2.5 dargestellt. Um nun die Werte der Paritätsbits konkret zu ermitteln wird beispielsweise für das erste Kontrollbit, beginnend mit dem ersten Datenbit, jedes zweite Datenbit miteinander verknüpft. Für das zweite Kontrollbit wird der Wert des Nutzdatenbits, welches sich im Codewort rechts des Kontrollbits befindet, eingerechnet. Im Anschluss werden zwei Datenbits übersprungen und die darauf folgenden zwei Datenbits einbezogen, dies wird bis zum Ende des Codewortes wiederholt. Das dritte Kontrollbit, beginnt ebenfalls mit dem Nutzdatenbit rechts von diesem, bezieht jedoch direkt die folgenden drei Datenbits ein, überspringt dann vier weitere und bezieht im Anschluss die folgenden vier Bits wieder in die Rechnung ein. Das System kann für beliebige weitere Kontrollbits analog weitergeführt werden. Es wird deutlich, dass nicht jedes Datenbit in jedes Kontrollbit einfließt, manche Datenbits jedoch auch mehrfachen Einfluss in verschiedene Kontrollbits haben können. Die durch diesen einfachen Hamming-Code entstandenen Codewörter haben untereinander den Hamming-Abstand drei. Wird ein Codewort nun übertragen, kann somit durch Prüfen dieses Abstandes erkannt werden zu welchem Codewort es zugeordnet werden soll. Aus dem festen Abstand von drei zwischen allen Codewörtern ergibt sich ebenfalls die Korrekturleistung, durch welche das Codewort selbst mit einem Bitfehler noch richtig zuordnen kann. Treten jedoch zwei Bitfehler auf, wird das empfangene Wort dem falschen Codewort zugeordnet, es findet dann eine sogenannte Falschkorrektur statt.[10]

Vier Jahre später entwickelten Irving S. Reed und David E. Muller die Reed-Muller-Codes welche, wie die Golay-Codes, eine Familie von linearen, fehlerkorrigierenden Codes beschreiben. Im Vergleich zu den meisten vorherigen Verfahren sind Reed-Muller-Codes, aufgrund ihrer effizienten Konstruktion und hohen Feh-

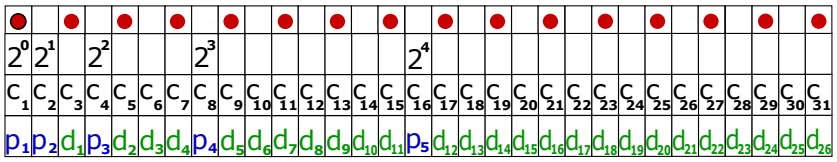

Abbildung 2.5.: Anordnung der Paritätsbits p_n sowie der Datenbits d_n innerhalb eines durch den Hamming-Code erstellten Codewortes

lerkorrektur, gut für große Blocklängen geeignet. Für das Senden einer Nachricht n wird diese, wie auch in vorherigen Verfahren üblich, zunächst in ein Codewort c übersetzt. Dieses Codewort kann nun durch eine Abbildung f aus der Menge $V = \{f\,Abbildung \mid f : \mathbb{F}_2^m \to \mathbb{F}_2\}$ beschrieben werden. Die so erhaltene Abbildung besitzt einen Bildvektor $(f(0), f(1), ..., f(2^m - 1))$, welcher die zur Fehlerkorrektur benötigte Redundanz besitzt. Dieser Bildvektor wird nun über den gestörten Kanal übertragen, wodurch der übertragenen Bildvektor mit einem Fehlervektor e verknüpft wird. Für die Dekodierung des empfangenen Vektors ist es nötig die Koordinatenfunktionen Z_i der Menge V zu bestimmen. Durch bilden des Skalarproduktes zwischen empfangenen Vektor und den Koordinatenfunktionen ist es möglich die Monomkoeffizienten und somit die ursprüngliche Abbildung $f = c$ zu ermitteln, solange der Fehler e nicht zu groß ist. Aus dem so berechneten Codewort kann dann auf die Nachricht n geschlossen werden.[30] Durch die hohe Fehlerkorrekturrate und effiziente Dekodierung der Reed-Muller-Codes wurden diese zu einem geeigneten Kandidaten für die Datenübertragung der Mariner 9 Mission. Hierbei kam der Hadamard-Code, eine Unterklasse der Reed-Muller-Codes, im Bereich der Fehlerkorrektur bei der Bildübertragung zum Einsatz. Die mit 6 Bit codierten Grauwerte der Bildpixel wurden durch 32 Bit lange Codewörter kodiert, welche einen Hamming-Abstand von mindestens 16 besaßen. Vergleicht man den verwendeten Hadamard-Code mit einem 5-Wiederholungscode fällt auf das die Informationsrate ähnlich, jedoch die Anzahl der korrigierbaren Fehler deutlich größer ist. Während ein 5-Wiederholungscode maximal zwei Bitfehler richtig korrigieren kann, können mit dem Hadamard-Code bis zu sieben Bitfehler pro Wort korrigiert werden. Unter Anwendung einer schnellen Fourier-Transformation war es ebenfalls möglich die empfangenen Daten effizient zu Dekodieren.[12]

Bereits 1960 entstand ein weiterer Meilenstein aus der Zusammenarbeit von Reed und Gustave Solomon. Die Reed-Solomon-Codes gehören zur Oberklasse der BCH-Codes und bilden die ersten zyklischen, fehlerkorrigierenden Codes. Im Unterschied zu vorhergehenden Verfahren sind die Codesymbole Teil eines endlichen Körpers und nicht einzelne Bits. Ein Beispiel für die Übertragung von Text ist hierbei die Verwendung des ASCII-Standards als endlicher Körper für die übertragenen Zeichen. Die zur Fehlerkorrektur benötigte Redundanz der Codes entsteht, da die Zahlenwerte der Nachricht durch ein Polynom des Grades k beschrieben werden. Diese festgelegten Stützstellen, welche die Nachricht kodieren, werden im Anschluss durch extrapolieren des Polynoms mit redundanten Stützstellen ergänzt. Werden bei der

Übertragung einige der n Stützstellen ausgelöscht, ist es durch Interpolation möglich das Polynom zu berechnen. Hierbei kann eine Nachricht mit $(n-k)$ Auslöschungen oder $\frac{(n-k)}{2}$ Fehlern sicher rekonstruiert werden, sodass die Codes ausgezeichnete Fehlerkorrektureigenschaften besitzen. Der Einsatz der Codes war jedoch zunächst nicht sinnvoll möglich, da erst 1969 ein effizienter Dekodieralgorithmus durch Elwyn Berlekamp und James Massey vorgestellt wurde.[2, 20] In Folge dessen kamen RS-Codes sowohl in wissenschaftlichen als auch in kommerziellen Gebieten zum Einsatz und finden bis heute in CDs oder dem DVB-Standard für digitales fernsehen Anwendung.

Ein weiteres Beispiel für Kanalcodes welche erst durch einen, viele Jahre später entwickelten, Dekodierungsalgorithmus verwendbar wurden sind Faltungscodes. Diese wurden bereits 1955 von Peter Elias für Datenströme ohne feste Blocklänge beschrieben. Die praktische Anwendung war jedoch erst 1967 mit dem Viterbi-Algorithmus möglich, welcher das effiziente Dekodieren der Codes ermöglichte. Dieser entstand während der Erforschung der Fehlerwahrscheinlichkeiten der Faltungscodes und ermöglicht eine Soft-Input-Dekodierung. Der Algorithmus dekodiert hierbei basierend auf den Wahrscheinlichkeiten der einzelnen Symbole anstatt auf Basis der Bitwerte. Die Fehlerkorrekturrate des Algorithmus ist durch das Dekodierungsverfahren jedoch begrenzt, da diese die Dekodierkomplexität exponentiell erhöht.[32] Einsatz findet dieser Algorithmus beispielsweise als Leseschutz für Festplatten sowie in Mobilfunk- und Satellitenübertragungen, bei welchen die Symbolwahrscheinlichkeiten bekannt sind.

1966 zeigte der US-amerikanische Informationstheoretiker Dave Forney, das durch serielle Verkettung der Codes immer bessere Codes entworfen werden konnten. Da sich die verketteten Codes durch verknüpfen der bisher bekannten Dekodierverfahren dekodieren ließen, entstand wenig Mehraufwand. Für die Verkettung wird zunächst ein äußerer Kanalcode, beispielsweise ein Reed-Solomon-Code, gewählt und anschließend durch einen inneren Faltungscode kodiert.[6] Wie bereits vorherige fehlerkorrigierende Codes fand auch dieses Verfahren Anwendung in der Raumfahrt, beispielsweise wurden 1977 serielle Codeverkettungen für beide Voyager-Raumsonden eingesetzt.

Erst zu Beginn der 1990er wurden seriell verkettete Codes durch eine neue Kodierungtheorie abgelöst. Im Vergleich zu allen bisherigen Codes, welche bisher immer mehrere Dezibel im Signal-Rausch-Verhältnis von der 1948 von Shannon definierten Kanalkapazität entfernt waren, ist es mit Turbo-Codes möglich diese Differenz auf weniger als ein Dezibel zu schließen. Der Kodiervorgang, wie in Abbildung 2.6 dargestellt, einer Nachricht findet hierbei in mindestens zwei seriell oder parallel geschalteten Kodierern statt bei welchen die Nutzdaten zunächst im ersten Kodierer mit einem Faltungscode kodiert werden. Anschließend werden die so erhaltenen kodierten Daten durch einen pseudo-zufälligen Interleaver umgeformt. Es folgt eine erneute Kodierung mittels Faltungscode im zweiten Kodierer, welche die finalen zu übertragenden Daten bereitstellt. Basierend auf der gewählten Anzahl der Kodierer wird zur Dekodierung der Nachricht die gleiche Anzahl an Dekodierern benötigt.

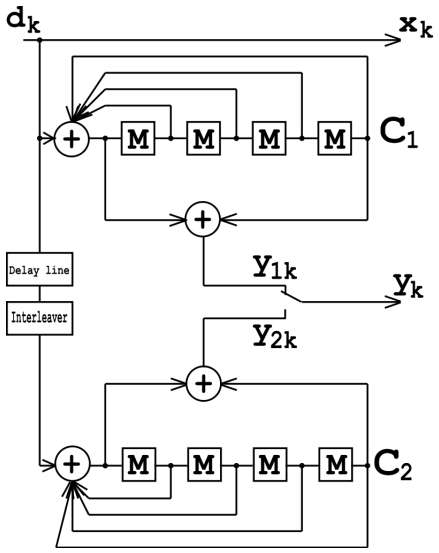

Abbildung 2.6.: Schematische Darstellung des Kodiervorgang eines Turbo-Encoders mit zwei parallelen Kodierern

Die ebenfalls seriell oder parallel geschalteten Dekodierer arbeiten iterativ und tauschen untereinander statistische Daten zur Fehlerkorrektur aus. Durch den in der Kodierung eingesetzten Interleaver sind die Codewörter voll verschränkt, wodurch der Dekodieraufwand linear zur Codewortlänge bleibt.[3] Diese einem Turbolader gleichende Funktionsweise der Dekodierung gab dem entwickelten Code den Namen Turbo-Code und ermöglicht unter geringem Aufwand eine sehr leistungsstarke Fehlerkorrektur. Gleich den Faltungscodes, auf welchen das Verfahren basiert, ist die Dekodierung des Turbo-Codes nur über Soft-Decision möglich. Da die Dekodierer nur statistische Informationen austauschen, können anschließend die einzelnen Symbole der Codewörter nur mit bestimmten Wahrscheinlichkeiten verarbeitet werden. Des Weiteren ermöglicht der lineare Dekodieraufwand der Turbo-Codes erstmals den Einsatz sehr langer Codewörter von mehreren Kilobit pro Wort, wodurch Turbocodes vor allem bei den Mobilfunkstandards UMTS und LTE eingesetzt werden.[31]

Die bereits Anfang der 1960er Jahre entwickelten LDPC-Codes wurden auf Grund der fehlenden technischen Möglichkeiten zur Implementierung der damaligen Zeit als nicht einsatzfähig klassifiziert. Als jedoch 1996 durch David J.C. MacKay bei diesen eine ähnlich gute Leistungsfähigkeit wie Turbo-Codes nachgewiesen wurde, entstand in den folgenden Jahren durch Richardson und Urbanke eine umfangreiche Theorie zur Konstruktion dieser.[17] LDPC-Codes können neben Turbo-Codes als Bestandteil des aktuellen Standes der Technik klassifiziert werden und finden im 5G-Standard sowie den WLAN-Standards 802.11n und 802.11ac Einsatz.

Ebenfalls Teil des Standes der Technik sind die 2008 von Erdal Arıkan eingeführten Polar Codes. Für diese ist es, erstmalig in der Geschichte der Kanalkodierung, gelungen nachzuweisen, dass für asymptotisch gegen unendlich strebende Blocklängen die von Shannon beschrieben Kanalkapazität erreicht werden kann.[1] Durch

Verknüpfung mit einer zyklisch redundanten Prüfsumme sind die dann als CRC-Polar-Codes bezeichneten Codes auch für kurze Blocklängen sehr gut geeignet. Einsatz finden diese CRC-Polar-Codes beispielsweise in den Steuerkanälen der aktuellen 5G-Mobilfunknetze.

3. Konzept

Ziel der Arbeit ist die Verbesserung der Sprachqualität der durch CASA beschädigten Sprachsignale, welche durch einen oder mehrere der, im vorherigen Kapitel, vorgestellten Algorithmen realisiert werden soll. Für eine ideale Rekonstruktion des Originalsignals wäre der Einsatz eines Kanalkodierungsalgorithmus mit perfekter Kodierung und großer Fehlertoleranz zweckmäßig, da mit diesem das Signal vollständig fehlerfrei rekonstruiert werden könnte. Für das vorliegende Problem sind diese jedoch nicht einsetzbar, da für das Korrigieren der Fehler Informationen über das Originalsignal in Form von Paritätsbits oder Ausgangswahrscheinlichkeiten der Symbole vorliegen müssen. Diese sind jedoch durch das Anwendungsgebiet von CASA, bei welchem ein gemischtes Signal ohne Vorkenntnisse der einzelnen Originalsignale zerlegt werden soll, vollständig unbekannt. Es ist somit nicht möglich einen der in Abschnitt 2.4 vorgestellten Kanalkodierungsalgorithmen einzusetzen. Ein weiterer Algorithmus, welcher im vorherigen Kapitel betrachtet wurde ist Linear Predictive Coding. Der Algorithmus ermöglicht die Komprimierung von Signalen und benötigt im Gegensatz zu den vorherigen Algorithmen keinerlei Informationen über das Originalsignal und fand ursprünglich Einsatz als Verfahren zur Komprimierung von Signalen in der Telefonie. Es sollten dabei Sprachsignale möglichst stark komprimiert werden um trotz der damaligen geringen Kanalkapazitäten eine hohe Anzahl an gleichzeitigen Signalübertragungen zu ermöglichen. Das eingehende Signal wird hierbei in Form eines Vektors bestehend aus Koeffizienten und Variablen zur Bestimmung des Signaltyps (vergleiche 2.3.2) kodiert. Unter Verwendung des LPC-Algorithmus wäre es also möglich das Signal zu analysieren und anschließend aus dem erhaltenen Vektor neu zu synthetisieren. Durch die Kodierung mit einer festgelegten Anzahl an Parametern findet eine Interpolation des Frequenzspektrums statt, wodurch fehlende Informationen repariert werden können. Im Folgenden werden einige Evaluationsparameter für den Einsatz des LPC-Vocoders ermittelt. Diese müssen so gewählt werden, dass der durch die Komprimierung mit LPC entstehende Qualitätsverlust minimiert und das bestmögliche Ergebnis erzielt wird.

3.1. Linear Predictive Coding

Der erste zu betrachtende Parameter dieser Kategorie bestimmt die Anzahl der LPC-Koeffizienten des Filters und muss auf die Abtastrate des Eingabesignales angepasst werden. Die Anzahl der Koeffizienten hat direkten Einfluss auf den Grad der Komprimierung durch die Analyse, dabei steigt der Kompressionsfaktor mit sinkender Anzahl an Koeffizienten. Die Bestimmung der Anzahl der Koeffizienten ist für eine

Ursprung des Wertes	Koeffizientenanzahl
Näherungsformel über Formanten	44
Speech Processing[21]	16
Web-Based-Vocoder [16]	20
Mohammadi[23]	18

Tabelle 3.1.: Vergleich der berechneten mit praktisch eingesetzten LPC-Koeffizientenanzahl

vorgegebene Abtastrate nicht eindeutig möglich, es existiert jedoch eine Formel zur Annäherung des Wertes. Für die Abtastrate der in dieser Arbeit verwendeten Eingabesignale von 44100 Hz ergibt sich eine Bandbreite von 22050 Hz. Des Weiteren wird angenommen, dass pro 1000 Hertz Bandbreite je ein Formant existiert, somit kann auf ungefähr 22 Formanten pro Signal geschlossen werden. Jeder Formant kann durch zwei Koeffizienten repräsentiert werden, sodass näherungsweise eine Anzahl von 44 Koeffizienten angenommen wird. Im Vergleich mit verschiedenen Implementation wird jedoch deutlich, dass dieser berechnete Wert deutlich oberhalb der dort eingesetzten Koeffizientenanzahl ist (siehe Tabelle 3.1). Die Bestimmung eines finalen Wertes wird durch systematisches Ausprobieren im Rahmen des theoretisch ermittelten und der praktisch eingesetzten Werte durchgeführt.

Ein weiterer Parameter der Analyse bestimmt die Größe der sich überlappenden Zeitfenster in welche das Eingabesignal zerlegt wird. Die Fenstergröße beeinflusst direkt die zeitliche Veränderung des Ausgabesignals. Dabei gilt, dass mit abnehmender Fenstergröße sich das Signal schneller und mit zunehmender Fenstergröße langsamer ändert. Zur Bestimmung des Wertes wird zunächst die Erzeugung der menschlichen Sprache betrachtet (siehe 2.3.1). Der Vokaltrakt verändert dabei alle zehn bis 100 Millisekunden seine Form und erzeugt somit einen neuen Ton. [25] Die Fenstergröße wird basierend auf dem physikalischen Modell auf 100 Millisekunden festgelegt. Des Weiteren ist es möglich eine Aussage über die Art des Grundsignals für die LPC-Synthese zu treffen. Die zu reparierenden Signale enthalten Sprache, sodass anstatt weißes Rauschen ein Impulssignal, dessen Frequenz der Grundfrequenz des Eingabesignals entspricht, eingesetzt wird. Bevor ein eingehendes Signal analysiert und anschließend synthetisiert werden kann, wird dieses zunächst durch eine Vorverarbeitung angepasst. Die notwendige Korrektur des Frequenzspektrums wird mit Hilfe eines, der Analyse vorgeschalteten, Filters umgesetzt.

3.2. Abtastratenkonvertierung

Im Anschluss an die vorhergehende Synthese durch LPC soll die Abtastrate des Signals reduziert werden. Die Reduktion der Abtastrate führt zu einer Erhöhung der Bandbreite der Frequenzbänder des Spektrums, wodurch diese zusammengefasst werden und somit der Einfluss der fehlenden Segmente reduziert sowie die Sprachqualität leicht verbessert wird. Die Abtastrate des Ursprungssignales beträgt

44100 Hz und muss auf eine für alle Signale gleiche, vorher festgelegte Abtastrate reduziert werden. Für die anschließende Evaluation wird die Abtastrate auf 8000 Hz festgelegt, dies ergibt sich durch den Einsatz eines automatischen Spracherkennungssystems, dessen Modell für Signale mit dieser Abtastrate ausgelegt ist.[13] Man unterscheidet in der Abtastratenkonvertierung zwei Fälle basierend auf der Art der Veränderung der Abtastrate. Die Erhöhung der Abtastrate eines Signals wird als Abtastratenerhöhung, die Reduktion der Rate als Abtastratenreduktion bezeichnet. Für beide Fälle ist es nötig, das vorliegende zeitdiskrete Signal zu bearbeiten und dieses zunächst in Zeitfenster zu zerlegen. Dies soll mit Hilfe einer schnellen Fourier-Transformation realisiert werden, welche die effiziente Berechnung der diskreten Fourier-Transformation ermöglicht.

3.2.1. Abtastratenerhöhung

Die Erhöhung der Abtastrate wird bestimmt durch einen Interpolationsfaktor, welcher das Verhältnis des Zielsignales mit höherer Abtastrate zum Ausgangssignal mit niedrigerer Abtastrate beschreibt. Es werden die Fälle eines ganzzahligen und eines rationalen Faktors unterschieden, da für diesen Anwendungszweck nur ganzzahlige Interpolationsfaktoren auftreten, kann der letztere Fall ignoriert werden. Die Abtastratenerhöhung mit ganzzahligem Interpolationsfaktor findet in zwei Schritten statt. Im ersten Schritt wird das Ausgangssignal in das Signal mit der gewünschten Abtastrate konvertiert. Für die praktische Umsetzung muss dieses Verfahren auf das zuvor in Zeitfenster zerlegte Signal angewendet werden. Hierbei bleibt die Anzahl der Zeitfenster für das Signal konstant, jedoch innerhalb jedes Fensters wird die Größe des Frequenzbereiches mit dem Interpolationsfaktor skaliert. Der ursprüngliche Frequenzbereich wird nun auf den skalierten Bereich gestreckt, wobei durch das Strecken die Intensität des Signales ebenfalls um den Interpolationsfaktor skaliert wird. Zur Korrektur dessen wird entweder jeder Eintrag des Frequenzbereiches durch den Interpolationsfaktor geteilt oder die duplizierten Einträge genullt, wodurch ebenfalls die Intensität auf das Original reduziert wird. Alternativ wird eine geringere Laufzeit pro Zeitfenster erzielt, indem die ursprünglichen Frequenzen im Abstand des Interpolationsfaktors N in ein mit Nullen gefülltes Feld einfügt werden. Vergleicht man die Abläufe beider Implementationen an einem Beispiel (siehe 3.1) wird die verringerte Laufzeit deutlich.

Das so entstandene Signal wird im zweiten Schritt durch einen Tiefpass geleitet, welcher die Nyquist-Frequenz des Ursprungssignales als Grenzfrequenz besitzt. Die Nyquist-Frequenz des Ursprungssignales ist hierbei als die Hälfte der Abtastrate definiert und beträgt für diesen Anwendungsfall 22050 Hz. Nach der Filterung des Signals ergibt sich das fertige Signal mit erhöhter Abtastrate.

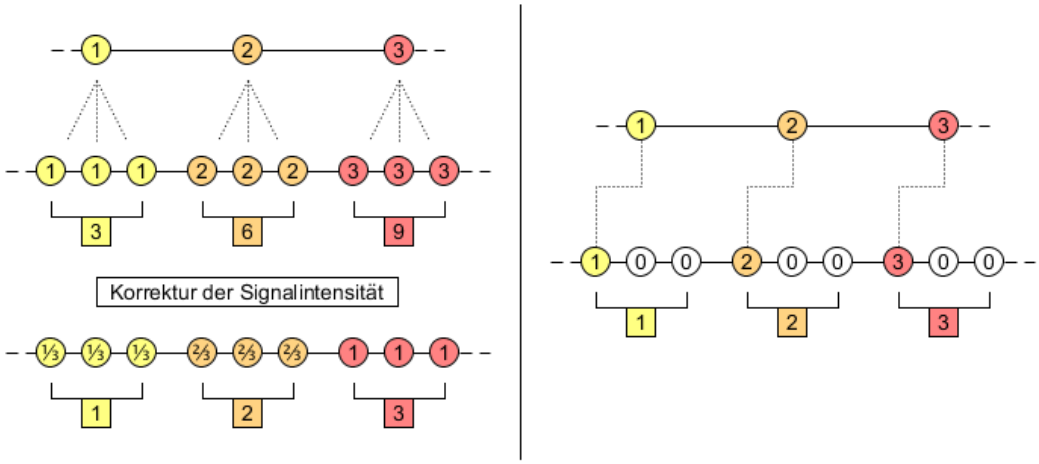

Abbildung 3.1.: Schematischer Vergleich beider Methoden zur Streckung des Frequenzbereiches um den Faktor 3 für einen Abschnitt eines Fensters. (links: Duplikation mit Intensitätskorrektur, rechts: direktes Einfügen in ein genulltes Feld)

3.2.2. Abtastratenreduktion

Die für diesen Anwendungsfall benötigte Reduktion der Abtastrate kann gleich der Abtastratenerhöhung in zwei Fälle, basierend auf dem Interpolationsfaktor, eingeteilt werden. Der Interpolationsfaktor beträgt für den Anwendungsfall $\frac{44100\,Hz}{8000\,Hz} = 5,5125$ und ist somit rational. Es wird jedoch im Folgenden zunächst die Abtastratenreduktion für ganzzahlige Interpolationsfaktoren betrachtet, da diese Bestandteil für die Reduktion mit rationalem Faktor ist. Im ersten Schritt wird das Signal mit einem Tiefpassfilter, welcher gleich der Abtastratenerhöhung die Nyquist-Frequenz als Grenzfrequenz besitzt, gefiltert. Dies ist nötig um alle Frequenzen oberhalb der halben Abtastfrequenz zu entfernen, da es sonst zu einer Verzerrung des Signals, dem Alias-Effekt, kommt.[11] Für den zunächst betrachteten Fall des ganzzahligen Faktors N wird für jedes Zeitfenster nur jede N-te Frequenz aus dem Frequenzbereich jedes Zeitfensters abgetastet und aus diesen anschließend ein neues Zeitfenster gebildet. Eine weitere Veränderung der Frequenzen ist nicht nötig, da sich die Intensität des Signales bei dieser Konvertierung nicht verändert. Der Fall eines rationalen Interpolationsfaktors wie in diesem Anwendungsfall lässt sich nun lösen, indem zunächst eine ganzzahlige Abtastratenerhöhung um einen Faktor U (siehe 3.2.1) durchführt wird, gefolgt von einer ganzzahligen Reduktion um den Faktor D. Das Verhältnis beider Interpolationsfaktoren erfüllt die folgende Gleichung:

$$f_{Ausgang} * \frac{D}{U} = f_{Ziel}$$

Es wird deutlich, dass die gesuchten Interpolationsfaktoren direkt aus der Ausgangs- bzw. Zielabtastrate ermittelt werden können:

$$\frac{D}{U} = \frac{f_{Ausgang}}{f_{Ziel}} = \frac{44100}{8000} = \frac{441}{80} \tag{3.1}$$

Um die Laufzeit des Algorithmus zu verkürzen wird das Verhältnis der Abtastraten soweit wie möglich gekürzt. Es ergibt sich folglich, dass für das Reduzieren der Abtastrate von 44100 Hz auf 8000 Hz zunächst eine Abtastratenerhöhung um den Faktor 80 und anschließend eine Reduktion um den Faktor 441 stattfindet. Eine weitere Optimierung ist hierbei möglich, da durch das Verketten der Algorithmen nach der Erhöhung und vor Beginn der Reduktion jeweils ein Tiefpassfilter angewendet wird. Das doppelte Filtern mit verschiedenen Grenzfrequenzen ist überflüssig und kann durch ein einmaliges Filtern mit der geringeren der beiden Filterfrequenzen ersetzt werden.

3.3. Sampleerzeugung

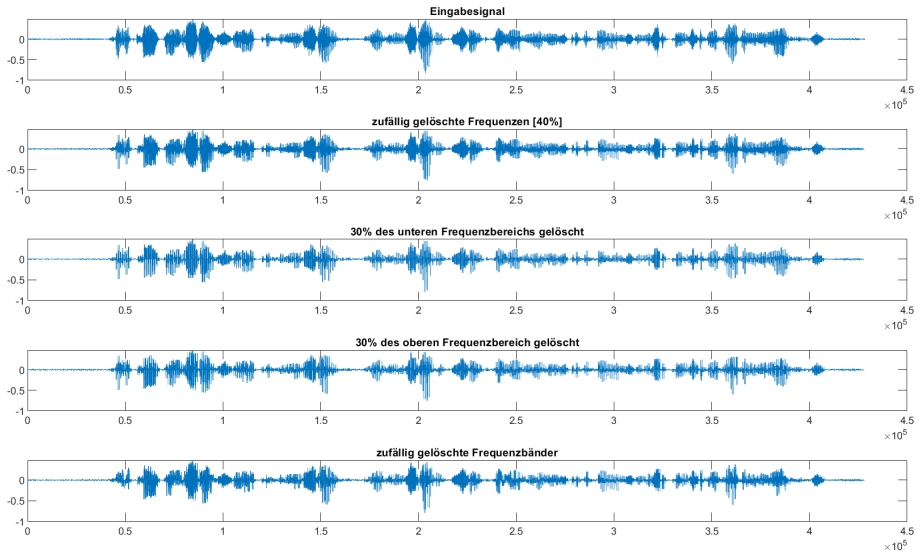

Abbildung 3.2.: Grafische Darstellung eines Originalsignals mit männlichem Sprecher mit den, auf vier unterschiedliche Arten, zerstörten Signalen

Die in den vorherigen Abschnitten vorgestellten Algorithmen zur Reparatur der Signale müssen auf ihre tatsächliche Tauglichkeit evaluiert werden, hierfür wird zunächst eine ausreichend große Menge an Beispielsignalen benötigt. Ausgangspunkt

für den Einsatz der Algorithmen sind durch CASA separierte Sprachsignale, welche jeweils eine einzelne Stimme enthalten. Diese haben durch das Trennen mittels Zeit-Frequenz-Maskierung Fehler in Form von fehlenden oder zusätzlichen Frequenzbereichen anderer Stimmen erhalten. Anstatt ein vollständiges CASA-System zu simulieren ist es für diese Arbeit zweckmäßig Audiosignale, in welchen nur eine Stimme vorkommt, gezielt zu zerstören und somit die Auswirkungen der Zeit-Frequenz-Maskierung zu simulieren. Für die Abdeckung aller möglichen Szenarien und Grenzfälle wird jedes Ausgangssignal auf vier verschiedene Arten zerstört, dies ist in Abbildung 3.2 für das Audiosignal eines männlichen Sprechers dargestellt. Das erste Erzeugungsverfahren löscht zufällige Frequenzen des Signales bis ein festgelegter prozentualer Anteil der Gesamtheit des Frequenzbereiches gelöscht wurde. Diese Methode erzeugt Signale, welche durch die fehlenden Einzelfrequenzen ein erhöhtes Signal-Rausch-Verhältnis besitzen. Sie werden als Testfall für die Ausgabe eines CASA-Systems betrachtet, bei welchem die vorhandenen Hintergrundgeräusche oder anderen Stimmen nicht perfekt gefiltert wurden. Die beiden folgenden Arten der Sampleerzeugung können aufgrund ihrer Ähnlichkeit zusammen betrachtet werden. Es wird hierbei erneut prozentual basierend auf der Gesamtmenge der Frequenzen der obere bzw. untere Frequenzbereich gelöscht. Dadurch werden die Fälle simuliert, bei denen durch die Zeit-Frequenz-Maskierung überlappende Stimmen im hohen bzw. niederen Frequenzbereich einem anderen Signal zugeordnet wurden und folglich im Ausgabesignal des eigentlich Sprechenden fehlen. Die Unterscheidung für hohe und niedrige Frequenzen ist hierbei für die Betrachtung von hohen und tiefen Stimmen, wie sie bei Männern und Frauen vorkommen, nötig. Das vierte Verfahren unterscheidet sich durch dessen Funktionsweise von den vorhergehenden drei Verfahren. Hierbei wird erstmals der Frequenzbereich nicht als Gesamtheit betrachtet, sondern es findet eine Zerlegung in Frequenzbänder statt. Dies ermöglicht es Frequenzteilbereiche, welche mit dem Frequenzumfang einer Stimme verglichen werden können, aus der Mitte des Frequenzspektrums zu entfernen. Folglich wird ein Signal generiert, welches fehlende Stimmsegmente im gesamten Signal aufweist. Es wird hierbei der Fall simuliert, dass der Frequenzumfang eines Sprechers falsch erkannt wurde und Frequenzbereiche fehlen. Die Verfahren decken somit sowohl das Fehlen von Stimmsegmenten sowie das Anfügen von Störsignalen in Form von anderen Stimmen oder Hintergrundgeräuschen ab. Folglich bilden die mit den vorgestellten Verfahren generierten Beispiele eine ausreichend große Menge an fehlerbehafteten Ausgaben für die Simulation der durch die Zeit-Frequenz-Maskierung auftretenden Fehler.

3.4. Plan zur Analyse

Zur Klärung der Forschungsfrage, wie CASA separierte Sprachsignale repariert werden können, werden fehlerfreie Ausgangssamples mit hoher Sprachqualität verwendet. Jedes Sample enthält nur einen Sprecher und wird zunächst mit den, in den vorherigen Abschnitten vorgestellten, Methoden gezielt zerstört. Dieser Vorgang er-

zeugt Beispielsignale, welche die Ausgabe eines CASA Systems simulieren. Die so generierten Beispielsignale werden anschließend durch Kombination der vorgestellten Verfahren repariert. Der erste Reparationsschritt wird hierbei durch die Kodierung des Signals unter Verwendung eines LPC-Vocoders realisiert. Im zweiten Schritt wird das kodierte Signal durch eine Abtastratenkonvertierung auf eine Abtastrate von 8000 Hz reduziert. Zur Überprüfung der Reparaturverfahren werden folgende Hypothesen aufgestellt:

H1 Das Reparieren von Signalen durch die Kodierung mittels LPC-Vocoder oder die Reduktion der Abtastrate führt zu einer Erhöhung der Sprachqualität.

H2 Das Reparieren von Signalen durch die Kodierung mittels LPC-Vocoder und anschließender Reduktion der Abtastrate führt zu einer Erhöhung der Sprachqualität.

H3 Das Reparieren von Signalen durch die Kodierung mittels LPC-Vocoder oder die Reduktion der Abtastrate wirkt der Zerstörung des Signales entgegen und erzeugt ein dem Ursprungssample ähnlicheres Signal.

H4 Das Reparieren von Signalen durch die Kodierung mittels LPC-Vocoder und anschließender Reduktion der Abtastrate wirkt der Zerstörung des Signales entgegen und erzeugt ein dem Ursprungssample ähnlicheres Signal.

Zum Prüfen der aufgestellten Hypothesen werden verschiedene statistische Verfahren eingesetzt und ausgewertet. Für jede der eingesetzten Metriken werden jeweils die Signale nach der Kodierung mittels LPC-Vocoder, nach Reduktion auf 8000 Hertz sowie der Kombination beider Reparaturschritte verglichen. Das Prüfen der Sprachqualität geschieht zunächst durch den Vergleich der Signal-Rausch-Verhältnisse der Signale, sodass die Veränderung der Signalenergie im Vergleich zur Rauschenergie untersucht werden kann. Im Anschluss werden die Signal transkribiert und die erhaltenen Texte analysiert. Hierzu werden die richtig erkannten Worte, die Wortfehlerrate sowie der Wortinformationsverlust berechnet. Jedes der Verfahren gibt hierbei Aufschluss über einen anderen Aspekt der Worterkennung. Die Analyse der transkribierten Werte kann anschließend mit der Auswertung der Signal-Rausch-Verhältnisse verglichen werden. Das Überprüfen der Hypothesen H3 und H4 wird durch die Berechnung des RMSE realisiert. Dieser statistische Wert zeigt basierend auf seiner Größe die Ähnlichkeit der beiden zuvergleichenden Signale und kann somit direkt ausgewertet werden.

4. Implementation

4.1. Sampleerzeugung

Die Erzeugung der Beispielsignale für die Evaluation der Reparationsverfahren soll durch vier verschiedene Methoden realisiert werden. Jede der Methoden erhält ein fehlerfreies Eingabesignal und erzeugt durch das gezielte Löschen verschiedener Frequenzbereiche ein Beispielsignal. Unabhängig von der konkreten Implementation der Modulation des Frequenzbereiches kann der Programmablauf durch das in Abbildung 4.1 dargestellte Blockdiagramm zusammengefasst werden. Im ersten Schritt wird das Signal durch die Hamming-Fensterfunktion in Abschnitte, sogenannte Fenster zerlegt. Für jedes generierte Fenster wird anschließend eine schnelle Fourier-Transformation (engl. fast Fourier transform, daher kurz FFT) durchgeführt, wodurch der Frequenzbereich des Signalabschnittes bearbeitet werden kann. Es findet anschließend das gezielte Löschen der Frequenzen bzw. Frequenzbereiche statt, bei welchem die, je nach Methode bestimmten, Frequenzbereiche durch Nullen ersetzt werden. Nach Abschluss des Löschvorgangs wird das Fenster durch eine inverse schnelle Fourier-Transformation (IFFT) rücktransformiert und der Vorgang für alle weiteren Fenster wiederholt. Nach Bearbeitung aller Fenster werden diese zur Generierung eines zusammenhängenden Signals miteinander verknüpft. Das so erstellte fertige Ausgabesignal unterscheidet sich nur durch die fehlenden Frequenzen vom Eingabesignal, andere Eigenschaften wie die Länge wurden durch das Verfahren nicht verändert.

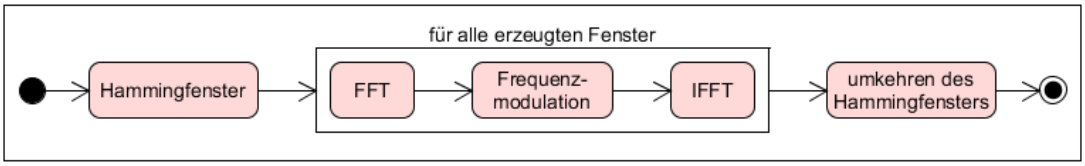

Abbildung 4.1.: Flussdiagramm der Erzeugung der Beispielsignale für die Reparaturverfahren

4.2. Linear Predictive Coding

Für diese Arbeit wird die Implementation des LPC-Vocoders durch eine bereits existierende Bibliothek bereitgestellt werden. Für die Auswahl einer geeigneten Bibliothek werden im Folgenden verschiedene Bibliotheken gegenübergestellt und über verschiedene Kriterien untereinander verglichen. Die für die Auswahl entscheidenden Kriterien sind hierbei, dass die gewählte Bibliothek Free und Open Source Software (FOSS) ist, der vorhandene Quelltext hinreichend dokumentiert ist und die Bibliothek Möglichkeiten zur Anpassung des Ergebnisses durch verändern von Parametern bereitstellt. Die erste Bibliothek wurde vom Sprachverarbeitungsteam des Mathworks-Projektes erstellt und ist in Matlab geschrieben. Neben der enthaltenen Dokumentation des Quelltextes bietet diese ebenfalls die Möglichkeit der individuellen Parametrisierung der Ergebnisse und erfüllt somit bereits zwei Auswahlkriterien. Zusätzlich ermöglicht diese Bibliothek das einfache Finden der Parameter durch bereitstellen einer grafischen Benutzeroberfläche. Das Verwenden dieser Bibliothek setzt jedoch den Zugang zu einem Rechner mit Matlab voraus und ist folglich nicht vollständig kostenfrei möglich. Es existieren weitere zahlreiche Implementationen eines LPC-Vocoders in Matlab, allerdings bieten diese alle einen geringeren Funktionsumfang und erfüllen durch die Wahl der Programmiersprache nicht das Auswahlkriterium einer Free and Open Source Software. Weitere Implementationen sind beispielsweise in Python, C++ oder JavaScript zu finden. Eine einfache Implementation des Algorithmus in Python wird durch John Williamson [34] bereitgestellt. Die Bibliothek ist sehr gut dokumentiert und unter der MIT Lizenz lizenziert, somit handelt es sich hierbei um Free und Open Source Software. Die Bibliothek ermöglicht es jeden einzelnen Aspekt der LPC-Analyse und Synthese durch Parameter zu beeinflussen und ist durch die Möglichkeit der Parameterübergabe als Aufrufargumente sehr einfach in andere Projekte einzubinden. Eine weitere in Python verfasste Bibliothek entstand im Zuge des Papers von Mohammadi, Seyed Hamidreza und Kain, Alexander [23], welche neben LPC auch weitere Vocoder implementiert, welche durch die Verwendung eines vor trainierten neuralen Netzes umgesetzt werden. Weiterhin erwähnenswert ist der in JavaScript verfasste Vocoder für Webanwendungen, welcher die webbasierten Echtzeitkodierung von Sprachaufnahmen ermöglichen wird [16]. Aus den vorgestellten Bibliotheken soll nun eine, basierend auf den drei Auswahlkriterien, ausgewählt und als Implementierung des LPC-Vocoders genutzt werden. Die webbasierte Kodierung ist für diese Arbeit nicht brauchbar, da sämtliche Operationen auf lokaler Hardware umsetzbar sein sollen somit kann diese Implementierung direkt ausgeschlossen werden. Für den Vergleich der anderen Bibliotheken wurden diese in Tabelle 4.1, anhand der Auswahlkriterien aufgelistet. Es ergibt sich direkt, dass aus den verglichenen Implementationen des Algorithmus nur die Bibliothek von John Williamson alle drei Kriterien erfüllt. Folglich wird in dieser Arbeit die Bibliothek für die Implementierung des LPC-Vocoders eingesetzt.

Im Folgenden werden die konkreten Aufrufparameter für die Kodierung eines Eingabessignals mit der Bibliothek diskutiert. Die Parameter werden zur Verbesserung der Analyse basierend auf ihrem Verwendungszweck zunächst in drei Kategorien un-

Bibliothek	Sprache	Parametrisierung	FOSS	Dokumentiert
Speech Processing[21]	Matlab	Ja	Nein	Ja
DeJanu	Matlab	Nein	Nein	Ja
John Williamson[34]	Python	Ja	Ja	Ja
Mohammadi[23]	Python,TCL	Nein	Ja	Nein

Tabelle 4.1.: Vergleich der geprüften Bibliotheken für die Implementation eines LPC-Vocoders

terteilt. Die erste Kategorie umfasst alle Parameter, welche die Art des verwendeten Grundsignals für die LPC-Synthese verändern. Standardmäßig findet die Synthese des Signales durch den Algorithmus der Bibliothek auf Basis eines Sinussignales statt. Es wird dabei jedoch nicht der für die Arbeit gewünschte LPC-Vocoder eingesetzt, sondern es wird ein Synthesizer emuliert. Durch das Verwenden der Parameter –buzz bzw. –noise wird der LPC-Vocoder aktiviert und basierend auf der Wahl des Parameters wird das Grundsignal in Form eines Impulssignales auf einer übergebenen Frequenz bzw. als weißes Rauschen initialisiert. Durch die Vorbetrachtungen im Konzept ergibt sich direkt, das der Parameter –buzz verwendet werden muss, da die Eingabesignale Sprache enthalten. Die Erzeugung des Trägersignales erfordert die Übergabe eines Parameters, welcher die Frequenz des Grundsignales bestimmt. Die Bestimmung des exakten Wertes geschieht durch eine Frequenzanalyse des Eingabesignals. Betrachtet man das untere Diagramm der in Abbildung 4.2 dargestellte Analyse eines Beispielsignals mit männlichem Sprecher wird die Frequenzmodulation während der Aussprache von Vokalen und Konsonanten deutlich. Zur Repräsentation des Frequenzspektrums wird daher die Grundfrequenz des Signales übergben. Die Berechnung dieser erfolgt idealerweise durch das Filtern des Signales basierend auf der Analyse der harmonischen Schwingungen und anschließendem aussortieren der Zeitbereiche, welche keine Sprache enthalten. Dieser Schritt wird für diese Arbeit übersprungen, da die für die Reparatur verwendeten Signale partiell zerstört sind und somit eine Erkennung der harmonischen Schwingung zu Fehlern führt. Die Grundfrequenz wird folglich durch das Bilden des arithmetischen Mittels über den gesamten zeitlichen Verlauf der Frequenz berechnet und variiert für jedes Eingabesignal, sodass diese für jeden Anwendungsfall neu ermittelt werden muss. Für das in Abbildung 4.2 dargestellte Beispiel ergibt sich durch die beschriebene Berechnung eine Grundfrequenz von 131.1024 Hz, welche an die Bibliothek übergeben wird. Nach der Auswahl des Grundsignals werden nun alle Parameter betrachtet, welche direkten Einfluss auf die Analyse und Synthese haben. Die größe des Zeitfensters wurde bereits im Konzept auf 100 ms festgelegt, es bleibt daher nur die Bestimmung der konkreten Koeffizientenanzahl. Diese wird im Rahmen der im Konzept bestimmten Werte (vergleiche Tabelle 3.1) durch systematisches Ausprobieren bestimmt. Hierbei ergab eine Anzahl von 25 Koeffizienten das beste Ergebnis im Bezug auf das Signal-Rausch-Verhältnis sowie die allgemeine Sprachqualität. Die dritte und finale Kategorie umfasst alle Parameter, welche das Filtern des Eingabesignales

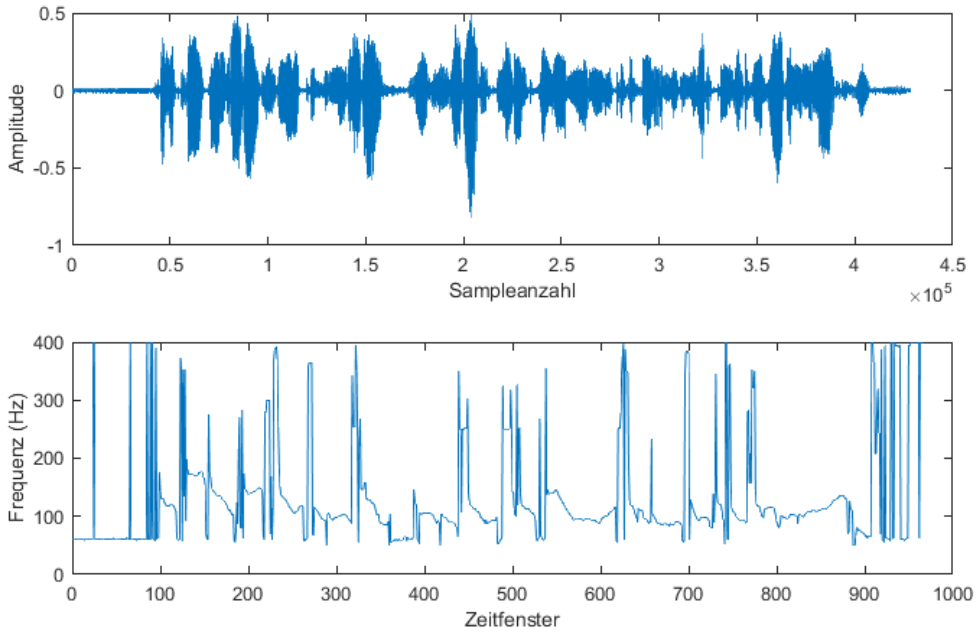

Abbildung 4.2.: Frequenzanalyse eines Signales mit männlichem Sprecher.

vor der Analyse beeinflussen. Die Vorverarbeitung zur Korrektur des Frequenzspektrums wird durch einen Bandbreitenfilter in Form eines Butterworth-Filters vierten Grades realisiert. Dieser filtert das Eingabesignal unter Angabe einer oberen und unteren Grenzfrequenz und flacht die Bereiche außerhalb des Filterbandes ab. Durch anschließendes normalisieren des Signales werden gleichmäßigere Amplituden in allen Frequenzbereichen erzielt. Das Frequenzband wird auf den Bereich zwischen 100 Hz und 4000 Hz festgelegt, sodass sowohl der untere als auch der obere Frequenzbereich angepasst wird und mögliche Störsignale reduziert werden. Das auf diese Weise gefilterte Signal wird anschließenden durch einen weiteren Verarbeitungsschritt der Vorverarbeitung moduliert. Diese Dezimierung wird durch einen ganzzahligen Faktor beschrieben, welcher die Abtastrate des Eingabesignales um diesen Faktor reduziert. Das gefilterte Signal mit reduzierter Abtastrate wird an die Analyse- und Synthesefunktionen übergeben und durch eine abschließende Abtastratenerhöhung um den verwendeten Dezimierungsfaktor auf die Eingabeabtastrate zurück konvertiert. Der Dezimierungsfaktor wird auf fünf festgelegt, sodass der LPC-Vocoder das gefilterte Signal mit einer Abtastrate von 8820 Hz als Eingabesignal erhält. Ein höherer Faktor sorgt für eine stärkere Glättung des Signales und wurde so gewählt das ein gutes Verhältnis zwischen der Reduktion des Signal-Rausch-Verhältnisses und erhaltener Sprachqualität entsteht. Zusammenfassend ist es nun möglich die Aufrufparameter für den Einsatz der Bibliothek als „--buzz f_0 --window 100 --order 25 --hp 4000 --lp 100 --decimate 5“ zu definieren, das Platzhalterzeichen f_0 repräsentiert hierbei die berechnete Grundfrequenz des Eingabesignals. Das durch den Vocoder erzeugte

Ausgabesignal wird in Form einer Audiodatei im WAVE-Format abgespeichert und wird bereits nach diesem Teilschritt für die spätere Evaluation zwischengespeichert.

4.3. Abtastratenkonvertierung

Neben der Kodierung durch den LPC-Vocoder wurde im Konzept die Abtastratenkonvertierung als weiteres Reparaturverfahren vorgestellt. Diese werden sowohl einzeln, als auch miteinander verknüpft eingesetzt. Die durch die Kodierung erhaltene Audiodatei muss zunächst eingelesen werden, damit diese als Eingabesignal für die Abtastratenkonvertierung genutzt werden kann. Der Funktionsaufruf geschieht durch Übergabe des Eingabesignals, der aktuellen Abtastrate von 44100 Hz sowie der Zielabtastrate von 8000 Hz. Die Funktion bestimmt anhand der im Konzept erklärten Formel 3.1 die Interpolationsfaktoren für die Erhöhung und Reduktion der Abtastrate. Die Umsetzung der Formel sowie das Kürzen der Faktoren ist in Pseudocode 4.1 in Zeile 1 zu sehen. In Zeile 3 wird die Abtastrate des Signales um den entsprechenden Faktor U erhöht und das Signal anschließend durch einen Tiefpass mit Nyquist-Frequenz als Grenzfrequenz gefiltert. Im Anschluss muss die Abtastrate um den Reduktionsfaktor D verringert werden. Das erhaltene Ausgabesignal besitzt nun eine Abtastrate von 8000 Hz und kann in Zeile 7 zurückgegeben werden. Das Verfahren wird sowohl als einzelnes Reparaturverfahren als auch in Kombination mit dem LPC-Vocoder untersucht.

Algorithmus 4.1 Pseudocode der Implementation der Abtastratenkonvertierung

```
1  [D,U] = ratio(44100 / 8000)
2
3  signal = upsample(signal, U)
4  signal = filter(signal.nyquist)
5  signal = downsample(signal, D)
6
7  return signal
```

4.4. Implementation der Metriken

Betrachtet man den Datenfluss für ein unzerstörtes Ausgangssignal ergibt sich der in Abbildung 4.3 dargestellte Ablauf, bei welchem das Signal zunächst durch die im Konzept erklärten vier Methoden zerstört wird. Anschließend werden die Beispielsignale wie dargestellt durch den Einsatz des LPC-Vocoders, der Abtastratenkonvertierung sowie einer Kombination beider repariert. Jedes der aufgerufenen Reparaturverfahren erzeugt ein Ausgabesignal, welche mit den Nummern (1) bis (3) gekennzeichnet sind. Zur Überprüfung der Hypothesen werden die reparierten Signale an die jeweiligen Metriken übergeben und die Ergebnisse dieser ausgewertet.

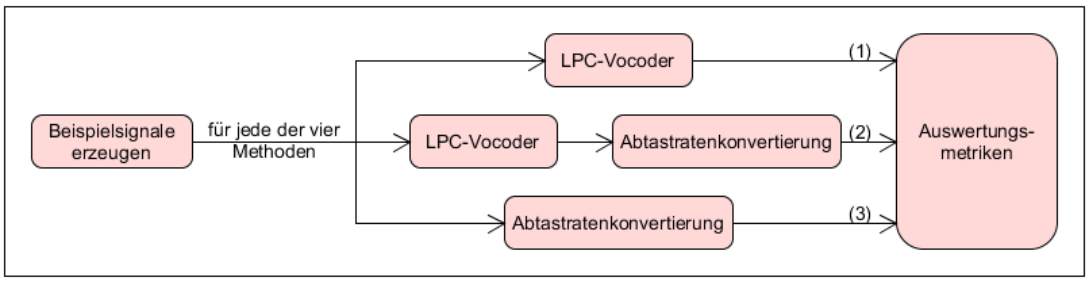

Abbildung 4.3.: Vollständiger Datenfluss eines Ausgangssamples durch alle Programmabschnitte

4.4.1. Signal-Rausch-Verhältnis

Die erste zu betrachtende Metrik ist das Signal-Rausch-Verhältnis (engl. signal-to-noise-ratio, daher kurz SNR). Dieses beschreibt das Verhältnis der Signalenergie σ_s zur Rauschenergie σ_r eines gegebenen Signals und kann durch

$$SNR = \frac{\sigma_s^2}{\sigma_n^2}$$

berechnet werden. Es gilt dabei, dass mit zunehmendem Signal-Rausch-Verhältnis die technische Qualität des Signals ebenfalls steigt. Die Verhältnisse der Energien unterscheiden sich hierbei meist um mehrere Größenordnungen, sodass eine logarithmische Darstellung der Größe zweckmäßig ist. Dies wird durch die Berechnung der Formel

$$SNR_{dB} = 10 * log_{10}\left(\frac{\sigma_s^2}{\sigma_n^2}\right) \tag{4.1}$$

realisiert und gibt einen Wert in Dezibel wieder. Durch die logarithmische Darstellung ist es nun möglich, dass trotz der positiven quadrierten Energien, ein negatives Ergebnis auftreten kann. Dies ist für die Gleichung 4.1 nur möglich, wenn der Logarithmus ebenfalls negativ ist. Es ergibt sich

$$0 < \frac{\sigma_s^2}{\sigma_n^2} < 1$$

und folglich

$$\sigma_s^2 < \sigma_n^2$$

Das negative Signal-Rausch-Verhältnis entsteht somit dadurch, dass die quadrierte Signalenergie kleiner ist als die quadrierte Rauschenergie. Für die so erhaltenen Werte gilt weiterhin, dass mit zunehmendem Verhältnis die Qualität zunimmt, sodass die Werte direkt miteinander verglichen werden können.

4.4.2. Transkription

Des Weiteren werden die Signale unter Einsatz eines automatischen Spracherkennungssystems transkribiert und die erkannten Worte verglichen, wodurch eine direkte Aussage über die vorhandene Sprachqualität getroffen werden kann. Für die Auswahl eines dem aktuellen Standes der Technik entsprechenden Spracherkennungssystems werden im Folgenden die cloudbasierten Systeme von Google, IBM und Microsoft betrachtet. Verglichen werden die kostenfreien Angebote der jeweiligen Anbieter basierend auf der Menge der Audiominuten, welche pro Monat mit diesen transkribiert werden können (vergleiche Tabelle 4.2).

Anbieter	Audiominuten pro Monat
Google Cloud	60
IBM Cloud Watson	500
Microsoft Azure	300

Tabelle 4.2.: Vergleich der kostenfreien Angebote von cloudbasierten Anbietern für Sprachtranskription

Es wird deutlich, dass mit dem kostenfreien Angebot von IBM Watson das größte Kontingent an Audiominuten pro Monat transkribiert werden kann. Folglich wird dieses für die Evaluation der zerstörten und reparierten Audiosignale eingesetzt. Nach erfolgreicher Registrierung erhält man Authentifizierungstoken und Adresse des persönlichen Endpunktes, welche die Interaktion mit dem System ermöglichen. Die konkrete Ansteuerung der API wird mit Python realisiert und verwendet die von IBM bereitgestellte Entwicklerbibliothek zur Nutzung der cloudbasierten Dienstleistungen. Die zu übergebenen Parameter für den Aufruf ergeben sich direkt aus den zu transkribierenden Signalen. Diese liegen in Form von WAVE-Audiodateien vor, sodass der Parameter content_type den Wert 'audio/wav' erhält. Die Wahl des für die Transkription zu verwendeten Modells ist hierbei basierend auf der Abtastrate des Signales unterschiedlich. Für die durch den LPC-Vocoder generierten Signale mit einer Abtastrate von 44100 Hz wird das Breitbandmodell en-US_BroadbandModel verwendet. Für die Signale mit reduzierter Abtastrate muss das korrespondierende Schmalbandmodell en-US_NarrowbandModel eingesetzt werden. Weiterhin werden die Optionen für das Übermitteln der Wortkonfidenz sowie der erkannten zeitlichen Wortpositionen aktiviert, sodass diese für eventuelle nähere Betrachtungen zur Verfügung stehen. Zusammengefasst ergibt sich der in 4.2 dargestellte Quelltext für den Aufruf der API. Dieser gibt das Ergebnis der Transkription in Form eines Datenstrings im JSON Format zurück, welcher für den anschließenden Vergleich analysiert werden muss.

Algorithmus 4.2 Aufruf der IBM-API zur Transkription einer Audiodatei

```
1  from ibm_watson import SpeechToTextV1
2
3  authenticator=IAMAuthenticator('Authentifizierungstoken')
4  service       =SpeechToTextV1(authenticator=authenticator)
5  service.set_service_url('API-Endpunkt-URL')
6
7  if (audio_file.samplerate==8000)
8    model_name = 'en-US_NarrowbandModel'
9  else
10   model_name = 'en-US_BroadbandModel'
11
12 service.recognize(audio=audio_file, model=model_name)
13 service.get_result()
```

4.4.3. Worterkennungsrate

Für die Analyse muss zunächst der transkribierte Text aus der erhaltenen JSON-Datei extrahiert werden. Im Anschluss werden die erhaltenen Sätze durch den Algorithmus 4.3 auf Ähnlichkeit verglichen. Die an die Funktion übergebenen, zu vergleichenden Sätze werden zunächst in den Zeilen 1 und 2 jeweils in ihre Worte aufgeteilt. Anschließend werden die beiden Wortmengen durch die Funktion in Zeile 4 auf die Anzahl der übereinstimmenden Wörter geprüft und die gefundenen Wörter aus beiden Mengen entfernt. Zum Erhalt einer sehr simplen Ähnlichkeitsfunktion ist es möglich die Menge der gefundenen Worte mit der Menge der gesamten Worte zu vergleichen um so einen Genauigkeitswert zu ermitteln. Durch dieses Verfahren werden jedoch nur exakt gleiche Wörter erkannt. Für eine genauere Bestimmung der Ähnlichkeit ist es ebenfalls nötig sehr ähnliche Wörter, welche sich beispielsweise nur durch einen Buchstaben unterscheiden, zu erkennen. Es werden hierfür aus den verbleibenden Wörtern, in den bereits analysierten Wortmengen, alle möglichen Paare gebildet und die Levenshtein-Distanz dieser gebildet. Die Levenshtein-Distanz beschreibt die minimale Anzahl an Zeichenoperationen, welche nötig sind um eine der Zeichenketten in die jeweils andere umzuwandeln. Folglich kann diese zur Bestimmung der Ähnlichkeit eingesetzt werden, dabei gilt: Je geringer die Distanz, desto größer die Ähnlichkeit der beiden Worte. In Zeile 7 wird nun für jedes Wort des zu vergleichenden Satzes das Paar mit der geringsten Levenshtein-Distanz ermittelt. Reduziert man die Länge des ursprünglichen Wortes um die ermittelte Levenshtein-Distanz und teilt diesen Wert durch die Länge des ursprünglichen Wortes, ergibt sich der richtig erkannte Anteil für jedes Wort des zu vergleichenden Satzes. Es ist jedoch möglich, dass die Levenshtein-Distanz größer ist als die Länge des ürsprünglichen Wortes, wodurch die erhaltene Wortähnlichkeit negativ wäre. Dies tritt auf, wenn beispielsweise keine Übereinstimmungen vorhanden sind und die zu verglei-

chenden Worte unterschiedliche Längen besitzen. Für diese Sonderfälle kann davon ausgegangen werden, dass die verglichenen Worte keinerlei Ähnlichkeit besitzen und die Wortähnlichkeit somit null gesetzt werden kann. Die so berechneten Wortanteile werden anschließend in Zeile 16 zur vorher erkannten Menge der gemeinsamen Worte addiert. Zur Berechnung der Ähnlichkeit der beiden Sätze wird der so berechnet Wert der richtig erkannten Worte und Teilworte durch die Gesamtanzahl der Worte des vollständigen Satzes geteilt.

Algorithmus 4.3 Funktion zum Vergleich zweier Sätze auf übereinstimmende Worte zur Berechnung der Ähnlichkeit

```
1  words_full = SplitSentence(complete_sentence);
2  words_comp = SplitSentence(sentence_to_compare);
3
4  words_common=FindRemoveCommonWords(words_full,words_comp);
5
6  for(word in word_comp)
7      [word_pair, lev_distance] =
                FindSmallestLevenshteinPair(word,words_full);
8      length = length(word_pair.word_full);
9
10     partial_word = (length - lev_distance)/length;
11
12     if partial_word < 0
13         partial_word = 0;
14     end
15
16     words_common = words_common + partial_word;
17 end
18
19 return (100 * (words_common / words_full.count));
```

4.4.4. Wortfehlerrate

Neben der Berechnung der richtig erkannten Worte ist die Berechnung der Wortfehlerrate (engl. word error rate, daher kurz WER) eine häufig eingesetzte Metrik für automatische Spracherkennungssysteme. Im Vergleich zur vorherigen Metrik beschreibt die WER einen statistischen Kostenwert, welcher nötig wäre um das erhaltene Ausgabewort in das Eingabewort zurückzuführen. Dieser Kostenwert wird durch die Formel

$$WER = \frac{S + D + I}{S + D + H}$$

berechnet. Die enthaltenen Variablen S,D und I beschreiben die Substituierungen, Löschungen und Einfügungen von Zeichen, welche nötig sind um das Ausgangswort

wieder herzustellen. Im Kontext der Gleichung bezeichnet H die Anzahl der exakt erkannten Worte. Die Werte der Variabelen S,D und I können beispielsweise durch eine aufbereitete Berechnung der Levenshteindistanz ermittelt werden, da für diese die Werte ebenfalls benötigt werden. Der so berechnete Wert der Wortfehlerrate kann ungleich der zuvor beschriebenen Worterkennungsrate größer als 1 werden. Die obere Schranke wird unter normalen Umständen gegeben durch

$$WER_{max} = \frac{max(N_1, N_2)}{N_1}$$

Die verwendeten Werte N_1 und N_2beschreiben die Menge der eingelesenen Eingabe- bzw. Ausgabewörter. Als Eingabewörter werden die transkribierten Texte der Urspsprungssignale (m1, m2, w1, w2) und als Ausgabewörter die transkribierten Texte der entsprechenden reparierten Signale verwendet. Die Realisierung der Gleichungen geschieht durch die Verwendung der Pythonbibliothek JiWER [24]. Mit dieser ergibt sich für den Vergleich der reparierten Signale mit dem Urspsrungssignal der im Pseudocode 4.4 aufgestellte vereinfachte Quelltext.

Algorithmus 4.4 Berechnung der Wortfehlerrate unter Einsatz der Pythonbibliothek JiWER

```
1  from jiwer import wer
2
3  error = wer(ursprungssignal.text, repariertessignal.text)
4
5  return error
```

4.4.5. Wortinformationsverlust

Eine weitere Metrik zur Analyse der transkribierten Texte ist der Wortinformationsverlust (engl. word information lost, daher kurz WIL). Dieser beschreibt eine Annäherung an den relativen Informationsverlust, welcher durch die Gleichung

$$RIL = \frac{H(X|Y)}{H(Y)}$$

berechnet werden kann. Die Funktion $H(...)$ berechnet für den übergebenen Parameter die Entropie, wobei X bzw. Y hierbei jeweils die Eingabe- bzw. Ausgabewörter repräsentieren. Aus dem Paper von Morris, Maier und Green [24] wird deutlich, dass eine Annäherung an diesen komplexen Wert auf Basis der Wichtung der in der Wortfehlerrate verwendeten Variablen H,S,D und I möglich ist. Der Wortinformationsverlust wird beschrieben durch

$$WIL = 1 - \left(\frac{H}{N_1} * \frac{H}{N_2}\right) \cong RIL$$

Die in der Gleichung auftauchenden Variablen entsprechen den in der Wortfehlerrate erläuterten. Die Umsetzung dieser Gleichungen wird erneut durch die Verwendung der Pythonbibliothek JiWER realisiert. Der zur Realisierung der Gleichungen verwendete Code (siehe Pseudocode 4.5) ergibt sich analog zum Algorithmus 4.4, welcher für die Wortfehlerrate eingesetzt wurde. Es wird dabei lediglich die in den Zeilen 1 und 3 importierte und aufgerufene Funktion zu wil(...) abgeändert, die Reihenfolge der übergebenen Aufrufparameter bleibt jedoch bestehen.

Algorithmus 4.5 Berechnung des Wortinformationsverlustes unter Einsatz der Pythonbibliothek JiWER

```
1  from jiwer import wil
2
3  error = wil(ursprungssignal.text, repariertessignal.text)
4
5  return error
```

4.4.6. RMSE

Zum Überprüfen der Hypothesen H3 und H4 werden die entsprechenden reparierten Signale mit dem Originalsample verglichen. Es existieren verschiedene Methoden zur Bestimmung der Ähnlichkeit zweier Signale. Für diese Arbeit wird die Ähnlichkeit zweier Signale A und B durch die Berechnung der Wurzel des Mittelwertes der quadrierten Differenz beider Signale (engl. root mean square error, kurz RMSE)

$$RMSE = \sqrt{\frac{1}{n} * \sum_{i=1}^{n} (A_i - B_i)^2} \tag{4.2}$$

realisiert. Der berechnete Wert ist aussagekräftig über die Abweichung der Signale zueinenader und wird mit zunehmender Ähnlichkeit der Eingabesignale kleiner. Der erhaltene Wert muss jedoch relativ zur Datensatzweite ausgewertet werden. Diese weicht für die verwendeten Signale ab, sodass zum Vergleich der erhaltenen Werte dieser zunächst normalisiert wird

$$RMSE_{norm} = \frac{RMSE}{(max(A,B) - min(A,B))}$$

Der Vergleich zweier Signale durch die vorgestellten Formeln ist jedoch nur möglich, wenn diese die exakt selbe Datensatzanzahl besitzen. Für die zerstörten und reparierten Signale ist es jedoch möglich, dass diese durch diverse Berechnungsschritte nicht übereinstimmen. Zur Behebung des Problems wird das kürzere beider Signale durch lineare Interpolation an die Länge des zweiten Signales angeglichen. Durch die Reduktion der Abtastrate wurde die Datenmenge des reparierten Signales jedoch um den verwendeten Interpolationsfaktor reduziert. Die Angleichung an das Originalsignal würde hierbei zu einem stark fehlerhaften Ergebnis führen, sodass

für den Vergleich das Originalsignal ebenfalls auf 8000 Hz reduziert wird. Angewendet auf die verwendeten Signale führt dies zu einem maximalen Fehler von circa 0,2%, welcher für die abschließende Evaluation toleriert wird. Die in Algorithmus 4.6 dargestellte Funktion realisiert die Berechnung des normalisierten RMSE. In Zeile 2 ist ebenfalls die zuvor beschriebene Angleichung der Signallängen durch lineare Interpolation sichtbar.

Algorithmus 4.6 Funktion zur Berechnung des normalisierten RMSE

```
1  rmse(data_real, data_predicted)
2      linInterp(data_real, data_predicted);
3
4      RMSE = sqrt(mean((data_real - data_expected)^2));
5
6      data_max = max(data_real, data_predicted);
7      data_min = min(data_real, data_predicted);
8
9      return (RMSE / (data_max - data_min));
10 end
```

Für den tatsächlichen Vergleich der Werte muss für jedes Ursprungssignal und jede der vier Erzeugungsmethoden der RMSE berechnet werden. Zur Klärung der Hypothesen wird als Bezugspunkt das jeweils zugehörige Ursprungssignal verwendet. Dieses Verhalten wird durch Algorithmus 4.7 beschrieben. Für die Berechnung des RMSE für das durch den LPC-Vocoder kodierten Signals in Zeile 9 ist dies ohne Problem möglich. Dies ist jedoch nicht für alle Signale der Fall. Für die Berechnung des RMSE des nach der Kodierung entstandenen Signals ist eine Reduktion des Ursprungssignales für den Vergleich (siehe Zeile 5) nötig, da durch die Reduktion der Abtastrate die Größe des Datensatzes um den Interpolationsfaktor reduziert wurde. Der Vergleich des reparierten Signales, welches nur durch die Abtastratenkonvertierung bearbeitet wurde ergibt jedoch ein Problem. Für die Berechnung des RMSE wäre es nötig das reduzierte Usprungssignal als Bezugssignal einzusetzen. Dieses ist jedoch, da zur Berechnung dessen die selbe Funktion eingesetzt wird, identisch mit dem reparierten Signal. Folglich wäre der Wert des RMSE für diese Fälle immer 0. Es wird daher das reparierte Signal in Zeile 15 wieder auf eine Abtastrate von 44100 Hz angehoben und mit dem unveränderten Ursprungssignal verglichen. Nach der Berechnung der Werte für jeweils eine Erzeugungsmethode eines Ursprungssignales und der zugehörigen zwei reparierten Signale werden diese in Form eines Diagrammes visualisiert.

Algorithmus 4.7 Funktion zur Berechnung der normalisierten RMSE-Werte der reparierten Signale

```
1  for(sample = [m1, m2, w1, w2])
2      data_orig = readAudio(sample);
3      rmse_orig = rmse(data_orig, data_orig);
4
5      data_orig_res = resample(data_orig, 8000Hz);
6
7      for(method = [band, low, high, rand])
8          data_voc = readAudio(sample.method.vocoded);
9          rmse_voc = rmse(data_voc, data_orig);
10
11         data_voc_res = readAudio(sample.method);
12         rmse_voc_res = rmse(data_voc_res, data_orig_res);
13
14         data_res = readAudio(sample.method.resampled);
15         data_res = resample(data_res, 441000Hz);
16         rmse_res = rmse(data_res, data_orig);
17     end
18
19     diagram(rmse_voc,rmse_voc_res,rmse_res);
20 end
```

5. Evaluierung

In diesem Kapitel werden die Ergebnisse der im vorherigen Kapitel implementierten Metriken präsentiert. Für den Einsatz der Metriken werden als Grundlage vier Urspsprungssignale verwendet, welche durch die Abkürzungen m1, m2, w1 und w2 gekennzeichnet sind. Diese sind in Abbildung 5.1 dargestellt. Jedes der Signale beinhaltet einen einzelnen Sprecher und besitzt zudem verschiedene Texte und Längen. Je zwei der insgesamt vier Ursprungssignale besitzen einen weiblichen (w1, w2) und zwei einen männlichen Sprecher (m1, m2), dabei existiert in diesen Paaren jeweils ein langes und ein kurzes Signal. Auf diese Ursprungssignale werden die vier Methoden zur Beispielerzeugung angewendet. Es entstehen hierbei die beschädigten Signale, welche anschließend durch die drei vorgestellten Varianten repariert werden. Die erzeugten, beschädigten Signale werden durch eine Abkürzung der Form „Ursprungssignale-Methode“ in den folgenden Diagrammen gekennzeichnet. Die Suffixe „high“ bzw. „low“ stehen hierbei stellvertretend für die Methoden, in welchen der obere bzw. untere Frequenzbereich entfernt wird. Durch „band“ wird das Löschen der Frequenzbänder und durch „rand“ das vollständig zufällige Löschen von Frequenzen gekennzeichnet. Einige der verwendeten Methoden entfernen einen gewissen prozentualen Anteil durch zufälliges Löschen von Frequenzen. Es ist daher für das Aufstellen der Statistiken nötig die verwendeten Methoden mehrmalig einzusetzen. Für die folgenden Diagramme wurde jedes der Ursprungssignale insgesamt 50 mal mit jeder der Methoden in ein beschädigtes Beispielsignal umgewandelt und anschließend repariert. Für jedes Ursprungssignal ergeben sich folglich für jede der 50 Wiederholungen vier Beispielsignale, aus welchen jeweils 12 reparierte Signale erzeugt werden. Zur Präsentation der Ergebnisse werden die Daten, nach Reparaturverfahren getrennt, in jeweils drei Diagramme unterteilt, in welchen die einzelnen Beispielsignale gegenübergestellt werden.

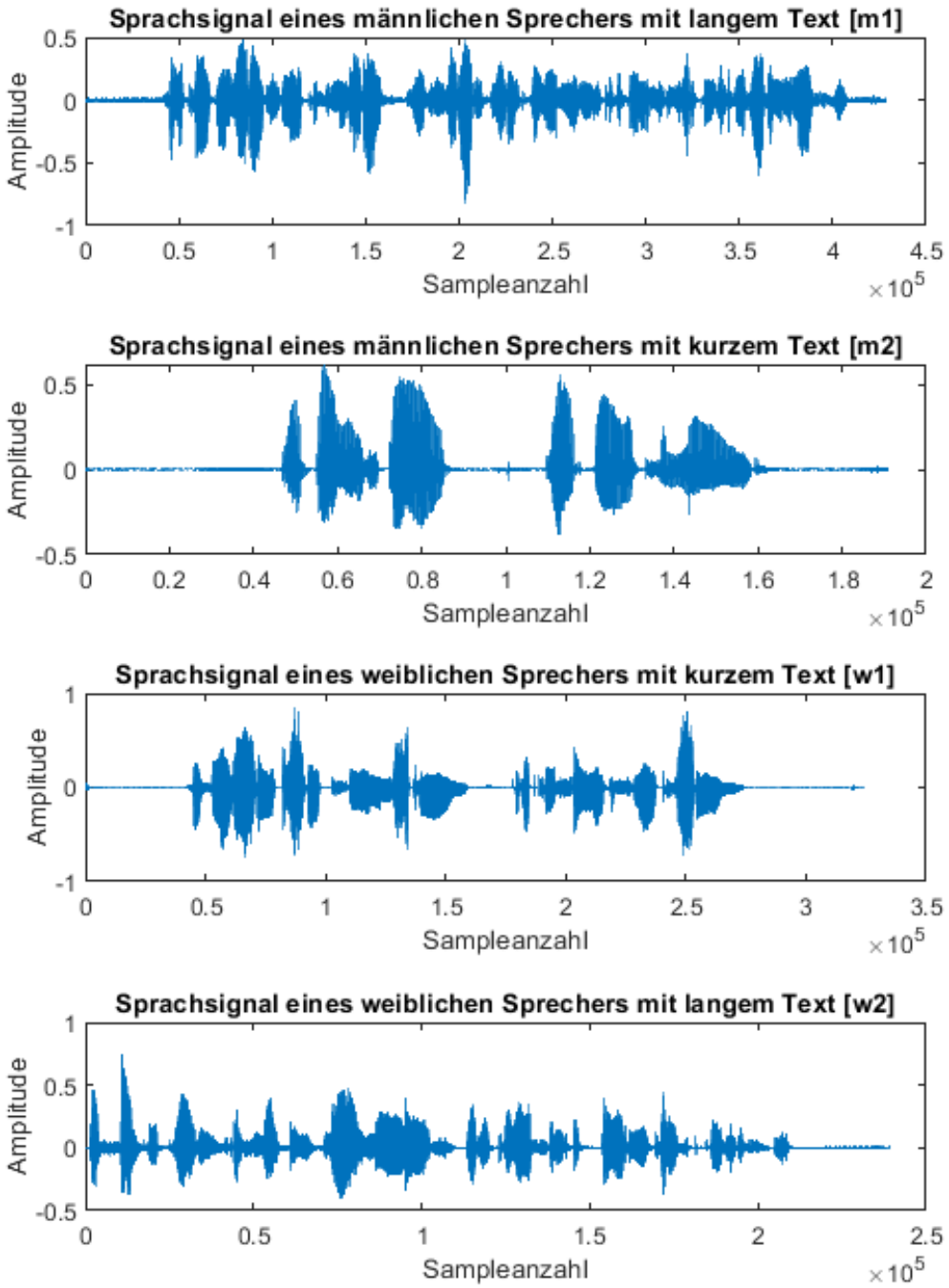

Abbildung 5.1.: Darstellung der für die Evaluation verwendeten Grundsignale

5.1. Signal-Rausch-Verhältnis

Zur Überprüfung der Hypothesen werden zunächst die Signal-Rausch-Verhältnisse der reparierten Signale analysiert. Für jede der 4 Methoden und jedes der drei Reparaturverfahren wurde ein Kastendiagramm aus den Ergebnissen der 50 Durchläufe generiert. Die durch diese Metrik erhaltenen Daten sind in Abbildung 5.2 für das Ursprungssignal w1 dargestellt. Auf Grund der Ähnlichkeit der Ergebnisse der nicht dargestellten Diagramme der anderen Ursprungssignale werden diese hier nicht dargestellt, sondern sind in Anhang A.1 zu finden.

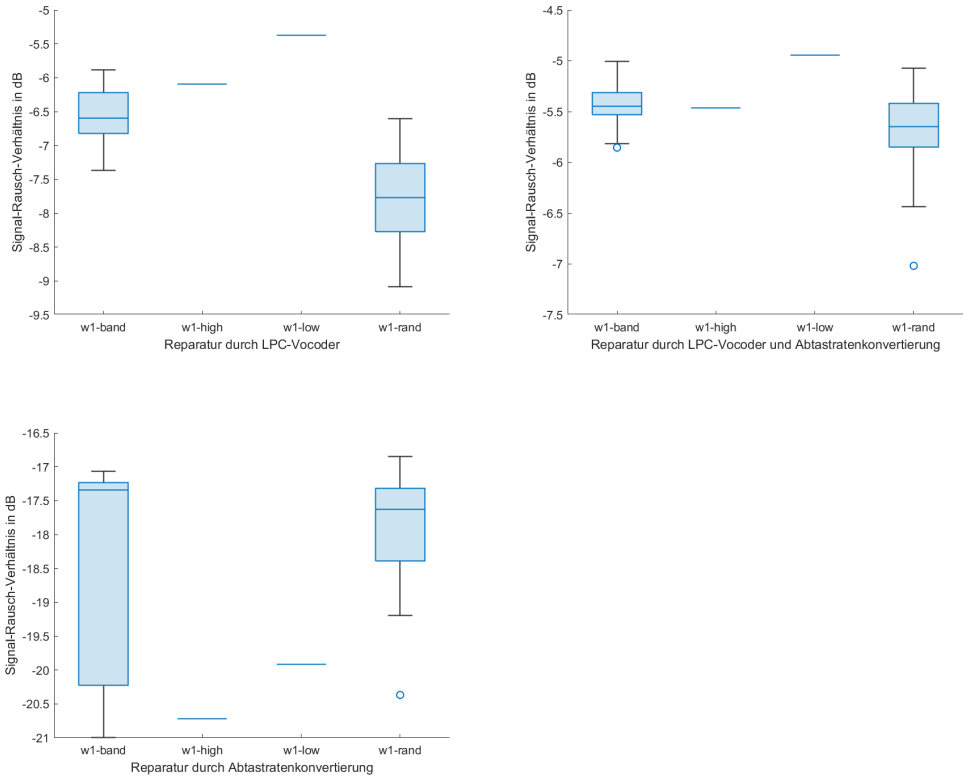

Abbildung 5.2.: Signal-Rauschverhältnisse aller reparierten Beispielsignale des Ursprungssignales „w1“

Bei Betrachtung der Ergebnisse fällt direkt auf, dass die Werte der Signal-Rausch-Verhältnisse negativ sind. Aus der Formel zur Berechnung der SNR in Dezibel (Formel 4.1) ergibt sich folglich, dass die Rauschenergie größer als die Signalenergie ist. Eine Zunahme des Signal-Rausch-Verhältnisses kann folglich als Erhöhung der technischen Qualität des Signales aufgefasst werden.

In den Diagrammen der drei Reparaturverfahren für das Ursprungssignal „w1“ (Abbildung 5.2) ist ersichtlich, dass die Kastendiagramme der Methoden der Fre-

quenzbandlöschung („band“) sowie des zufälligen Löschens („rand“) eine variierende Ergebnissmenge zeigen. Im Vergleich dazu wird für das Löschen des oberen bzw. unteren Frequenzbereiches des Frequenzspektrums („high“ bzw. „low“) nur ein einzelner Wert erhalten. Die Ursache liegt in der unterschiedlichen Funktionsweise der Erzeugungsmethoden. Die Frequenzbandlöschung sowie das zufällige Löschen der Frequenzen besitzen zufallsbasierte Aspekte, welche eine Streuung der Ergebnisse verursachen. Für beide Methoden mit statischem Ergebnis ist dies nicht der Fall, da sich die Größe des Frequenzspektrums für ein Ursprungssignal nicht verändert. Folglich bleibt der entfernte Frequenzbereich ebenfalls konstant, sodass die Ausgabe dieser Methoden konstant über alle Durchläufe bleibt.

Die Kastendiagramme zeigen deutlich die Streuung der Signal-Rausch-Verhältnisse, welche für jede der beiden zufälligrn Methoden innerhalb eines Reparaturverfahrens ähnlich ist. Eine Ausnahme bildet hierbei die Reparatur durch das Konvertieren der Abtastrate, welche im unteren linken Diagramm dargestellt ist. Es ist hierbei ersichtlich, dass für die erhaltenen Ergebnisse die Streuung der Ergebnisse des zufälligen Löschens deutlich kleiner als die Streuung der Frequenzbandlöschung ist. Der Ausreißer im Kastendiagramm des zufälligen Löschens zeigt jedoch, dass dies nicht für alle Werte gilt. Die scheinbare geringere Streuung kann folglich nur durch die gewählte Anzahl an Durchläufen verursacht worden sein. Das Wählen einer deutlich größeren Versuchsanzahl würde dies korrigieren.

Für jedes der drei Reparaturverfahren ergibt sich aus den Diagrammen in Abbildung 5.2, dass die berechneten Signal-Rausch-Verhältnisse für das Löschen des unteren Frequenzbereiches größer sind, als die des Löschens des oberen Frequenzbereiches. Betrachtet man jedoch die Ergebnisse für das Ursprungssignal „w2“ in Abbildung 5.3, zeigt sich der exakt umgekehrte Fall. Diese Erscheinung tritt unabhängig des Geschlechtes des Spreches sowie der Länge des vorgelesenen Textes auf. Aus den Diagrammen ist es möglich für jedes der Reparaturverfahren die Energiereduktion für jede der verwendeten Erzeugungsmethoden zu vergleichen. Jede der Methoden simuliert jedoch das Auftreten von unterschiedlichen Beschädigungen der Signale, sodass die Reparaturverfahren untereinander verglichen werden müssen.

Vergleicht man die Gesamtheit der Reparaturverfahren zunächst mit den ursprünglichen beschädigten Signalen ergibt sich, dass für alle Verfahren das Signal-Rausch-Verhältnis und somit die technische Qualität zugenommen hat. Im Vergleich der Reparaturverfahren untereinander wird deutlich, dass das Kodieren des Signales mit anschließender Abtastratenkonvertierung auf 8000 Hz die größten Werte erzeugt und somit die größte Menge an Energie reproduziert. Dies wird bei Betrachtung der vier verwendeten Generierungsmethoden erneut deutlich, da der schlechteste Wert jeder Methode über den korrespondierenden Median beider anderer Reparaturmethoden liegt. Aus den dargestellten Werten lässt sich weiterhin schließen, dass das Kodieren der Signale unter Verwendung des LPC-Vocoders zu einer starken Zunahme des Signal-Rausch-Verhältnisses und somit einer starken Erhöhung der technischen Qualität führt. Die vorliegenden Daten zeigten jedoch, dass dieses Ergebnis durch die anschließende Reduktion der Abtastrate weiter verbessert werden kann. Der durch den Einsatz des Vocoders auftretende Zuwachs an Energie ist durch dessen

Funktionsweise erklärbar, da hierbei zusätzliche Informationen generiert und eingefügt werden. Die reine Reduktion der Abtastrate als Reparaturverfahren führt nur zu einer geringen Erhöhung des Signal-Rausch-Verhältnisse, da durch diese einige Daten entfernt werden, der Inhalt jedoch auf das Relevante beschränkt wird.

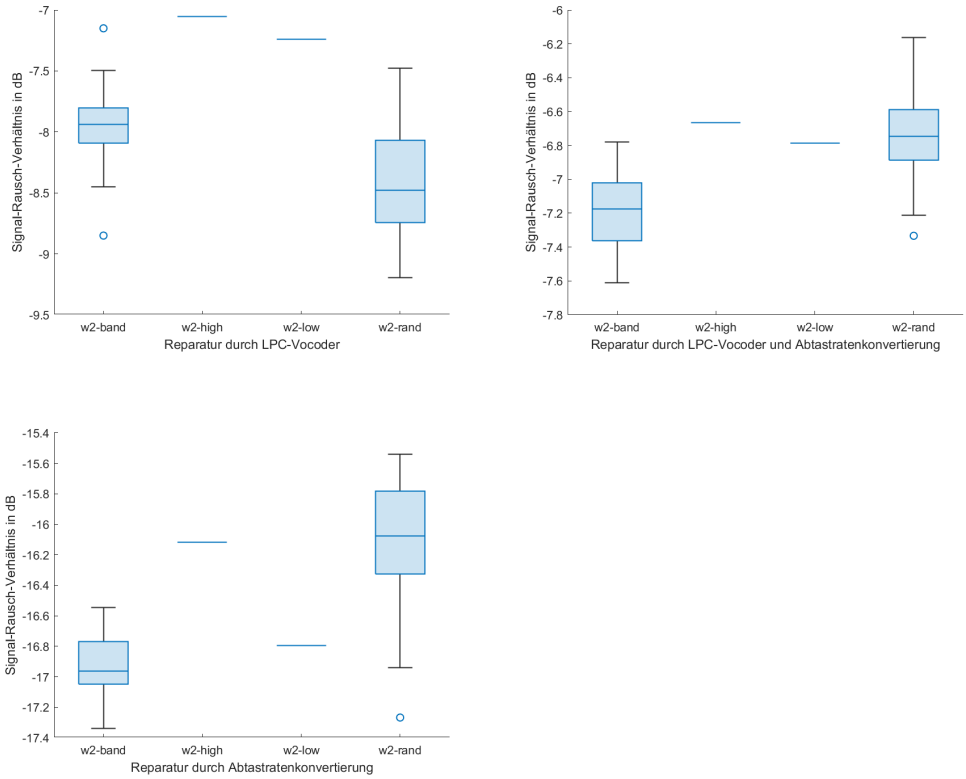

Abbildung 5.3.: Signal-Rauschverhältnisse aller reparierten Beispielsignale des Ursprungssignales „w2“

5.2. Transkription

Die im Folgenden ausgewerteten Metriken benötigen als Eingabedaten die transkribierten Texte der Signale, welche durch die beschriebenen Algorithmen gewonnen werden. Für jedes Ursprungssignal sowie für alle beschädigten und reparierten Signale aller 50 Durchläufe wurden die erkannten Texte zwischengespeichert und an die nachfolgenden Metriken übergeben. Somit wurden alle diese Metriken auf den exakt gleichen Datensatz angewendet. Die erhaltenen Ergebnisse werden im Folgenden für die Worterkennungsrate, die Wortfehlerrate sowie den Wortinformationsverlust dargestellt.

5.2.1. Worterkennungsrate

Als erste Metrik werden die Worterkennungsraten der transkribierten Texte verglichen. Für die aus den Ursprungssignalen erzeugten und anschließend reparierten Signale sind die erhaltenen Ergebnisse in Abbildung 5.4 dargestellt. Die Ergebnisse für jedes der einzelnen Ursprungssignale sind in Anhang A.2 zu finden.

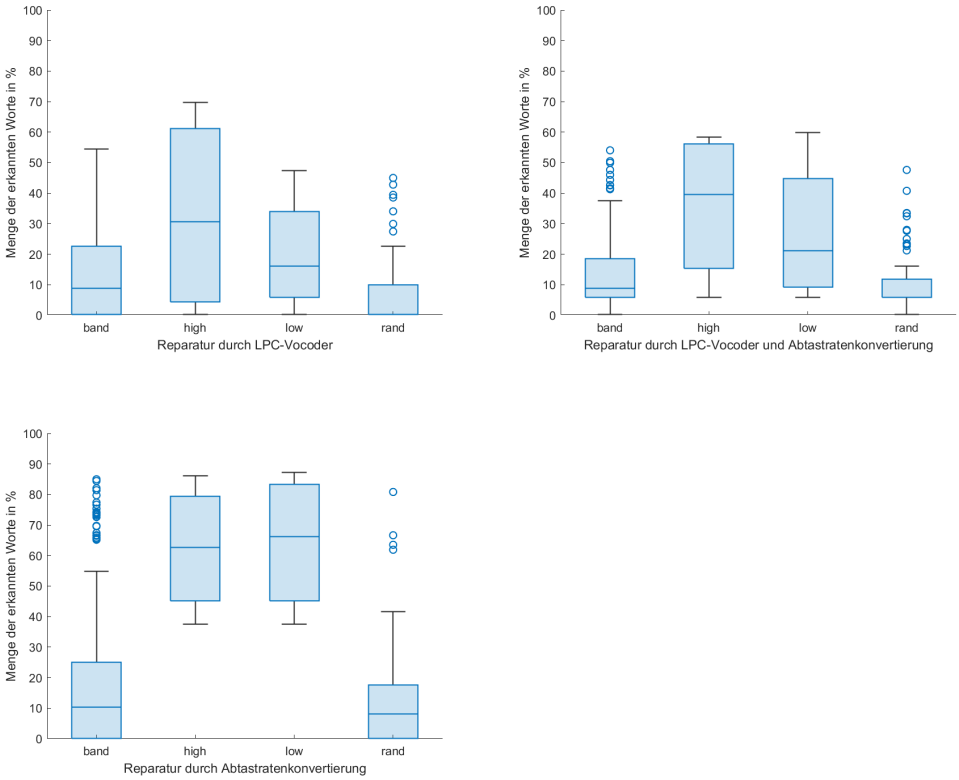

Abbildung 5.4.: Gesamtübersicht der Worterkennungsraten der transkribierten Texte für alle Ursprungssignale

In den Diagrammen in Abbildung 5.4 ist zunächst ersichtlich, dass für alle verwendeten Methoden eine große Streuung der Ergebnisse erzielt wird. Auf Grundlage der in den Kastendiagrammen enthaltenen Ausreißer der Methoden „band“ und „rand“ ist es möglich darauf zu schließen, dass durch den zufälligen Aspekt der verwendeten Methoden diese variierende Ergebnisse erzeugen. Durch Erhöhung der Versuchsdurchläufe würde sich die normale Menge der Kastendiagramme so erweitern, dass diese die hier als Ausreißer dargestellten Ergebnisse umfassen würde. Die Größe der Streuung der Ergebnisse ist am Beispiel der Reparatur durch Konvertieren der Abtastrate deutlich zu erkennen. Hierbei deckt die Ergebnisspanne fast den gesamten möglichen Wertebereich zwischen null und 100 Prozent ab. Die Ursache dieser star-

ken Schwankungen ist auf das zufällige Löschen zurückzuführen. Werden bei diesem einige oder sogar alle harmonischen Frequenzen des Signales entfernt, dann reduziert sich die Menge der Worte, welche durch das automatische Spracherkennungssystem erkannt werden. Dieses Verhalten wird erneut bei Betrachtung der Worterkennungsraten im Diagramm für die Reparatur durch die Kombination von LPC-Vocoder und Abtastratenkonvertierung deutlich. Während bei den anderen beiden Reparaturmethoden der Median der Worterkennungsraten der Methoden „high“ und „low“ recht nah beieinander ist, ist in diesem Diagramm ein erheblicher Unterschied sichtbar. Dieser ist auf das zuvor erklärte Verhalten zurückzuführen. Die auftretende starke Streuung für die Methoden, welche den oberen bzw. unteren Frequenzbereich entfernen („high“ bzw. „low“) ergibt sich durch betrachten der Worterkennungsraten für die Ursprungssignale „m2“ und „w1“. Die Ergebnisse für alle Signale, welche sich aus dem Ursprungssignal „m2“ ergeben sind in Abbildung 5.5 dargestellt. Die Ergebnisse für das Ursprungssignal „w1“ sind aufgrund ihrer Ähnlichkeit zum Ursprungssignal „m2“ in Anhang A.2 beigefügt. Betrachtet man die Ergebnisse in Abbildung 5.5 zeigt sich, dass für alle Signale aus diesen Ursprungssignalen eine sehr geringe Worterkennungsrate auftritt. Die Ursache für dieses Verhalten liegt darin, dass das automatische Spracherkennungssystem für diese Signale keine bzw. stark fehlerhafte Ausgaben ermittelt hat. Es handelt sich hierbei um die Vertreter der jeweiligen Sprecher mit kürzerem Text. Eine mögliche Ursache für das nicht Erkennen der gesprochenen Texte liegt in der Länge der Texte. Automatische Spracherkennungsysteme besitzen neben einer reinen Worterkennung auch eine Grammatikerkennung, welche für längere Sätze eine Korrektur des Ergebnisses vornehmen kann. Durch die Kürze der gesprochenen Texte in Kombination mit der geringen Texterkennung ergibt sich folglich das schlechte Ergebnis. Während für die Reparatur durch den LPC-Vocoder (Abbildung 5.5, oben links) eine Worterkennungsrate von null Prozent für jede Methode auftritt, erreicht die Reparatur durch die Reduktion der Abtastrate (Abbildung 5.5, unten links) für die Methoden „low“ und „high“ knapp unter 40 Prozent. Der Einfluss dieser Werte ist deutlich in der Gesamtübersicht in Abbildung 5.4 in Form der Minimalwerte in den dargestellten Kastendiagrammen zu sehen.

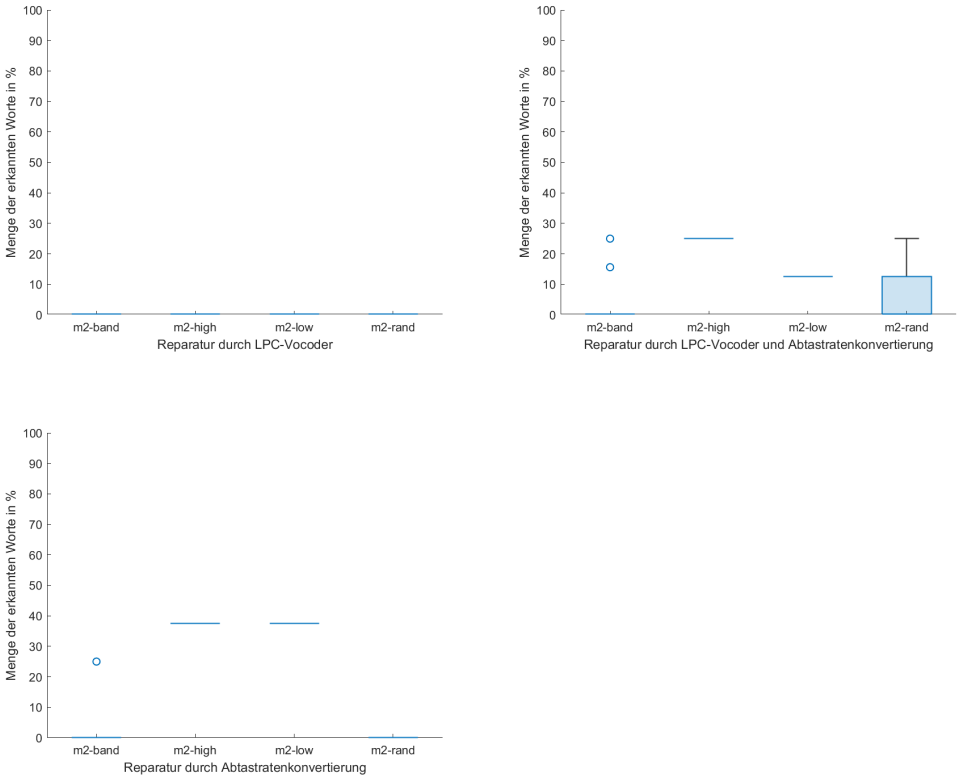

Abbildung 5.5.: Worterkennungsraten der transkribierten Texte für das Ursprungssignal „m2“

Vergleicht man die Reparaturmethoden der Gesamtübersicht in Abbildung 5.4 miteinander so wird direkt deutlich, dass das Kodieren der Signale zu einer recht niedrigen Worterkennungsrate und somit einem starken Verlust der Erkennbarkeit führt. Betrachtet man nun das Diagramm für die Kombination des LPC-Vocoders mit der Abtastratenkonvertierung (Abbildung 5.4, oben rechts) wird deutlich, dass die anschließende Reduktion der Abtastrate einen weiteren Abfall der Worterkennungsrate bewirkt und somit der richtig erkannte Textanteil weiter sinkt. Wendet man jedoch die Abtastratenkonvertierung ohne vorherige Kodierung des Signales an, wird anhand der Worterkennungsraten deutlich, dass für alle Methoden mehr als 80 Prozent des Originaltextes im besten Fall wiedererkannt wurden. Für die Methoden „high“ und „low“ ergeben sich außerdem sehr gute Minimalwerte. Die Worterkennungsraten der zufallsbasierten Methoden schwanken hierbei jedoch stark. Während der Median der Ergenisse dieser Methoden recht niedrig ist, liegt er dennoch oberhalb des Medians der anderen Reparaturverfahren. Des Weiteren erreichen die Spitzenwerte der Methoden ebenfalls Werte von über 80 Prozent und liegen somit weit über den Spitzenwerten der korrespondierenden Methoden der anderen Reparaturverfahren. Aus dieser Metrik ergibt sich somit, dass durch den Einsatz

der Abtastratenkonvertierung die größte Menge an richtig erkannten Worten erzielt werden kann. Dies wiederum lässt darauf schließen, dass im Kontext dieser Metrik durch die große Menge an richtig erkanntem Originaltext, diese Reparaturmethode die Signale mit der besten Qualität erzeugt.

Betrachtet man nun die in Abbildung 5.6 dargestellten Worterkennungsraten der beschädigten Signale vor dem Einsatz der Reparaturverfahren werden die Auswirkungen der Beschädigungen auf die Worterkennungsrate deutlich. Vergleicht man nun die Werte vor der Reparatur mit den Ergebnissen nach der Reparatur so wird deutlich, dass für das beste der drei Reparaturverfahren nur eine geringe Verringerung der Worterkennungsrate sichtbar ist. Im Vergleich mit den anderen Reparaturverfahren findet eine stärkere Reduktion der Worterkennungsrate statt.

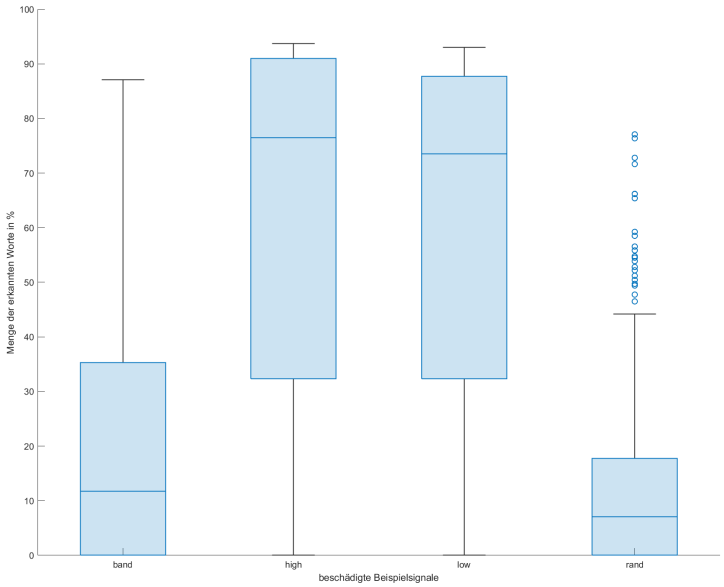

Abbildung 5.6.: Gesamtübersicht der Worterkennungsraten der transkribierten Texte aller Ursprungssignale vor der Reparatur

Dies lässt darauf schließen, dass keines der Reparaturverfahren zu einer grundsätzlichen Verbesserung der Sprachqualität des Signales beiträgt. Vergleicht man die Ergebnisse vor der Reparatur des Signales „m2“ mit den zuvor in Abbildung 5.5 dargestellten Worterkennungsraten nach den Reparaturverfahren zeigt sich jedoch eine Besonderheit. Abbildung 5.7 zeigt, dass für die aus dem Ursprungssignal „m2“ generierten Beispielsignale kein Text erkannt wurde. Betrachtet man nun die in Abbildung 5.5 dargestellten Reparaturverfahren wird deutlich, dass durch die Reduktion der Abtastrate eine deutliche Erhöhung der Worterkennungsrate auftritt. Im Kontext dieser Metrik zeigt sich demnach, dass die Reduktion der Abtastrate in bestimmten Fällen zu einer Verbesserung der erkannten Texte führt.

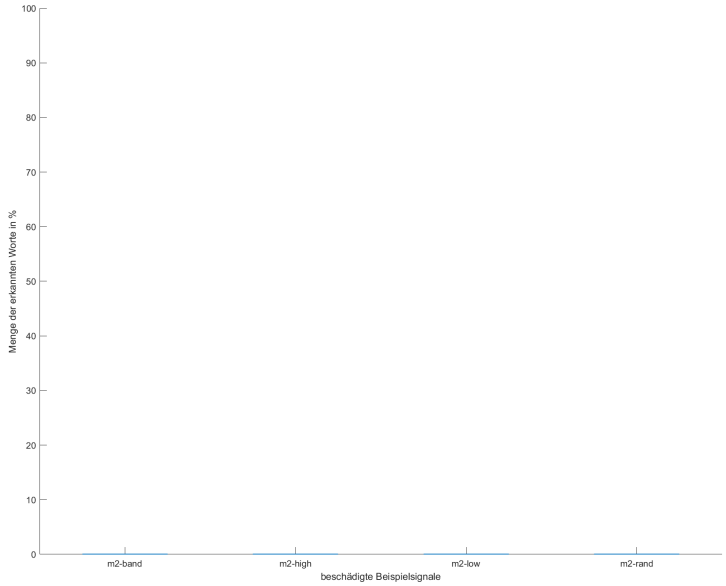

Abbildung 5.7.: Worterkennungsraten der transkribierten Texte vor der Reparatur für das Ursprungssignal „m2“

5.2.2. Wortfehlerrate

Neben der Worterkennungsrate wird nun die Wortfehlerrate für die Signale berechnet. Der in den Diagrammen beschriebene Wert beschreibt die fehlerhaften Worte in Prozent. In Abbildung 5.8 sind die Wortfehlerraten für alle Ursprungssignale dargestellt. Die Diagramme der einzelnen Urpsrungssignale sind in Anhang A.3 beigefügt. Betrachtet man zunächst allgemein die Werte der Reparaturverfahren in Abbildung 5.8 wird deutlich, dass viele der berechneten Wortfehlerraten in der oberen Hälfte des Wertebereiches liegen. Es handelt sich hierbei jedoch um Fehlerraten, sodass ein hoher Wert schlechter als ein niedrigerer ist. Des Weiteren ist eine große Anzahl von Ausreißern in den Methoden „band“ und „rand“ auffällig. Diese Methoden besitzen eine zufallsbasierte Komponenten, sodass durch Erhöhen der Versuchsanzahl die Ausreißer Bestandteil der normalen Menge des Kastandiagrammes würden. Es zeigt sich zudem eine starke Streuung der Ergebnisse für die Methoden „high“ und „low“. Diese Methoden löschen jeweils einen festen Frequenzbereich und sind somit nicht durch Zufall veränderlich. Die auftretende Varianz der Ergebnisse ergibt sich durch die unterschiedlich gute Erkennung der einzelnen Beispielsignale. Die schlechtesten Werte ergeben sich hierbei aus den Beispielsignalen der Ursprungssignale „m2“ bzw. „w1“ (siehe Anhang A.3).

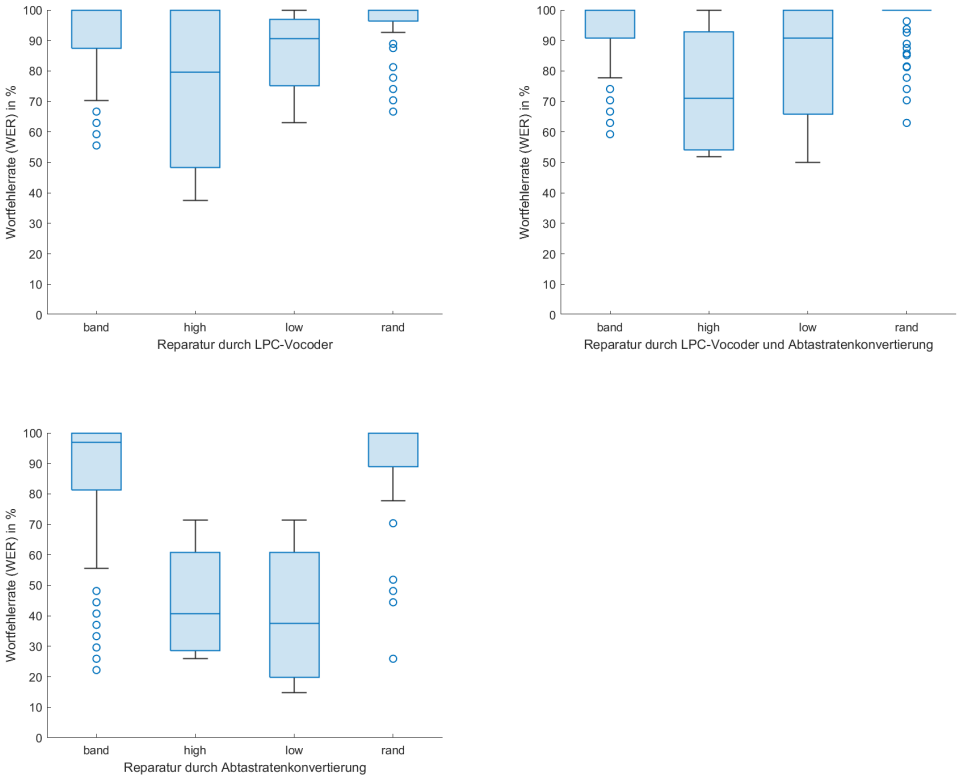

Abbildung 5.8.: Gesamtübersicht der Wortfehlerraten der transkribierten Texte aller Ursprungssignale

Aus den Diagrammen ist es möglich die drei Reparaturverfahren im Kontext dieser Metrik miteinander zu vergleichen. Es zeigt sich, dass die Reparatur durch Kodierung des Signals mittels LPC-Vocoder im linken oberen Diagramm in Abbildung 5.8 Spitzenwerte von circa 38 Prozent für die Methode „high“ besitzt. Im Vergleich mit der Reparatur durch kodieren und anschließende Abtastratenkonvertierung zeigt sich eine Erhöhung der Wortfehlerraten in allen Aspekten. Ausgenommen ist hierbei der schlechteste Wert, welcher für beide Reparaturverfahren für jede Erzeugungsmethode 100 Prozent ist. Im Diagramm für die Reparatur durch die Abtastratenkonvertierung ohne vorherige Kodierung des Signals zeigt sich jedoch eine deutliche Verbesserung im vergleich zu beiden anderen Verfahren. Während die schlechtesten Werte der zufallsbasierten Methoden „rand“ und „band“ immernoch 100 Prozen sind, heben sich die Methoden „high“ und „low“ stark vom bisherigen Gesamtbild ab. Der Median sowie die Spitzenwerte sind ebenfalls höher als die korrespondierenden Werte aller anderen Reparaturverfahren.

Vergleicht man die Werte nach der Reparatur nun mit den in Abbildung 5.9 dargestellten Wortfehlerraten der Signale vor der Reparatur, zeigt sich das für alle Reparaturverfahren eine Erhöhung der Fehlerrate und somit eine Verschlechterung

der Qualität im Kontext dieser Metrik stattgefunden hat. Während durch die Reduktion der Abtastrate (Abbildung 5.8, links unten) meist nur eine sehr geringe Veränderung zu sehen ist, ist die Erhöhung der Fehlerrate für die beiden anderen Verfahren deutlich erkennbar.

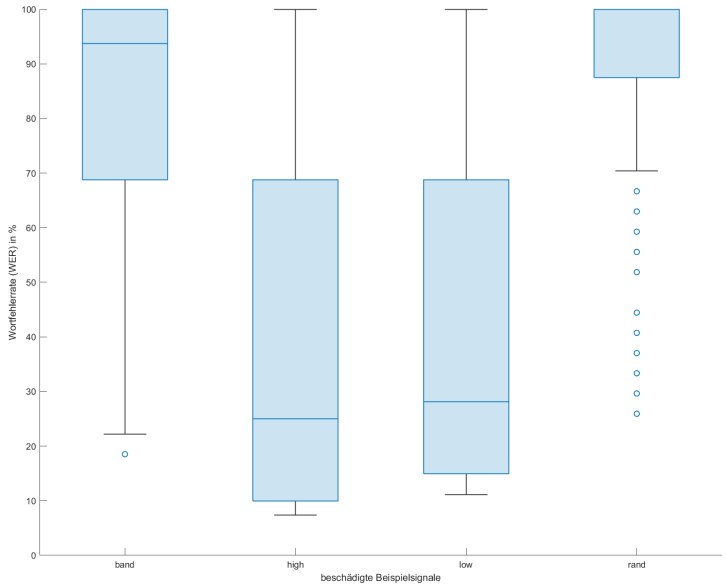

Abbildung 5.9.: Gesamtübersicht der Wortfehlerraten der transkribierten Texte aller Ursprungssignale vor der Reparatur

Betrachtet man auch für diese Metrik die Ergebnisse für das Ursprungssignal „m2“ vor und nach der Reparatur durch die Abtastratenkonvertierung (siehe Anhang A.3) zeigt sich die fehlende Erkennung der Signale durch das automatische Spracherkennungssystem vor der Reparatur durch eine Fehlerrate von 100 Prozent für jede der Methoden. Betrachtet man nun die in Abbildung 5.10 dargestellten Diagramme nach den Reparaturverfahren wird erneut eine Verbesserung der Erkennbarkeit des Signales sichtbar. Hierbei kann durch die Abtastratenkonvertierung allein, als auch im Einsatz nach der Kodierung des Signals durch den LPC-Vocoder eine Reduktion der Wortfehlerrate für zuvor vollständig nicht erkennbare Signale festgestellt werden. Durch dieses Verhalten und dem nur geringen Einfluss auf die Wortfehlerrate erkennbarer Signale bietet die Reparatur durch Abtastratenkonvertierung viel Potential.

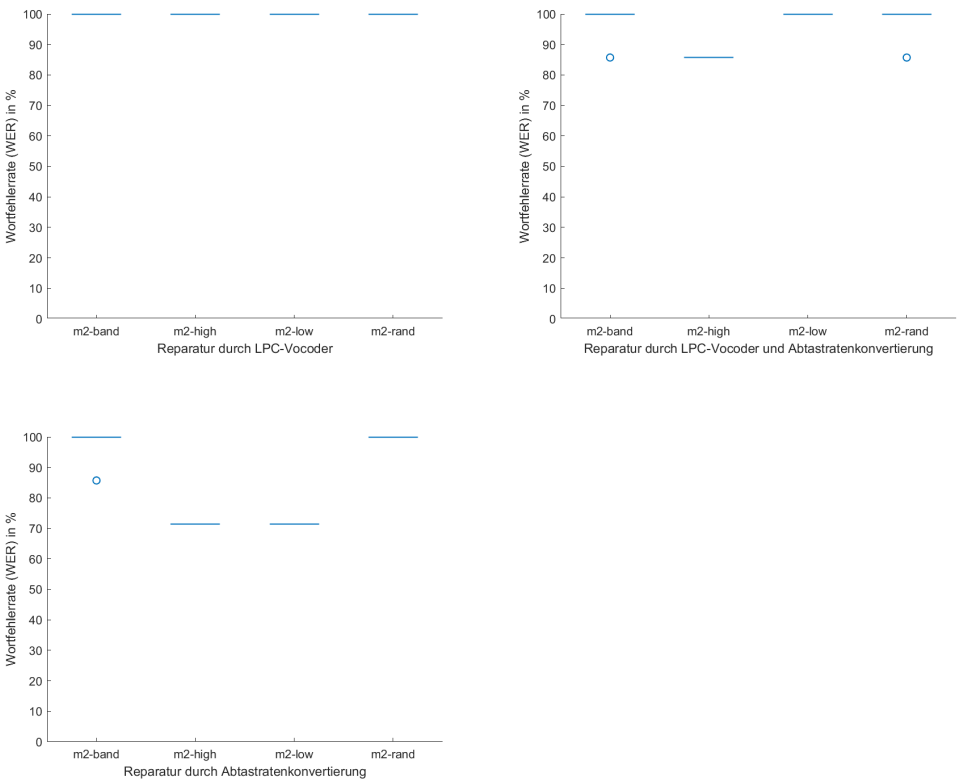

Abbildung 5.10.: Gesamtübersicht der Wortfehlerraten der transkribierten Texte der Ursprungssignale

5.2.3. Wortinformationsverlust

Die letzte Metrik, welche auf die durch das automatische Spracherkennungssystem generierten Texte angewendet wird, ist der Wortinformationsverlust. Die Gesamtübersicht für alle Ursprungssignale ist in Abbildung 5.11 dargestellt. Bei allgemeiner Betrachtung der Diagramme fällt zunächst die Ähnlichkeit mit der Gesamtübersicht der Wortfehlerraten (siehe Abbildung 5.8) auf. Die Wortfehlerrate gibt keinerlei Auskunft über den Informationsgehalt der verglichenen Texte. Durch die Berechnung der Wortinformationsverluste wird dies jedoch ermöglicht. Die im Folgenden nicht dargestellten Einzeldiagramme der Ursprungssignale sind in Anhang A.4 beigefügt.

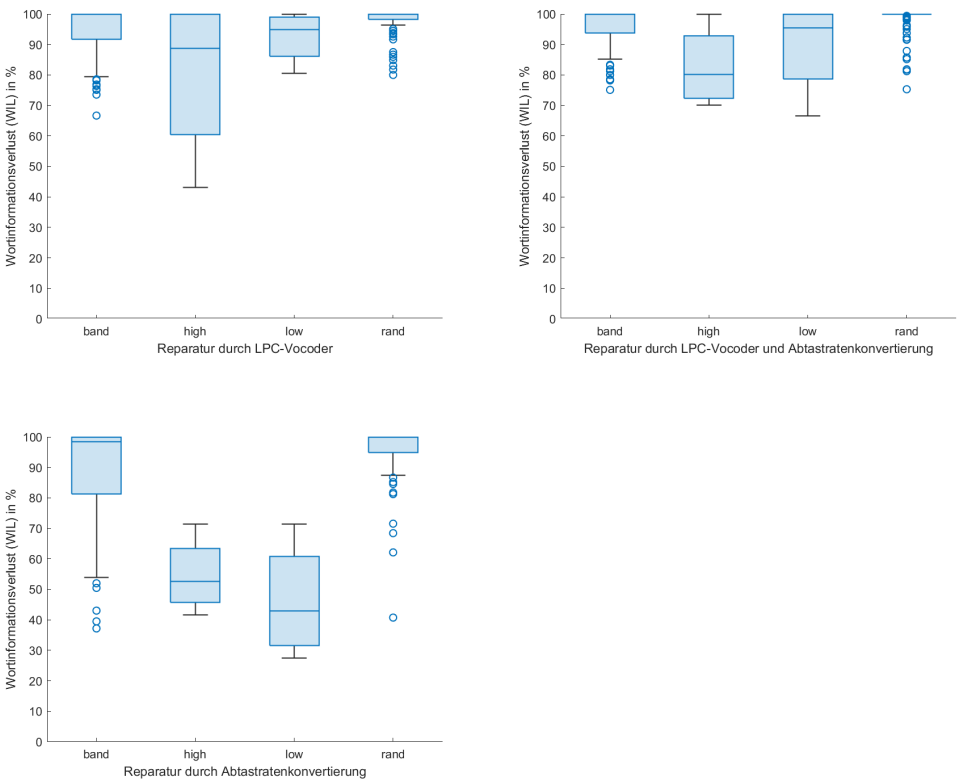

Abbildung 5.11.: Gesamtübersicht des Wortinformationsverlustes der transkribierten Texte für alle Ursprungssignale

Analog zu den vorherigen beiden Metriken, welche auf die transkribierten Texte angewendet wurden, ist es möglich die Ursache der Ausreißer auf die zufallsbasierten Komponenten der Methoden „band“ und „rand“ zurückzuführen. Durch eine Erhöhung der Versuchsdurchläufe würde die Anzahl der Werte im Bereich der aktuellen Ausreißer steigen, sodass diese in die normale Menge zwischen beiden Whiskern eingegliedert würden. Es zeigt sich weiterhin, dass für die Reparatur durch das Kodieren mittels LPC-Vocoder (Abbildung 5.11, links oben) sowie die Reparatur durch Kodierung des Signals und Reduktion der Abtastrate (Abbildung 5.11, rechts oben) für viele Signale zu einem Wortinformationsverlust von 100 Prozent führen. Im unteren linken Diagramm der Reparatur durch die Abtastratenkonvertierung zeigt sich jedoch, vorallem für die Methoden „high“ und „low“ eine deutliche Verbesserung der Werte.

Im Vergleich der Werte nach dem Einsatz der jeweiligen Reparaturverfahren mit den in Abbildung 5.12 dargestellten Werten kann der Einfluss der Reparaturverfahren gezeigt werden. Für die Reparatur durch Kodieren des Signals mittels LPC-Vocoder zeigt sich eine deutliche Erhöhung der Werte für jede der verwendeten Methoden. Durch die anschließende Abtastratenkonvertierung steigt der Wortin-

formationsverlust weiter an. Es lässt sich foglich sagen, dass diese beiden Verfahren einen stark negativen Einfluss auf die Signalerkennung und somit die Sprachqualität im Kontext dieser Metrik haben. Betrachtet man die Reparatur, bei welcher nur die Abtastrate reduziert wurde, so zeigen sich mehrere Sachverhalte auf. Es zeigt sich, dass nur eine geringe Erhöhung der Wortinformationsverluste auftritt. Des Weiteren zeigt sich, dass für die Methoden „high“ und „low“, welche den oberen bzw. unteren Frequenzbereich entfernen, eine Verbesserung im schlechtesten Fall auftritt. Signale, welche vor der Reparatur nicht von der automatischen Spracherkennung erkannt wurden, können nach der Abtastratenkonvertierung zumindest teilweise wieder erkannt werden. Dies tritt beispielsweise für das Ursprungssignal „m2“ auf (siehe Anhang A.4). Es zeigt sich weiterhin, dass durch die Abtastratenreduktion die Streuung der Daten reduziert und der Median zum Mittelwert verschoben wird, sodass eine symmetrische Verteilung der Werte aufzutreten scheint.

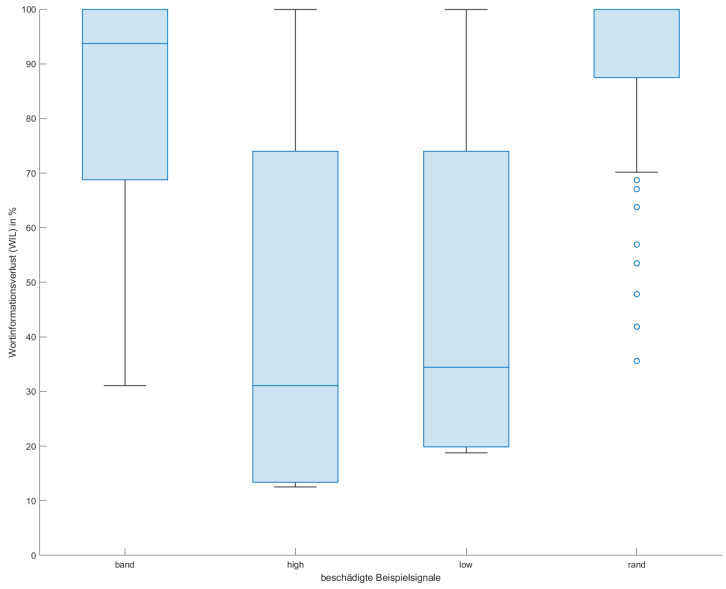

Abbildung 5.12.: Gesamtübersicht des Wortinformationsverlustes der transkribierten Texte aller Ursprungssignale vor der Reparatur

5.3. RMSE

Durch die in Abschnitt 4.4.6 implementierten Algorithmen wurden die Diagramme in Abbildung 5.13 für alle Ursprungssignale erzeugt. Diese stellen für jedes Ursprungssignal und jede Erzeugungsmethode die normalisierten RMSE-Werte im Bezug zum Ursprungssignal dar. Der Wert eines jeden Ursprungssignales im Vergleich mit sich selber ist null, da im Vergleich eines Signales mit sich selbst keine Differenz zwischen beiden Signalen auftritt.

Die in Abbildung 5.13 dargestellten Ergebnisse für alle Ursprungssignale zeigen, dass der Median für alle Methoden nahezu gleich ist. Eine etwas größere Abweichung ist hierbei im Diagramm der Reparatur durch Abtastratenkonvertierung für das zufällige Löschen der Frequenzen ersichtlich, bei welcher der Median leicht verringert im Vergleich zu den anderen Methoden ist. Des Weiteren ist eine starke Ähnlichkeit in der Streuung der Werte sichtbar.

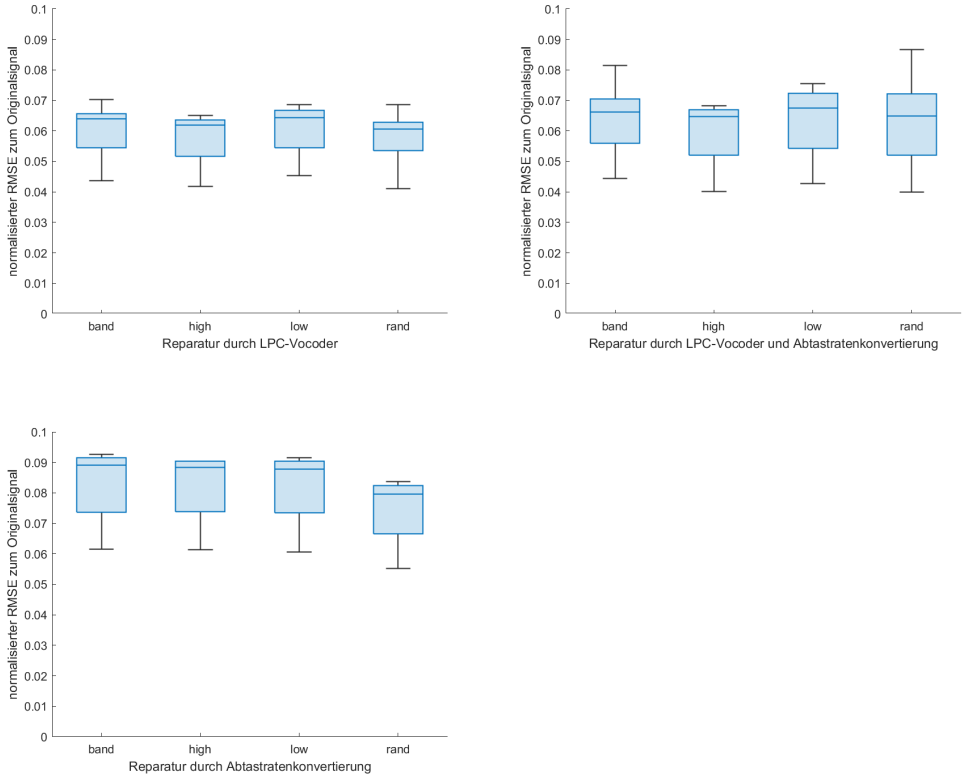

Abbildung 5.13.: Gesamtübersicht der normalisierten RMSE Werte für alle Ursprungssignale

Aus dem in den Diagrammen dargestellten Werten ergibt sich weiterhin, dass keines der Reparaturverfahren ein vom Ursprungssignal stark abweichendes Signal erzeugt. Dies ergibt sich, da der normalisierte RMSE einen maximalen Wert von eins besitzen kann. Alle in den Diagrammen der Reparaturverfahren dargestellten Werte liegen jedoch unter einem Wert von 0,1 und sind somit recht niedrig. Im Vergleich der einzelnen Reparaturverfahren untereinander zeigt sich, dass die Reparatur durch den Einsatz des LPC-Vocoders die geringsten RMSE Werte aufweist. Durch die anschließende Abtastratenkonvertierung (siehe Abbildung 5.13, links oben) wird deutlich, dass diese nur geringen Einfluss auf die Werte in Form einer Erhöhung sowie einer Vergrößerung der Streuung hat. Für die Reparatur durch die Abtastra-

tenkonvertierung im unteren linken Diagramm wurden unter allen Methoden die höchsten Werte ermittelt.

Durch den Vergleich der Werte nach der Reparatur mit den Werten vor dem Einsatz der Verfahren (siehe Abbildung 5.14) kann gezeigt werden, ob ein Signal vor oder nach dem Einsatz eines Reparaturverfahrens dem Ursprungssignal ähnlicher ist. Vergleicht man zunächst die Reparatur durch die Reduktion der Abtastrate so zeigt sich eine starke Ähnlichkeit der Werte. Bei genauerer Betrachtung wird jedoch deutlich, dass der Median für jede der vier Methoden vor der Reparatur niedriger ist als nach der Reparatur. Folglich führt das Reparaturverfahren zu einer Erhöhung des RMSE und die reparierten Signale sind dem Ursprungssignal weniger ähnlich als vor der Reparatur. Der Vergleich der beiden anderen Reparaturverfahren zeigt jedoch deutlich, dass die berechneten Werte nach der Reparatur leicht unter den Werten vor dem Einsatz der jeweiligen Verfahren liegt. Dies bedeutet, dass die Kodierung mittels LPC-Vocoder sowie die Kodierung und anschließende Reduktion der Abtastrate ein Signal erzeugt, welches dem Ursprungssignal ähnlicher ist als vor diesen.

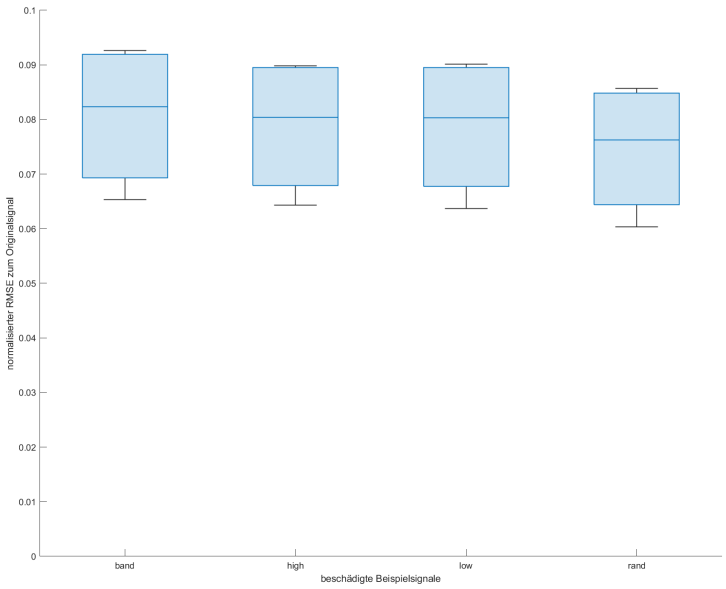

Abbildung 5.14.: Gesamtübersicht der normalisierten RMSE Werte aller Ursprungssignale vor der Reparatur

6. Diskussion

Im Folgenden werden nun die Ergebnisse der Metriken im Kontext der aufgestellten Hypothesen betrachtet und verglichen. Die Hypothesen H1 und H2 beschäftigen sich hierbei zunächst mit der Veränderung der Sprachqualität durch die drei verwendeten Reparaturverfahren. Die Ergebnisse der ersten Metrik der Signal-Rausch-Verhältnisse zeigte auf, dass für die Reparatur durch LPC-Vocoder und anschließendem Reduzieren der Abtastrate das höchste Signal-Rausch-Verhältnis und somit die größte Energiereproduktion der Signale erreicht werden kann. Die Reparatur durch reines konvertieren der Abtastrate führte zur geringesten Veränderung des Verhältnisses und somit zur geringsten Energiereproduktion. Vergleicht man nun die so gewonnenen Erkenntnisse mit den Ergebnissen der drei Metriken, welche zur Analyse der transkribierten Texte eingesetzt wurden, so zeigt sich ein Unterschied. Die Metriken der Worterkennungsrate, Wortfehlerrate und der Wortinformationsverlust zeigten eindeutig, dass für jede der drei geprüften Reparaturverfahren für alle Methoden im allgemeinen Fall eine Verschlechterung der Ergebnisse im Vergleich zu den Signalen vor dem Einsatz der Reparaturverfahren führt. Dies wiederum lässt darauf schließen, dass keines der Verfahren eine Verbesserung der Erkennbarkeit durch das automatische Spracherkennungssystem ermöglicht. Die Erkennbarkeit durch ein automatisches Spracherkennungssystem kann hierbei direkt mit der Sprachqualität des Signales verglichen werden, da sich eine hohe Sprachqualität durch eine hohe Erkennbarkeit der gesprochenen Texte auszeichnet. Der auftretende Unterschied der Ergebnisse der Metriken wird somit deutlich. In den Ergebnissen der SNR war die Reparatur durch LPC-Vocoder und anschließende Abtastratenkonvertierung das beste der drei Reparaturverfahren darstellte. Im Kontext der Worterkennungsrate, WER sowie WIL ist dieses jedoch das schlechteste der drei Verfahren, welches zur stärksten Minderung der Erkennbarkeit der gesprochenen Texte führt. Im Gegensatz dazu wurde die Reparatur durch das ausschließliche Reduzieren der Abtastrate in den Ergebnisse der SNR als schlechtestes der drei Verfahren eingestuft, da es die geringsten Änderungen der SNR aufwies, als bestes Verfahren im Kontext der anderen Metriken aufgefasst. Die Gemeinsamkeit der Metriken ist hierbei, dass in der Analyse der transkribierten Texte, dass Verfahren ebenfalls den geringsten Einfluss auf die Ergebnisse hatte. Dies führt jedoch im Falle dieser Metriken dazu, dass die Abtastratenkonvertierung die besten Ergebnisse aller geprüften Verfahren ergibt. Das Verfahren hat weiterhin eine besondere Eigenschaft für Signale, welche durch das automatische Spracherkennungssystem vollständig nicht erkannt werden konnten. Es zeigt sich, dass die Reparatur durch Reduktion der Abtastrate für diese Fälle einen positiven Einfluss auf die Erkennbarkeit haben kann. Zur Klärung der Hypothesen H1 und H2 werden die Ergebnisse der SNR mit den Ergebnissen der

drei Metriken zur Analyse der transkribierten Texte abgewogen. Das Signal-Rausch-Verhältnis gibt hierbei Auskunft über die technische Qualität während die anderen Metriken die Erkennbarkeit des Signales zeigen. Für die Sprachqualität eines Signales ist hierbei der Informationsgehalt der gesprochenen Texte von größter Relevanz. Folglich werden die Metriken der Worterkennungsrate, WER und WIL für die Klärung der Hypothesen mehr gewichtet als das Signal-Rausch-Verhältnis. Unter diesen Erkenntnissen kann die Hypothese H2 vollständig abgelehnt werden. Die Hypothese H1 umfasst hierbei die Reparatur durch LPC-Vocoder als auch die Reparatur durch Abtastratenkonvertierung. Diese Hypothese kann teilweise angenommen werden, da nur die Abtastratenkonvertierung in speziellen Fällen einen positiven Einfluss und im allgemeinen Fall einen minimal negativen Einfluss aufweist. Die Reparatur durch LPC-Vocoder hingegen wies in allen Metriken nur mittelmäßige Ergebnisse auf und eignet sich somit nicht als Reparaturverfahren.

Zur Klärung der Hypothesen H3 und H4 sollen die Ähnlichkeiten der Signale durch die Berechnung des RMSE zwischen diesen ermittelt werden. Die Ergebnisse zeigten zunächst eindeutig die bereits in den vorherigen Metriken festgestellte Ähnlichkeit des durch die Abtastratenkonvertierung reparierten Signales mit dem nicht reparierten Signal. Des Weiteren ergaben die Daten, dass die Reparatur durch LPC-Vocoder mit und ohne anschließender Abtastratenkonvertierung zu einer Verringerung des RMSE zum Ursprungssignal im Vergleich zum nicht reparierten Signal führt. Die Metrik ergab somit, dass diese beiden Verfahren ein Signal erzeugen, welches dem Ursprungssignal ähnlicher ist als vor der Anwendung dieser. Die Ähnlichkeit der Signale kann beispielsweise jedoch auch durch Betrachtung der erkannten Worte gezeigt werden. Für jedes der Ursprungssignale wurden 100 Prozent der Worte richtig erkannt, sodass folglich eine Erhöhung der erkannten Worte nach der Reparatur im Vergleich zu davor ebenfalls ein ähnlicheres Signal im Kontext dieser Betrachtungsweise erzeugen würde. Der RMSE ist hierbei jedoch die deutlich genauere Metrik und wird daher vorrangig betrachtet. Folglich ergibt sich, dass die Hypothese H4 akzeptiert wird. Die Hypothese H3 wiederum kann nur teilweise akzeptiert werden, da die Reparatur durch LPC-Vocoder die Bedingung der Hypothese erfüllt, die reine Abtastratenkonvertierung jedoch zu einer Erhöhung des normalisierten RMSE führt und somit nicht das gewünschte Kriterium erfüllt.

7. Zusammenfassung

Selektives Hören beschreibt die menschliche Fähigkeit aus einer Vielzahl von Geräuschen eines gezielt besser wahrzunehmen und die wahrgenommene Lautstärke der anderen zu reduzieren. Das Gebiet der computergestützen auditiven Szenenanalyse beschäftigt sich mit der Realisierung dieser Fähigkeit unter Einsatz von Computern. Eine solche Realisierung findet in verschiedenen Schritten statt, dabei findet die konkrete Trennung der Signale durch eine Zeit-Frequenz-Maskierung statt. Hierbei werden Signalabschnitte im Zeit- und Frequenzbereich aus dem Grundsignal herausgefiltert und in einzelne Signale getrennt. Jedes der getrennten Signale beinhaltet dann nur noch eine Stimme. Durch das begrenzte menschliche Sprachfrequenzspektrum ist es jedoch nicht unwahrscheinlich, dass die Stimmen zweier Sprecher zum selben Zeitpunkt überlappende Frequenzen besitzen. Durch die Zeit-Frequenz-Maskierung kann dieses Segment jedoch nur einem der Sprecher zugeordnet werden, sodass fehlerbehaftete Ausgaben entstehen.

Im Zuge dieser Arbeit wurden Verfahren auf ihre Tauglichkeit als Verfahren zur Verbesserung der Sprachqualität der, durch ein CASA-System generierten, beschädigten Signale überprüft. Konkret wurden ein LPC-Vocoder, welcher durch eine Bibliothek realisiert wurde sowie die Reduktion der Taktrate des Signales überprüft. Der Einsatz des LPC-Vocoders ergab sich durch dessen Funktionsweise. Das zu kodierende Signal wird hierbei im Analyseschritt zunächst stark komprimiert. Anschließend wird im Syntheseschritt des Verfahrens auf Basis des komprimierten Signales ein vollständig neues Signal synthetisiert. Das Verfahren fand zudem bereits praktischen Einsatz als Komprimierungsverfahren in den Anfängen der Telefonie, es wurde hiermit die zu übertragene Datenmenge für die Übertragung der Gespräche reduziert. Die Reduktion der Taktrate schien als geeignetes Verfahren, da durch die Veränderung der Taktrate eine Änderung des Frequenzspektrums des Signales auftritt. Mit den gewählten Verfahren sollten nun die aufgestellten Hypothesen und somit die Tauglichkeit der Verfahren als Reparaturverfahren überprüft werden.

Es wurden die Hypothesen aufgestellt, dass die aufgestellten Verfahren zu einer Verbesserung der Sprachqualität führen bzw. diese eine dem Ursprungssignal ähnlichere Ausgabe erzeugen. Für die verwendeten Metriken der SNR, Worterkennungsrate, WER, WIL und RMSE wurden die Werte der durch die Verfahren reparierten Signale mit den Signalen vor der Reparatur verglichen. Aus dem Vergleich der Signal-Rausch-Verhältnisse der Signale sowie der durch die IBM-Cloud transkribierten Texte war es möglich eine Aussage über die Veränderung der Sprachqualität der Signale zu treffen. Es ergab sich das der Einsatz der Abtastratenkonvertierung in einigen bestimmten Fällen eine positive Wirkung auf die Sprachqualität des Signal haben kann, im allgemeinen Fall jedoch eine minimal negative Wirkung besitzt. Der

Vergleich der Signale auf Ähnlichkeit wurde mittels RMSE realisiert. Die Analyse ergab hierbei, dass die Reparatur durch LPC-Vocoder sowie wahlweise anschließende Abtastratenkonvertierung zu einer Reduktion des RMSE zum Ursprungssignal im Vergleich zu den Werten vor der Reparatur aufweist. Aus der Analyse der verwendeten Metriken zeigt sich, dass die Reduktion der Abtastrate Potential für die Reparatur der beschädigten Signale hat, sofern diese starke Sprachqualitätsverluste aufweisen.

Die geprüften Reparaturverfahren zeigen hierbei jedoch nur einen kleinen Ausschnitt aus der Menge der möglichen Algorithmen. Eine weitere Forschung unter Einsatz eines modifizierten Vocoders sowie der Einsatz anderer Vocoder ist als weiterführende Arbeiten in diesem Gebiet denkbar. Des Weiteren könnten vollständig andere Ansätze zur Reparatur der Signale, beispielsweise auf Grundlage von Deep Learning implementiert und geprüft werden.

Literaturverzeichnis

[1] Arikan, E.: Channel polarization: A method for constructing capacity-achieving codes. In: 2008 IEEE International Symposium on Information Theory. pp. 1173–1177 (July 2008)

[2] Berlekamp, E.R.: Nonbinary bch decoding. IEEE Transactions on Information Theory 14, 242 (1968)

[3] Berrou, C., Glavieux, A., Thitimajshima, P.: Near shannon limit error-correcting coding and decoding: Turbo-codes. 1. In: Proceedings of ICC '93 - IEEE International Conference on Communications. vol. 2, pp. 1064–1070 vol.2 (May 1993)

[4] Bregman, A.S.: Auditory scene analysis: The perceptual organization of sound. MIT press (1994)

[5] Deller, J., Proakis, J., Hansen, J.: Discretetime processing of speech signals,"mcmillan publish (1993)

[6] Forney, G.D.: Concatenated codes. Tech. rep., Massachu-setts Institute of Technology (1996)

[7] GOLAY, M.J.E.: Notes on digital coding. Proc. IEEE 37, 657 (1949), https://ci.nii.ac.jp/naid/10018343533/en/

[8] Goldberg, R.: A practical handbook of speech coders. CRC press (2019)

[9] Gray, R.M.: A history of realtime digital speech on packet networks: Part II of linear predictive coding and the internet protocol. Foundations and Trends® in Signal Processing 3(4), 203–303 (2009), https://ee.stanford.edu/~gray/lpcip.pdf

[10] Hamming, R.W.: Error detecting and error correcting codes. The Bell System Technical Journal 29(2), 147–160 (April 1950)

[11] Harris, F.J.: Multirate signal processing for communication systems. River Publishers (2021)

[12] Horadam, K.J.: Hadamard Matrices and Their Applications. Princeton University Press (2006), https://www.ebook.de/de/product/5521019/k_j_horadam_hadamard_matrices_and_their_applications.html

[13] IBM Cloud Watson Docs: Abtastratenkonvertierung der spracherkennungsmodelle (12.07.2021), https://cloud.ibm.com/docs/speech-to-text-data?topic=speech-to-text-data-audio-formats#samplingRate

[14] Jeffress, L.A.: A place theory of sound localization. Journal of Comparative and Physiological Psychology 41(1), 35–39 (1948)

[15] Kegler, M., Beckmann, P., Cernak, M.: Deep speech inpainting of time-frequency masks. arXiv preprint arXiv:1910.09058 (2019)

[16] Llorach, G., Ohlenbusch, M.: Web-based vocoder with audio worklets and live input (2020), https://github.com/Web-based-vocoder/web-based-vocoder.github.io

[17] MacKay, D.J., Neal, R.M.: Near shannon limit performance of low density parity check codes. Electronics letters 33(6), 457–458 (1997)

[18] MacWilliams, F., Sloane, N.: The theory of error-correcting codes north-holland (1977)

[19] Markel, J.D., Gray, A.J.: Linear prediction of speech, vol. 12. Springer Science & Business Media (2013)

[20] Massey, J.L.: Shift-register synthesis and bch decoding. IEEE Transactions on Information Theory 15(1), 122–127 (1969)

[21] MATLAB Central File Exchange: Speech processing (04.07.2021), https://www.mathworks.com/matlabcentral/fileexchange/45321-lpc-vocoder

[22] Meddis, R., Hewitt, M.J.: Virtual pitch and phase sensitivity of a computer model of the auditory periphery. i: Pitch identification. The Journal of the Acoustical Society of America 89(6), 2866–2882 (jun 1991)

[23] Mohammadi, Seyed Hamidreza und Kain, A.: Voice conversion using deep neural networks with speaker-independent pre-training. In: 2014 IEEE Spoken Language Technology Workshop (SLT). pp. 19–23. IEEE (2014)

[24] Morris, A., Maier, V., Green, P.: From wer and ril to mer and wil: improved evaluation measures for connected speech recognition (10 2004)

[25] Nam Phamdo: Speech compression (26.08.2021), https://faculty.uml.edu/jweitzen/16.548/ClassNotes/SpeechCompression.htm

[26] Phamdo, N.: Speech compression (2001), http://faculty.uml.edu/jweitzen/16.548/ClassNotes/SpeechCompression_files/mmodel.gif

[27] Shannon, C.E.: A mathematical theory of communication. The Bell System Technical Journal 27(4), 623–656 (Oct 1948)

[28] Slaney, M., Lyon, R.F.: On the importance of time-a temporal representation of sound. Visual representations of speech signals 95116 (1993)

[29] Slaney, M.: An introduction to auditory model inversion. Tech. rep., Interval Technical Report IRC 1994-014 (1994)

[30] Stolte, N.: Rekursive codes mit der Plotkin-konstruktion und ihre decodierung. Ph.D. thesis, Technische Universität Darmstadt (2002)

[31] TSGR, E.: Lte: Evolved universal terrestrial radio access (e-utra). Multiplexing and channel coding (3GPP TS 36.212 version 10.3. 0 Release 10) ETSI TS 136(212), V10 (2011)

[32] Viterbi, A.: Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. IEEE Transactions on Information Theory 13(2), 260–269 (apr 1967)

[33] Wang, D., Brown, G.J.: Computational Auditory Scene Analysis: Principles, Algorithms, and Applications. Wiley-IEEE Press (2006)

[34] Williamson, J.: Simple lpc vocoder in python (2017), `https://github.com/johnhw/lpc_vocoder`

A. Anhang

A.1. Signal-Rausch-Verhältnisse

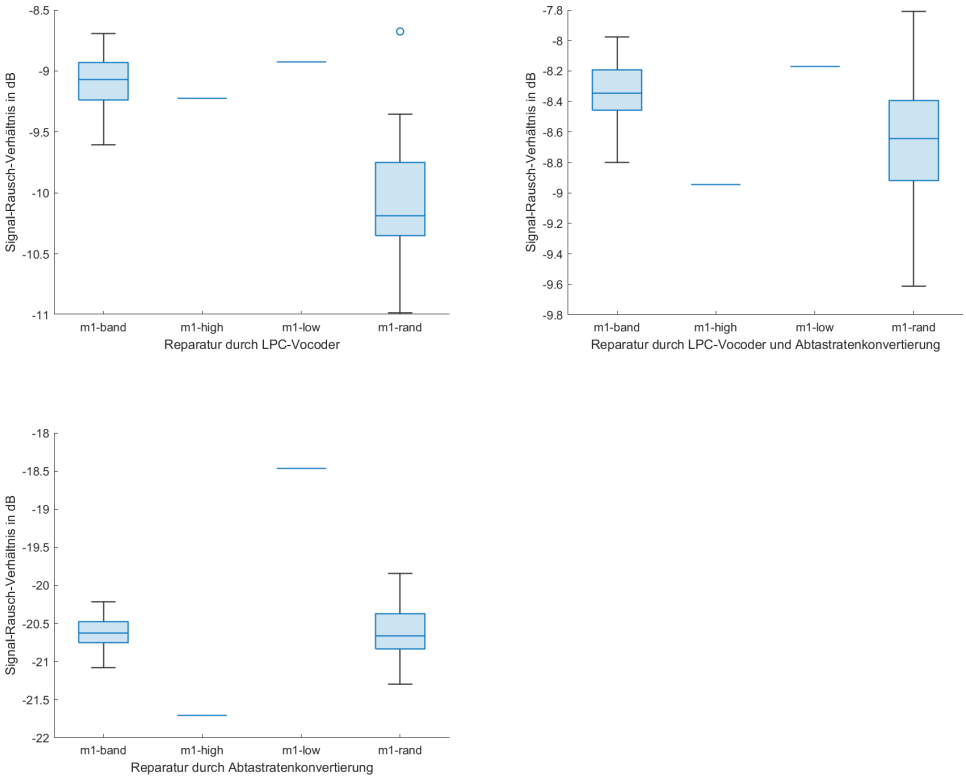

Abbildung A.1.: Signal-Rausch-Verhältnisse für das Ursprungssignal „m1“

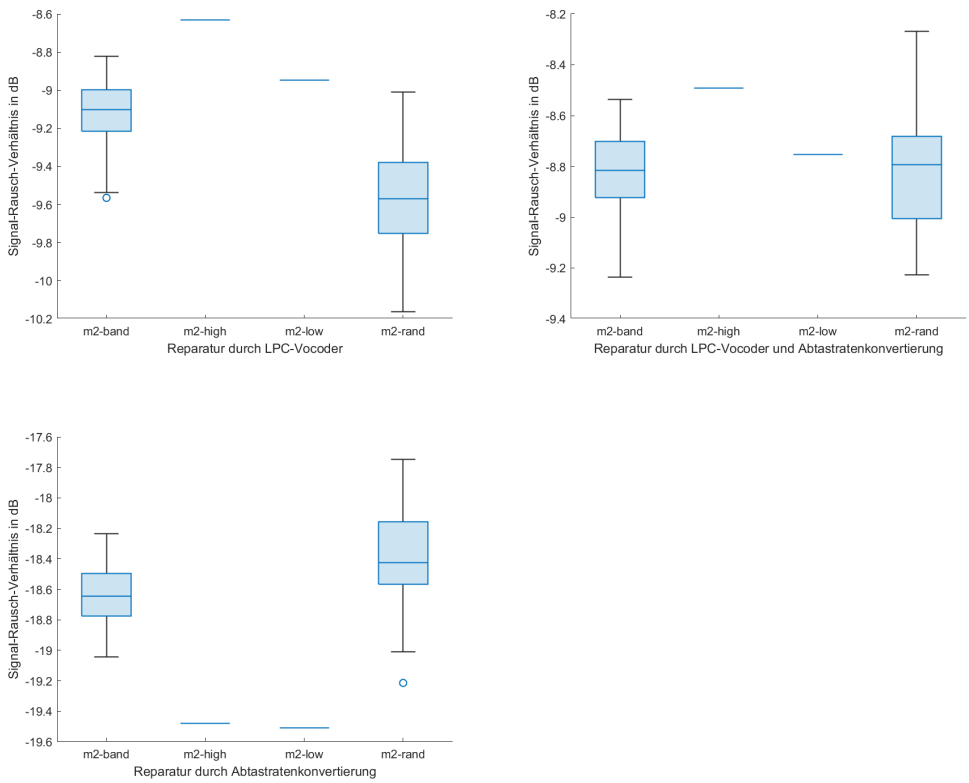

Abbildung A.2.: Signal-Rausch-Verhältnisse für das Ursprungssignal „m2“

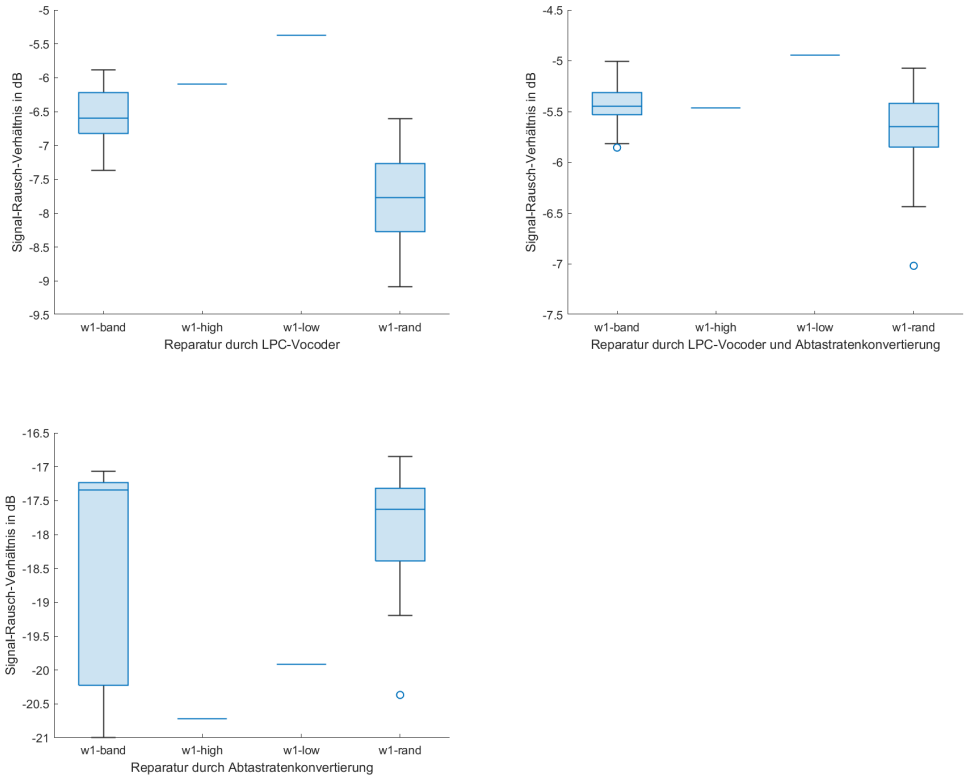

Abbildung A.3.: Signal-Rausch-Verhältnisse für das Ursprungssignal „w1“

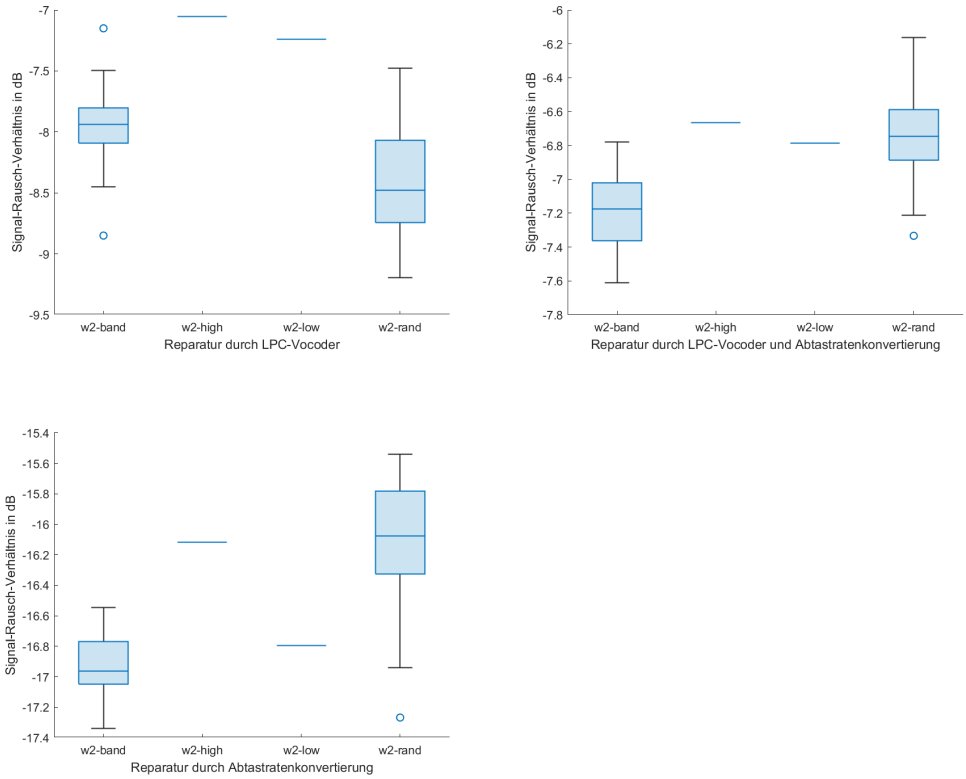

Abbildung A.4.: Signal-Rausch-Verhältnisse für das Ursprungssignal „w2“

A.2. Worterkennungsraten

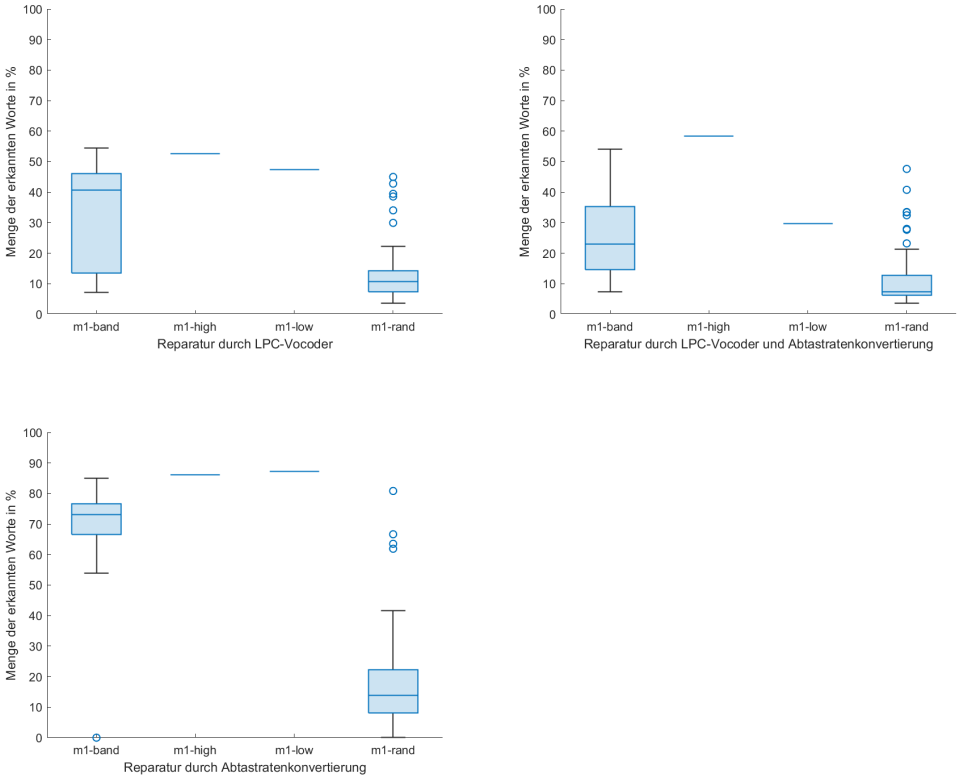

Abbildung A.5.: Worterkennungsraten der transkribierten Texte für das Ursprungssignal „m1“

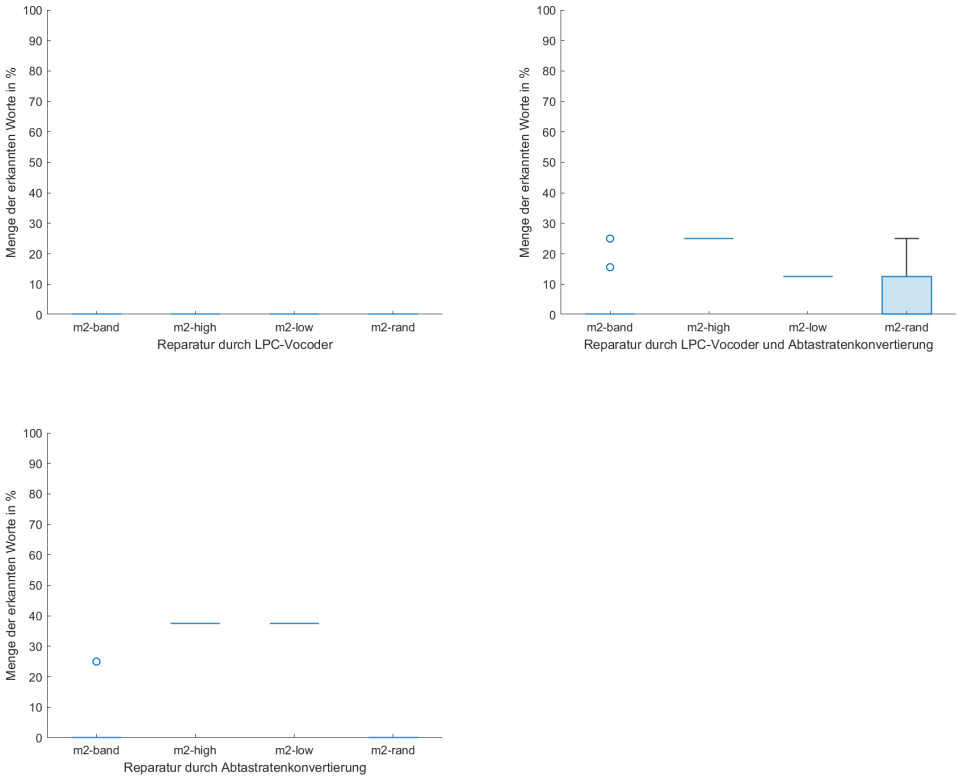

Abbildung A.6.: Worterkennungsraten der transkribierten Texte für das Ursprungssignal „m2“

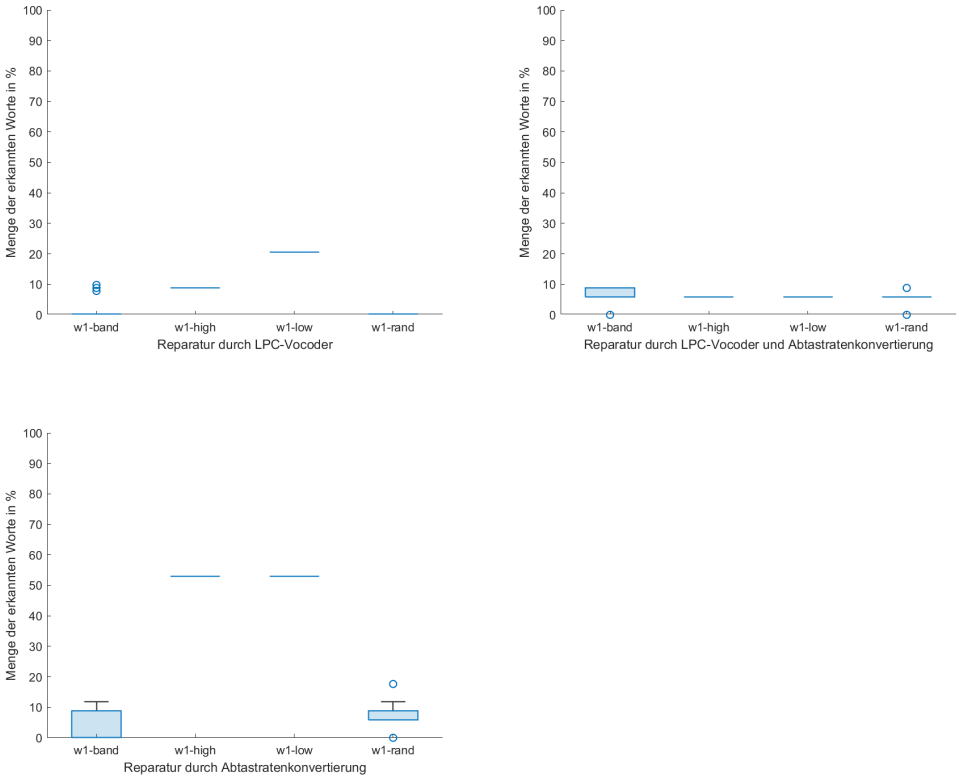

Abbildung A.7.: Worterkennungsraten der transkribierten Texte für das Ursprungssignal „w1“

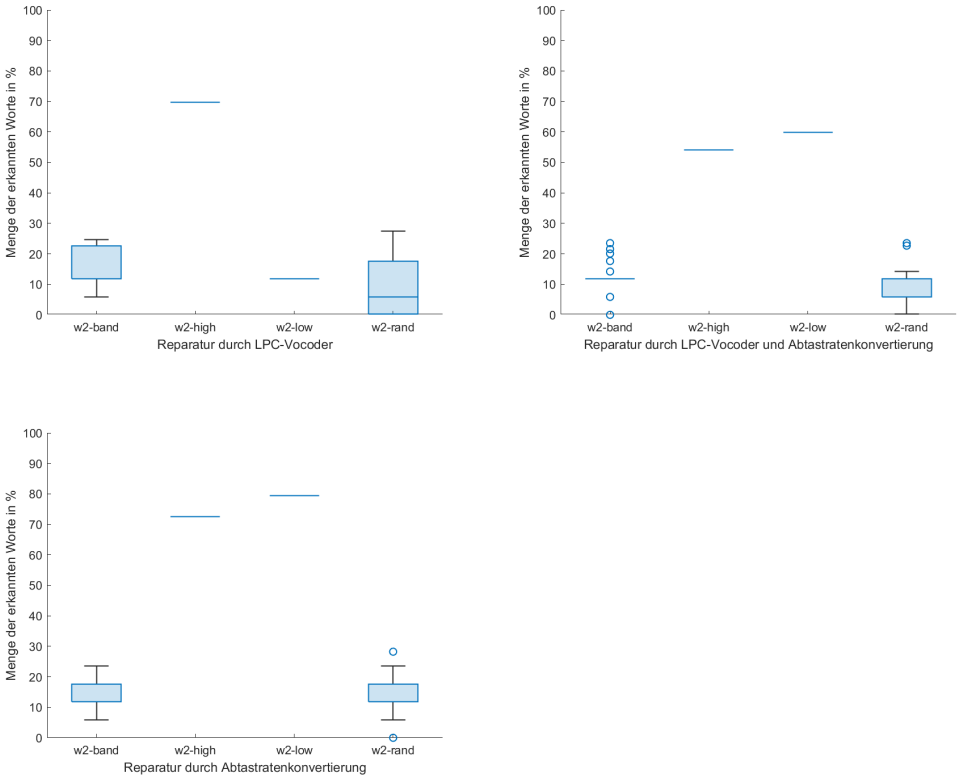

Abbildung A.8.: Worterkennungsraten der transkribierten Texte für das Ursprungssignal „w2“

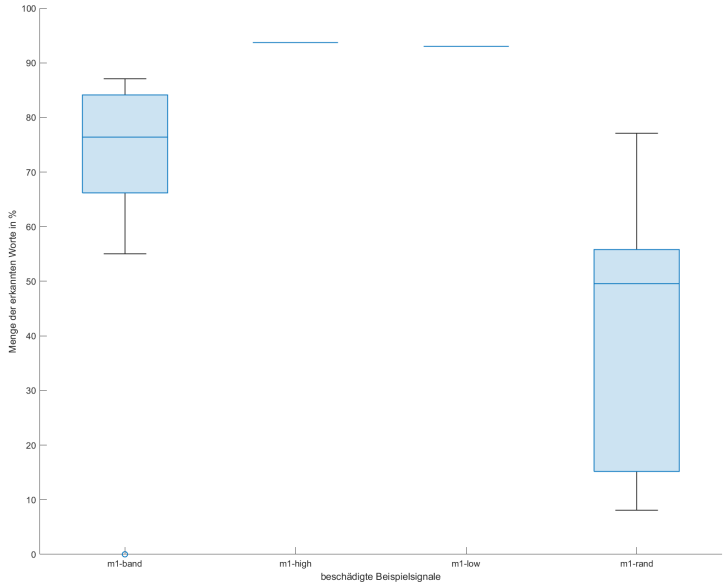

Abbildung A.9.: Worterkennungsraten der transkribierten Texte vor der Reparatur für das Ursprungssignal „m1“

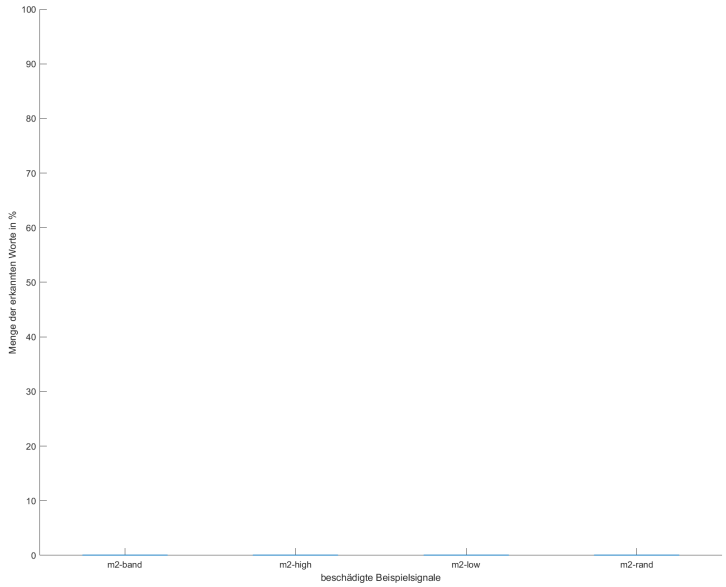

Abbildung A.10.: Worterkennungsraten der transkribierten Texte vor der Reparatur für das Ursprungssignal „m2“

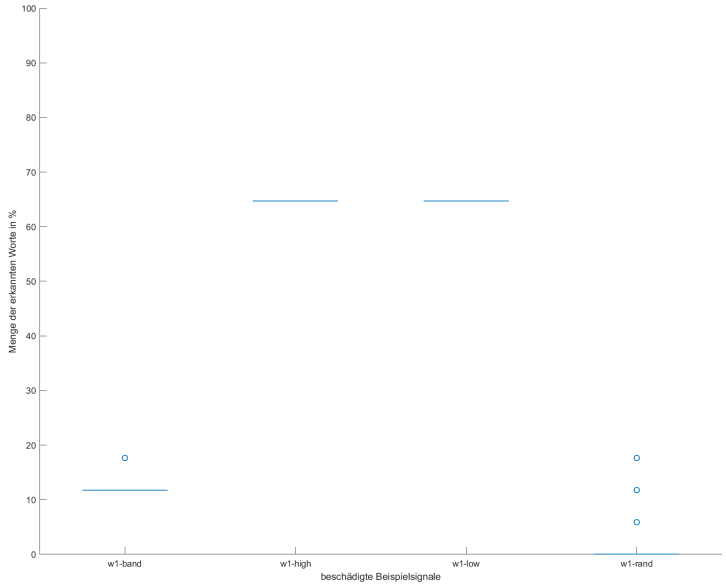

Abbildung A.11.: Worterkennungsraten der transkribierten Texte vor der Reparatur für das Ursprungssignal „w1“

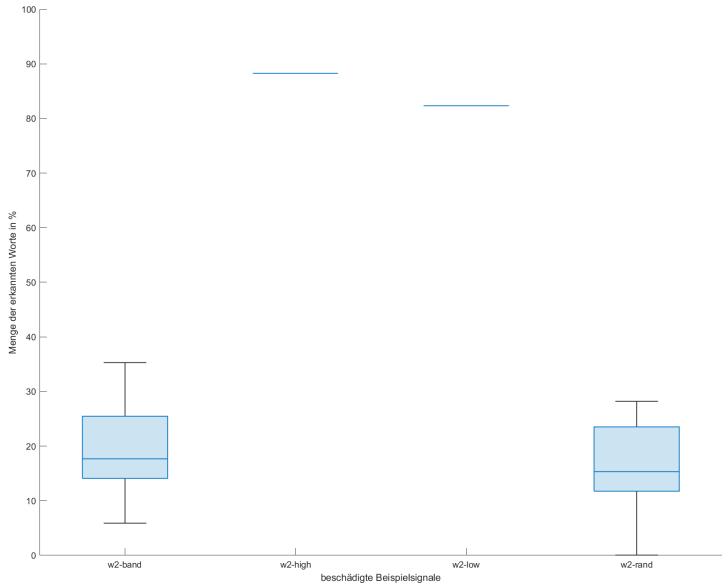

Abbildung A.12.: Worterkennungsraten der transkribierten Texte vor der Reparatur für das Ursprungssignal „w2“

A.3. Wortfehlerraten

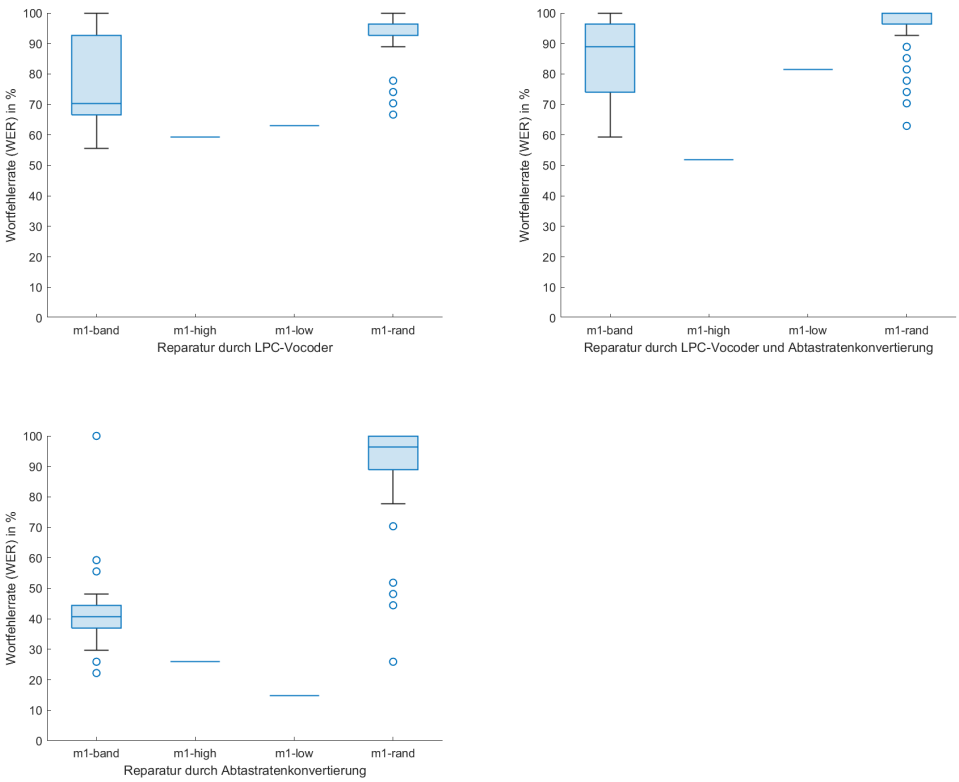

Abbildung A.13.: Wortfehlerraten der transkribierten Texte für das Ursprungssignal „m1“

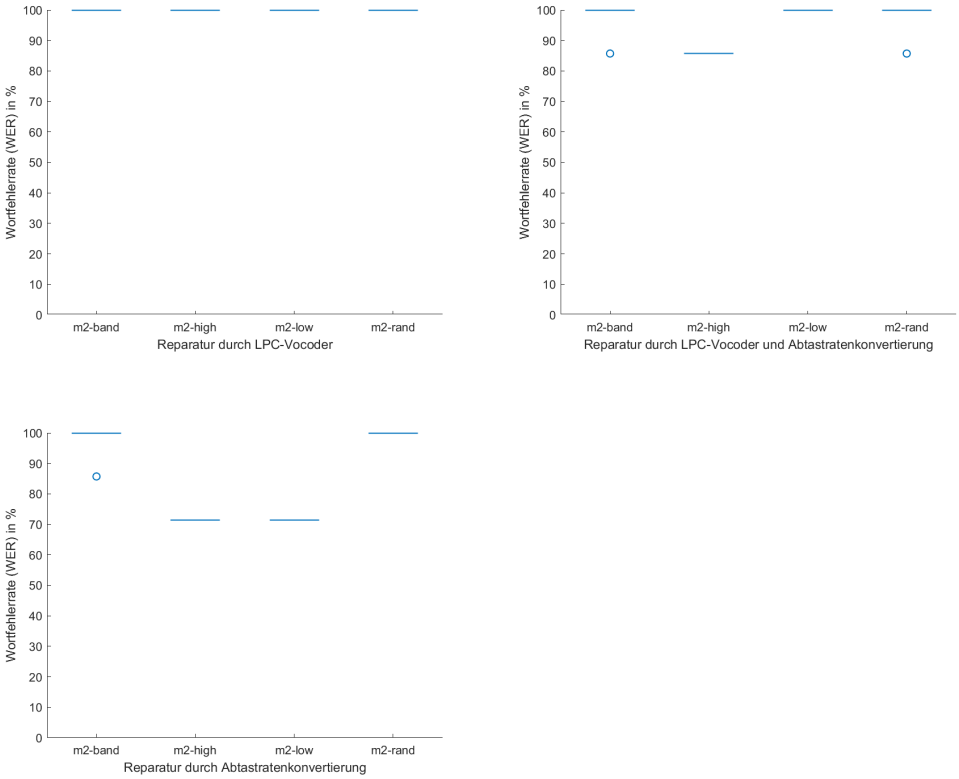

Abbildung A.14.: Wortfehlerraten der transkribierten Texte für das Ursprungssignal „m2“

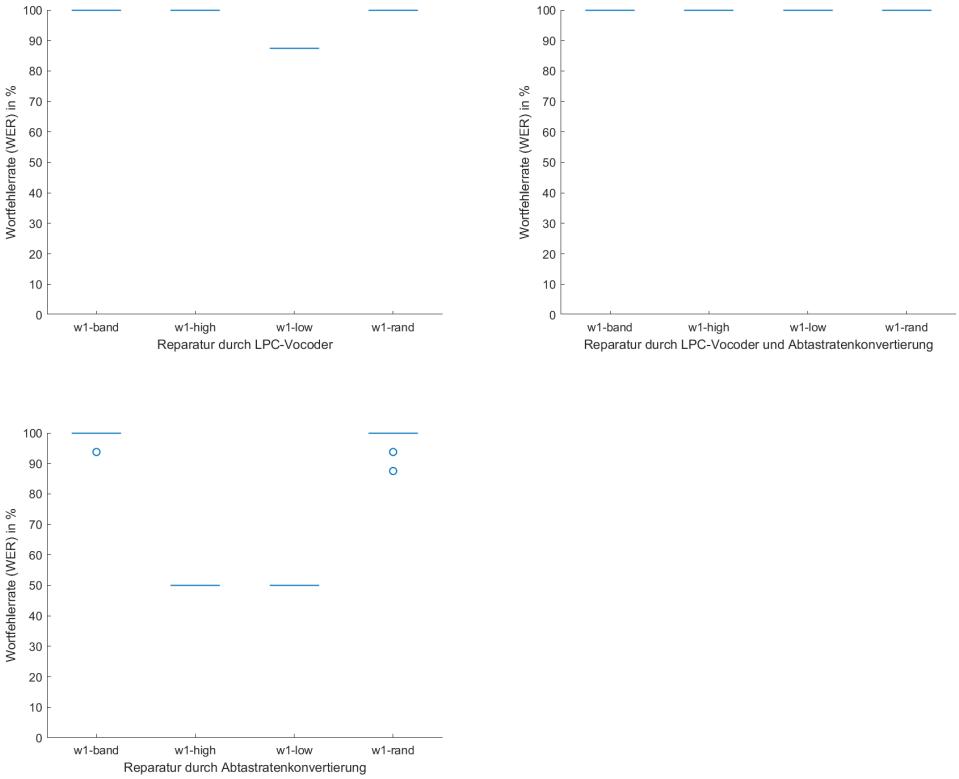

Abbildung A.15.: Wortfehlerraten der transkribierten Texte für das Ursprungssignal „w1“

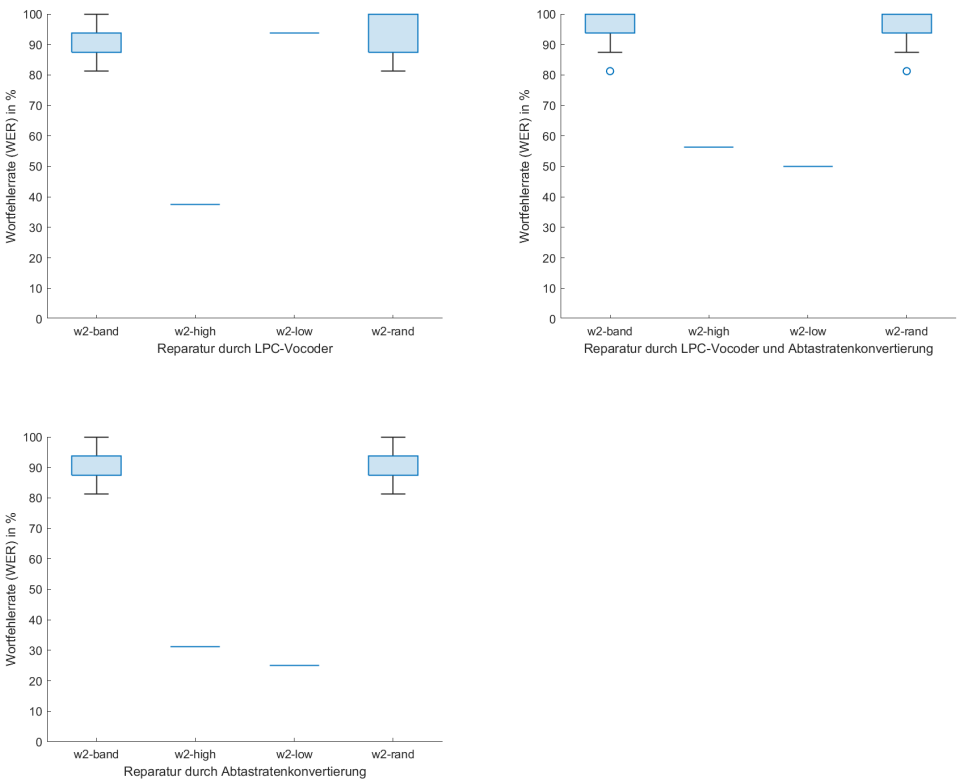

Abbildung A.16.: Wortfehlerraten der transkribierten Texte für das Ursprungssignal „w2“

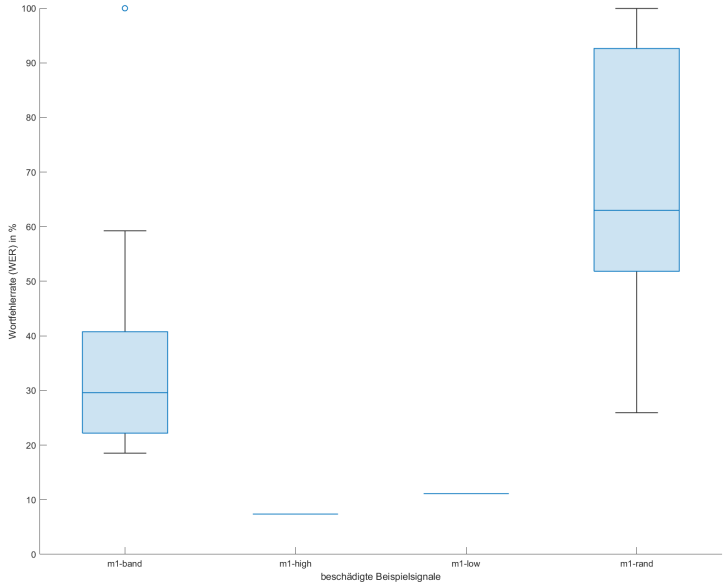

Abbildung A.17.: Wortfehlerraten der transkribierten Texte vor der Reparatur für das Ursprungssignal „m1“

Abbildung A.18.: Wortfehlerraten der transkribierten Texte vor der Reparatur für das Ursprungssignal „m2“

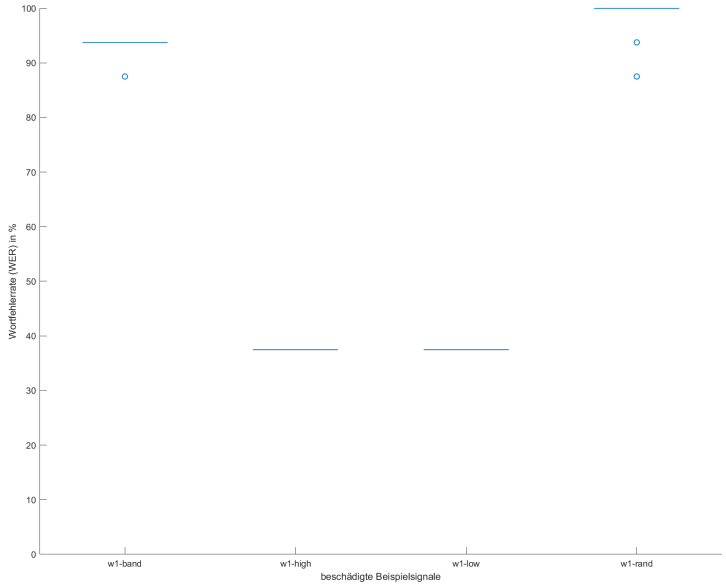

Abbildung A.19.: Wortfehlerraten der transkribierten Texte vor der Reparatur für das Ursprungssignal „w1“

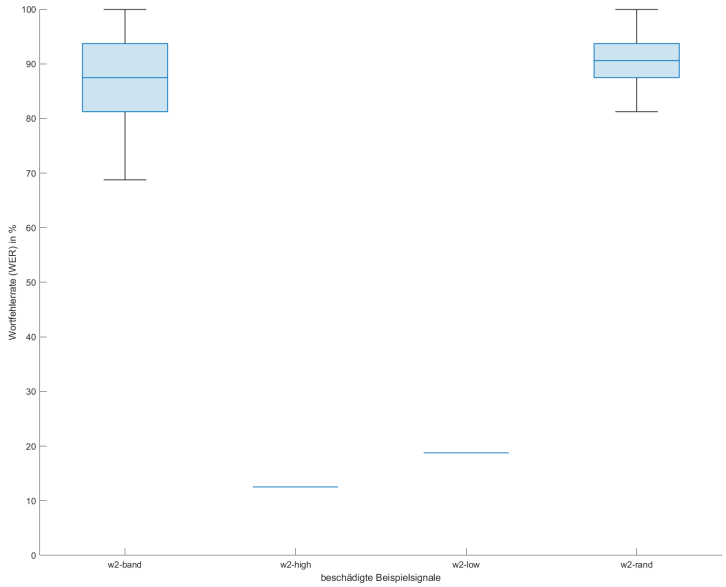

Abbildung A.20.: Wortfehlerraten der transkribierten Texte vor der Reparatur für das Ursprungssignal „w2“

A.4. Wortinformationsverlust

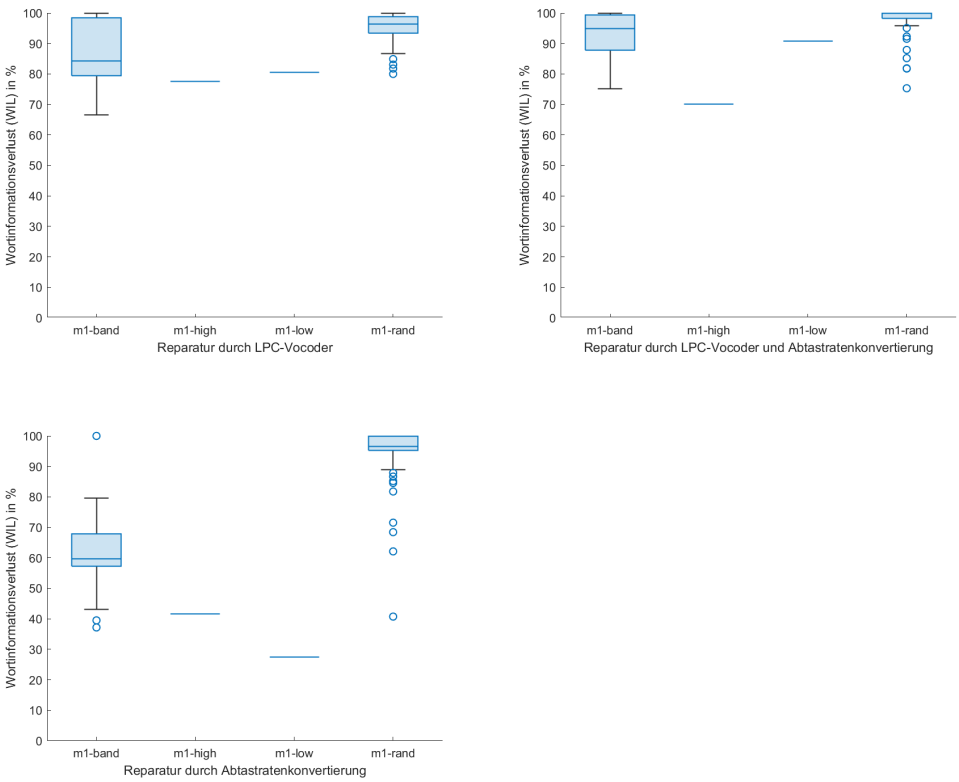

Abbildung A.21.: Wortinformationsverluste der transkribierten Texte für das Ursprungssignal „m1“

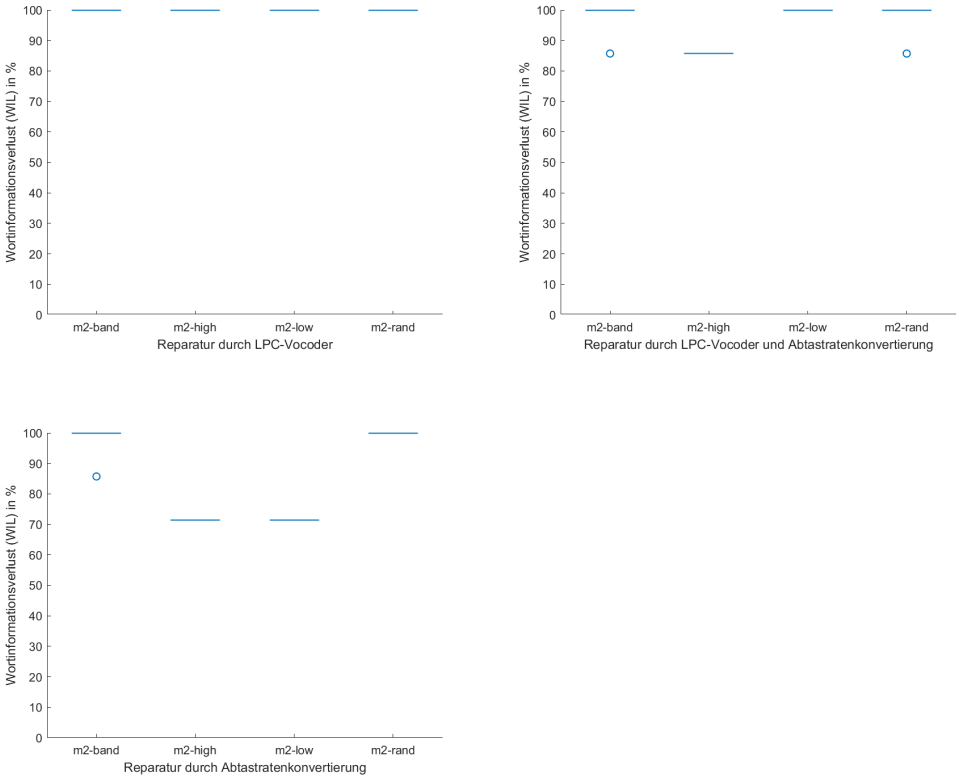

Abbildung A.22.: Wortinformationsverluste der transkribierten Texte für das Ursprungssignal „m2“

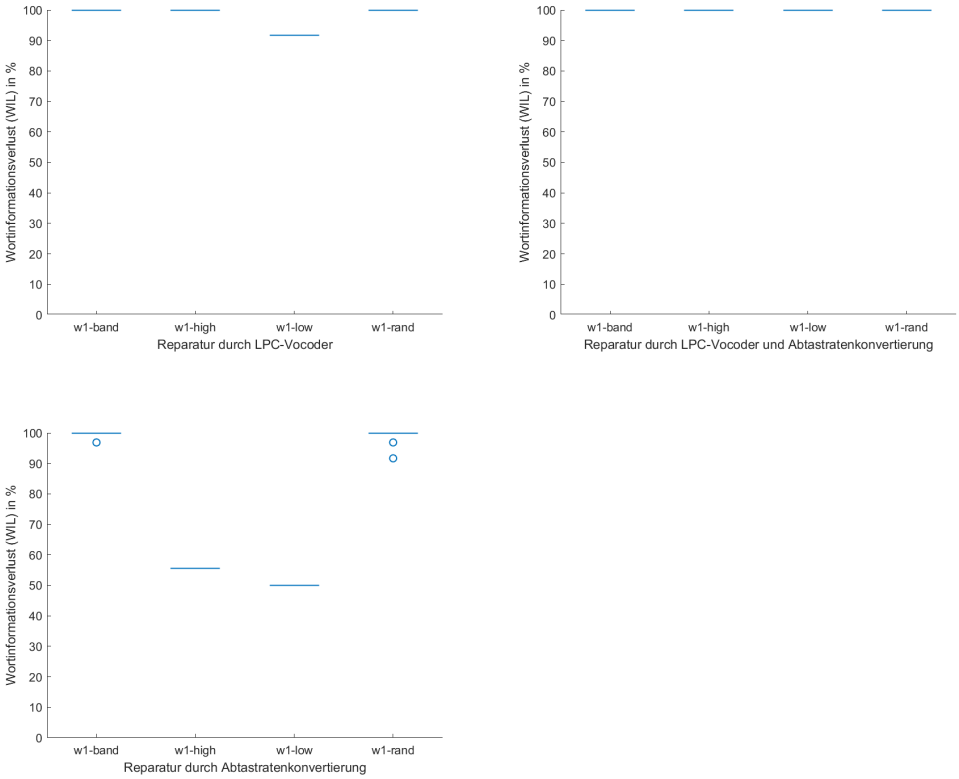

Abbildung A.23.: Wortinformationsverluste der transkribierten Texte für das Ursprungssignal „w1“

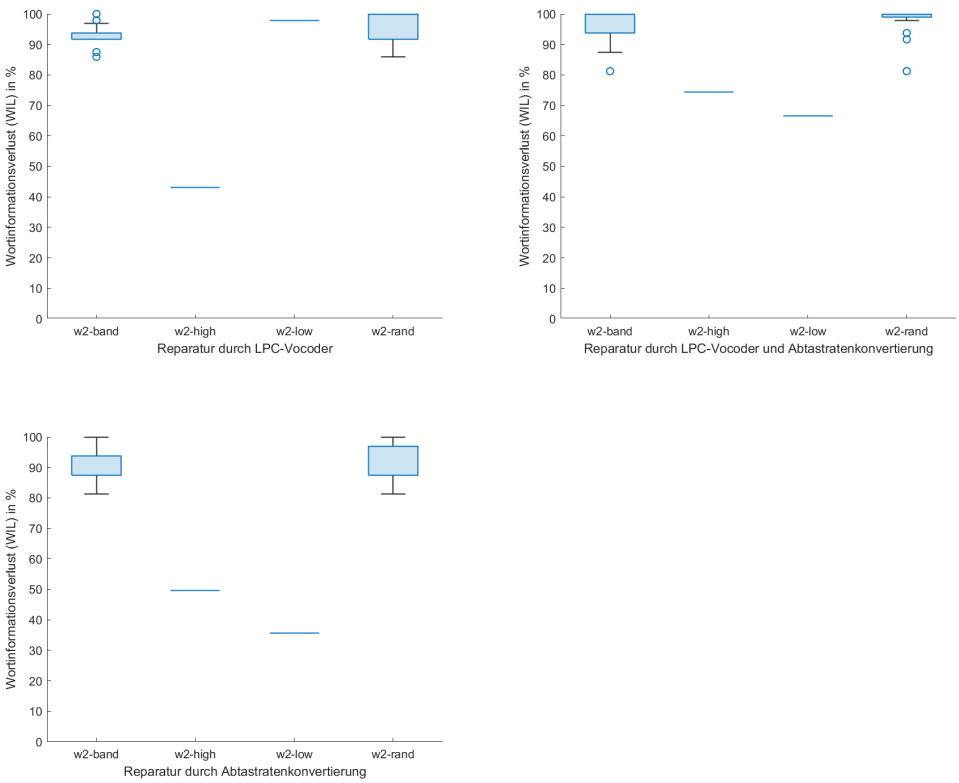

Abbildung A.24.: Wortinformationsverluste der transkribierten Texte für das Ursprungssignal „w2“

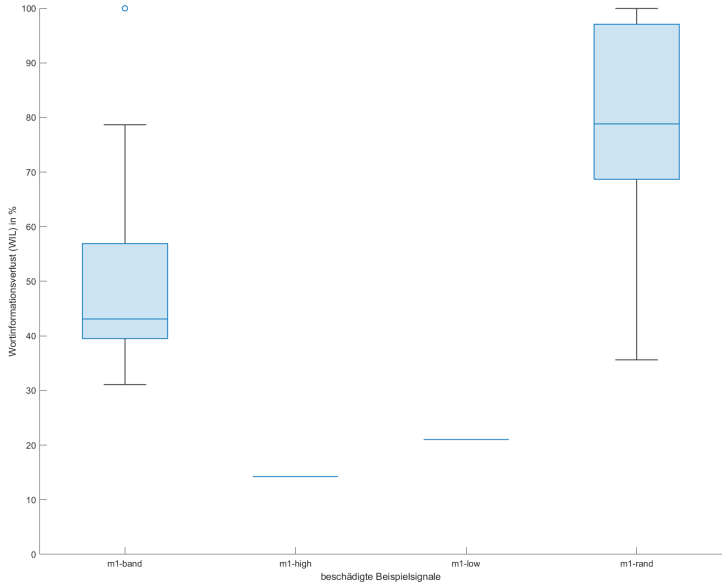

Abbildung A.25.: Wortinformationsverluste der transkribierten Texte vor der Reparatur für das Ursprungssignal „m1“

Abbildung A.26.: Wortinformationsverluste der transkribierten Texte vor der Reparatur für das Ursprungssignal „m2“

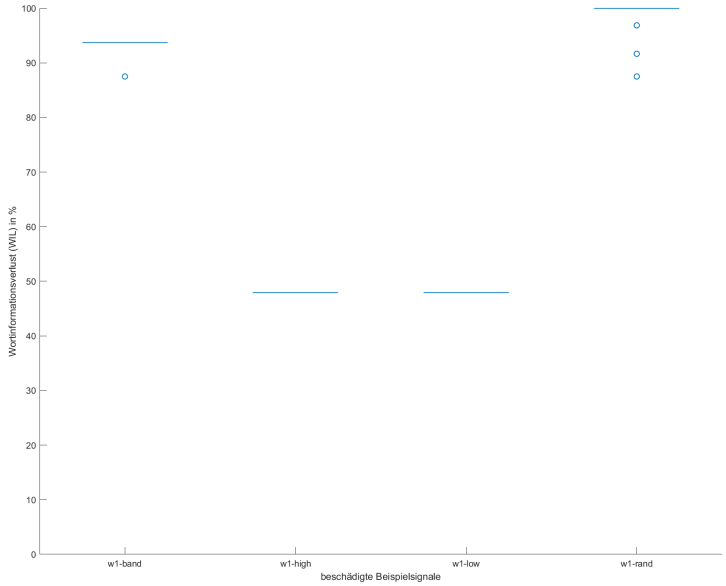

Abbildung A.27.: Wortinformationsverluste der transkribierten Texte vor der Reparatur für das Ursprungssignal „w1“

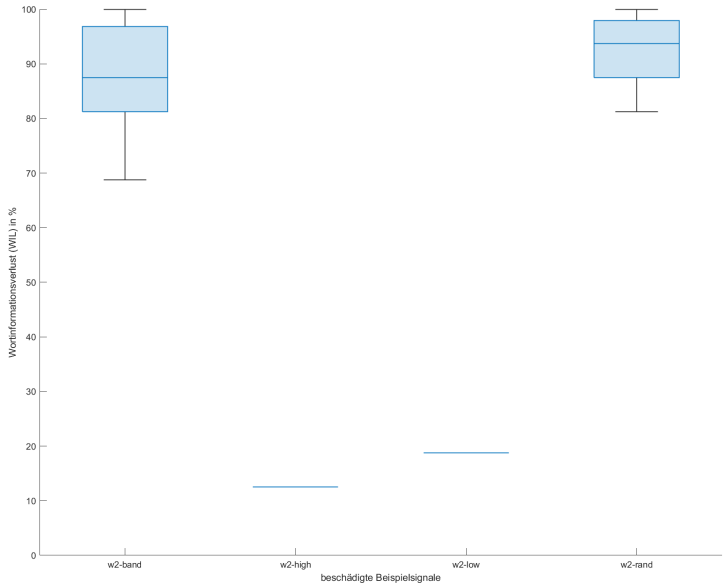

Abbildung A.28.: Wortinformationsverluste der transkribierten Texte vor der Reparatur für das Ursprungssignal „w2“

TECHNISCHE UNIVERSITÄT CHEMNITZ

Studentenservice – Zentrales Prüfungsamt

Selbstständigkeitserklärung

Name: van der Smissen Vorname: Jonas geb. am: 19.12.1997 Matr.-Nr.: 436768	Bitte beachten: 1. Bitte binden Sie dieses Blatt am Ende Ihrer Arbeit ein.

Selbstständigkeitserklärung*

Ich erkläre gegenüber der Technischen Universität Chemnitz, dass ich die vorliegende Bachelorarbeit selbstständig und ohne Benutzung anderer als der angegebenen Quellen und Hilfsmittel angefertigt habe.

Die vorliegende Arbeit ist frei von Plagiaten. Alle Ausführungen, die wörtlich oder inhaltlich aus anderen Schriften entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch nicht als Prüfungsleistung eingereicht und ist auch noch nicht veröffentlicht.

Datum: Unterschrift: ……

* Statement of Authorship

I hereby certify to the Technische Universität Chemnitz that this thesis is all my own work and uses no external material other than that acknowledged in the text.

This work contains no plagiarism and all sentences or passages directly quoted from other people's work or including content derived from such work have been specifically credited to the authors and sources.

This paper has neither been submitted in the same or a similar form to any other examiner nor for the award of any other degree, nor has it previously been published.