TECHNISCHE UNIVERSITÄT DRESDEN
FAKULTÄT ELEKTROTECHNIK UND INFORMATIONSTECHNIK
Institut für Akustik und Sprachkommunikation

DIPLOMARBEIT

„Vergleich maschineller Lernverfahren für die Grundfrequenzvorhersage in der Sprachsynthese“

von

Patrick Schmager
geboren am 25.06.1992 in Torgau

zur Erlangung des akademischen Grades

DIPLOMINGENIEUR
(Dipl.-Ing.)

Tag der Einreichung: 30.10.2017
Betreuer der Diplomarbeit: Jun.-Prof. Dr.-Ing. Peter Birkholz
Verantwortlicher Hochschullehrer: Jun.-Prof. Dr.-Ing. Peter Birkholz

TECHNISCHE UNIVERSITÄT DRESDEN

Fakultät Elektrotechnik und Informationstechnik
Juniorprofessur Kognitive Systeme

Aufgabenstellung für die Diplomarbeit

Für: Herrn Patrick Schmager, Matrikelnr.: 3755038
Studiengang: Elektrotechnik, PO 2010

Thema: **Vergleich maschineller Lernverfahren für die Grundfrequenz-Vorhersage in der Sprachsynthese**

Die Grundfrequenz (F0) des Stimmtons ist ein wichtiger Parameter für den prosodischen Ausdruck von Sprachäußerungen. Der zeitliche Verlauf der F0 hängt dabei von mehreren Einflussgrößen wie der Phrasierung und den Wort- und Satzakzenten der Äußerung ab. In der Sprachsynthese hat die Vorhersage des F0-Verlaufs einen ganz entscheidenden Einfluss auf die wahrgenommene Natürlichkeit synthetischer Sprache: Bei aktuellen Synthesesystemen erkennt man, dass es sich um eine künstliche Stimme handelt oft daran, dass einzelne Silben oder Wörter falsch betont werden.
In dieser Arbeit soll ein neues Verfahren für die F0-Vorhersage auf Basis des Target-Approximation-Modells für die Grundfrequenz von Einzelwörtern in Kombinationen mit einem maschinellen Lernverfahren entwickelt und bewertet werden. Das Target-Approximation-Modell definiert im Zeitintervall jeder Silbe eine lineare Funktion als „Zielfunktion" für die F0, wobei der tatsächliche F0-Verlauf durch die sukzessive asymptotische Annäherung an die Ziele entsteht. Die Parameter der Zielfunktionen (Offset und Steigung) sollen automatisch für alle Silben der Wörter anhand relevanter Einflussgrößen vorhergesagt werden. Als Datenbasis dient ein annotierter Sprachkorpus' von 2000 Wörtern.

Folgende Teilaufgaben sind zu lösen:

- Ausführliche Literaturrecherche zum Stand der Forschung
- Erweiterung eines Tools zur Schätzung der Target Approximation Parameter für Sprachäußerungen (PentaTrainer)
- Bestimmung geeigneter Eingabemerkmalvektoren für die Lernverfahren
- Implementierung und Test von mindestens zwei aktuellen Lernverfahren mit geeigneten Toolboxen (z.B. SVMs und MLPs) und Bestimmung der optimalen Hyperparameter durch Kreuzvalidierung
- Perzeptionstest zur subjektiven Bewertung der Vorhersagegüte der Lernverfahren.

1. Prüfer: Jun.-Prof. Dr.-Ing. Peter Birkholz
2. Prüfer: PD Dr.-Ing. Ulrich Kordon

Ausgehändigt: 22.05.2017
Einzureichen: 30.10.2017

Peter Birkholz

Prof. Dr.-Ing. Steffen Bernet
Vorsitzender des Prüfungsausschusses

Jun.-Prof. Dr.-Ing. Peter Birkholz
Verantwortlicher Hochschullehrer

Selbständigkeitserklärung

Hiermit erkläre ich, Patrick Schmager, dass die heute beim Prüfungsausschuss der Fakultät Elektrotechnik und Informationstechnik eingereichte Diplomarbeit

„Vergleich maschineller Lernverfahren für die Grundfrequenzvorhersage in der Sprachsynthese“

vollkommen selbständig von mir verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel verwendet und Zitate kenntlich gemacht wurden.

Dresden, den 30.10.2017 Patrick Schmager

Inhaltsverzeichnis

Verwendete Bezeichner

Mathematische Symbole

$\mathbb{R}$	Menge der reellen Zahlen
$\mathbb{R}_{++}$	Menge der echt positiven reellen Zahlen
$\mathbb{Z}$	Menge der ganzen Zahlen
$\mathbb{H}$	Hilbertraum mit reproduzierbaren Kern
$\langle x, y \rangle$	Skalarprodukt zweier Elemente aus $\mathbb{H}$
$\lVert \mathbf{x} \rVert_p$	ℓ_p-Norm des Vektors $\mathbf{x}$
$\lVert \mathbf{x} \rVert$	ℓ_2-Norm des Vektors $\mathbf{x}$
X^{T}	Transponierte der Matrix X
X^{-1}	Inverse der Matrix X
$\mathbf{x}_i$	i-ter Zeilenvektor der Matrix X
$\mathbf{x}_{\cdot i}$	i-ter Spaltenvektor der Matrix X
x_i	i-tes Element des Vektors $\mathbf{x}$
$\mathrm{diag}(\mathbf{x})$	Diagonalmatrix mit den Elementen x_i auf der Hauptdiagonalen
$x^{(i)}$	i-tes Element einer Stichprobe
$f^{(i)}(x)$	i-te Ableitung der Funktion f nach x
$\delta(x)$	Delta-Distribution („Dirac-Impuls“)
x^*	Optimalwert aller möglichen Werte x einer Suche
$\hat{x}$	Schätzwert der Größe x
$\bar{x}$	Mittelwert der Größe x
$\tilde{x}$	Median der Größe x

Abkürzungen

f_0	*Grundfrequenz*
AKF	*Autokorrelationsfunktion*
BOBYQA	*Bounded Optimization by Quadratic Approximation*
BPA	*Backpropagation Algorithm*
CVE	*Cross Validation Error*
DBN	*Deep Belief Network*
DNN	*Deep Neural Network*
ERM	*Empirical Risk Minimization*
HPC	*High Performance Computer*
IPA	*International Phonetic Alphabet*
KRR	*Kernel Ridge Regression*
KNN	*Künstliche Neuronale Netze*
LRR	*Linear Ridge Regression*
LSTM	*Long Short Term Memory*
MLP	*Multilayer Perceptron*
MOS	*Mean Opinion Score*
MSE	*Mean Squared Error*
MVUE	*Minimum Variance Unbiased Estimator*
PENTA	*Parallel Encoding Target Approximation*
PDA	*Pitch Detection Algorithm*
PSOLA	*Pitch Synchronous Overlap and Add*
PT	*Pitch Target*
RBM	*Restricted Boltzmann Machine*
RKHS	*Reproducing Kernel Hilbert Space*
RMSE	*Root Mean Square Error*
RNN	*Recurrent Neural Network*
SAMPA	*Speech Assessment Methods Phonetic Alphabet*
SSE	*Sum of Squared Errors*
SVR	*Support Vector Regression*
TAM	*Target-Approximation-Modell*
TD-PSOLA	*Time Domain - Pitch Synchronous Overlap and Add*
ToBI	*Tones and Break Indices*
TTS	*Text-to-Speech*
VGP	*Vanishing Gradient Problem*

Einführung

„In der Sprachsynthese ist eine gute Prosodiesteuerung das größte Problem, das derzeit einer guten Qualität synthetischer Sprache noch im Wege steht.“ (Vary et al., 1998)

Gut 20 Jahre nach der Veröffentlichung dieses Textes lässt sich feststellen, dass erhebliche Fortschritte in der Sprachsynthese gemacht wurden; die Qualität synthetisierter Sprache jedoch immer noch nicht an jene natürlich artikulierter Sprache mit ihrer großen Variabilität heranreicht. Dabei liegt das Hauptproblem nach wie vor in einer mangelhaften Generierung der Prosodie. Aus diesem Grund versucht die vorliegende Arbeit einen Beitrag zur Prosodiesteuerung in der Sprachsynthese zu leisten, wobei der Fokus auf der Grundfrequenzvorhersage liegen soll.

Die Grundfrequenz ist die niedrigste Frequenz einer Überlagerung von harmonischen Schwingungen, wobei sich stimmhaft artikulierte Sprache als solche auffassen lässt. Der zeitliche Grundfrequenzverlauf einer sprachlichen Äußerung charakterisiert maßgeblich deren Intonation und andere prosodische Parameter. In diesen prosodischen Eigenschaften sprachlicher Äußerungen sind eine Vielzahl von Informationen codiert, die sich nicht in den einzelnen Lauten widerspiegeln, sondern im gesamten Klangbild einer Äußerung und auch Sprache. Beispielsweise werden durch die Prosodie Informationen über Betonung und Sprechrhythmus, aber ebenso über die Befindlichkeit und Emotionen des Sprechers, übertragen. Damit hat die Prosodie und somit die Grundfrequenz als eine akustische Realisierung dieser, einen entscheidenden Einfluss auf die wahrgenommene Natürlichkeit von artikulierter Sprache. Dies begründet folglich die zentrale Rolle der Prosodiesteuerung in der Sprachsynthese, die den Anspruch hat, möglichst natürliche Sprache zu generieren.

In den meisten praktischen Anwendungen ist die Sprachsynthese innerhalb eines Text-to-Speech Systems eingebettet. Ein solches System enthält Textinformation als Eingabe, verarbeitet diese und generiert ein Sprachsignal als Ausgabe. Aktuelle Text-to-Speech Systeme sind sehr komplex und bedienen sich in der Regel Methoden der Computerlinguistik, der Sprachsignalverarbeitung und des Maschinenlernens, wie in Abbildung 0.1 angedeutet. Die Grundfrequenzvorhersage stellt in einem solchen komplexen System einen zentralen Baustein dar, der sich ebenfalls den Methoden der genannten Disziplinen bei einer praktischen Umsetzung bedient.

Das Hauptproblem bei der Grundfrequenzvorhersage besteht darin, dass Textinformation eigentlich keine bzw. kaum Informationen über die Prosodie beinhaltet. Aus diesem Grund stoßen Ansätze, die versuchen explizite Regeln zwischen Textinformation und Grundfrequenzverlauf zu bestimmen, schnell an ihre Grenzen. Menschen erlernen die Prosodiesteuerung mit dem Spracherwerb, welche damit in ihrer Erfahrung verankert ist. Aus diesem Grund liegt die Idee nahe, dass ein Training maschineller Lernverfah-

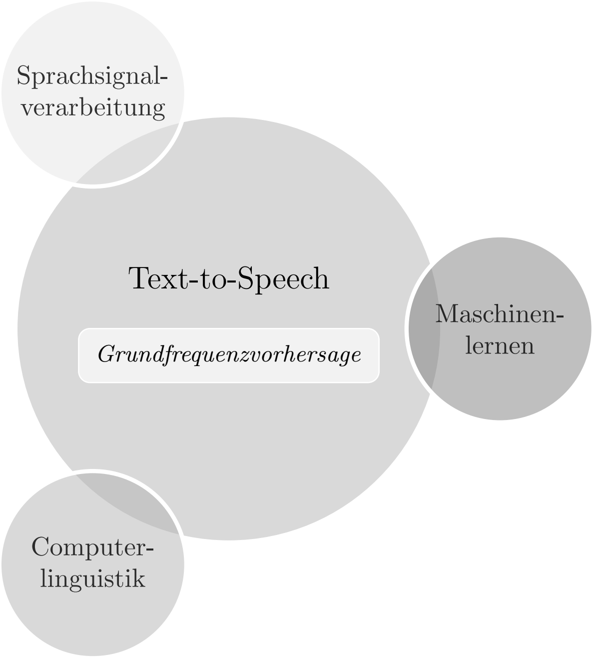

Abbildung 0.1: Thematische Einordnung der Problemstellung „Grundfrequenzvorhersage“.

ren auf Basis natürlicher Äußerungen diesen Prozess abbilden kann. Dabei muss jedoch bedacht werden, dass das Problem nicht zu stark abstrahiert werden sollte, denn die Grundfrequenz wird durch den menschlichen Sprechapparat erzeugt, welcher mögliche Grundfrequenzverläufe determiniert. Diese beiden Ideen von empirischen Erwerb und artikulatorischen Grenzen der Grundfrequenzproduktion bilden den Kern des in dieser Arbeit entwickelten Systems zur Grundfrequenzvorhersage.

Das Phänomen der Prosodie ist keineswegs vollends verstanden und damit Gegenstand aktueller Forschung. Aus diesem Grund konnte bisher auch noch kein universelles Modell entwickelt werden, welches die Grundfrequenz in all ihren Facetten mathematisch beschreibt. Einen vielversprechenden Weg dorthin stellt jedoch das Target-Approximation-Modell von Xu und Wang (2001) dar, welches versucht den Prozess der Grundfrequenzproduktion im menschlichen Sprechapparat zu modellieren und gleichzeitig eine elegante mathematische Form aufweist. Damit ist dieses Modell sehr gut für den Zweck der Grundfrequenzvorhersage geeignet. Auf Basis dieses Modells sollen verschiedene maschinelle Lernverfahren auf deren Eignung für die Grundfrequenzvorhersage untersucht werden, wobei der Fokus auf verschiedenen Kernmethoden und künstlichen neuronalen Netzen liegt.

Die Untersuchungen dieser Arbeit dienen dabei dem Projekt „Deutsche Aussprachedatenbank“ (Förster, 2014) der Abteilung Sprechwissenschaft und Phonetik der Martin-Luther-Universität Halle-Wittenberg in Kooperation mit dem Lehrstuhl für Kognitive System der Technischen Universität Dresden. Ziel dieses Projektes ist der Aufbau einer Online-Datenbank, mit der die Inhalte des Deutschen Aussprachewörterbuches von

Krech et al. (2009) einer breiten Nutzergruppe zugänglich gemacht werden können. Die Einträge sollen neben der phonetischen Transkription und Angabe der Wortherkunft auch Hörbeispiele bereitstellen. Qualität, Vergleichbarkeit und Erweiterbarkeit sind dabei die zentralen Anforderungen an die Produktion der Aufnahmen. Alle Hörbeispiele sollen idealerweise von einer einheitlichen Frauenstimme gesprochen werden. Bei rund 130.000 enthaltenen Wörtern mit ständig wachsender Zahl lassen sich diese Anforderungen durch ein Einsprechen der Wörter kaum umsetzen, wobei der damit verbundene, finanzielle und zeitliche Aufwand noch gar nicht berücksichtigt ist. Aus diesem Grund wurde der Versuch getätigt mithilfe artikulatorischer Sprachsynthese ein System zu entwickeln, welches die geforderten Aufnahmen automatisch synthetisiert und damit alle Anforderungen erfüllt. Bei der artikulatorischen Sprachsynthese wird der gesamte Sprechapparat des Menschen in einem dreidimensionalen Modell simuliert und liefert damit eine hohe Qualität synthetisierter Sprache. Das zu lösende Problem besteht folglich in der Vorhersage der Parameter für die artikulatorische Sprachsynthese auf Basis der im Ausspracheswörterbuch enthaltenen Transkriptionen. An diesem Punkt knüpfen die Untersuchungen der vorliegenden Arbeit an, welche sich im Speziellen auf die Vorhersage der prosodischen Parameter fokussieren.

Bei der Einzelwortsynthese, wie im Falle eines Wörterbuchs, kommt der Grundfrequenzvorhersage eine zentrale Rolle zu, gerade weil eine hohe Qualität gefordert wird. Die in dieser Arbeit entwickelte Methode zur Grundfrequenzvorhersage soll damit nicht nur die Leistungsfähigkeit des Target-Approximation-Modells untersuchen, sondern auch ein Werkzeug für weitere Arbeiten innerhalb des beschriebenen Projekts sowie anderer Projekte bereitstellen. Für die Umsetzung der Methode steht ein Korpus von rund 2.000 Äußerungen zur Verfügung, das von einer professionellen Sprecherin eingesprochen wurde. Diese Daten bilden die Grundlage der zu vergleichenden Lernverfahren. Die Vorhersage findet auf Basis der Transkription aus dem erwähnten Ausspracheswörterbuch statt.

Die vorliegende Arbeit gliedert sich in sechs Kapitel. Im ersten Kapitel werden kurz die notwendigen Grundlagen aus Linguistik und Phonetik skizziert, die zur Beschreibung der Grundfrequenz und deren Vorhersage notwendig sind. Kapitel 2 beschäftigt sich mit dem aktuellen Stand der Technik zur Grundfrequenzvorhersage. Darin werden die verschiedenen Grundfrequenzmodelle erläutert, unterschiedliche maschinelle Lernverfahren diskutiert sowie verschiedene existierende Vorhersagelösungen verglichen. Im dritten Kapitel wird detailliert die entwickelte Lösungsmethode, aufbauend auf dem Target-Approximation-Modell, vorgestellt und sowohl formal als auch aus Implementierungssicht beschrieben. Entsprechende Untersuchungsergebnisse werden in Kapitel 4 präsentiert und diskutiert. Eine Einordnung und Beurteilung der Ergebnisse wird im vorletzten Kapitel vorgenommen. Schließlich befasst sich das Kapitel 6 mit Erweiterungen und Verbesserungsvorschlägen der entwickelten Lösungsmethode.

Kapitel 1

1 Grundlagen

Um sich dem Problem der Grundfrequenzvorhersage zu nähern, sind verschiedene Aspekte der Linguistik und Phonetik notwendig, die in diesem Grundlagenkapitel diskutiert werden und sich nach den Beschreibungen von Vary et al. (1998), Krech et al. (2009) und Pompino-Marschall (2009) richten.
Betrachtet man den Aufbau gesprochener Sprache, so lässt sich feststellen, dass diese aus gewissen Grundbausteinen besteht, die sich nach bestimmten Regeln kombinieren lassen und damit die Grundlage einer Sprache bilden. Diese Grundbausteine werden als Laute oder Phone bezeichnet und stellen ein minimales, eigenständig wahrnehmbares Schallsegment dar, bezogen auf das menschliche Gehör. Eine symbolische Abstrahierung dieser Laute sind die sogenannten Phoneme, welche nach Definition die kleinsten bedeutungsunterscheidenden Einheiten des Lautsystems einer Sprache darstellen. Zwei Wörter, die sich durch ein Phonem unterscheiden, haben also auch verschiedene Bedeutungen. Gesprochene Sprache lässt sich durch Phoneme transkribieren, wobei das Phon als akustische Realisierung des zugehörigen Phonems aufzufassen ist. Das Internationale Phonetische Alphabet (engl. *International Phonetic Alphabet*, IPA) ist eine Zusammenstellung aller Phoneme und ordnet diesen ein eindeutiges Symbol zu. Zu Zwecken der maschinellen Verarbeitung der IPA Symbole wurde eine auf 7-bit ASCII-Zeichen basierte Darstellung der Symbole entwickelt, deren Gesamtheit als SAMPA (engl. *Speech Assessment Methods Phonetic Alphabet*) benannt wird. Auch in der schriftlichen Realisierung einer Sprache, die beispielsweise auf dem lateinischen Alphabet basiert, lassen sich solche kleinsten bedeutungsunterscheidenden Einheiten feststellen, die als Grapheme bezeichnet werden. (Vary et al., 1998)

Phoneme

In der deutschen Sprache existieren in etwa 50 unterschiedliche Phoneme. Prinzipiell lassen sich diese in zwei Klassen unterteilen, nämlich Vokale und Konsonanten. Grundsätzlich wird ein Phonem als Vokal bezeichnet, wenn die Artikulation stimmhaft ist, das heißt eine Schwingung der Stimmlippen (Glottis) vorliegt und keine wesentlichen Engstellen im Vokaltrakt aufweist, sodass der Luftstrom ungehindert passieren kann. Die Artikulation von Konsonanten ist dagegen durch Engstellen und Verschlüsse im Vokaltrakt gekennzeichnet, die entweder stimmhaft oder stimmlos sein kann. Weiterhin ist es möglich, diese beiden Klassen innerhalb gewisser Merkmale feiner zu kategorisieren, welche sich im Wesentlichen auf Ort und Art der Artikulation innerhalb des Vokaltrakts beziehen.
Vokale lassen sich sehr gut nach drei Merkmalen einteilen, welche die Position von Zunge

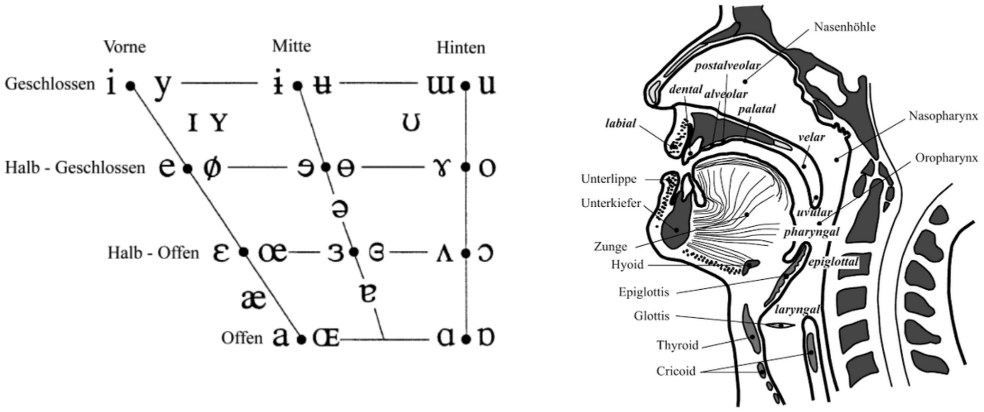

Abbildung 1.1: Vokaltrapez zur Kategorisierung der Vokale (links) und seitlicher Querschnitt des Vokaltrakts zur Kategorisierung der Konsonanten (rechts) aus Pompino-Marschall (2009).

und Lippen während der Artikulation beschreiben. Die sich ergebenden Kategorien sind Zungenhöhe, Zungenlage sowie Lippenrundung und lassen sich im sogenannten Vokaltrapez schematisch darstellen. Dieses kann als seitlicher Längsschnitt des Mundraums innerhalb eines kartesischen Koordinatensystems gedacht werden und ist in Abbildung 1.1 dargestellt. Auf der Abszisse ist die Zungenlage abzulesen und auf der Ordinate die Zungenhöhe. Die Phoneme selbst sind dabei durch ihr IPA Symbol im Diagramm eingezeichnet. Linksseitig eines jeden Koordinatenpunktes ist das äquivalente Phonem für eine ungerundete Lippenstellung angegeben, rechtsseitig jenes für die gerundete. Konsonanten lassen sich ebenfalls nach ihrem Artikulationsort einteilen, jedoch muss dafür der komplette Vokaltrakt in Betracht gezogen werden, welcher in der rechten Skizze aus Abbildung 1.1 angedeutet ist. Die Bezeichner entstammen hierbei der Biologie des Vokaltraktes. Zusätzlich unterscheiden sich Konsonanten in der Art der Artikulation, welche u.a. plosiv, frikativ, nasal, lateral oder gleitend erfolgen kann. (Pompino-Marschall, 2009)

Distinktive Merkmale

Die eben beschriebenen artikulationsspezifischen Kategorien der Phoneme gliedert diese in gewisse Gruppen, die sich jedoch überschneiden. In Anbetracht vieler praktischer Probleme ist hingegen eine eindeutige, artikulationsbezogene Unterscheidung der einzelnen Phoneme vonnöten. Eine solche eindeutige Einteilung wird durch die distinktiven Merkmale der Phonologie beschrieben. Distinktive Merkmale sind artikulatorische, binäre Kategorien, welche ein Phonem eindeutig charakterisieren. Dabei lassen sich für jede Sprache andere distinktive Merkmale identifizieren, wobei diese optimalerweise so gewählt werden sollten, dass sie bedeutungsunterscheidend sind. Wird also ein Phonem beispielsweise durch 12 distinktive Merkmale beschrieben, so sollte die Änderung eines

dieser binären Merkmale ein anderes Phonem beschreiben. Nichtbedeutungsunterscheidende Merkmale werden als redundant bezeichnet. Anzahl und Art der distinktiven Merkmale einer Sprache hängen von der Anzahl der Phoneme ab, welche die Sprache charakterisieren. Für das Deutsche findet man variierende Systeme, die jeweils zwischen 12 und 22 Merkmalen unterscheiden, wobei es einen Kompromiss zwischen Genauigkeit und Redundanz der Beschreibung gibt, der jeweils von der konkreten Anwendung abhängig ist.
Die im vorherigen Abschnitt beschriebenen Kategorien zur Einteilung der Vokale und Konsonanten nach dem Ort und der Art der Artikulation dienen sehr häufig als distinktive Merkmale. Aber auch zusätzliche Kategorien wie beispielsweise Stimmhaftigkeit, Nasalität, Gespanntheit, Tonalität oder Glottisposition können als Merkmale gewählt werden. Die Darstellung aller Phoneme einer Sprache mittels distinktiver Merkmale lässt sich in einer binären Merkmalmatrix festhalten, die sehr gut zur maschinellen Verarbeitung in der Computerlinguistik geeignet ist. (Vary et al., 1998)

Silben

Natürlicherweise lässt sich eine gesprochene Äußerung in meist mehrere Sprecheinheiten eindeutig unterteilen, die in einem Zug gesprochen werden und aus mehreren Lauten bestehen. Solche rhythmischen Sprecheinheiten werden als Silben bezeichnet. In der Linguistik unterteilt man eine Silbe in drei Komponenten; nämlich Onset (Silbenanlaut), Nukleus (Silbengipfel) und Koda (Silbenauslaut), wobei Nukleus und Koda zusammengefasst als Reim bekannt sind. Der Nukleus ist der vokalische Kern einer Silbe und besteht entweder aus einem Vokal, einem Diphthong oder einem speziellen, stimmhaften Konsonanten (Sonorant). Onset und Koda hingegen beinhalten eine variable Anzahl von Konsonanten, wobei sich der Onset vor dem Nukleus befindet, die Koda entsprechend dahinter. Onset und Koda einer Silbe können auch leer sein, ein vokalischer Kern hingegen ist in jeder Silbe identifizierbar und charakterisiert diese. Der eben beschriebene Aufbau einer Silbe wird in Abbildung 1.2 schematisch veranschaulicht. Dabei werden die Abkürzungen für Konsonant (K) und Vokal (V) verwendet.
Aus der Phonotaktik ist bekannt, dass im Deutschen maximal drei Konsonanten im Onset und fünf Konsonanten in der Koda auftreten. Jedoch treten fünf Konsonanten in

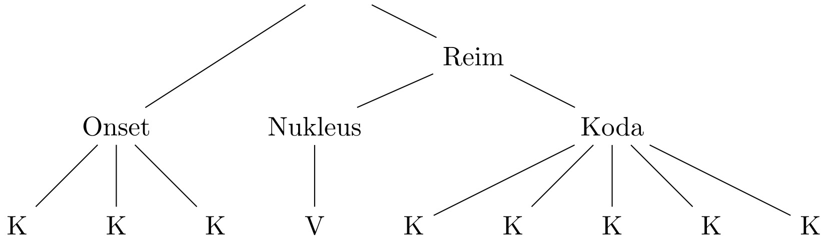

Abbildung 1.2: Aufbau einer Silbe innerhalb der deutschen Sprache.

der Koda sehr selten auf und so gut wie nie in der Grundform eines Wortes, sondern lediglich bei flektierten Wörtern wie beispielsweise *des Herbsts* [dɛs hɛʶpsts]. (Krech et al., 2009)

Prosodie

Zwar eignen sich Phoneme sehr gut um artikulierte Sprache zu charakterisieren und transkribieren, so ist es aber nicht möglich durch Phoneme allein, gesprochene Sprache in ihrer gesamten Vielseitigkeit zu erfassen. Sprache übermittelt darüber hinaus Informationen, die sich nicht als zeitliche Abfolge von Phonemen darstellen lässt, sondern sich im Klangbild der Sprache präsentiert. Die Gesamtheit all dieser lautlichen Parameter, die nicht an Phoneme gebunden sind, wird als Prosodie bezeichnet. Nach heutigen Forschungsstand gibt es keine minimalen bedeutungsunterscheidenden Einheiten für die Prosodie, was eine Kategorisierung prosodischer Merkmale erschwert. Hauptmerkmal vieler prosodischer Parameter ist ihre Suprasegmentalität, d.h. sie wirken über Phonem-, Silben-, Wort- und sogar Satzgrenzen hinweg. Je nach Blickwinkel lassen sich unterschiedliche prosodische Parameter feststellen, wobei die Darstellungen in der Literatur sehr unterschiedlich dazu sind. Oft benannte linguistische Parameter der Prosodie sind Akzentuierung, Phrasierung, Fokus und Intonation, welche sich hierbei nicht klar voneinander trennen lassen und auch ineinandergreifen. Akzente sind Stellen einzelner, lokaler Betonung und Phrasen zeichnen sich durch einen Tonhöhenverlauf mit gruppierender Wirkung aus. Die wichtigsten akustischen Parameter der Prosodie sind Grundfrequenz (engl. *Fundamental Frequency*, f_0) und Lautdauer. Die Grundfrequenz ist die tiefste Frequenz einer Überlagerung harmonischer Schwingungen. Stimmhafte Laute stellen sich als eine solche Überlagerung von Schwingungen dar, die durch einen Luftstrom, welcher an der Glottis ins Schwingen versetzte wird, erzeugt wird. Eine konkrete Zuordnung zwischen linguistischen und akustischen Parametern der Prosodie lässt sich nicht angeben, jedoch korrelieren Intonation und Grundfrequenz miteinander. Es sei noch erwähnt, dass die psychoakustische Größe der Tonhöhe (engl. Pitch) ebenfalls mit der Grundfrequenz korreliert, aber dennoch eine andere physikalische Größe darstellt. In der englischsprachigen Literatur werden die Begriffe für Tonhöhe und Grundfrequenz oft synonym verwendet.

Das Phänomen der Prosodie lässt sich jedoch nicht allein durch linguistische Parameter erfassen, da die Prosodie ebenso Träger extralinguistischer Information wie beispielsweise Emotionen, Alter, Geschlecht und physische Befindlichkeit des Sprechers ist. Betrachtet man den Parameter der Grundfrequenz, so lässt sich feststellen, dass sich dieser nicht als reine, harmonische Schwingung darstellt, sondern geringen Schwankungen in Frequenz und Amplitude unterliegt, die als Jitter bzw. Shimmer benannt werden. Diese Effekte werden gemeinhin als Mikroprosodie bezeichnet. Eng verbunden damit ist die Beobachtung, dass der Verlauf der Grundfrequenz in gewisser Weise von den artikulierten Lauten abhängt. So treten beispielsweise plötzliche Erhöhungen der Grundfrequenz an Grenzen zwischen stimmhaften und stimmlosen Lauten auf, die meist explizit für den Menschen nicht wahrnehmbar sind, aber scheinbar dennoch das Gesamtbild des Klangs beeinflussen. Solche Effekte werden als koartikulatorische Effekte bezeichnet und können auch als eine Form der Mikroprosodie betrachtet werden. (Taylor, 2009; Vary et al., 1998)

Signaltheorie

Abschließend seien noch kurz die signaltheoretischen Grundlagen erläutert, die zur Beschreibung von Sprachsignalen und der Grundfrequenz nötig sind. Eine artikulierte Äußerung kann durch ein reelles, zeitkontinuierliches Signal $x(t) \in \mathcal{X}$ beschrieben werden, wobei $\mathcal{X} := \{x(t) : x \in \mathbb{R}; t \in \mathbb{R}\}$ die Menge aller zeitkontinuierlichen Signale ist. Dabei handelt es sich zunächst um ein Schalldrucksignal, welches durch ein Mikrophon in ein adäquates Spannungssignal umgewandelt werden kann. Um dieses Signal den Methoden der digitalen Sprachsignalverarbeitung zugänglich zu machen, muss eine Abtastung des Signals erfolgen, wobei im Abstand einer Abtastperiode $\Delta t \in \mathbb{R}_{++}$ jeweilige Signalwerte bestimmt werden. Die Abtastperiode wird meist durch den Kehrwert der Abtastfrequenz $f_A = \frac{1}{\Delta t}$ dargestellt. Als Ergebnis der Abtastung liegt dann das abgetastete, zeitkontinuierliche Signal $x_A(t) = \sum_{k=-\infty}^{\infty} x(k)\delta(t - k\Delta t) \in \mathcal{X}$ vor. Das abgetastete Signal weist damit nur an den Abtastzeitpunkten $k\Delta t$ Signalwerte ungleich Null auf. Dabei stellt $x(k) \in \{x(k) : x \in \mathbb{R}; k \in \mathbb{Z}\}$ ein reelles, zeitdiskretes Signal dar, welches durch die Abtastwerte des zeitkontinuierlichen Signals definiert ist, d.h. $x(k) := x_A(k\Delta t)$. Die Signalwerte werden oft als Amplitude bezeichnet und in normierter Form dargestellt. Abgetastete, zeitkontinuierliche Signale werden im Weiteren mit $x(k\Delta t)$ bezeichnet. Ist die Abtastperiode konstant, spricht man von einer äquidistanten Abtastung.

Die durch eine Abtastung erhaltenen Sprachsignale können dann durch Methoden der digitalen Signalverarbeitung analysiert und manipuliert werden. Abbildung 1.3 zeigt ein solches Sprachsignal, welches mit 44,1 kHz abgetastet wurde. Die normierte Amplitude ist dabei proportional zum Schalldruckpegel des ursprünglichen Signals. (Hoffmann und Wolff, 2014)

Aus den abgetasteten Sprachsignalen kann durch Einsatz spezieller Algorithmen die Grundfrequenz bestimmt werden. Die Funktionsweise eines solchen Algorithmus ist in Kapitel 3 näher beschrieben. Die vom Algorithmus bestimmte Grundfrequenz stellt sich

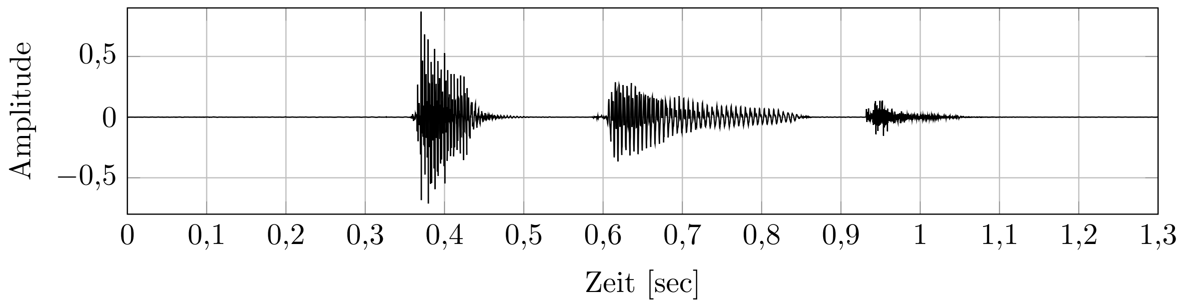

Abbildung 1.3: Digitales Sprachsignal der Äußerung *Abbild* [ˈap͡b̥ɪlt].

ebenfalls wieder als ein abgetastetes Zeitsignal $f_0(k\Delta t) \in \mathcal{X}$ mit zugehörigen Abtastintervall dar, wobei die Amplituden als Frequenzwerte interpretiert werden müssen. In der Sprachsignalverarbeitung ist es üblich Frequenzwerte in Halbtönen (engl. Semitones [st]) anzugeben, da diese normierte Skala sprecherunabhängig ist und sich damit besser für Vergleiche eignet als die Hertz-Skala. Die Halbton-Skala stellt ein normiertes, logarithmisches Maß dar, wobei meist auf 1 Hz normiert wird, und ist folgendermaßen definiert.

$$f_{0(\mathrm{st})} = 12 \cdot \log_2 \left(\frac{f_{0(\mathrm{Hz})}}{1\mathrm{Hz}} \right) \tag{1.1}$$

Die Grundfrequenz des Signals aus Abbildung 1.3 ist in Abbildung 1.4 visualisiert. Das Signal weist hierbei eine Abtastrate von $f_A = 100$ Hz auf. Es ist deutlich zu erkennen, dass die Grundfrequenz nur in stimmhaften Bereichen definiert ist, d.h. dort, wo das zugehörige Sprachsignal harmonische Schwingungen aufweist. Die Abtastwerte der stimmlosen Abschnitte haben einen nicht definierten Wert und werden bei entsprechenden Berechnungen nicht berücksichtigt. Zur mathematischen Beschreibung sei daher die Menge aller Indizes innerhalb stimmhafter (engl. voiced) Signalabschnitte durch $\mathcal{V} := \{k : k \in \mathbb{Z}; f_0(k) \in \mathbb{R}_{++}\}$ beschrieben, welche damit alle Indizes umfasst, für die die jeweilige f_0 einen wohldefinierten Wert aufweist. Das kleinste Element dieser Menge sei im Besonderen durch den Index $k_{\mathrm{v}0}$ beschrieben. Für das gegebene Beispiel gilt somit $k_{\mathrm{v}0}\Delta t = 0{,}375$ sec und $f_0(k_{\mathrm{v}0}\Delta t) = 94{,}3$ st.

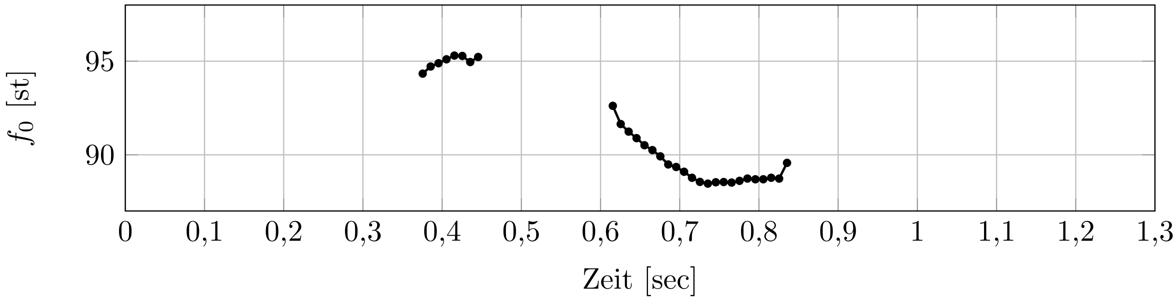

Abbildung 1.4: Grundfrequenzverlauf des Sprachsignals aus Abbildung 1.3.

2 Stand der Technik

2.1 Grundfrequenzmodelle

Wie bereits im Grundlagenkapitel ausgeführt, existieren keine minimalen bedeutungsunterscheidenden Einheiten für die Prosodie. Aus diesem Grund wurden zahlreiche Modelle entwickelt, die alle jeweils unterschiedliche Aspekte der Prosodie hervorheben und deshalb ihre Berechtigung haben. Dennoch lässt sich feststellen, dass es der Forschung bisher noch nicht gelungen ist, ein universelles Modell der Prosodie zu entwickeln, was primär an ihrer vielseitigen Erscheinung liegt. In diesem Abschnitt werden hauptsächlich solche Modelle beschrieben, die den Fokus auf der Modellierung der Grundfrequenz bzw. Intonation legen. Die hier betrachteten Modelle lassen sich in phonologische, phonetische und quantitative Modelle unterscheiden. Phonologische Modelle versuchen, in Anlehnung an die Phoneme, die Intonation durch eine zeitliche Folge bestimmter Töne und Akzentformen zu modellieren. Diese Töne und Akzentformen stellen dabei jedoch keine minimalen bedeutungsunterscheidenden Einheiten dar, wodurch ihre Zuordnung ebenfalls nicht eindeutig ist. Phonetische Modelle hingehen versuchen den Prozess der Artikulation zu modellieren, durch den die Grundfrequenz entsteht. Die Modellierung erfolgt dabei anhand der konkreten, kontinuierlichen Grundfrequenzkontur im Gegensatz zu den diskreten Tönen der phonologischen Modelle. Eine Modellierung der Grundfrequenzkontur erfolgt ebenso bei den quantitativen Modellen. Im Unterschied zu den Phonetischen wird jedoch der Prozess der Artikulation vernachlässigt und durch eine rein quantitative Beschreibung versucht, die Grundfrequenzkontur im Sinne einer mathematischen Kurve zu approximieren.

Phonetische Modelle wurden in erster Linie dafür entwickelt, um real auftretende Grundfrequenzverläufe möglichst gut zu beschreiben. Somit versuchen diese Modelle den Prozess der Artikulation nachzuvollziehen und verständlich zu machen, wohingegen quantitative Modelle immer im Zusammenhang mit künstlich generierten Grundfrequenzverläufen, vor allem für die Sprachsynthese, entwickelt wurden. Die Parameter werden jeweils aus einem geeigneten Korpus natürlicher Äußerungen gelernt oder durch gewisse Regeln determiniert. Eine weitere Kategorisierung der Prosodiemodelle kann nach ihrer zeitlichen Struktur erfolgen. So lassen sich superpositionelle und sequentielle Modelle unterscheiden. Superpositionelle Modelle betonen eher den Aspekt der Suprasegmentalität der Prosodie, wohingegen sequentielle Modelle vielmehr die temporale Art der Grundfrequenzproduktion abbilden. Prinzipiell lässt sich feststellen, dass ein gutes Modell in der Lage sein sollte, alle natürlich auftretenden Grundfrequenzverläufe zu beschreiben, dabei jedoch nicht zu stark überbestimmt sein sollte und viele unnatürliche Konturen ebenfalls charakterisiert. Weiterhin sollte auch eine Eindeutigkeit zwischen Modellpara-

metern und erzeugter Grundfrequenzkontur vorliegen. Das Problem vieler Modelle ist, dass unterschiedliche Parameterkombinationen die selben Konturen ergeben können, was sich nachteilig auf ein etwaiges Training eines Lernverfahrens auswirkt.
Im Folgenden sind einige Prosodiemodelle zur Beschreibung der Grundfrequenz vorgestellt, die überwiegend Anwendung in praktischen Systemen zur Prosodiegenerierung gefunden haben. Das in der vorliegenden Arbeit entwickelte System zur Grundfrequenzvorhersage basiert auf dem Target-Approximation-Modell, weshalb dieses detaillierter beschrieben ist. (Taylor, 2009)

ToBI-Modell (Pierrehumbert, 1980)

Das *Tone-and-Break-Indices*-Modell (ToBI) beschreibt die Intonation als eine Folge spezieller Töne und Akzente, wobei zwischen Tonakzenten, Phrasentönen und Randtönen unterschieden wird. Es ist damit ein Tonsequenzmodell und gleichzeitig das bekannteste phonologische f_0-Modell. Die Betonung wird hierbei also nicht in Form einer Grundfrequenzkontur, sondern als Folge von Tönen und Akzenten modelliert. Die Elemente dieser Folge, im Weiteren als Label bezeichnet, setzen sich dabei aus zwei Grundtönen, nämlich Hochtöne (H) und Tieftöne (L), zusammen und können durch diese transkribiert werden. Die verschiedenen Label können mit einer oder mehreren Silben assoziiert werden. Tonakzente bestehen entweder aus einem oder zwei der Grundtöne, wobei Töne akzentuierter Silben immer mit einem Stern (*) gekennzeichnet sind. Die Menge der Tonakzente ist durch die Label $\{H^*, L^*, H^* + L, H + L^*, L^* + H, L + H^*\}$ gegeben. Phrasengrenzen werden durch Randtöne beschrieben, die durch ein Prozentzeichen (%) kenntlich gemacht sind, wovon genau zwei existieren und durch die Label $\{H\%, L\%\}$ gegeben sind. Phrasentöne kennzeichnen schließlich den Verlauf der Grundfrequenz zwischen dem letzten Tonakzent einer Phrase und dem Randton. Dabei ergeben sich wiederum genau zwei

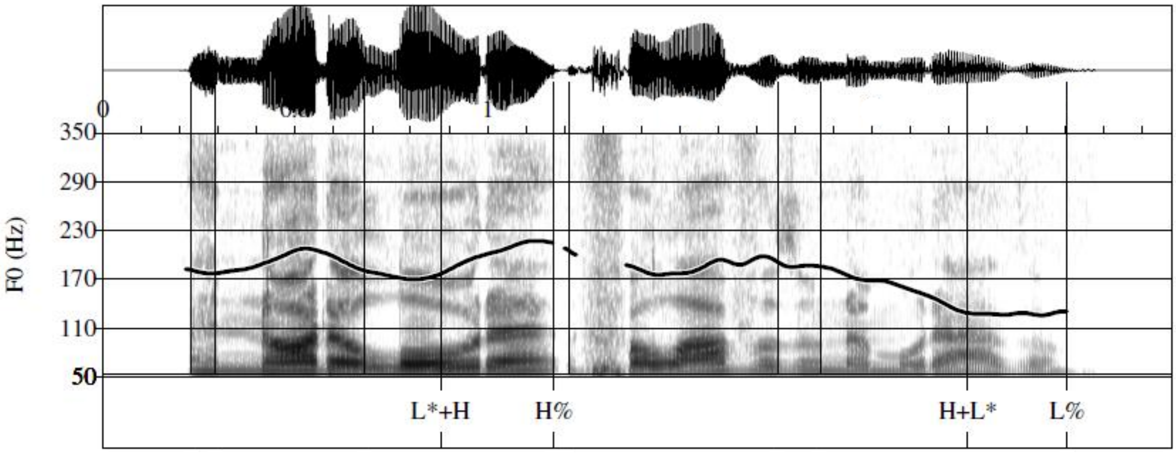

Abbildung 2.1: Annotierter Grundfrequenzverlauf nach dem ToBI-Modell aus Frota et al. (2015).

mögliche Töne, die durch ein Minuszeichen (-) charakterisiert werden, wodurch sich die Label $\{H-, L-\}$ ergeben.
Ein Beispiel einer annotierten Grundfrequenz nach dem ToBI-Modell ist in Abbildung 2.1 dargestellt. Für eine korrekte Annotation ist im Allgemeinen Expertenwissen und Erfahrung notwendig.

Fujisaki Modell (Fujisaki und Hirose, 1984)

Beim Fujisaki-Modell handelt es sich um ein klassisches phonetisches Modell, welches versucht den Prozess der Artikulation zu modellieren um natürlich auftretende Grundfrequenzverläufe zu erklären. Das Fujisaki-Modell modelliert die prosodischen Parameter Phrasierung und Akzentuierung direkt und geht davon aus, dass diese den Grundfrequenzverlauf im Wesentlichen bestimmen. Der Grundfrequenzverlauf stellt sich hierbei als Superposition von Phrasen-, Akzent- und Basiskomponente dar. Die Basiskomponente spiegelt die minimale, sprecherspezifische Grundfrequenz wieder, welche durch die Schwingung der Glottis entsteht. Die Phrasenkomponente beschreibt eine globale Intonationskontur, wohingegen die Akzentkomponente lokal akzentuierte Silben charakterisiert. Die Steuerung der Phrasenkomponente erfolgt durch sogenannte Phrasenkommandos, die als Impulse definiert sind und eine feste zeitliche Position aufweisen. Die Akzentkomponente wird durch Akzentkommandos gesteuert, welche durch eine passend positionierte Rechteckfunktion definiert ist. Die Filterung der jeweiligen Kommandos durch einen kritisch gedämpften Tiefpass zweiter Ordnung erzeugt dann die charakteristischen Grundfrequenzkomponenten. Mathematisch lässt sich das Fujisaki-Modell damit durch folgende Gleichungen beschreiben.

$$\ln f_0(t) = \ln f_\mathrm{b} + \sum_{i=1}^{I} A_{\mathrm{p}i} G_\mathrm{p}(t - T_{0i}) + \sum_{j=1}^{J} A_{\mathrm{a}j}[G_\mathrm{a}(t - T_{1j}) - G_\mathrm{a}(t - T_{2i})] \tag{2.1}$$

$$G_\mathrm{p}(t) = \begin{cases} \alpha^2 t e^{-\alpha t} & \text{für } t \geq 0 \\ 0 & \text{für } t < 0 \end{cases} \qquad G_\mathrm{a}(t) = \begin{cases} 1 - (1 + \beta t) e^{-\beta t} & \text{für } t \geq 0 \\ 0 & \text{für } t < 0 \end{cases} \tag{2.2}$$

f_b	Wert der Basiskomponente
I, J	Anzahl der Phrasen- bzw. Akzentkomponenten
$G_\mathrm{p}(t), G_\mathrm{a}(t)$	Phrasen- bzw. Akzentfilter
α, β	Dämpfungsfaktoren der Filter
$A_{\mathrm{p}i}$	Amplitude der i-ten Phasenkomponente
T_{0i}	Zeitpunkt der i-ten Phasenkomponente
$A_{\mathrm{a}j}$	Amplitude der j-ten Akzentkomponente
T_{1j}, T_{2j}	Start- und Endzeit der j-ten Akzentkomponente

Da das Fujisaki-Modell den Prozess der Artikulation modelliert und auch davon abgeleitet ist, lässt es sich demnach physiologisch interpretieren, wobei sich die nachfolgenden

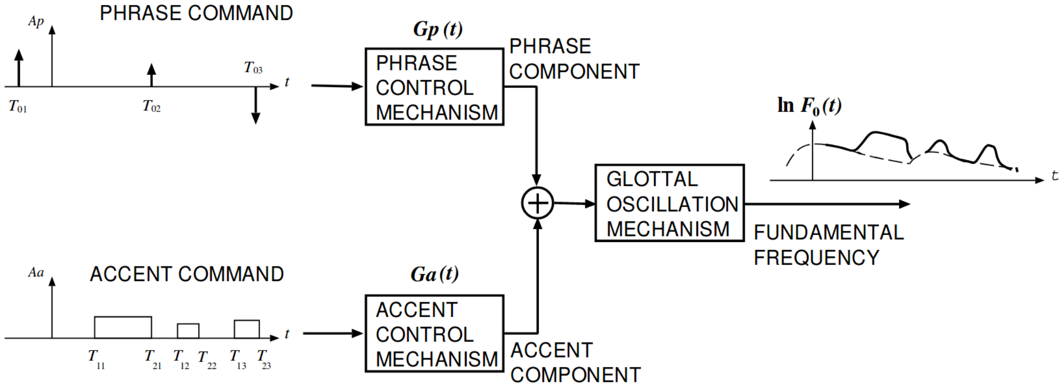

Abbildung 2.2: Blockdiagramm des Fujisaki-Modells aus Mixdorff (1998).

Beschreibungen nach Mixdorff (1998) richten. Die Grundfrequenz wird durch die Schwingung der Stimmlippen erzeugt, welche veränderlich in ihrer Oberflächenspannung und Dehnung bzw. Dicke ist. Spannung und Dicke der Glottis werden über die Kehlkopfmuskulatur eingestellt und damit die Grundfrequenz. Diese Muskulatur wird über neuronale Impulse gesteuert und übt im Wesentlichen zwei Kräfte auf die Glottis aus, die sich als eine Rotations- und eine Translationskomponente darstellen. Fujisaki und Hirose (1984) haben gezeigt, dass die logarithmierte Grundfrequenz proportional zur Glottisdicke mit einem additiven, konstanten Term ist. Diese Konstante wird beim Fujisaki-Modell durch die Basiskomponente modelliert. Weiterhin wird die Rotationskomponente mit der Akzentkomponente assoziiert und die Translationskomponente folglich mit der Phrasenkomponente, was insgesamt die Veränderung der Glottis durch die Kehlkopfmuskulatur modelliert. Eine schematische Darstellung des Modells ist in Abbildung 2.2 zu sehen.

Tilt-Modell (Taylor, 1994)

Das Tilt-Modell ist ein klassisches quantitatives Modell zur Beschreibung der Grundfrequenzkontur, das hauptsächlich für die praktische Anwendung in der Sprachsynthese entwickelt wurde. Quantitative Modelle versuchen eine parametrierte, mathematische, meist silbenbezogene Funktion eines Grundfrequenzabschnitts zu definieren, die auf der einen Seite alle möglichen, natürlichen Grundfrequenzverläufe gut abbilden kann und auf der anderen Seite möglichst wenige Parameter aufweist. Optimalerweise sollten diese freien Parameter möglichst gut mit verschiedenen, prosodischen Parametern korreliert sein, da dies vorteilhaft für die Grundfrequenzvorhersage sein kann.

Beim sogenannten rise/fall/connection-Modell, welches Taylor (2009) zusammenfassend als Grundlage des Tilt-Modells beschreibt, wird der Grundfrequenzverlauf einer Silbe durch vier Parameter codiert, die auch in Abbildung 2.3 dargestellt sind. Dabei wird zwischen Amplitude und Dauer eines monoton steigenden und fallenden Abschnitts unterschieden. Dieses Modell versucht in erster Linie die im Grundfrequenzverlauf charakteristischen Akzente abzubilden. Solche Charakteristika in der f_0-Kontur werden dabei

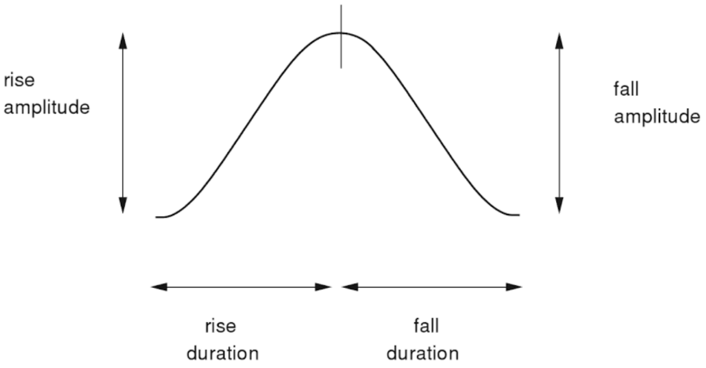

Abbildung 2.3: Ereignisbezogener Grundfrequenzabschnitt nach dem rise/fall/connection Modell aus Taylor (2009).

als Ereignisse (engl. Events) bezeichnet, wobei jedem Ereignis eine Silbe zugeordnet wird, aber nicht zwangsweise jeder Silbe ein Ereignis. Das auf diesem Modell aufbauende Tilt-Modell versucht den Parametersatz zu reduzieren und gleichzeitig diese Parameter linguistisch interpretierbar zu machen. Dazu werden zwei neue Parameter wie folgt definiert.

$$\mathrm{tilt}_{\mathrm{amp}} = \frac{|A_{\mathrm{rise}}| - |A_{\mathrm{fall}}|}{|A_{\mathrm{rise}}| + |A_{\mathrm{fall}}|} \qquad \mathrm{tilt}_{\mathrm{dur}} = \frac{|D_{\mathrm{rise}}| - |D_{\mathrm{fall}}|}{|D_{\mathrm{rise}}| + |D_{\mathrm{fall}}|} \tag{2.3}$$

Der Parameter $\mathrm{tilt}_{\mathrm{amp}}$ kann als ein Maß für die Akzentuierung interpretiert werden. Es zeigt sich, dass die beiden Parameter aus den Gleichungen (2.3) stark miteinander korrelieren und zu einem Parameter zusammengefasst werden können, welcher dann ein Maß für die Neigung (engl. Tilt) der f_0-Kontur darstellt und als $\mathrm{tilt} = (\mathrm{tilt}_{\mathrm{amp}} + \mathrm{tilt}_{\mathrm{dur}})/2$ gegeben ist. In Abbildung 2.4 sind verschiedene ereignisbezogene Grundfrequenzabschnitte für einen variierenden Tilt-Parameter veranschaulicht. Die Berechnung von Grundfrequenzverläufen aus den Tilt-Parametern kann durch Einsatz spezieller Filter erfolgen.
Als weitere quantitative Modelle lassen sich beispielsweise zwei stützstellenbasierte Modelle nennen. Das Modell von Traber (1991) definiert dabei pro Silbe acht Stützstellen und damit acht Parameter. Das Modell von Bailly und Holm (2005) definiert drei Stützstellen und einen Streckungsfaktor pro Silbe und kommt dabei mit vier Parametern aus. Jedoch werden mehrere Grundfrequenzkonturen, die unterschiedliche prosodische Funktionen abbilden, überlagert, um die Suprasegmentalität zu modellieren.

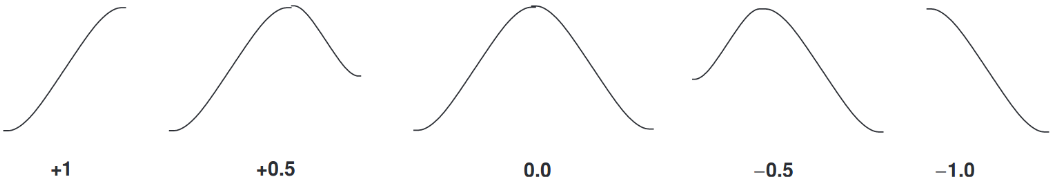

Abbildung 2.4: Ereignisbezogene Grundfrequenzabschnitte für unterschiedliche Tilt-Parameterwerte aus Taylor (2009).

Target-Approximation-Modell (Xu und Wang, 2001)

Das *Target-Approximation-Modell* (TAM) ist wie das Fujisaki-Modell ein phonetisches und versucht den Artikulationsprozess zur Erzeugung der Grundfrequenz abzubilden. Andererseits ist es silbenweise definiert und weist wenige freie Parameter auf, wodurch es, ähnlich wie die quantitativen Modelle, gut für die Grundfrequenzvorhersage geeignet ist. Das Modell wurde zur Beschreibung der Grundfrequenz im Mandarin-Chinesischen entwickelt. Chinesisch ist eine tonale Sprache, was bedeutet, dass nicht nur die einzelnen Laute, sondern auch die Betonung bzw. Töne bedeutungstragend sind. Die Änderung der Betonung eines Wortes kann also auch dessen Bedeutung ändern, wodurch der Intonation eine besondere Rolle zukommt. Für die Prosodiegenerierung und Sprachsynthese im Allgemeinen ergeben sich dadurch weitere Schwierigkeiten. Im Mandarin-Chinesischen existieren fünf lexikalische Töne zur Beschreibung der unterschiedlichen bedeutungstragenden Töne. Diese fünf Töne werden als hoch (engl. high, H), steigend (engl. rising, R), tief (engl. low, L), fallend (engl. falling, F) und neutral (N) kategorisiert und in dieser Reihenfolge entsprechend als Erster, Zweiter, Dritter, Vierter und Fünfter Ton benannt. Das TAM geht davon aus, dass sich die lexikalischen Töne des Mandarin-Chinesischen im Artikulationsprozess widerspiegeln und diesen steuern. Die modellhaften, artikulatorischen Realisierungen der Töne werden als *Pitch-Target*s (PT) bezeichnet. Es wird dabei zwischen zwei Arten von PTs unterschieden und mittels eckiger Klammer gekennzeichnet, nämlich dynamische ([rise], [fall]) und statische ([low], [mid], [high]) Targets. Im Rahmen des TAM wird angenommen, dass jeder lexikalische Ton durch ein konkretes Target realisiert wird, wobei Abbildung 2.5 die Zuordnung zwischen Tönen und PTs sowie deren Bezeichner verdeutlicht.

So wie eine Sequenz von Phonemen gesprochene Laute einer Äußerung eindeutig codiert, wird durch die Sequenz von PTs der Grundfrequenzverlauf einer Äußerung eindeutig codiert bzw. charakterisiert. PTs, als Realisierung der lexikalischen Töne, können also genutzt werden, um Grundfrequenzverläufe quantitativ abzubilden. Ein PT definiert silbenweise den Verlauf der Grundfrequenzkontur und kann durch eine lineare Funk-

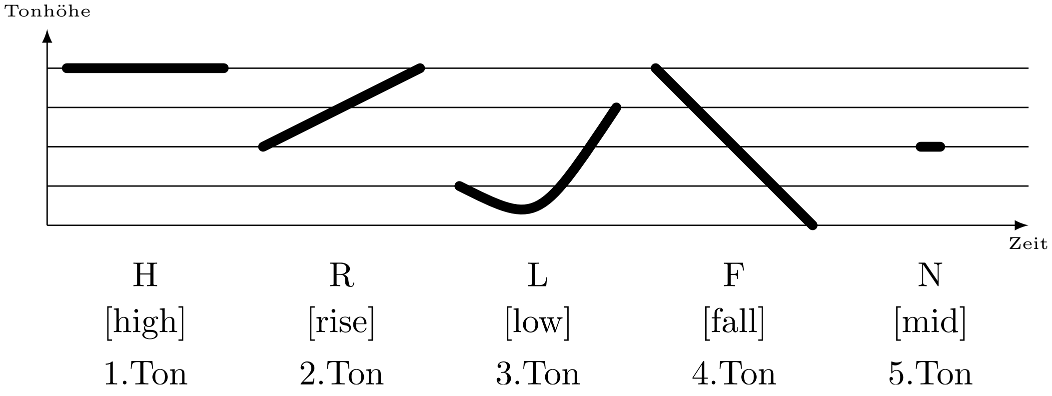

Abbildung 2.5: Darstellung und Bezeichnungen der fünf lexikalischen Töne des Mandarin-Chinesischen sowie deren Zuordnung zu den PTs des TAM.

tion $p(t) = mt + b$ beschrieben werden, welche über die Silbendauer d definiert ist. Die Parameter der linearen Funktion, Anstieg m und Verschiebung b, bestimmen somit die Kategorie des beschriebenen Targets und damit den Verlauf der Grundfrequenz. Betrachtet man jedoch natürliche Grundfrequenzverläufe, so lässt sich erkennen, dass die artikulierten, lexikalischen Töne nicht oder nur teilweise im f_0-Verlauf erkennbar sind, sich der Grundfrequenzverlauf aber keinesfalls als eine Aneinanderreihung linearer Targets darstellt. Xu und Wang (2001) führen diese Beobachtung auf den Artikulationsprozess zurück, wie im Folgenden beschrieben wird.

Die Erzeugung der Grundfrequenz erfolgt durch die Schwingung der Stimmlippen, welche hauptsächlich von der Dicke und Spannung derselben abhängt, aber auch vom subglottalen Druck. Die Spannung der Stimmlippen wird im Wesentlichen durch zwei Muskeln der Kehlkopfmuskulatur, Musculus cricothyroideus und Musculus thyroarytaenoideus, gesteuert, deren ausgeübte Kräfte einander entgegenwirken. Eine Erhöhung der Oberflächenspannung durch Wirken der Muskeln verursacht somit eine höhere Grundfrequenz, eine verringerte Spannung eine dementsprechend niedrigere f_0. Ein solches biomechanisches System von entgegenwirkenden Muskelpaaren, welche Energie hin- und hertransportieren, kann als ein lineares System Nter Ordnung beschrieben werden. Das TAM nimmt infolgedessen an, dass die beschriebenen PTs dem Artikulationsprozess unterliegen und diesen steuern. Die Kehlkopfmuskulatur, charakterisiert durch ein System Nter Ordnung, weist jedoch eine Verzögerung auf. Dies äußert sich im Grundfrequenzverlauf durch eine Kontur, die sich mit zunehmender Zeit dem unterliegenden Target asymptotisch nähert, was durch die Zeitkonstante des Systems gesteuert wird. Im Rahmen des TAM wird bei der mathematischen Beschreibung üblicherweise die inverse Zeitkonstante verwendet, die hier als Annäherungsrate $\boldsymbol{\lambda}$ bezeichnet wird. Das Prinzip der asymptotischen Näherung ist in Abbildung 2.6 verdeutlicht. Mathematisch lässt sich dieses Verhalten durch ein lineares, kritisch gedämpftes System Nter Ordnung mit Tiefpasscharakter modellieren. Prom-On et al. (2009) verwendeten zweckmäßigerweise ein Filter dritter Ordnung bei einer quantitativen Umsetzung des Modells, welches durch nachfolgende Gleichungen beschrieben ist.

$$f_0(t) = \overbrace{(mt + b)}^{\text{Pitch-Target}} + \overbrace{(c_0 + c_1 t + c_2 t^2) e^{-\boldsymbol{\lambda} t}}^{\text{Systemantwort}} \tag{2.4}$$

$$c_0 = f_0(0) - b \qquad c_1 = f_0'(0) + c_0\boldsymbol{\lambda} - m \qquad c_2 = \frac{1}{2}(f_0''(0) + 2c_1\boldsymbol{\lambda} - c_0\boldsymbol{\lambda}^2)$$

Im Mandarin-Chinesischen ist jede Silbe durch einen der fünf beschriebenen Töne gekennzeichnet. Dies legt die Annahme nahe, dass die zugeordneten PTs ebenfalls silbenweise definiert werden. Charakteristisch für Grundfrequenzverläufe ist, dass diese nur abschnittsweise definiert sind, nämlich nur dort, wo stimmhafte Laute artikuliert werden. Streng genommen sollten also auch die Targets nur an Stellen stimmhafter Laute definiert sein. Laut Prom-On et al. (2009) belegen empirische Studien jedoch, dass die f_0-Kontur sich eher vom Zeitpunkt des Silbenbeginns als vom Zeitpunkt des beginnenden stimmhaften Bereichs an das Pitch-Target annähert und äquivalent am Silbenende endet anstatt am Ende des stimmhaften Bereichs innerhalb der Silbe. Aus diesem Grund wird für das TAM angenommen, dass der Grundfrequenzverlauf an den Silbengrenzen

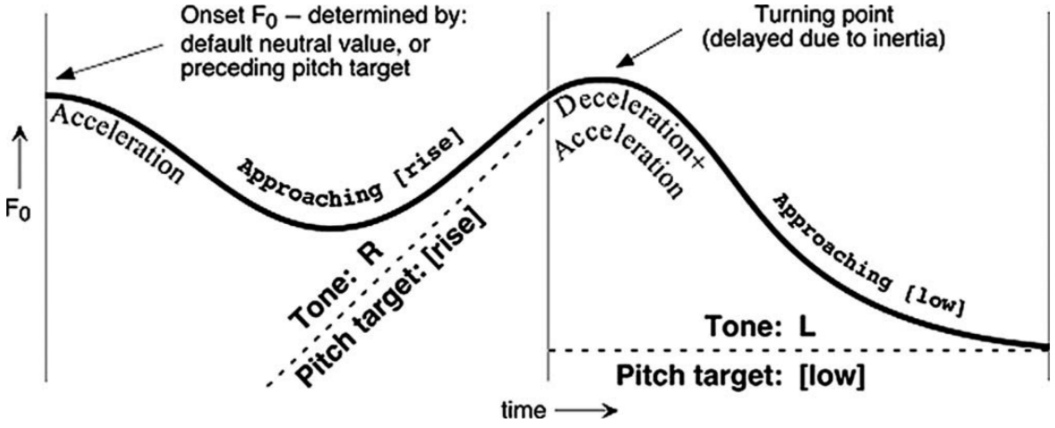

Abbildung 2.6: Grundfrequenzverlauf nach dem TAM aus Prom-On et al. (2009).

synchronisiert ist, d.h., der Zustand der f_0 wird an den Silbengrenzen übertragen. Die so synchronisierte, sequentielle Folge von Targets ist damit in der Lage die Suprasegmentalität der Prosodie zu erklären, da nicht jedes PT als separate, lokale prosodische Einheit betrachtet wird, sondern durch die Synchronisation den gesamten, folgenden Artikulationsprozess beeinflusst.

Das beschriebene TAM eignet sich gut für eine artikulatorische Modellierung der Grundfrequenz, macht jedoch keine Aussage über abstraktere Prozesse der Sprachproduktion, die den Verlauf der Grundfrequenz bestimmen. Ein von Xu (2004) entwickeltes, auf dem TAM aufbauendes Modell erfüllt genau diesen Zweck und schafft einen Zusammenhang zwischen Mechanismen der Artikulation und Prozessen der Sprachproduktion bzgl. der f_0. Das *Parallel-Encoding-Target-Approximation*-Modell (PENTA) versucht dabei nicht nur die Töne, sondern auch andere prosodische Parameter wie Akzentuierung oder Fokus zu modellieren. Die Realisierung prosodischer Parameter werden im PENTA-Modell als kommunikative Funktionen bezeichnet, wobei angenommen wird, dass jeder Silbe ein Wert für eine kommunikative Funktion zugeordnet werden kann und mehrere Funktionen parallel betrachtet werden können. Ein nach diesem Schema annotiertes Sprachsignal mit zugehöriger Grundfrequenz ist in Abbildung 2.7 dargestellt. Das PENTA-Modell geht davon aus, dass zwischen den Kategorien der parallel betrachteten kommunikativen Funktionen und den Parametern des TAM (Anstieg, Verschiebung, Annäherungsrate) ein Zusammenhang besteht.

Ein solcher quantitativer Zusammenhang wurde von Prom-On et al. (2009) sowie Xu und Prom-On (2014) untersucht, wobei jeder möglichen Kombination von Kategorien der parallelen, kommunikativen Funktionen pro Silbe genaue Werte der TAM-Parameter zugeordnet wurden. Als Grundlage diente ein zweckmäßig annotiertes Korpus von natürlichen Äußerungen. Mit speziellen Algorithmen wurde der Grundfrequenzverlauf aus den natürlichen Äußerungen bestimmt, die dann wiederum zur Ermittlung des gesuchten Zusammenhangs verwendet werden konnten. Mithilfe des ermittelten Zusammenhangs konnten so also Grundfrequenzverläufe erzeugt werden, die den natürlichen sehr nahe kommen und das allein auf Basis von zwei oder drei kommunikativen Funktionen, was

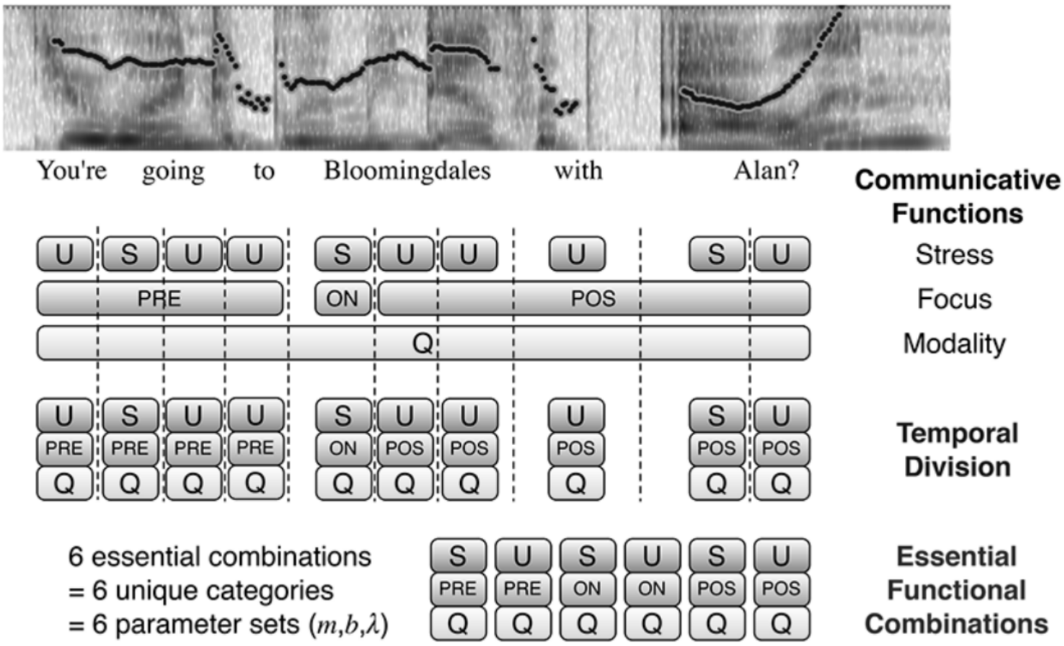

Abbildung 2.7: Realisierung des PENTA-Modells aus Xu und Prom-On (2014).

die Leistungsfähigkeit des Modells unterstreicht. Die Kombination von Kategorien der parallelen, kommunikativen Funktionen wird im Folgenden als funktionale Kombination bezeichnet.

Zur Ermittlung des Zusammenhangs zwischen funktionaler Kombination und Werten der TAM-Parameter wurden die Tools PENTAtrainer1 (Prom-On et al., 2009; Xu und Prom-On, 2015) und PENTAtrainer2 (Xu und Prom-On, 2014) entwickelt, welche im Wesentlichen zwei Hauptfunktionen implementieren. Zum einen wird eine Oberfläche zur Annotation von Äußerungen bereitgestellt, um die Kategorien der kommunikativen Funktionen pro Silbe zeitlich festzulegen. Zum anderen wird für jede mögliche funktionale Kombination ein optimales PT geschätzt, wobei die Schätzung anhand der von Praat bestimmten Grundfrequenzverläufe natürlicher Äußerungen stattfindet. Beim PENTAtrainer1 werden dazu zuerst die optimalen Targets aus den natürlichen Äußerungen geschätzt, wobei diejenigen ausgewählt werden, die eine f_0 erzeugen, die der natürlichen am nächsten kommen. Bei diesem Vorgang wird die natürliche f_0 in den stimmlosen Abschnitten interpoliert und anschließend das gesamt Signal geglättet, um ein durchgängig definiertes Signal als Grundlage der Optimierung zu erhalten. Die Optimierung selbst erfolgt durch eine erschöpfende Suche über einem Gitter ganzer Zahlen. Als Optimierungskriterium dient der summierte quadratische Fehler (engl. *Sum of Squared Error*, SSE) zwischen der von Praat bestimmten, natürlichen f_0-Kontur und der durch die Folge von PTs erzeugten, modellierten Grundfrequenzkontur. In der Sprachsignalverarbeitung ist eine solche iterative Optimierung von Modellparametern als Analyse-durch-Synthese Ansatz bekannt. Die Optimierung erfolgt hierbei sequentiell, für jede Silbe einzeln, wobei der jeweilige Zustand der f_0 an den Silbengrenzen übertragen wird. Ein mittels PENTAtrainer1 modellierter, optimaler Grundfrequenzverlauf ist in Abbildung 2.8 dargestellt.

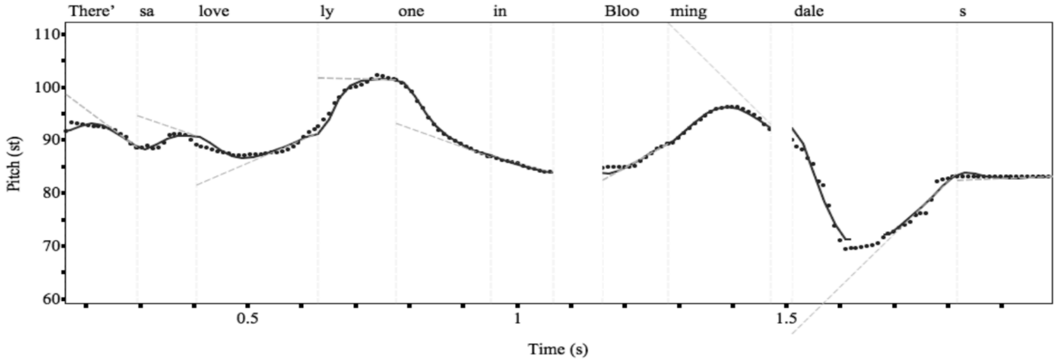

Abbildung 2.8: Optimale PTs und modellierter Grundfrequenzverlauf ermittelt mit PENTAtrainer1 aus Xu und Prom-On (2015).

Nach der Parametersuche erfolgt die Ermittlung des Zusammenhangs zwischen funktionalen Kombinationen und Pitch-Targets durch eine einfache Mittelwertbildung, wobei die Werte der PTs jeweils über alle Silben gemittelt werden, welche die selbe funktionale Kombination aufweisen. Im PENTAtrainer2 hingegen wird ein stochastisches Optimierungsverfahren eingesetzt, um den Zusammenhang zu bestimmen. Zufällig wird jeder funktionalen Kombination ein konkretes PT zugeordnet und der Fehler zwischen modellierten und natürlichen Grundfrequenzverläufen über das gesamte Korpus gemessen. Durch ein iteratives Verfahren werden die Targets solange verändert, bis ein Minimum für den Gesamtfehler gefunden ist.

2.2 Grundfrequenzvorhersage

Wie bereits im ersten Kapitel besprochen, stellt die Grundfrequenz einen entscheidend Parameter für die wahrgenommene Natürlichkeit von Sprache dar. In Text-zu-Sprache (engl. *Text-to-Speech*, TTS) Systemen ist die Vorhersage der Grundfrequenz also entscheidend für die Qualität der synthetisierten Sprache. Da im Text selbst wenig Information über die Prosodie codiert ist, stellt sich die Vorhersage der Prosodie, im speziellen der Grundfrequenz, rein aus Textinformation als schwierig heraus. Der erste Verarbeitungsschritt in den meisten TTS Systemen ist eine syntaktische Analyse des Texts, durch die linguistische Informationen über den Text gewonnen werden können. Die so gewonnenen Informationen über Wortarten, Wortpositionen, Silbenstruktur, Satzzeichen usw. können für die Grundfrequenzvorhersage genutzt werden. In einem weiteren Verarbeitungsschritt kann eine Graphem-zu-Phonem Konvertierung erfolgen, wodurch lautbezogene Informationen gegeben durch Phonemfolgen, aus dem Text extrahiert werden können. All diese Informationen stehen dann für die Vorhersage der Grundfrequenz und anderer prosodischer Parameter zur Verfügung. Folgendes Schema visualisiert diesen allgemeinen Aufbau eines TTS Systems.

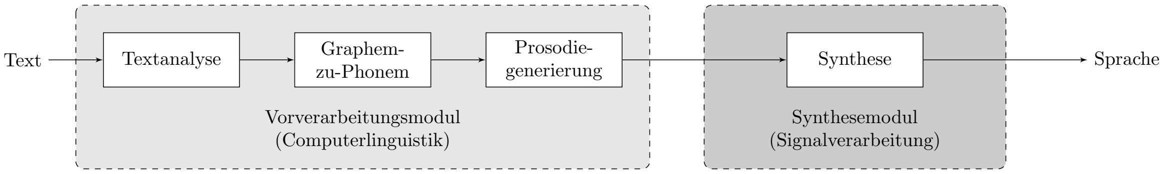

Abbildung 2.9: Allgemeiner Aufbau eines TTS Systems nach Balyan et al. (2013).

Im Prinzip stellt sich die Grundfrequenzvorhersage als ein Regressionsproblem dar, welches eine explizite Abbildung von den aus der Textverarbeitung gewonnenen linguistischen Informationen auf einen charakteristischen Grundfrequenzverlauf berechnet. Eine direkte Berechnung der Grundfrequenzkurve, d.h. die Vorhersage der einzelnen Abtastwerte, stellt sich jedoch als problematisch heraus, da die Anzahl der Abtastwerte pro Silbe stark variiert und mögliche Zusammenhänge zwischen diesen vernachlässigt werden. Ferner ist bei einem solchen Ansatz der Ausgangsraum sehr groß, was eine präzise Vorhersage erschwert. Aus diesen Gründen wird oft ein Grundfrequenzmodell zur Hilfe genommen, welches in der Lage ist, Grundfrequenzverläufe aus einem Satz von Parametern zu generieren. Dadurch können also die Modellparameter anstatt der Abtastwerte vorhergesagt werden, was die Dimension des Problems deutlich reduziert. Ein weiterer Vorteil bei der Verwendung eines Grundfrequenzmodells ist, dass sich die Modellparameter gezielt manipulieren lassen, um bestimmte Effekte in der synthetisierten Sprache zu erreichen, wie beispielsweise das Einbringen spezieller Emotionen. Korrelieren die Parameter des Modells darüber hinaus mit bestimmten Prosodieparametern, können letztere damit auch gezielt manipuliert werden. Die Vorhersage selbst wird davon nicht beeinflusst, da die Manipulation in einem Nachverarbeitungsschritt erfolgt. Dieses Gebiet ist Gegenstand aktueller Forschung und wird bisher kaum oder gar nicht in TTS Systemen berücksichtigt.

Prinzipiell lassen sich beliebige Grundfrequenzmodelle für die Vorhersage nutzen, wobei Modelle, welche die Grundfrequenzkontur direkt modellieren, von Vorteil sind. Wird beispielsweise ein Tonsequenzmodell verwendet, müssen zunächst die Label der Tonmuster bestimmt werden, aus welchen dann wiederum erst eine Grundfrequenzkontur geschätzt werden kann. Dies macht das Problem unnötig komplex und anfälliger für Fehler.

Die Entwicklung von Systemen zur Grundfrequenzvorhersage begann in den 1980ern und konnte bis heute noch nicht zufriedenstellend gelöst werden. Die verschiedenen Lösungen lassen sich in zwei Kategorien unterteilen: regelbasierte und datengetriebene Ansätze. Diese sollen im Folgenden kurz besprochen werden. Aktuelle TTS Systeme stützen sich vorrangig auf datengetriebene Ansätze, die sich zumeist als Regressionsproblem beschreiben lassen. (Taylor, 2009)

Regelbasierte Systeme

Regelbasierte Ansätze versuchen Charakteristika in natürlichen Grundfrequenzverläufen zu identifizieren, um darauf aufbauend einen Satz von Regeln abzuleiten, der zur Generierung möglichst natürlicher f_0-Verläufe genutzt werden kann. Anderson et al. (1984) beschrieben erstmals ein solcher Ansatz und implementierten diesen. Als Modell wurde das besprochene ToBI-Modell verwendet und damit versucht, aus den ToBI-Label f_0-Verläufe vorherzusagen. Der in dieser Veröffentlichung abgeleitete Satz von Regeln ordnet jedem Label eine bestimmte Anzahl an Abtastwerten der f_0-Kurve zu, welche dabei auf einen vorher festgelegten Wertebereich begrenzt sind. Die Kontur wird dann schließlich durch einfache, lineare Interpolation bestimmt und anschließend geglättet. Dieser Ansatz wurde in zahlreichen Arbeiten weiterentwickelt, wobei beispielsweise die Vorhersage nicht auf einzelnen, isolierten Label basiert, sondern auch vorherige und folgende mit betrachtet. Davon abgesehen wurden auch andere Interpolations- und Glättungsverfahren eingesetzt, um die f_0-Kontur zu generieren. Eine ausführliche Beschreibung eines solchen weiterentwickelten Ansatzes ist beispielsweise in Jilka et al. (1999) zu finden. Auf diese Art und Weise arbeiten auch einige aktuelle Systeme, so beispielsweise die quelloffene Software Mary TTS (Schröder und Trouvain, 2003). In diesem System wurden die Regeln durch eine automatische Korpusanalyse abgeleitet.

Ein weiterer regelbasierter Ansatz wurde von Mixdorff (1998) entwickelt. Dieser basiert auf einem für die deutsche Sprache angepassten Fujisaki-Modell und versucht die Modellparameter aus geeigneter linguistischer Information vorherzusagen. Aus natürlichen Grundfrequenzverläufen wurden dabei wiederum gewisse Regeln abgeleitet, die abhängig von der Eingangsinformation typische Werte für die Fujisaki-Parameter angeben. Eine Umsetzung dieser Art der Prosodiegenerierung erfolgte im Sprachsynthesesystem DreSS, welches an der TU Dresden entwickelt wurde.

Datengetriebene Systeme

Aufgrund der großen Variabilität natürlich auftretender Grundfrequenzverläufe kommen die regelbasierten Ansätze schnell an ihre Grenzen, da eine riesige Anzahl an Regeln abgeleitet werden muss, um gute Vorhersagen zu treffen. Der damit verbundene zeitliche Aufwand zur Ableitung und Implementierung der Regeln lohnt sich in den meisten Fällen nicht. Aus diesem Grund wurde bereits recht früh in der Forschung dazu übergegangen, datengetriebene Lösungen zu entwickeln, auch wenn diese mit teilweise hohen Rechenaufwand verbunden sind und große Korpora von Sprachdaten als Grundlage benötigen. Die heutzutage zur Verfügung stehende Rechenleistung begünstigt jedoch den Einsatz solcher Verfahren.

Erste Ansätze wurden aufbauend auf dem System NETtalk entwickelt (Scordilis und Gowdy, 1989). In dieser Veröffentlichung wurden zwei mehrschichtige, neuronale Netze verwendet, um aus einer Folge von Phonemen den Grundfrequenzverlauf vorherzusagen. Hierbei wurde jedoch kein Modell verwendet, sondern direkt die Abtastwerte der f_0 vorhergesagt. Da die Vorhersage der Grundfrequenz ein Problem mit einer zeitlichen Struktur ist, liegt es nahe rückgekoppelte Systeme zu verwenden, um diese Struktur zu erfassen und für die Vorhersage zu nutzen. Dieser Ansatz wurde erstmals von Traber (1991) verfolgt, wobei ein rekurrentes, neuronales Netz für die Grundfrequenzvorhersage genutzt wurde. Dieses System wurde in der Synthesesoftware SVOX implementiert. Als

f_0-Modell diente ein quantitatives, stützstellenbasiertes. Quantitative Modelle sind oft überbestimmt, wodurch sehr unnatürliche Grundfrequenzverläufe erzeugt werden können, weshalb phonologische und phonetische Modelle in der folgenden Zeit bevorzugt wurden. So beschreiben Black und Hunt (1996) die Vorhersage der f_0 Kontur aus den Label des Tonsequenzmodells mittels linearer Regression. Aber auch Jokisch et al. (2000) stellen ein System zur Vorhersage der Fujisaki-Parameter vor, welches auf einem mehrschichtigen Perzeptron basiert. Dieses System wurde darüber hinaus zu einem Hybriden weiterentwickelt, welches den regelbasierten und datengetriebenen Ansatz kombiniert. Beide Prosodiesysteme wurden ebenfalls im Rahmen der Synthesesoftware DreSS implementiert.

In der neueren Forschung liegt der Fokus auf der statistisch-parametrischen Sprachsynthese, welche versucht, die Parameter eines Vocoders auf Framebasis vorherzusagen, der dann wiederum das Sprachsignal generiert. Diese Verfahren haben jedoch große Eingangs- und Ausgangsräume und benötigen riesige Datenmengen in der Lernphase. Die Vorhersage der Grundfrequenz taucht dabei nicht mehr als ein separates Problem auf, sondern ist in den Gesamtkontext eingebettet. Bei dieser Art der Sprachsynthese werden primär Hidden-Markov-Modelle und Deep-Belief-Netzwerke eingesetzt. Die Generierung der Grundfrequenz erfolgt somit ebenfalls framebasiert, was bedeutet, dass aller 5 oder 10 msec ein Abtastwert für die f_0 berechnet wird. Es werden also keine expliziten f_0-Modelle mehr verwendet. Da die Grundfrequenz jedoch nur innerhalb stimmhafter Bereiche der Sprache definiert ist, existiert zumeist noch ein Flag, welches dies anzeigt. Alternativ können auch sogenannte Multi-Space-Wahrscheinlichkeitsverteilungen zur Modellierung genutzt werden, die diese Art des nicht Vorhandenseins der f_0-Kontur in stimmlosen Bereichen abbilden können.

Erste Arbeiten über diese Art der Grundfrequenzgenerierung sind in Yoshimura et al. (1999) zu finden, wobei ein Hidden-Markov-Modellansatz genutzt wird. Eine der ersten Arbeiten, welche die Verwendung von Deep-Belief-Netzwerken untersucht, stammt von Google (Ze et al., 2013). Ebenso implementiert die quelloffene Sprachsynthese-Software Merlin ein solches tiefes Netzwerk, welches u.a. die f_0 vorhersagt. In der aktuellen Forschung wird vornehmlich der Einsatz von bidirektionalen Long-Short-Term-Memories untersucht, die eine spezielle Form rekurrenter, neuronaler Netze darstellen und gut für Probleme mit ausgeprägter zeitlicher Struktur, wie es bei der Prosodiegenerierung der Fall ist, geeignet sind.

2.3 Regressionsverfahren

Regressionsanalysen beschreiben statistische Verfahren zur Untersuchung quantitativer Zusammenhänge zwischen einer abhängigen Variablen und einer oder mehreren unabhängige Variablen. Im Rahmen dieser Arbeit werden die Unabhängigen als Merkmale und die Abhängige als Messwert bezeichnet. Die Darstellungen in diesem Abschnitt stützen sich vor allem auf Grundlagenwerke über Maschinenlernen (Bishop, 2006; Russell und Norvig, 2012; Theodoridis, 2015).

Ziel der Regressionsanalyse ist die Ermittlung eines funktionalen Zusammenhangs zwischen Messwert und zugehörigen Merkmalen, welcher dann beispielsweise für Vorhersagen genutzt werden kann. Aus Sicht des Maschinenlernens bzw. der statistischen Lerntheorie folgt die Bestimmung des funktionalen Zusammenhangs bei der Regressionsanalyse dem Prinzip der Minimierung des empirischen Verlusts (engl. *Empirical Risk Minimization*, ERM). Als Grundlage dienen dabei Daten, aus denen ein solcher funktionaler Zusammenhang geschätzt bzw. gelernt werden soll. Ein Datensatz mit N Messwerten und zugehörigen L-dimensionalen Merkmalvektoren wird als Lernstichprobe oder Trainingsdatensatz bezeichnet und in folgender Weise beschrieben.

$$\mathcal{D} = \{(y^{(n)}, \mathbf{x}^{(n)}) : y \in \mathbb{R}; \mathbf{x} \in \mathbb{R}^L\} \quad n = 1...N \tag{2.5}$$

Die untersuchten Daten stammen in der Regel aus Messungen der realen Welt, was bedeutet, dass diese Daten einen Messfehler bzw. Rauschen enthalten. Würde man nun bekannte numerische Verfahren der Interpolation, wie beispielsweise die Polynominterpolation, auf solche Daten anwenden, führt dies zu einer Überanpassung.
Um also funktionale Zusammenhänge in verrauschten Daten zu bestimmen, ist eine Regressionsanalyse notwendig, welche durch das folgende, allgemeine Modell beschrieben werden kann.

$$y = f(\mathbf{x}) + e \tag{2.6}$$

Für eine gute Schätzung $\hat{y} = f(\mathbf{x})$ des funktionalen Zusammenhangs sollte die Modellfunktion f passend gewählt werden, sodass sie den Daten am besten entspricht. Dabei gibt es zwei unterschiedliche Ansätze, um die Modellfunktion für die Regression zu bestimmen, welche als parametrische bzw. parameterfreie Modelle bzw. Methoden bezeichnet werden. Bei der Regression mit parametrischen Modellen wird eine konkrete parametrierte Funktion als Modell angenommen, welche eine Familie von Funktionen $\mathcal{F} := \{f_{\boldsymbol{\theta}}(\cdot) : \boldsymbol{\theta} \in \mathbb{R}^L\}$ als mögliche Lösungen beschreibt. Bei der Regression mit parameterfreien Modellen wird hingegen kein konkreter funktionaler Zusammenhang vor der Schätzung angenommen, sondern dieser direkt aus der Lernstichprobe ermittelt. Mathematisch gesehen wird somit die Wahl der Modellfunktion auf einen passenden Funktionenraum $\mathbb{H}$ eingeschränkt, wobei sich spezielle Hilberträume als zweckmäßig erwiesen haben. In beiden Fällen erfolgt die Schätzung im Sinne der ERM, was der Minimierung einer Kostenfunktion $J_N(f) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{L}(y_n, f(\mathbf{x}_n))$, auch als empririscher Verlust bekannt, entspricht. Die Messung des Fehlers erfolgt durch eine Fehlerfunktion $\mathcal{L} : \mathbb{R}^2 \rightarrow \mathbb{R}_{++}$, die ein Abstandsmaß zwischen Messwerten und Modellfunktionswerten bietet. In vielen Anwendungen wird die quadratische oder absolute Differenz gewählt, um diesen Abstand zu definieren. Aber auch die Huber'sche Fehlerfunktion oder die ϵ-intensive Fehlerfunktion kommen oft zum Einsatz. Die unterschiedlichen Fehlerfunktionen sind in Abbildung 2.10 gegenübergestellt. Damit lässt sich das zu lösende Schätzproblem der Regressionsaufgabe im Sinne der ERM folgendermaßen definieren.

$$\min_{f} J_N(f) := \sum_{n=1}^{N} \mathcal{L}\Big(y_n, f(\mathbf{x}_n)\Big) \tag{2.7}$$

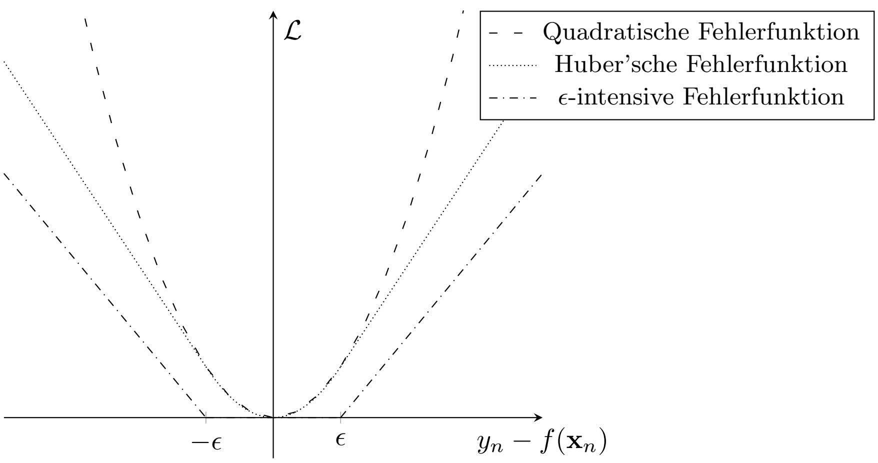

Abbildung 2.10: Beispiele verschiedener Fehlerfunktionen.

Dabei gilt je nach Verfahren entweder $f \in \mathcal{F}$ oder $f \in \mathbb{H}$. Verschiedene Regressionsverfahren unterscheiden sich nun in der Wahl der Modellfunktion f sowie der Fehlerfunktion $\mathcal{L}$. Im Folgenden sind mehrere Regressionsverfahren, die im Rahmen dieser Arbeit untersucht wurden, näher beschrieben. Kriterien zur Auswahl eines passenden Verfahrens für eine konkrete Problemstellung sind beispielsweise die Größe der Lernstichprobe, die Dimension der Merkmalvektoren, die zur Verfügung stehende Rechenleistung, die Struktur der Daten sowie eventuelles Vorwissen über die Lösung des Problems (Theodoridis, 2015).

Lineare Regression (um 1800)

Erste Regressionsanalysen wurden bereits von Gauß im Jahre 1809 zur Berechnung von Planetenlaufbahnen aus Beobachtungen eingesetzt. Dieses Verfahren ist heute als Kleinste-Quadrate-Methode bekannt. Die klassische lineare Regression basiert auf dieser Idee und stellt das einfachste Regressionsverfahren dar, wobei ein linearer Zusammenhang zwischen Merkmalvektoren und Messwerten angenommen wird, womit die Familie der Modellfunktionen als linear vorgegeben wird. Die Wichtung der einzelnen Merkmale erfolgt durch den Parametervektor $\boldsymbol{\theta} := (\theta_0, \theta_1, \ldots, \theta_L)^{\mathrm{T}} \in \mathbb{R}^{L+1}$, denn ein Gleichanteil θ_0 sollte bei der Modellierung nicht vernachlässigt werden. Aus Gründen einer kompakteren Formulierung sei in diesem Unterabschnitt der Merkmalvektor $\mathbf{x} := (1, x_1, \ldots, x_L)^{\mathrm{T}} \in \mathbb{R}^{L+1}$ in erweiterter Form definiert. An allen anderen Stellen sei auf die Definition verwiesen, die am Anfang des Abschnitts 2.3 gemacht wurde. Als Fehlerfunktion wird die quadratische Differenz gewählt. Damit ergibt sich als Kostenfunktion der bekannte mittlere quadratische Fehler (engl. *Mean Squared Error*, MSE).

$$f(\mathbf{x}) = \boldsymbol{\theta}^{\mathrm{T}}\mathbf{x} \tag{2.8}$$

$$\mathcal{L}(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2 \tag{2.9}$$

Mit diesen Informationen lässt sich das Optimierungsproblem zur Lösung des Regressionsproblems wie in Gleichung 2.10 beschreiben. Gleichung 2.11 zeigt die geschlossene Lösung dieses Problems, welche auch als Kleinste-Quadrate-Lösung bekannt ist. Im Folgenden sei $\mathbf{y} := (y_1, \ldots, y_n)^\mathrm{T} \in \mathbb{R}^N$ und $X := (\mathbf{x}_1^\mathrm{T}, \ldots, \mathbf{x}_N^\mathrm{T})^\mathrm{T} \in \mathbb{R}^{N\times(L+1)}$.

$$\begin{aligned}\hat{\boldsymbol{\theta}} &:= \arg\min_{\boldsymbol{\theta}} \sum_{n=1}^{N} (y^{(n)} - \boldsymbol{\theta}^\mathrm{T}\mathbf{x}^{(n)})^2 &\quad (2.10)\\ &= (X^\mathrm{T}X)^{-1}X^\mathrm{T}\mathbf{y} &\quad (2.11)\end{aligned}$$

Es lässt sich zeigen, dass unter der Annahme von weißem Gauß'schem Rauschen die beschriebene Lösung den gleichmäßig besten erwartungstreuen Schätzer (engl. *Minimum Variance Unbiased Estimator*, MVUE) darstellt und die Cramer-Rao-Schranke mit Gleichheit erfüllt. Das bedeutet jedoch nicht, dass dieser Schätzer optimal im Sinne des MSE Kriteriums ist. Gemäß dem Verzerrung-Varianz-Dilemma lässt sich ein solcher Schätzer nur verbessern, d.h. die Varianz des Schätzers verringern, indem man gleichzeitig die Verzerrung, auch Bias genannt, erhöht. Der erhaltene verzerrte Schätzer mit verringerter Varianz kann so einen geringeren MSE aufweisen als der MVUE. Der MSE lässt sich als Summe von Verzerrung und Varianz darstellen, wie im Folgenden symbolhaft angedeutet ist (Kay, 1993).

$$\mathrm{MSE} := \mathbb{E}[(y - f(\mathbf{x}))^2] \equiv \mathrm{Verzerrung}^2 + \mathrm{Varianz} + \sigma^2 \qquad (2.12)$$

Der Term σ^2 wird als irreduzibler Fehler bezeichnet und resultiert aus den verrauschten Messwerten und ist somit unabhängig von jeglicher Wahl der Modellfunktion. In der Praxis erreicht man eine Reduzierung der Varianz durch das Einbringen der Nebenbedingung $\|\boldsymbol{\theta}\|^2 \leq \varrho$ in das Optimierungsproblem aus 2.10. Berechnet man daraufhin die Lagrang'sche Funktion, um ein unbedingtes Optimierungsproblem zu erhalten, ergibt sich die folgende Form, welche wiederum eine geschlossene Lösung aufweist.

$$\begin{aligned}\hat{\boldsymbol{\theta}} &:= \arg\min_{\boldsymbol{\theta}} \sum_{n=1}^{N} (y^{(n)} - \boldsymbol{\theta}^\mathrm{T}\mathbf{x}^{(n)})^2 + C\,\|\boldsymbol{\theta}\|^2 &\quad (2.13)\\ &= (X^\mathrm{T}X + CI)^{-1}X^\mathrm{T}\mathbf{y} &\quad (2.14)\end{aligned}$$

Diese regularisierte Variante der linearen Regression ist in der Literatur als Ridge-Regression bzw. *Linear-Ridge-Regression* (LRR) bekannt, wobei die Wahl der ℓ_2-Norm als Regularisierungsfunktion als Tikhonov-Regularisierung bezeichnet wird. Durch das Einbringen einer zusätzlichen Nebenbedingung wird der Raum möglicher Lösungen für das Optimierungsproblem eingeschränkt. Konkret werden nur Lösungen zugelassen, deren euklidische Norm kleiner als ein bestimmter Schwellwert ist. Dabei sind die bedingte und unbedingte Form des Optimierungsproblem für eine konkrete Wahl von ϱ bzw. $C > 0$ äquivalent, wobei ein direkter Zusammenhang analytisch nicht angegeben werden kann. Es werden Lösungen bevorzugt, die eine kleine quadratische Norm aufweisen. Je größer der Regularisierungsparameter C gewählt wird, desto stärker ist dieser Effekt. Man spricht von einer geschrumpften Lösung gegenüber der ursprünglichen, da viele

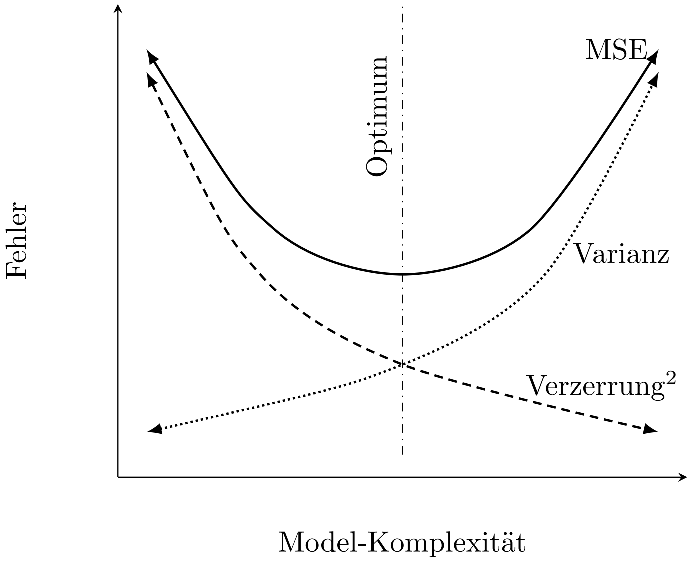

Abbildung 2.11: Veranschaulichung des Verzerrung-Varianz-Dilemmas.

Komponenten der geschätzten Lösung $\hat{\boldsymbol{\theta}}$ gegen den Wert Null gedrückt werden, was zu einer Verzerrung des Schätzers führt, die Komplexität des Modells verringert und damit auch die Varianz, da weniger Parameter für die Schätzung benötigt werden. Die Modellkomplexität sei dabei ein Maß für die Anzahl der zu schätzenden Parameter.

Wird die Lösung jedoch zu stark geschrumpft, überwiegt der Einfluss der Verzerrung auf den Gesamtfehler. Dieser Zusammenhang ist in Abbildung 2.11 dargestellt. Leider lässt sich analytisch kein optimaler Wert für den Regularisierungsparameter C ermitteln, da dieser von den zugrunde liegenden Daten abhängt, weshalb in der Praxis oft auf statistische Verfahren, wie beispielsweise die Kreuzvalidierung, zurückgegriffen wird, um diesen zu bestimmen. Dieser Prozess zur Findung des Schätzers mit dem geringstem Gesamtfehler ist als Modellselektion oder Modellauswahl bekannt und resultiert aus dem Verzerrung-Varianz-Dilemma (Bishop, 2006).

Abschließend sei noch der Schätzer des gesuchten funktionalen Zusammenhangs angegeben, der durch die Lineare Regression berechnet wird, wobei der Gleichanteil mit in den Parametervektor integriert wurde.

$$\hat{y}(\mathbf{x}) = \hat{\boldsymbol{\theta}}^{\mathrm{T}}\mathbf{x} \tag{2.15}$$

Mehrschichtiges Perzeptron (1980er)

Unterliegt den Daten der Lernstichprobe kein linearer Zusammenhang, sind nichtlineare Regressionsmethoden erforderlich, um eine gute Schätzung $\hat{y}$ zu bestimmen. Der naheliegendste Ansatz ist die Verwendung einer nichtlinearen Modellfunktion f, die in der

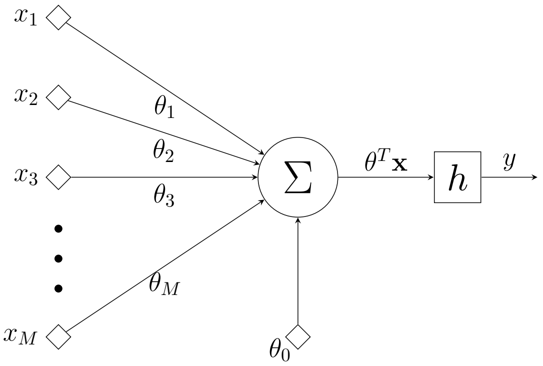

Abbildung 2.12: Modell eines künstlichen Neurons.

Lage ist, den gesuchten funktionalen Zusammenhang abzubilden.

Neuronale Netze stellen eine Klasse mathematischer Modelle dar, die versuchen, die Informationsverarbeitung in biologischen Systemen zu beschreiben. Die Idee basiert auf der Tatsache, dass natürliche, informationsverarbeitende Systeme, wie beispielsweise das menschliche Gehirn, aus einer Vielzahl von verbundenen Neuronen bestehen, die als Knoten eines Netzes aufgefasst werden können. Technische Systeme zur Informationsverarbeitung, die auf diesen Modellen basieren, werden als *Künstliche Neuronale Netze* (KNN) bezeichnet. KNN können u.a. auch eingesetzt werden, um Regressionsprobleme zu lösen, da sie als sehr flexible, parametrische Modelle interpretiert werden können. Aus Sicht der Regressionsanalyse macht es jedoch keinen Sinn zu versuchen, dabei möglichst wirklichkeitsgetreue Modelle zu verwenden, da diese nur zu unnötigen Limitierungen führen würden. Sinnvoller ist es, die flexiblen Modelle der KNN optimal im Sinne der statistischen Lerntheorie zu nutzen.

In Abbildung 2.12 ist das mathematische Modell eines Neurons dargestellt, welches einen Knoten in einem neuronalen Netz beschreibt. In Anlehnung an sein biologisches Vorbild besitzt ein Neuron mehrere Eingänge, welche unterschiedlich stark gewichtet werden. Der Ausgang wird dann mittels Aktivierungsfunktion h aus der Linearkombination der gewichteten Eingänge berechnet. Ein einzelnes Neuron stellt dabei eine simple Modellfunktion dar, die für eine Regressionsanalyse genutzt werden kann, welche als $f(\mathbf{x}) = h(\boldsymbol{\theta}^{\mathrm{T}}\mathbf{x} + \theta_0)$ gegeben ist, wobei $\boldsymbol{\theta} := (\theta_1, \ldots, \theta_L)^{\mathrm{T}} \in \mathbb{R}^L$ und $\mathbf{x} := (x_1, \ldots, x_L)^{\mathrm{T}} \in \mathbb{R}^L$ gelte. Bei Wahl einer linearen Funktion für h entspricht dieses Modell dem der bereits besprochenen linearen Regression.

Eine Zusammenschaltung mehrere dieser Neuronen wird als neuronales Netz bzw. KNN bezeichnet. Eine wichtige Klasse solcher Netze stellen die sogenannten Mehrschichtigen Perzeptronen (engl. *Multilayer Perceptron*, MLP) dar. Das Perzeptron war einer der ersten Klassifikationsalgorithmen basierend auf der Theorie neuronaler Netze. Das Perzeptron stellt einen linearen, binären Klassifikator dar, welcher durch eine Signum-Funktion für h definiert ist. In der heutigen Literatur wird als MLP jedoch nicht nur die Zusammenschaltung mehrerer Perzeptronen bezeichnet, sondern meist auch der allgemeinere Fall mit beliebiger Aktivierungsfunktion h. Die verschalteten Neuronen lassen sich dabei einzelnen Schichten zuordnen, wobei zwischen Eingabeschicht, Ausgabeschicht und

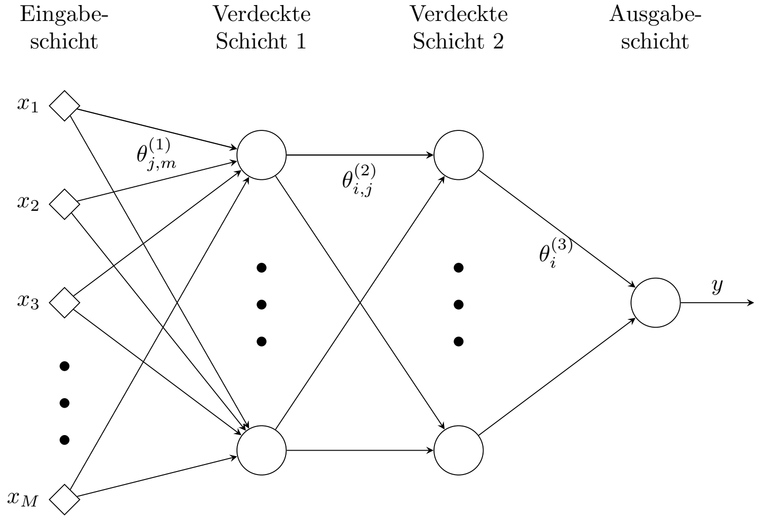

Abbildung 2.13: Zweischichtiges MLP mit einem Ausgang.

mehreren verdeckten Schichten unterschieden werden kann. Ein zweischichtiges MLP mit einem Ausgang, wie es für eine klassische, univariate Regressionsanalyse verwendet werden kann, ist in Abbildung 2.13 dargestellt.

Die Modellfunktion des zweischichtigen MLP mit einem Ausgang ist gegeben als

$$f(\mathbf{x}) = h\Big(\sum_{i=1}^{I} \theta_i^{(3)} h\Big(\sum_{j=1}^{J} \theta_{i,j}^{(2)} h(\sum_{m=1}^{L} \theta_{j,m}^{(1)} x_m + \theta_{j,0}^{(1)}) + \theta_{i,0}^{(2)} \Big) + \theta_0^{(3)} \Big). \tag{2.16}$$

Mithilfe des Universal-Approximation-Theorems kann gezeigt werden, dass selbst mittels eines einschichtigen MLP die Approximation einer großen Klasse kontinuierlicher Funktionen möglich ist. Dieses Theorem hat jedoch keine wirklich praktische Relevanz, belegt aber die Flexibilität des Modells Theodoridis, 2015. Für viele praktische Anwendungen hat sich gezeigt, dass sich die Einführung einer zweiten verdeckten Schicht als vorteilhaft erweist. Die Verwendung weiterer Schichten führt zu Problemen, die im Abschnitt über Deep-Learning besprochen werden.

Wie bei anderen parametrischen Regressionsverfahren können die Modellparameter im Sinne der ERM geschätzt werden, wodurch ein Optimierungsproblem definiert wird, wie in Gleichung 2.7 gegeben. Die $L + 1$ Parameter eines jeden Neurons bilden in ihrer Gesamtheit den Parameterraum und eine dementsprechende Familie von Funktionen, aus der jene mit den geringsten Kosten gewählt werden soll. Als Fehlerfunktion sei wiederum die quadratische Differenz angenommen. Durch die im Allgemeinen nicht konvexe Modellfunktion (2.16) weist die Kostenfunktion J viele lokale Extrempunkte auf, was die Optimierung erschwert. Eine analytische Lösung ist deshalb nicht möglich. Für die Lösung müssen also numerische Optimierungsverfahren herangezogen werden, wie in diesem Fall das Gradientenabstiegsverfahren. Zur Bestimmung des steilsten Abstiegs ist es nötig, den Gradienten der Kostenfunktion zu berechnen, weshalb es von Vorteil ist, eine stetig differenzierbare Funktion h als Aktivierungsfunktion zu wählen. Durch die komplexe Modellfunktion (2.16) ist es dennoch sehr aufwendig, den Gradienten durch einfaches Differenzieren zu bestimmen. Abhilfe schafft ein Algorithmus basierend auf der Idee der Fehlerrückverfolgung (engl. *Backpropagation Algorithm*, BPA), der eine modifizierte Form des Gradientenabstiegsverfahrens darstellt. Dabei wird der Gradient auf eine einfache Weise in einer rekursiven Form bestimmt. Ausgehend vom Netzausgang wird hierbei ein Fehlerterm Schicht für Schicht durch das Netz propagiert, der dann zur Bestimmung der jeweiligen Ableitung benutzt wird. Erst durch die Entwicklung des BPA im Jahr 1986 haben MLPs und andere KNN eine breite Anwendung erfahren. Durch Anwenden des BPA ist die Möglichkeit des sogenannten Online-Lernens gegeben. Dabei wird eine Optimierung und damit eine Anpassung der Gewichte für jedes Element aus $\mathcal{D}$ einzeln, nacheinander ausgeführt. Das Ergebnis der vorherigen Optimierung dient dabei jeweils als Initialisierung der folgenden.

Die komplexe Modellstruktur ermöglicht zwar einen flexiblen Einsatz, bringt aber auch einige Nachteile mit sich. Durch die nicht konvexe Kostenfunktion gibt es keinerlei Gewährleistung, dass durch die numerische Optimierung wirklich die optimalen Parameter gefunden werden, sprich der Algorithmus gegen das globale Extremum konvergiert. Etwas Abhilfe können jedoch sogenannte beschleunigte Gradientenverfahren schaffen, die in der Lage sind, lokale Extrema in einer gewissen Größenordnung zu überwinden. Ein weiterer Nachteil ist die hohe Anzahl der Freiheitsgrade bei der Wahl des konkreten Modells sowie des Optimierungsalgorithmus. Außerdem ist ein gewisses Maß an Erfahrung beim Einsatz KNN nötig, da viele implementierungsspezifische Fragen von der konkreten Problemstellung und Netzarchitektur abhängen, wie etwa Online- oder Stapel-Lernen, Initialisierung der Gewichte, Skalierung und Normierung der Daten, Regularisierung der Kostenfunktion und die Wahl des Trainingsalgorithmus, um nur einige zu nennen. (Bishop, 2006)

Kernel-Regression (1960er)

Eine andere Möglichkeit der nichtlinearen Regression ergibt sich durch eine Transformation der Merkmalvektoren. Bei diesem Ansatz können die Merkmalvektoren durch eine nichtlineare Abbildung in einen höherdimensionalen Vektorraum abgebildet werden, in dem dann eine lineare Regression durchgeführt werden kann, wie in diesem Abschnitt diskutiert wird. Diese Methoden sind als sogenannte Kernmethoden bekannt.

Üblicherweise wählt man Hilberträume mit reproduzierenden Kern (engl. *Reproducing Kernel Hilbert Space*, RKHS) für diesen Zweck aus, da diese eine Reihe nützlicher Eigenschaften mit sich bringen. Ein Hilbertraum ist ein RKHS, wenn er die Reproduktionseigenschaft

$$f(\mathbf{x}) = \langle f, \kappa(\cdot, \mathbf{x}) \rangle; \quad \forall f \in \mathbb{H}; \quad \forall \mathbf{x} \in \mathbb{R}^L \quad \kappa(\cdot, \mathbf{x}) \in \mathbb{H} \tag{2.17}$$

erfüllt. Dabei sei $\mathbb{H}$ der RKHS der reellen Funktionen über der Menge $\mathbb{R}^L$ und die Funktion κ der sogenannte Kern (engl. Kernel) des RKHS. Mithilfe der Kernfunktion ist es nun möglich, die Merkmalvektoren nach $\mathbb{H}$ abzubilden. Diese Abbildung $\phi : \mathbb{R}^L \rightarrow \mathbb{H}$ wird als Merkmaltransformation (engl. Feature Map) bezeichnet und wird folgendermaßen definiert.

$$\phi(\mathbf{x}) := \kappa(\cdot, \mathbf{x}) \tag{2.18}$$

Jedem Merkmalvektor wird so also eine Funktion zugeordnet, die Element des RKHS ist. Die Lösung des ursprünglichen linearen Regressionsproblems beinhaltet die Berechnung der inneren Produkte aller Merkmalvektoren beschrieben durch $X^{\mathrm{T}}X$. Die Berechnung aller inneren Produkte ist ebenso in $\mathbb{H}$ nötig, um das lineare Problem zu lösen, nur eben mittels der transformierten Merkmalvektoren $\phi(\mathbf{x})$. Durch Einsatz des „Kern-Tricks“ (engl. Kernel-Trick) können diese inneren Produkte sehr effizient und ohne explizite Berechnung der Merkmaltransformation angegeben werden. Diese Eigenschaft ist im Folgenden dargestellt und leicht durch Anwendung von Reproduktionseigenschaft (2.17) und Merkmaltransformation (2.18) nachzuvollziehen.

$$\langle \phi(\mathbf{x}^{(n)}), \phi(\mathbf{x}^{(m)}) \rangle = \kappa(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) \tag{2.19}$$

Eine Auswertung der Kernfunktion an den Merkmalvektoren reicht also aus, um das innere Produkt der Abbilder in $\mathbb{H}$ zu berechnen, eben auch wenn $\mathbb{H}$ unendlich viele Dimensionen aufweist. Dadurch wird eine sehr flexible Variante der nichtlinearen Regression ermöglicht.

Allerdings verbleibt jedoch die Frage, wie das lineare Regressionsproblem im RKHS gelöst werden kann. Um diese Frage zu beantworten, kann das sogenannte Representer-Theorem herangezogen werden. Dieses sagt aus, dass sich die Lösung des linearen Regressionsproblems im RKHS durch folgende Gleichung repräsentieren lässt.

$$f(\mathbf{x}) = \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\kappa}(\mathbf{x}) \tag{2.20}$$

Dabei sei $\boldsymbol{\theta} \in \mathbb{R}^N$ und $\boldsymbol{\kappa}(\mathbf{x}) = [\boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}^{(1)}), \dots, \boldsymbol{\kappa}(\mathbf{x}, \mathbf{x}^{(N)})]^{\mathrm{T}}$. Damit ist eine explizite Lösung des Regressionsproblems in $\mathbb{H}$ gegeben. Die geschätzte Modellfunktion stellt sich als gewichtete Summe der Kernfunktion, ausgewertet an den Merkmalvektoren, dar und ist damit direkt von der Lernstichprobe abhängig, was kennzeichnend für parameterfreie Regressionsverfahren ist. Die Komplexität der Modellfunktion, gemessen an der Anzahl der zu bestimmenden Gewichtsfaktoren $\boldsymbol{\theta}$, wächst im Unterschied zu parametrischen Verfahren mit der Größe der Lernstichprobe.

Zur Lösung der Regressionsaufgabe verbleibt nun noch die Bestimmung der optimalen Gewichtsfaktoren, welche durch die Kleinste-Quadrate-Methode geschätzt werden können. Das resultierende Problem mit geschlossener Lösung ist im Folgenden angegeben, wobei wiederum eine Regularisierung eingefügt wurde. Aus diesem Grund ist die beschriebene Methode auch als *Kernel-Ridge-Regression* (KRR) benannt. (Theodoridis, 2015)

$$\begin{align} \hat{\boldsymbol{\theta}} &:= \arg\min_{\boldsymbol{\theta}} \sum_{n=1}^{N} \left(y^{(n)} - \boldsymbol{\theta}^{\mathrm{T}} \boldsymbol{\kappa}(\mathbf{x}^{(n)}) \right) + C \boldsymbol{\theta}^{\mathrm{T}} \mathcal{K}^{\mathrm{T}} \boldsymbol{\theta} \tag{2.21} \\ &= (\mathcal{K} + CI)^{-1} \mathbf{y} \tag{2.22} \end{align}$$

Dabei sind alle inneren Produkte durch die Kernmatrix $\mathcal{K}$ gegeben.

$$\mathcal{K} = \begin{pmatrix} \kappa(\mathbf{x}_1, \mathbf{x}_1) & \dots & \kappa(\mathbf{x}_1, \mathbf{x}_N) \\ \vdots & \ddots & \vdots \\ \kappa(\mathbf{x}_N, \mathbf{x}_1) & \dots & \kappa(\mathbf{x}_N, \mathbf{x}_N) \end{pmatrix} \tag{2.23}$$

Die Wahl der Kernfunktion ist entscheidend bei der Kernel-Regression und anderen Kernmethoden. In vielen praktischen Problemen wird der Gauß-Kern (engl. Gaussian Kernel o. Radial Basis Function Kernel) verwendet, gegeben als $\kappa(\mathbf{x}, \mathbf{y}) = \exp(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2})$, da der RKHS, welcher durch den Gauß-Kern erzeugt wird, unendlichdimensional ist und damit sehr flexibel. Benutzt man einen linearen Kern $\kappa(\mathbf{x}, \mathbf{y}) = \mathbf{x}^{\mathrm{T}}\mathbf{y}$, so lässt sich leicht prüfen, dass die Ergebnisse der linearen Regression und Kernel-Regression übereinstimmen. In diesen Fall stellt die Kernel-Regression die duale Form des linearen Regressionsproblems dar, was auch explizit gezeigt werden kann, und hat deshalb natürlich auch die selbe Lösung. Aber erst durch die Darstellung des Problems in seiner dualen Form mittels Kernfunktionen gelingt es, durch Nutzung zweckmäßiger Kerne die Regression für den nichtlinearen Fall zu erweitern. Dieser Zusammenhang zwischen primaler, parametrischer Form und dualer, parameterfreien Form besteht übrigens für viele lineare Modelle (Bishop, 2006).

Der Schätzer der Kernel-Regression ist durch nachfolgende Gleichung gegeben.

$$\hat{y}(\mathbf{x}) = \hat{\boldsymbol{\theta}}^{\mathrm{T}} \boldsymbol{\kappa}(\mathbf{x}) \tag{2.24}$$

Support-Vektor-Regression (Vapnik, 1995)

Ein generelles Problem parameterfreier Regressionsverfahren ist, dass die Komplexität des Modells mit der Größe der Lernstichprobe wächst. Deshalb stellt sich die Frage, ob es überhaupt nötig ist, alle Daten der Lernstichprobe für eine Vorhersage zu verwenden oder ob sich nicht auch eine Untermenge relevanter Trainingsdaten bestimmen lässt, die dann für die Vorhersage genutzt werden. Diese Überlegung führt zu den sogenannten Kernmethoden mit dünnbesetzter Lösung (engl. Sparse Kernel Machines).

Der wesentliche Unterschied zur eben beschriebenen KRR ist die Wahl einer linearen ϵ-intensiven Fehlerfunktion anstatt einer quadratischen Differenz. Eine Lösung dieses Regressionsproblems kann nun wieder mittels Representer-Theorem hergeleitet werden oder aber auch eleganter über die angesprochene duale Form, was im Folgenden kurz beschrieben ist. Dazu wird anfangs eine lineare Modellfunktion angenommen und die Lösung dafür bestimmt. Anschließend wird gezeigt, wie die gefundene Lösung für einen beliebigen RKHS verallgemeinert werden kann. Die folgenden Gleichungen charakterisieren das lineare Regressionsproblem, welches zunächst gelöst werden kann.

$$f(\mathbf{x}) = \boldsymbol{\theta}^{\mathrm{T}} \mathbf{x} + \theta_0 \tag{2.25}$$

$$\mathcal{L}\big(y, f(\mathbf{x})\big) = \begin{cases} |y - f(\mathbf{x})| - \epsilon, & \text{if } |y - f(\mathbf{x})| > \epsilon \\ 0, & \text{if } |y - f(\mathbf{x})| \leq \epsilon \end{cases} \tag{2.26}$$

Die Kostenfunktion des resultierenden Optimierungsproblems ist durch die Verwendung der ϵ-intensiven Fehlerfunktion nicht stetig differenzierbar, was das Auffinden einer Lösung erschwert. Abhilfe schaffen sogenannte Schlupfvariablen, die eingeführt werden können, um das Problem umzuformulieren, ohne dessen Lösung zu verändern. Anschließend kann die Berechnung des dualen Problems erfolgen, dessen Lösung rechentechnisch günstiger ist aber auch vorteilhaft für die spätere Verallgemeinerung für den nichtlinearen Fall. Die Kostenfunktion des dualen Problems ist durch das Infimum der Lagrang'schen Funktion gegeben und hängt vom inneren Produkt der Merkmalvektoren ab. Das resultierende duale Problem stellt sich als quadratisches Programm dar, welches beispielsweise durch ein Innere-Punkte-Verfahren effizient gelöst werden kann (Boyd und Vandenberghe, 2004).
Nimmt man nun den allgemeineren Fall der linearen Regression in einem RKHS an, d.h. $f(\mathbf{x}) = \langle \theta, \phi(\mathbf{x}) \rangle + \theta_0, \theta \in \mathbb{H}$ sei als Modellfunktion gegeben, so kann man unter Ausnutzung des Kern-Tricks die inneren Produkte der dualen Kostenfunktion durch die charakteristische Kernfunktion des erzeugten RKHS ersetzen. So erhält man auf eine einfache und intuitive Weise die Lösung für das verallgemeinerte Problem. Nachfolgend ist das verallgemeinerte, duale Problem mit entsprechender Kostenfunktion und Nebenbedingungen angegeben.

$$\begin{aligned} \max_{\boldsymbol{\lambda},\tilde{\boldsymbol{\lambda}}} \sum_{n=1}^{N} (\tilde{\lambda}_n - \lambda_n) y^{(n)} - \epsilon(\tilde{\lambda}_n + \lambda_n) - \frac{1}{2} \sum_{n=1}^{N} \sum_{m=1}^{N} (\tilde{\lambda}_n - \lambda_n)(\tilde{\lambda}_m - \lambda_m) \kappa(\mathbf{x}^{(n)}, \mathbf{x}^{(m)}) \quad (2.27) \\ \text{s.t.} \quad 0 \leq \tilde{\lambda}_n \leq C, 0 \leq \lambda_n \leq C, \quad n = 1, \ldots, N \\ \sum_{n=1}^{N} \tilde{\lambda}_n = \sum_{n=1}^{N} \lambda_n \end{aligned}$$

$\tilde{\lambda}_n$ und λ_n sind dabei die Lagrang'schen Multiplikatoren, welche durch Lösen des quadratischen Programms bestimmt werden. Mithilfe der Lagrang'schen Multiplikatoren kann nun die Gewichtsfunktion θ geschätzt werden, welche durch folgenden Ausdruck gegeben ist

$$\hat{\theta}(\cdot) = \sum_{n=1}^{N} (\tilde{\lambda}_n - \lambda_n) \kappa(\cdot, \mathbf{x}^{(n)}) \quad (2.28)$$

Die Schätzung des Bias-Terms $\hat{\theta}_0$ ist als Mittelwert der Lösungen des Gleichungssatzes $y^{(n)} - \langle \theta, \phi(\mathbf{x}) \rangle - \theta_0 = \epsilon$ gegeben. Mit diesen Ergebnissen lässt sich unter Ausnutzung des Kern-Tricks und der Reproduktionseigenschaft der Schätzer für die nichtlineare, ϵ-intensive Regression angeben. (Theodoridis, 2015)

$$\hat{y}(\mathbf{x}) = \sum_{n=1}^{N_s} (\tilde{\lambda}_n - \lambda_n) \kappa(\mathbf{x}, \mathbf{x}^{(n)}) + \hat{\theta}_0 \quad (2.29)$$

Dabei ist $N_s \leq N$ die Anzahl Lagrang'scher Multiplikatoren ungleich Null, was bedeutet, dass nicht alle Trainingsdaten für die Schätzung der Regressionsfunktion notwendig

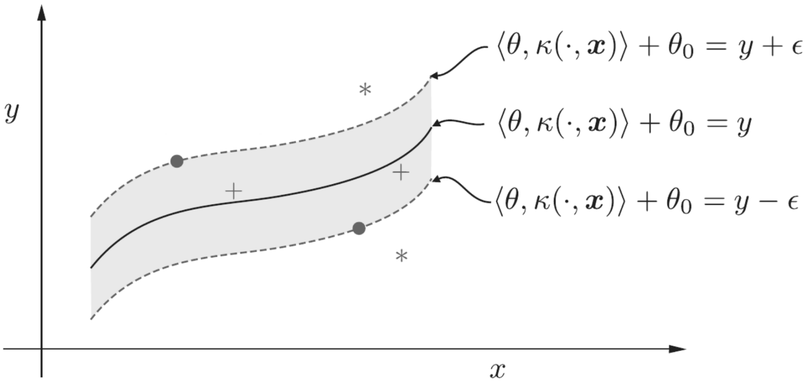

Abbildung 2.14: Veranschaulichung der ϵ-SVR aus Theodoridis (2015).

sind, so wie zu Anfang beabsichtigt. Es kann sogar gezeigt werden, dass die Multiplikatoren nur für solche Merkmalvektoren $\mathbf{x}^{(n)}$ ungleich Null sind, für die der gemessene Fehler größer oder gleich ϵ ist. Da nur diese Vektoren einen Beitrag zur Schätzung leisten, werden sie als sogenannte Stütz- oder Support-Vektoren bezeichnet, wodurch das beschriebene Verfahren auch als *Support-Vektor-Regression* (SVR) bekannt ist. Abbildung 2.14 veranschaulicht diesen Zusammenhang. Über den Parameter ϵ kann implizit eingestellt werden, wie viele Merkmalvektoren einen Beitrag zur Schätzung der gesuchten Funktion leisten, d.h., wie viele Stützvektoren verwendet werden. Der Parameter ϵ definiert dabei einen Bereich um die Schätzung $\hat{y}$ herum, welcher in Abbildung 2.14 grau hinterlegt ist. Nur Vektoren außerhalb dieses Bereichs leisten einen Beitrag zur Schätzung und sind somit als Stützvektoren definiert.

Es existiert auch eine abgewandelte, dennoch äquivalente, Form der ϵ-SVR, die als ν-SVR bekannt ist (Smola und Schölkopf, 2004). Bei dieser Form der SVR kann über den Parameter ν explizit angegeben werden, wie viele Stützvektoren verwendet werden sollen. Dieser Wert definiert dann wiederum ein konkretes ϵ, das den Bereich gültiger Stützvektoren determiniert. Da sich jedoch bei den meisten Problemen vorher keine Aussage über die Anzahl der benötigten Stützvektoren machen lässt, müssen die Parameter ϵ bzw. ν durch statistische Methoden ermittelt werden und es ist egal, welche Form der SVR verwendet wird.

Deep-Learning (seit 2000er)

Hinter dem Schlagwort „Deep-Learning“ verbergen sich eine Reihe von Modellen, die auf sogenannten tiefen Netzarchitekturen basieren und spezielle Varianten des Trainings nutzen und natürlich auch eingesetzt werden können, um Regressionsprobleme zu lösen. Eine Netzarchitektur wird als tief bezeichnet, wenn sie mehr als zwei verdeckte Schichten aufweist. Solche Netze werden als tiefe neuronale Netze (engl. *Deep Neural Networks*, DNN) bezeichnet. Trotz der Aussage des Universal-Approximation-Theorems erweisen sich DNNs in zahlreichen praktischen Anwendungen als äußerst nützlich. Besonders bei

Problemstellungen, in denen es darum geht, komplexe Muster und Strukturen in den Daten zu finden, wie beispielsweise Objekte in einem Video, erweisen sich DNNs als vielversprechender Ansatz.

Die Verarbeitung der Merkmale durch eine verdeckte Schicht entspricht einer Merkmaltransformation, die als eine Abstrahierung der Merkmale interpretiert werden kann. In jeder Schicht des Netzwerks erfolgt also eine unterschiedlich starke Abstrahierung der Merkmale, wodurch eine semantische Struktur entsteht, die es dann beispielsweise ermöglicht, Gesichter zu klassifizieren. Diese semantische Struktur kann als eine Hierarchie von Merkmalen verstanden werden, welche die Struktur der Daten analysiert. Während des Lernprozesses entscheidet damit das Netzwerk von selbst, welche Abstrahierung sinnvoll ist und welche Merkmale relevant sind. Damit wird eine manuelle Merkmaltransformation überflüssig, da diese vom Netz eigenständig gelernt wird. Voraussetzung dafür ist eine angemessen große Lernstichprobe.

Beim Training tiefer neuronaler Netze treten zahlreiche Schwierigkeiten auf. Zum einen wird die Modellfunktion durch die weiteren Schichten komplexer, was das Risiko erhöht, dass die Optimierung in einem „schlechten“ lokalen Minimum endet. Hinzu kommt, dass der Gradient durch den BPA in rekursiver Weise bestimmt wird, was dazu führen kann, dass der durch das Netz propagierte, multiplikative Fehler in den höheren Netzschichten sehr klein werden kann, wodurch auch der Gradient sehr klein wird. Dies führt dazu, dass beim Training des Netzes nur eine geringe Änderung der Gewichte erzielt wird, was sich besonders in den eingangsnahen Schichten bemerkbar macht. Dieser Sachverhalt wird als Problem des verschwindenden Gradienten (engl. *Vanishing Gradient Problem*, VGP) bezeichnet. Für die unterschiedlichen tiefen Netzarchitekturen gibt es unterschiedliche Trainingsmethoden, um dem VGP zu entgehen oder abzumildern, wie im Folgenden kurz für drei Varianten von DNNs erläutert wird. (Theodoridis, 2015)

Deep-Belief-Network

Eine Möglichkeit zur Vermeidung des VGP bei tiefen, vorwärtsgerichteten KNN ist ein sogenanntes Vortraining. Anstatt die Gewichte des Netzes zufällig zu initialisieren und dann ein Training mittels BPA durchzuführen, findet ein unüberwachtes Vortraining statt, welches die Gewichte für nachfolgende überwachte Trainingsphase initialisiert. Das überwachte Training erfolgt wie gehabt mittels BPA und dient der Feineinstellung der Gewichte. Das unüberwachte Vortraining wird durch Einsatz beschränkter Boltzmann Methoden (engl. *Restricted Boltzmann Machines*, RBM) erreicht. Eine RBM ist eine spezielle Form eines zufälligen Markov-Feldes und damit ein graphisches, statistisches Modell zur Darstellung von Wahrscheinlichkeitsverteilungen, dessen Struktur man als KNN interpretieren kann. RBMs sind, unter der Voraussetzung einer ausreichend großen Lernstichprobe, in der Lage, beliebige diskrete Verteilungen abzubilden. Die meisten statistischen Methoden versuchen durch Maximierung der Likelihood-Funktion die gesuchte Verteilung zu schätzen, was zu sehr komplexen Optimierungsproblemen führen kann. Um den Rechenaufwand zu minimieren, wird beim Training der RBM die Likelihood-Funktion durch zwei Kullback-Leibler-Divergenzen ersetzt. Der resultierende Algorithmus ist als Contrastive-Divergence Algorithmus bekannt und ermöglicht ein effizientes Training von RBMs und damit erst die praktische Anwendung tiefer neuronaler

Netze für viele Problemstellungen.

Beim Training des tiefen KNN selbst werden nun immer jeweils zwei Schichten als RBM aufgefasst. In einer rekursiven Form, beginnend mit den N Merkmalvektoren, werden nun die RBMs nacheinander vom Netzeingang aus trainiert. Der Eingang der nächsten RBM ist dabei immer durch N Samples der gelernten Verteilung der vorherigen RBM gegeben. Es sei hervorgehoben, dass die Messwerte in dieser Phase des unüberwachten Lernens nicht beteiligt sind. Dieser Prozess stellt das angesprochene hierarchische Lernen relevanter Merkmale dar, wodurch jede RBM eine höhere Stufe der Abstraktion darstellt. Nach Abschluss des Vortrainings werden die Gewichte der letzten Schicht des Netzwerks mithilfe der Messwerte durch ein überwachtes Training ermittelt. Diese Gewichte werden dann benutzt, um alle anderen Gewichte des Netzes zu initialisieren, wodurch eine vorteilhafte Ausgangslage für die folgende Optimierung erreicht wird. Durch Anwendung des BPA werden dann schließlich die finalen Werte der Gewichte gefunden.

Die eben beschriebene Methode des Lernens mittels Vortraining wurde zum ersten mal von Hinton et al. (2006) im Zusammenhang mit speziellen Bayes Netzwerken vorgestellt, die als *Deep-Belief-Network*s (DBN) bekannt geworden sind. DBNs sind also generative Modelle, die damit nicht nur auf Anwendungen der Regression und Klassifikation beschränkt sind, wie die anderen hier vorgestellten diskriminativen Modelle, sondern können zusätzlich zur Generierung neuer Daten benutzt werden.

Recurrent-Neural-Network

*Recurrent-Neural-Network*s (RNN) sind rückgekoppelte neuronale Netze, die besonders für Probleme geeignet sind, denen eine zeitliche Struktur unterliegt, wie beispielsweise der Spracherkennung. Die Rückkopplung erlaubt es dem Netz, Informationen für eine gewisse Zeit zu speichern und somit Abhängigkeiten zwischen zeitlich aufeinanderfolgenden Daten herzustellen. „Entfaltet“ man solch ein Netzwerk in Gedanken über die Zeit, ergibt sich eine tiefe Struktur, wodurch ein spezieller Trainingsalgorithmus angewendet werden muss. Dieser stellt eine modifizierte Variante des BPA dar und wird als Backpropagation-through-Time bezeichnet. Dieser Algorithmus leidet jedoch ebenfalls unter dem VGP. Um dieses Problem zu vermeiden, können sogenannte *Long-Short-Term-Memory* (LSTM) Blöcke verwendet werden, die einfache Neuronen ersetzen. Diese Blöcke besitzen einen speziellen Mechanismus, ausgestattet mit einem Speicher, der den durch das Training propagierten Fehler über einen längeren Zeitraum auf einem konstanten Wert hält, was das Verschwinden des Gradienten vermeidet. (Lipton et al., 2015)

Convolutional-Neural-Network

Diese Klasse von Netzen kann ebenfalls als tief bezeichnet werden, da sie zumeist viele Schichten aufweist. Trotzdessen können solche Netze mittels einfachen BPA trainiert werden. Dabei wird Gebrauch von einer Form der Gewichteverteilung gemacht, d.h. viele Parameter des Netzes haben den selben Wert. Dies reduziert die Komplexität und damit das VGP, wodurch der Einsatz tiefer Architekturen durch eine spezielle Netzstruktur ermöglicht wird. Anwendung finden diese Netze zumeist in der Bildverarbeitung und seien hier deshalb nur der Vollständigkeit halber erwähnt. (Bishop, 2006)

Kapitel 3

3 Lösungsmethode

3.1 Formale Beschreibung

Zunächst soll eine rein formale Beschreibung der erarbeiteten Lösungsmethode erfolgen. Diese dient dem Überblick über das System und ist gleichzeitig Grundlage der softwareseitigen Implementierung. Details zu den einzelnen Komponenten und deren Implementierung werden in den folgenden Abschnitten dieses Kapitels besprochen. Die vorgestellte Methode wurde für die Grundfrequenzvorhersage von Einzelwörter entwickelt, lässt sich jedoch durch Hinzunahme geeigneter Eingangsinformationen auch für Sätze erweitern, zweckmäßige Trainingsdaten vorausgesetzt. Abbildung 3.1 veranschaulicht das entwickelte System in einem Blockschaltbild.

Ziel der Grundfrequenzvorhersage in einem TTS System ist die Vorhersage bzw. Schätzung einer Grundfrequenzkontur, welche als zeitkontinuierliches Signal $\hat{f}_0(t) \in \mathcal{X}$ modelliert wird. Für die Vorhersage dienen Informationen, die aus der linguistischen Textanalyse und Graphem-zu-Phonem Konvertierung gewonnen werden. Im Speziellen liegt als Eingangsinformation die SAMPA-Transkription der Äußerung mit Silbengrenzen und Akzentsymbolen vor, welche durch einen Vektor $\mathbf{q} \in \mathcal{Q}^U$ gegeben ist. Dabei sei $\mathcal{Q}$ die Menge aller SAMPA-Symbole, U die Anzahl an SAMPA-Symbolen in der Äußerung und S die jeweilige Anzahl der Silben. Durch Angabe von Hauptakzenten und Nebenakzenten in der SAMPA-Transkription, lässt sich für jede Äußerung ein eindeutiges Akzentmuster bestimmen, welches jeder Silbe einen Akzentwert zuordnet und durch einen Vektor $\mathbf{c} \in \{1, 2, 3, 4\}^S$ charakterisiert wird. Informationen über die Phrasierung spielen bei Einzelwörtern keine Rolle. Es sei zusätzlich erwähnt, dass alle Äußerungen des Korpus in einem neutralen Kontext gesprochen wurden. Die Eingangsinformationen werden einer Merkmaltransformation $o : \mathcal{Q}^U \rightarrow \mathbb{R}^{L \times S}$ unterzogen, um die Dimension des Eingangsraums zu reduzieren und gleichzeitig artikulationsspezifische Merkmale zu erhalten. Jede Äußerung lässt sich damit durch eine Merkmalmatrix $M = (\mathbf{m}_{\cdot 1}, \ldots, \mathbf{m}_{\cdot S})$

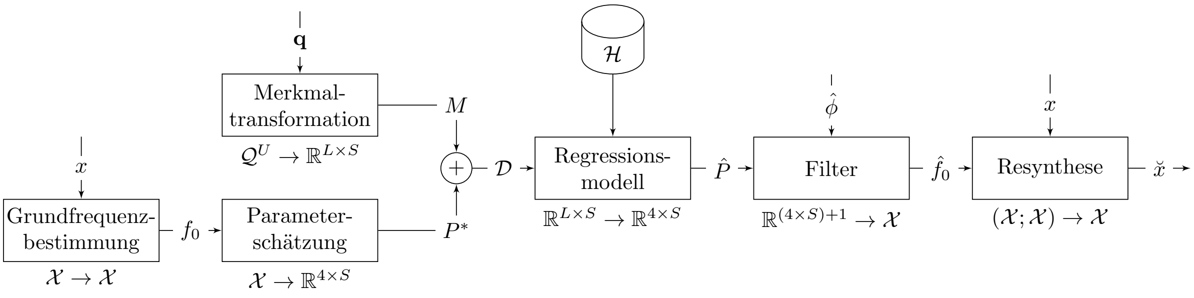

Abbildung 3.1: Blockschaltbild der entwickelten Lösungsmethode.

beschreiben, wobei die L-dimensionalen Spaltenvektoren $\mathbf{m} \in \mathbb{R}^L$ jeweils die Merkmale einer der S Silben charakterisieren. Als Grundlage der Merkmaltransformation dienen die distinktiven Merkmale der Phonologie.

Da die entwickelte Methode auf einem maschinellen Lernalgorithmus basiert, sind geeignete Ausgangsdaten des Systems notwendig, um ein Training durchzuführen. Dabei sollen silbenweise die Parameter des TAMs, beschrieben durch einen Parametervektor $\mathbf{p} = (m, b, \boldsymbol{\lambda}, d)^{\mathrm{T}} \in \mathbb{R}^4$, vorhergesagt werden. Die modellierte Grundfrequenz einer Äußerung, bestehend aus S Silben, ist damit durch eine Parametermatrix

$$P = (\mathbf{p}_{\cdot 1}, \mathbf{p}_{\cdot 2}, \ldots, \mathbf{p}_{\cdot S}) = \begin{pmatrix} m_1 & m_2 & \ldots & m_S \\ b_1 & b_2 & \ldots & b_S \\ \boldsymbol{\lambda}_1 & \boldsymbol{\lambda}_2 & \ldots & \boldsymbol{\lambda}_S \\ d_1 & d_2 & \ldots & d_S \end{pmatrix} \in \mathbb{R}^{4 \times S}$$

gegeben. Für die vollständige Charakterisierung der modellierten Grundfrequenz ist jedoch ein weiterer Parameter notwendig, welcher als Onset ϕ beschrieben wird. Dieser definiert den Wert der modellierten Grundfrequenz zu Äußerungsbeginn. Durch PTs und Onset ist ein modellierter Grundfrequenzverlauf also eindeutig definiert und durch $f_0(t; P, \phi) \in \mathcal{X}$ beschrieben. PTs und Onset werden im Weiteren zusammengefasst als TAM-Parameter bezeichnet und sind äußerungsspezifische bzw. quantitative Parameter. Im Gegensatz dazu stellt die Filterordnung N einen qualitativen Parameter des TAM dar und definiert dementsprechend ein Filter für die Target-Approximation.

Zur Ermittlung der Trainingsdaten werden die TAM-Parameter aus den aufgenommenen Sprachsignalen des Korpus bestimmt. In einem ersten Schritt wird dabei durch einen Grundfrequenzbestimmungsalgorithmus (engl. *Pitch Detection Algorithm*, PDA), basierend auf einer Autokorrelation, die natürliche Grundfrequenz $f_0(k\Delta t) \in \mathcal{X}$ einer jeden Äußerung auf Grundlage des abgetasteten Sprachsignals $x(k\Delta t) \in \mathcal{X}$ bestimmt. Dieser Algorithmus kann also als eine Abbildung zwischen Signalen $u : \mathcal{X} \to \mathcal{X}$ betrachtet werden. Anschließend werden durch Lösen der Optimierungsaufgabe $(P^*, \phi^*) :=$ $\arg\min \|f_0(k\Delta t) - f_0(k\Delta t; P, \phi)\|_2^2$ die optimalen Modellparameter bestimmt bzw. geschätzt, wobei nur die Zeitpunkte $k\Delta t$ stimmhafter Abschnitte $k \in \mathcal{V}$ der natürlichen f_0 betrachtet werden, was ein Abtasten der modellierten f_0 an diesen Stellen voraussetzt. Die Menge der Indizes innerhalb stimmhafter Signalanteile $\mathcal{V}$ wurde in Kapitel 1 definiert. Bei der Parameterschätzung können die Silbendauern d vernachlässigt werden, da diese bereits durch das annotierte Korpus gegeben sind und somit als optimal betrachtet werden können. Der um die Silbendauer reduzierte Parametervektor wird im Weiteren zur Unterscheidung in rekursiver Schreibweise mit $\boldsymbol{p} = (m, b, \boldsymbol{\lambda})^{\mathrm{T}} \in \mathbb{R}^3$ bezeichnet. Die so erhaltenen optimalen Modellparameter können dann für das Training des maschinellen Lernverfahrens benutzt werden. Dazu werden die Parameter mit den jeweiligen, transformierten Merkmalen silbenweise miteinander verknüpft und als Lernstichprobe $\mathcal{D} = \{(\mathbf{m}_{\cdot s}^{(a)}, \mathbf{p}_{\cdot s}^{*(a)})\}$, $s = 1...S^{(a)}$, $a = 1...A$ zusammengefasst, wobei A die Anzahl der im Korpus enthaltenen Äußerung ist.

Mithilfe der Lernstichprobe soll nun ein silbenbezogener, funktionaler Zusammenhang zwischen Merkmalvektoren $\mathbf{m}$ und Parametervektoren $\mathbf{p}$ ermittelt werden, der durch die Abbildung $g : \mathbb{R}^L \rightarrow \mathbb{R}^4$ beschrieben ist und zur Vorhersage neuer Parameter genutzt werden soll. Dieses Problem kann durch eine multivariate Regression gelöst werden. Da die vier Komponenten des Parametervektors $\mathbf{p}$ als unabhängig angenommen werden können, lässt sich das multivariate Regressionsproblem entkoppeln und in Form vier separater, univariater Probleme betrachten. Diese sind durch die Gleichung $\hat{\mathrm{p}}_i = g_i(\mathbf{m}) + e, i \in \{m, b, \boldsymbol{\lambda}, d\}$ gegeben, wobei das Ziel der Regression die Bestimmung der vier Regressionsfunktionen g_i ist. Das Training erfolgt durch die Minimierung einer Kostenfunktion, wie in Kapitel 2 beschrieben, welche hier als Vorhersagefehler bezeichnet werden soll. Durch eine Kreuzvalidierung werden die optimalen Hyperparameter der Regressionsmethode ermittelt. Der reellwertige Vektorraum aller möglichen Hyperparameter sei durch die Menge $\mathcal{H}$ beschrieben, dessen Dimension von der Regressionsmethode und dem Lernalgorithmus abhängig ist.

Das mithilfe der Lernstichprobe trainierte Modell kann nun zur Vorhersage der TAM-Parameter neuer Daten verwendet werden. Der ermittelte, silbenbasierte Zusammenhang wird für jede Silbe einer Äußerung separat angewandt, wodurch schließlich aus der Merkmalmatrix M eine Schätzung der Parametermatrix $\hat{P}$ berechnet werden kann. Durch eine Tiefpassfilterung können aus den durch die Parametermatrix beschriebenen PTs und einem geeigneten Onset-Wert eine Grundfrequenzkurve berechnet werden, was einer Abbildung $v : \mathbb{R}^{(4\times S)+1} \rightarrow \mathcal{X}$ entspricht, wobei die geschätzte Grundfrequenz nach dem TAM als $\hat{f}_0(t) = f_0(t; \hat{P}, \hat{\phi}) \in \mathcal{X}$ bezeichnet ist. Die Berechnung der Onset-Schätzung $\hat{\phi}$ erfolgt dabei vorher in einem separaten Schritt. Nach diesem Schritt ist die Grundfrequenzvorhersage abgeschlossen und kann in TTS Systemen für die Synthese von Sprache verwendet werden. Formal ist die Abbildung eines SAMPA-Vektors auf eine Grundfrequenzkontur als als $z : \mathcal{Q}^U \rightarrow \mathcal{X}$ gegeben und kann durch $\hat{f}_0(t; \hat{P}, \hat{\phi}) = v(g(o(\mathbf{q})))$ berechnet werden. Mithilfe von Ähnlichkeitsmaßen für Signale lässt sich die vorhergesagte Grundfrequenz mit der natürlichen vergleichen und somit die Leistungsfähigkeit des gesamten Systems beurteilen. Dabei dienen üblicherweise die Wurzel aus dem mittleren quadratischen Fehler (engl. *Root Mean Square Error*, RMSE) oder der Korrelationskoeffizient ρ als Ähnlichkeitsmaß bzw. Vergleichskriterium. Die für den Vergleich benutzten Daten werden dabei natürlich nicht für das Training benutzt, da man prüfen will, wie gut das System generalisiert.

Problematisch ist, dass RMSE und Korrelationskoeffizient mathematische Abstandsmaße darstellen und keinerlei Aussage über die wahrgenommene Natürlichkeit der vorhergesagten Grundfrequenz machen. Deshalb soll im Rahmen dieser Arbeit die vorhergesagte Grundfrequenz wieder auf die Originaläußerung der Korpus aufgeprägt werden und der Ergebnisbeurteilung innerhalb eines Perzeptionstests dienen. Diese Manipulation der Prosodie in einem Sprachsignal kann durch einen sogenannten PSOLA (engl. *Pitch Synchronous Overlap and Add*) Algorithmus erfolgen, der eine Signalabbildung $w : (\mathcal{X}, \mathcal{X}) \rightarrow \mathcal{X}$ durchführt und das manipulierte Sprachsignal $\breve{x}(k\Delta t)$ ausgibt.

Abbildung 3.2 stellt die soeben erläuterte Methode nochmals in anschaulicher Form dar und betont dabei die verschiedenen Ebenen zur Evaluierung des Systems, welche sich einmal auf die vorhergesagten Modellparameter, zum anderen auf die daraus berechnete Grundfrequenzkontur und schließlich auf das resynthetisierte Sprachsignal beziehen.

Ferner sind die verarbeiteten Elemente jedes Blocks explizit dargestellt. Dabei sei auch nochmal auf die verschiedenen Betrachtungsebenen bzgl. Silben und Äußerung hingewiesen. So wird eine Silbe jeweils mit einem Merkmal- bzw. Parametervektor assoziiert, eine Äußerung dagegen mit einer Merkmal- bzw. Parametermatrix.

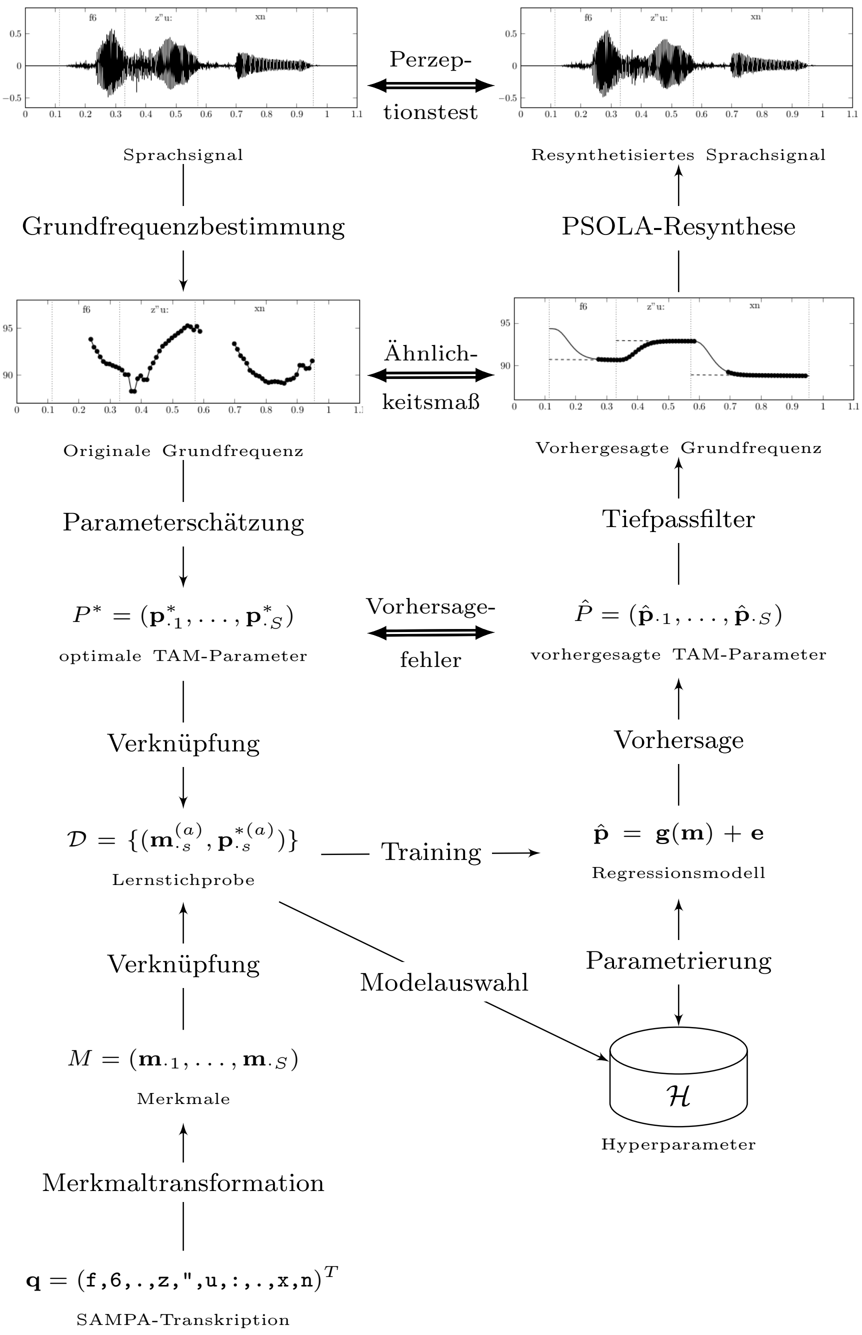

Abbildung 3.2: Informationsfluss der Lösungsmethode mit Fokus auf den verschiedenen Evaluationsebenen.

3.2 Bestandteile

Merkmaltransformation

Als Eingangsinformation der Grundfrequenzvorhersage dient die SAMPA-Transkription einer Äußerung, auf deren Basis eine Grundfrequenzkontur der Äußerung vorhergesagt werden soll. Die Transkription beinhaltet die transkribierte Phonemfolge der Äußerung, Silbengrenzen sowie die Silbenakzente. Da die Vorhersage der TAM-Parameter auf Basis einer Silbe erfolgt, ist es sinnvoll, eine feste Anzahl von Merkmalen zu identifizieren, die eine Silbe eindeutig beschreiben. Silbengrenzen sind in der SAMPA-Darstellung durch einen Punkt `[.]` gekennzeichnet. Unter Anbetracht des in Kapitel 1 vorgestellten Silbenaufbaus (Onset, Nukleus, Koda), sowie die Notwendigkeit spezieller SAMPA-Symbole zur Charakterisierung von Diphthongen, Stimmhaftigkeit, Nasalität, Lautlängen, Akzenten etc., ist die Anzahl der SAMPA-Symbole zur Beschreibung einer Silbe sehr variabel und kann theoretisch bis zu 15 Symbole oder auch mehr betragen.

Für Problemstellungen des maschinellen Lernens ist es üblich, Eingangsinformationen, die sich als diskrete Mengen darstellen, was für die Menge der SAMPA-Symbole zutrifft, eine sogenannte 1-aus-n Codierung zu verwenden, damit die einzelnen Eingangssymbole präzise unterschieden werden können. Die für das Training verwendeten Äußerungen sind mit einer Menge von 52 SAMPA-Symbolen zur Beschreibung einer Silbe transkribiert, was bedeutet, dass ein Symbol durch einen 52-dimensionalen Vektor, in dem genau eine Vektorkomponente 1 ist und alle anderen 0, repräsentiert ist. Geht man also nun davon aus, dass Silben mit bis zu 15 Symbolen berücksichtigt werden müssen, ergibt dies eine Eingangsdimension von 780, wobei noch keine Positionsinformationen berücksichtigt wurden. Eine Alternative wäre die Verwendung der 7-Bit ASCII Repräsentation der SAMPA-Symbole, die es erlaubt, jedes Symbol durch einen 7-dimensionalen Vektor darzustellen, was zu einer Eingangsdimension von 105 führen würde. Problem bei dieser Darstellung ist jedoch, dass beispielsweise die Phoneme [a] und [b] durch die Vektoren `(1100001)` und `(1100010)` ASCII-codiert sind, also im Eingangsraum sehr nahe beieinander liegen, obwohl Ersteres ein Vokal und Letzteres ein Verschlusskonsonant ist und damit völlig verschiedene prosodische Parameter realisieren. Ziel eines Lernalgorithmus ist es, Verallgemeinerungen zu finden, was um so besser funktioniert, wenn ähnliche Eingänge zu ähnlichen Ausgängen führen. Dies erlaubt es dem Lernverfahren auch, dass etwaige Korrelationen der Eingangsdaten gefunden werden können, die ein Generalisieren unterstützen.

Ziel der Merkmaltransformation ist eine Reduktion der Dimension des Eingangsraums sowie die Identifizierung artikulationsspezifischer Merkmale, die relevant für die Grundfrequenzvorhersage sein können und ähnliche Phoneme durch ähnliche Vektoren codieren, um die Vorhersage zu verbessern. Um eine passende Darstellung der SAMPA-Transkription für den Eingang des Regressionsproblems zu finden, können die distinktiven Merkmale herangezogen werden, welche artikulatorisch ähnlichen Phonemen nahe beieinander liegende Vektoren zuordnen. Ein weiterer Vorteil ist, dass durch Diakritika gegebene Informationen dabei mit codiert werden können, wobei sich eine Silbe also aus maximal neun Phonemen von Onset, Nukleus und Koda zusammensetzt. Da im Rahmen dieser Arbeit nur Wörter in ihrer Grundform betrachtet werden, ist die Anzahl der Phoneme in der Koda auf vier beschränkt, wodurch eine Silbe also maximal durch acht Phoneme beschrieben ist. Bei der Annahme von beispielsweise 12 distinktiven Merkmalen resultieren daraus 96 Merkmale zur Darstellung der phonetischen Information. Als Grundlage der in dieser Arbeit ausgearbeiteten Merkmaltransformation dienen die iden-

	IPA	SAMPA	Stimm	Art der Artikulation					Ort
				Nasal	Plosiv	Frikativ	Gleitlaut	Lateral	
			S	N	P	F	G	L	O
Konsonanten	p	`p`	0	0	1	0	0	0	0
	b	`b`	1	0	1	0	0	0	0
	t	`t`	0	0	1	0	0	0	1
	d	`d`	1	0	1	0	0	0	1
	k	`k`	0	0	1	0	0	0	2
	ɡ	`g`	1	0	1	0	0	0	2
	ʔ	`?`	1	0	1	0	0	0	3
	f	`f`	0	0	0	1	0	0	0
	v	`v`	1	0	0	1	0	0	0
	s	`s`	0	0	0	1	0	0	1
	z	`z`	1	0	0	1	0	0	1
	ʃ	`S`	0	0	0	1	0	0	1
	ʒ	`Z`	1	0	0	1	0	0	1
	θ	`T`	0	0	0	1	0	0	1
	ð	`D`	1	0	0	1	0	0	1
	ç	`C`	0	0	0	1	0	0	2
	ʝ	`j\`	1	0	0	1	0	0	2
	x	`x`	0	0	0	1	0	0	2
	h	`h`	0	0	0	1	0	0	3
	m	`m`	1	1	0	0	0	0	0
	n	`n`	1	1	0	0	0	0	1
	ŋ	`N`	1	1	0	0	0	0	2
	l	`l`	1	0	0	0	0	1	1
	ʁ	`R`	1	0	0	1	0	0	2
Affrikaten	pf	`pf`	0	0	1	1	0	0	0
	ps	`ps`	0	0	1	1	0	0	1
	ts	`ts`	0	0	1	1	0	0	1
	tʃ	`tS`	0	0	1	1	0	0	1
	pʃ	`pS`	0	0	1	1	0	0	1
	dʒ	`dZ`	1	0	1	1	0	0	1
	·	·	0	0	0	0	0	0	0

Tabelle 3.1: Konsonantenmerkmale

tifizierten Vokal- und Konsonantenmerkmale aus der unveröffentlichten Arbeit von Birkholz, Reichel et al. (2014), welche für die Lautdauervorhersage erarbeitetet wurden und sich an die Kategorien der distinktiven Merkmale anlehnen. Die im Rahmen der vorliegenden Arbeit veränderten Konsonantenmerkmale lassen sich nach Stimmhaftigkeit, Art sowie Ort der Artikulation einteilen und sind in Tabelle 3.1 für alle verwendeten Konsonanten dargestellt. In der ursprünglichen Form haben Birkholz, Reichel et al. (2014) den Ort der Artikulation durch vier binäre Merkmale beschrieben {Labial, Koronal, Dorsal, Laryngal}, die hier jedoch zu einem reellwertigen Merkmal zusammengefasst wurden, da der Ort der Artikulation eine natürliche Ordnung aufweist, wie in Abbildung 1.1 zu erkennen ist. Das Intervall [0,3] beschreibt somit den Ort der Artikulation beginnend bei den Lippen (Labial) entlang des Vokaltrakts bis hin zur Glottis (Laryngal) als ein singuläres Merkmal. Dies verringert weiterhin die Dimension des Eingangs und ermöglicht das Lernen möglicher Korrelationen. Die Art der Artikulation ist durch die fünf binären Merkmale {Nasal, Plosiv, Frikativ, Gleitlaut, Lateral} beschrieben, denen keine natürliche Ordnung unterliegt. Außerdem bildet die Stimmhaftigkeit des Konsonanten ein separates Merkmal. Die Vokalmerkmale erklären zum einen in Anlehnung an das Vokaltrapez aus Abbildung 1.1 die Zungenstellung mit zwei reellen Merkmalen {Position, Höhe} sowie die Lippenrundung durch ein binäres Merkmal und wurden dabei aus Birkholz, Reichel et al. (2014) übernommen. Außerdem ist ein reellwertiges Merkmal für die Vokallänge vorhanden sowie ein binäres Merkmal zur Beschreibung nasaler Vokale, welches vor allem bei den im Korpus enthaltenen französischen Fremdwörtern auftritt. Die Zusammenstellung der Merkmale für alle verwendeten Vokale ist in Tabelle 3.2 zu

IPA	SAMPA	Position	Höhe	Rundung	Länge	Nasal
		ZP	ZH	LR	LL	NL
i	`i`	-2	3	0	0	0
ɪ	`I`	-1	2	0	0	0
e	`e`	-2	1	0	0	0
ɛ	`E`	-2	-1	0	0	0
y	`y`	-2	3	1	0	0
ʏ	`Y`	-1	2	1	0	0
ø	`2`	-2	1	1	0	0
œ	`9`	-2	-1	1	0	0
ə	`@`	0	0	0	0	0
ɐ	`6`	0	-2	0	0	0
a	`a`	0	-3	0	0	0
ɑ	`A`	2	-3	0	0	0
u	`u`	2	3	1	0	0
ʊ	`U`	1	2	1	0	0
o	`o`	2	1	1	0	0
ɔ	`O`	2	-1	1	0	0

Tabelle 3.2: Vokalmerkmale

IPA	SAMPA	Bezeichnung	Merkmal
ː	:	langer Vokal	Lautlänge (LL) inkrementieren
˜	~	nasalierter Vokal	Nasal (NL) negieren
⌢	_^	nichtsilbischer Vokal	Arithmetisches Mittel; LL++
ˈ	=	silbischer Konsonant	Vokalmerkmale von ə
˳	_0	entstimmlichter Konsonant	Stimmhaftigkeit (S) negieren

Tabelle 3.3: Vokal- und Konsonantenmodifizierer

sehen.

Außerdem existieren in der SAMPA-Transkription sogenannte Modifizierer, die spezielle Merkmale eines Phonems ändern und u.a. die Diakritika repräsentieren. Diese Vokal- und Konsonantenmodifizierer sind in Tabelle 3.3 zusammengefasst. Hervorzuheben sei der Vokalmodifizierer für nichtsilbische Vokale, welcher beispielsweise zur Beschreibung von Diphthongen eingesetzt wird, wobei es immer einen silbischen und einen nichtsilbischen Vokal innerhalb eines Diphthongs gibt. Bei der Implementierung der Lösungsmethode wird der arithmetische Mittelwert der Vokalmerkmale der beteiligten Vokale gebildet, um Diphthonge zu charakterisieren. Dies ist damit zu begründen, da sich Diphthonge als ein Verlauf von einem Vokal zum anderen Vokal innerhalb des Vokaltrapez aus Abbildung 1.1 darstellen und somit die jeweils mittleren Positionen der Artikulatoren eine sinnvolle Beschreibung bilden. Es treten sogar Wörter auf, bei denen drei Vokale im Nukleus enthalten sind, wie z.B. *Ingenieur* [ɪnʒeni̯ˈøːᵄ], und die Mittelwertbildung ebenfalls als sinnvoll betrachtet werden kann. Zusätzlich wird bei Diphthongen auch die Lautlänge inkrementiert. Bei den Konsonantenmodifizierern sei jene Beschreibung silbischer Konsonanten hervorzuheben, welche bei speziellen, stimmhaften Konsonanten (Sonoranten) bei fehlenden vokalischen Silbenkern auftreten kann. Die Vokalmerkmale werden dabei alle auf Null gesetzt.

Aus den gezeigten Tabellen wird deutlich, dass sich ein Konsonant durch sieben und ein Vokal durch fünf Merkmale charakterisieren lässt. Unter Berücksichtigung des beschriebenen Silbenaufbaus ergeben sich damit 21 Merkmale zur phonetischen Beschreibung des Onset, 5 für den Nukleus und 28 für die Koda. Innerhalb der Merkmaltransformation werden diese 54 phonetischen Merkmale einer Silbe durch einen Vektor angegeben, wobei jedes Merkmal eine feste Position aufweist. Nicht vorhandene Konsonanten werden durch eine 0 in allen Merkmalen notiert.

Über die Phoneme und den Silbengrenzen hinaus enthält die Transkription auch Informationen über die Silbenakzente, welche durch Haupt- und Nebenakzente gekennzeichnet werden. Ein Hauptakzent steht dabei für eine stark betonte Silbe und wird durch den Wert 4 gekennzeichnet, Nebenakzente kennzeichnen nebenbetonte Silben und erhalten den Wert 3. Weiterhin werden unbetonte Silben mit dem Wert 2 und reduzierte mit dem Wert 1 versehen. Aus Kennzeichnung und Position der Akzente lässt sich so für jede Äußerung ein eindeutiges Akzentmuster ableiten, das einer jeden Silbe eine Akzentstufe zuordnet und die Betonung einer Äußerung beschreibt. Wie von Birkholz, Reichel et al. (2014) vorgeschlagen, wurden die Akzentstufen der vorherigen, aktuellen und folgenden Silbe als weitere silbenspezifische, reellwertige Merkmale aufgenommen. Zusätzlich sind Positionsinformationen der Äußerung wichtig für die Prosodiegenerierung. Dabei wurden folgende relevante Merkmale identifiziert: Anzahl der Wörter in der Äußerung, Positionsnummer des aktuellen Wortes in der Äußerung, Anzahl der Silben im aktuellen Wort, Positionsnummer der aktuellen Silbe innerhalb des Wortes, Anzahl der Phoneme im Onset, Anzahl der Phoneme in der Koda, Anzahl der Phoneme in der vorherigen Silbe sowie die Anzahl der Phoneme der folgenden Silbe. Die Positionsmerkmale wurden dabei ebenfalls von Birkholz, Reichel et al. (2014) übernommen. Es sei noch erwähnt, dass die vorliegende Arbeit die Grundfrequenzvorhersage für Einzelwörter untersucht und die Merkmale wortabhängigen Positionsmerkmale für eine geeignete Charakterisierung nicht notwendig wären. Dennoch eignen sich diese Merkmale zur Beschreibung von Komposita, die eine Zusammensetzung mehrerer Grundwörter darstellen, aber dennoch als Einzelwort zu werten sind. Da die Grundwörter dabei nicht durch ein Leerzeichen getrennt sind, wurde ein neues Symbol `[ | ]` zum verwendeten SAMPA-Zeichensatz hinzugefügt, welches Wortgrenzen in Komposita markiert.

Fasst man die soeben beschriebenen Merkmale zusammen, lassen sich diese also in 54 phonetische, drei Akzent- und acht Positionsmerkmale unterteilen, wodurch also eine Silbe durch 65 Merkmale beschrieben ist. Für jede Silbe einer Äußerung wird ein 65-elementiger Vektor bestimmt, der dann als Eingangsgröße eines maschinellen Lernalgorithmus fungiert. Das Ergebnis der Merkmaltransformation der SAMPA-Transkription einer Äußerung ist in Abbildung 3.3 veranschaulicht, wobei nochmals alle verwendeten Merkmale in tabellarischer Form beschrieben sind. Jedes Merkmal ist durch zwei Buchstaben abgekürzt, wobei die verschiedenen Konsonanten in Onset und Koda einfach durchnummeriert wurden. Die Merkmale einer mehrsilbigen Äußerung lassen sich der Darstellung nach als eine Merkmalmatrix auffassen.

	1	2		S	
1	S1	S1	…	S1	Stimmhaftigkeit 1. Konsonant, Binäres Merkmal (Onset)
2	N1	N1	…	N1	Nasalität 1. Konsonant, Binäres Merkmal (Onset)
3	P1	P1	…	P1	Plosivität 1. Konsonant, Binäres Merkmal (Onset)
4	F1	F1	…	F1	Frikativität 1. Konsonant, Binäres Merkmal (Onset)
5	G1	G1	…	G1	Gleitlaut 1. Konsonant, Binäres Merkmal (Onset)
6	L1	L1	…	L1	Lateralität 1. Konsonant, Binäres Merkmal (Onset)
7	O1	O1	…	O1	Ort der Artikulation 1. Konsonant, Reeles Merkmal (Onset)
8	S2	S2	…	S2	Stimmhaftigkeit 2. Konsonant, Binäres Merkmal (Onset)
	⋮	⋮		⋮	
21	O3	O3	…	O3	Ort der Artikulation 3. Konsonant, Reeles Merkmal (Onset)
22	ZP	ZP	…	ZP	Zungenposition Vokal, Reeles Merkmal (Nukleus)
23	ZH	ZH	…	ZH	Zungenhhe Vokal, Reeles Merkmal (Nukleus)
24	LR	LR	…	LR	Lippenrundung Vokal, Binäres Merkmal (Nukleus)
25	LL	LL	…	LL	Lautlänge Vokal, Reeles Merkmal (Nukleus)
26	NL	NL	…	NL	Nasalität Vokal, Binäres Merkmal (Nukleus)
27	S4	S4	…	S4	Stimmhaftigkeit 4. Konsonant, Binäres Merkmal (Koda)
	⋮	⋮		⋮	
54	O7	O7	…	O7	Ort der Artikulation 7. Konsonant, Reeles Merkmal (Koda)
55	AA	AA	…	AA	Akzentstufe der vorherigen Silbe, Reeles Merkmal
56	AV	AV	…	AV	Akzentstufe der aktuellen Silbe, Reeles Merkmal
57	AF	AF	…	AF	Akzentstufe der folgenden Silbe, Reeles Merkmal
58	WA	WA	…	WA	Wortanzahl in Äußerung, Reeles Merkmal
59	WN	WN	…	WN	Wortnummer innerhalb der Äußerung, Reeles Merkmal
60	AS	AS	…	AS	Silbenanzahl in Wort, Reeles Merkmal
61	NS	NS	…	NS	Silbennummer innerhalb des Wortes, Reeles Merkmal
62	PO	PO	…	PO	Phonemanzahl in Onset der Silbe, Reeles Merkmal
63	PC	PC	…	PC	Phonemanzahl in Koda der Silbe, Reeles Merkmal
64	PV	PV	…	PV	Phonemanzahl der vorheriger Silbe, Reeles Merkmal
65	PF	PF	…	PF	Phonemanzahl der folgenden Silbe, Reeles Merkmal

Abbildung 3.3: Merkmalmatrix einer Äußerung bestehend aus einem Merkmalvektor pro Silbe mit zugehörigen Abkürzungen und Bedeutungen der Merkmale.

Grundfrequenzbestimmung

Ein erster Schritt zur Bestimmung der optimalen PTs, die als Trainingsmaterial für den Lernalgorithmus dienen sollen, ist die Bestimmung der Grundfrequenz der im Korpus enthaltenen Sprachsignale. Ein solches Sprachsignal ist in Abbildung 3.4 dargestellt und ist in der Regel nicht stationär, was bedeutet, dass das Spektrum des Zeitsignals eine Zeitabhängigkeit aufweist.

Zur Bestimmung der Grundfrequenz wurde der von Boersma (1993) vorgestellte Algorithmus zur Grundfrequenzbestimmung auf Basis einer Kurzzeitanalyse verwendet. Dazu wird das eingehende Sprachsignal in sogenannte Frames unterteilt, die jeweils eine Länge von $\frac{0{,}75}{f_{0,\mathrm{min}}}$ haben und auf dieser Framebasis verarbeitet werden. Die genaue Länge der Frames ist ein einstellbarer Parameter des Algorithmus und hängt also von der niedrigsten zu erwartenden Grundfrequenz ab, welche standardmäßig auf 75 Hz gewählt wird, wodurch sich eine Framelänge von 10 msec ergibt. Innerhalb eines Frames wird das Signal als quasistationär angenommen, was bedeutet, dass sich das Spektrum des Signals in diesem Zeitabschnitt nur unwesentlich ändert. Für jeden Frame wird die *Autokorrelationsfunktion* (AKF) berechnet, woraus dann die Grundfrequenz für den Frame ermittelt werden kann. Die AKF wird dabei durch den hohen Rechenaufwand nicht direkt berechnet, sondern indirekt unter Ausnutzung des Wiener-Chintschin-Theorems, wodurch eine diskrete Fourier-Hin- und Rücktransformation mittels FFT- bzw. IFFT-Algorithmus nötig ist. Die Fourier-Transformation eines zeitlich begrenzten Frames führt zu einem sogenannten Verschmieren des Spektrums, da die Unterteilung in Frames als eine entsprechende Faltung des Signals mit einer Rechteckfunktion angesehen werden kann. Nach dem Faltungssatz der Fourier-Transformation erfolgt im Bildbereich eine Multiplikation der Spektren von Rechteckfunktion und Signal, wodurch das Signalspektrum mit der einer Spaltfunktion multipliziert wird und störende Signalanteile verursacht. Dieser Effekt kann durch Einsatz geeigneter Fensterfunktionen minimiert werden. Nach der Fensterung wird im Bildbereich der Betrag des Spektrums gebildet und anschließend rücktransformiert, was nach dem Wiener-Chintschin-Theorem die AKF des Frames lie-

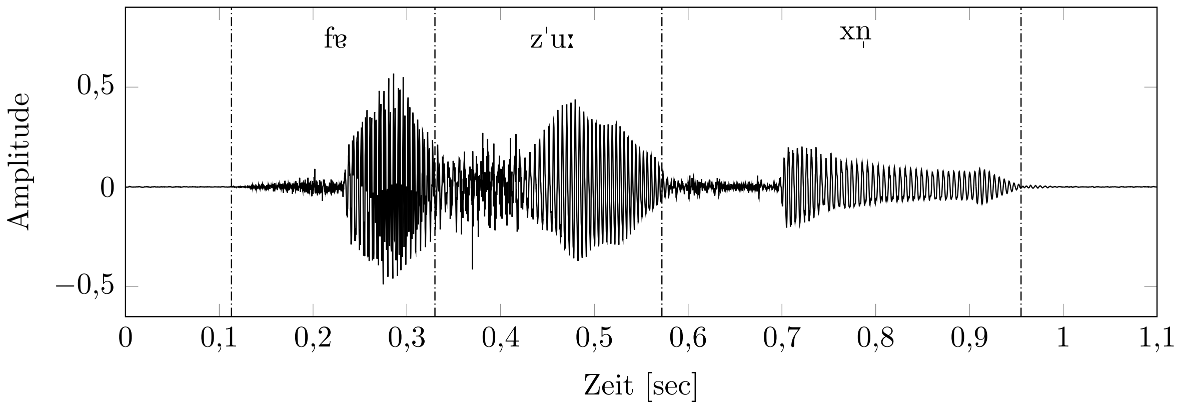

Abbildung 3.4: Beispielhaftes, digitales Sprachsignal der Äußerung *versuchen* [fɐzˈuːxn̩] als Ausgangspunkt der Grundfrequenzbestimmung.

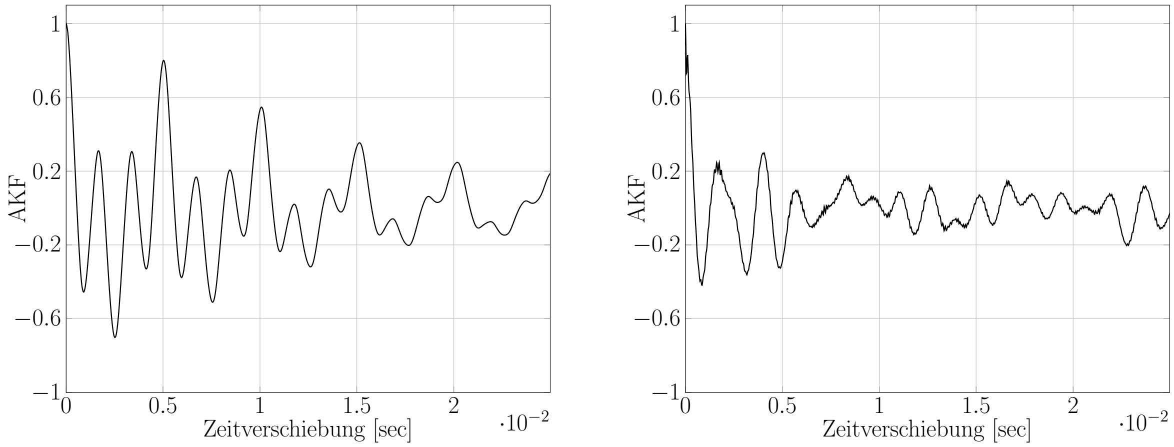

Abbildung 3.5: Autokorrelationsfunktion eines Frames mit stimmhaften (links) und stimmlosen Signalanteil (rechts).

fert. Die AKF stellt sich dann als eine symmetrische Folge von Extrema dar, welche Rückschlüsse auf die Periodizität des Signals zulässt. Der Abstand eines Maximums von der Ordinate entspricht dabei der Periodendauer. In Abbildung 3.5 sind beispielhaft die AKFs der 50 msec langen Frames [0,25; 0,3] sec und [0,6; 0,65] sec des Signals aus Abbildung 3.4 dargestellt. Ersterer ist Teil eines stimmhaften Sprachabschnittes, welcher durch deutliche, periodische Maxima mit abfallender Amplitude gekennzeichnet ist. Die Grundfrequenz dieses Frames lässt sich aus dem Diagramm als $f_0 = \frac{1}{5 \text{ msec}} = 200$ Hz ablesen, da diese die Inverse der größten erkennbaren Periodendauer darstellt. Der zweite Frame ist Teil eines stimmlosen Sprachabschnitts, der keine Grundfrequenz aufweist, was auch in der AKF erkennbar ist, die keine Struktur periodischer Maxima erkennen lässt.

Im stimmhaften Beispiel der obigen Grafik lässt sich die größte Periodendauer zwar gut erkennen, dennoch ist dies im Allgemeinen schwieriger. Das Problem hierbei ist, dass die AKF viele weitere Maxima aufgrund der Harmonischen im Signal aufweist. Weisen also die verschiedenen Maxima der AKF, hervorgerufen durch Harmonische und Grundfrequenz, ähnliche Amplituden auf, sind diese nicht klar unterscheidbar. Aus diesem Grund werden pro Frame also mehrere mögliche Kandidaten für die Grundfrequenz ausgewählt. Nachdem so alle Frames analysiert wurden, kann mithilfe einer dynamischen Suchen, implementiert durch den Viterbi-Algorithmus, der günstigste Pfad bzw. die günstigste Folge von Kandidaten berechnet werden. Als Kosten werden dabei u.a. die Amplituden der Kandidaten, mögliche Oktavfehler und die Stimmhaftigkeit berücksichtigt. Oktavfehler beschreiben ebensolche, an denen falsche Maxima ausgewählt werden und treten häufig an Übergängen zwischen stimmhaften und stimmlosen Signalabschnitten auf. Als Ergebnis der dynamischen Suche liegt ein Grundfrequenzwert pro Frame mit stimmhaften Signalanteil vor, welche als Folge betrachtet eine Schätzung des Grundfrequenzverlaufs darstellen. Abbildung 3.6 zeigt den Grundfrequenzverlauf des obigen Sprachsignals und wurde mit der eben beschriebenen Methode ermittelt. Die stimmhaften Abschnitte eines solchen Signals werden, wie in der Einführung definiert, durch die Menge $\mathcal{V}$ be-

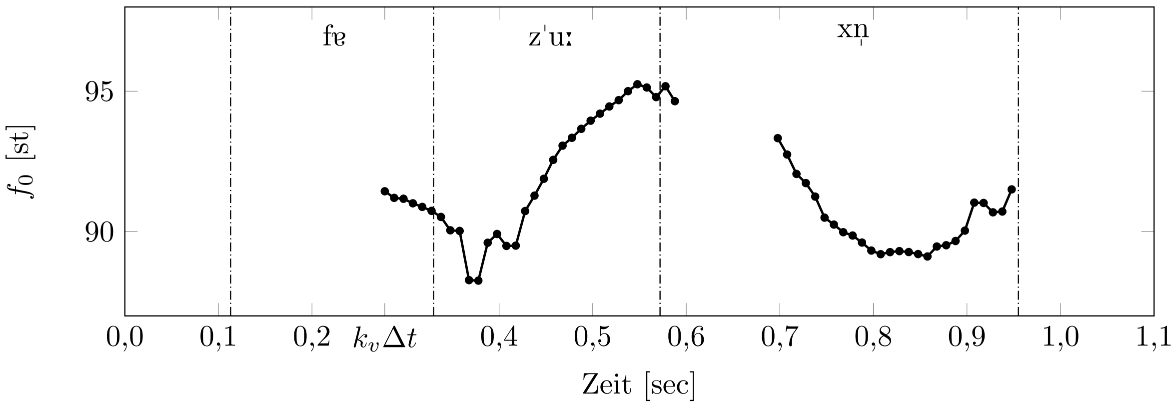

Abbildung 3.6: Ermittelter Grundfrequenzverlauf des Signals aus Abbildung 3.4 durch Einsatz des PDA nach Boersma (1993).

schrieben.

Der Algorithmus benötigt als Eingangsgröße einen erwarteten f_0-Bereich. Boersma und Weenink (2017) geben als Empfehlung für Frauenstimmen ein Bereich von [100,500] Hz an, der bei der Implementierung für die Verarbeitung des Korpus verwendet wurde, da der gesamte Korpus durch eine Frau eingesprochen wurde. Somit benutzt der Algorithmus eine Framelänge von 7,5 msec.

Grundfrequenzmodell

Wie bereits erwähnt, ist das Grundfrequenzmodell von Xu und Wang (2001) Basis der hier entwickelten Lösungsmethode. Eine erste, quantitative Umsetzung des Modells wurde von Prom-On et al. (2009) vorgestellt, jedoch wurden die quantitativen Modellparameter für diese Arbeit abweichend definiert und sind deshalb in Abbildung 3.7 nochmals veranschaulicht. Die Verschiebung b wird dabei zu Beginn einer Silbe definiert und nicht am Silbenende. Zusätzlich wurde der Onset explizit mit als Modellparameter aufgefasst. Zu Beginn einer modellierten Äußerung befindet sich der Filterzustand im Nullzustand, d.h. alle Zustandsvariablen außer dem Onset haben den Wert 0.

Eine modifizierte Form des Modells wurde von Birkholz, Kröger et al. (2011) zur Modellierung der Trajektorien verschiedener supraglottaler Artikulatoren (z.B. Zunge, Kiefer) vorgestellt. Dabei wurden konstante Targets betrachtet und der Verlauf der Trajektorien durch ein allgemeines kritisch gedämpftes System Nter Ordnung erzeugt. Birkholz und Hoole (2012) konnten schließlich zeigen, dass die Verwendung eines Filters der Ordnung $N = 5$ für die Target-Approximation artikulatorischer Trajektorien zu den besten Ergebnissen führt. Diese Untersuchungsergebnisse sollen im Rahmen dieser Arbeit auch auf das TAM der Grundfrequenz angewendet und die Abhängigkeit der Filterordnung untersucht werden. Die Gleichungen zur Beschreibung des TAM mit einem kritisch gedämpften Filter Nter Ordnung wurden in Birkholz, Kröger et al. (2011) für konstante Targets beschrieben und sind im Folgenden in modifizierter Form für lineare angegeben.

$$f_0(t;\mathbf{p}) = (mt+b) + (c_0 + c_1 t + \cdots + c_{N-1} t^{N-1}) e^{-\boldsymbol{\lambda} t} \tag{3.1}$$

$$f_0^{(n)}(t;\mathbf{p}) = \frac{d^n}{dt^n}(mt+b) + e^{-\boldsymbol{\lambda} t} \sum_{i=0}^{N-1} t^i \left(\sum_{j=0}^{\min\{N-1-i,n\}} (-\boldsymbol{\lambda})^{n-j} \binom{n}{j} c_{i+j} \frac{(i+j)!}{i!} \right) \tag{3.2}$$

$$\begin{aligned} c_0 &= f_0(0;\mathbf{p}) - b \\ c_n &= \Big(f_0^{(n)}(0;\mathbf{p}) - \frac{d^n}{dt^n}(mt+b) - \sum_{i=0}^{n-1} c_i (-\boldsymbol{\lambda})^{n-i} \binom{n}{i} i! \Big) / n! \end{aligned} \tag{3.3}$$

Gleichung 3.1 beschreibt wiederum die Grundfrequenz innerhalb einer Silbe charakterisiert durch ein silbenbezogenes PT $\mathbf{p}$. Der f_0-Verlauf ist an den Silbengrenzen synchronisiert, wobei der Zustand an den Silbengrenzen durch die Ableitungen definiert ist. Eine geschlossene Form des Grundfrequenzverlaufs einer mehrsilbigen Äußerung $f_0(t;P)$ lässt sich dadurch nicht angeben. Der Zeitpunkt $t=0$ in den obigen Gleichungen bezieht sich auf den Beginn der jeweiligen Targets und ist damit relativ definiert. Dies muss bei einer Implementierung berücksichtigt werden. Der Onset ϕ taucht in der obigen, mathematischen Beschreibung in direkter Form nur für die erste Silbe auf, wobei für diese dann $\phi = f_0(0;\mathbf{p})$ gilt.
Xu und Liu (2006) schlagen eine Erweiterung des TAM vor, indem sie die Frage aufwerfen, wie eine Silbengrenze definiert ist. Dabei schlussfolgern sie, dass die konventionelle Definition der Silbengrenze in artikulatorischen Äußerungen, wie sie auch zur Annotation des Korpus verwendet wurde, nicht optimal im Sinne des TAM ist. Dies liegt daran, dass durch den Artikulationsprozess, der im TAM durch ein lineares System Nter Ordnung modelliert wird, eine Verzögerung auftritt und die eigentlichen Silbengrenzen daher zeitlich eher angesetzt werden müssen. Die Untersuchungen von Xu und Liu (2006) legen nahe, dass dazu eine Verschiebung der konventionellen Grenzen, festgemacht an der akustischen Realisierung der Phoneme innerhalb eines Sprachsignals, um 26-48 msec nach vorn nötig ist. Da es dazu noch keine näheren quantitativen Untersuchungen gibt, soll dies im Rahmen dieser Arbeit geschehen. Die Silbengrenzenverschiebung τ stellt sich

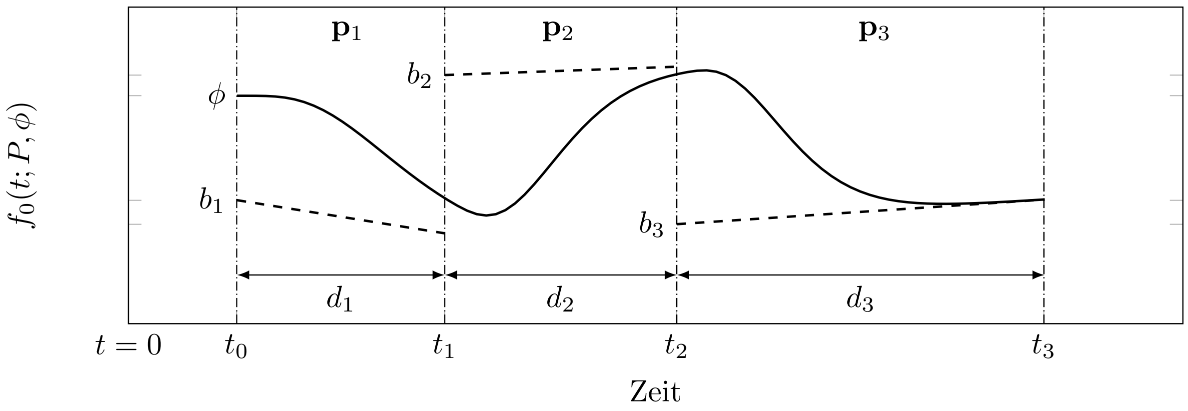

Abbildung 3.7: Modellierter Grundfrequenzverlauf nach dem TAM.

dabei wie die Filterordnung als qualitativer Parameter des TAM dar und kann auch als Freiheitsgrad bei der Modellierung betrachtet werden. Bei einer solchen Verschiebung werden nur die inneren Silbengrenzen einer Äußerung verschoben, d.h., die Grenzen von Äußerungsbeginn und -ende bleiben fixiert. Durch die Silbengrenzenverschiebung werden also die Silbendauern aller PTs verändert, die im Rahmen dieser Arbeit dem Korpus entnommen werden.

Parameterschätzung

Die Schätzung der TAM-Parameter, welche für das Training des Lernalgorithmus dienen soll, kann anhand der durch einen PDA ermittelten Grundfrequenzverläufe natürlicher Äußerungen geschehen. Da eine nach dem TAM modellierte Grundfrequenz eindeutig durch eine Folge von PTs und einem Onset-Wert beschrieben ist, kann die Schätzung anhand eines Abstandsmaßes, gemessen zwischen natürlicher und modellierter Grundfrequenz, erfolgen. Es existieren unterschiedliche Abstandsmaße um den Abstand zweier Signale zu bestimmen. Da die Signalwerte eines abgetasteten Signals einen Vektor bilden, können bekannte Vektornormen zur Abstandsbestimmung eingesetzt werden. Üblicherweise wird dabei auf die Klasse der p-Normen zurückgegriffen, die als $\|x\|_p = (\sum_{k=1}^{K} |x_k|^p)^{1/p}$ definiert ist. Welcher Wert für p gewählt wird, hängt dabei von der konkreten Anwendung ab, wobei die Summennorm ($p = 1$), euklidische Norm ($p = 2$) oder Maximumnorm ($p \to \infty$) oft eingesetzt werden. Der Abstand zweier Vektoren bzw. Signale ist damit als Norm des Differenzvektors gegeben. Für das hier besprochene Problem soll die euklidische Norm als Abstandsmaß dienen, da die Evaluation des Systems anhand des RMSE Kriteriums erfolgt, gemessen zwischen natürlicher und vorhergesagter f_0. Eine Minimierung des euklidischen Abstands geht dabei mit der Minimierung des RMSE einher, da beides quadratische Abstands-, bzw. Fehlermaße sind, und liefert damit die optimalen Ergebnisse.

Die Schätzung der Parameter ergibt sich als Lösung eines Optimierungsproblems, welches die Werte der PTs und Onset ermittelt, die eine modellierte Grundfrequenz erzeugen, welche einen minimalen euklidischen Abstand zur Natürlichen aufweist. Die so geschätzten Parameter können dann für ein Training verwendet werden. Eine Schätzung der Silbendauern muss dabei nicht erfolgen, da diese dem Korpus entnommen und somit als optimal angenommen werden können. Es sei darauf hingewiesen, dass für die Optimierung nur solche Abtastwerte verwendet werden, die in den stimmhaften Bereichen der natürlichen f_0 vorliegen. Die erhaltenen, optimalen Targets minimieren zwar das mathematische Abstandsmaß und erzeugen eine optimale Anpassung der Modellkurve an die natürliche f_0, dennoch bedeutet dies nicht, dass die erhaltenen Parameter auch physiologisch sinnvoll sind. An dieser Stelle sei nochmals auf den in Abschnitt 2.1 besprochenen Unterschied zwischen phonetischen und quantitativen Modellen hingewiesen. Im Rahmen eines quantitativen Modells würde man sich mit den so bestimmten Parametern zufrieden geben, da sie die natürliche Kurve optimal nachbilden und deren Parameter keinerlei physiologische oder artikulatorische Bedeutung haben. Im Rahmen eines phonetischen Modells hingegen ist es nicht nur das Ziel möglichst optimale, sondern auch möglichst natürlich interpretierbare Parameter zu bestimmen. Als unnatürlich werden hierbei PTs mit extrem großen Anstiegen oder Zeitkonstanten betrachtet. Ziel der späteren Vorhersage ist es darüber hinaus, möglichst natürlich interpretierbare Targets vorherzusagen. Dies wird zum einen dadurch erreicht, dass der Optimierung lineare Nebenbedingungen hinzugefügt werden, die den Parametersuchraum begrenzen. Zum anderen werden weitere nichtlineare Nebenbedingungen eingeführt, die solche extremen Werte der einzelnen Parameter, die als unnatürlich erachtet werden, zunehmend bestraft. Das Optimierungsproblem mit zugehörigen Nebenbedingungen ist damit wie folgt definiert.

$$
\begin{aligned}
(P^*, \phi^*) := \underset{\substack{P \in \mathbb{R}^{4 \times S} \\ \phi \in \mathbb{R}}}{\arg\min} \quad & \| f_0(k\Delta t) - f_0(k\Delta t; P, \phi) \|_2^2 && k \in \mathcal{V} && (3.4) \\
\text{s.t.} \quad & \phi_{\min} \le \phi \le \phi_{\max}; \quad \boldsymbol{p}_{\min} \le \boldsymbol{p}_s \le \boldsymbol{p}_{\max} && s = 1...S && (3.5) \\
& (\boldsymbol{p}_s - \bar{\boldsymbol{p}})^{\mathrm{T}} W (\boldsymbol{p}_s - \bar{\boldsymbol{p}}) \le \varrho && s = 1...S && (3.6)
\end{aligned}
$$

Die Quadrierung der Zielfunktion dient der Auflösung der Wurzel und spielt für die Minimierung keine Rolle, da die quadratische Funktion konvex ist. Die Nebenbedingungen wurden silbenweise definiert, da sie für alle Silben identisch sind. Dabei definiert $\boldsymbol{p}_{\min} = (m_{\min}, b_{\min}, \boldsymbol{\lambda}_{\min})^{\mathrm{T}}$ die untere Schranke des Suchraums physiologisch sinnvoller PTs, $\boldsymbol{p}_{\max} = (m_{\max}, b_{\max}, \boldsymbol{\lambda}_{\max})^{\mathrm{T}}$ die jeweils obere Grenze und $\bar{\boldsymbol{p}} = (\bar{m}, \bar{b}, \bar{\boldsymbol{\lambda}})^{\mathrm{T}}$ die jeweiligen unbestraften Parameterwerte, welche somit bevorzugt gewählt werden. Die Parameter $\phi_{\min}$ und $\phi_{\max}$ legen den Suchraum für den optimalen Onset fest und können ohne Einschränkungen mit Suchraumgrenzen des Verschiebungsparameters gleich gesetzt werden. Der durch die Nebenbedingungen 3.5 definierte Suchraum der Parameter einer gesamten Äußerung wird im folgenden mit $\mathcal{P}$ bezeichnet, wobei die Dimension dieser Menge von der Silbenanzahl S abhängt. Da die Silbengrenzen nicht explizit mit optimiert werden müssen, besitzt das Problem 3.4 genau $3S + 1$ Optimierungsvariablen. Die Matrix $W := \mathrm{diag}(w_m, w_b, w_{\boldsymbol{\lambda}})$ dient einer unterschiedlich starken Wichtung der Parameter, wodurch eine unterschiedlich starke Bestrafung der verschiedenen TAM-Parameter möglich ist und von den Autoren des Artikels MathWorks (2017) als Designmatrix beschrieben wird. Über den Parameter ϱ kann die Stärke der Bestrafung eingestellt werden. All diese Parameter müssen für eine praktische Umsetzung geeignet gewählt werden. In Gleichung 3.5 sei die Kleiner-Gleich-Operation elementweise definiert.

Aus den Gleichungen 3.1 - 3.3 ist erkennbar, dass die modellierte Grundfrequenz eine nicht konvexe Funktion beschreibt, wodurch auch die Zielfunktion des Optimierungsproblems 3.4 nicht konvex ist, was das Auffinden eines globalen Minimums erschwert. Eine Lösung für dieses nichtlineare Problem kann nur durch ein numerisches Optimierungsverfahren ermittelt werden. Erschwerend kommt hinzu, dass die Ableitung der Zielfunktion nicht in geschlossener Form angegeben werden kann, da die modellierte f_0 abschnittsweise definiert ist, was den Einsatz eines Gradientenabstiegsverfahrens verhindert. Es bleibt also nur die Klasse numerischer, ableitungsfreier Algorithmen zur Lösung des Problems. Der bekannteste Algorithmus dieser Klasse ist der Nelder-Mead-Simplex

Algorithmus. In dieser Arbeit soll jedoch der BOBYQA (engl. *Bounded Optimization by Quadratic Approximation*) Algorithmus von Powell (2009) angewendet werden, da dieser nach den Untersuchungen von Rios und Sahinidis (2013) im Allgemeinen eine bessere Leistungsfähigkeit zeigt. Außerdem ist der BOBYQA Algorithmus in vielen Software Bibliotheken implementiert. BOBYQA setzt voraus, dass der Parametersuchraum des Problems beschränkt ist, was für Problem 3.4 zutrifft, da die TAM-Parameter auf physiologisch sinnvolle Werte beschränkt werden.
Der BOBYQA stellt ein sogenanntes Trust-Region-Verfahren dar, bei denen die Zielfunktion durch eine Modellfunktion, in diesem Falle eine quadratische, approximiert wird. Das approximierte Optimierungsproblem stellt ein quadratisches Programm dar und kann durch Einsatz geeigneter numerischer Verfahren effizient gelöst werden, wobei beispielsweise ein Innere-Punkte-Verfahren angewendet werden kann (Boyd und Vandenberghe, 2004). In einem iterativen Prozess wird so jeweils über einem bestimmten Definitionsbereich, Trust-Region genannt, die Approximation bestimmt und die Lösung berechnet. Je nach Güte der Approximation erfolgt im nächsten Iterationsschritt eine Vergrößerung oder Verkleinerung der Trust-Region, bis schließlich ein Abbruchkriterium erreicht wird. Da bei einer solchen lokalen Suche das Auffinden eines globalen Minimums einer nicht konvexen Funktion nicht garantiert ist, führt man in der Praxis diesen Optimierungsprozess mehrmals hintereinander durch und variiert dabei beispielsweise den Initialisierungspunkt zufällig innerhalb des spezifizierten Suchraums. Da die Zielfunktion mit wachsender Silbenanzahl in der Äußerung zunehmend mehr lokale Extrema aufweist, wird hier die Anzahl der Zufallsinitialisierungen von der Silbenanzahl abhängig gemacht und auf $5S + 10$ festgelegt. Empirische Untersuchungen haben gezeigt, dass eine höhere Anzahl an Suchvorgängen zu keinen besseren Ergebnissen führen.
Möchte man nun also den BOBYQA Algorithmus auf das Problem 3.4 anwenden, muss dieses modifiziert werden, da der Algorithmus keine beliebigen Nebenbedingungen erlaubt. Die linearen Nebenbedingungen 3.5 stellen den begrenzten Parametersuchraum $\mathcal{P}$ dar und sind eine direkte Eingangsgröße für den Algorithmus. Die nichtlinearen Ungleichungsnebenbedingungen aus 3.6 hingegen müssen in die Zielfunktion eingearbeitet werden, damit der Algorithmus angewendet werden kann. Dies kann durch Bildung der Lagrang'schen Funktion geschehen, welche die selben Lösungen aufweist, wie das ursprüngliche Problem. Das resultierende Problem ist in Gleichung 3.7 gezeigt und für konkrete Werte von $\lambda > 0$ und ϱ äquivalent mit 3.4, wobei jedoch keine explizite Abhängigkeit zwischen den beiden Größen formuliert werden kann (Theodoridis, 2015).

$$\begin{aligned}(P^*, \phi^*) &:= \arg\min_{\mathcal{P}} \; \|f_0(k\Delta t) - f_0(k\Delta t; P, \phi)\|_2^2 + \sum_{s=1}^{S} \lambda_s \Big((\boldsymbol{p}_s - \bar{\boldsymbol{p}})^{\mathrm{T}} W (\boldsymbol{p}_s - \bar{\boldsymbol{p}}) - \varrho \Big) \quad k \in \mathcal{V} \\ &= \arg\min_{\mathcal{P}} \; \sum_{k \in \mathcal{V}} \Big(f_0(k\Delta t) - f_0(k\Delta t; P, \phi) \Big)^2 + \lambda \sum_{s=1}^{S} (\boldsymbol{p}_s - \bar{\boldsymbol{p}})^{\mathrm{T}} W (\boldsymbol{p}_s - \bar{\boldsymbol{p}}) \qquad (3.7)\end{aligned}$$

Jede der silbenbezogenen Nebenbedingungen wird durch Hinzufügen eines Lagrang'schen Multiplikators additiv in die Zielfunktion eingebracht. Diese können als Regularisierungsparameter aufgefasst werden. Um die Anzahl der Freiheitsgrade zu begrenzen, soll nur ein einziger Regularisierungsparameter betrachtet werden, der für alle Parameter

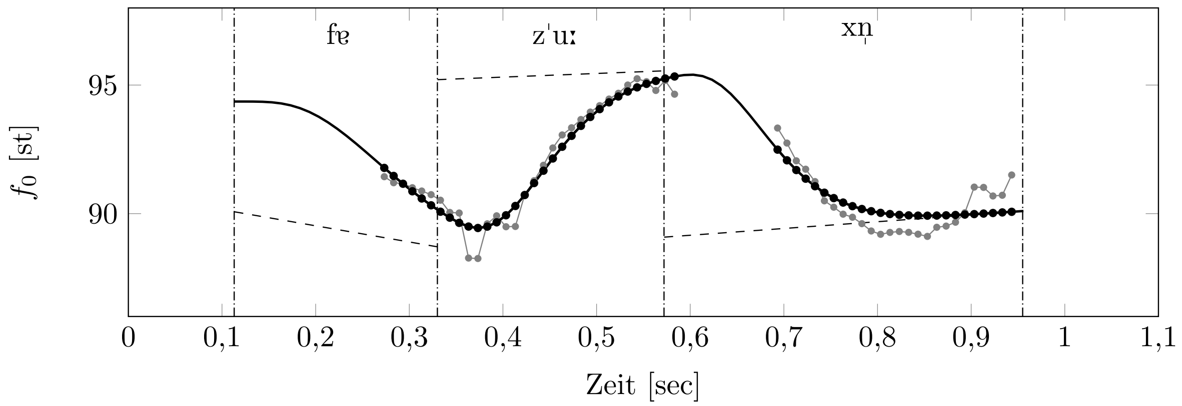

Abbildung 3.8: Modellierter Grundfrequenzverlauf nach dem TAM mit optimalen PTs und zugehöriger, natürlicher f_0 (grau hinterlegt).

gleich groß gewählt werden soll. Außerdem kann eine unterschiedlich starke Bestrafung der einzelnen Parameter über die Designmatrix W eingestellt werden, wodurch die Beschränkung $\lambda = \lambda_s$ ohne Einschränkungen vorgenommen werden kann. Die Summanden $\lambda_s \varrho$ sind konstant und haben keinen Einfluss auf die Position des Minimums, weshalb diese vernachlässigt werden können. Auch der Faktor $\frac{1}{K}$, resultierend aus der Norm, beeinflusst die Minimierung nicht und kann entfernt werden bzw. geht wiederum als konstanter Faktor mit in die Designmatrix W ein. Über den Regularisierungsparameter λ kann die Stärke der Regularisierung eingestellt werden.

Das modifizierte Optimierungsproblem 3.7 kann in dieser Form also mittels BOBYQA Algorithmus gelöst werden. Der zweite Summand der Zielfunktion sorgt dafür, dass die nichtlinearen Ungleichungsnebenbedingungen eingehalten und somit als Regularisierungsterm betrachtet werden kann. Abhängig vom Regularisierungsparameter λ werden extreme Werte der TAM-Parameter zunehmend bestraft, wobei sich dieser Prozess mittels $\bar{\boldsymbol{p}}$ und W steuern lässt. Das Ergebnis einer solchen Optimierung ist beispielhaft in Abbildung 3.8 dargestellt, wobei die optimalen TAM-Parameter für die Grundfrequenzkontur aus Abbildung 3.6 bestimmt wurden. Die gefundenen PTs bestimmen den Grundfrequenzverlauf dabei so, dass der Abstand zwischen natürlicher und modellierter Grundfrequenz minimiert wird. Aus Abbildung 3.8 wird ebenfalls ersichtlich, dass man sich den Optimierungsprozess in Gedanken gut veranschaulichen kann, indem man die Targets variiert und die erzeugte f_0 dementsprechend ändert.

Für eine praktische Implementierung stellt sich natürlich die Frage, wie die Parameter der Nebenbedingungen des Problems 3.7 zu wählen sind. Tabelle 3.4 listet die Zahlenwerte, die im Rahmen dieser Arbeit verwendet wurden und nachfolgend diskutiert werden. Die Implementierung wurde so angelegt, dass diese Werte vom Nutzer einstellbar sind. Die Wahl der Grenzen des Verschiebungsparameters $b_{\min}$ und $b_{\max}$ richtet sich nach dem erwarteten Frequenzbereich der f_0. Wiederum wurde nach der Angabe von Boersma und Weenink (2017) ein Suchbereich für die Frauenstimme von [100,500] Hz angenommen. Daraus wurden für die Implementierung $b_{\min} = 75$ st und $b_{\max} = 115$ st abgeleitet. Dies entspricht ebenfalls der Empfehlung im PENTAtrainer1, wo ein Suchraum von $b = \pm 20$

st um den mittleren f_0-Wert zu Äußerungsbeginn angegeben wird. Wie schon bereits erwähnt, können diese Grenzen ebenfalls ohne Einschränkungen für den Onset verwendet werden. Xu und Sun (2000) haben Untersuchungen vorgenommen, um eine maximale Änderungsrate der Grundfrequenz anzugeben, wobei ein Wert von $75\frac{\mathrm{st}}{\mathrm{sec}}$ ermittelt wurde. Dieser Wert kann als Grundlage dienen, um den Suchraum für den Anstieg der PTs abzuschätzen. Der Suchraum wurde auf $m_{\min} = -50\frac{\mathrm{st}}{\mathrm{sec}}$ und $m_{\max} = 50\frac{st}{sec}$ begrenzt, da Untersuchungen gezeigt haben, dass bei einer Wahl dieser Werte die zuvor genannte Bedingung fast nie verletzt wird. Auch die Annäherungsrate, bzw. inverse Zeitkonstante, sollte so gewählt werden, dass die erzeugten Grundfrequenzverläufe eine Rate von $75\frac{\mathrm{st}}{\mathrm{sec}}$ nicht überschreiten. Direkte Werte lassen sich leider nicht ableiten, da die maximale Änderungsrate der f_0 Kontur nicht nur von dem Target abhängt, sondern auch von der Systemordnung und dem Systemzustand. Prinzipiell lässt sich feststellen, je größer die Differenz zwischen f_0-Wert von Silbenanfang und Silbenende, desto größer muss die Zeitkonstante sein, damit die von Xu und Sun (2000) ermittelte Bedingung nicht verletzt wird. Aus diesen Gründen wurde die für den PENTAtrainer1 vorgeschlagenen Werte $\boldsymbol{\lambda}_{\min} = 1\frac{1}{\mathrm{sec}}$ und $\boldsymbol{\lambda}_{\max} = 80\frac{1}{\mathrm{sec}}$ übernommen, die sich darüber hinaus auf empirische Untersuchungen stützen (Prom-On et al., 2009). Alternativ wäre es ebenso möglich, bei der Optimierung f_0-Verläufe, die diese Nebenbedingung nicht erfüllen, zu bestrafen und eine zusätzliche Art der Regularisierung einzuführen. Eine Analyse der modellierten f_0-Verläufe des Korpus zeigt jedoch, dass diese Bedingung so gut wie nie verletzt wird und eine etwaige Bestrafung keinen Effekt hat.

Der Parameter $\bar{m}$ wurde so gewählt, dass Anstiege um den Wert 0 $\frac{\mathrm{st}}{\mathrm{sec}}$ nicht bestraft und damit bevorzugt werden. Für $\bar{b}$ wurde ein Werte von 95 st festgelegt, da dieser die Mitte des Suchbereichs für die Verzerrung darstellt und in der Nähe des sprecherspezifischen Mittelwertes des Korpus liegt. Eine Verbesserung kann erreicht werden, indem man den Mittelwert der natürlichen f_0, bezogen auf die jeweilige Äußerung, für $\bar{b}$ wählt, wodurch dieser sprecherunabhängig wird. Als bevorzugter Wert für die Annäherungsrate wurden 80 $\frac{1}{\mathrm{sec}}$ gewählt, da dieser einer Zeitkonstante von 12,5 msec entspricht, die einen üblichen Wert darstellt und davon abgesehen besonders oft bei einer unregularisierten Optimierung als optimal identifiziert wurde. Die absoluten Werte der Parameter w_m, w_b und w_λ spielen grundsätzlich keine Rolle, sondern nur deren Verhältnis zueinander, da dieses beschreibt, wie stark sich die Regularisierung auf die einzelnen Parameter auswirkt. Multipliziert man die entsprechenden Werte aus Tabelle 3.4 z.B. mit 80 ergeben sich die Zahlenwerte $\{1{,}6;\ 1{,}0;\ 0{,}25\}$, die besser erkennen lassen, dass der Parameter Anstieg am stärksten und der Parameter Annäherungsrate am schwächsten regularisiert wird. Dies liegt wohl auch vor allem daran, dass die einzelnen Parameter unterschiedliche Wertebereiche haben und betragsmäßig verschiedene Anteile zum Regularisierungsterm beisteuern. Eine willkürliche Multiplikation mit 80 hätte lediglich zur Folge, dass der äquivalente Regularisierungsparameter λ im selben Maße skaliert wird. Da dieser empirisch bestimmt wird und keinerlei natürliche Bedeutung aufweist, kann dessen Dimension beliebig gewählt werden, welche durch die Designmatrix W eingestellt werden kann. Durch die in Tabelle 3.4 gezeigten Werte wird eine zweckmäßige Skalierung vorgenommen, sodass die Parameterwerte im Bereich [-1,1] liegen und zusätzlich verschieden

$m_{\mathrm{min}} =$	-50 $\frac{\mathrm{st}}{\mathrm{sec}}$	$m_{\mathrm{max}} =$	50 $\frac{\mathrm{st}}{\mathrm{sec}}$	$\bar{m} =$	0 $\frac{\mathrm{st}}{\mathrm{sec}}$	$w_m =$	$\frac{1}{50}\ \frac{\mathrm{sec}^2}{\mathrm{st}^2}$	$N =$	var.
$b_{\mathrm{min}} =$	75 st	$b_{\mathrm{max}} =$	115 st	$\bar{b} =$	95 st	$w_b =$	$\frac{1}{80}\frac{1}{\mathrm{st}^2}$	$\tau =$	var.
$\lambda_{\mathrm{min}} =$	1 $\frac{1}{\mathrm{sec}}$	$\lambda_{\mathrm{max}} =$	80 $\frac{1}{\mathrm{sec}}$	$\bar{\lambda} =$	80 $\frac{1}{\mathrm{sec}}$	$w_\lambda =$	$\frac{1}{320}\ \mathrm{sec}^2$	$\lambda =$	var.
$f_{0,\mathrm{min}} =$	100 Hz	$f_{0,\mathrm{max}} =$	500 Hz					$\hat{\phi} \rightarrow$	var.

Tabelle 3.4: Freiheitsgrade der TAM-Parameterschätzung.

gewichtet werden. Beispielsweise könnte man versuchen, die Skalierung so zu wählen, dass regularisierter und unregularisierter Term der Zielfunktion die selbe Größenordnung aufweisen, was sich für das vorliegende Problem jedoch als schwierig erwiesen hat, da die Parameter sehr unterschiedliche Wertebereiche aufweisen. Prinzipiell ist jedoch nur deren Verhältnis zueinander wichtig und dem absoluten Wert keinerlei Bedeutung zuzumessen.

Onset-Schätzung

Bei einer quantitativen Modellierung nach dem TAM muss in jedem Falle eine Schätzung des Onsets erfolgen. Je nach Problemstellung stehen jedoch unterschiedliche Informationen für die Schätzung zur Verfügung. Bei einer Schätzung der TAM-Parameter auf Basis einer natürlichen f_0 können somit auch alle Abtastwerte dieser Grundfrequenz für die Onset-Schätzung benutzt werden, wodurch diese als äußerungsspezifische Schätzmethoden bezeichnet werden. Dies wurde beispielsweise im vorherigen Abschnitt angewendet, in dem der Onset durch eine Optimierung über der gesamten natürlichen f_0 berechnet wird. Diese Art der Onset-Schätzung wird im Folgenden mit $\hat{\phi} = \phi^*$ beschrieben. Alternativ könnte man die Onset-Schätzung von der Schätzung der restlichen TAM-Parameter separieren und den Wert unter Ausnutzung aller Abtastwerte der natürlichen f_0 durch eine Extrapolation berechnen, was im Weiteren als $\hat{\phi} = \phi_{\mathrm{ext}}$ bezeichnet wird. Prom-On et al. (2009) haben die Onset-Schätzung lediglich auf Basis des ersten aller stimmhaften Abtastwerte der natürlichen f_0 vorgenommen, d.h. $\hat{\phi} = f_0(k_{\mathrm{v0}}\Delta t)$. Diese Art der Schätzung liefert überwiegend bei Äußerungen, die mit stimmlosen Lauten beginnen, ungeeignete Werte.

Im Rahmen einer Grundfrequenzvorhersage basierend auf dem TAM muss eine Onset-Schätzung ohne die Informationen natürlicher f_0-Kurven auskommen, da diese dabei nicht bereitstehen. Aus diesem Grund muss die Schätzung durch geeignete, statistische Methoden ermittelt werden. Die einfachste Art ist die Ermittlung eines sprecherspezifischen Mittelwertes $\hat{\phi} = \bar{\phi}_{\mathrm{corp}}$, welcher vom jeweiligen Korpus abhängt. Für den verwendeten Korpus wurde dafür ein Wert von 94,06 st ermittelt. Alternativ könnte man auch eine Abhängigkeit des Onsets von der Anzahl der Silben vermuten und eine statistische Betrachtung dahingehend erweitern und die silbenabhängigen, sprecherspezifischen Mittelwerte des Onsets für die Schätzung verwenden, welche als $\hat{\phi} = \bar{\phi}_{\mathrm{corp}}(s)$ gekennzeichnet wird. Dadurch wird die zuvor beschriebene Schätzung verfeinert. Die

statistischen Schätzmethoden können dabei natürlich ebenfalls für die Problemstellung der Parameterschätzung verwendet werden.
Im Rahmen der vorliegenden Arbeit wurden nur die Schätzverfahren $\hat{\phi} = f_0(k_{\text{v0}}\Delta t)$, $\hat{\phi} = \bar{\phi}_{\text{corp}}$ und $\hat{\phi} = \bar{\phi}_{\text{corp}}(s)$ untersucht. Die Idee der optimalen Schätzung $\hat{\phi} = \phi^*$ wurde erst später entwickelt und aus zeitlichen Gründen war eine Implementierung und Wiederholung aller Untersuchungen nicht mehr möglich.

Lernstichprobe

Die aus Merkmaltransformation und Parameterschätzung ermittelten Werte werden paarweise miteinander zu Tupeln verknüpft, welche dann die Lernstichprobe bilden. Ein solches Tupel ist ein Element der Lernstichprobe und wird auch als Sample bezeichnet. Diese Lernstichprobe $\mathcal{D}$ kann dann für das Training eines Regressionsverfahrens benutzt werden, welches versucht, eine allgemeine Abbildung zwischen Merkmalen und TAM-Parametern einer Silbe zu ermitteln.
Die Lernstichprobe wird auf Basis von Hörbeispielen erstellt, welche als Beilage für das Aussprachewörterbuch von Krech et al. (2009) erstellt wurden und unter DeGruyter (2009) im wav-Dateiformat veröffentlicht. Alle Äußerungen wurden von einer professionell ausgebildeten Sprecherin eingesprochen und dienen den Erläuterungen über die deutsche Standardaussprache im Aussprachewörterbuch. Die Hörbeispiele umfassen überwiegend Einzelwörter, enthalten aber auch Phrasen, und beinhalten primär solche Beispiele, an denen der Nutzer des Buches auf Besonderheiten in der Aussprache aufmerksam gemacht werden soll. Darunter treten Wörter vieler verschiedener Kategorien auf, wobei im Besonderen Lehnwörter aus dem Englischen und Französischen, Namen von Persönlichkeiten und Orten sowie verschiedene Komposita hervorzuheben sind. Manche Wörter sind auch mehrfach vorhanden, unterscheiden sich jedoch in der Aussprache. Aus der genannten Funktion der Hörbeispiele wird klar, dass diese Zusammenstellung nicht als repräsentativ für die deutsche Sprache im Allgemeinen gelten kann. Dennoch bieten sie zahlreiche Beispiele für natürlich betonte Einzelwörter, die für das Training eines Lernalgorithmus verwendet werden sollen.
Alle Hörbeispiele wurden von einer Doktorandin der Abteilung Sprechwissenschaft und Phonetik der Martin-Luther-Universität Halle-Wittenberg unter Verwendung der Software Praat nach den Laut- und Silbengrenzen annotiert, und in einem Praat-spezifischen TextGrid-Dateiformat gespeichert. Außerdem wurde die SAMPA-Darstellung aller Hörbeispiele aus der IPA-Transkription des Wörterbuchs manuell bestimmt und zusätzlich die Silbengrenzen eingefügt, da diese im Aussprachewörterbuch nicht angegeben werden. Auf Basis der Akzentsymbole wurden die wortspezifischen Akzentmuster ebenfalls manuell bestimmt. SAMPA-Transkription und Akzentmuster eines jeden Hörbeispiels wurden in einer csv-Datei abgelegt. Die Zusammenstellung von Hörbeispielen, Silbengrenzenannotation und SAMPA-Transkription mit Akzentmustern bilden den Korpus, der für die Ermittlung der Lernstichprobe im Rahmen dieser Arbeit zur Verfügung stand. Das Aussprachewörterbuch beinhaltet insgesamt 133.909 Äußerungen, die sich in 400.123 Silben aufteilen, und damit eine Äußerung im Mittel aus 2,98 Silben besteht. Das Korpus

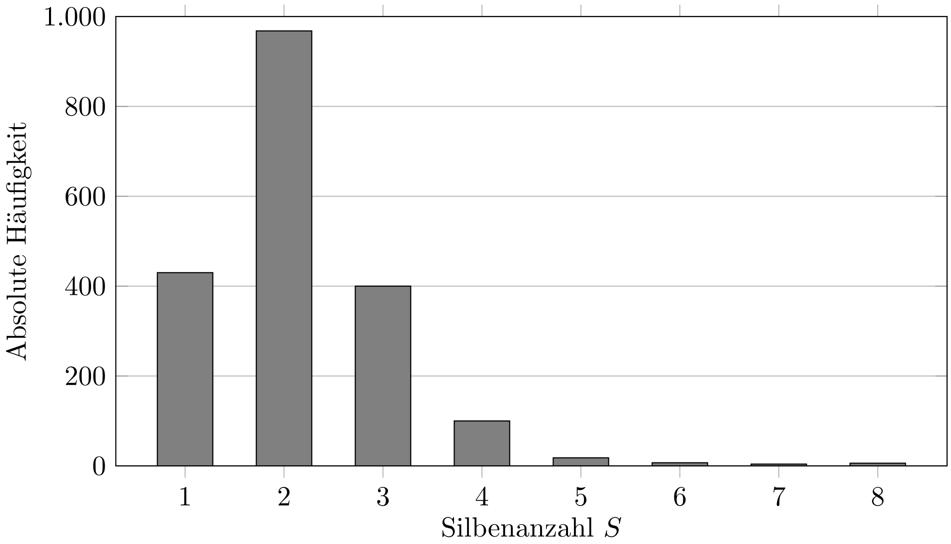

Abbildung 3.9: Absolute Häufigkeit der Äußerungen mit Silbenanzahl S für die 1.934 Elemente des Korpus.

beinhaltet insgesamt 2002 Äußerungen, von denen jedoch nur 1934 verwendet wurden, da sich die entwickelte Lösungsmethode auf Einzelwörter bezieht und alle Phrasen aussortiert wurden. Die Phrasen sind davon abgesehen nicht im Wörterbuch gelistet. Damit stehen für das Training des Lernverfahrens 4175 Silben zur Verfügung, wobei eine Äußerung im Mittel aus 2,16 Silben besteht. Abbildung 3.9 zeigt die genaue Häufigkeitsverteilung der Silbenanzahl für das verwendete Korpus. Entsprechende Silbeninformationen für das gesamte Korpus lagen nicht vor. Jedoch ist bekannt, dass das Wörterbuch 9.743 unterschiedliche Silben aufweist, von denen das Korpus 1.540 beinhaltet. Berücksichtigt man die Akzent- und Positionsinformation einer jeden Silbe, so umfasst das Wörterbuch mehrere zehntausend verschiedene Silben. Unter Berücksichtigung dieser Information lassen sich für das Korpus des Ausspracheworterbuchs 3.067 verschiedene Silben identifizieren.

Die genannten Zahlen verdeutlichen, dass der Umfang der zur Verfügung stehenden Lernstichprobe eher als gering einzustufen ist und diese durch die beschriebene Funktion der Hörbeispiele nicht repräsentativ für die deutsche Sprache ist. Aus diesem Grund sind keine herausragenden Ergebnisse für eine auf dem Korpus basierende Grundfrequenzvorhersage zu erwarten. Dennoch lässt sich die Leistungsfähigkeit der Methode daran abschätzen und auch ggf. auf eine größere Datenbasis anwenden.

Regressionsmodell

Ziel der vorliegenden Arbeit ist ein Vergleich verschiedener maschineller Lernverfahren bzw. Regressionsverfahren bzgl. deren Eignung zur Vorhersage einer Grundfrequenzkontur. Dabei sollen die in Abschnitt 2.3 beschrieben Verfahren untersucht und miteinander verglichen werden. Je nach Verfahren wird das multivariate Regressionsproblem in mehrere univariate Problem geteilt, die dann unabhängig voneinander gelöst werden können. Beim Einsatz eines MLP könnte das multivariate Problem theoretisch auch geschlossen gelöst werden, indem man vier Ausgangsneuronen für die vier Vorhersageparameter $\{m, b, \boldsymbol{\lambda}, d\}$ benutzt würde. Dies führt jedoch zu Einschränkungen in der Berechenbarkeit und kann als Spezialfall von vier separaten Netzen betrachtet werden. Vor dem Einsatz eines solchen Verfahrens zur Lösung eines konkreten Problems müssen im Vorhinein dessen Hyperparameter gewählt werden, um optimale Ergebnisse zu erzielen. Dies geschieht im Rahmen einer Modellauswahl, welche im nachfolgenden Abschnitt näher beschrieben ist.

Mit geeignet gewählten Hyperparametern kann das Regressionsmodell dann trainiert werden und für den Einsatz neuer Daten, die nicht in der Lernstichprobe enthalten sind, verwendet werden. Der Vorhersagefehler ist wie besprochen ein Gütemaß zur Beurteilung der vorhergesagten Parameter. Ein aussagekräftigeres Gütemaß zur Beurteilung des Gesamtsystems ist jedoch die Ähnlichkeit zwischen einer natürlichen und einer äquivalenten, vorhergesagten Grundfrequenzkontur, wobei letztere durch die bereits beschriebene Tiefpassfilterung aus den vorhergesagten TAM-Parametern erzeugt wird. Als Ähnlichkeitsmaße dienen RMSE und Korrelationskoeffizient ρ. Abbildung 3.2 veranschaulicht nochmals die unterschiedliche Betrachtung von Vorhersagefehler des Regressionsmodells und Ähnlichkeitsmaß bezogen auf das Gesamtsystem. Um für das Gesamtsystem charakteristische Werte der Ähnlichkeitsmaße zu erhalten, kann die Methode der k-fachen Kreuzvalidierung eingesetzt werden, wobei die Lernstichprobe randomisiert und in k Teilmengen geteilt wird. Durch dieses Verfahren werden die gesamten Daten der Lernstichprobe zur Ermittlung eines validen Evaluierungskriteriums genutzt. $k-1$ der Teilmengen (Trainingsmenge) werden dann für ein Training des Regressionsverfahrens genutzt und für die übrige Teilmenge (Testmenge) werden jeweilige Vorhersagen der TAM-Parameter durch das trainierte Modell gemacht. Anschließend werden die Ähnlichkeitsmaße zwischen den vorhergesagten und natürlichen f_0-Verläufen berechnet. Dieser Prozess wird insgesamt k-mal durchgeführt, wobei jede Teilmenge genau einmal als Testmenge fungiert und damit für jedes Element des Korpus eine Vorhersage vorliegt. Am Ende werden die mittleren Ähnlichkeitsmaße über das gesamte Korpus berechnet, welche dann eine Aussage über die Leistungsfähigkeit des Systems machen. In der Praxis hat sich ein Wert von $k = 10$ für die Kreuzvalidierung etabliert, welcher auch für die beschriebene Berechnung genutzt wurde (Theodoridis, 2015).

Das Beispiel einer mittels trainierten Modell vorhergesagten Grundfrequenz ist in Abbildung 3.10 gegeben. Die zugrunde liegende Äußerung ist dieselbe wie die der natürlichen Grundfrequenz aus Abbildung 3.6 und kann demnach mit dieser verglichen werden. Es sei nochmals darauf hingewiesen, dass die optimalen Targets aus Abbildung 3.8 dabei nicht für das Training verwendet wurden, sondern der Testmenge angehörten.

Damit ein Lernalgorithmus jedoch gut funktioniert, sind einige weitere Faktoren zu beachten wie beispielsweise eine angemessene Datenskalierung. Diese ist notwendig, um alle Merkmale auf den selben Wertebereich zu skalieren, damit nicht schon im Vorhinein einige Merkmale stärker als andere gewichtet werden. Hierbei soll die Skalierung wie in Hsu et al. (2016) vorgeschlagen verwendet werden, die als sogenannte Min-Max-Skalierung

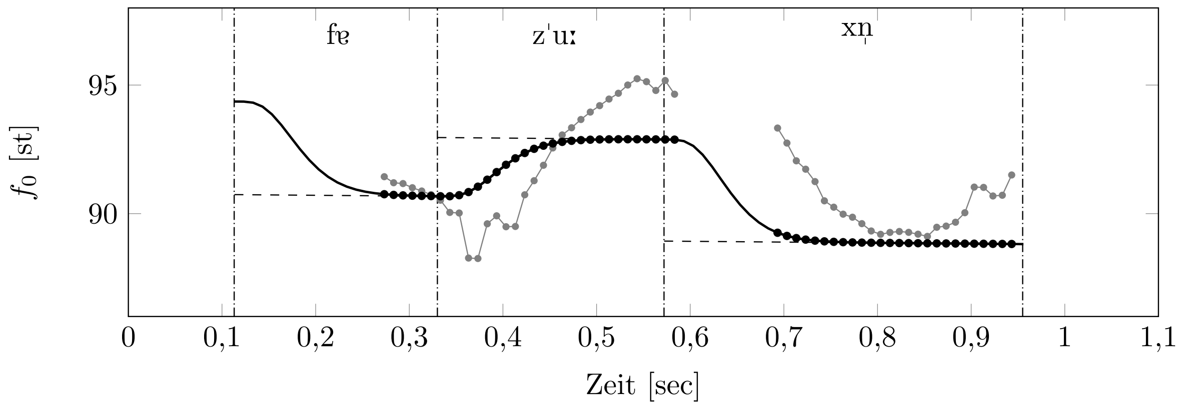

Abbildung 3.10: Vorhergesagter Grundfrequenzverlauf nach der entwickelten Lösungsmethode und äquivalente, natürliche f_0 (grau hinterlegt).

bekannt ist. Beim Einsatz neuronaler Netze werden die Daten vor dem Training auch oft bzgl. ihres Mittelwertes und der Varianz normiert.

Modellauswahl

Die verschiedenen Regressionsverfahren weisen eine unterschiedliche Anzahl von Hyperparametern auf, die in der Regel vor dem eigentlich Training des Modells bestimmt werden müssen und von der Lernstichprobe abhängig sind. Da sich keine analytisch optimalen Hyperparameter bestimmen lassen, muss auf geeignete, statistische Methoden zurückgegriffen werden, um diese zu bestimmen. Wichtig ist, dass die Hyperparameter für die vier univariaten Regressionsprobleme separat bestimmt werden müssen. So kann beispielsweise ein MLP zur Vorhersage der Verschiebung b eine völlig andere Struktur aufweisen als jenes zur Vorhersage des Anstiegs m.
Die LRR Methode enthält nur den Regularisierungsparameter C als Hyperparameter. In Rifkin und Lippert (2007) wird eine Methode erläutert, die auf einer Leave-One-Out-Kreuzvalidierung beruht und die Ermittlung von C mit in den Trainingsalgorithmus einbaut. Dieser Algorithmus soll auch bei der Implementierung der LRR genutzt werden, wodurch keine Hyperparameter in einer separaten Modellauswahl ermittelt werden müssen. Das selbe Prinzip soll ebenfalls für die KRR eingesetzt werden. Bei der KRR hängen die restlichen zu bestimmenden Hyperparameter vom verwendeten Kern ab. Für alle benutzten Kernmethoden wurde hierbei der Gauß-Kern verwendet, da dieser einen unendlich-dimensionalen RKHS generiert und damit eine große Flexibilität bei der nichtlinearen Regression aufweist. Außerdem konnten für viele praktische Probleme die besten Ergebnisse durch Einsatz eines Gauß-Kerns erzielt werden (Theodoridis, 2015). Es kann sogar analytisch gezeigt werden, dass die Verwendung eines Gauß-Kerns in vielen Fällen zu optimalen Ergebnissen führt (Schröder und Trouvain, 2003). Daher wird die Auswahl eines geeigneten Kerns für die Kernmethoden nicht in der Modellauswahl betrachtet, sondern dieser vorher festgelegt. Der Gauß-Kern ist auch daher gut geeignet,

da er im Gegensatz anderer nichtlinearer Kerne, wie beispielsweise einem Polynom-Kern, nur einen zusätzlichen Hyperparameter in das Problem einbringt, welcher dabei eine Varianz beschreibt. Alternativ zu der Definition in Abschnitt 2.3 wird der Gauß-Kern oft durch den Parameter $\gamma = \frac{1}{2\sigma}$ definiert, der eine inverse Varianz beschreibt. Im Rahmen der Modellauswahl der KRR muss also nur ein Wert für γ beim Einsatz der KRR bestimmt werden. Eine oft angewandte Heuristik bei der Wahl von γ ist als sogenannter „Median-Trick“ bekannt. Dabei wird durch die Berechnung des Mittelwerts der quadrierten Abstände zwischen allen paarweisen Merkmalvektoren der Lernstichprobe eine Schätzung für σ berechnet. Durch den Einsatz dieser Methode kommt auch die KRR ohne eine zusätzliche Modellauswahl aus (King, 2017).
Anders sieht dies für die SVR und das MLP aus, da sich dessen Hyperparameter nicht im Vorhinein auf plausible Werte festlegen lassen bzw. deren Bestimmung mit in den Trainingsalgorithmus integriert werden können. Für die Wahl solcher Parameter bleibt am Ende nur eine erschöpfende Suche auf Basis eines geeigneten Kriteriums. Die SVR beinhaltet von Grund auf einen Regularisierungsparameter C und die sogenannte Intensität ϵ, welche die Anzahl der Stützvektoren determiniert, als Hyperparameter. Bei

Methode	Parameter	Beschreibung	Bestimmung
LRR	C	Regularisierungsparameter	Methode nach (Rifkin und Lippert, 2007)
KRR	C	Regularisierungsparameter	Methode nach Rifkin und Lippert (2007)
	γ	Inverse Varianz des Gauß-Kerns	Median-Trick
SVR	C	Regularisierungsparameter	Erschöpfende Suche auf Basis des CVE
	ϵ	Intensität, determiniert Anzahl der Stützvektoren	
	γ	Inverse Varianz des Gauß-Kerns	
MLP	L_1	Neuronenanzahl der 1. verdeckten Schicht	Erschöpfende Suche auf Basis des CVE
	L_2	Neuronenanzahl der 2. verdeckten Schicht	
	α	initiale Lernrate bzw. Schrittweite	
	β	Wichtung des Momentum-Terms	

Tabelle 3.5: Übersicht über die Hyperparameter aller verwendeten Regressionsmethoden.

Verwendung eines Gauß-Kerns kommt zusätzlich der Parameter γ hinzu. Für die Bestimmung von γ für die SVR wurde nicht der Median-Trick verwendet, sondern auch die erschöpfende Suche angesetzt. Für das MLP ist ein Modell mit zwei verdeckten Schichten vorgesehen, wodurch die Anzahl der Neuronen in den Schichten als frei wählbare Parameter gegeben sind und mit L_1 bzw. L_2 bezeichnet werden. Zusätzlich bringt der Trainingsalgorithmus des MLP zwei weitere, freie Parameter ein, die durch die initiale Lernrate bzw. Schrittweite α und dem Wichtungsfaktor β des Momentum-Terms gegeben sind. Außerdem hat auch die Wahl der Aktivierungsfunktion einen großen Einfluss auf die Leistungsfähigkeit eines MLP. Für diese Arbeit wurde die Aktivierungsfunktion auf die Sigmoid-Funktion fixiert und nicht im Rahmen der Modellauswahl variiert. Die Sigmoid-Funktion zeigt für viele praktische Probleme sehr gute Ergebnisse und wurde deshalb verwendet . Eine Übersicht aller betrachteten Hyperparameter der vier verschiedenen Verfahren ist in Tabelle 3.5 zusammengestellt.

Um die Hyperparameter von SVR und MLP zu bestimmen, kann das Verfahren der k-fachen Kreuzvalidierung eingesetzt werden, wodurch ein geeignetes Kriterium für die Suche der Parameter definiert werden muss. Hierbei erfolgt wiederum eine Unterteilung der Lernstichprobe in $k = 10$ Teilmengen, die der Bildung geeigneter Trainings- und Testmengen dienen. Für alle k Testmengen wird also eine Vorhersage der TAM-Parameter berechnet und auf Basis der Lernstichprobe ein Gütemaß ermittelt, wobei in der Regel das MSE-Kriterium verwendet wird und hier als Vorhersagefehler bezeichnet wird. Durch den MSE wird ermittelt, wie gut das Lernverfahren generalisiert und für Vorhersagen unbekannter Daten geeignet ist. Der arithmetische Mittelwert aller k Vorhersagefehler wird als Kreuzvalidierungsfehler (engl. *Cross Validation Error*, CVE) bezeichnet. Die Verwendung aller Daten der Lernstichprobe innerhalb der Kreuzvalidierung garantiert dabei die bestmöglichen Ergebnisse, da alle vorliegenden Informationen genutzt werden.

Ziel ist es nun also, die Menge an Hyperparametern zu bestimmen, die den geringsten CVE aufweist und damit im Mittel den geringsten Vorhersagefehler. Dieses Problem stellt sich wiederum als ein nichtlineares Optimierungsproblem dar, für das keine explizite Ableitung angegeben werden kann. Im Unterschied zur vorher beschriebenen Parameterschätzung kann jedoch bei der Modellauswahl meist vorab keine plausible Einschränkung des Suchraums angeben werden, was den Einsatz lokaler, gradientenfreier Optimierungsverfahren ausschließt. Aus diesem Grund wird meist eine erschöpfende Suche über einem groben, logarithmischen Gitter (engl. Grid Search) eingesetzt, um passende Hyperparameter zu bestimmen. Der Aufwand einer solchen Suche steigt exponentiell mit der Anzahl der Hyperparameter. Wurden so jedoch optimale Werte auf dem groben Gitter bestimmt, kann das Ergebnis beispielsweise verfeinert werden, indem in der lokalen Umgebung des Optimums auf dem groben Gitter eine Suche über einem feineren Gitter initiiert wird oder aber auch das grobe Optimum zur Initialisierung eines lokalen, gradientenfreien Optimierungsverfahrens, wie beispielsweise dem BOBYQA-Algorithmus, dient. Für die Modellauswahl in dieser Arbeit wurde letztgenanntes Suchverfahren angewandt, wobei zu erwähnen sei, dass es eine Vielzahl weiterer Suchstrategien gibt. Ziel der Modellauswahl ist es jedoch nicht, ein eindeutiges, globales Minimum zu finden, sondern vielmehr die Hyperparameter auf eine gewisse Größenordnung festzulegen, die zu vernünftigen Ergebnissen führt. Praktisch zeigt sich nämlich, dass eine geringe Variation eines Hyperparameters kaum zu Änderungen des Vorhersagefehlers führt.

Resynthese

In bestimmten TTS-System können Laut- und Prosodiegenerierung als unabhängige Probleme betrachtet und auch gelöst werden. Eine Möglichkeit der Zusammenführung der beiden Ergebnisse ist die Anwendung des *Time-Domain-PSOLA*-Algorithmus (TD-PSOLA) nach Moulines und Charpentier (1990). Der TD-PSOLA-Algorithmus kann ebenfalls zur Änderung der Prosodie eines Sprachsignals benutzt werden. Um die wahrgenommene Natürlichkeit der vorhergesagten Grundfrequenzverläufe zu bewerten, sollen diese mittels TD-PSOLA auf die originalen Sprachsignale aufgeprägt werden, wodurch die originale mit der vorhergesagten Grundfrequenz ersetzt wird. Eine solche Untersuchung ist notwendig, da die mathematischen Ähnlichkeitsmaße zur Bewertung des Systems keinerlei Aussagen über die Natürlichkeit der generierten Grundfrequenz machen.

Der TD-PSOLA-Algorithmus teilt das zu manipulierende Sprachsignal in kleine, überlappende Segmente. Eine Veränderung der Grundfrequenz kann durch geeignetes Zusammen- oder Auseinanderrücken der Segmente erreicht werden. So kann die Grundfrequenz einer Sprachäußerung verändert bzw. manipuliert werden. Als Ergebnis dieses Prozesses liegt dann das originale Sprachsignal mit der vorhergesagten Grundfrequenz vor. Das so manipulierte Signal kann dann dazu dienen, die Natürlichkeit der vorhergesagten Grundfrequenz zu beurteilen. Es sei zu erwähnen, dass die vorhergesagte f_0 eine kontinuierliche Kurve darstellt, die beliebig abgetastet werden kann. Informationen über stimmhafte und stimmlose Abschnitte liegen nicht vor, können jedoch durch eine Lautdauervorhersage abgeschätzt werden. Die Abtastzeitpunkte der natürlichen f_0, wie sie auch in Abbildung 3.10 eingezeichnet sind, werden nur zur Bestimmung der Ähnlichkeitsmaße benutzt. Das Fehlen dieser Information spielt beim Einsatz der PSOLA-Resynthese jedoch keine Rolle, da die synthetisierten Laute bereits in Form des Sprachsignals vorliegen und bei stimmlosen Lauten keine f_0 vorliegt. Dadurch wird trotz des Aufprägens in diesen Abschnitten keine Stimmhaftigkeit wahrnehmbar.

Perzeptionstest

Um letztendlich die Natürlichkeit der vorhergesagten Grundfrequenz zu beurteilen, wurde ein Perzeptionstest mit mehreren Probanden durchgeführt. Der Test wurde dabei ähnlich wie von Prom-On et al. (2009) beschrieben konzipiert. Um ein statistisch aussagekräftiges Ergebnis zu erhalten, nahmen 25 Probanden am Versuch teil, was eine gängige Größe für Versuche dieser Art ist. Die Probanden waren deutsche Muttersprachler und im Alter zwischen 22 und 40 Jahren, wobei 7 weibliche und 18 männliche Probanden am

Versuch teilnahmen. Jeder Proband bekam 20 zufällig aus dem Korpus ausgewählte Äußerungen in sieben verschiedenen Varianten präsentiert. Dabei sollten die natürlichen, die optimal geschätzten, sowie die zwei besten vorhergesagten Grundfrequenzverläufe miteinander verglichen werden. Mit der besten vorhergesagten f_0 ist dabei jene mit den besten Ähnlichkeitsmaßen gemeint. Außerdem sollte für alle Varianten der Einfluss der Silbengrenzenverschiebung untersucht werden. Damit ergaben sich die in Tabelle 3.6 zusammengefassten Stichproben, welche im Perzeptionstest untersucht wurden.
Jeder Proband bekam somit insgesamt 140 Wörter präsentiert, deren Reihenfolge randomisiert wurde, und sollte auf einer Skala von 5 bis 1 die subjektiv empfundene Natürlichkeit aller Äußerungen beurteilen. 5 steht dabei für sehr natürlich und 1 für sehr unnatürlich. Jedes Wort durfte maximal einmal wiederholt werden. Bei der Verwendung der erwähnten nominalen Skala können die Ergebnisse anschließend nach dem *Mean-Opinion-Score* (MOS) verglichen werden. Die Ergebnisse eines solchen Vergleichs müssen jedoch differenziert betrachtet werden, da der Vergleich basierend auf dem MOS gewisse methodische Schwächen aufweist (Rosenberg und Ramabhadran, 2017). Da im Rahmen dieser Arbeit durch das verwendete Korpus jedoch nur Tendenzen anstatt absoluter Grenzen untersucht werden können, ist ein Vergleich auf Basis des MOS grundsätzlich legitimiert. Umfangreicher angelegte Untersuchungsverfahren waren im zeitlichen Rahmen dieser Arbeit leider nicht durchführbar. Durch den ermittelten MOS können außerdem Vergleiche zu Ergebnissen anderer Untersuchungen gemacht werden. Zusätzlich zum MOS sollen die Ergebnisse durch geeignete Hypothesentests statistisch ausgewertet werden.

Nr.	Stichprobe	N	τ/msec	λ	$\hat{\phi}$
1	originale f_0	-	-	-	-
2	modellierte f_0 mit optimal geschätzten PTs	5	0	0	$f_0(k_{\mathrm{v0}}\Delta t)$
3	modellierte f_0 mit optimal geschätzten PTs und verschobenen Silbengrenzen	5	-40	0	$f_0(k_{\mathrm{v0}}\Delta t)$
4	modellierte f_0 mit vorhergesagten PTs durch SVR	5	0	75	$\bar{\phi}_{\mathrm{corp}}$
5	modellierte f_0 mit vorhergesagten PTs durch SVR und verschobenen Silbengrenzen	5	-40	75	$\bar{\phi}_{\mathrm{corp}}$
6	modellierte f_0 mit vorhergesagten PTs durch MLP	5	0	75	$\bar{\phi}_{\mathrm{corp}}$
7	modellierte f_0 mit vorhergesagten PTs durch MLP und verschobenen Silbengrenzen	5	-40	75	$\bar{\phi}_{\mathrm{corp}}$

Tabelle 3.6: Untersuchte Stichproben im Perzeptionstest.

Der Perzeptionstest fand in einem schallisolierten Audio-Studio statt, um störende Effekte auf die Probanden zu minimieren. Außerdem wurde die externe USB-Soundkarte Terratec Aureon XFire 8.0 HD sowie Studiokopfhörer der Modellreihe STAX SR-202 verwendet, um eine hohe Qualität der abgespielten Äußerungen zu gewährleisten. Die Probanden wurden nicht darüber informiert, dass sich auch die originalen Äußerungen unter den gehörten Beispielen befinden. Der Perzeptionstest wurde mit der Software Praat durchgeführt, wobei die Version 6.0.28 verwendet wurde. Die 20 verschiedenen Äußerungen für jeden Probanden wurden zufällig ausgewählt, wobei jeder Proband 20 andere Äußerungen zu hören bekam. Damit wurden also 500 der 1934 Äußerungen für den Test verwendet. Die im Korpus enthaltenen Fremdwörter und Namen wurden vorher weitestgehend aussortiert, um sicherzustellen, dass die gehörten Wörter und dessen Betonung dem Probanden bekannt sind.

3.3 Implementierung

Programmstruktur

Im Blockdiagramm 3.1 wurden alle Komponenten des Gesamtsystems zusammengefasst. Für eine praktische Implementierung des vorgeschlagenen Systems muss jede Komponente in geeigneter Weise softwareseitig umgesetzt werden. Dabei wurden verschiedene Anforderungen an die Implementierung gestellt, wobei ausschließlich quelloffene Software für die Umsetzung verwendet werden sollte. Dies dient in erster Linie einer nahtlosen Integration der entstandenen Software in andere Projekte sowie einer unproblematischen Weiterentwicklung. Außerdem sollte eine nutzerfreundliche Bedienoberfläche nicht vernachlässigt werden. Zusätzlich ist zu beachten, dass das Lösen der verschiedenen Optimierungsaufgaben äußerst rechenintensiv ist und die Möglichkeit einer Parallelisierung der Software in Betracht gezogen werden muss. An der TU Dresden besteht die Möglichkeit, Berechnungen auf einem linuxbasierten Hochleistungsrechner (engl. *High Performance Computer*, HPC) durchzuführen, welcher für dieses Projekt genutzt werden sollte. Auch die Plattformunabhängigkeit der Software ist ein wichtiger Aspekt, da am betreuenden Lehrstuhl ausschließlich Windows-Rechner benutzt werden und die Software auch auf diesen lauffähig sein soll. Für den Einsatz der teilweise komplexen Signalverarbeitungs-, Optimierungs- und Maschinenlernalgorithmen, welche für die Implementierung notwendig sind, soll auf passende Softwarebibliotheken oder Programme zurückgegriffen werden.

Aus den beschriebenen Anforderungen wurde abgeleitet, die Programmiersprache C++ für die Implementierung zu verwenden. Entsprechende quelloffene Compiler sowie plattformunabhängige Bibliotheken stehen dabei zur Verfügung. Für die Implementierung wurde die Dlib-Bibliothek intensiv genutzt, welche unter der Boost Software Lizenz verwendet werden darf. Außerdem lässt sich C++ Code gut parallelisieren und ermöglicht eine effiziente Realisierung der vorgeschlagenen Lösungsmethode. Zusätzlich ist es ohne größeren Aufwand möglich, den Code in andere Projekte zu integrieren, gerade in

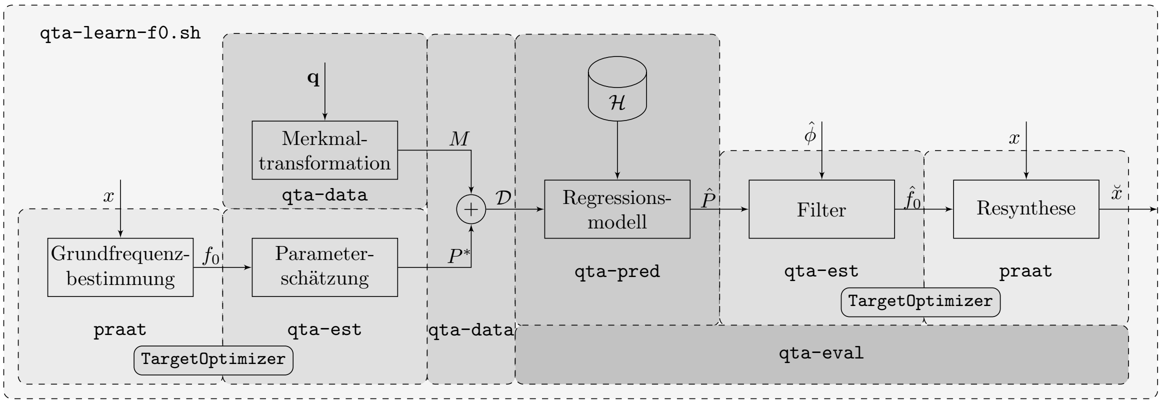

Abbildung 3.11: Softwareseitige Einteilung des entwickelten Systems.

Hinblick darauf, dass andere Projekte des betreuenden Lehrstuhls ebenfalls in C++ umgesetzt wurden. Die Entwicklung wurde auf einem Linux-System vorgenommen, wobei der gcc Compiler der Version 6.3 verwendet wurde. Bei der Programmierung wurden die Möglichkeiten und Paradigmen des C++11-Standards verwendet, sowie ein strikter, objektorientierter Stil eingehalten. Rechenintensive Berechnungen wurden auf dem Hochleistungsrechner des Zentrums für Informationsdienste und Hochleistungsrechnen (ZIH) der TU Dresden vorgenommen. Dabei standen zehn Rechenknoten zur Verfügung, jeweils ausgestattet mit einem Intel-Prozessor der Haswell-Architektur mit 24 Kernen und 62 TB Speicher.

Bei der Umsetzung war es wichtig, einen strikt modularen Aufbau zu erzielen, damit die einzelnen Systemkomponenten auch unabhängig voneinander ausgeführt und untersucht werden können. Eingangs- und Ausgangsdaten einer jeden Komponente müssen dazu in jeweiligen Dateien gespeichert werden und können somit für eine Analyse genutzt werden. Die Software wurde aus diesen Gründen so konzipiert, dass jede Systemkomponente in ein eigenständiges Programm abgebildet wird. Die Funktionalität des Gesamtsystems wird dann in einem bash-Skript realisiert, welches die einzelnen Programme aufruft und sich um die Dateieingabe und -ausgabe kümmert. Alle Komponenten können so zentral an einer Stelle im bash-Skript konfiguriert und auch entsprechend ein- und ausgeschaltet werden. Somit ist ein effizientes Werkzeug für die verschiedenen Untersuchungen gegeben. Die Software wurde dabei so angelegt, dass eine einfache Integration zu einem einheitlichen C++ Programm in Form einer Binärdatei möglich ist. Abbildung 3.11 visualisiert die Einteilung des Gesamtsystems in die einzelnen, unabhängigen Programme. Die Umsetzung und Funktionalität der separaten Programme wird im Folgenden detaillierter beschrieben. Jedes Programm ist dabei kommandozeilenbasiert und kann über Kommandozeilenargumente konfiguriert werden.

Nachfolgend wird die Funktionsweise der einzelnen Programme näher erläutert. Für weiterführende implementierungsspezifische Details sei der Leser auf den kommentierten Programmcode sowie den Anwendungsbeschreibungen der einzelnen Programme verwiesen, die über eine entsprechende Hilfe-Option abrufbar sind. All diese Informationen finden sich auf dem zugehörigen Datenträger zu dieser Arbeit.

Dlib

Zur Entwicklung aller hier beschriebenen Programme wurde die C++ Bibliothek Dlib eingesetzt (King, 2017). Diese stellt eine große Auswahl nützlicher Funktionen bereit, ist sehr gut dokumentiert, übersichtlich implementiert und wurde bereits in einer Vielzahl wissenschaftlicher und kommerzieller Projekte verwendet. Unter anderen wurden die Module für String-, XML- und Kommandozeilen-Parsing, Statistik, Lineare Algebra und Optimierung verwendet. Außerdem bietet die Bibliothek zahlreiche Möglichkeiten zur Parallelisierung des Codes an, welche auf dem POSIX-Threads-Modell basiert. Ein weiterer großer Vorteil ist die Plattformunabhängikeit aller betriebssystemspezifischer Funktionen, die durch geeignete API-Wrapper bereitgestellt werden. Damit lässt sich der auf dieser Bibliothek basierende Code auf verschiedensten Betriebssystemen problemlos kompilieren. Außerdem hat es sich als sehr vorteilhaft erwiesen, dass viele Algorithmen auch auf Datentypen der C++ Standard-Template-Library (STL) anwendbar sind bzw. diese einfach in von der Bibliothek bereitgestellte Datentypen umgewandelt werden können. Die Dlib verfügt darüber hinaus über ein Maschinenlern-Modul, welches eine Vielzahl maschineller Lernverfahren bereitstellt, die von verschiedenen Kernmethoden bis zu tiefen Neuronalen Netzen reichen (King, 2009). Details dieses Moduls werden im Weiteren noch beschrieben und mit anderen Bibliotheken verglichen.

Alles in allem stellt die Dlib ein umfangreiches Werkzeug zur Implementierung vielfältiger Problemstellungen dar und erfüllt alle im vorigen Abschnitt benannten Anforderungen. Unter Verwendung der dieser Bibliothek wurde eine übersichtliche Implementierung des Gesamtsystems umgesetzt. Die Dlib-Implementierung selbst kann zusätzlich als sehr schlank beurteilt werden, da sie sich auf die wesentlichen Funktionalitäten konzentriert. Viele Teile der Bibliothek sind header-only und müssen damit nicht extra kompiliert werden. Dadurch ist eine Implementierung auf Basis der Dlib gut für eine Integration innerhalb anderer Projekte geeignet.

Praat

Praat ist ein quelloffenes Software-Paket zur wissenschaftlichen Untersuchung von Sprache aus phonetischer Sicht (Boersma und Weenink, 2017). Es wurde an der Universität Amsterdam entwickelt und ist hauptsächlich in C++ geschrieben, wodurch es sich auch auf dem benannten HPC kompilieren lässt. Es lässt sich durch eine Skriptsprache steuern und stellt viele Routinen zur Bearbeitung von Sprachkorpora bereit, wobei sich die Skripte auch in eigenständige C++ Programme in Form einer Binärdatei kompilieren lassen. Dadurch lässt sich Praat sehr gut in den beschriebenen Programmaufbau integrieren. Praat stellt ein diverses Angebot an Algorithmen der digitalen Sprachsignalverarbeitung zur Verfügung worunter sich unter anderen der in Abschnitt 3.2 beschriebene PDA sowie der TD-PSOLA-Algorithmus befindet. Da Praat eine sehr breite Anwendung in der Phonetik findet, liegen die im Korpus annotierten Silbengrenzen in einem Praat-spezifischen TextGrid-Dateiformat vor.

Zur Vorbereitung der vollautomatischen Datenverarbeitung im Rahmen der Grundfrequenzvorhersage wurde ein Praat-Skript erstellt, welches alle Äußerungen des Korpus einliest, durch Anwendung des beschriebenen PDA die natürliche Grundfrequenz bestimmt und diese schließlich wieder in einem Praat-spezifischen PitchTier-Dateiformat speichert. Zur Beschreibung der Grundfrequenz wurde eine Halbtonskala, normiert auf 1 Hz, verwendet. Da es bei der Bestimmung der Grundfrequenz zu Oktavfehlern kommen kann, wurden die so bestimmten Grundfrequenzverläufe nochmals manuell kontrolliert und wenn nötig korrigiert. Eine Korrektur des Algorithmus war hierbei bei rund 5% der Äußerungen notwendig.

TargetOptimizer

Ursprünglich war vorgesehen das Tool PENTAtrainer1 für die Schätzung der optimalen TAM-Parameter zu verwenden. Dieses Tool sollte jedoch angepasst werden, so dass nicht nur ein lineares System dritter Ordnung für die Target-Approximation verwendet werden kann, sondern ein System beliebiger Ordnung N. Da der Quellcode des Programms qtalearn jedoch nicht zugänglich war, wurde ein eigenes Tool entwickelt, ebenfalls bestehend aus einem Praat-Skript (TargetOptimizer.praat) und einem C++ Programm (qta-est), welche die jeweiligen Aufgaben übernehmen. Im Zuge dessen wurden mehrere Veränderungen und Verbesserungen vorgenommen, die im Nachfolgenden näher beschrieben sind.

Die Grundfunktionalität des PENTAtrainer1 zur Bestimmung der optimalen Parameter wurde für die Neuimplementierung übernommen. Somit wird dem Nutzer eine grafische Oberfläche angeboten, in der alle nötigen Freiheitsgrade für die Parameterschätzung eingegeben werden können, welche in Tabelle 3.4 aufgelistet sind. Zusätzlich muss der Nutzer die Dateipfade der Ein- und Ausgabedaten angeben und kann wählen, welche Daten generiert werden sollen. Durch das Praat-Skript wird das Korpus verarbeitet, wobei zunächst die Praat-spezifischen Eingangsdaten eingelesen werden, die durch das Sprachsignal, die Silbengrenzen sowie den Grundfrequenzverläufen gegeben sind. Werden keine Grundfrequenzverläufe bereitgestellt, werden diese automatisch von Praat ermittelt. Die relevanten Informationen aller Eingangsdaten für die TAM-Parameterschätzung werden dann in kompakter Form in eine Textdatei geschrieben und an das Programm qta-est übergeben, welches diese einliest und die Werte der optimalen Parameter, zugehöriger f_0 und Ähnlichkeitsmaße berechnet, welche schließlich wieder dem Praat-Skript durch eine Textdatei übergeben werden. Der zugehörige Informationsfluss ist in Abbildung 3.12 dargestellt. Im Unterschied zum PENTAtrainer1 werden die Grundfrequenzverläufe vor der Optimierung nicht interpoliert und geglättet, da dadurch Informationen an den stimmlosen Abschnitten angegeben werden, obwohl keine f_0 vorhanden sind, was die folgende Optimierung beeinflusst und damit nicht die optimalen Ergebnisse liefert. Die hier entwickelte Lösung betrachtet nur die vom PDA bestimmten Abtastzeitpunkte des Grundfrequenzverlaufs bzw. die korrigierten Abtastzeitpunkte. Ziel der Optimierung ist es vordergründig, möglichst gute PTs für die stimmhaften Abschnitte zu finden, denn in den stimmlosen Abschnitten spielt die modellierte f_0 weitestgehend keine Rolle. Nach

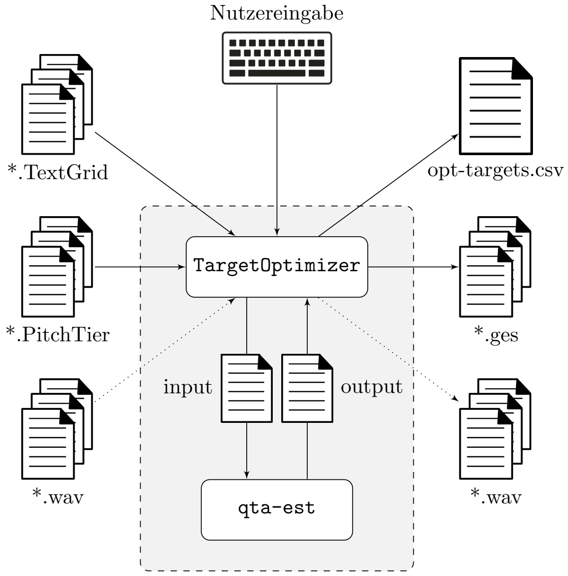

Abbildung 3.12: Informationsflussdiagramm des entwickelten Tools TargetOptimizer.

der Optimierung werden die optimalen Targets sowie der modellierte Grundfrequenzverlauf in separate Textdateien geschrieben. Dieser Prozess wird nun für alle Elemente des Korpus durchgeführt und am Ende schließlich in einer csv-Datei mit allen optimalen PTs des gesamten Korpus gespeichert.

Eine weitere Funktionalität des Praat-Skripts, neben der Schätzung der optimalen Parameter, ist die PSOLA-Resynthese. In diesem Modus wird neben den Korpus-Eingangsdaten eine csv-Datei mit geschätzten oder vorhergesagten Targets eingelesen, die modellierte Grundfrequenz berechnet und schließlich eine PSOLA-Resynthese durchgeführt. Für die Berechnung des Grundfrequenzverlaufs wird wiederum das Programm qta-est aufgerufen, welches geeignete Filter implementiert. Alle nötigen Informationen werden dabei abermals durch Textdateien ausgetauscht. Für die PSOLA-Resynthese werden die Sprachsignale des Korpus eingelesen und mit der modellierten Grundfrequenz manipuliert, wobei der in Praat implementierte TD-PSOLA-Algorithmus Anwendung findet. Als Ergebnis liegt dann für jedes Element des Korpus ein manipuliertes Sprachsignal vor, welches für den Perzeptionstest verwendet werden kann. Zusätzlich werden die Ähnlichkeitsmaße RMSE und Korrelationskoeffizient ρ für alle modellierten Grundfrequenzverläufe bestimmt und ebenfalls in einer csv-Datei gespeichert.

Außerdem wurde als problematisch angesehen, dass im PENTAtrainer1 die Ähnlichkeitsmaße nicht bezogen auf eine Äußerung berechnet werden, sondern silbenweise mit anschließender Mittelwertbildung. Dies führt zu einer Verzerrung des Fehlers und wurde für das Tool TargetOptimizer abgeändert.

Abschließend sei noch erwähnt, dass das Skript um die Funktionalität der Silbengrenzenverschiebung, wie in Abschnitt 3.2 beschrieben, erweitert wurde. Diese kann als freiwählbarer Parameter eingestellt werden und verändert entsprechend die Werte der im Korpus vorgegeben Silbengrenzen.

qta-est

Das Tool qta-est implementiert die in Abschnitt 3.2 beschriebene Parameterschätzung und stellt dabei eine Erweiterung des Tools qtalearn von Xu und Prom-On, 2015 dar. Die Optimierung selbst erfolgt dabei nicht durch eine erschöpfende Suche über einem Gitter ganzer Zahlen, sondern durch Anwendung des BOBYQA Algorithmus über einem reellen Zahlenraum. Eine Implementierung des BOBYQA Algorithmus wurde dabei durch die Dlib bereitgestellt. Der Algorithmus wird mehrfach hintereinander angewendet, wobei jedes mal ein zufälliger Initialisierungspunkt innerhalb des Suchraums gewählt wird. Dieser Prozess lässt sich sehr einfach und effizient durch eine parallele for-Schleife beschleunigen. Außerdem erfolgt die Optimierung nicht silbenweise, sondern betrachtet die komplette Äußerung mit all ihren Silben gleichzeitig. Zusätzlich wurde im Tool qta-est die in Gleichung 3.7 beschriebene Regularisierung umgesetzt, die eine Optimierung mit passenden Nebenbedingung darstellt, um physiologisch sinnvolle PTs zu finden. Eine letzte Erweiterung stellt die Implementierung des kritisch gedämpften Filters Nter Ordnung nach Gleichung 3.1 dar. Darüber hinaus wird der Filterzustand, durch analytische Ableitungen an den Silbengrenzen berechnet, im Gegensatz zur Implementierung in qtalearn, die numerische Ableitungen benutzt.
Als Voraussetzung für die PSOLA-Resynthese bietet das Tool die Funktionalität der Target-Approximation bzw. Filterung von eingegebenen PTs. Dabei werden keine optimalen Targets gesucht, sondern für gegebene der damit verbundene Grundfrequenzverlauf ausgegeben. Dieser Verlauf kann dann für eine PSOLA-Resynthese verwendet werden.

qta-data

Das Programm qta-data erfüllt im Wesentlichen die Funktion der Merkmaltransformation aber auch die Verknüpfung von Merkmalen und optimalen PTs zur Lernstichprobe. Da die Merkmaltransformation eine Konvertierung der SAMPA-Darstellung in eindeutige, mathematische Vektoren darstellt, verarbeitet dieses Tool vor allem Strings. Dazu wurden viele von der Dlib bereitgestellten Stringverarbeitungsalgorithmen sowie die regex-Implementierung der STL zur Verarbeitung regulärer Ausdrücke eingesetzt.
Die Merkmaltransformation entspricht im Prinzip einer Umsetzung der in Abschnitt 3.2 abgebildeten Tabellen. Das Akzentmuster einer Äußerung, eindeutig beschrieben durch die jeweiligen Symbole für Haupt- und Nebenakzente, musste dabei nicht gesondert bestimmt werden, da dieses bereits explizit dem Korpus entnommen werden konnte. Die Muster wurden dabei manuell von den Erstellern des Korpus ermittelt. Damit waren alle Akzentmerkmale explizit gegeben. Hauptaufgaben waren somit die Zerlegung der SAMPA-Darstellung in die einzelnen Wörter bzw. Silben, die Identifizierung der Silbenbestandteile Onset, Nukleus und Koda, die Zuordnung der Phoneme zu den Vektoren nach den Tabellen 3.1 und 3.2 basierend auf den distinktiven Merkmalen sowie die Verarbeitung der jeweiligen Vokal- und Konsonantenmodifizierer aus Tabelle 3.3. Durch Auszählen der Silben- und Wortgrenzen wurden so zusätzlich alle Positionsmerkmale

bestimmt. Abschließend erfolgte eine Zusammenfassung von phonetischen, Akzent- und Positionsmerkmalen für jede Silbe in einen eindeutigen Merkmalvektor. Die Gesamtheit aller Vektoren, welche auf die beschriebene Weise für das Korpus bestimmt wurden, wurde in einer csv-Datei gespeichert und stand für die weitere Verarbeitung zur Verfügung.

Liegen die optimalen TAM-Parameter jeder Silbe des Korpus vor, können diese gleich mit den passenden Merkmalen verknüpft werden, wodurch die Lernstichprobe erhalten wurde. Diese Lernstichprobe lag dann ebenfalls silbenweise in einer csv-Datei vor.

qta-pred

Das Programm qta-pred stellt den Kern des entwickelten Systems dar und implementiert die verschiedenen maschinellen Lernverfahren mit den geeigneten Trainingsalgorithmen. Vor der Implementierung wurden verschiedene C++ Bibliotheken für Maschinenlernen verglichen, um geeignete auszuwählen. Bei der Umsetzung von Verfahren die auf einer Support-Vektor-Methode basieren, hat sich die LIBSVM Bibliothek (Chang und Lin, 2011) als Standardwerkzeug etabliert und soll dabei auch im Rahmen dieser Arbeit Anwendung finden. Die LIBSVM ist jedoch nach dem C89-Standard implementiert, weshalb die Umsetzung der LIBSVM innerhalb der Dlib-Bibliothek verwendet wurde, die eine komfortablere Schnittstelle als das Original bietet und sich besser in den vorhandenen C++ Code integrieren lässt. Ferner sind im Maschinenlern-Modul der Dlib die Methoden LRR sowie KRR umgesetzt, welche ebenfalls für die Implementierung genutzt wurden. Vorteilhaft bei dieser Implementierung ist die Umsetzung der Methode nach Rifkin und Lippert (2007), wodurch der Regularisierungsparameter während des Trainings gleich mit berechnet wird.

Die Dlib implementiert darüber hinaus ein Modell des MLP, welches jedoch gewissen Einschränkungen unterliegt. So ist es beispielsweise nur möglich, Neuronen mit einer Sigmoid-Aktivierungsfunktion zu verwenden. Da dadurch in der Ausgabeschicht nur Werte im Bereich [0,1] ausgegeben werden können, ist eine zweckmäßige Skalierung und Reskalierung der Parameterwerte nötig. Für das Training ist ein BPA mit Schrittweitensteuerung und Momentum-Term zur Beschleunigung des Gradientenverfahrens implementiert. Zwar existieren Bibliotheken mit flexibleren Modellen, die unterschiedliche Aktivierungsfunktionen, Trainingsalgorithmen und Suchstrategien bereitstellen, jedoch wurden diese aus Gründen von meist unübersichtlicheren Programmierschnittstellen nicht eingesetzt. Als Beispiel ist hier die Bibliothek OpenNN zu nennen. Die einheitliche Verwendung einer Bibliothek macht sich auch im Sinne einer eleganten Gesamtlösung bezahlt, da beispielsweise ein einheitliches Format für die trainierten Modelle verwendet werden kann. Die MLP-Implementierung der Dlib reicht für die in dieser Arbeit betrachteten Untersuchungen zum Vergleich verschiedener Lernverfahren völlig aus.

Das Maschinenlern-Modul der Dlib ist damit bestens geeignet, um die in Abschnitt 2.3 betrachteten Algorithmen zu vergleichen. Alternativen, die ebenfalls eine einheitliche Lösung des Gesamtsystems in C++ ermöglichen würden, wären das Maschinenlern-Modul der Bibliothek OpenCV sowie die portable Maschinenlern-Bibliothek mlpack.

Erstere wurde nicht verwendet, da die Bibliothek als sehr umfangreich mit großen Overhead angesehen wird und ein riesiges Softwarepaket darstellt. Dies ist im Besonderen bei der Integration in andere Projekte, die nicht OpenCV verwenden, problematisch. Die mlpack-Bibliothek bietet hingegen ebenfalls eine elegante Programmierschnittstelle als auch eine effiziente Implementierung. Zum Zeitpunkt der Erstellung dieser Arbeit war jedoch eine Implementierung verschiedener Kernmethoden innerhalb der Bibliothek noch nicht umgesetzt und erst für kommende Releases angekündigt, weshalb mlpack ebenfalls nicht für die Umsetzung verwendet wurde.

In der Konzeption dieser Arbeit war anfangs die Implementierung tiefer neuronaler Netzarchitekturen vorgesehen, deren Umsetzung jedoch von den Zwischenergebnissen und zeitlichen Fortschritt abhängig gemacht wurden. Auch hier bietet die Dlib die Möglichkeit zur Implementierung entsprechender Deep-Learning-Lösungen innerhalb des Maschinenlern-Moduls. Dabei werden verschiedene Eingangs-, Verarbeitungs- und Kostenschichten bereitgestellt, die beliebig miteinander verknüpft werden können. Zusätzlich sind unterschiedliche Trainingsalgorithmen implementiert, die durch Bereitstellung von GPU-Kernel ohne zusätzlichen Implementierungsaufwand auch auf NVIDIA Grafikkarten ausgeführt werden können. Die Möglichkeit zum Training rekurrenter Netzwerke ist dabei nicht gegeben. Alternativen dazu stellen die Bibliotheken TensorFlow und Caffe dar, welche beide eine große Auswahl an Deep-Lerning-Methoden anbieten. Caffe ist dabei wiederum durch eine als unsauber empfundene Programmierschnittstelle gekennzeichnet und TensorFlow stellt leider ein riesiges Softwarepaket dar, welches sich schlecht in bestehende Anwendungen integrieren lässt. Eine ausschließliche Implementierung rekurrenter Netzwerke ist durch die Bibliothek LIBRNN gegeben. Diese Bibliothek enthält jedoch einige Bugs und hat Probleme beim Kompilieren verursacht und wurde deshalb als nicht tauglich bewertet.

Das Tool qta-pred implementiert folglich die vier Lernverfahren LRR, KRR, SVR und MLP für die Vorhersage der TAM-Parameter. Die Daten der Lernstichprobe, gegeben als csv-Datei, werden dabei in Vektoren eingelesen. Das Programm benötigt zusätzlich eine xml-Datei als Eingang, die den benutzten Lernalgorithmus definiert. Diese Algorithmus-Datei bestimmt das zu nutzende Lernverfahren, dessen Hyperparameter sowie einen Dateipfad für das trainierte Modell. Das Tool implementiert dabei drei verschiedene Funktionen. So können durch ein Optimierungsverfahren beruhend auf dem Kreuzvalidierungsfehler, wie in 3.2 beschrieben, die Hyperparameter des jeweiligen Verfahrens berechnet werden. Sind diese bestimmt, kann das Tool zur Vorhersage der TAM-Parameter einer kompletten Lernstichprobe genutzt werden, wobei ebenfalls ein Kreuzvalidierungsverfahren benutzt wird. Schließlich lässt sich aus einer Lernstichprobe auch ein trainiertes Modell im Dlib-spezifischen, binären Dateiformat erzeugen, welches dann innerhalb anderer Projekte einfach eingelesen werden kann und für Vorhersagen nutzbar ist.

qta-eval

Zum Zweck der Auswertung dient das Tool qta-eval. Dieses berechnet relevante Statistiken aus den generierten Daten wie beispielsweise den mittleren RMSE oder Korrelationskoeffizienten, welche zur Bewertung des Gesamtsystems herangezogen werden. Davon abgesehen werden jedoch auch Momente der Parameterverteilungen und andere Häufgkeitsanalysen ermittelt. Zusätzlich werden Grafiken der modellierten Grundfrequenzverläufe mit den jeweiligen PTs erzeugt und der originalen f_0 gegenübergestellt, was eine visuelle Analyse ermöglicht. Ferner werden Histogramme der TAM-Parameter generiert, die Aufschluss über die Verteilung der Daten geben. Für die Implementierung wurde das Statistik-Modul der Dlib-Bibliothek verwendet. Zur Generierung aller Grafiken wurde die Software Gnuplot eingesetzt, welche aus geeigneten plot-Dateien Grafiken erzeugt. qta-eval generiert diese plot-Dateien und startet dann Gnuplot automatisch über einen Systemaufruf.

Zusammenfassung

Ein Informationsflussgraph der vollständigen Implementierung ist in Abbildung 3.13 illustriert. Die Programm TargetOptimizer.praat ist dabei in kompakter Form, ohne innere Struktur abgebildet. Die Grafik veranschaulicht die gesamte Verarbeitung des Korpus sowie die daraus gewonnen Informationen. Zwei voneinander getrennte Verarbeitungsebenen sind hierbei hervorgehoben. Zunächst erfolgt die Grundfrequenzbestimmung mit Praat und einer anschließenden, manuellen Korrektur der Oktavfehler. Danach erst beginnt die vollautomatische Verarbeitung des Korpus durch das bash-Skript qta-learn-f0.sh. Dieses Skript startet die einzelnen Programme kommandozeilenbasiert. Die vom Praat-Skript TargetOptimizer.praat benötigten, nutzerspezifischen Eingabeparameter sind im bash-Skript als Variablen deklariert und können dem Praat-Skript bei einem Kommandozeilenaufruf als Argumente übergeben werden, wodurch keine graphische Nutzeroberfläche gestartet werden muss. Dies ist hinsichtlich einer Berechnung auf dem HPC nötig, da dort keinerlei Grafik-Bibliotheken vorliegen. Auch die benötigten Kommandozeilenargumente der Tools qta-data, qta-pred und qta-eval werden als Variablen im bash-Skript deklariert, wodurch alle Freiheitsgrade des gesamten Systems an einer einzigen Stelle konfiguriert werden. Bei diesen Tools spezifizieren die Kommandozeilenargumente benötigte Ein- und Ausgabedateien bzw. -pfade oder verschiedene Betriebsmodi. Die genaue Spezifikation ist der Hilfe-Option der jeweiligen Anwendung zu entnehmen.

Alle beteiligten Dateien lassen sich in Ein- bzw. Ausgangsdaten gliedern, wobei alle Eingangsdaten durch das Korpus gegeben sind und alle Ausgangsdaten die Ergebnisse der Korpusverarbeitung darstellen. Diese beiden Ebenen sind ebenfalls in der Grafik hervorgehoben.

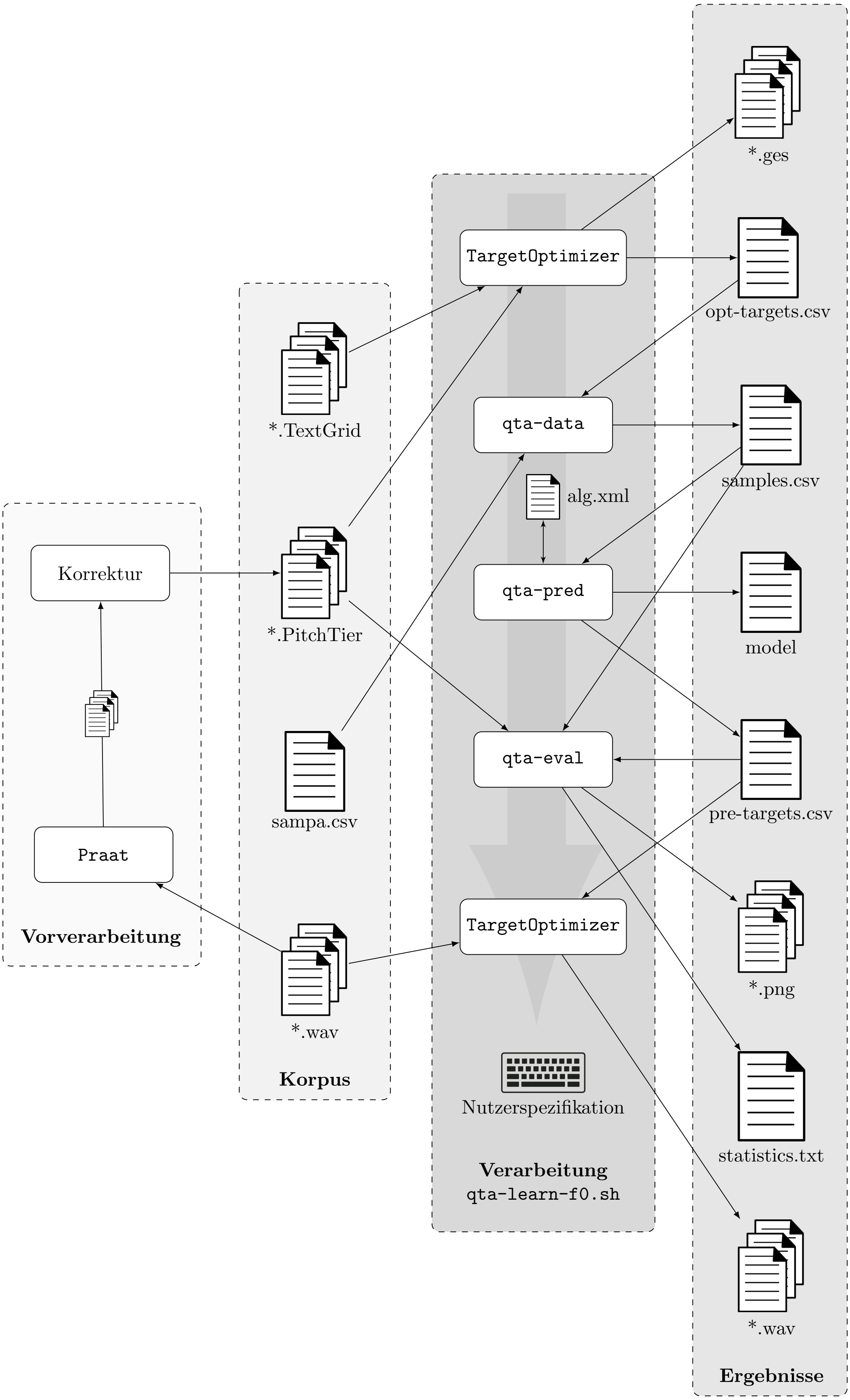

Abbildung 3.13: Informationsflussdiagramm des Gesamtsystems.

Kapitel 4

4 Untersuchungsergebnisse

4.1 Statistische Onset-Schätzung

Bei der Grundfrequenzvorhersage liegen keine Informationen der natürlichen f_0 für die Onset-Schätzung vor. Daher müssen geeignete statistische Werte ermittelt werden, die dann der Onset-Schätzung dienen. Wie in Kapitel 3 besprochen, sollen hierbei die zwei Methoden $\hat{\phi} = \bar{\phi}_{\text{corp}}$ und $\hat{\phi} = \bar{\phi}_{\text{corp}}(s)$ untersucht werden. Um den sprecherspezifischen Mittelwert zu ermitteln, welcher vom verwendeten Korpus abhängig ist, wurde der Wert $f_0(k_{\text{v0}}\Delta t)$ aller natürlichen Grundfrequenzverläufe des Korpus gemittelt und ein Wert von $\bar{\phi}_{\text{corp}} = \bar{f}_0(k_{\text{v0}}\Delta t) = 94{,}06$ st für den Onset bestimmt. Die zugehörige Verteilung aller Werte $f_0(k_{\text{v0}}\Delta t)$ ist in Abbildung 4.1 dargestellt. Ferner wurde eine Abhängigkeit dieser Werte von der Silbenanzahl untersucht, wobei jeweils die Verteilungen der Werte $f_0(k_{\text{v0}}\Delta t; s)$, $s = 1...8$ für alle Äußerungen mit gleicher Silbenanzahl betrachtet wurde. Die Ergebnisse sind ebenfalls in Abbildung 4.1 verdeutlicht. Als Onset-Schätzung würde nun wiederum der jeweilige silbenabhängige Mittelwert der Verteilung dienen, welcher als $\bar{\phi}_{\text{corp}}(s) = \bar{f}_0(k_{\text{v0}}\Delta t; s)$, $s = 1...8$ gegeben ist. Es zeigt sich jedoch, dass keine wesentlichen Unterschiede in den Mittelwerten bestehen und die Wahl eines einzelnen sprecherspezifischen Wertes vernünftig scheint. Auffällig ist jedoch, dass einsilbige Wörter im Mittel einen um 0,5 st höheren Wert $f_0(k_{\text{v0}}\Delta t)$ aufweisen.

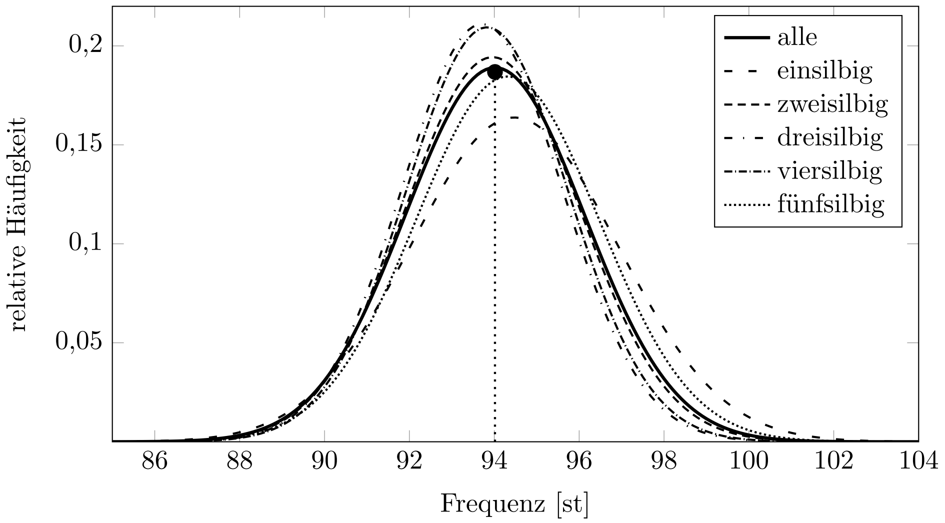

Abbildung 4.1: Verteilungen der Werte $f_0(k_{\text{v0}}\Delta t)$ in Abhängigkeit von der Silbenanzahl.

4.2 Parameterschätzung

Vergleich PENTAtrainer1
($N = 5$; $\tau = 0$ msec; $\lambda = 0$; $\hat{\phi} = f_0(k_{\mathrm{v0}}\Delta t)$)

Da ursprünglich eine modifizierte Variante des PENTAtrainer1 für die Schätzung der TAM-Parameter dienen sollte, der Quellcode jedoch nicht zugänglich war, wurde das Tool TargetOptimizer entwickelt und mit PENTAtrainer1 verglichen. Ein beispielhafter, visueller Vergleich einer geschätzten f_0 ist in Abbildung 4.2 gegeben. Für dieses Beispiel ist deutlich zu erkennen, dass die im TargetOptimizer umgesetzte Variante der Parameterschätzung zu besseren Ergebnissen führt, wenn man vergleicht, wie gut die modellierte Grundfrequenz an die natürliche angenähert wurde. Beim PENTAtrainer1 wird die optimale, modellierte f_0 nicht durch die vom PDA bestimmten f_0-Kontur ermittelt, sondern anhand einer interpolierten und geglätteten Kontur, welche ebenfalls in Abbildung 4.2 dargestellt ist. Dies führt zu dem Problem, dass auch in stimmlosen Bereichen eine Minimierung des RMSE an der interpolierten Kurve angestrebt wird, obwohl der Verlauf an diesen Stellen vernachlässigbar ist. Daraus resultieren unverhältnismäßige Ausmittellungen markanter Bereiche des angestrebten natürlichen f_0-Verlaufs.
Um einen aussagekräftigen Vergleich zwischen den beiden Tools mit deren unterschiedlichen Schätzverfahren zu erhalten, wurden für alle Elemente des Korpus die jeweiligen

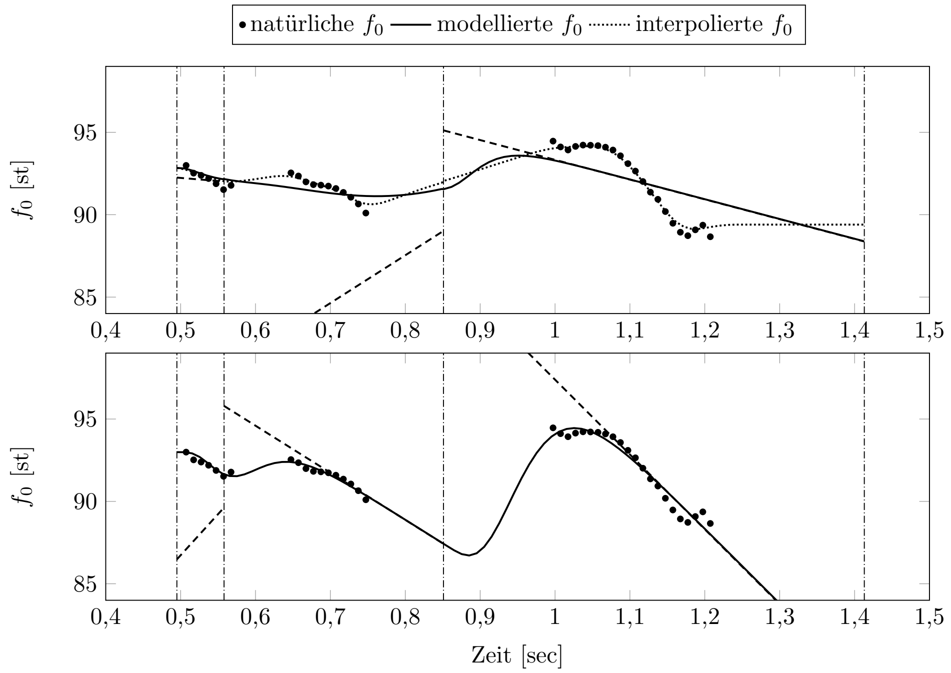

Abbildung 4.2: Vergleich der modellierten f_0-Verläufe mit PENTAtrainer1 (oben, RMSE = 1,109 st) und TargetOptimizer (unten, RMSE = 0,362 st) für die Äußerung *Apostroph* [apo.stʁ̥ˈoːf].

optimalen, modellierten f_0-Konturen ermittelt und entsprechende Ähnlichkeitsmaße zu den natürlichen Verläufen berechnet. Die Vergleichsmaße wurden hierbei ausschließlich auf Basis der vom PDA ermittelten Abtastzeitpunkte bestimmt. Die mittleren Ähnlichkeitsmaße dienen dann einem angemessenen Vergleich. Zusätzlich wurde untersucht, wie sich die Verwendung der nicht manuell korrigierten f_0-Verläufe auf die mittleren Ähnlichkeitsmaße auswirkt. Die Ergebnisse sind in Tabelle 4.1 zusammengefasst. Bei der Verwendung des TargetOptimizers konnte der mittlere RMSE für das Korpus fasst halbiert werden, was die Qualität der modellierten f_0-Verläufe belegt. Auch der mittlere Wert des Korrelationskoeffizienten konnte deutlich gesteigert werden. Ferner lässt sich aus den Zahlen erkennen, dass sich eine fehlende, manuelle Korrektur auf die Schätzmethode des TargetOptimizers stärker auswirkt und diese damit anfälliger für Oktavfehler des PDA ist.

Abschließend sei noch darauf hingewiesen, dass nicht die vom PENTAtrainer1 ausgegebenen Werte von Korrelationskoeffizient und RMSE für den Vergleich genutzt wurden, da diese anhand der interpolierten f_0 ermittelt wurden. Für den modellierten Grundfrequenzverlauf aus Abbildung 4.2 gibt das Tool beispielsweise einen RMSE von 0,46 st aus. Zudem wurde dieser Wert nicht für die gesamte Äußerung berechnet, sondern die RMSE Werte der Silben wurden gemittelt, wofür in diesem Falle die Werte {0,06; 0,38; 0,95} st für die Silben-RMSE Werte bestimmt wurden. Eine Mittelung dieser Werte zur Berechnung des RMSE einer Äußerung führt jedoch zu einer unzulässigen Verzerrung und kann demnach nicht für einen fairen Vergleich verwendet werden. Gleiches gilt auch für die Berechnung des Korrelationskoeffizienten.

Tool	RMSE [st]	ρ
TargetOptimizer(unkorrigiert)	0,582	0,944
TargetOptimizer	0,557	0,946
PENTAtrainer1 (unkorrigiert)	1,042	0,877
PENTAtrainer1	1,028	0,883

Tabelle 4.1: Vergleich der mittleren Ähnlichkeitsmaße zwischen modellierten und natürlichen f_0-Verläufen.

Einfluss Filterordnung

($N \to$ var.; $\tau = 0$ msec; $\lambda = 0$; $\hat{\phi} = f_0(k_{\mathrm{v}0}\Delta t)$)

Die Filterordnung N stellt sich als freier Parameter des Systems dar und sollte deshalb optimal gewählt werden. Ein beispielhafter Vergleich zweier modellierter Grundfrequenzen mit der Filterordnung $N = 3$ bzw. $N = 5$ ist in Abbildung 4.3 veranschaulicht und verdeutlicht, wie sich eine höhere Filterordnung positiv auf das Ergebnis auswirken kann. Im Allgemeinen ist ein System höherer Ordnung deutlich flexibler als ein System niedrigerer Ordnung und damit besser geeignet, um beliebige Kurven anzunähern. Im

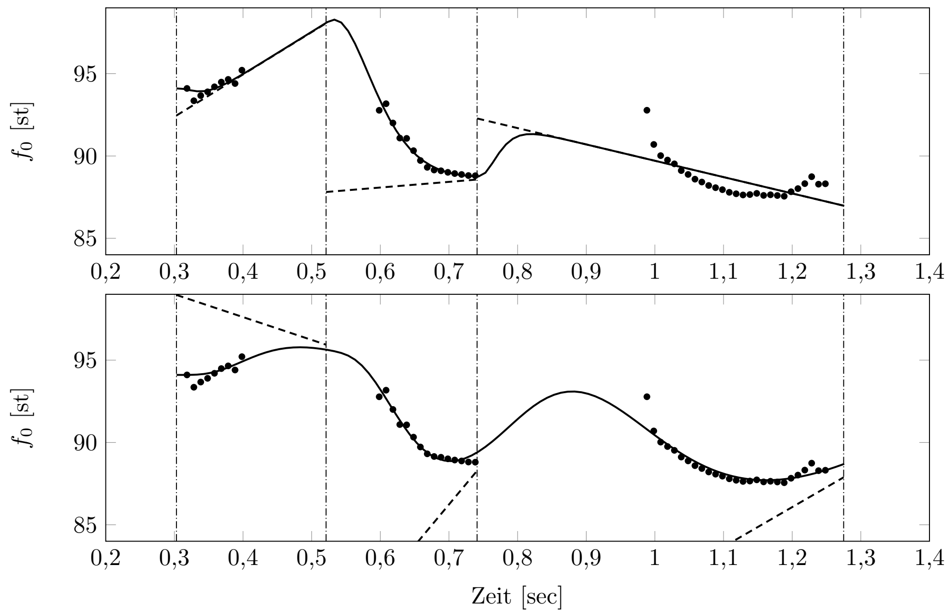

Abbildung 4.3: Vergleich der modellierten f_0-Verläufe mit Filterordnung $N = 3$ (oben, RMSE = 0,521 st) und $N = 5$ (unten, RMSE = 0,381 st) für die Äußerung *abplatzen* [ˈap͡platsn̩].

vorliegenden Beispiel gelingt es so mit einem System dritter Ordnung nicht, den markanten Verlauf in der letzten Silbe abzubilden, was sich natürlich auch am RMSE Wert äußert.

Eine optimale Filterordnung wurde dadurch bestimmt, dass die mittleren Ähnlichkeitsmaße für das gesamte Korpus bei unterschiedlichen Filterordnungen berechnet wurden. Die Ergebnisse sind in Abbildung 4.4 visualisiert. Besonders beim RMSE zeigt sich ein minimaler Fehler bei einer Ordnung von $N = 5$. Auch der Korrelationskoeffizient ist für

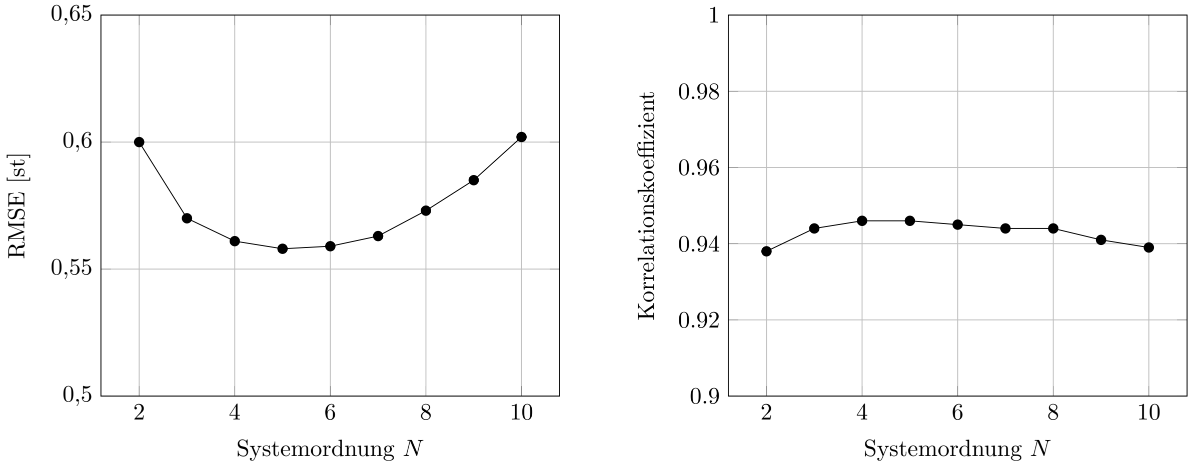

Abbildung 4.4: Mittlere Ähnlichkeitsmaße in Abhängigkeit der Systemordnung.

ein solches System fünfter Ordnung am höchsten. Bei Filterordnungen größer als fünf verschlechtern sich die Ähnlichkeitsmaße und der Rechenaufwand erhöht sich deutlich. Im ersten Moment wäre es nicht zu erwarten, dass eine Erhöhung der Filterordnung zu schlechteren Ergebnissen führt, da das System nicht an Flexibilität verliert, sondern im Gegenteil über mehr Freiheitsgrade verfügt. Mathematisch gesehen liegt scheinbar genau darin das Problem, da sich diese erhöhte Anzahl von Freiheitsgraden negativ auf die Optimierung auswirken. Die Zielfunktion des Optimierungsproblems 3.7 enthält dadurch noch mehr lokale Extrema, was das Auffinden eines globalen Minimums erschwert. Demnach stellt ein Filter fünfter Ordnung einen guten Kompromiss zwischen Komplexität und Berechenbarkeit dar.

Einfluss Silbengrenzenverschiebung

$(N = 5;\ \tau \to \text{var.};\ \lambda = 0;\ \hat{\phi} = f_0(k_{\text{v}0}\Delta t))$

Auch bei der Untersuchung der Silbengrenzenverschiebung τ wurde als erstes wieder ein visueller Vergleich angestellt, um den Einfluss einzuschätzen. Ein solches Beispiel ist in Abbildung 4.5 dargestellt. Man erkennt, dass eine Verschiebung der Silbengrenzen nach vorn für diesen Fall einen völlig anderen Grundfrequenzverlauf erzeugt, wobei auch die gefunden PTs sehr unterschiedlich sind. Dennoch wurde bei beiden Varianten die originale f_0 ähnlich gut angenähert, die schließlich nur in den stimmhaften Bereichen definiert ist. Auch die jeweiligen RMSE Werte unterschieden sich nicht wesentlich.

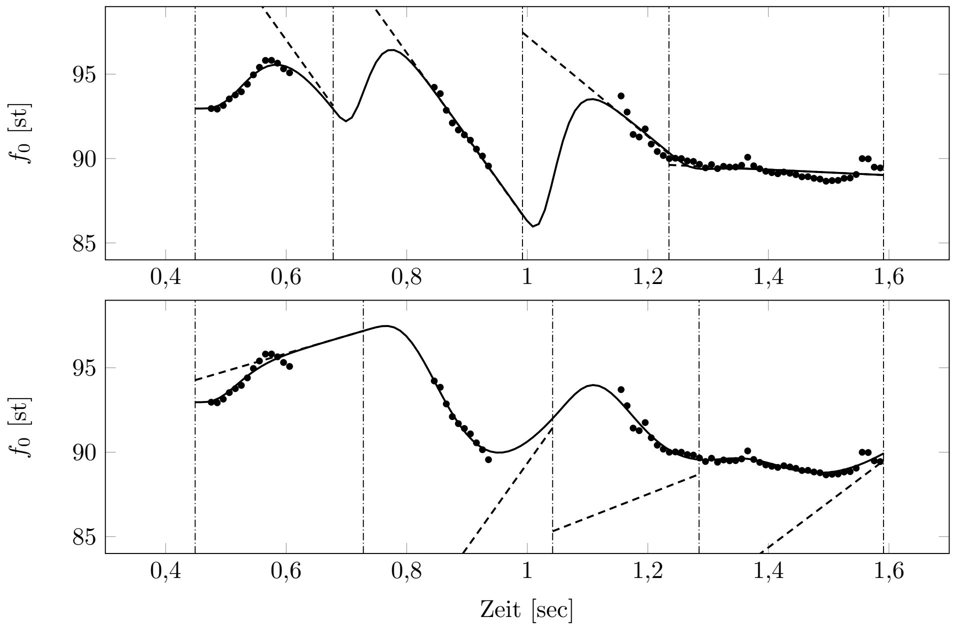

Abbildung 4.5: Vergleich der modellierten f_0-Verläufe mit Silbengrenzenverschiebung $\tau = 50$ msec (oben, RMSE = 0,352 st) und $\tau = 0$ msec (unten, RMSE = 0,282 st) für die Äußerung *Auspuffflamme* [ˈaɔ̯spʊf͜flamə].

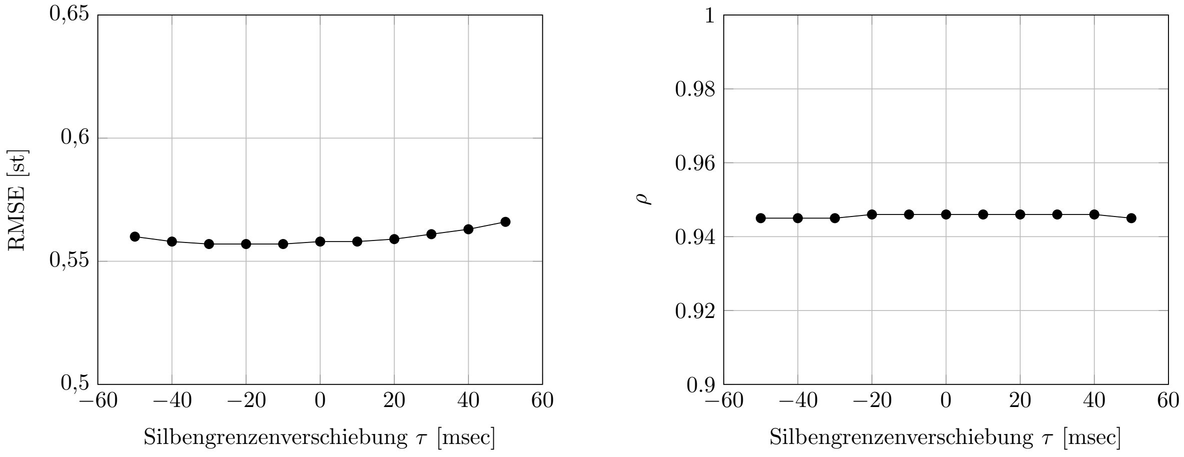

Abbildung 4.6: Mittlere Ähnlichkeitsmaße in Abhängigkeit der Silbengrenzenverschiebung.

Um hierbei den Einfluss wiederum quantitativ festzustellen, wurden die mittleren Ähnlichkeitsmaße des Korpus für verschiedene Werte der Silbengrenzenverschiebung untersucht. Die zugehörigen Ergebnisse sind in Abbildung 4.6 dargestellt. Aus diesen Graphen wird ersichtlich, dass es keinen relevanten Einfluss der Silbengrenzenverschiebung auf die Qualität der modellierten Grundfrequenzen gibt.

Einfluss Onset-Schätzung

($N = 5$; $\tau = 0$ msec; $\lambda = 0$; $\hat{\phi} \rightarrow$ var.)

Da sich der Einfluss des Onset-Schätzverfahrens der Parameterschätzung auf das Training eines Lernalgorithmus auswirken könnte, wurde dieser ebenfalls genauer untersucht. Für die Parameterschätzung liegen die Information der natürlichen f_0 bereit und können für die Onset-Schätzung verwendet werden, wodurch alle in Abschnitt 3.2 diskutierten Verfahren eingesetzt werden könnten. Die Methode $\hat{\phi} = \phi*$ würde die besten Ergebnisse erwarten lassen, wurde jedoch nicht im Rahmen dieser Arbeit implementiert und untersucht, da die Idee zu spät entwickelt wurde. Das gleiche gilt für die Methode $\hat{\phi} = \phi_{\text{ext}}$. Die statistische Methode $\hat{\phi} = \bar{\phi}_{\text{corp}}(s)$ bringt nach den Untersuchungen aus Abschnitt 4.1 keinerlei Vorteil. Aus diesen Gründen wurden nur die beiden Methoden $\hat{\phi} = f_0(k_{\text{v0}}\Delta t)$

Methode	RMSE [st]	ρ
$\hat{\phi} = f_0(k_{\text{v0}}\Delta t)$	0,557	0,945
$\hat{\phi} = \bar{\phi}_{\text{corp}}$	0,672	0,918

Tabelle 4.2: Vergleich der mittleren Ähnlichkeitsmaße zwischen den verschiedenen Onset-Schätzverfahren.

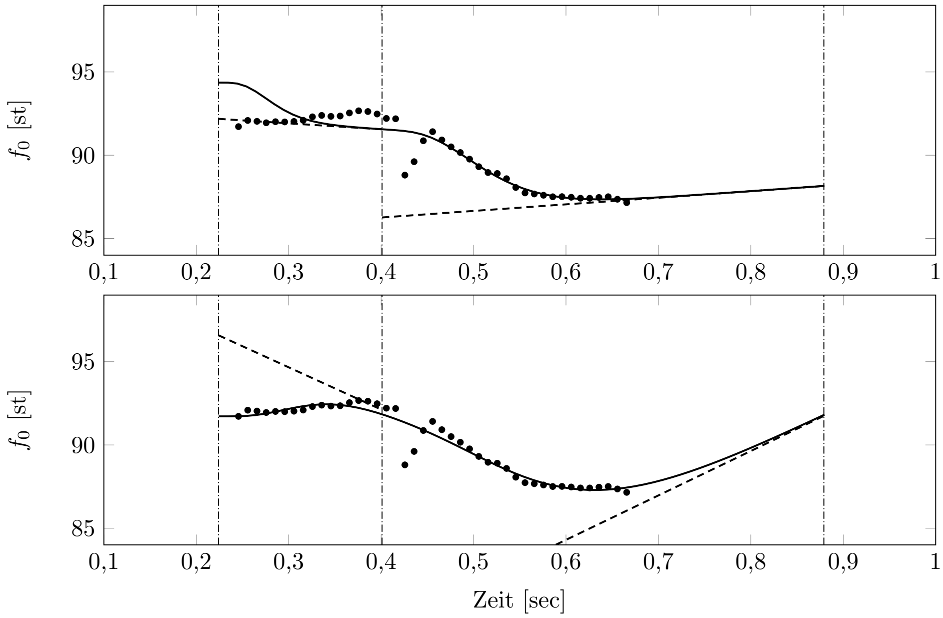

Abbildung 4.7: Vergleich der modellierten f_0-Verläufe mit Korpus-Mittelwert $\bar{\phi}_{corp}$ (oben, RMSE = 0,881 st) und ersten Abtastwert der natürlichen Grundfrequenz $f_0(k_{\mathrm{v0}}\Delta t)$ (unten, RMSE = 0,529 st) für die Äußerung *Abend* [ˈaːbmt].

und $\hat{\phi} = \bar{\phi}_{\mathrm{corp}}$ miteinander verglichen, wobei erstere auf der natürlichen f_0 basiert und letztere rein statistisch ist.
In Abbildung 4.7 ist wiederum ein illustrierendes Beispiel gegeben. Darin erkennt man, dass hauptsächlich für Äußerungen mit stimmhaften Erstlaut die Verwendung des ersten Abtastwertes der natürlichen f_0 eine bessere Kurvenanpassung ermöglicht. Diese Beobachtung spiegelt sich auch in den mittleren Ähnlichkeitsmaßen wieder, welche für das Korpus bestimmt wurden und in Tabelle 4.2 gegeben sind.

Einfluss Regularisierungsparameter
$(N = 5;\ \tau = 0$ msec; $\lambda \rightarrow$ var.; $\hat{\phi} = f_0(k_{\mathrm{v0}}\Delta t))$

Aus den bisher betrachteten, beispielhaft gezeigten f_0-Verläufe nach dem TAM wird ersichtlich, dass die gefundenen, optimalen PTs nicht unbedingt physiologisch so interpretiert werden können, wie man es erwartet. Eine möglichst genaue Anpassung an die natürliche f_0 wird oft nur auf Kosten „unnatürlicher" Targets mit betragsmäßig großem Anstieg oder kleinen Annäherungsraten erreicht, die nicht im Sinne des Modells sind. Deshalb liegt die Überlegung nahe, bei der Optimierung natürlichere Targets zu bevorzugen und dafür eine suboptimale Kurvenannäherung in Kauf zu nehmen. Dieser Kompromiss wird durch das Einbringen der Regularisierung erreicht, wie in Abschnitt 3.1 beschrieben wurde und in diesem Abschnitt analysiert.

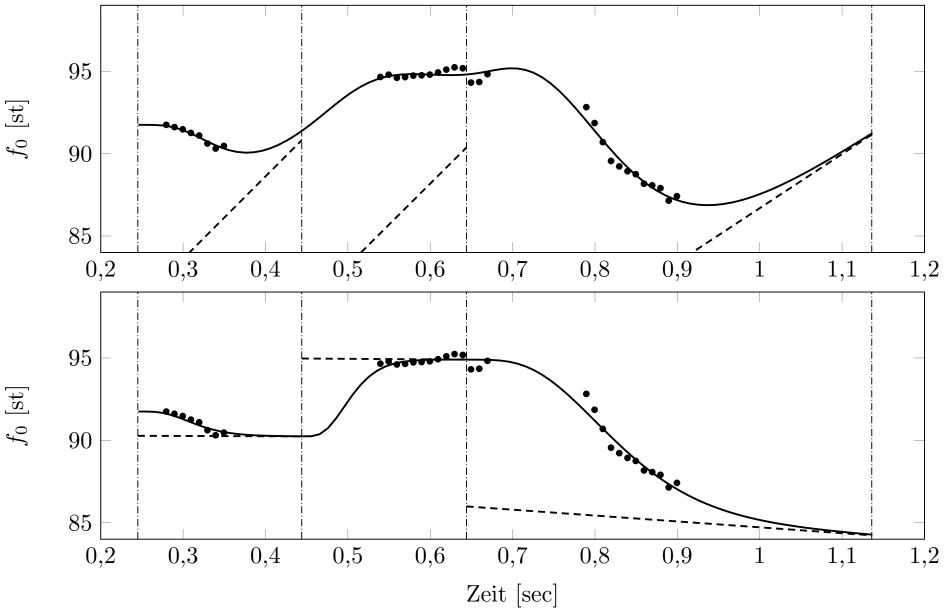

Abbildung 4.8: Vergleich der modellierten f_0-Verläufe mit Regularisierungsparameter $\lambda = 0$ (oben, RMSE = 0,303 st) und $\lambda = 0,75$ (unten, RMSE = 0,348 st) für die Äußerung *Ästhetik* [ɛstˈeːtik].

Dazu ist wiederum ein Beispiel gegeben, welches in Abbildung 4.8 dargestellt ist. Man erkennt, dass durch eine minimale Verschlechterung der Fehlermaße, hervorgerufen durch den Regularisierungsterm, wesentlich plausiblere PTs gefunden werden.
Eine Analyse der Verteilungen der gefundenen Parameter erklärt diesen Effekt. Abbildung 4.9 zeigt die Verteilungen der Parameter Anstieg, Verschiebung und Annäherungsrate in Abhängigkeit des Regularisierungsparameters. Ohne Regularisierung werden für alle Parameter primär die Grenzen des Suchraums als optimal identifiziert, was auch im Beispiel aus Abbildung 4.8 in der zweiten Silbe zu beobachten ist, wo ein PT mit steilen Anstieg und kleiner Annäherungsrate gefunden wurde, was nicht dem Sinne des Modells entspricht. Die nicht regularisierten Verteilungen weisen demnach auch eine dementsprechend hohe Varianz auf. Die Erhöhung des Regularisierungsparameters bewirkt eine zunehmende Bestrafung dieser extremen Werte für die TAM-Parameter, wodurch eher Werte gefunden werden, die um einen bestimmten Mittelwert verteilt sind. Die Masse der Verteilung wird dadurch hin zu einem festgelegten Wert gedrückt. Für die Parameter Anstieg und Verschiebung bedeutet dies die Annäherung an eine Normalverteilung, wodurch ebenfalls die Varianz deutlich reduziert wird. Somit werden Anstiege um den Wert 0 $\frac{\text{st}}{\text{sec}}$ und Verschiebungen mit Werten um 95 st, welches die mittlere f_0 des Suchbereichs, bevorzugt. Beim Parameter der Annäherungsrate bzw. inversen Zeitkonstante wird der Wert gegen 80 $\frac{1}{\text{sec}}$ gedrückt, der physiologisch sehr plausibel ist, wie bereits in Kapitel 3 erörtert. Sehr kleine Werte treten eher selten auf und werden dadurch bei der Regularisierung stärker bestraft. Zusätzlich zur Wahl dieser Mittelwerte, kann durch die Designmatrix W ein Wichtungsfaktor eingestellt werden, der die Stärke der Regularisierung für die jeweiligen Parameter einstellt. Andernfalls müssten drei

separate Regularisierungsterme eingeführt werden, was die Freiheitsgrade der Optimierung unnötig erhöht. Die Einträge der Diagonalmatrix W wurden dabei empirisch so

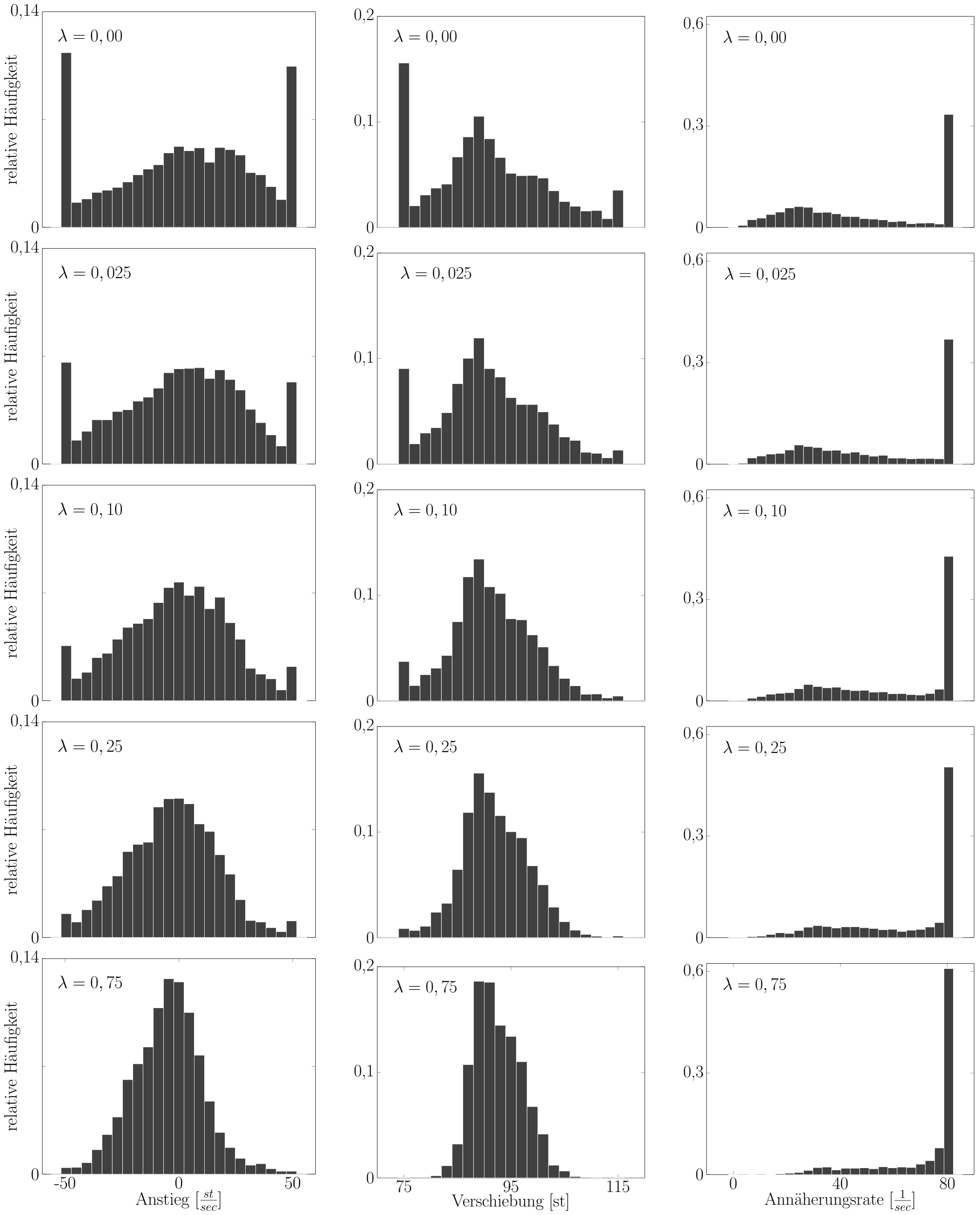

Abbildung 4.9: Einfluss des Regularisierungsparameters auf die Parameterverteilung der geschätzten TAM-Parameter.

gewählt, dass die Verteilungen ungefähr im gleichen Maße geändert werden. Es hat sich herausgestellt, dass dabei die Parameter Anstieg und Annäherungsrate stärker bestraft werden müssen als der Parameter Verschiebung. Alle freien Parameter wurden bereits in Tabelle 3.4 zusammenfassend dargestellt.
Ein Verlauf der mittleren Ähnlichkeitsmaße in Abhängigkeit vom Regularisierungsparameter λ ist in Abbildung 4.12 gegeben und wird später noch im Zusammenhang mit der Vorhersage diskutiert.

Modellkomplexität

($N = 5$; $\tau = 0$ msec; $\lambda = 0$; $\hat{\phi} = f_0(k_{v0}\Delta t)$)

Im Zusammenhang mit Abbildung 4.9 wurden bereits die typischen Verteilungen der als optimal identifizierten TAM-Parameter im Sinne der mathematischen Optimierung diskutiert. Im ursprünglichen Modell ist keine Regularisierung vorgesehen und wie gezeigt wurde, wird ohne Regularisierung auch das optimale Ergebnisse bzgl. der angenäherten Kurve erzielt. Betrachtet man die Werte der unregularisierten Verteilungen aus Abbildung 4.9 nochmals genauer in einem Streudiagramm in gegenseitiger Abhängigkeit, ergeben sich weitere Aufschlüsse über die geschätzten TAM-Parameter. Zwei solcher Diagramme sind Abbildung 4.10 dargestellt. Aus diesen wird zum einen ersichtlich, dass fast ausschließlich PTs gefunden werden, die bei geringer Verschiebung einen starken positiven Anstieg aufweisen, bei mittlerer Verschiebung vornehmlich Anstiege um Null und bei großen Verschiebungen insbesondere starke negative Anstiege. Stellt man sich dies für eine f_0-Kontur vor dem inneren Auge vor, ergibt dies auch Sinn. So wären beispielsweise Targets mit großer Verschiebung und großen positiven Anstieg sehr unnatürlich. Dennoch deutet dieser Zusammenhang auf eine gewisse Abhängigkeit der beiden Parameterschätzungen und eine eventuelle, damit verbundene Überbestimmtheit des Modells hin.

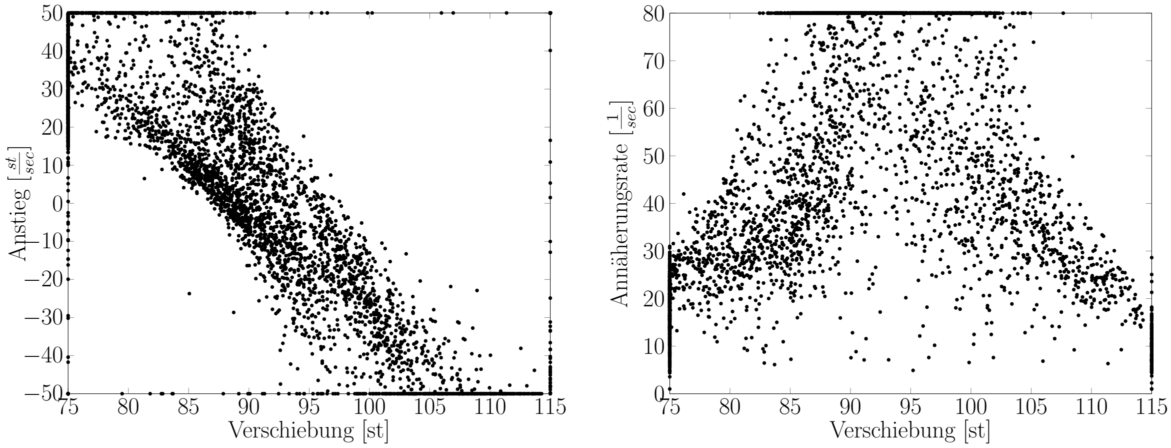

Abbildung 4.10: Streudiagramme der geschätzten TAM-Parameter des Korpus für $\lambda = 0$.

Diese Vermutung wird ebenfalls dadurch gestützt, indem man die Annäherungsrate in Abhängigkeit der Verschiebung darstellt. So wird deutlich, dass über einem großen Bereich der Verschiebung [85;105 st] sehr oft der maximale zugelassene Wert der Annäherungsrate von 80 $\frac{1}{sec}$ als optimal identifiziert wird. Jedoch für sehr kleine bzw. sehr große Werte der Verschiebung werden geringere Werte der Annäherungsrate als optimal empfunden. Gerade PTs, die diesen Charakter aufweisen, werden als „unnatürlich“eingestuft, da eine zu geringe Annäherungsrate bei starken Verschiebungen dazu führen kann, dass sich die f_0-Kontur effektiv gar nicht an das Target annähert, sondern weit davon entfernt bleibt. Ein solcher Fall ist beispielsweise in der zweiten Silbe der unregularisierten Variante aus Abbildung 4.8 zu sehen.

Die Beobachtungen legen nahe, dass das TAM eventuell überbestimmt ist. Dies äußert sich darin, wie in Abschnitt 2.1 diskutiert, dass verschiedene Parameterkombinationen zu den selben f_0-Verläufen führen können. Wie bereits beschrieben, definiert die Anzahl der Freiheitsgrade die Komplexität des Modells. Die in Abschnitt 2.3 besprochene Regularisierung des Regressionsmodells, beschrieben durch den Parameter C, dient einer Reduktion der Modellkomplexität, um den Gesamtfehler zu minimieren. Auch die für die Bestimmung der optimalen PTs eingeführte Regularisierung, charakterisiert durch λ, schränkt die Freiheitsgrade des Modells ein und dient damit einer Reduktion der Modellkomplexität und wirkt damit der Überbestimmtheit des Modells entgegen.

Die Regularisierung ist hierbei jedoch eine rein empirische Methode zur Reduktion der Modellkomplexität. Auf Basis der aus Abbildung 4.10 ableitbaren Regeln stellt sich natürlich die Frage, ob nicht auch eventuell eine analytische Reduktion der Freiheitsgrade, die sich in der Definition des Modells äußert, finden lässt. Geeignete Beschränkungen des Modells müssen dabei in den physiologischen Mechanismen der Grundfrequenzproduktion gesucht und angemessen mathematisch umgesetzt werden. Die aus Abbildung 4.10 abgeleiteten Regeln bietet einen Hinweis darauf.

Grenzen des Modells
($N = 5$; $\tau = 0$ msec; $\lambda = 0$; $\hat{\phi} = f_0(k_{\mathrm{v}0}\Delta t)$)

Die bisher gezeigten Beispiele wurden jeweils immer benutzt, um den positiven Einfluss einer Größe auf die Parameterschätzung zu verdeutlichen. Deshalb sind in Abbildung 4.11 zwei Beispiele gegeben, bei denen die umgesetzte Parameterschätzung nicht zu den gewünschten Ergebnissen führt. So wird zum einen deutlich, dass besonders bei einsilbigen Äußerungen oft keine zufriedenstellende Kurvenannäherung im Rahmen des TAM gefunden wird. Dies liegt meist daran, dass sich der Initialzustand des Filters für die erste Silben einer jeden Äußerung im Nullzustand befindet und damit die Möglichkeiten der Kurvenannäherung einschränkt. Bei mehrsilbigen Äußerungen fällt eine schlechtere Schätzung der ersten Silbe weniger ins Gewicht. Um diesem Problem entgegenzuwirken, ließe sich zusätzlich zum Onset der gesamte Initialzustand des Filters schätzen. Dieser könnte dabei mit in das vorhandene Optimierungsproblem eingebaut werden und einfach durch zusätzliche Optimierungsvariablen dargestellt werden. Damit würde auch die Onsetschätzung verbessert werden, da diese Teil des Filterinitialzustands ist und damit ebenfalls optimal gewählt werden würde. Für die Schätzung würde so auch die komplette natürliche f_0 verwendet werden, wodurch alle verfügbaren Informationen in die Onset-Schätzung einfließen, was in jedem Fall besser ist als die bloße Verwendung des ersten Abtastwertes. Anstatt $3S$ hätte die Optimierung aus Gleichung 3.7 dann $3S + N$ Optimierungsvariablen, wobei N der Filterordnung entspricht. Dieser Ansatz würde wahrscheinlich die Parameterschätzung weiter verbessern, hätte jedoch nur sehr wenig oder keinen Einfluss auf die Grundfrequenzvorhersage, da bei dieser sowieso nur ein sprecherspezifischer Wert für die Onset-Schätzung verwendet werden kann. Aus diesem Grund wurde die optimale Schätzung des Filterinitialzustands nicht mit in die Optimierungsaufgabe der TAM-Parameterschätzung eingebaut.

Neben der Modellierung einsilbiger Äußerungen versagt das Modell oft bei komplett stimmhaften Äußerungen, wo die Grundfrequenz mehrere Wendepunkte pro Silbe aufweist. Da das Modell solche Verläufe nicht abbilden kann, wird meist eine Konstante f_0-Kontur als optimal identifiziert, da diese zwar die Kosten minimiert und damit den RMSE, aber nicht die markanten Stellen der natürlichen Kontur trifft. Dieses Problem lässt sich nicht ohne Weiteres umgehen, sondern stellt eine intrinsische Grenze des Modells dar. Prinzipiell lässt sich beobachten, dass sobald die natürliche Grundfrequenz eine Äußerung mehr als $2S$ Wendepunkte aufweist, eine adäquate Modellierung mittels TAM nicht möglich ist.

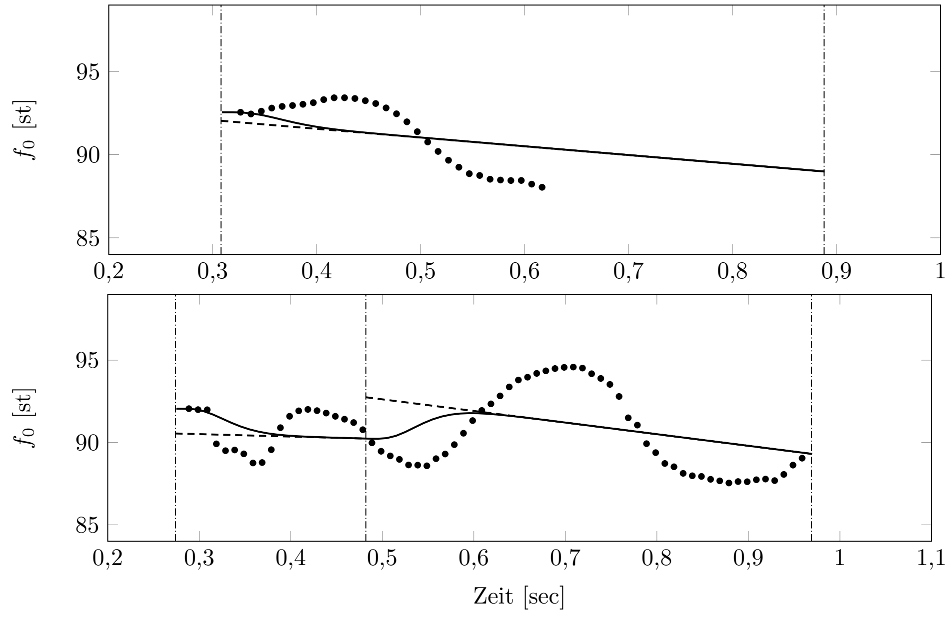

Abbildung 4.11: Modellierte f_0-Verläufe für die Äußerungen *Barsch* [baːʶʃ] (oben, RMSE=1,560 st) und *sowohl* [zoːv ˈoːl] (unten, RMSE=1,898 st).

4.3 Modellauswahl

Bestimmung der Hyperparameter

($N = 5$; $\tau = 0$ msec; $\lambda \to$ var.; $\hat{\phi} = f_0(k_{\mathrm{v}0}\Delta t)$)

Die Hyperparameter der eingesetzten Lernmethode hängen von der Lernstichprobe ab. Die optimalen Hyperparameter lassen sich durch Einsatz der Kreuzvalidierungsmethode und Minimierung des Vorhersagefehlers, wie in Abschnitt 3.2 diskutiert, ermitteln. Da die Hyperparameter meist vorher nicht sinnvoll abgeschätzt werden können, wird ein

Modell	Parameter	Min	Max	Mittelwert	Median	(Min+Max)/2
Anstieg m	C	0,150	4,298	1,261	0,447	2,224
	γ	0,088	0,484	0,287	0,339	0,286
	ϵ	0,011	0,216	0,069	0,024	0,114
	L_1	1	88	15	1	45
	L_1	1	10	2	2	6
	α	0,002	2,380	0,505	0,425	1,191
	β	10^{-5}	0,735	0,284	0,388	0,367
Verschiebung b	C	0,059	100,0	33,67	0,257	50,03
	γ	0,020	0,527	0,222	0,269	0,274
	ϵ	0,013	0,121	0,048	0,031	0,067
	L_1	1	38	15	12	20
	L_1	1	3	1	1	2
	α	0,283	2,983	0,970	0,464	1,633
	β	0,002	0,736	0,326	0,455	0,369
Annäherungsrate λ	C	0,135	0,837	0,340	0,248	0,486
	γ	0,210	3,943	0,601	0,361	2,077
	ϵ	0,006	0,186	0,091	0,095	0,096
	L_1	1	100	9	1	51
	L_1	1	18	6	4	10
	α	0,014	16,80	3,147	0,496	8,407
	β	10^{-5}	0,492	0,199	0,077	0,246
alle	$\gamma_{(\text{Median-Trick})}$	0,154				

Tabelle 4.3: Statistik der optimalen Hyperparameter der 15 Messungen $\lambda = \{0,0; 0,005; 0,01; 0,05; 0,1; 0,5; 1; 5; 10; 50; 100; 500; 1000; 5000; 10000\}$.

Suchraum über mehrere Zehnerpotenzen veranschlagt, wie es auch in Hsu et al. (2016) beschrieben ist. Ziel ist es, eine gewisse Größenordnung für die Hyperparameter festzulegen. Eine Variation der Parameter innerhalb dieser Größenordnung ändert zumeist das Vorhersageergebnis kaum.

Durch die Regularisierung wurde jedoch ein Freiheitsgrad eingeführt, der die Lernstichprobe beeinflusst, wodurch auch die Hyperparameter vom Regularisierungsparameter λ abhängen. Die Untersuchungen haben gezeigt, dass keine explizite Abhängigkeit der Hyperparameter von λ vorliegt. Vielmehr wurden bei der Modellauswahl unabhängig von λ die Hyperparameter in gewissen Intervallen als geeignet identifiziert. Die Statistiken dieser Untersuchung sind in Tabelle 3.5 dargestellt. Für alle weiteren Untersuchungen wurde jeweils der Mittelwert eines jeden Parameters aus Tabelle 4.3 benutzt.

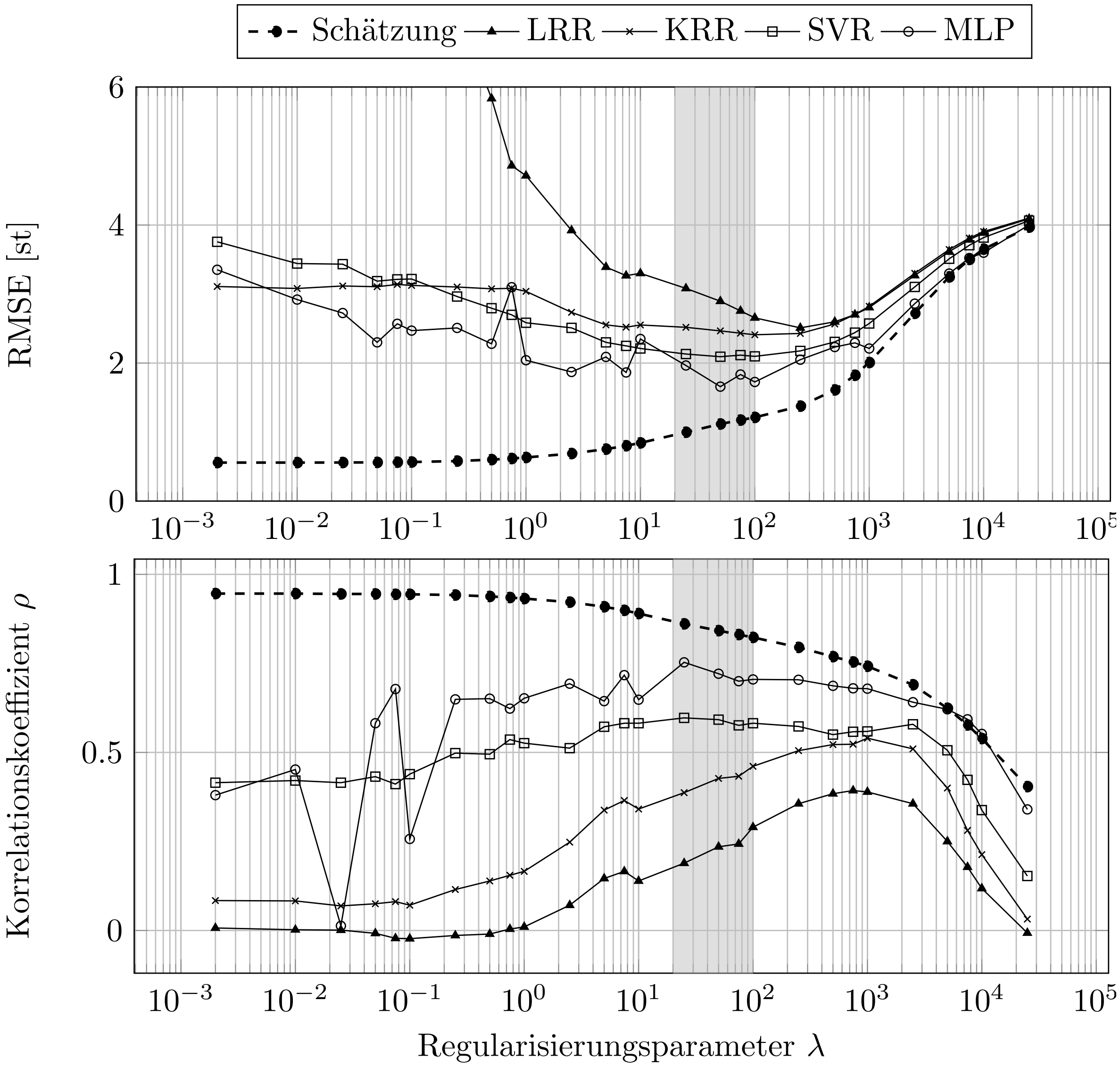

Abbildung 4.12: Abhängigkeit der mittleren Ähnlichkeitsmaße von der Regularisierung.

Bestimmung des Regularisierungsparameters
($N = 5$; $\tau = 0$ msec; $\lambda \rightarrow$ var.; $\hat{\phi} = f_0(k_{\mathrm{v0}}\Delta t)$)

Nachdem für die Lernstichprobe passende Hyperparameter festgelegt wurden, kann eine Bestimmung des optimalen Wertes für die Regularisierung erfolgen, wobei der Parameter so gewählt werden sollte, dass er die Ähnlichkeitsmaße für die vorhergesagten Grundfrequenzverläufe, wiederum gemittelt über das Korpus, minimiert. Entsprechende Untersuchungsergebnisse sind in Abbildung 4.12 gezeigt. Da keinerlei Informationen über den Parameter vorliegen, wurde wiederum eine Suche über mehrere Dekaden veranschlagt. Dabei wurden die mittleren Ähnlichkeitsmaße der für das Training verwendeten, optimal geschätzten f_0-Verläufe sowie der vorhergesagten f_0-Verläufe aller vier untersuchten Regressionsmethoden ermittelt. Aus den Graphen ist deutlich zu erkennen, dass eine stärkere Regularisierung zu einer Verringerung der mittleren Ähnlichkeitsmaße der Vorhersage führt. Durch die Regularisierung werden zwar nicht mehr die bestmöglichen modellierten f_0-Konturen für das Training verwendet, aber dennoch führen diese zu besseren Vorhersageergebnissen. Dies liegt wohl vor allem an der Reduktion der Varianz der Parameterverteilungen der Trainingsdaten, die durch die Regularisierung erreicht wird. Der mit grau markierte Bereich in den Diagrammen wurde als optimal für die Wahl des Regularisierungsparameters identifiziert, da dieser zu den besten Ähnlichkeitsmaßen führt.

Bestimmung der Onset-Schätzung
($N = 5$; $\tau = 0$ msec; $\lambda = 75$; $\hat{\phi} \rightarrow$ var.)

Für die Vorhersage der Grundfrequenz wurde ein sprecherspezifischer Wert für den Onset ermittelt, wie in Abschnitt 4.1 besprochen. Es wurde ebenfalls gezeigt, dass die Verwendung dieses sprecherspezifischen Wertes nicht von Vorteil für die Schätzung der optimalen TAM-Parameter ist. Daraus lässt sich jedoch nicht schlussfolgern, dass die so ermittelten optimalen Parameterwerte auch nicht vorteilhaft für das Training eines Lernverfahrens wären. Aus diesem Grund wurde untersucht, ob sich die Verwendung des sprecherspezifischen Onset Wertes bei der Parameterschätzung positiv auf die Vorhersageergebnisse auswirkt, da bei der Vorhersage ebenfalls diese Art der Onset-Schätzung verwendet wird. Tabelle 4.4 belegt, dass durch die Verwendung des sprecherspezifischen

Methode	RMSE [st]		ρ	
	Training	Vorhersage	Training	Vorhersage
$\hat{\phi} = f_0(k_{\mathrm{v0}}\Delta t)$	1,118	1,684	0,841	0,735
$\hat{\phi} = \bar{\phi}_{\mathrm{corp}}$	1,178	2,256	0,810	0,709

Tabelle 4.4: Vergleich der mittleren Ähnlichkeitsmaße zwischen den verschiedenen Onset-Schätzverfahren.

Onset Wertes für die Bestimmung der Trainingsdaten die mittleren Ähnlichkeitsmaße sowohl für die Trainingsdaten selbst, als auch für die Vorhersage schlechter werden und demnach nicht anzuwenden ist.

4.4 Grundfrequenzvorhersage

Mit den ermittelten Werten von Hyperparameter und Regularisierungsparameter lassen sich die bestmöglichen Vorhersageergebnisse nach der vorgeschlagenen Methode erzielen. Zwar wurden auch die Silbendauern als TAM-Parameter mit vorhergesagt, aber dennoch die annotierten Silbendauern des Korpus zur Erzeugung der vorhergesagten f_0-Verläufe verwendet. Die Laut- und Silbendauervorhersage stellt ein separates Problem dar und stand nicht im Fokus der hier getätigten Untersuchungen. Die Silbendauervorhersage wurde zwar mit implementiert, jedoch nicht weiter untersucht, da dies nicht Teil der Aufgabenstellung war und der zeitliche Rahmen der Arbeit nicht zuließ.
Ein Beispiel vorhergesagter Grundfrequenzverläufe ist in Abbildung 4.13 dargestellt. Wie bereits Abbildung 4.12 nahelegt, werden die besten Vorhersageergebnisse durch die Regressionsmethoden SVR und MLP erzielt. Besonders im Verlauf der Kurve für das MLP sind ersichtliche Schwankungen zu erkennen, die auf eine nicht ausreichend große Lernstichprobe hindeuten. Verschiedene Zufallsinitialisierungen sowie verschiedene Teilmengen der Lernstichprobe bei der Kreuzvalidierung führen zu teils sehr unterschiedlichen Vorhersageergebnissen. Setzt man eine ausreichend große Lernstichprobe voraus,

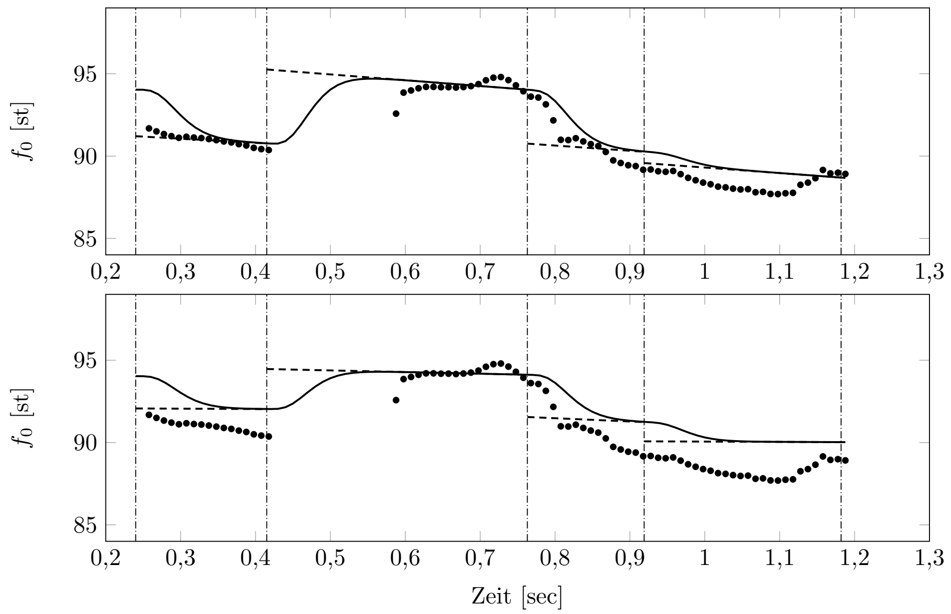

Abbildung 4.13: Vergleich der vorhergesagten f_0-Verläufe von MLP (oben, RMSE=0,958 st) und SVR (unten, RMSE=1,573 st) für die Äußerung *Arterie* [aʶtˈeːʁiə].

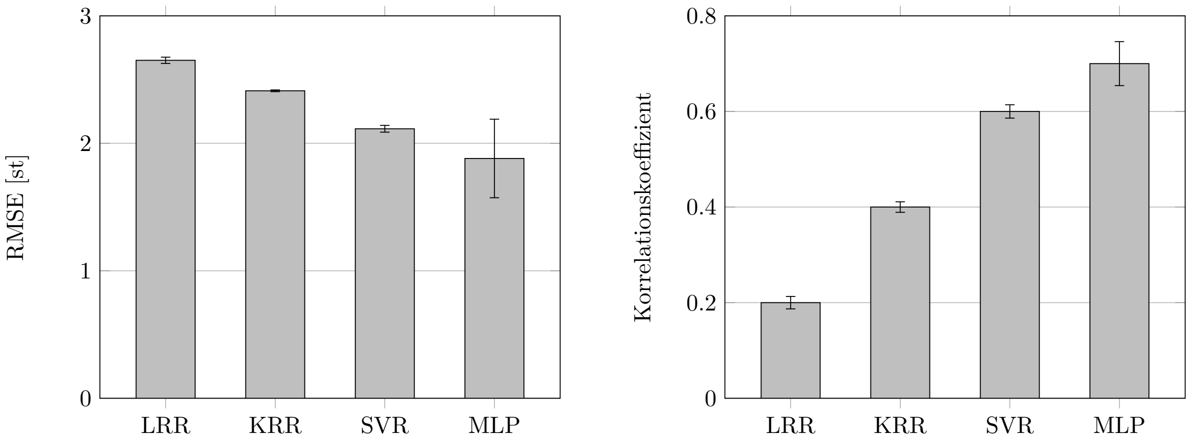

Abbildung 4.14: Mittlere Ähnlichkeitsmaße der vorhergesagten Grundfrequenzverläufe gemittelt über 10 Messungen.

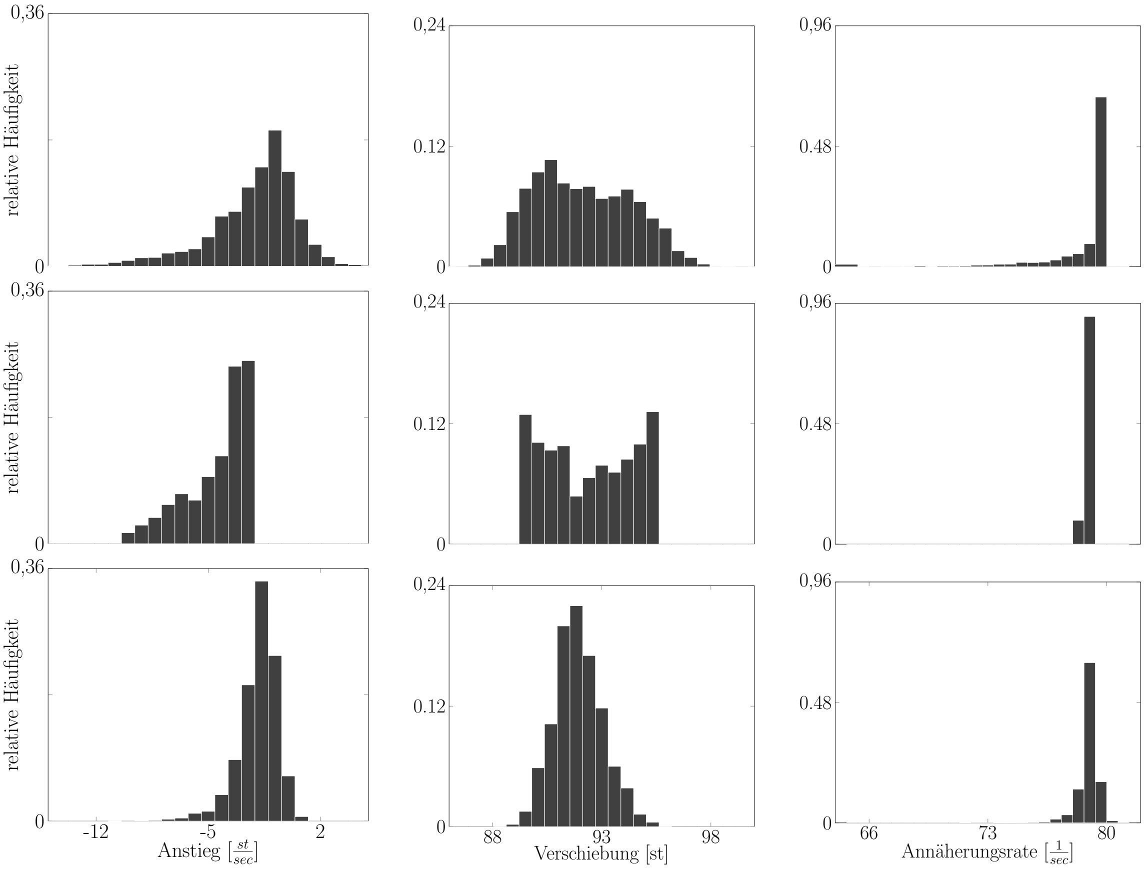

Abbildung 4.15: Vergleich der Parameterverteilungen von Trainingsdaten (oben) MLP-Vorhersage (mitte) und SVR-Vorhersage (unten).

sollten sich diese Einflüsse minimieren und nur wenig um einen mittleren Wert variieren. Um dennoch vergleichbare Ergebnisse zu erzielen, wurden die mittleren Ähnlichkeitsmaße der Vorhersage, erhalten durch eine Kreuzvalidierung, nochmals über zehn unabhängige Messungen gemittelt. Die jeweiligen Ergebnisse sind in Abbildung 4.14 dargestellt. Die besten Vorhersageergebnisse werden durch Einsatz des MLP erzielt gefolgt von der SVR. Dabei wird auch deutlich, dass die Ergebnisse des MLP stärkeren Schwankungen unterliegen als die der anderen, deterministischen Methoden, die keiner Zufallsinitialisierung oder ähnlichen unterliegen.
Darüber hinaus wurden die Verteilungen der vorhergesagten Parameter untersucht und mit denen der Trainingsdaten verglichen, wie in Abbildung 4.15 dargestellt. Es zeigt sich, dass die Verteilungen der vorhergesagten Parameter mittels SVR annähernd normalverteilt sind und eine große Ähnlichkeit zu den Verteilungen der Trainingsdaten aufweisen. Die vorhergesagten Parameter mittels MLP hingegen weisen andere Formen der Verteilungen auf.
Abschließend soll nochmals visuell verdeutlicht werden, dass durch die untersuchte Implementierung keineswegs zufriedenstellende Vorhersageergebnisse in allen Fällen erreicht werden, so wie es die mittleren Ähnlichkeitsmaße auch schon andeuten. Ein solches Negativbeispiel ist in Abbildung 4.16 dargestellt. Die vorhergesagten Grundfrequenzverläufe ähneln keineswegs den natürlichen, sondern bilden eher einen konstanten Verlauf, der sich in einer monoton klingenden Betonung äußert. Damit tritt bei dieser Umsetzung dasselbe Problem auf, unter dem alle Systeme zur Prosodiesteuerung leiden.

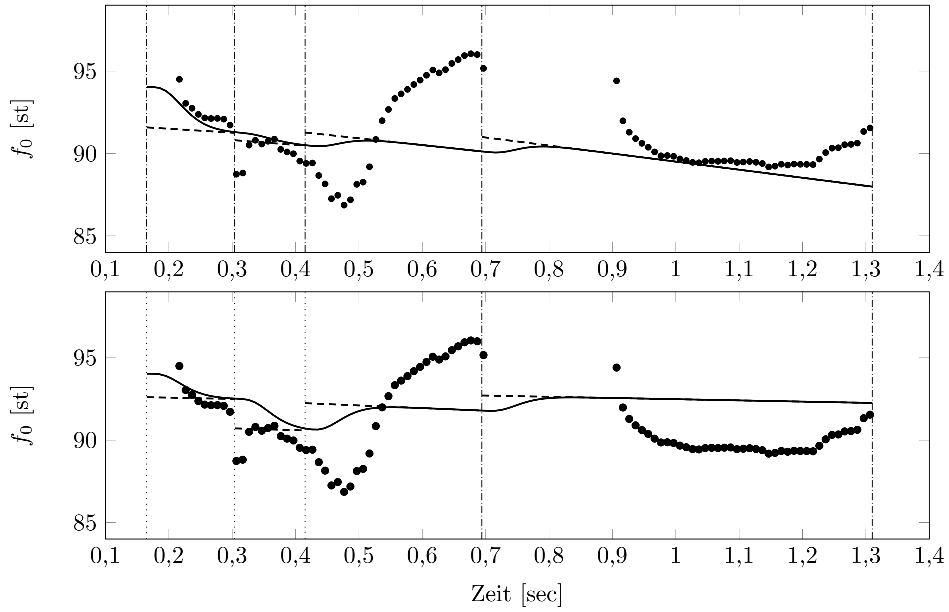

Abbildung 4.16: Vergleich der vorhergesagten f_0-Verläufe von MLP (oben, RMSE=2,373) und SVR (unten, RMSE=2,527) für die Äußerung *wiederholen* [viːdɐhˈeːᵄʃtɛln].

4.5 Perzeptionstest

Um die mathematischen Untersuchungskriterien besser einschätzen zu können, wurde ein Perzeptionstest durchgeführt. Dazu wurde der MOS der sieben beschriebenen Stichproben ermittelt, welcher in Abbildung 4.17 dargestellt ist. Dabei zeigt sich, dass bereits ein signifikanter Unterschied in der wahrgenommenen Natürlichkeit zwischen Äußerungen mit natürlicher f_0 und optimal modellierter f_0 besteht, was auf einen großen Einfluss mikroprosodischer Effekte für die wahrgenommene Natürlichkeit von Sprache hindeutet. Der Unterschied der wahrgenommenen Natürlichkeit zwischen optimal modellierten und vorhergesagten Grundfrequenzverläufen ist dabei nicht derartig herausstechend ausgeprägt. Ferner wurde der Einfluss der Silbengrenzenverschiebung auf die wahrgenommene Natürlichkeit untersucht. Die Ergebnisse legen nahe, dass keine relevanten Unterschiede bei verschobenen und nicht verschobenen Silbengrenzen bestehen.

Die aus den MOS abgeleiteten Ergebnisse spiegeln sich auch in der statistischen Analyse des Ergebnisse wieder. Nach den Ausführungen von Rosenberg und Ramabhadran (2017) muss für statistische Auswertungen der Wilcoxon-Mann-Whitney-Test, auch bekannt als U-Test, verwendet werden. Dieser kann bei statistischen Untersuchungen mit nominalen Skalen, was für die vorgegebene Natürlichkeitsskala mit den Werten $\{5, 4, 3, 2, 1\}$ zutrifft, angewendet werden und setzt keine Normalverteilung der Werte voraus. Der Test prüft zwei Stichproben auf Median-Unterschiede in der Grundgesamtheit. Die Ergebnisse sind in Tabelle 4.5 dargestellt und belegen die bereits diskutierten Ergebnisse auf Grundlage des MOS. Eine Korrektur des Signifikanzniveaus nach der Bonferroni-Holm-Prozedur würde im vorliegenden Fall zu keiner Änderung der Ergebnisse führen.

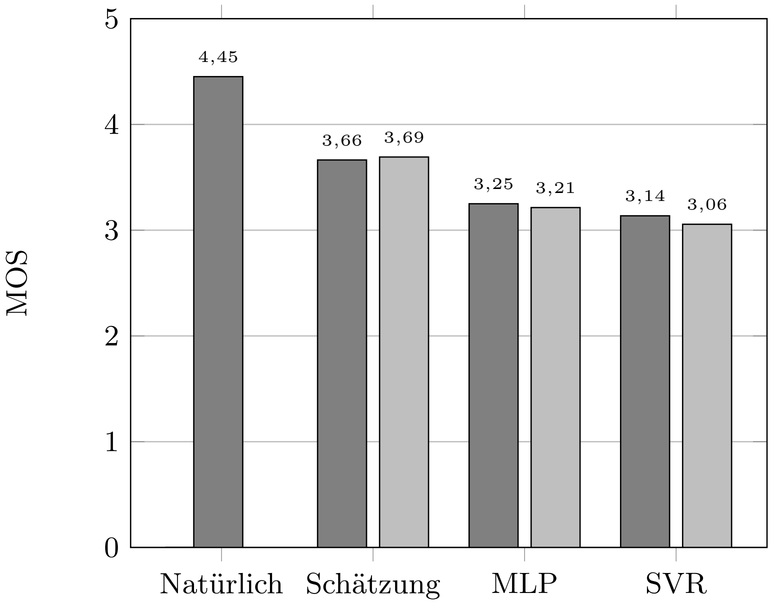

Abbildung 4.17: MOS der untersuchten Stichproben. Hellgraue Balken kennzeichnen eine Silbengrenzenverschiebung von $\tau = 40$ msec nach vorn.

1. Stichprobe	2. Stichprobe	p-Wert	Hypothese
Natürlich	Schätzung	$7 \cdot 10^{-34}$	H_1
Schätzung	MLP	$5 \cdot 10^{-8}$	H_1
Schätzung	SVR	$2 \cdot 10^{-12}$	H_1
Schätzung	Schätzung-40	0,697	H_0
MLP	MLP-40	0,652	H_0
SVR	SVR-40	0,280	H_0

Tabelle 4.5: Ergebnisse des Wilcoxon-Mann-Whitney-Test (U-Test) bei 5% Signifikanzniveau mit der Nullhypothese $H_0 : \tilde{x}_1 = \tilde{x}_2$ und Alternativhypothese $H_1 : \tilde{x}_1 \neq \tilde{x}_2$.

5 Diskussion der Ergebnisse

Die in Kapital 4 präsentierten Untersuchungsergebnisse zeigen, dass das eingesetzte Verfahren zur Parameterschätzung sehr gute Ergebnisse liefert und die vorgestellte Methode aus Prom-On et al. (2009) in vielen Punkten verbessert. Dieses Ergebnis wird durch Tabelle 4.1 belegt. Wie bereits erwähnt, kann durch einen überschaubaren Aufwand die Schätzung weiterhin verbessert werden, indem man den Initialzustand des Filters mit zugehörigen Onset als freie Parameter in das Optimierungsproblem mit einbaut. Somit wird insbesondere die Schätzung für die erste Silbe deutlich verbessert. Ein direkter Vergleich mit den Werten aus Prom-On et al. (2009) ist nicht möglich, da wie schon erläutert wurde, der mittlere RMSE über alle Silben des verwendeten Korpus gebildet wurde und nicht äußerungsbezogen. Davon abgesehen unterschieden sich die verwendeten Korpora in Sprache und Umfang.

Die gleichzeitige Schätzung aller $3S$ TAM-Parameter einer Silbe erhöht natürlich den Rechenaufwand deutlich. Auf einem handelsüblichen Prozessor liegt die Dauer einer Parameterschätzung für Einzelwörter mit bis zu acht Silben dennoch im einstelligen Sekundenbereich. Bei der Untersuchungen längerer Äußerungen auf Phrasen- oder Satzebene könnte sich die benötigte Rechenzeit jedoch stark erhöhen. Im Rahmen dieser Arbeit wurde der Code deshalb massiv parallelisiert und auf Prozessoren eines Hochleistungsrechners mit 24 Kernen ausgeführt, was das benannte Problem vernachlässigbar macht.

Außerdem konnte die Ergebnisse von Birkholz und Hoole (2012) belegt werden, dass ein System fünfter Ordnung für die Modellierung des TAM optimal ist. Dies lässt sich zum einen dadurch erklären, dass ein System der Ordnung fünf einen guten Kompromiss zwischen Berechenbarkeit und Komplexität darstellt. Andererseits lässt sich das komplette System vom Nervenimpuls bis zur artikulierten Grundfrequenz, hervorgerufen durch Muskelbewegungen, als ein System fünfter Ordnung modellieren, was den Gedanken nahelegt, dass ein solches System also völlig ausreichend und möglicherweise optimal für die Wahl des Filters nach dem TAM ist.

Die These von Xu und Liu (2006), welche eine Verbesserung des TAM durch nach vorn verschobene Silbengrenzen postuliert, da diese den eigentlich Silbengrenzen der Artikulation näher kommen, konnte hingegen nicht belegt werden. Die Ergebnisse aus Abbildung 4.6 zeigen, dass es keinen messbaren Zusammenhang zwischen Silbengrenzenverschiebung und Qualität der Modellanpassung gibt. Auch perzeptiv wurden von den Probanden keine Unterschiede zwischen den unterschiedlich modellierten f_0-Verläufen wahrgenommen, wie die Resultate aus Abbildung 4.17 zeigen.

Ein auffallendes Ergebnis des Perzeptionstest ist der relativ große Unterschied zwischen der wahrgenommen Natürlichkeit der originalen Äußerung und der optimal modellierten Äußerung. Dieses Ergebnis deutet darauf hin, dass mikroprosodische Effekte einen

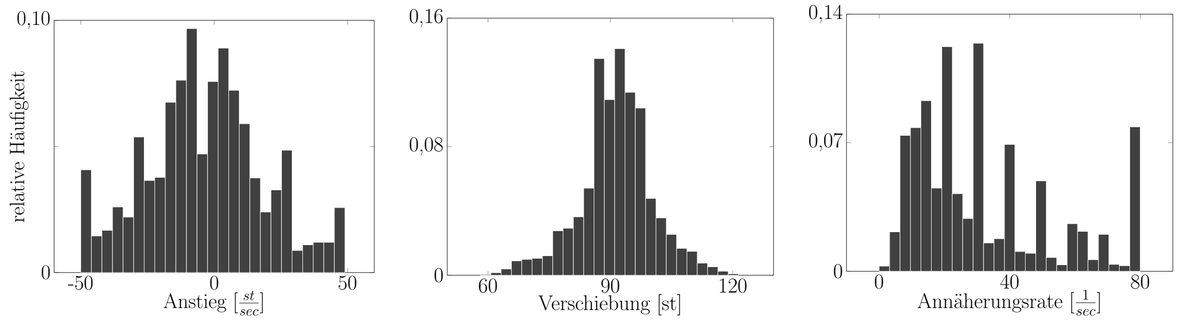

Abbildung 5.1: Relative Häufigkeiten der optimalen TAM-Parameter bestimmt durch PENTAtrainer1.

großen Einfluss auf die wahrgenommene Natürlichkeit von Sprache haben, da diese im TAM nicht berücksichtigt werden.

Zusammenfassend sei an dieser Stelle nochmals auf die Unterschiede zwischen der Parameterschätzung aus Prom-On et al. (2009) und der hier vorgestellten hingewiesen. Die Interpolation des natürlichen Grundfrequenzverlaufs liefert plausible Werte in stimmlosen Abschnitten. Außerdem ist durch die Verwendung der interpolierten f_0 als Grundlage der Schätzung eine silbenweise Bestimmung der PTs eindeutig und keine gleichzeitige Schätzung aller TAM-Parameter einer Äußerung notwendig, da die interpolierte Kurve über den gesamten Definitionsbereich definiert ist und damit keine weiteren Freiheitsgrade für eine Optimierung offenlässt. Nachteilig wirkt sich hingegen die Minimierung des RMSE in stimmlosen Abschnitten aus, da dies dazu führt, dass markante Stellen des natürlichen f_0-Verlaufs herausgemittelt werden. Es kann jedoch auch argumentiert werden, dass gerade dadurch natürlichere PTs gefunden werden, die nicht so starke Extremwerte aufweisen, wie sie bei einer unregularisierten Schätzung nach der in dieser Arbeit auftretenden Methode entstehen, wie Abbildung 4.9 zeigt. Dies wird deutlich, wenn man die Verteilung der mittels PENTAtrainer1 bestimmten Targets betrachtet, wie in Abbildung 5.1 veranschaulicht.

Wendet man jedoch die erweiterte, in dieser Arbeit entwickelte Schätzung an und berücksichtigt nur die Abtastpunkte der natürlichen f_0 für die Schätzung bei gleichzeitiger Betrachtung aller TAM-Parameter, kann zwar eine deutlich bessere Anpassung der modellierten f_0 an die natürliche erzielt werden, jedoch nur auf Kosten unnatürlicher PTs, wie in Kapitel 4 besprochen. Die verbesserte Anpassungsfähigkeit beruht auf der Tatsache, dass der modellierte f_0-Verlauf in stimmlosen Abschnitten als Freiheitsgrad betrachtet werden kann und durch die Optimierung ein optimaler Verlauf gefunden wird, der wiederum vorteilhaft für die Anpassung an stimmhafte Abschnitte ist. Um jedoch natürliche Targets zu erhalten, wurde eine Regularisierung notwendig, was wiederum die Qualität der Anpassung und damit die Möglichkeit zur Abbildung markanter Stellen im f_0-Verlauf verringert hat. Wie bereits besprochen, ist das TAM ohne Regularisierung eher als quantitatives Modell zu werten und nicht als phonetisches. Aus diesen Gedanken lässt sich schlussfolgern, dass eine silbenweise Schätzung auf Basis der interpolierten f_0 zu den selben Ergebnissen führt wie eine gemeinsame Schätzung auf Basis der Abtastwerte des PDA mit zusätzlicher Regularisierung. Beide Ansätze schränken das Schätzproblem gewisser Maßen ein, wobei die regularisierte Variante jedoch eine höhere Flexibilität bietet und besser im Sinne einer Modellreduktion erklärt werden kann.

Ferner ermöglichten die regularisierten PTs eine besser Vorhersage. Im Rahmen dieser Arbeit wurde auch untersucht, wie sich die Verwendung der mittels PENTAtrainer1 bestimmten PTs auf die Vorhersage auswirkt, wobei festgestellt wurde, dass die Vorhersageergebnisse in allen untersuchten Werten schlechter waren, als die in Abschnitt 2.2 vorgestellten Ergebnisse. Aus Gründen der Übersichtlichkeit wurde auf eine detaillierte Darstellung der Ergebnisse verzichtet, da sie keine zusätzlichen Schlussfolgerungen erlauben. Betrachtet man die Streudiagramme für die Daten, die mittels PENTAtrainer1 bestimmt wurden, ergeben sich die selben Beobachtungen wie in Abbildung 4.10.

Weiterhin ist in Anbetracht der Untersuchungsergebnisse aus Kapitel 4 festzustellen, dass die entwickelte Lösungsmethode zur Grundfrequenzvorhersage teils zu guten Ergebnissen führt, aber dennoch nicht in der Lage ist, natürliche Grundfrequenzverläufe in hoher Qualität zu generieren. Oftmals ist das System nicht in der Lage, markante Stellen im Grundfrequenzverlauf vorherzusagen und generiert monotone Verläufe. Damit weist es ähnliche Probleme wie existierende Ansätze auf. Ein Hauptproblem für die mangelhafte Leistungsfähigkeit des Systems liegt sicher in der geringen Größe des Korpus, welcher für das Training verwendet wurde, wie in Abschnitt 3.2 bereits erörtert wurde. Dies zeigt sich auch u.a. daran, dass bei der SVR teilweise bis zu 50% der Eingangsdaten als Stützvektoren berechnet wurden. Das wiederum belegt, dass nicht genügend Information vorlag, um den gesuchten Zusammenhang zu beschreiben. Da das Korpus erstellt wurde, um auf Besonderheiten in der deutschen Aussprache hinzuweisen, ist dieser zudem nicht optimal für die Zwecke dieser Arbeit geeignet.

Dennoch lässt sich feststellen, dass mit der entwickelten Lösungsmethode aufbauend auf dem beschriebenen Korpus teils bessere Ergebnisse als in der Literatur vorgefunden erzielt werden konnten. Das vorgestellte Systeme zur Vorhersage der Grundfrequenz im Deutschen, aufbauend auf dem Fujisaki Modell, von Mixdorff und Jokisch (2001b) erreicht einen mittleren Korrelationskoeffizienten von $\rho = 0,55$ und einen MOS von weniger als 3,0 im Perzeptionstest bei größeren Korpusumfang. Bei den Untersuchungen in Mixdorff und Jokisch (2001a) wurden ähnliche MOS Werte erzielt, wie die im Rahmen dieser Arbeit ermittelten. Ebenso haben Bailly und Holm (2005) mit ihrer Implementierung nur ein Korrelationswert von $\rho = 0.63$ erreicht, der hinter dem Ergebnis dieser Arbeit liegt. Fairerweise muss jedoch betont werden, dass sich die Untersuchungen in den genannten Quellen stets auf die Grundfrequenzvorhersage ganzer Sätze bezogen hat und nicht nur auf Einzelwörter. Darüber hinaus wurden in den benannten Referenzen auch andere Korpora verwendet. Trotzdessen belegt ein Vergleich der Zahlenwerte, dass die entwickelte Lösung durchaus kompetitiv zu bestehenden Ansätzen ist. Bei Verwendung eines geeigneteren und vor allem umfangreicheren Korpus sind nochmals deutlich bessere Ergebnisse zu erwarten. Darüber hinaus ist eine Erweiterung der vorgestellten Lösung für die Grundfrequenzvorhersage von Phrasen bzw. Sätzen problemlos möglich. Leider lassen sich in der Literatur keine Werte für Einzelwortvorhersagen für einen direkten Vergleich finden, um die Ergebnisse einzuordnen. Viele Veröffentlichungen zu dem Thema stellen erst gar keine detaillierten Untersuchungsergebnisse vor, machen

Parameter	SVR	MLP
m	8,294 $\frac{\mathrm{st}^2}{\mathrm{sec}^2}$	16,634 $\frac{\mathrm{st}^2}{\mathrm{sec}^2}$
b	6,252 st^2	9,148 st^2
λ	15,405 $\frac{1}{\mathrm{sec}^2}$	14,964 $\frac{1}{\mathrm{sec}^2}$

Tabelle 5.1: Vorhersagefehler (MSE) für SVR und MLP der einzelnen TAM-Parameter.

keine Aussagen über das verwendete Korpus oder haben nur Spezialfälle im Fokus. Eine weitere Schwierigkeit bei solchen Vergleichen ergibt sich dadurch, dass die vorgestellten Systeme immer jeweils für andere Sprachen verwendet wurde und die Prosodie sehr von der Sprache abhängig ist.

Ein auffälliges Ergebnis dieser Arbeit ist, dass die besten Grundfrequenzvorhersagen mittels MLP erzielt werden konnten, was sowohl für die mathematischen Fehlerkriterien, als auch für den Perzeptionstest zutrifft. Die Ergebnisse waren dabei für das MLP immer leicht besser als für die SVR, obwohl die Lernstichprobe relativ gering war und ein besseres Abschneiden der Kernmethode zu erwarten gewesen wäre. Diesen Unterschied genau zu begründen, erweist sich jedoch als schwierig. Dabei kommt hinzu, dass die gezeigten Verteilungen der vorhergesagten Parameter für MLP und SVR sehr unterschiedlich sind. Die Verteilung der SVR Parameter ähnelt dabei der Verteilung der Trainingsdaten deutlich mehr, wodurch man vermuten würde, dass die SVR-Vorhersage besser geeignet sei. Diese Vermutung wird unter anderen durch einen Vergleich des Vorhersagefehlers gestützt, der in Tabelle 5.1 für die beiden Verfahren gegenübergestellt ist. Erstaunlicherweise zeigt sich, dass der Vorhersagefehler für die einzelnen TAM-Parameter mittels SVR kleiner oder annähernd gleich denen des MLP ist, was sich auch mit den Ergebnissen von Lazaridis et al. (2014) deckt. Dennoch würde dieses Ergebnis erwarten lassen, dass die aus den geschätzten TAM-Parametern erzeugten Grundfrequenzverläufe für die SVR-Schätzung bessere Ähnlichkeitsmaße zwischen vorhergesagter f_0-Kontur und natürlicher f_0 aufweisen. Die Ergebnisse aus Abbildung 4.14 belegen jedoch, dass mittels MLP-Parameterschätzung bessere Ähnlichkeitsmaße für die Grundfrequenzvorhersage erzielt wurden. Der geringere Vorhersagefehler der SVR-Vorhersage gegenüber der MLP-Vorhersage scheint vernünftig, was sich hauptsächlich auf den geringen Korpusumfang zurückführen lässt. Kernmethoden nutzen in der Regel die Informationen der Lernstichprobe besser aus, da alle Daten für die Vorhersage verwendet werden. Parametrische Verfahren hingegen benötigen meist eine größere Anzahl von Trainingsbeispielen, um die Gewichte gut einzustellen. Da das zur Verfügung stehende Korpus relativ klein ist, spiegelt sich dieses Phänomen auch im Vorhersagefehler aus Tabelle 5.1 wieder.

Die Tatsache, dass mittels MLP-Vorhersage bessere Ähnlichkeitsmaße bzgl. der natürlichen f_0 erzielt wurden, deutet im Allgemeinen darauf hin, dass die Lernmethoden auf Grundlage der gegebenen Daten nicht in der Lage sind, das Problem ausreichend zu generalisieren. Dies wiederum bedeutet entweder, dass kein wirklicher Zusammenhang zwischen der SAMPA-Transkription und den zugehörigen PTs besteht oder wie bereits erwähnt, der Korpusumfang zu gering war um einen solchen zu bestimmen. Auch wenn die Zahlenwerte aus Tabelle 5.1 nicht direkt miteinander vergleichbar sind, so zeigen die Beobachtungen, dass die Vorhersage der Verschiebung besser funktioniert als die Vorhersage des Anstiegs. Dies deutet daraufhin, dass die Lernmethoden auf Basis der bestimmten Merkmale keine wirklichen Zusammenhänge zum TAM-Parameter Anstieg ermitteln konnten.

Eine visuelle Analyse der vorhergesagten f_0-Verläufe zeigt, dass das MLP offensichtlich besser die Positions- und Akzentinformationen über die jeweilige Silbe nutzt. Damit ist beispielsweise gemeint, dass im Deutschen die meisten zweisilbigen Äußerungen in der ersten Silbe betont sind und eine höhere Grundfrequenz aufweisen als in der zweiten Silbe. Allgemeiner lässt sich feststellen, dass oft die vorletzte Silbe betont wird und in der letzten Silbe die Grundfrequenz stark abfällt. Diese Regeln wurden vom MLP besser gelernt als von der SVR und sind durch zwei Beispiele in Abbildung 5.2 gezeigt. Aus den gegeben Beispielen wird auch ersichtlich, dass die Vorhersage der Verschiebung den größten Einfluss auf den damit verbundenen f_0-Verlauf aufweist. Dies liegt in erster Linie daran, dass die Annäherungsrate durch die Regularisierung auf einen sehr kleinen Bereich um 79 $\frac{1}{sec}$ begrenzt wurde und auch der Anstieg zumeist im negativen Bereich nahe 0 liegt und den grundsätzlichen Verlauf nicht entscheidend beeinflussen. Es ist jedoch sehr auffällig, dass durch die SVR Vorhersage sehr monotone Verläufe erzeugt werden. Dieses Ergebnis spiegelt sich auch in der Verteilung der vorhergesagten Verschiebungen aus Tabelle 4.15 wieder. Man erkennt, dass sehr oft Verschiebungen von 92 st gewählt werden und die Variation um diesen Wert geringer ausfällt als bei den entsprechenden Trainingsdaten. Die Verteilung des vorhergesagten Verschiebungsparameters mittels MLP weist jedoch eine interessante Verteilung auf, durch welche zu erkennen ist, dass überdurchschnittlich oft Werte unter 90 st und über 95 st vorhergesagt werden. Gerade durch diese PTs wird in vielen Fällen die richtige Tendenz des f_0-Verlaufs vorhergesagt und beispielsweise ein deutlicher Unterschied zwischen vorletzter und letzter Silbe in der Verschiebung. Aus diesem Grund sind auch die gemessenen Ähnlichkeitsmaße für die MLP-Vorhersage besser als für die SVR. Dass der Vorhersage-

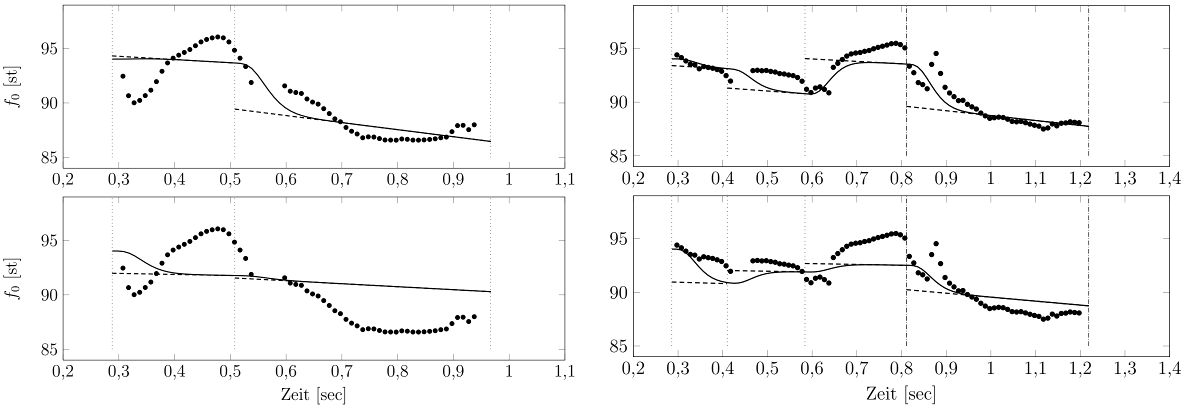

Abbildung 5.2: Vergleich der vorhergesagten f_0-Verläufe von MLP (oben) und SVR (unten) für die Äußerungen *Visum* [vˈiːzʊm] (links) und *Unterredung* [ʊntɐʁˈeːdʊŋ] (rechts).

fehler der einzelnen TAM-Parameter für die SVR dennoch geringer ist, lässt sich damit begründen, dass diese besonders häufig um einem spezifischen Mittelwert liegen und auch im Mittel eine geringere Abweichung aufweisen, was den Gesamtfehler klein hält. Da bei der MLP-Vorhersage extremere Werte vorhergesagt werden, wirkt sich ein Fehler natürlich stärker aus.

Da die Leistungsfähigkeit der Vorhersage nicht direkt am Vorhersagefehler gemessen wird, sondern an den Ähnlichkeitsmaßen der generierten Grundfrequenzverläufe, könnten die Hyperparameter auch beispielsweise anhand des mittleren RMSE optimiert werden. Eine direkte Optimierung anhand der natürlichen f_0 minimiert etwaige Fehler, die bei der Parameterschätzung gemacht werden. In Abbildung 3.2 kann dieser Unterschied im Optimierungskriterium nochmals nachvollzogen werden. Das Problem wird dadurch jedoch komplexer, da eine Filterung der TAM-Parametern in jedem Optimierungsschritt für das gesamte Korpus nötig ist. Damit bleibt die Frage zu klären, wie sich die so ermittelten Hyperparameter auf die Vorhersagefehler von MLP und SVR auswirken.

Die Untersuchungsergebnisse werfen insgesamt die Frage auf, wie gut sich die aus der Parameterschätzung erhaltenen PTs überhaupt für die Vorhersage eignen bzw. ob den Daten grundsätzlich ein untersagter Zusammenhang unterliegt. Dies könnte zum einen darauf zurückzuführen sein, dass, wie anfangs besprochen, eigentlich nur sehr wenig Information über die Prosodie im Text codiert ist. Eine andere Einschränkung ergibt sich durch die quantitative Umsetzung des TAM. Es gibt nämlich keinerlei Gewährleistung, dass durch die vorgenommene Parameterschätzung im Sinne einer mathematischen Optimierung die besten Werte der Targets im Sinne des Modells bestimmt werden. Dies zeigt sich allein schon in der Notwendigkeit der Regularisierung, um überhaupt einigermaßen plausible Vorhersageergebnisse zu erhalten. Der Einfluss der Regularisierung kann nämlich auch wie nachfolgend beschrieben, interpretiert werden. Ohne Regularisierung werden PTs bestimmt, die f_0-Konturen mit einem minimalen Fehler zur natürlichen Kontur beschreiben und markante Stellen im Verlauf erfassen. Dies trifft jedoch nicht für die Vorhersage zu. Zwar werden durch die unregularisierten Targets für das Training auch deutlich markantere PTs vorhergesagt, bloß sind diese meistens völlig abwegig und erzeugen sehr unnatürliche Grundfrequenzverläufe mit sehr geringer bis keiner Ähnlichkeit zu äquivalenten, natürlichen Verläufen. Die Regularisierung dämmt dieses Phänomen ein und sorgt dafür, dass der Wertebereich der Parameter wesentlich kleiner wird, wodurch auch kaum extreme oder markante Targets vorhergesagt werden und durch eine vorteilhafte Schwankung um einen Mittelwert Verläufe gefunden werden, die eher monoton sind, aber dafür keine zu starken Abweichungen aufweisen. Damit stellt sich die Frage, ob durch die gefundenen PTs überhaupt virtuelle, lexikalische Töne für das Deutsche darstellen, oder das TAM nicht auch vielmehr nur als rein quantitativen Grundfrequenzmodell fungiert und die damit verbunden Probleme aufweist. Somit wäre für weitere Untersuchungen erst einmal die Frage zu klären, ob die Annahme virtueller, lexikalischer Töne für das Deutsche gerechtfertigt ist und diese durch die Art der Parameterschätzung überhaupt gefunden werden. Ein aufschlussreicher Vergleichswert für eine Vorhersage ergibt sich dadurch, die Ähnlichkeitsmaße für einen komplett monotonen f_0-Verlauf mit einem mittleren, sprecherspezifischen Wert zu bestimmen. Ein solcher Wert könnte als Grenzfall betrachtet werden und ermöglicht eine bessere Einordnung der Ergebnisse. Im Rahmen dieser Arbeit wurde ein solcher Vergleichswert jedoch nicht ermittelt. Weiterhin müssen zusätzliche Einschränkungen für das Modell gefunden werden, um eine Vorhersage zu verbessern und mögliche Zusammenhänge besser heraushebt. Diese These wird durch die Ergebnisse aus Abbildung 4.10 gestützt, die auf eine Überbestimmtheit der geschätzten TAM-Parameter hinweist.

Ursprünglich war im Rahmen dieser Arbeit geplant, auch komplexere Modelle neuronaler Netze zu implementieren und für die Problemstellung zu evaluieren. Die Bestimmung der optimalen Hyperparameter aus Tabelle 4.3 zeigt jedoch, dass gerade MLPs mit nur 1 bis 20 Neuronen pro Schicht als optimal für die Vorhersage identifiziert wurden. Dieses Ergebnis deutet nicht gerade darauf hin, dass tiefe neuronale Netze bei dem gegebenen Problem erfolgversprechender sind. Gerade in Hinblick auf den geringen Umfang der zur Verfügung stehenden Lernstichprobe wurde die Entscheidung getroffen, kein tiefes neuronales Netz zu implementieren, da sich keine Vorteile absehen lassen. Darüber hinaus war es für dieses Problem möglich, relevante Merkmale manuell zu identifizieren, wodurch keine automatische Merkmalauswahl durch ein Deep-Belief-Network notwendig ist.

Auch auf eine Implementierung eines rekurrenten Netzwerks wurde verzichtet, da die Äußerungen im Durchschnitt nur aus 2,16 Silben bestehen und damit keine ausgeprägte, zeitliche Struktur aufweisen. Außerdem wurden bereits jeweils Akzent- und Positionsinformationen aus der jeweiligen vorherigen und folgenden Silbe mit in der Merkmaltransformation einbezogen, wodurch die zeitliche Dimension in den Merkmalen selbst erfasst wurde. In Anbetracht dessen hätte ein rückgekoppelter Ansatz im Falle der Einzelwortvorhersage keinerlei Vorteil gebracht. Bei einer Erweiterung auf ein System für die Grundfrequenzvorhersage auf Phrasen oder Sätze ist die zeitliche Struktur des Problems jedoch wesentlich stärker ausgeprägt und ein rückgekoppeltes System sicher zu bevorzugen.

6 Zusammenfassung und Ausblick

„Most popular machine-learning algorithms have been applied to prosodic prediction. The results are often similar, and difficult to improve upon because of the inherent difficulties in representing prosodic form and underspecification in the text.“ (Taylor, 2009)

Die Untersuchungsergebnisse dieser Arbeit zeigen, dass obiges Zitat aus dem Standardwerk „Text-to-Speech Synthesis“ nach wie vor zutreffend ist. Auch die Verwendung des vielversprechenden Target-Approximation-Modells für die Grundfrequenzvorhersage bietet keine allumfassende Lösung der Problemstellung. Wie von Taylor angesprochen liegen die zwei Hauptschwierigkeiten zum einen darin, dass im Text fast keine Information über die Prosodie codiert ist und zum anderen in der unzureichenden Modellierung der Grundfrequenz. So hat sich zum einen gezeigt, dass auch das TAM gewissermaßen überbestimmt ist und viele unnatürliche Grundfrequenzverläufe durch entsprechende Parameterwahl zulässt und wie fast alle Modelle eine Modellierung der Mikroprosodie vernachlässigt.

Trotz aller Probleme konnte jedoch gezeigt werden, dass es dennoch möglich ist, auf Basis des TAM ein Vorhersagesystem zu entwickeln, welches trotz geringer Korpusgröße und -eignung relativ gute Ergebnisse liefert und vergleichbar und teilweise besser als vorhandene Systeme für die Grundfrequenzvorhersage ist. Einen entscheidenden Anteil spielt dabei eine angemessene Parameterschätzung der TAM-Parameter aus natürlichen Äußerungen, die als Trainingsmaterial dienen. Dabei konnte die Schätzmethode aus Prom-On et al. (2009) deutlich verbessert werden und durch das Einbringen einer Regularisierung dafür gesorgt werden, dass physiologisch sinnvollere PTs bestimmt werden, die wiederum besser für eine darauf basierende Vorhersage geeignet waren. Dabei kann die Regularisierung als eine empirische Modellreduktion interpretiert werden. Die Kernidee der verbesserten Schätzung bestand primär darin, die Parameter ausschließlich auf Grundlage der vom PDA ermittelten Abtastwerte der natürlichen f_0 zu ermitteln, wobei alle Targets einer Äußerung gleichzeitig geschätzt wurden und nicht nacheinander. Dies sorgt insbesondere dafür, dass in stimmlosen Sprachabschnitten die f_0 so modelliert wird, dass sie einen vorteilhaften Verlauf in den stimmhaften Abschnitten aufweist, an denen ihr Einfluss hörbar ist. Die implementierte Parameterschätzung lässt sich sogar noch weiter verbessern, indem man den optimalen Onset der Modellkurve mit in die Schätzung einbezieht. Weiterhin konnte gezeigt werden, dass die Schätzung durch Verwendung eines Filters fünfter Ordnung optimiert werden kann und dass ein Einfluss einer Silbengrenzenverschiebung nicht feststellbar ist. Ein überraschendes Ergebnis liegt darin, dass eine Vorhersage mittels MLP zu besseren Ergebnissen bzgl. der Grundfrequenzkonturen als eine SVR-Vorhersage geführt hat, obwohl der Vorhersagefehler bzgl. der TAM-Parameter für die SVR geringer war als für das MLP.

Die Untersuchungen zeigen auch, dass das entwickelte System Potential für weitere Verbesserung bietet. So kann beispielsweise die Parameterschätzung erweitert werden, indem man eine Schätzung des initialen Filterzustands mit in das Optimierungsproblem einbaut. Die Vorhersage ließe sich evtl. weiterhin verbessern, wenn man die Hyperparameter auf die Ähnlichkeitsmaße hin optimiert anstatt auf den Vorhersagefehler. Darüber hinaus kann das System auch für die Schätzung der Silbendauern eingesetzt werden, welche ebenfalls einen TAM-Parameter darstellen. Die Silbendauern waren durch das annotierte Korpus vorgegeben. In den meisten TTS Systemen erfolgt eine Schätzung von Phonem- und Silbendauern in einem separaten Schritt, so dass diese dann für die Grundfrequenzvorhersage zur Verfügung stehen. Aufgrund mangelnder Zeit wurden die geschätzten Silbendauern im Rahmen dieser Arbeit jedoch nicht evaluiert. Ebenso lässt sich das Modell für die Vorhersage phrasen- oder satzbezogener Grundfrequenzverläufe erweitern. Dabei müssen jedoch noch weitere Merkmale mit in die Merkmaltransformation eingebaut werden. Ferner könnte die Identifikation relevanter Merkmale durch Methoden der Merkmalauswahl (engl. Feature Selection) zu einer Steigerung der Leistungsfähigkeit führen. In jedem Falle wäre es sehr aufschlussreich, die Relevanz der phonetischen Merkmale näher zu untersuchen und deren Einfluss zu messen. Für all die vorgeschlagenen weiterführenden Untersuchungen sollte jedoch ein größeres und geeigneteres Korpus verwendet werden. Einen solches Korpus vorausgesetzt macht es auch Sinn, Methoden des Deep-Learning näher zu untersuchen.

Schließlich hat sich in den Ergebnissen auch gezeigt, dass das TAM zwar ein theoretisches Modell aus phonetischer Sicht ist, jedoch bei einer quantitativen Umsetzung noch zahlreiche Probleme auftreten. Aus der Perspektive der Grundfrequenzvorhersage wäre es wünschenswert, noch mehr artikulatorische Grenzen für das Modell zu untersuchen, um die Möglichkeiten generierter Konturen weiter zu beschränken. Jedoch ist wohl auch klar, dass sich solche Beschränkungen oder Abhängigkeiten der TAM-Parameter nur schwer identifizieren lassen und noch schwieriger in eine elegante, explizite mathematische Beschreibung zu bringen sind. Wahrscheinlich bleiben am Ende zur Lösung dieses Problems nur empirische Methoden wie die Regularisierung oder statistische, wie beispielsweise in Prom-On et al. (2009), wo ein Mittelwert über alle PTs gleichen Typs gebildet wurde.

Als Ergebnis dieser Arbeit wurde das Praat-Tool TargetOptimizer entwickelt, welches eine verbesserte Schätzmethode zur Bestimmung der TAM-Parameter implementiert. Zusätzlich können die Ergebnisse in einem VocalTractLab spezifischen Format ausgegeben werden, was der weiteren Forschung am betreuenden Lehrstuhl dient. Auch eine Funktionalität zur Vorhersage der Grundfrequenz und Silbendauer wird bereitgestellt, wobei diese unter Anbetracht der Untersuchungsergebnisse jedoch noch weiter verbessert werden sollte. Dies ist in sofern einfach möglich, da das trainierte Modell als Binärdatei vorliegt und einfach ausgetauscht werden kann, falls ein anderes Korpus verwendet werden soll.

Anhang A

A Kurzdokumentation der Software

Verwendete Software

Linux Kernel	4.4.0-96
Ubuntu	16.04 LTS
gcc	6.2.0
Dlib	19.4
GnuPlot	4.6.1
Praat	6.0.28
GNU bash	4.3.48
GNU make	4.1
PENTAtrainer1	1.9.3.7

Projekt-Struktur

```
qta-learn-f0
 ├─ bin
 │   └─ TargetOptimizer, qta-data, qta-pred, qta-stat
 ├─ build
 │   └─ *.o
 ├─ corpus
 │   └─ *.wav, *.TextGrid, *.PitchTier, sampa.csv
 ├─ include
 │   └─ *.h
 ├─ learn
 ├─ lib
 │   └─ Dlib
 ├─ src
 │   └─ *.cpp
 ├─ test
 ├─ tools
 │   └─ Praat, TargetOptimizer.praat
 ├─ job.batch
 ├─ Makefile
 └─ qta-learn-f0.sh
```

Anwendung

Die nötigen Befehle zum Kompilieren der Software wurden in ein Makefile geschrieben. Als erstes kann bei Bedarf eine spezielle Version der Software Praat gebaut werden, die keinerlei Grafik- und Soundbibliotheken benötigt und demnach auf Server-Platformen lauffähig ist. Dies ist insbesondere notwendig, wenn die Software auf einem HPC System ausgeführt werden soll. Durch den nachfolgenden Befehl wird die entsprechende Version kompiliert.

```
make praat
```

Alle anderen Teile der erstellten Software werden in einem separaten Schritt kompiliert, wodurch die Binärdateien TargetOptimizer, qta-data, qta-pred und qta-eval erzeugt werden. Dieser Prozess wird durch den Befehl

```
make
```

ausgelöst. Durch ein Kompilieren mit der Präprozessoranweisung -DDEBUG_MSG werden Zusatzinformationen bei der Ausführung der Programme ausgegeben. Danach stehen alle Softwarekomponenten zur Anwendung bereit. Als erstes sollte die Funktionsfähigkeit der Software durch einen Test überprüft werden. Dabei werden alle Funktionalitäten ausgeführt und auf einem kleinen Testdatensatz angewendet. Durch den Befehl

```
make test
```

wird das Hauptskript qta-learn-f0.sh aufgerufen, welches den Gesamtablauf steuert. Ist dieser Test erfolgreich durchgelaufen, so können im Skript qta-learn-f0.sh alle Nutzerparameter beliebig eingestellt werden, so wie sie für die nachfolgende Verarbeitung des gesamten Korpus benötigt werden. Zusätzlich lässt sich auswählen, welche der Programmblöcke ausgeführt werden sollen (Parameterschätzung, Modellauswahl, Vorhersage). Der Befehl

```
make learn
```

startet dann die vollautomatische Korpusanalyse und Grundfrequenzvorhersage, wiederum durch Ausführen des Skripts qta-learn-f0.sh, wobei alle Daten im Verzeichnis `learn` für die Berechnungen verwendet werden. Diese müssen vor der Berechnung also in diesem Verzeichnis abgelegt bzw. symbolische Links auf das `corpus`-Verzeichnis gesetzt werden. Alle Ergebnisse finden sich anschließend ebenfalls im Verzeichnis `learn`.
Werden die Berechnungen auf dem HPC des ZIH ausgeführt, muss die Software slurm zur Verwaltung von Rechenaufträgen verwendet werden, damit das Programm auf entsprechenden Rechenknoten laufen kann. Durch

```
make job
```

kann ein solcher Rechenauftrag aufgegeben werden. Dieser Befehl startet das bash-Skript job.batch, welches dann das eigentliche Programm in einem slurm-Kontext startet.

Literaturverzeichnis

Anderson, Mark, Janet Pierrehumbert und Mark Liberman (1984). „Synthesis by rule of English intonation patterns“. In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'84.* Bd. 9. IEEE, S. 77–80.

Bailly, Gérard und Bleicke Holm (2005). „SFC: a trainable prosodic model“. In: *Speech communication* 46.3, S. 348–364.

Balyan, Archana, S. Agrawal und Amita Dev (2013). „Speech synthesis: A review“. In: *International Journal of Engineering Research & Technology (IJERT)* 2.6, S. 57–75.

Birkholz, Peter und Phil Hoole (2012). „Intrinsic velocity differences of lip and jaw movements: preliminary results“. In: *Thirteenth Annual Conference of the International Speech Communication Association.*

Birkholz, Peter, Bernd J. Kröger und Christiane Neuschaefer-Rube (2011). „Model-based reproduction of articulatory trajectories for consonant–vowel sequences“. In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.5, S. 1422–1433.

Birkholz, Peter, Uwe Reichel und Christiane Neuschaefer-Rube (2014). „Phone duration prediction for speech synthesis with echo state networks vs. multilayer perceptrons“.

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning.* Springer Science + Business Media.

Black, Alan W. und Andrew J. Hunt (1996). „Generating F_0 contours from ToBI labels using linear regression“. In: *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on.* Bd. 3. IEEE, S. 1385–1388.

Boersma, Paul (1993). „Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound“. In: *Proceedings of the institute of phonetic sciences.* Bd. 17. 1193. Amsterdam, S. 97–110.

Boersma, Paul und David Weenink (2017). *Praat: doing phonetics by computer.* URL: `http://www.fon.hum.uva.nl/praat/` (besucht am 05. 10. 2017).

Boyd, Stephen und Lieven Vandenberghe (2004). *Convex optimization.* Cambridge university press.

Chang, Chih-Chung und Chih-Jen Lin (2011). „LIBSVM: a library for support vector machines“. In: *ACM transactions on intelligent systems and technology (TIST)* 2.3, S. 27.

DeGruyter (2009). *Deutsches Aussprachewörterbuch. Verlagswebsite.* URL: `https://www.degruyter.com/view/product/19839` (besucht am 22. 10. 2017).

Dusterhoff, Kurt E. und Alan W. Black (1997). „Generating f0 contours for speech synthesis using the tilt intonation theory.“ In:

Förster, Johannes (2014). „Aufbau und Entwicklung der Deutschen Aussprachedatenbank (DAD)“. In: *Aussprache und Sprechen im interkulturellen, medienvermittelten und pädagogischen Kontext. Beiträge zum 1. Doktorandentag der Halleschen Sprechwissenschaft.* Hrsg. von Alexandra Ebel.

Frota, Sónia, Pedro Oliveira, Marisa Cruz und Marina Vigário (2015). *P-ToBI: tools for the transcription of Portuguese prosody.* URL: `http://labfon.letras.ulisboa.pt/InAPoP/P-ToBI/index.html` (besucht am 09. 22. 2017).

Fujisaki, Hiroya und Keikichi Hirose (1984). „Analysis of voice fundamental frequency contours for declarative sentences of Japanese“. In: *Journal of the Acoustical Society of Japan (E)* 5.4, S. 233–242.

Hinton, Geoffrey E., Simon Osindero und Yee-Whye Teh (2006). „A fast learning algorithm for deep belief nets“. In: *Neural computation* 18.7, S. 1527–1554.

Hirose, Keikichi und Hiroya Fujisaki (1982). „Analysis and synthesis of voice fundamental frequency contours of spoken sentences“. In: *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'82.* Bd. 7. IEEE, S. 950–953.

Hirose, Keikichi und Jianhua Tao, Hrsg. (2015). *Speech Prosody in Speech Synthesis: Modeling and generation of prosody for high quality and flexible speech synthesis.* Springer.

Hoffmann, Rüdiger und Matthias Wolff (2014). *Intelligente Signalverarbeitung 1. Signalanalyse.* Springer.

– (2015). *Intelligente Signalverarbeitung 2. Signalerkennung.* Springer.

Hsu, Chih-Wei, Chih-Chung Chang und Chih-Jen Lin (2016). *A Practical Guide to Support Vector Classification.* URL: `https://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf` (besucht am 16. 08. 2017).

Jilka, Matthias, Gregor Möhler und Grzegorz Dogil (1999). „Rules for the generation of ToBI-based American English intonation“. In: *Speech Communication* 28.2, S. 83–108.

Jokisch, Oliver, Hansjörg Mixdorff, Hans Kruschke und Ulrich Kordon (2000). „Learning the parameters of quantitative prosody models.“ In: *INTERSPEECH*, S. 645–648.

Kay, Steven M. (1993). *Fundamentals of statistical signal processing. Estimation Theory.* Prentice Hall PTR.

King, Davis E. (2009). „Dlib-ml: A machine learning toolkit“. In: *Journal of Machine Learning Research* 10.Jul, S. 1755–1758.

– (2017). *Dlib C++ Library.* URL: `dlib.net` (besucht am 27. 09. 2017).

Krech, Eva-Maria, Eberhard Stock, Ursula Hirschfeld und Lutz Christian Anders (2009). *Deutsches Aussprachewörterbuch.* Walter de Gruyter.

Lazaridis, Alexandros, Pierre-Edouard Honnet und Philip N. Garner (2014). *SVR vs MLP for Phone Duration Modelling in HMM-based Speech Synthesis.* Techn. Ber. Idiap.

Lipton, Zachary C., John Berkowitz und Charles Elkan (2015). „A critical review of recurrent neural networks for sequence learning“. In: *arXiv preprint arXiv:1506.00019.*

MathWorks (2017). *Regularized Estimates of Model Parameters.* URL: `https://de.mathworks.com/help/ident/ug/regularized-estimates-of-model-parameters.html` (besucht am 21. 08. 2017).

Mixdorff, Hansjörg (1998). „Intonation Patterns of German - Quantitative Analysis and Synthesis of F0 Countours“. Diss. TU Dresden.

– (2000). „A novel approach to the fully automatic extraction of Fujisaki model parameters“. In: *Acoustics, Speech, and Signal Processing, 2000. ICASSP’00. Proceedings. 2000 IEEE International Conference on.* Bd. 3. IEEE, S. 1281–1284.

Mixdorff, Hansjörg und Oliver Jokisch (2001a). „Comparing a data-driven and a rule-based approach to predicting prosodic features of German“. In: *Tagungsband der 12. Konferenz Elektronische Sprachsignalverarbeitung*, S. 298–305.

– (2001b). „Implementing and evaluating an integrated approach to modeling German prosody“. In: *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis.*

Moulines, Eric und Francis Charpentier (1990). „Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones“. In: *Speech communication* 9.5-6, S. 453–467.

Pierrehumbert, Janet (1980). „The phonology and phonetics of English intonation“. Diss. Massachusetts Institute of Technology.

Pompino-Marschall, Bernd (2009). *Einführung in die Phonetik.* de Gruyter.

Powell, Michael J. (2009). „The BOBYQA algorithm for bound constrained optimization without derivatives“. In: *Cambridge NA Report NA2009/06, University of Cambridge, Cambridge.*

Prom-On, Santitham, Yi Xu und Bundit Thipakorn (2009). „Modeling tone and intonation in Mandarin and English as a process of target approximation“. In: *The Journal of the Acoustical Society of America* 125.1, S. 405–424.

Rifkin, Ryan M. und Ross A. Lippert (2007). „Notes on regularized least squares“. In:

Rios, Luis Miguel und Nikolaos V. Sahinidis (2013). „Derivative-free optimization: a review of algorithms and comparison of software implementations“. In: *Journal of Global Optimization* 56.3, S. 1247–1293.

Rosenberg, Andrew und Bhuvana Ramabhadran (2017). „Bias and Statistical Significance in Evaluating Speech Synthesis with Mean Opinion Scores“. In: *Proc. Interspeech 2017*, S. 3976–3980.

Russell, Stuart und Peter Norvig (2012). *Künstliche Intelligenz. Ein moderner Ansatz.* Pearson.

Schröder, Marc und Jürgen Trouvain (2003). „The German text-to-speech synthesis system MARY: A tool for research, development and teaching“. In: *International Journal of Speech Technology* 6.4, S. 365–377.

Scordilis, Michael S. und John N. Gowdy (1989). „Neural network based generation of fundamental frequency contours“. In: *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on.* IEEE, S. 219–222.

Smola, Alex J. und Bernhard Schölkopf (2004). „A tutorial on support vector regression“. In: *Statistics and computing* 14.3, S. 199–222.

Taylor, Paul (1994). „The rise/fall/connection model of intonation“. In: *Speech Communication* 15.1, S. 169–186.

– (2009). *Text-to-Speech Synthesis.* Cambridge University Press.

Theodoridis, Sergios (2015). *Machine Learning. A Bayesian and Optimization Perspective.* Elsevier.

Traber, Christof (1991). „F0 generation with a data base of natural F0 patterns and with a neural network“. In: *The ESCA Workshop on Speech Synthesis.*

Vapnik, Vladimir N. (1995). *The Nature of Statistical Learning Theory.* Springer New York.

Vary, Peter, Ulrich Heute und Wolfgang Hess (1998). *Digitale Sprachsignalverarbeitung.* B. G. Teubner Stuttgart.

Wu, Zhizheng, Oliver Watts und Simon King (2016). „Merlin: An open source neural network speech synthesis system“. In: *Proc. SSW, Sunnyvale, USA.*

Xu, Yi (2004). „The PENTA model of speech melody: Transmitting multiple communicative functions in parallel“. In: *Proceedings of from sound to sense* 50, S. 91–96.

Xu, Yi und Fang Liu (2006). „Tonal alignment, syllable structure and coarticulation: Toward an integrated model“. In: *Italian Journal of Linguistics* 18.1, S. 125.

Xu, Yi und Santitham Prom-On (2014). „Toward invariant functional representations of variable surface fundamental frequency contours: Synthesizing speech melody via model-based stochastic learning“. In: *Speech Communication* 57, S. 181–208.

– (2015). *PENTAtrainer1 - A Praat script for automatic analysis and synthesis of intonation based on the PENTA model.* URL: `http://www.homepages.ucl.ac.uk/~uclyyix/PENTAtrainer1/` (besucht am 16. 08. 2017).

Xu, Yi und Xuejing Sun (2000). „How fast can we really change pitch? Maximum speed of pitch change revisited“. In: *Sixth International Conference on Spoken Language Processing.*

Xu, Yi und Q Emily Wang (2001). „Pitch targets and their realization: Evidence from Mandarin Chinese“. In: *Speech communication* 33.4, S. 319–337.

Yoshimura, Takayoshi et al. (1999). „Simultaneous modeling of spectrum, pitch and duration in HMM-based speech synthesis“. In: *Sixth European Conference on Speech Communication and Technology.*

Ze, Heiga, Andrew Senior und Mike Schuster (2013). „Statistical parametric speech synthesis using deep neural networks“. In: *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on.* IEEE, S. 7962–7966.