



Jan Simon Ročnik, BSc

Dynamische Intonation von digitalen Konzertorgeln

MASTERARBEIT

zur Erlangung des akademischen Grades

Diplom-Ingenieur

Masterstudium Telematik

eingereicht an der

Technischen Universität Graz

Betreuer:

Mag. art. Martin Rumori

Institut für Kirchenmusik und Orgel

Universität für Musik und darstellende Kunst Graz

Begutachter:

Ao. Univ.-Prof. Dipl.-Ing. Dr. techn. Gerhard Graber

Signal Processing and Speech Communications Laboratory

Technische Universität Graz

Graz, Dezember 2015

Für all die Frauen, die mich liebten.

Romi und Laura.

Ich bedanke mich bei Gerhard Graber, Manfred Ifkovits, Margreth Pichler, Rosalia Maria Ročnik, Gunther Rost, Martin Rumori, Klaus Ziegler und Laura Charlotte Naomi Daniel für den inhaltlichen und seelischen Beistand den ich während meines Studiums erfahren durfte. Für jedes Glas Rotwein und jeden Moment.
Ohne euch wär's nur Telematik gewesen.

Zusammenfassung

Tastenteinstrumente haben einen diskreten, keinen kontinuierlichen Tonumfang. Durch die Einschränkung von 12 Tasten pro Oktave ist es nicht möglich in allen Tonarten die gleichen Frequenzverhältnisse der Intervalle einzuhalten. Heute wird von Tastenteinstrumenten eine Kompromisstimmung (Temperierung) verwendet, die den unausweichlichen Fehler gleichmäßig auf alle Töne verteilt. Diese Stimmung ist als gleichstufige Stimmung bekannt und bringt ebenso ihre Probleme mit sich. Diese zeigen sich beim Zusammenspiel mit frei intonierten Instrumenten, aber auch an der fortgesetzten Entwicklung von mikrotonalen Stimmungssystemen im 20. Jahrhundert. Es liegt nahe, die Intonation auch auf Tastenteinstrumenten zu einem freien musikalischen Gestaltungsparameter zu emanzipieren und dafür gegenwärtige technische Möglichkeiten zu nutzen. Am Institut für Kirchenmusik und Orgel existieren verschiedene digitale Orgelsysteme, unter anderem ein hoch integriertes System mit Spieltisch (Rodgers Masterpiece), aber auch Software-Orgeln, die über einen modularen MIDI-Spieltisch gesteuert werden können (Hauptwerk, GrandOrgue). Gegenüber Pfeifenorgeln haben sie den Vorteil, dass sich dynamische, mikrotonale Verstimmungen wesentlich leichter implementieren lassen.

In der Arbeit sollen Entwicklungsschritte zu frei und dynamisch intonierten Tastenteinstrumenten am Beispiel der digitalen Orgel unternommen werden. Im Fokus stehen dabei die Erschließung von Interfaces zur Stimmungssteuerung und Möglichkeiten der teilweisen oder vollständigen Automation von Intonationsanpassungen während des Spiels.

Abstract

Keyboard instruments provide a discrete, not a continuous tone range. Due to the restriction of 12 keys per octave it is not possible to fulfill the defined physical frequency ratios of thirds and fifths for all tonalities. Today, keyboard instruments implement a compromise tuning (temperament) which distributes the unavoidable error over all tones. This tuning is known as equal tuning and also comes with some problems. They come up when playing with instruments that have no tuning restriction, but also in the development of microtonal tuning systems in the 20th century. It stands to reason to emancipate the tuning of keyboard instruments to a dynamic musical design parameter by the use of today's technical possibilities.

The institute for church music and organ provides different digital organ systems, one of them a highly integrated system within an organ console (Rodgers Masterpiece), but also a software organ, which is controlled by a modular MIDI console (Hauptwerk, GrandOrgue). These instruments have the advantage that dynamic tuning and microtonal tunings are easier to implement.

The present thesis is thought to show the development of a free and dynamic tunable keyboard instrument, taking the digital organ as an example. The focus is held on the exploration of interfaces for controlling dynamic tuning and the possibilities of a partwise or full automated tuning during a performance.

Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe. Das in TUGRAZonline hochgeladene Textdokument ist mit der vorliegenden Masterarbeit identisch.

Graz, am

(Unterschrift)

Inhaltsverzeichnis

1	Einleitung	11
1.1	Motivation	11
1.2	Aufgaben	11
1.3	Begriffsklärung	12
2	Musikalische und psychoakustische Grundlagen	13
2.1	Musikalische Grundlagen	13
2.1.1	Musikalische Skalen	13
2.1.2	Musikalische Intervalle	13
2.1.3	Das Cent-Maß	15
2.2	Psychoakustische Grundlagen der Tonhöhenwahrnehmung	16
2.2.1	Tonhöhenwahrnehmung	16
2.2.2	Tonhöhenauflösung	19
2.2.3	Tonheit	19
2.2.4	Klangfarbe	20
2.2.5	Konsonanz und Dissonanz	21
2.2.6	Zusammenfassung	23
3	Stimmungssysteme	25
3.1	Probleme von reinen Intervallen	25
3.1.1	Das pythagoreische Komma	26
3.1.2	Das syntonische Komma	26
3.1.3	Die kleine und große Diësis	26
3.1.4	Zusammenfassung	27
3.2	Stimmung, Intonation, Temperierung	28
3.2.1	Intonation von Tasteninstrumenten	28
3.2.2	Reine Stimmung	31
3.2.3	Pythagoreische Stimmung	31
3.2.4	Mitteltönige Stimmungen	32
3.2.5	Silbermann-Stimmung	33
3.2.6	Werckmeister-Stimmung (wohltemperierte Stimmung)	34
3.2.7	Gleichstufige Stimmung	35
3.2.8	Neidhardt-Stimmungen	36
3.2.9	Vergleich der Abweichung von der reinen Stimmung	37
3.2.10	Enharmonische Verwechslung	39
3.2.11	Lösungsversuche auf der Basis von Tasteninstrumenten	39
3.2.12	Mikrotonale Musik	42
3.2.13	Zusammenfassung	42
3.3	Inharmonizität	44
3.3.1	Auswirkungen der Inharmonizität auf reine Intervalle	45
3.3.2	Streckung der Temperierung	45
3.3.3	Zusammenfassung	45
3.4	Adaptives Tuning	46
3.4.1	Ansätze für adaptives Tuning	46
3.4.2	Hermode-Tuning	47
3.4.3	Spring-Tuning	48
3.4.4	Zusammenfassung	49

4	Dynamische Intonation digitaler Konzertorgeln	51
4.1	Laborsystem	52
4.1.1	Die digitalen Orgeln am Institut für Kirchenmusik und Orgel	52
4.1.2	Nicht verfügbare Systeme	54
4.1.3	Beschreibung des Laborsystems	55
4.2	DynTune-Software	58
4.2.1	Benutzeroberfläche	58
4.2.2	Konfigurationsdatei	61
4.3	Inbetriebnahme der Software	63
4.3.1	Steuerung über externe Interfaces	64
4.4	Technische Implementierung	64
4.4.1	Programmablauf	67
5	Epilog	69
5.1	Ergebnisse erster Anwendung	69
5.2	Diskussion	70
5.2.1	Approximationsfunktion	70
5.2.2	Aktualisierungsrate	72
5.2.3	Normalton	72
5.2.4	Ideale Stimmung	72
5.3	Zusammenfassung	73
5.4	Ausblick	73
A	Klangbeispiele	75
B	Hauptwerk XML-Format	77
C	DynTune-Dokumentation	79
C.1	DynTuneGUI	79
C.1.1	Static Class Members	79
C.1.2	Class Variables	79
C.1.3	Class Functions	80
C.2	IntervallPlotter	83
C.2.1	Static Class Members	83
C.2.2	Class Variables	83
C.2.3	Class Functions	83
C.3	HWXml	84
C.3.1	Static Class Members	84
C.3.2	Class Variables	84
C.3.3	Class Functions	84
C.4	DynTune	85
C.4.1	Static Class Members	85
C.4.2	Class Variables	85
C.4.3	Class Functions	86
C.5	DynTuneHWUpdate	92
C.5.1	Static Class Members	92
C.5.2	Class Variables	92
C.5.3	Class Functions	92
C.6	DynTuneMidiReceiver	93
C.6.1	Static Class Members	93
C.6.2	Class Variables	93
C.6.3	Class Functions	94

1

Einleitung

*"Now I've heard there was a secret chord
That David played, and it pleased the Lord
But you don't really care for music, do you?
It goes like this, the fourth, the fifth
The minor falls, the major lifts . . ."*
– Leonard Cohen, *Hallelujah*

1.1 Motivation

Das Telematikstudiums als wählbare Brücke zwischen Technischer Universität und Universität für Musik und darstellende Kunst bietet eine Basis zur Durchführung von interuniversitären Masterarbeiten. Die Orgelforschung sowie die neuen digitalen Konzertorgeln am Institut für Kirchenmusik und Orgel eröffnen Möglichkeiten, sich auf elektrotechnischer und akustischer Ebene mit dem Instrument Orgel zu befassen. Ein Forschungsgegenstand dabei ist, die einzelnen Töne des Instruments während des Spiels frei intonierbar zu gestalten. Ausgangspunkte für die freie Intonation sind zum einen unterschiedliche Tonartencharakteristiken historischer Stimmungen sowie die Beobachtung, dass Streichquartette oder Chöre die dynamische Intonation als musikalischen Gestaltungsparameter einsetzen. Die technische Herausforderung ist die Entwicklung eines interaktiven Systems, das nach seiner Fertigstellung im Konzert- und Forschungsbetrieb Nutzung findet.

1.2 Aufgaben

- (i) Recherche zu bisherigen Versuchen der dynamischen Intonation von Tasteninstrumenten.
- (ii) Einrichtung eines frei intonierbaren Laborsystems auf Basis der Institutsorgeln.
 - Erschließung der vorhandenen elektronischen Orgeln hinsichtlich dynamischer Verstimmbarkeit
 - Abbildung einer dynamisch veränderbaren Stimmungstabelle auf der Basis einer statischen Grundstimmung
 - Implementierung einer Schnittstelle zwischen Stimmungstabelle und Orgel
- (iii) Möglichkeiten von Interfaces zur (händischen) dynamischen Intonation.
 - Eindimensionaler kontinuierlicher Stimmungsparameter zwischen statischen Grundstimmungen, gesteuert durch freies Schweller-Pedal
 - Stimmung der Einzeltöne über Anschlagsdynamik oder Aftertouch (Tastendruck auf der angeschlagenen Taste)
- (iv) Ausblick auf Automatisierung der Intonation (Beispiel: Hermode-Tuning) und Strategien für alternative Stimmungsnachführungen.

1.3 Begriffsklärung

Nachfolgend werden ausgewählte Begriffe erklärt. Einigen dieser liegen verschiedene Bedeutungen zu Grunde, sie werden daher hier für die Arbeit festgelegt.

Stimmung Als Stimmung wird, bei Instrumenten, die Festlegung der Tonhöhe bezeichnet.

Intonation Intonation bezeichnet das Anpassen einer Summe von Parametern wie Tonhöhe, Lautstärke und Klangfarbe einzelner Töne. Bei Instrumenten wie Klavier, Orgel, u a. wird die Intonation vor dem Spiel vorgenommen. Dabei werden Tonhöhe und Klangfarbe eingestellt. Streich- oder Blasinstrumente werden auch vor dem Gebrauch gestimmt, Intonation bezeichnet hier aber die vom Musiker selbst während des Spiels vorgenommene Festlegung der Parameter wie Lautstärke, Klangfarbe und Tonhöhe.

Temperierung Vor allem Tasteninstrumente verlangen eine statische Zuordnung von Tonhöhe zu Taste, die in der Regel während des Spiels nicht mehr geändert werden kann¹. Diese Festlegung wird als Temperierung bezeichnet.

¹ Methoden der dynamischen Umstimmung sind in Abschnitt 3.4 der Arbeit erklärt.

2

Musikalische und psychoakustische Grundlagen

2.1 Musikalische Grundlagen

2.1.1 Musikalische Skalen

Skalen teilen den Tonumfang (eines Instruments, des menschlichen Gehörs, ...) in Teile. Definierte Töne und Tonverhältnisse finden sich nicht nur in der Sprache wieder, in der Musik hat sich eines der Tonsysteme, das das Komponieren, Wiedergeben und gemeinsame Musizieren vereinfacht, herausgebildet. Heute wird in der westlichen (abendländischen) Musik ein 12-stufiges, *chromatisches* Tonsystem verwendet, mithilfe welchem sich auf jeder Tonstufe eine 8-stufige *diatonische* Tonleiter aufbauen lässt. Das chromatische Tonsystem unterscheidet sich von Systemen, die z. B. Mönche im Mittelalter verwendeten, aber auch heute noch sind in anderen Kulturen andere Tonsysteme sowie Skalen in Gebrauch. Dazu gehören der Gamelan auf Java oder der Qawwali im Sufismus².

Von allen Möglichkeiten Tonverhältnisse festzulegen, also musikalische Skalen zu definieren, werden in der westlichen Musik nur wenige verwendet. Das Wissen um den Aufbau der chromatischen Tonleiter sei für das Verständnis der nachfolgenden Ausführungen vorausgesetzt. Im weiteren soll aber auf die gebräuchlichen musikalischen Intervalle näher eingegangen werden.

2.1.2 Musikalische Intervalle

Musikalische Intervalle beschreiben den Abstand von zwei Tönen. „Ein Ton entsteht durch periodische Schwingungen eines elastischen Körpers“ [1, S. 47]. Dieser Körper kann z. B. eine Saite eines Instruments sein. Teilt man eine Saite in zwei gleich große Teile, so erklingt jeder dieser Teile im Verhältnis einer Oktave zum Ton der ganzen Saite. Die Frequenz, mit der die halbe Saite schwingt, entspricht der doppelten der ganzen Saite. Saitenlänge und Frequenz verhalten sich verkehrt proportional. Die Oktave als eines der bekanntesten Intervalle entspricht demnach einem Frequenzverhältnis von genau 1:2. Die Oktave wird in die uns bekannte Tonleiter unterteilt. Würde man eine Saite in drei gleich große Teile teilen, würden $\frac{2}{3}$ der Saite das Quinten-Intervall bilden.

Musikalisch versteht man unter *Ton* nicht nur das Ergebnis einzelner periodischer Schwingungen, sondern einen *Klang* als Summe einzelner Töne. Diese einzelnen Töne werden *Obertöne*, *Teiltöne* oder auch *Naturtöne* genannt. In Abbildung 2.1 sind diese Obertöne für den Grundton C dargestellt.

Diese Obertonreihe enthält (bis auf die Töne 7, 11, 13, 14) alle Töne der C-Dur-Tonleiter: Daraus lassen sich auch die Frequenzverhältnisse einzelner Intervalle ablesen [1].

Die Intervalle beschreiben also das Tonhöhenverhältnis des Oktaven-Grundtons zu den einzelnen Stufen der Tonleiter. Das kleinste dieser Intervalle ist die Prime, welche auch als *Einklang*

² Sammelbegriff für stark spirituell orientierte Strömungen des Islam.

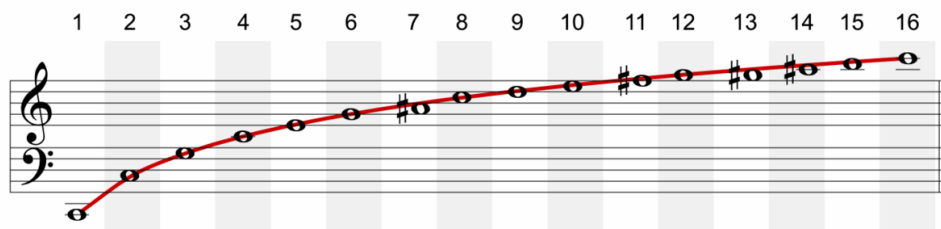


Abbildung 2.1: Obertonreihe für C.
 Bildnachweis: dbenzhuser
 Wikimedia Commons, lizenziert unter CreativeCommons-Lizenz by-sa-2.0-de

Intervall	Frequenzverhältnis
Prime	$1/1$
Oktave	$1/2$
Quinte	$2/3$
große Terz	$4/5$

Tabelle 2.1: Frequenzverhältnisse ausgewählter Intervalle und deren Bezeichnung. Nach [1, S. 49].

bezeichnet wird, sie beschreibt das Verhältnis zur ersten Stufe mit dem Frequenzverhältnis 1:1. Die Namensgebung der Intervalle folgt den lateinischen Ordnungszahlen. Prime (lat. Prima, die Erste), Sekund (lat. Secunda, die Zweite), usw.

Tonstufe	C	D	E	F	G	A	H	c
Intervall	Prime	gr. Sekunde	gr. Terz	Quarte	Quinte	gr. Sexte	gr. Septime	Oktave

Tabelle 2.2: Intervalle der C-Dur-Tonleiter.

Tabelle 2.2 zeigt die C-Dur-Tonleiter mit den Tonstufen zugehörigen Intervallen. Mithilfe der Intervalle wird jedem Ton eine eindeutiges Frequenz (im Verhältnis zum Grundton) zugeordnet. Terz, Quinte und Oktave werden als Grundintervalle angesehen. Sie folgen den Ordnungszahlen: 3, 5 und 8. Das Notensystem ist so aufgebaut, dass alle Töne durch Aneinanderreihung von Intervallen erreicht werden können, z. B. ergeben große Terz und kleine Terz eine Quinte.

$$\frac{5}{4} \cdot \frac{6}{5} = \frac{30}{20} = \frac{3}{2} \tag{2.1}$$

Pro Oktave liegt eine Verdoppelung der Frequenz vor. Die Frequenz wächst also exponentiell mit der Größe des Intervalls. Dies wird auch in Abbildung 2.1 deutlich. Das Frequenzverhältnis von k Oktaven beträgt 2^k . Weiters lässt sich sagen, dass das Intervall q $\text{ld}(q)$ Oktaven entspricht. Eine Quinte entspricht 0,585 Oktaven (vgl. Gl. 2.2). Es ist also möglich, jedes Intervall als Vielfaches einer Oktave zu beschreiben.

$$\text{ld}(q) = \text{ld}\left(\frac{3}{2}\right) = 0,585 \text{ Oktaven} \tag{2.2}$$

2.1.3 Das Cent-Maß

Um die Berechnung von Frequenzverhältnissen zu erleichtern, wurde 1875 von Alexander Ellis die Maßeinheit Cent vorgeschlagen; diese dient heute zur Beschreibung der additiven Struktur von Intervallen.

Eine Oktave entspricht 12 Halbtonschritten. Die Größe einer Oktave wurde daher auf 1200 Cent festgelegt. Durch diese Wahl ist es möglich die Tonhöhe mit dem Cent-Maß hinreichend genau zu beschreiben. Hinreichend bedeutet, dass erst Abweichungen von mehr als 3 Cent für den Menschen hörbar werden (vergleiche dazu Abschnitt 2.2.2 auf Seite 19.)

$$1 \text{ Cent} = \sqrt[1200]{2} = 1,00057779 \quad (2.3)$$

Der Abstand zwischen zwei Tönen eines Intervalls kann also in Cent angegeben werden. Entspricht das Intervall einem Frequenzverhältnis q , berechnet sich der Abstand d der beiden Töne in Cent nach Gleichung 2.4.

$$d = 1200 \cdot \text{ld}(q) \text{ [Cent]} \quad (2.4)$$

Die umgekehrte Berechnung des Frequenzverhältnisses erfolgt nach Gleichung 2.5. Besitzt ein Intervall die Größe d , so beträgt sein Frequenzverhältnis den Faktor q .

$$q = 2^{\frac{d}{1200}} \quad (2.5)$$

Tabelle 2.3 erweitert Tabelle 2.1 von S. 14 um die Größen der Intervalle in Cent.

Intervall	Frequenzverhältnis	Größe in Cent
Oktave	$1/2$	1200,000
Quinte	$2/3$	701,995
Quarte	$3/4$	498,045
große Terz	$4/5$	386,314
kleine Terz	$5/6$	315,641

Tabelle 2.3: Abstand ausgewählter Intervalle in Cent.

Die Brechung der Größe von höheren Intervallen kann nun als Addition erfolgen. Kleine Terz plus große Terz berechnet sich als 316 Cent + 368 Cent = 702 Cent = 1 Quinte.

2.2 Psychoakustische Grundlagen der Tonhöhenwahrnehmung

Die Psychoakustik befasst sich mit den Zusammenhängen zwischen dem physikalischen Schallfeld und der menschlichen Empfindung. Die Verarbeitung vom Schallereignis zum Höreindruck wird von ihr beschrieben und erforscht. Für diese Arbeit ist insbesondere das Teilgebiet der Tonhöhenwahrnehmung interessant.

Grob kann ein wahrgenommener Ton in verschiedene Bestandteile zerlegt werden: die Intensität, also die Lautstärke, und die Frequenz, mit der die Luftteilchen schwingen, und die im Zusammenhang mit der Tonhöhe steht. Der Frequenzumfang, der von einem erwachsenen Menschen wahrgenommen werden kann, erstreckt sich dabei von 20 Hz bis 20 kHz, die obere Grenze verschiebt sich mit dem Alter, aber auch durch Beanspruchung allerdings nach unten. Zum Vergleich dazu reicht der Frequenzumfang der menschlichen Stimme (ungefähr) von 80 Hz bis 12 kHz, wobei der Grundton der männlichen Stimme bei etwa 125 Hz, der der weiblichen Stimme etwa bei 250 Hz liegt. Der im wesentlichen für Musik relevante Bereich erstreckt sich von 40 Hz bis 10 kHz.

Bei der Wahrnehmung von Klangereignissen spielen mehrere Komponenten des menschlichen Körpers eine Rolle: so werden Vibrationen und Druck (als Teile einer Schallwelle) über den gesamten Körper aufgenommen, zentrales Element ist aber das Ohr. Aufbau und Funktion des Ohres können anhand dreier Bereiche beschrieben werden:

Außenohr Das Außenohr besteht aus der Ohrmuschel, welche die weiteren Bereiche des Ohres vor Verschmutzungen schützt und durch ihre Form einen Teil zum Richtungshören beiträgt. Schallwellen werden über den Gehörgang an das Trommelfell weitergeleitet.

Mittelohr Im Mittelohr befindet sich das Trommelfell sowie die drei Gehörknöchelchen: Hammer, Amboss und Steigbügel. Diese drei Knochen bilden ein an das Trommelfell angeschlossenes mechanisches System, welches eine Impedanzwandlung vollführt. Große Amplituden mit wenig Druck (am Trommelfell) werden in eine kleine Auslenkung mit großem Druck umgewandelt und an das Ovale Fenster weiterleitet.

Innenohr Hinter dem Ovalen Fenster sitzt die Cochlea oder auch Gehörschnecke. Die Schnecke ist gefüllt mit einer Flüssigkeit, die schwerer in Schwingung zu versetzen ist als Luft und die vorhergehende Impedanzwandlung im Mittelohr notwendig macht. In der Cochlea befindet sich die Basilarmembran mit dem Cortischen Organ, hier erfolgt letztendlich die Wandlung von Druck in Nervenimpulse durch kleine Härchen, die in Reihen angeordnet sind. Die Auswertung geschieht dabei nach Frequenzen, je nachdem an welcher Stelle der Basilarmembran sich das Erregungsmaximum eines Signals befindet kann dessen Frequenz bestimmt werden.

Ebenfalls relevant für die Wahrnehmung von Audiosignalen sind Signale, die wir über andere Sinnesorgane aufnehmen, z. B. ob und wie wir die Schallquelle sehen oder sogar welche Farbe sie besitzt [2]. Aus all den Informationen setzt sich im Gehirn ein Klangeindruck zusammen.

2.2.1 Tonhöhenwahrnehmung

Nach dem ANSI³ ist die Tonhöhe der Teil der Hörempfindung, der es ermöglicht Töne auf einer musikalischen Tonleiter zu ordnen (ANSI S1.1). Intuitiv fällt es leicht ein Verständnis für Tonhöhe zu entwickeln. Jeder Mensch ist in der Lage Töne miteinander zu vergleichen und auch nach ihrer Tonhöhe zu sortieren. Dabei kommen verschiedene Menschen aber nicht exakt zum selben Ergebnis und umso schwieriger fällt es zu definieren, nach welchen Kriterien die Tonhöhe vom Menschen beurteilt wird.

³ American National Standards Institute

Der Klang eines musikalischen Tones, also einer gesungenen oder gespielten Note, setzt sich aus einem Grundton und mehreren Obertönen (Teiltönen) zusammen. Obertöne sind zumeist Bestandteile, die durch Eigenschwingungen des Instruments (Klangkörper, Saiten, usw.) entstehen, und ganzzahlige Vielfache des Grundtons. Man spricht in diesem Fall auch von harmonischen Obertönen. Näherungsweise kann vorerst die Aussage getroffen werden, dass die Tonhöhe vom Grundton abhängig ist. Besteht ein Klang aus harmonischen Obertönen, ist die Tonhöhe genau die Tonhöhe des Grundtones [3]. Diese Definition verliert an Gültigkeit, sobald der Klang nicht mehr aus harmonischen Komponenten besteht, es sich also um ein so genanntes komplexes Signal handelt. Allerdings sind die meisten Klänge in Sprache und Musik harmonisch. Wie bereits in Abschnitt 2.1.2 erwähnt bilden die Obertöne einen Zusammenhang zur s. g. Obertonreihe, einer Skala, die nach Intervallen (Oktave, Quinte, ...) geordnet ist.

Theorien zur Tonhöhenwahrnehmung

Einer der ersten, die sich mit der Wahrnehmung von Tönen beschäftigten, war Pythagoras von Samos. (Auf seinen Erkenntnissen baute später auch der Pythagoreer Philolaos auf, vgl. Abschnitt 3.1.1). Er stellte bereits im Jahr 600 vor Christus fest, dass sich die Tonhöhe einer schwingenden Saite verdoppelt, also um eine Oktave erhöht, wenn er die Saite halbiert. Er folgerte daraus, dass sich die Frequenz, mit der die Saite schwingt verdoppelt. Später (1640) deklarierte Galileo Galilei ebenfalls, dass die Tonhöhe mit der Frequenz, also der Anzahl der Schwingungen pro Sekunde zusammenhängt.

Das menschliche Gehör nutzt die Gesamtheit eines Klanges, also auch die Obertöne um auf die Tonhöhe zurückzuschließen, dabei muss die Grundfrequenz im Klang selbst nicht vorhanden sein. Der deutsche Physiker Thomas Johann Seebeck beschäftigte sich um 1841 mit Experimenten rund um das Fehlen des Grundtons und der Tatsache, dass die Tonhöhe nicht von der Periodendauer des Grundtones abhängt.

Seebeck benutzte in seinen Experimenten eine Lochsirene mit drei Signalen. Für das erste Signal waren die Löcher in regelmäßigen Abständen angeordnet. Die Tonhöhe des resultierenden Schallereignisses entspricht der Frequenz $\frac{1}{T}$. Für das zweite Signal waren die Löcher ebenfalls in regelmäßigen Abständen angeordnet, nur enthielt sie doppelt so viele Löcher, was zu einer Tonhöhe mit der Frequenz $\frac{2}{T}$ führte, also einer Oktave höher. Für das dritte Signal verschob Seebeck jedes zweite Loch des zweiten Signals. Das dritte Signal war also gegeben durch wenig Energie bei der Frequenz $\frac{1}{T}$ und viel Energie bei der Frequenz $\frac{2}{T}$. Abbildung 2.2 zeigt die drei von Seebeck verwendeten Signale.

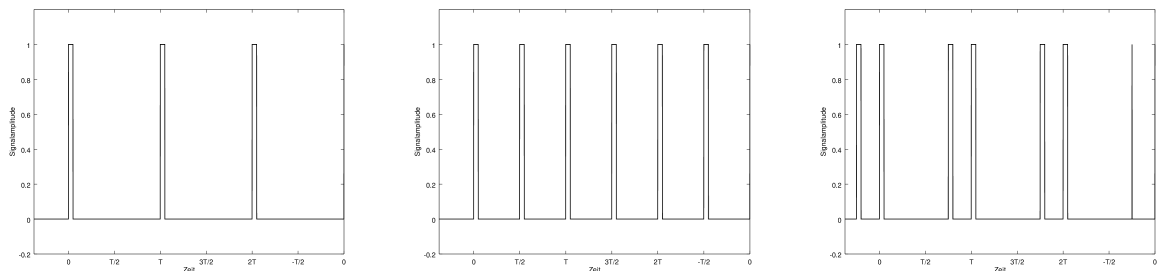


Abbildung 2.2: Die drei von Seebeck verwendeten Signale, nach [4].

Seebeck vermutete, dass Signal zwei und drei durch die gleiche Lochzahl auch die gleiche Tonhöhe besäßen. In Wirklichkeit besitzen aber Signal eins und drei die gleiche Tonhöhe. Er vermutete, dass es sich dabei um eine Art akustische Täuschung handeln müsste [5].

Nach Seebeck untersuchte auch Hermann von Helmholtz Mitte des 19. Jahrhunderts die Tonhöhenwahrnehmung. Er ging davon aus, dass die Basilarmembran aus einer Reihe gespannter Saiten besteht, ähnlich einer Harfe, die unterschiedliche Resonanzfrequenzen besitzt. Bringt eine

komplexe Schallwelle diese Saiten zum Schwingen, so werden nur Saiten mit gleicher Resonanzfrequenz angeregt. Es fände also eine Art Fourier-Analyse der Schallwelle statt. Helmholtz ging weiters davon aus, dass die Impedanzwandlung im Mittelohr ein nichtlinearer Prozess ist. Bei der Wandlung würden, laut ihm, nichtlineare Verzerrungen auftreten, die in der Cochlea weiterverarbeitet werden. Er hielt fest, dass diese nichtlinearen Verzerrungen für komplexe Töne Störfrequenzen auftreten lassen, die der Differenz nebeneinander liegender Frequenzen entsprechen. Bei harmonischen Tönen hingegen entstünden Störfrequenzen, die den harmonischen Obertönen entsprechen und im weiteren also nicht auffallen würden. Die Störfrequenzen oder Kombinations-töne bezeichnete er auch als „quadratische Komponenten nichtlinearer Verzerrung“ [6]. In den Experimenten von Seebeck könne also die Energie, die bei der Frequenz $\frac{2}{T}$ zustande kommt, zur Energie der Frequenz $\frac{1}{T}$ addiert werden. Die Grundfrequenz wäre also dominierend. Die Theorie von Seebeck, die von Helmholtz weiterentwickelt wurde, ist heute als *Ortstheorie* bekannt.

Für das Phänomen, dass die Tonhöhe auch ausgewertet werden kann, selbst wenn die Grundfrequenz im Signal nicht vorhanden ist, kann als Beispiel das Telephon angeführt werden. Die Bandbreite die beim Telefon übertragen wird beginnt bei 300 Hz. Obwohl die menschliche Stimme, wie eingangs erwähnt, Grundfrequenzen von 125 Hz bzw. 250 Hz besitzt, ist es möglich diese in der originalen Tonhöhe wahrzunehmen, ohne dass bemerkt wird, dass die eigentliche Grundfrequenz im Signal nicht vorhanden ist.

Aktueller Stand der Forschung

Die Forschungen von Seebeck, Helmholtz und Nachfolgern gingen stets davon aus, dass die Verarbeitung des Schallereignisses nur im (jeweiligen) Ohr passieren würde. Houtsma und Goldstein konnten dies 1972 am MIT⁴ widerlegen. In ihren Experimenten wurde Testpersonen ein aus zwei Komponenten bestehendes Signal vorgespielt: jeweils eine Komponente des Signals, an einem Ohr. Die beiden Komponenten bestanden dabei aus je einem einfachen Sinus-Signal beliebiger Frequenz. Dabei stellten sie fest, dass die von den Testpersonen wahrgenommene Tonhöhe aus der Differenz der beiden Komponenten bestand. Houtsma und Goldstein schlussfolgerten, dass die Analyse zur Tonhöhenwahrnehmung nicht im Ohr stattfinden kann, sondern an einem zentralen Ort erfolgen muss. Dies widerlegt die von Helmholtz aufgestellte Theorie, dass Verzerrungen und Störfrequenzen für die Wahrnehmung der Tonhöhe verantwortlich sind [7].

Richtig ist jedoch, dass es im Ohr zu nichtlinearen Verzerrungen kommt. Die von Helmholtz beschriebenen quadratischen Komponenten nichtlinearer Verzerrung sind allerdings unauffällig [6]. Von Relevanz sind hingegen kubische Verzerrungen. Die Frequenz des auftretenden Störtones von zwei Komponenten f_1 und f_2 berechnet sich aus $2 \cdot f_1 - f_2$. Diese resultierende Störfrequenz tritt aber lediglich 20 dB bis 30 dB unter den beiden Komponenten auf. Trotzdem beeinflussen diese kubischen Verzerrungen die Auswertung der Frequenzanteile im Ohr und dadurch die Analyse der Tonhöhe.

Moderne Theorien

Der Großteil der aktuellen Theorien wie der von Houtsma und Goldstein geht davon aus, dass das Erkennen der Tonhöhe ein zweistufiger Prozess ist. In diesem Prozess wird auf die Grundtonhöhe aus den im Schallfeld enthaltenen Komponenten geschlossen. Im ersten Schritt findet, ähnlich wie von Helmholtz beschrieben, eine Auswertung der einzelnen im Signal vorkommende Frequenzen statt. In einem zweiten Schritt erfolgt die Verarbeitung an zentraler Stelle. Es wird davon ausgegangen, dass dabei Muster abgeglichen werden und die Tonhöhe aufgrund bereits bekannter Tonhöhen erkannt wird. Aufgrund der alleinigen Auswertung der in den Signalen vorhandenen Frequenzen werden diese modernen Theorien *spektrale Theorien* genannt [3].

⁴ Massachusetts Institute of Technology

2.2.2 Tonhöhenauflösung

Die Auflösung der wahrgenommenen Tonhöhe wird durch zwei Faktoren beschränkt, erstens der Dichte an Haarzellen auf der Basilarmembran, zweitens der Unmöglichkeit des Gehirns pro Haarzelle eine Auswertung vornehmen zu können. Die hörbare Bandbreite des Menschen kann in zwei Gruppen eingeteilt werden (nach [8]):

Frequenzen bis 500 Hz Im Bereich von Frequenzen im unteren Bereich des Frequenzspektrums entspricht die Länge der Haarzellenreihen weniger als 1 mm pro Oktave, der Mensch ist in der Lage etwa 270 Tonhöhen im Abstand von 1,8 Hz zu unterscheiden. Abgesehen von der Auswertung des Erregungsmaximums kann das Ohr in diesem Bereich auch die Periodendauer des Signals erfassen. Die Auswertung ist somit genauer.

Frequenzen ab 500 Hz Im Bereich höherer Frequenzen beträgt die Länge der Haarzellenreihen ca. 6 mm pro Oktave. Bis 15 kHz können insgesamt etwa 520 Tonhöhen unterschieden werden. Dies entspricht in Bereich von 500 Hz bis 15 kHz 70 Tonhöhen pro Oktave. Die Intervallgröße entspricht ungefähr 3 Cent.

2.2.3 Tonheit

Für die wahrgenommene Tonhöhe von harmonischen Tönen wurde die psychoakustische Größe der Tonheit (Formelzeichen Z) gewählt, mit ihrer Einheit *Mel*. Die Skala wurde 1937 von Stanley Smith Stevens begründet, der 1000 Hz mit einem Wert von 1000 Mel festlegte. Eberhard Zwicker skalierte später (um 1982) auf den Ton C mit 131 Hz, um mit der Bark-Skala⁵ (einer weiteren Größe für die Tonheit) kompatibel zu werden, später wurden aber 125 Hz als Basis gewählt. Näherungsweise kann die Mel-Skala mit Gleichung 2.6 beschrieben werden.

$$Z = 2595 \cdot \log\left(1 + \frac{f}{700}\right) \quad (2.6)$$

Abbildung 2.3 zeigt den Zusammenhang zwischen Tonheit in Mel und Frequenz in Herz. Für Frequenzen bis 500 Hz existiert ein linearer Zusammenhang. Eine Verdoppelung der Frequenz wird als eine Verdoppelung der Tonhöhe empfunden. Ab Frequenzen von 500 Hz ist der Zusammenhang logarithmisch, Tonhöhen werden kleiner wahrgenommen als die tatsächliche Frequenzänderung. Dies hängt vor allem mit der Art der Auswertung zusammen (vgl. dazu Abschnitt 2.2.2). In diesem Bereich ist also ein musikalisches Intervall von mehr als zwei Oktaven notwendig, um eine Verdoppelung der empfundenen Tonhöhe zu erreichen [8].

(Anm.: Die Bestimmung der Mel-Skala geht auf psychoakustische Versuchsreihen zurück. An solchen Versuchsreihen nehmen nach sorgfältiger Auswahl sowohl Musiker als auch Laien teil. Das Gehör eines Musikers kann so geschult sein, dass er die Nichtlinearität der Mel-Skala korrigiert.)

Insgesamt existieren 620 unterscheidbare Tonhöhen, die in Schritten von 3,9 Mel unterschieden werden. Der Hörbereich umfasst 2400 Mel.

⁵ 1,31 Bark = 131 Mel = 131 Hz

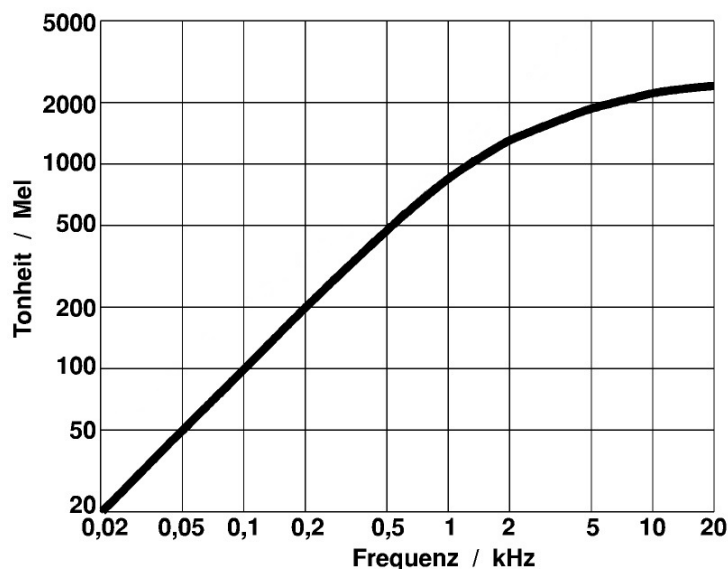


Abbildung 2.3: Zusammenhang zwischen Frequenz und Tonheit nach [8].

2.2.4 Klangfarbe

Auch wenn die einzelnen Töne eines Instruments als eine einzelne Frequenz definiert werden, versteht man darunter mehrere Bestandteile. Der Klang eines einzelnen Tones setzt sich aus einem Grundton, mehreren Obertönen, Rauschen, dem zeitlichen Verlauf des Spektrums u. v. m. zusammen. Diese Summe von Parametern wird als *Klangfarbe* oder *Timbre* bezeichnet. Wenn vom (musikalischen) Ton als einer einzelnen Tonstufe gesprochen wird, ist meistens der vom Instrument geformte Klang dieser Tonstufe gemeint, also nicht der physikalische Ton.

Resonanzen eines Instruments, die sich durch seine Bauform ergeben, aber auch Resonanzen im menschlichen Sprachtrakt sind prägend für den sich ergebenden Klangeindruck. Die stärksten von ihnen werden als Formanten bezeichnet, sie sind charakteristisch für das Obertonspektrum eines Klanges. Formanten sind ein Grund, warum verschiedene Instrumente bei gleichem Grundton unterschiedlich erklingen.

Rauschanteile steuern ebenfalls ihren Teil zum Klangeindruck bei, bei Blasinstrumenten oder Orgeln sind Anblasgeräusche prägend für den Klang eines Instrumentes. Aber auch das Heulen des Windes oder Meeresrauschen hat seine charakteristische Klangfarbe.

Der Klangeindruck entsteht letztendlich im Gehirn als Auswertung verschiedener Parameter. Dieser Klangeindruck lässt sich nicht als einzelnes Kontinuum beschreiben, sondern ist ein mehrdimensionaler Vektor.

kalt	⇔	warm
stumpf	⇔	scharf
weich	⇔	hart
kompakt	⇔	vielschichtig
statisch	⇔	dynamisch
farbig	⇔	schwarz-weiß

Tabelle 2.4: Beispiel eines Mehrdimensionalen Vektors zur Beschreibung des Klangeindrucks, nach [9].

Wie viele Dimensionen genau nötig sind, um den Klangeindruck zu beschreiben, ist dabei Thema anhaltender Diskussionen [9].

Die hier beispielhaft aufgeführten Parameter sind keine physikalischen Größen. In einer Reihe von Forschungen [9, S. 29] wurden Testpersonen damit beauftragt die Klangfarbe eines Klangbeispiels multidimensional zu beschreiben.

Aus dieser Beschreibung errechnete ein Algorithmus den Bezug der einzelnen Parameter zu physikalischen Größen mit dem Ergebnis, dass die Klangfarbe folgende Klangattribute umfasst:

- (i) Synchronisation im Ein- und Ausschwing-Vorgang der Klangfragmente
- (ii) Grad der spektralen Fluktuation
- (iii) Präsenz von hochfrequenter, unharmonischer Energie im Einschwingvorgang
- (iv) Bandbreite des Signals
- (v) Ausgeglichenheit der Energie in hohen und tiefen Komponenten des Klangspektrums

Dies stellt aber nur einen Versuch dar die Klangfarbe zu beschreiben. Andere Theorien beschäftigen sich mit Analogien zu Selbstlauten oder pragmatischer Synthese. Im weiteren Verlauf dieser Arbeit wird die Klangfarbe, vor dem Hintergrund, dass auf die Entwicklung einer Orgelspielhilfe Bezug genommen wird, vor allem über den emotionalen Eindruck beschrieben werden.

2.2.5 Konsonanz und Dissonanz

Das westliche Tonsystem begründet sich wesentlich auf dem Prinzip von Konsonanz und Dissonanz. Während die Prime als einziges Intervall unbestritten als Konsonant bezeichnet werden kann (es erklingt nur ein Ton), kann bei den weiteren Intervallen keine klare Abgrenzung getroffen werden.

Konsonanz und Dissonanz wurde über die Jahrhunderte immer wieder neu beurteilt und eine Abgrenzung ist bis heute nicht gänzlich möglich. Grundlegend kann die Aussage getroffen werden: Analysiert man den Gesamtklang von zwei Tönen eines Intervalls, würde festgestellt werden, dass bei Intervallen kleiner Ordnung (vgl. Abschnitt 2.1.2) gemeinsame Obertöne auftreten, mit steigender Ordnung des Intervalls aber immer mehr unterschiedliche Obertöne erklingen – man spricht allgemein von Dissonanz [10]. Die Oktav als Intervall niedriger Ordnung (Frequenzverhältnis $1/2$) wird als besonders ruhig und konsonant empfunden. Dies ist aber nur eine der gebräuchlichen Definitionen⁶.

James Tenney fasste 1934 fünf Konzepte (Consonance and Dissonance Concepts) zusammen, die Konsonanz und Dissonanz beschreiben: [9], [11].

Melodic Consonance (CDC-1) Die früheste Definition, die in Tenneys Arbeit auftaucht, ist eine melodische Konsonanz. Melodische Intervalle sind konsonant oder dissonant, je nachdem, in welchem musikalischen Kontext sie in einem Werk vorkommen.

Polyphonic Consonance (CDC-1) Mit dem Aufkommen der Mehrstimmigkeit ging man in der Definition der Konsonanz von einer rein melodischen Betrachtung dazu über, Intervalle von gleichzeitig erklingenden Tönen als ganzes zu untersuchen. Besonders zwischen dem 13. und 15. Jahrhundert folgten eine Reihe von Abstraktionen dieser Theorie. Dazu zählt auch die Definition, dass besonders Intervalle mit niedriger Ordnung (Prime, Oktave, Quinte, Quart) konsonant sind und die Dissonanz mit der Ordnung des Intervalls ansteigt.

Contrapuntal Consonance (CDC-3) Zum 16. Jahrhundert hin wurde Konsonanz vor allem nach ihrer Rolle im Kontrapunkt definiert. Dabei galten Quinte und Quarte als perfekt konsonant und kleine, große Terz sowie die Sexte als unvollkommen konsonant.

Functional Consonance (CDC-4) Im 17. Jahrhundert führte Jean-Philippe Rameau die Konsonanz eines Tones auf sein Verhältnis zu einer „Wurzel“ bzw. zum Grundton zurück. Konsonante Töne seien solche, die in einem einfachen Verhältnis zum Grundton stünden, dissonante Töne besäßen ein komplexes Verhältnis zu diesem. Dabei wird als einfaches Verhältnis, jenes bezeichnet welches sich aus Quinten und Terzen zusammensetzen lässt.

⁶ In der vorliegenden Arbeit wird, soweit nicht anders erwähnt, auf diese Bezug genommen.

Psychoacoustic Consonance (CDC-5) Die psychoakustische Konsonanz (auch sensorische Dissonanz) basiert auf den Wahrnehmungseigenschaften des menschlichen Gehörs und beschreibt das aktuellste Konzept von Konsonanz und Dissonanz, welches maßgeblich auf den Forschungen von Hermann Helmholtz (vgl. Abschnitt 2.2.1) beruht, aber auch auf der Arbeit von Plomb und Levelt Mitte des 20. Jahrhunderts in den Niederlanden. Darauf aufbauend ergeben sich daraus drei Schlussfolgerungen:

- (i) Einzelne komplexe Töne sind grundsätzlich dissonant. Im Unterschied zu den anderen CDCs wird hier das Verhältnis von Konsonanz und Dissonanz nicht auf das Verhältnis der Töne untereinander zurückgeführt, sondern unter dem Blickwinkel der *Rauheit* betrachtet: Jeder Ton mit mehr als einem Teilton besitzt Dissonanz, die durch das Zusammenspiel der Teiltöne auftritt. Unter dem Blickwinkel der *Tonheit* betrachtet, bedeutet das: Je weiter Obertöne oder Teiltöne vom harmonischen Grundton abweichen, desto größer wird ihre Dissonanz.
- (ii) Konsonanz und Dissonanz hängen nicht nur vom Intervallgröße ab, sondern auch vom Spektrum der Töne. Unter dem Blickwinkel der *Tonheit* bedeutet das: Intervalle sind umso dissonanter, je mehr ihre Teiltöne von harmonischen Tönen abweichen. Unter dem Blickwinkel der Wahrnehmung hingegen bedeutet das: Intervalle sind dissonant, wenn durch ihre Teiltöne Rauheit entsteht.
- (iii) Konsonanz und Dissonanz sind nicht klar trennbar, sondern liegen auf einem Kontinuum.

Sensorische Dissonanz

Die Sensorische Dissonanz tritt durch das menschliche Gehörssystem auf und äußert sich in Effekten wie Rauigkeit und Schwebung.

Erklingen zwei Sinuswellen gleichzeitig, sind beide voneinander getrennt wahrnehmbar, solange die Frequenzen der beiden weit genug voneinander entfernt liegen. Liegen die beiden Frequenzen nahe beieinander, kann nur eine einzelne Frequenz wahrgenommen werden, aufgrund der mangelnden Frequenzauflösung unseres Gehörs kommt es zur Schwebung.

Fixiert man eine Sinusschwingung bei einer beliebigen Frequenz (z. B. 440 Hz) und variiert eine zweite (von z. B. 440 Hz bis 880 Hz), so stellen sich nacheinander folgende Phänomene ein:

- (i) Für eng beieinander liegende Frequenzen werden beide Schwingungen als eine wahrgenommen.
- (ii) Für nahe beieinander liegende Frequenzen stellt sich eine langsame Schwebung ein.
- (iii) Je weiter die beiden Frequenzen voneinander entfernt liegen, desto schneller wird die Schwebung und desto rauer wird der Gesamtklang wahrgenommen.
- (iv) Sind die beiden Frequenzen ausreichend voneinander entfernt, können beide Schwingungen getrennt wahrgenommen werden.

Plomb und Levelt stellten Untersuchungen an, in denen sie einer Reihe von Versuchspersonen Paare von Sinusschwingungen vorspielten und sie über ihre Empfindungen in Hinsicht auf Dissonanz befragten. Abbildung 2.4 zeigt, dass die Dissonanz ihr Minimum erreicht, wenn die beiden Schwingungen die gleiche Frequenz aufweisen. Unmittelbar danach steigt die Dissonanz stark bis zu ihrem Maximum an und fällt exponentiell ab. Das Experiment zeigt, dass bereits Septime und None die gleich geringe sensorische Dissonanz besitzen wie die Oktave (sofern es sich um reine Sinusschwingungen handelt), [12].

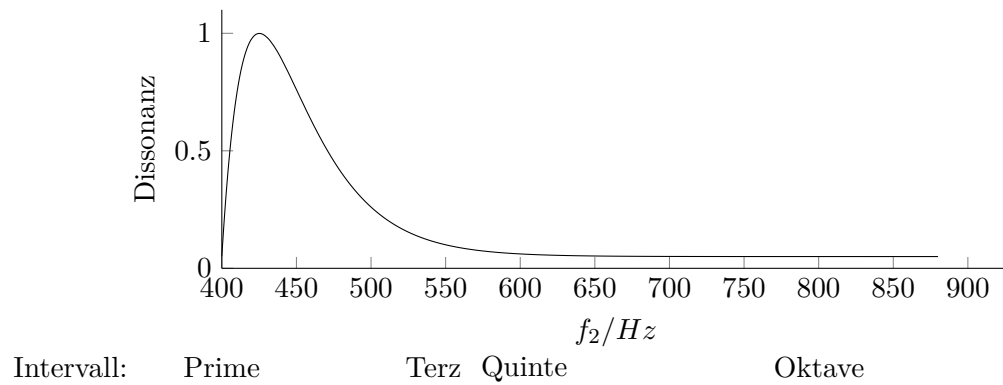


Abbildung 2.4: Experiment von Plomp und Levelt. Zwei Sinusschwingungen erklingen zeitgleich. $f_1 = 400$ Hz, $f_2 = 400 \leq x \leq 800$ Hz. Die X-Achse zeigt die Frequenz der zweiten Schwingung, die Y-Achse die sensorische Dissonanz, die von den Versuchspersonen wahrgenommen wurde, normiert auf 1 als Maximum.

2.2.6 Zusammenfassung

Teilbereiche der Tonhöhenwahrnehmung sind noch nicht eindeutig erklärt und Gegenstand aktueller Forschungen. Bereiche wie die Wahrnehmung im Bezug auf Klangfarbe, Konsonanz und Dissonanz sind stark vom persönlichen Empfinden abhängig und unterlagen in der Vergangenheit einem wechselnden Konsens.

Auch wenn die Dissonanz hier überwiegend als ein aus dem unvollkommenen Aufbau der chromatischen Tonleiter entstehendes Problem dargestellt wird, ist sie dennoch ein wichtiger Bestandteil der Musik.

„Im wechsellvollen Spiel der einzelnen Stimmen, in der Freizügigkeit der Melodie liegt es begründet, daß im Verlauf eines Stückes häufig die reine Konsonanz der tonalen Klänge durch Töne getrübt wird, die der konsonanten Einheit des Klanges widerstreben und daher als Fremdkörper angesehen werden müssen.“ [1, S. 112]

Der Einsatz von dissonanten Klängen ist vor allem als Stilmittel begründet, totale Konsonanz muss daher nicht unbedingt das Ziel sein.

3

Stimmungssysteme

"Die Obertonreihe . . . enthält noch viele Probleme, die eine Auseinandersetzung nötig machen werden. Und wenn wir diesen Problemen augenblicklich noch entrinnen, so verdanken wir das fast ausschließlich einem Kompromiß zwischen den natürlichen Intervallen und unserer Unfähigkeit sie zu verwenden."

– Arnold Schönberg [13, S. 22ff]

Die (gleichstufige) Stimmung von Tasteninstrumenten, wie wir sie heute kennen, beruht auf Kompromissen. Für die Verhältnisse zwischen den einzelnen Tönen gibt es eine klare Definition, die als *rein* angesehen wird. Bei Instrumenten mit statischer Stimmung ist es für das Spiel in verschiedenen Tonarten jedoch nicht möglich, alle Verhältnisse dieser Definition zu berücksichtigen. Es kommt zu einem Fehler, der nur durch Änderung einzelner Frequenzverhältnisse ausgeglichen werden kann.

Das nachfolgende Kapitel ist eine Einführung in das Thema der Intonation, Grundbegriffe und zugrunde liegende Probleme werden erklärt, des weiteren werden Lösungsansätze aus der Vergangenheit beleuchtet und es wird in die Möglichkeiten der dynamischen Intonation eingeleitet.

3.1 Probleme von reinen Intervallen

Als reine Intervalle bezeichnet man die in Abschnitt 2.1.2 aufgelisteten Frequenzverhältnisse. Die Obertöne dieser Intervalle treffen aufeinander und ergeben einen „reinen“ Klang (vgl. Abschnitte 2.1.2 und 2.2.5) Im Gegensatz dazu stehen „unreine“ Intervalle, deren Frequenzverhältnis verändert wurde.

Nach den vorangegangenen Erläuterungen erscheinen die Zusammenhänge zwischen den einzelnen Intervallen sinnvoll. Wieso sollten die Frequenzverhältnisse also geändert werden? Es ist unmöglich die beschriebenen Intervalle in einer 12-stufigen Skala unterzubringen. Es treten eine Reihe von Fehlern auf, die *Kommata* genannt werden. Diese bezeichnen ein kleines Intervall, das bei der Kombination von reinen Intervallen auftritt. Die wichtigsten dieser Fehler werden hier aufgelistet.

1	2	3	4	5	6	7	8	9	10	11	12
C	Cis	D	Es	E	F	Fis	G	Gis	A	B	H
Prime		Sekunde		Terz	Quarte		Quinte		Sexte		Septime

Tabelle 3.1: Die chromatische Tonleiter mit den acht Intervallen der C-Dur-Tonleiter.

3.1.1 Das pythagoreische Komma

Der Pythagoreer Philolaos ordnete die Verhältnisse der Saitenlängen an seiner Lyra als Oktave ($1/2$), Quinte ($2/3$), Quarte ($3/4$) und definierte die Differenz aus Quarte und Quinte als Ganzton mit dem Frequenzverhältnis ($3/2 : 4/3 = 9/8$). Zwölf aneinandergereihte reine Quinten (7 Halbtonschritte) enden abgezählt auf der Klaviatur auf der gleichen Taste wie sieben aneinandergereihte Oktaven (12 Halbtonschritte). Berechnet man die Intervallgrößen dieser beiden Sprünge, ergibt sich ein Abstand, der pythagoreisches Komma genannt wird. Aus Gleichung 2.4 ergibt sich die Größe des Intervalls von 12 Quinten als:

$$d_q = 12 \cdot (1200 \cdot \text{ld}\left(\frac{2}{3}\right)) = 8423,46 \text{ Cent} \quad (3.1)$$

Die Größe von sieben Oktaven ergibt sich aber als:

$$d_o = 7 \cdot (1200 \cdot \text{ld}\left(\frac{1}{2}\right)) = 8400 \text{ Cent} \quad (3.2)$$

Damit ergibt sich das pythagoreische Komma:

$$d_q - d_o = 23,46 \text{ Cent} \quad (3.3)$$

Das pythagoreische Komma ist also ein Intervall mit der Größe 23,46 Cent oder, nach Gleichung 2.3, einem Frequenzverhältnis von 1,0136. In den meisten Stimmungen entspricht dies einem Schritt von einem Viertel Halbton.

3.1.2 Das syntonische Komma

Das sich aus zwei (pythagoreischen) Ganztönen ergebende Intervall wird als *Ditonus* oder *pythagoreische große Terz* bezeichnet. Die Differenz zwischen der pythagoreischen großen Terz (zwei pythagoreische Ganztönen) und der reinen großen Terz wird syntonisches Komma genannt. Aus Gleichung 2.4 ergibt sich die Größe des Intervalls von zwei pythagoreische Ganztönen als:

$$d_p = 2 \cdot (1200 \cdot \text{ld}\left(\frac{9}{8}\right)) = 407,82 \text{ Cent} \quad (3.4)$$

Die Größe der reinen großen Terz aber ergibt sich als:

$$d_r = 1200 \cdot \text{ld}\left(\frac{5}{4}\right) = 386,31 \text{ Cent} \quad (3.5)$$

Damit ergibt sich das syntonische Komma:

$$d_p - d_r = 21,51 \text{ Cent} \quad (3.6)$$

Das syntonische Komma tritt auf, wenn ein Instrument auf C-Dur rein gestimmt wurde und nun auf G-Dur gespielt wird. In diesem Fall wird nicht nur F zu Fis sondern auch A müsste (um das syntonische Komma) verschoben werden, um die restlichen Bedingungen zu erfüllen. In den meisten Stimmungen entspricht das syntonische Komma einem Schritt von einem Fünftel Halbtonschritt.

3.1.3 Die kleine und große Diësis

Setzt man drei große Terzen (je 4 Halbtonschritte) hintereinander, so sollten sie auf dem gleichen Ton enden wie eine Oktave (12 Halbtonschritte). Dies ist jedoch nicht der Fall, der daraus entstehende Unterschied wird kleine Diësis genannt.

Reiht man vier kleine Terzen aneinander, so sollten sie ebenso auf dem gleichen Ton enden wie eine Oktave. Dies ist jedoch nicht der Fall, der daraus entstehende Unterschied wird große Diësis genannt.

Aus Gleichung 2.4 ergibt sich die Größe von drei großen Terzen als:

$$d_{gt} = 3 * 1200 * \text{ld}\left(\frac{5}{4}\right) = 1158,94 \text{ Cent} \quad (3.7)$$

Die Größe von vier kleinen Terzen ergibt sich als:

$$d_{kt} = 4 * 1200 * \text{ld}\left(\frac{6}{5}\right) = 1262,56 \text{ Cent} \quad (3.8)$$

Die Größe der Oktave ist 1200 Cent. Die kleine Diësis beträgt:

$$1200 - d_{gt} = 41,06 \text{ Cent} \quad (3.9)$$

Die große Diësis beträgt:

$$1200 - d_{kt} = 62,56 \text{ Cent} \quad (3.10)$$

Philolaos (siehe Abschnitt 3.1.1) verstand als Diësis den Unterschied zwischen Quarte und Ditonus, dieser wurde später als *Limma* oder pythagoreischer Halbton bezeichnet.

3.1.4 Zusammenfassung

In erster Linie treten die aufgezeigten Probleme bei Tasteninstrumenten und Saiteninstrumenten mit Bündeln auf. Es ist notwendig beim Stimmen dieser Instrumente einen Kompromiss einzugehen. Dieser Kompromiss wird als *Temperierung* bezeichnet und dient der Kompensation der einzelnen Kommata.

Steht die Reinheit der Oktave an erster Stelle und soll das pythagoreische Komma auszugleichen werden, müssen die 12 Quinten in den sieben Oktaven verstimmt werden. Diese Verstimmung beträgt:

$$1,995 \text{ Cent}$$

pro Oktave. Will man reine Terzen bewahren und das syntonische Komma ausgleichen, so muss das Komma auf vier Quinten aufgeteilt werden, die innerhalb von zwei Oktaven plus einer großen Terz liegen. Für die C-Dur-Tonleiter sind das die Quinten: C/G, G/D, D/A und A/E. Ihre Verstimmung beträgt für diesem Fall:

$$5,377 \text{ Cent.}$$

Diese Quinten werden auch mitteltönige Quinten genannt. Dieser Name entspringt der mitteltönigen Stimmung, auf die in Abschnitt 3.2.4 weiter eingegangen wird. Die kleine Diësis zeigt die Unmöglichkeit sowohl reine Terzen als auch reine Oktaven zu erhalten, ohne ein Restintervall von 41,06 Cent zu bilden.

Aus Abschnitt 2.2.2 auf S. 19 ist bekannt, dass bereits eine Änderung der Tonhöhe von 3 Cent wahrnehmbar ist. Pythagoräisches und syntonisches Komma gleichzeitig auszugleichen ist daher ausgeschlossen.

3.2 Stimmung, Intonation, Temperierung

Als Stimmen wird in der Musik das Festsetzen der Tonhöhe bei Instrumenten bezeichnet. Je nach Instrumententyp kann die Festlegung einer Normfrequenz (meistens der Kammerton $a^1 = 440$ Hz) ausreichend sein, bei Tasten- und Saiteninstrumenten mit Bündeln müssen zusätzlich noch die Tonhöhen der Saiten bzw. Tasten relativ zu einem Grundton festgelegt werden. Während Blasinstrumente und die meisten Saiteninstrumente nach Stimmung in der Lage sind (bedingt durch die Spieltechnik) alle Intervalle rein wiederzugeben, müssen bei anderen Instrumenten Kompromisse in Kauf genommen werden (vgl. Abschnitt 3.1). Werden Kompromisse eingegangen und sind die Intervalle nicht mehr rein, die Tonhöhen also vorherrschenden Ideen angepasst, spricht man von Temperierung. Im Kontext von Instrumenten, die eine solche Temperierung erfordern werden die Begriffe Stimmung und Temperierung meist gleich verwendet. Über die Jahrhunderte wurden viele verschiedene Möglichkeiten veröffentlicht, wie eine solche Temperierung auszusehen habe. Nachfolgender Abschnitt stellt einige wichtige Vertreter vor.

Der Begriff Intonation wird hier als das zusätzliche Anpassen von Tonparametern wie Klangfarbe oder Lautstärke verwendet. Je nach Instrument erfolgt die Intonation vom Musiker selbst oder bereits zuvor vom Instrumentenbauer oder Stimmmeister. Im Gegensatz dazu kann mit Intonation auch das Festlegen der Tonhöhe gemeint sein, sofern dies vom Spieler gesteuert wird (vgl. Abschnitt 1.3).

3.2.1 Intonation von Tasteninstrumenten

Tasteninstrumente besitzen einen diskreten Tonumfang. Durch ihre Konstruktion kann pro Taste exakt ein Ton wiedergegeben werden. Bei zwölf Tasten pro Oktave sind dies also genau zwölf Töne pro Oktave. Zusätzlich zur Stimmung werden die meisten Tasteninstrumente vor dem Spiel intoniert, da in den meisten Fällen auf Parameter wie Klangfarbe oder ggf. auch Lautstärke kein Einfluss mehr genommen werden kann.

Das Stimmen von Klavieren erfolgt über Variation der Saitenspannung. Zur Intonation gehört weiters das Bearbeiten der Hammerköpfe. Die größte Rolle hierbei spielt die Elastizität des Hammerfilzes: je härter der Filz, desto lauter erklingt die Saite. Zusätzlich relevant ist die Zeit, die der Filz benötigt, um wieder in seine Ausgangsform zurückzufinden: je elastischer der Filz, desto weicher und reicher an Obertönen wird der Klang.

Beim Cembalo werden die Saiten durch hervorstehende Plektren angerissen, d. h. das Plektrum hebt sich immer weiter hervor, bis die Saite darüber seitlich abrutscht. Das Stimmen erfolgt wie beim Klavier über die Änderung der Saitenspannung, eine Änderung der Klangfarbe über das Zuschneiden dieser Plektren. Die Elastizität dieser Plektren ist ebenfalls klangbestimmend.

Inharmonizität beschreibt das Auftreten der Erhöhung von Teiltönen. Sie tritt bei Saiteninstrumenten aufgrund der Steifigkeit der Saiten auf und ist ein wichtiges Merkmal der Intonation, aber auch der Betrachtung von reinen Intervallen. Abschnitt 3.3 auf S. 44 befasst sich weiter damit.

Orgel

Bei der Orgel unterscheidet man zwei große Gruppen von Pfeifen: Labialpfeifen (Lippenpfeifen) und Lingualpfeifen (Zungenpfeifen). Bei Labialpfeifen erfolgt die Tonerzeugung durch den Luftaustritt am Kernspalt, bei Lingualpfeifen durch das Schwingen einer Metallzunge. Grundsätzlich bestimmt die Mensur der Pfeife auch ihren Klang. Nachfolgend werden Stimm- und Intonations-Techniken angerissen.

Labialpfeifen existieren in verschiedenen Bauarten, grob kann zwischen gedeckt, offen und halboffen unterschieden werden. Je nach Bauart erfolgt auch die Stimmung.

- Stimmen mit Stimmdeckel: Gedeckte Metallpfeifen können durch Verschieben des Deckels gestimmt werden. Offene Holzpfeifen besitzen oft einen metallenen Stimmdeckel, welcher verbogen werden kann. Das Verschieben des Deckels bewirkt eine Veränderung der Pfeifenlänge.
- Stimmen mit Stimmring: Manche Metallpfeifen besitzen Stimmringe, welche wie der Deckel verschoben werden können. Auch hier wird durch das Verschieben die Länge der Pfeife variiert.
- Stimmen mit Stimmrolle oder Schieber: Metallpfeifen besitzen oft an ihrer Mündung einen Stimmschlitz, an dem das Metall gerollt wird. Die Stimmung erfolgt durch Ein- oder Ausrollen dieser Stimmrolle. Bei Holzpfeifen kann ein Schieber vorhanden sein.
- Stimmen mit Stimmhorn: Offene Metallpfeifen, die genau auf Tonhöhe zugeschnitten wurden, können mit einem Stimmhorn gestimmt werden. Mit dessen Hilfe wird das Ende der Pfeife geweitet oder verengt.

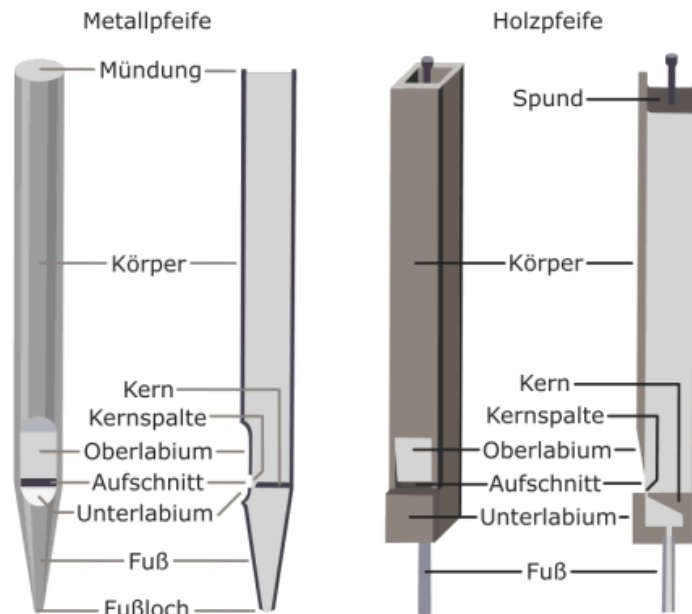


Abbildung 3.1: Aufbau von Labialpfeifen aus Metal und Holz.

Bildnachweis: Peter Bach

Creative Commons, lizenziert unter CreativeCommons-Lizenz by-nc-nd-3.0-de

Neben der Bauform existiert eine Vielzahl anderer Möglichkeiten den Klang der Labialpfeife zu beeinflussen. Tabelle 3.2 gibt eine Übersicht für gedeckte und für offene Pfeifen (i. d. R. Prinzipale), [14].

Lingualpfeifen besitzen im unteren Teil der Pfeife, dem Stiefel, eine Stimmvorrichtung. Mit einem langen festen Draht, der Stimmkrücke, wird die Zunge festgedrückt. Nur der frei bewegliche Teil der Zunge vor der Stimmkrücke kann frei schwingen und so den Ton erzeugen. Am oberen Ende der Pfeife ist die Stimmkrücke herausgeführt und bei deutscher Bauart abgewinkelt⁷. Mithilfe eines Stimmeisens kann die Krücke hinauf- oder hinabgeschoben (geschlagen)

⁷ Nicht bei französischer Bauart.

werden, durch diese Verschiebung ändert sich der Auflagepunkt auf der Zunge und so die Länge des freischwingenden Teils. Durch die Längenänderung kommt es zur Änderung der Tonhöhe [14].

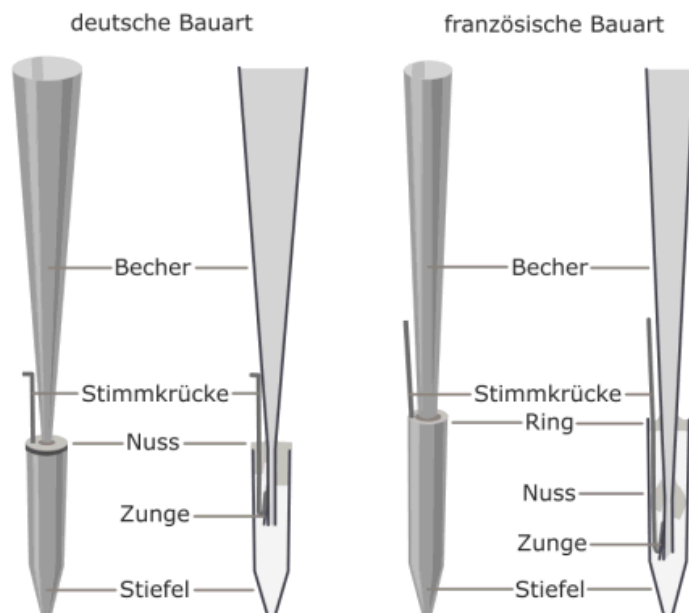


Abbildung 3.2: Aufbau von Lingualpfeifen deutscher und französischer Bauart.

Bildnachweis: Peter Bach

Creative Commons, lizenziert unter CreativeCommons-Lizenz by-nc-nd-3.0-de

Tabelle 3.3 zeigt eine Liste an Möglichkeiten den Klang von Lingualpfeifen weiter zu formen.

Baugruppe	Veränderung	Klangauswirkung
Gedckte Labialpfeifen		
Oberlabium	Eindrückung	längere und härtere Ansprache
	Anhebung	kürzere und weichere Ansprache
Kernspalt	Erweiterung	längere und härtere Ansprache, lauter
	Verengung	kürzere und weichere Ansprache, leiser
Kern	Erhöhung	längere und härtere Ansprache
	Erniedrigung	kürzere und weichere Ansprache
Offene Labialpfeifen		
Fußloch	Erweiterung	lauter, schärfer, forciertes
	Verengung	leiser, lieblicher, grundtonbehafteter
Oberlabium	Eindrückung	schärfer, stumpfere Ansprache
	Anhebung	flötiger, härtere Ansprache
Kern	Erhöhung	schärfer, lauter, stumpfere Ansprache
	Vertiefung	grundtonbehafter, flötiger, leiser, härtere Ansprache
Kernspalte	Verengung	leiser, diffuser, schnellere Ansprache
	Erweiterung	lauter, klarer, schärfer
Aufschnitt	Erhöhung	flötiger, grundtonbehafteter, langsamere Ansprache
	Erniedrigung	schärfer, schnellere Ansprache

Tabelle 3.2: Intonationsmittel für offene und geschlossene Labialpfeifen, aus [14].

Baugruppe	Veränderung	Klangauswirkung
Lingualpfeifen		
Zunge	Aufwurf verstärken	lauter, grundtonbehafter, langsamere Ansprache
	Aufwurf schwächen	leiser, obertonbehafter, schnellere Ansprache

Tabelle 3.3: Intonationsmittel für Lingualpfeifen, aus [14].

3.2.2 Reine Stimmung

Ende des 15. Jahrhunderts, mit dem Aufkommen der Mehrstimmigkeit, entstand die reine Stimmung. Sie beruht einzig auf den Grundintervallen Oktave, Quinte, Terz und darauf, dass diese Intervalle rein sind. Das Ziel einer reinen Stimmung ist es daher sowohl Quinten als auch Terzen rein zu stimmen, zumindest für ausgewählte Tonarten.

Durch Terz und Quinte sind der dritte und fünfte Ton der Oktave bereits festgelegt. Alle anderen lassen sich durch Addition (Subtraktion) von Quinten (Q), Terzen (T) und Oktaven (O) bilden:

Stufe	Tonbildung (musikalisch)	Tonbildung (mathematisch)	Frequenzverhältnis
2	$Q + Q - O$	$\frac{2}{3} \cdot \frac{2}{3} : \frac{1}{2}$	$\frac{8}{9}$
4	$O - Q$	$\frac{1}{2} : \frac{2}{3}$	$\frac{3}{4}$
6	$O + T - Q$	$\frac{1}{2} \cdot \frac{4}{5} : \frac{2}{3}$	$\frac{3}{5}$
7	$Q + T$	$\frac{2}{3} \cdot \frac{4}{5}$	$\frac{8}{15}$

Tabelle 3.5 ergänzt Tabelle 2.2 von Seite 14 durch die Frequenzverhältnisse bei reiner Stimmung für C-Dur.

C	D	E	F	G	A	H	c
Prime	gr. Sekunde	gr. Terz	Quarte	Quinte	gr. Sexte	gr. Septime	Oktave
$\frac{1}{1}$	$\frac{8}{9}$	$\frac{4}{5}$	$\frac{3}{4}$	$\frac{2}{3}$	$\frac{3}{5}$	$\frac{8}{15}$	$\frac{1}{2}$

Tabelle 3.5: Intervalle der C-Dur-Tonleiter und deren Frequenzverhältnisse bei reiner Stimmung.

Bei Modulation, beispielsweise von C-Dur nach G-Dur, kommt das syntonische Komma (vgl. Abschnitt 3.1.2) zum Tragen. Nicht nur F wird dabei durch Fis ersetzt, um die Frequenzverhältnisse der reinen Stimmung beizubehalten, müsste der Ton A ein syntonisches Komma höher gestimmt werden (vgl. Abschnitt 3.1.2). Ein Wechsel zwischen verschiedenen Tonarten ist daher auf Tasteninstrumenten nur nach Änderung der Stimmung möglich, während eines Stücks also praktisch unmöglich.

3.2.3 Pythagoreische Stimmung

Der Name pythagoreische Stimmung hat mit Philolaos (siehe Abschnitt 3.1.1) keinen direkten Zusammenhang, es wird aber in Erinnerung gerufen, dass das System der Skalenbildung bereits von den Pythagoreern erforscht wurde [15].

Die pythagoreische Stimmung fand vor allem in der Neuzeit, mit Beginn des 16. Jahrhunderts, ihre Anwendung. Als Hauptziel verfolgt sie reine Quinten zu erhalten. Ihr Tonsystem baut einzig auf Quinten auf und nicht wie bei der reinen Stimmung auf Quinten und Terzen. Die einzelnen Tonstufen bilden sich aus der Addition (bzw. Subtraktion) von Quinten und Oktaven. Alle Quinten rein zu halten ist, wie aus Abschnitt 3.1.1 hervorgeht, aber nicht möglich. Der Kompromiss besteht darin so viele Quinten wie möglich rein zu lassen und das verbleibende pythagoreische Komma einer einzigen Quinte anzulasten. Diese stark verstimmte Quinte wird auch als *Wolfsquinte* bezeichnet.

Ein weiterer Nachteil der pythagoreischen Stimmung ist es, dass durch die Reinheit der Quinten Terzen ebenfalls stärker verstimmt werden. Einzig die Terz H/Es, die das pythagoreische Komma enthält, ist annähernd rein.

Als Beispiel wird die Wolfsquinte auf den Schritt Gis/Es gelegt. Um die Frequenzverhältnisse und Intervallgrößen innerhalb der Oktave zu bekommen, werden von C ausgehend drei Quinten abwärts und acht aufwärts berechnet [10]. Durch die Division von Zweierpotenzen (Oktavierung) erhält man das Ergebnis innerhalb der betrachteten Oktave. Tabelle 3.6 verdeutlicht die Berechnung und zeigt die Intervallgrößen der pythagoreischen Stimmung bezogen auf C als Ausgangston.

Tonstufe	Berechnung	Frequenzverhältnis	Intervallgröße
gis	$1/16 \cdot (3/2)^8$	1,6018	815,64 Cent
cis	$1/16 \cdot (3/2)^7$	1,0679	113,73 Cent
fis	$1/8 \cdot (3/2)^6$	1,4238	611,70 Cent
h	$1/4 \cdot (3/2)^5$	1,8984	1109,74 Cent
e	$1/4 \cdot (3/2)^4$	1,2656	407,79 Cent
a	$1/2 \cdot (3/2)^3$	1,6875	905,87 Cent
d	$1/2 \cdot (3/2)^2$	1,1250	203,91 Cent
g	$1 \cdot (3/2)^1$	1,5000	701,96 Cent
c	$1 \cdot (3/2)^0$	1,0000	0 Cent
f	$2 \cdot (3/2)^{-1}$	1,3333	498 Cent
b	$4 \cdot (3/2)^{-2}$	1,7778	996,11 Cent
es	$4 \cdot (3/2)^{-3}$	1,1852	294,16 Cent

Tabelle 3.6: Intervallgrößen der pythagoreischen Stimmung, Wolfsquinte Gis/Es aus [10].

Abbildung 3.3 auf S. 37 zeigt die Abweichung der Quinten und Terzen von der reinen Stimmung. Der Arbeit liegt ein Klangbeispiel zur pythagoreischen Stimmung bei. Siehe dazu Anhang A, Klangbeispiel 1.

3.2.4 Mitteltönige Stimmungen

Als mitteltönige Stimmung bezeichnet man eine Gruppe von Stimmungen aus der Zeit des 15. bis 17. Jahrhunderts, deren Ziel es ist die reine Stimmung so gut wie möglich zu verwirklichen, dies allerdings nur für eine begrenzte Anzahl an Tonarten. Meistens ist die hier beschriebene $1/4$ -Komma basierende mitteltönige Stimmung gemeint, die das Ziel für die Tonarten C-, D-, F-, G-, A- und B-Dur realisiert.

Um eine reine große Terz (C/E) zu erhalten, wird das syntonische Komma über die vier Quinten C/G, G/D, D/A und A/E gleichmäßig verteilt. Diese „mitteltönigen“ Quinten bilden die Basis für die Tonstruktur der mitteltönigen Stimmung. Für die restlichen Tonstufen werden annähernd reine Intervalle verwendet.

Die große Terz entspricht zwei Ganztönen. Bei der reinen Stimmung werden diese beiden Töne, bedingt durch die Frequenzverhältnisse, in einen kleinen Ganzton (Frequenzverhältnis $10/9$) und einen großen Ganzton (Frequenzverhältnis $9/8$) aufgeteilt. Der Name „mitteltönige Stimmung“ kommt von der Teilung der großen Terz genau in der Mitte, in zwei gleich große Ganztöne. Das resultierende Frequenzverhältnis beträgt:

$$\sqrt{\frac{10}{9} \cdot \frac{9}{8}} = \sqrt{\frac{5}{4}} \quad (3.11)$$

Tabelle 3.7 zeigt die Tonhöhen der mitteltönigen Stimmung in Cent als Abstand zum Grundton. Jede Quinte wird demnach um ungefähr fünf Cent verstimmt und entspricht 697 Cent bzw. einem Frequenzverhältnis von 1,495. Vier dieser als mitteltönige Quinten bezeichneten Intervalle ergeben eine Terz plus zwei Oktaven. Nach Gleichung 3.12 und 3.13 ergibt sich nun eine Intervallgröße von 386 Cent für die Terz. Sie ist also rein. (Wie schon in Tabelle 3.6 wird in Gleichung 3.12 mit dem Faktor $1/4$, also zwei Oktaven, multipliziert, um das Ergebnis innerhalb der Ausgangsoktave zu erhalten.)

$$1/4 \cdot (1,495)^4 = 1,25 \quad (3.12)$$

$$1200 \cdot \text{ld}(1,25) = 386,6 \text{ Cent} \quad (3.13)$$

Das pythagoreische Komma wird nicht kompensiert, die Wolfsquinte (**Gis/Es**) bleibt daher erhalten. Innerhalb des Intervalls von elf Quinten ergeben sich acht reine Terzen. Die verbleibenden vier Terzen (**H/Es**, **Fis/B**, **Cis/F**, **Gis/C**) überschneiden sich mit der Quinte, die den Überschuss des syntonischen Kommas aufnimmt, und sind daher stärker verstimmt [10, S.26]. In Abwandlungen der mitteltönigen Stimmung wurde versucht auch die Wolfsquinte zu vermeiden, wobei die reinen Terzen erhöht und somit ebenfalls verstimmt wurden.

Tonstufe	Intervallgröße
c	0 Cent
cis	76 Cent
d	193 Cent
es	310 Cent
e	386 Cent
f	503 Cent
fis	580 Cent
g	697 Cent
gis	773 Cent
a	890 Cent
b	1007 Cent
h	1083 Cent

Tabelle 3.7: Tonhöhen der mitteltönigen Stimmung in Cent, als Abstand zum Grundton (C), aus [16].

Abbildung 3.4 auf Seite 37 zeigt die Abweichung der Quinten und Terzen bei mitteltöniger Stimmung zur reinen Stimmung. Andere Ableger der Gruppe der mitteltönigen Stimmungen ist z. B. die Schlick-Stimmung oder die im nächsten Abschnitt beschriebene Silbermann-Stimmung. Der Arbeit liegt ein Klangbeispiel zur mitteltönigen Stimmung bei. Siehe dazu Anhang A, Klangbeispiel 2.

3.2.5 Silbermann-Stimmung

Als einer der bedeutendsten braocken Orgelbauer im deutschsprachigen Raum gilt Gottfried Silbermann. Zusammen mit seinem Bruder und dessen Sohn erbaute er bis zu seinem Tod 1753 insgesamt ca. 50 Orgeln, Cembali, Clavichorde und Hammerflügel. Es existieren keine Veröffentlichungen oder Anweisungen Silbermanns zur Stimmung von Tasteninstrumenten. Auch wurden seine Orgeln im Laufe der Jahrhunderte mehrmals umgestimmt, daher beruhen die als Silbermann 1 und Silbermann 2 bekannten Stimmungen lediglich auf Rekonstruktionsversuchen.

Der Musiktheoretiker Andreas Sorge untersuchte Mitte des 18. Jahrhunderts zwei der Silbermann-Orgeln und veröffentlichte seine Ergebnisse. Die darin enthaltene Stimmanweisung wird als *Silbermann-Sorge-Temperatur* oder Silbermann-2-Stimmung bezeichnet. Es handelt sich dabei um eine mitteltönige Stimmung, in welcher ein Sechstel des pythagoreischen Kommas auf insgesamt elf Quinten aufgeteilt wird, die verbleibende Quinte (Gis/Es) wird um die verbleibenden fünf Sechstel verstimmt.

Tonstufe	Intervallgröße Silbermann 1	Intervallgröße Silbermann 2
c	0 Cent	0 Cent
cis	87 Cent	86 Cent
d	200 Cent	196 Cent
es	308 Cent	306 Cent
e	390 Cent	392 Cent
f	503 Cent	502 Cent
fis	590 Cent	588 Cent
g	700 Cent	365 Cent
gis	784 Cent	784 Cent
a	895 Cent	894 Cent
b	1006 Cent	1004 Cent
h	1090 Cent	1090 Cent

Tabelle 3.8: Tonhöhen der Silbermann-Stimmung in Cent als Abstand zum Grundton (C), aus [10].

In den Stimmungen erkennbar ist die Tendenz weder Quinten noch Terzen allzu stark zu verstimmen. Abbildung 3.5 auf Seite 37 zeigt die Abweichung der Quinten und Terzen von der reinen Stimmung. Der Arbeit liegt ein Klangbeispiel zur Silbermann-Stimmung bei. Siehe dazu Anhang A, Klangbeispiel 3.

3.2.6 Werckmeister-Stimmung (wohltemperierte Stimmung)

Andreas Werckmeister, selbst Organist und Musiktheoretiker, entwickelte zwischen 1681 und 1691 verschiedene wohltemperierte Stimmungen, die nach und nach die bis dahin vorherrschende mitteltönige Stimmung erweiterten und das Spielen in verschiedenen Tonarten erlaubten.

Tonstufe	Intervallgröße
c	0 Cent
cis	90 Cent
d	192 Cent
es	294 Cent
e	390 Cent
f	498 Cent
fis	588 Cent
g	696 Cent
gis	792 Cent
a	888 Cent
b	996 Cent
h	1092 Cent

Tabelle 3.9: Tonhöhen der Werckmeister-III-Stimmung in Cent als Abstand zum Grundton (C), aus [16].

Von den von Werckmeister hinterlassenen Stimmungen erlangte vor allem Nummer III an Bedeutung. Das pythagoreische Komma wird auf vier Quinten (C/G, G/D, D/A und H/Fis) glei-

chermaßen aufgeteilt. Diese verkleinerten Quinten werden auch als *Werckmeister-Quinten* bezeichnet. Alle anderen Quinten sind somit rein. Drei der Terzen sind um 22 Cent zu groß – dies entspricht dem pythagoreischen Komma. Die anderen sind um maximal 6 Cent zu groß, abhängig davon, wie viele der s. g. Werckmeister-Quinten sie enthalten.

Tabelle 3.9 zeigt die Tonhöhen der Werckmeister-III-Stimmung in Cent als Abstand zum Grundton.

Die Werckmeister-Stimmung stellt einen wesentlichen Schritt zur heute weitgehend verwendeten gleichstufigen Stimmung dar. Durch die große Flexibilität bei der Wahl der Tonart wurde sie vielseitig eingesetzt, so vermutlich auch von J. S. Bach für „das Wohltemperirte Clavier“ [17]. Auch von J. P. Kirnberger, F. A. Vallotti und J. G. Neidhardt existieren Vorschläge zur Temperierung, diese Stimmungen fallen ebenfalls in die Gruppe der wohltemperierten. Abbildung 3.6 auf Seite 38 zeigt die Abweichung der Quinten und Terzen von der reinen Stimmung.

Werckmeister fasste 1691 in seinem Werk „Musicalische Temperatur“ sechs Stimmungen zusammen. Die Stimmungen I und II waren gleich der reinen bzw. mitteltönigen Stimmung. Erst die Stimmungen III-VI wurden von ihm selbst entwickelt. In verschiedenen Quellen beginnt die Zählung seiner Stimmungen daher bei I, damit ist die in seinem Werk als Nummer III aufgeführte Stimmung gemeint. In obigen Ausführungen wird von Werckmeisters Zählung ausgegangen. Der Arbeit liegt ein Klangbeispiel zur Werckmeister-Stimmung bei. Siehe dazu Anhang A, Klangbeispiel 4.

3.2.7 Gleichstufige Stimmung

Als gleichstufige Stimmungen bezeichnet man Stimmungen, die eine Oktave in gleich große Teile aufteilen. Im deutschen Sprachraum wurden sie unter dem Begriff gleichschwebende Stimmung, 1707 von Andreas Werckmeister bekannt gemacht. Auch wenn verschiedene Quellen die Entstehung der gleichstufigen Stimmung bis in das Jahr 1584 in China zurückverfolgen, konnte diese sich erst gegen Ende des 18. Jahrhunderts behaupten und setzte sich zu Beginn des 19. Jahrhunderts als Standard für Tasteninstrumente durch [15].

Zur Verwendung der gleichstufigen Stimmung wird die Oktave in zwölf gleich große Teile (Halbtonschritte) geteilt (vgl. Abschnitt 2.1.3). Jeder Halbtonschritt entspricht demnach 100 Cent. Nach Gleichung 2.5 entsprechen 100 Cent einem Frequenzverhältnis q von:

$$q = 2^{\frac{100}{1200}} = 1,0595 \approx \frac{196}{185}$$

Tabelle 3.10 zeigt die Tonhöhen der gleichstufigen Stimmung in Cent als Abstand zum Grundton.

Tonstufe	Intervallgröße
c	0 Cent
cis	100 Cent
d	200 Cent
es	300 Cent
e	400 Cent
f	500 Cent
fis	600 Cent
g	700 Cent
gis	800 Cent
a	900 Cent
b	1000 Cent
h	1100 Cent

Tabelle 3.10: Tonhöhen der gleichstufigen Stimmung in Cent als Abstand zum Grundton (c), aus [10].

Das pythagoreische Komma wird auf zwölf aufeinanderfolgende Quinten verteilt, diese zwölf Quinten sind also um nur ca. zwei Cent tiefer gestimmt als reine Quinten. Auch die Quarte weicht mit 500 Cent in der gleichstufigen bzw. 498 Cent in der reinen Stimmung um nur zwei Cent ab. Größer sind die Abweichungen bei kleiner Terz (16 Cent), großer Terz (14 Cent), kleiner Sext (14 Cent) und großer Sext (16 Cent). Es existiert bis auf die Oktave, kein Intervall welches tatsächlich rein gestimmt ist. Abbildung 3.7 auf Seite 38 zeigt die Abweichung der Quinten und Terzen von der reinen Stimmung. Der Arbeit liegt ein Klangbeispiel zur gleichstufigen Stimmung bei. Siehe dazu Anhang A, Klangbeispiel 5.

3.2.8 Neidhardt-Stimmungen

Johann Georg Neidhardt lebte in der gleichen Zeit wie Andreas Werckmeister und war praktisch sein Gegenspieler. Grundlegend für die von ihm überlieferten Temperaturen ist das Streben nach uneingeschränkten Transpositionsmöglichkeiten in alle Tonarten. Er bereitete mit seinen Ideen den Weg zur gleichstufigen Stimmung. In Werckmeisters postum erschienenen Schriften erhebt dieser den Anspruch auf die Idee zur gleichstufigen Stimmung [18].

Von Neidhardt existieren drei Stimmungen von besonderer Bedeutung, diese fasste er in Stimmungsrichtlinien zusammen, angepasst an den Einsatzort. So entstanden Stimmungen „für das Dorf“, „für eine kleine Stadt“ und „für eine große Stadt“. Letztere ist an den bis dahin verbreiteten Einsatz der gleichstufigen Stimmung am Hofe angepasst. Tabelle 3.11 zeigt die Tonhöhen der Neidhardt-Stimmung „für eine große Stadt“ in Cent als Abstand zum Grundton.

Tonstufe	Intervallgröße
c	0 Cent
cis	96 Cent
d	196 Cent
es	298 Cent
e	394 Cent
f	498 Cent
fis	596 Cent
g	696 Cent
gis	796 Cent
a	894 Cent
b	998 Cent
h	1096 Cent

Tabelle 3.11: Tonhöhen der Neidhardt-Stimmung „für eine große Stadt“ in Cent als Abstand zum Grundton (C), aus [10].

Neidhardt änderte immer wieder seine Meinung zur gültigen Stimmung, er schlug zwar bereits früh die gleichstufige Stimmung vor, plädierte aber zwischenzeitlich immer wieder für wohltemperierte Stimmung [15]. Insbesondere aber die Stimmung „für eine große Stadt“ erlangte große Beliebtheit und kam in der Zeit nach Neidhardt auch immer wieder nach erfolgten Orgel-Renovierungen zum Einsatz. Abbildung 3.8 auf Seite 38 zeigt die Abweichung der Quinten und Terzen von der reinen Stimmung. Der Arbeit liegt ein Klangbeispiel zur Neidhardt-Stimmung bei. Siehe dazu Anhang A, Klangbeispiel 6.

3.2.9 Vergleich der Abweichung von der reinen Stimmung

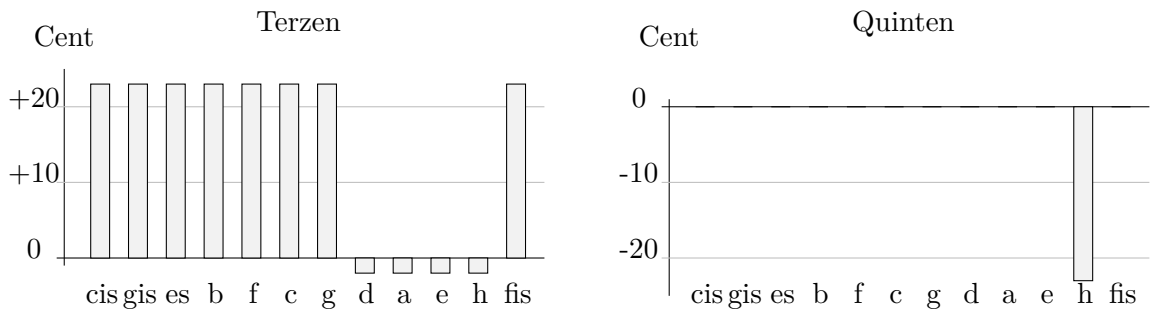


Abbildung 3.3: Abweichung von reinen Intervallen in pythagoreischer Stimmung, aus [15].

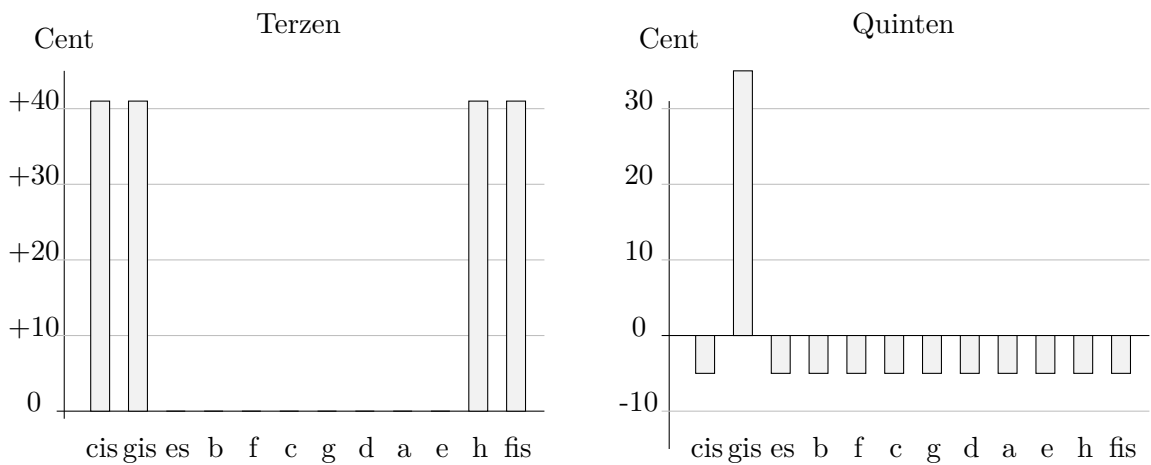


Abbildung 3.4: Abweichung von reinen Intervallen in mitteltöniger Stimmung, aus [15].

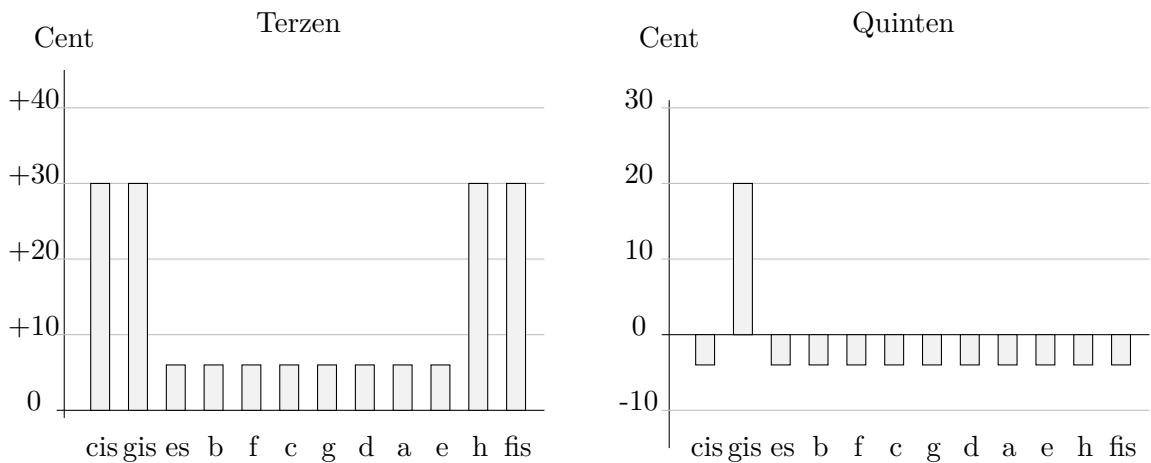


Abbildung 3.5: Abweichung von reinen Intervallen in Silbermann-1-Stimmung.

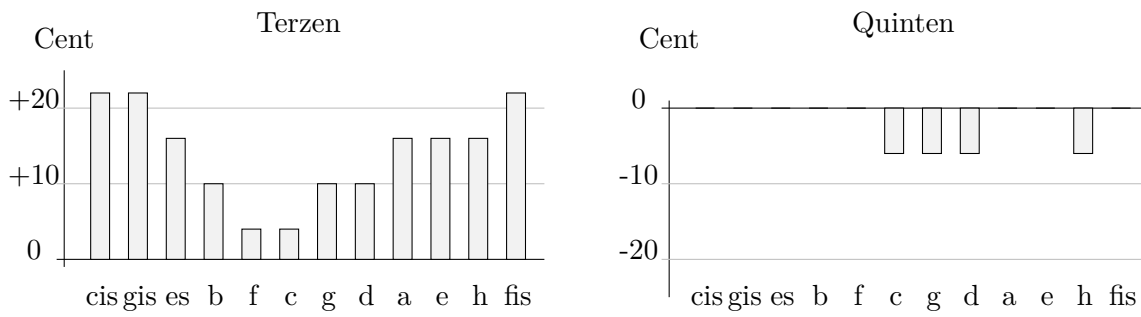


Abbildung 3.6: Abweichung von reinen Intervallen in Werckmeister-III-Stimmung, aus [15].

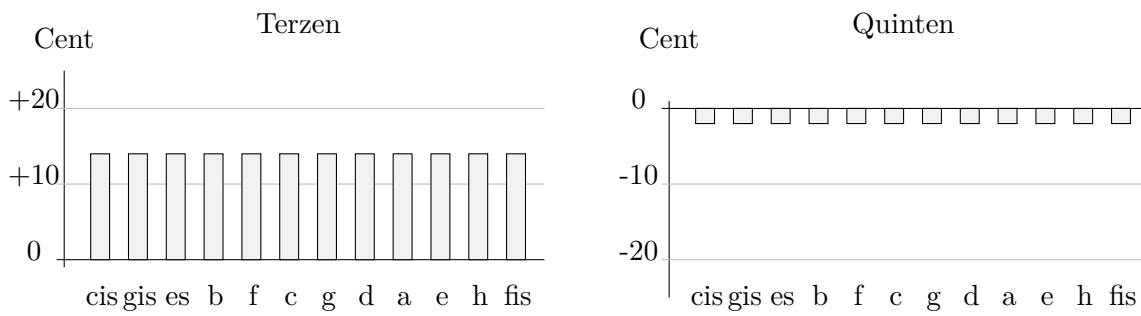


Abbildung 3.7: Abweichung von reinen Intervallen in gleichstufiger Stimmung, aus [15].

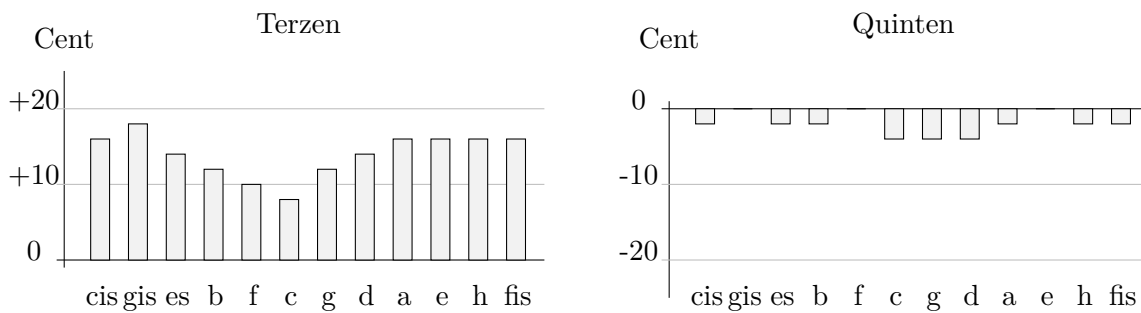


Abbildung 3.8: Abweichung von reinen Intervallen in Neidhardt-Stimmung „für die große Stadt“, aus [10].

3.2.10 Enharmonische Verwechslung

Durch die Beschränkung auf 12 Tasten pro Oktave kommt es dazu, dass eigentlich unterschiedliche Töne der gleichen Taste zugeordnet sind und damit trotz ihrer unterschiedlichen Bedeutung die gleiche Tonhöhe besitzen. Dieser Umstand wird enharmonische Verwechslung genannt. Tabelle 3.12 ist der Klaviatur nachempfunden und zeigt die unterschiedlichen Benennungen der zwölf Tonstufen. Die enharmonische Verwechslung bezeichnet die kompositorische Möglichkeit Töne umzudeuten. Dadurch lässt sich der musikalische Zusammenhang verändern und z. B. der Wechsel in eine andere Tonart bewirken.

his	hisis	cisis	dis	disis	eis	eisis	fisis	gis	gisis	ais	aisis
c	cis	d	es	e	f	fis	g	as	a	b	h
deses	des	eses	feses	fes	geses	ges	ases	as	heses	ceses	ces

Tabelle 3.12: Unterschiedliche Benennung der zwölf Tonstufen.

In pythagoreischer, reiner oder mitteltöniger Stimmung wären solche Wechsel deutlich hörbar, erst die wohltemperierten Stimmungen boten die Möglichkeit eines sanften Überganges, bei gleichstufiger Stimmung sind sie nicht mehr merklich.

Im Laufe des 19. Jahrhunderts begann sich die Festlegung auf einen Grundton bei einigen Kompositionen sogar komplett aufzulösen [13, S. 296f].

3.2.11 Lösungsversuche auf der Basis von Tasteninstrumenten

In den vorhergehenden Abschnitten wurde das Problem reine Intervalle an Tasteninstrumenten wiederzugeben hinreichend beschrieben; so auch der lange Prozess in der Musikgeschichte eine *optimale* Stimmung zu finden. Es wird klar, dass Tasteninstrumente mit ihrer Fixierung auf die 12-stufige Tonleiter, also mit ihren 12 Tasten pro Oktave dabei das Instrument mit der schwerwiegendsten Einschränkung darstellen. Vielfach wurde daher auch versucht nicht die Stimmung, sondern das Instrument anzupassen.

31-stufige Temperierung

Die 31-stufige Temperierung baut auf 30 mitteltönigen Quinten (vgl. Abschnitt 3.2.4) auf, diese bilden die Intervalle von **Geses**⁸ bis **Aisis**⁹. Das verbleibende Intervall **Aisis/Geses** ist eine um nur 1 Cent zu hohe, also fast reine Quinte. Durch diese Anordnung können die Quinten eine Oktave füllen, ohne dass dabei ein Komma entsteht. Diese Art der Temperierung kommt auf Instrumenten mit mindestens 31 Tasten zum Einsatz. Bereits im 16. Jahrhundert kam diese Art der Stimmung auf Instrumenten zum Einsatz, 1966 wurde sie vom niederländischen Musiker Adriaan Fokker in seinem Werk *Neue Musik mit 31 Tönen* weiter untersucht [19].

Cembalo universale, Cimbalo cromatico

Das Cembalo universale oder auch Cimbalo cromatico ist ein modifiziertes Cembalo aus der Zeit zwischen 1450 und 1700. Das Instrument zählt zu der Gruppe der *subsemitonischen* Instrumente, also Tasteninstrumenten mit unterteilten (gebrochenen) Halbtonschritten [20]. Sinn dieser gebrochenen Obertasten war es die enharmonische Verwechslung zu vermeiden. Die Modulation zu verschiedenen Tonarten in mitteltöniger Stimmung wird so ohne Auftreten der Wolfsquinte möglich. Ein Original des Instruments existiert heute nicht mehr. Nach Beschreibungen des

⁸ Das **Geses** ist ein zweifach erniedrigtes, also um zwei Halbtöne niedrigeres **G**.

⁹ Das **Aisis** ist ein zweifach erhöhtes, also um zwei Halbtöne höheres **A**.

Komponisten Michael Praetorius von 1619 verfügte das Instrument zusätzlich zu den fünf geteilten Obertasten (Cis/Des, Dis/Es, Fis/Ges, Gis/As und Ais/B) auch noch über Zusatz-Tasten für Eis und His. Eine Oktave verfügte so insgesamt über 19 Töne. [21].

Archicembalo

Um 1555 entwickelte der Musiktheoretiker Nicola Vicentino das Archicembalo, ein Cembalo mit insgesamt 36 Tasten pro Oktave, verteilt auf zwei Manuale. Ähnlich wie das Cembalo universale will auch das Archicembalo beliebige Modulationen der Grundtonart ermöglichen, indem es zwischen chromatischen Halbtönen über zwei zusätzliche Tonstufen (Tasten) und zwischen diatonischen Halbtönen ebenfalls über zwei zusätzliche Tonstufen (Tasten) verfügt. Das Instrument wurde mitteltönig temperiert, so gelang es Vicentino mit dem Ausnützen von „nur“ 31 Tönen pro Oktave fast ausschließlich reine Quinten zu erhalten (mit einer um 1 Cent verstimmtm Wolfsquinte). Durch diese Stimmung besaßen fünf Tasten pro Oktave die gleiche Tonhöhe. Abbildung 3.9 stellt die Tastatur des Archicembalos, mit Tonbezeichnung und Stimmung (in Cent zum Oktaven-Grundton), so dar, wie sie ursprünglich angedacht war [22].

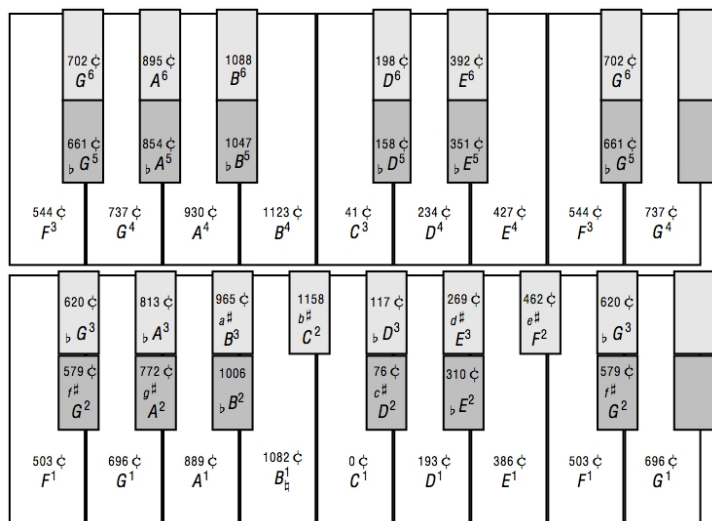


Abbildung 3.9: Archicembalo-Tastatur

Clavemusicum omnitonum

Das Clavemusicum omnitonum wurde 1606 als Auftragswerk des Herzogs von Novellara (Italien) von Vitus de Trasuntinis erbaut. Ähnlich wie auch das Archicembalo verfügte es über 31 Tonstufen auf 31 Tasten. Ziel des Instruments war es ebenfalls neben chromatischen auch diatonisches und enharmonisches Spielen zu ermöglichen. Das Instrument verfügte über zwei Reihen von Obertasten, in der ersten Reihe befanden sich sieben Tasten für die Ganztöne des Oktavenraums, darüber in zwei bzw. vier Reihen nochmals 24 Obertasten für die weitere Unterteilung der Halbtonschritte [23].

Enharmonicum

Ein anderes Ziel als die beschriebenen Cembali-Modifikationen der Renaissance verfolgte das Enharmonicum des japanischen Musiktheoretikers Tanaka Shōhei. Die reine Stimmung sollte möglichst akkurat auf einem Tasteninstrument realisiert werden. Das Enharmonicum verfügte über 20 Tasten aber 26 Tonhöhen pro Oktave. Per Fußtritt war es dem Spieler möglich den

erzeugten Klang der Obertasten von erhöht gestimmten Halbtonschritten (\sharp) zu erniedrig gestimmten (\flat) umzuschalten [24].

Clavichord

Zu den ältesten Tasteninstrumenten zählt das Clavichord, besonders im 17. und 18. Jahrhundert stand es auf dem Höhepunkt seiner Entwicklung. Mit *Tangenten*, die an den Tasten befestigt sind, wird eine gespannte Saite abgeteilt und zum Schwingen gebracht. Dabei schwingt aber nur ein Teil der Saite, der andere wird mit Filzpolstern fixiert. Die Tangente erfüllt dabei einen ähnlichen Zweck wie ein Steg.

Besondere Beachtung muss dem Clavichord geschenkt werden, da es als einziges Tasteninstrument über die Möglichkeit verfügt den Ton auch noch nach dem Tastenanschlag zu beeinflussen. Durch die Veränderung des Tastendrucks wird die Saitenspannung variiert und es entsteht ein dem Vibrato sehr ähnlicher Effekt. Die Tonhöhe des Clavichords ist somit also nicht vollkommen genau festgelegt [25].

Fokker-Orgel

Adriaan Fokker, ein niederländischer Musiker und Physiker, konstruierte mehrere Instrumente mit generalisierten Klaviaturen. Generalisierte Klaviaturen durchbrechen das Prinzip der typischen Anordnung von zwölf Tönen als sieben weiße und fünf schwarze Tasten. Durch eine kachelförmige Anordnung von rechteckigen Tasten entsteht Raum, um auch mikrotonale Skalen, also mehr als die zwölf Tonschritte der chromatischen Skala, anzuordnen. Das bekannteste von Fokkers Instrumenten ist die Fokker-Orgel, die über eine Klaviatur mit 31 gleichstufung verteilten Tönen pro Oktave verfügt. Abbildung 3.10 zeigt das Anordnungsschema der Tasten auf der Klaviatur. [26], [19]

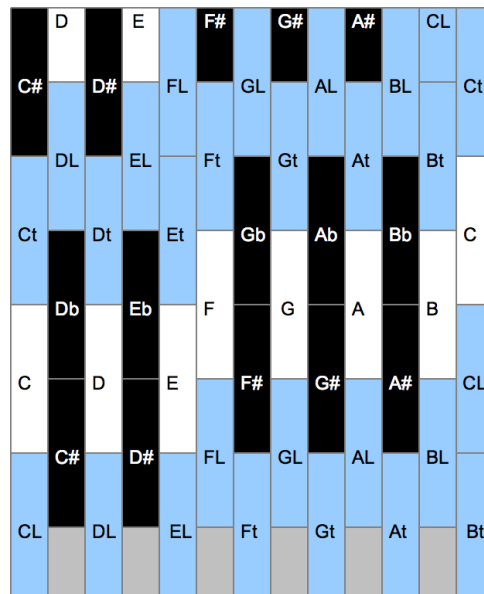


Abbildung 3.10: Schema der Tastenanordnung auf der Fokker-Orgel.

Bildnachweis: Hyacinth

Wikimedia Commons, lizenziert unter CreativeCommons-Lizenz by-sa-2.0-de.

MUTABOR

Das MUTABOR (*MUT*ierende *Automatisch Betriebene OR*gel)-Instrument ist ein seit 1980 an der Technischen Hochschule Darmstadt entwickeltes Versuchsinstrument für musiktheoretische Untersuchungen. Das Prinzip der *mutierenden Stimmungslogik* ist, dass erst nach betätigen einer Taste an der Klaviatur die entsprechende Tonhöhe auf Basis der vorangegangenen Spielfolge bzw. der anderen momentan betätigten Tasten berechnet wird. Mit anderen Worten: die Vergangenheit ist für die gegenwärtige Stimmung maßgebend. [27]

Die MUTABOR besteht aus einer Klaviatur, Synthesizer, Computer, Verstärkern, Lautsprechern und Software zur Steuerung des Systems. Das System ist dabei so konzipiert, dass die Berechnung der Tonhöhe in einer Geschwindigkeit erfolgt, in der keine hörbare Verzögerung wahrgenommen werden kann. Wie genau die Berechnung der Tonhöhe erfolgt, kann dabei vom Musiker selbst festgelegt werden. Die dafür vorgesehene „Programmiersprache“ wurde, nach anfänglicher Forschung, mit Musikern entwickelt und an deren Bedürfnisse angepasst [27].

Hermode-Orgel

Die Hermode-Orgel ist ein Instrument, welches in den vergangenen Jahren in Bad Liebenwerda (Deutschland) entwickelt wurde. Elektromagnetische Antriebe sind in der Lage die Pfeifen der Orgel in kürzester Zeit umzustimmen [28]. Gesteuert werden die Antriebe von einem Algorithmus, der die auf dem Manual betätigten Tastenkombination auswertet, dadurch die Tonart erkennt und eine dafür vorgesehene Stimmung an die Antriebe sendet. Diesem Algorithmus liegt das Hermode-Tuning zugrunde, eine Art des adaptiven Tunings. Auf den Algorithmus selbst wird in Abschnitt 3.4.2 näher eingegangen. Die Hermode-Orgel ist hier stellvertretend für eine ganze Reihe an Instrumenten aufgelistet, denen Hermode-Tuning zugrunde liegt.

Speziell bei der Orgel ist eine mikrotonale Veränderung der Stimmung aufgrund der verschiedenen Bauarten der Pfeifen schwierig. Es ist in der Literatur nicht genauer beschrieben, für welche Pfeifenarten die veränderbare Stimmung in der Hermode-Orgel geeignet ist.

3.2.12 Mikrotonale Musik

Mikrotonale Musik baut auf der Verwendung von *mikrotonalen Intervallen* kleiner als ein Halbtonschritt auf. Nicht zwangsweise muss eine Oktave dafür mehr als zwölf Töne aufweisen, entscheidend ist die Verwendung eines mikrotonalen Tonsystems.

Bereits im 11. Jahrhundert finden sich Nachweise der Verwendung von mikrotonalen Intervallen. So weisen aufgezeichnete Stücke gregorianischen Gesangs mikrotonale Vorzeichen auf. Erst das 20. Jahrhundert gilt, in der mikrotonalen Musik, als Beginn der Erweiterung des westlichen Tonsystems um mikrotonale Intervalle. In anderen Regionen der Erde sind solche Mikrotöne fester Bestandteil im Tonsystem, zu diesen Vertretern ausserhalb der westlichen Musik zählen die indische Gamelan-Musik oder die klassische arabische Musik.

Bedeutende Forscher auf dem Gebiet sind Adriaan Fokker (vgl. Abschnitt 3.2.11) oder Harry Partch, der ein Tonsystem mit 43 Tönen pro Oktave entwickelte [29].

3.2.13 Zusammenfassung

Anhand von Tabelle 3.13 werden die Unterschiede der einzelnen Stimmungen deutlich. Aus den Betrachtungen in Abschnitt 2.2.4 geht hervor, warum gerade die Oktave als einziges Intervall (niedriger Ordnung) immer rein gestimmt wird. Mit der Ordnung des Intervalls nimmt also auch das Maß an Akzeptanz von Verstimmung zu, weswegen nach der Oktave den Quinten der höchste Stellenwert bei der Stimmung zukommt.

Intervall	reine Intervalle	mitteltönige Stimmung	Werckmeister III	gleichstufige Stimmung
Prime	0	0	0	0
kl. Sekunde	70 (cis) / 112 (des)	76 (cis)	90 (des)	100
gr. Sekunde	204	193	192	200
kl. Terz		310 (es)	294 (es)	300
gr. Terz	386	386	390	400
Quarte	498	503	498	500
Tritonus	590 (fis) / 610 (ges)	579 (fis)	588 (ges)	600
Quinte	702	697	696	700
kl. Sexte	772 (gis) / 814 (as)	773 (gis) / 814 (as)	792 (as)	800
gr. Sexte	884	890	888	900
kl. Septime	976 (ais) / 996 (b)	1007 (b)	996 (b)	1000
gr. Septime	1088	1083	1092	1100
Oktave	1200	1200	1200	1200

Tabelle 3.13: Vergleich der Centzahlen verschiedener Stimmungen, aus [17].

In der heute weitgehend gebräuchlichen gleichstufigen Stimmung werden alle Intervalle hörbar unrein verstimmt. Trotz dieser Verstimmung werden sie dennoch als akzeptabel empfunden, was auf Hörgewohnheiten zurückzuführen ist. Ross W. Duffin geht davon aus, dass heutige Musiker die Verstimmung z. B. einer großen Terz nicht wahrnehmen (Konditionierung), aber sie auch nie rein gehört haben (Ignoranz) [30, S. 30].

Weiters ist in der musikalischen Praxis auch von *theoretisch richtiger* und *musikalisch richtiger* Stimmung die Rede. Die Ausführungen dieses Kapitels zeigen, dass es die richtige Temperierung aber nicht geben kann und die Temperierung lediglich ein Angebot darstellt, das Ohr aber entscheidet [31, S. 7].

*"Es führt also die gleichschwebende Temperatur
ihre Bequemlichkeit und Unbequemlichkeit
mit sich, wie der liebe Ehestand."
– Johann Georg Neidhardt [15, S. 40]*

3.3 Inharmonizität

Wie im vorangegangenen Abschnitt ausgeführt wurde, erzeugen Instrumente, abgesehen von rein elektronischen, immer ein Tongemisch, also einen Grundton und mehrere Obertöne (vgl. Abschnitt 2.2). In der Regel sind diese Obertöne harmonisch, also ganzzahlige Vielfache der Grundfrequenz, und folgen damit Gleichung 3.14, wobei f_n die Frequenz des Teiltons, n seine Ordnung und f_1 die Grundfrequenz bezeichnet [32]. Bei Saiteninstrumenten kommt es allerdings zur Inharmonizität. Die Inharmonizität beschreibt den Effekt der Frequenzerhöhung von Teiltönen aufgrund der Steifigkeit einer Saite.

$$f_n = n \cdot f_1 \quad (3.14)$$

Mathematisch erfasst ist diese Frequenzerhöhung durch die *Youngsche* Formel¹⁰ (siehe Gleichung 3.15), die die Abweichung E des Teiltones n von seiner harmonischen (Soll-)Frequenz in Cent beschreibt. Dabei stellt d den Saitendurchmesser in cm, l die Saitenlänge in cm, f die Frequenz des Grundtons in Hz und K eine Materialkonstante in Cent·cm²·Hz² dar. (Ein typischer Wert, der nach Material nur geringfügig variiert, ist $3,3 \cdot 10^{13}$ [10].)

$$E = \frac{K \cdot d^2}{l^4 \cdot f^2} \cdot n^2 \quad (3.15)$$

Aus der *Youngschen*-Formel wird ersichtlich: je größer der Saitendurchmesser ist desto steifer und somit inharmonischer werden die Teiltöne, je dünner der Saitendurchmesser ist, desto elastischer und harmonischer werden die Teiltöne. Für die Betrachtung von Instrumenten unter den Aspekten der Inharmonizität kann man auch einen Blick auf die Bauform werfen, die meistens direkt mit der Saitenlänge zusammenhängt.

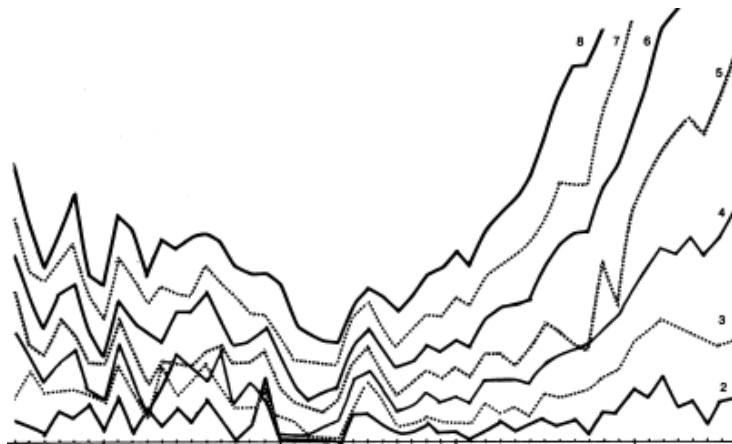


Abbildung 3.11: Inharmonizitätskurve eines Kleinklaviers, aus [31].

Die größte Saitensteifigkeit wird also im Bereich von Bass (größerer Saitendurchmesser) und Diskant (kürzere Saiten) erreicht, die Inharmonizität ist in diesen Bereichen am größten. Abbildung 3.11 zeigt die Inharmonizitäts-Kurve eines Kleinklaviers. Die X-Achse beschreibt die Töne von Bass nach Diskant, die Y-Achse die Abweichung der Teiltöne von dem Stimmwert, der sich bei harmonischen Verhältnissen ergeben würde. Die einzelnen Kurven stellen die Abweichung vom zweiten bis zum achten Teilton dar.

¹⁰ Benannt nach Robert W. Young.

3.3.1 Auswirkungen der Inharmonizität auf reine Intervalle

Allgemein gilt, dass bei reinen Intervallen auch die gemeinsamen Teiltöne gleich hoch und somit schwebungsfrei sind, also keinen Frequenzunterschied aufweisen. Diese Annahme kann aber nur für harmonische Teiltöne getroffen werden. Tabelle 3.14 zeigt die Verstimmung der Obertöne in Cent für den Grundton a' und den um eine reine Oktave höher gelegenen Ton a'' . Es wurden folgende Annahmen getroffen: $K = 3,3 \cdot 10^{13} \text{ Cent} \cdot \text{cm}^2 \cdot \text{Hz}^2$, $l = 41 \text{ cm}$ und $d = 0,0975 \text{ cm}$.

Teilton	Verstimmung (a')	Verstimmung (a'')
1	0,6 Cent	0,1 Cent
2	2,3 Cent	0,6 Cent
3	5,2 Cent	1,3 Cent
4	9,2 Cent	2,3 Cent
5	14,3 Cent	3,6 Cent
6	20,6 Cent	5,2 Cent

Tabelle 3.14: Verstimmung der Teiltöne von a' (440 Hz) und a'' (880 Hz) in Cent, nach Gl. 3.15.

Wie aus der Tabelle ersichtlich wird, trifft jeder gerade Teilton¹¹ auf einen Teilton des eine Oktave höher gelegenen Grundtons. Würde man die Stimmung verändern, um die ungeraden Teiltöne rein zu bekommen, wären die geraden wieder schwebend. Es ist also unmöglich Intervalle von Saiteninstrumenten rein zu stimmen. Die Definition von reinen Intervallen wird beim Aufkommen von Inharmonizität also hinfällig [31].

3.3.2 Streckung der Temperierung

Wie in Abbildung 3.11 zu sehen ist die Inharmonizität unterschiedlich stark ausgeprägt. Um dennoch befriedigende Intervalle zu erhalten, müssen auch die Oktaven verstimmt werden: in Mittellage um ca. zwei Cent, in Randlagen deutlich mehr, nämlich um bis zu ca. 30 Cent. Die Temperierung eines Klaviers wird deswegen „gestreckt“, d. h. die tiefsten Töne erklingen bis zu 30 Cent tiefer, die höchsten bis zu 30 Cent höher [31]. Man spricht von *Streckung* oder *Spreizung*.

3.3.3 Zusammenfassung

Das Spektrum von Instrumenten wie dem Klavier ist typischerweise obertonlastig (mehr Energie in den Teiltönen als im Grundton). Vor allem in den Bässen enthalten die Grundtöne wenig Energie. Im Rückblick auf Abschnitt 2.2.1 wird die Tonhöhe vom Gehirn mit Hilfe der Obertöne ermittelt, was am Beispiel des Klaviers zu einer Tonhöhenwahrnehmung führt, die vom eigentlichen Grundton abweicht. Das Spektrum der Orgel hingegen ist am Grundton energetisch stärker ausgeprägt, auch kommt es bei Orgelpfeifen nicht zur Inharmonizität¹², Orgeln sind also mischungsfähiger [33].

Die Inharmonizität soll die Frage der Reinheit von Intervallen neu aufwerfen, sie ist sowohl ein Nebeneffekt des Materials als auch Gestaltungsparameter.

¹¹ 2, 4, 6, 8, ...

¹² Prinzipiell tritt der Effekt bei jedem schwingenden Körper auf, bleibt aber aufgrund von Steifigkeit und Dominanz der Grundschwingung weit unter einem wahrnehmbaren Bereich.

3.4 Adaptives Tuning

Aus den vorangegangenen Abschnitten wird klar, dass egal welche Temperierung gewählt wird, immer Kompromisse eingegangen werden müssen. Was aber würde passieren, wenn die Töne einer Skala nicht fix festgelegt wären, also die Möglichkeit besäßen zu variieren?

Da der Umfang der Klaviaturen immer wieder als eine der schwerwiegendsten Beschränkungen auftaucht und es eine Vielzahl an gleichzeitig verfügbaren Tasten erfordern würde um alle Intervalle in sämtlichen Tonarten rein erklingen zu lassen, kam die Idee auf ein Umstimmen der Töne während des Spielens zu ermöglichen. Dieser Lösungsansatz geht bereits auf die Überlegungen der Pythagoreer zurück, wurde aber im Besonderen erst von Martin Vogel im letzten Jahrhundert wieder aufgegriffen, der vorschlag die Tonhöhen der üblichen 12er-Temperierung über eine Automatik einstellen zu lassen [27].

Auch hier gilt es wieder zwischen Stimmung und Intonation zu unterscheiden. Eines der Ziele von dynamischer Intonation ist die Beeinflussung der Klangfarbe, um maximale Konsonanz zu erhalten (vgl. Abschnitt 2.2.5 auf Seite 21). Adaptives Tuning bezeichnet die Adaption, also die Anpassung der Stimmung basierend auf bestimmten Parametern, in der Regel ist damit die Reaktion auf die gespielten Noten und das Erkennen der Tonart verbunden. Adaptives Tuning ist ein Vorgang, der in Echtzeit erfolgen muss, im Gegensatz zur Intonation, die vorweg auch ohne musikalischen Bezug zum Stück erfolgen kann.

Dynamische Intonation soll aber nicht nur das Problem der reinen Intervalle lösen, eine weitere Motivation liegt im Einsatz als Gestaltungsparameter. Musiker wie Streicher oder Bläser intonieren nicht immer gleich, es kommt während des Spielens immer wieder zu Abweichungen. Diese Abweichungen sind zum Großteil beabsichtigt, um der Musik eine gewisse „Lebendigkeit“ zu verleihen¹³. Adaptives Tuning eröffnet auch die Möglichkeit diese statistisch erfassbaren Abweichungen zu erkennen und zu reproduzieren, so dass sie bei Tasteninstrumenten (z. B. elektronischen mit stets fixer Tonbildung), die dieses einsetzen, als Stilmittel genutzt werden könnten [9]. Auch im Bereich der elektronischen Orgeln sind geringe Stimmaabweichungen ein zum Einsatz kommendes Stilmittel. Der amerikanische Orgelbauer Rodgers Instruments sowie der Orgelsampler *Hauptwerk* der Firma Milan Digital Audio implementieren ein statistisches Detune, welches die Töne während des Spiels mikrotonal verstimmt.

3.4.1 Ansätze für adaptives Tuning

Einer der einfachsten Ansätze für adaptives Tuning, der nicht einmal eines Computers oder eines besonderen Mechanismus bedarf, ist, das Instrument manuell für einzelne Teile eines Stückes neu zu stimmen. Es existieren mehrere historische Quellen, dass gerade diese Technik immer wieder zum Einsatz kam. Besonders Cembalo-Spieler stimmten ihre Instrumente während eines Stückes (zwischen den Sätzen) häufig um [9]. In diesem Zusammenhang kann auch das Enharmonicum (vgl. Abschnitt 3.2.11 auf Seite 40) erwähnt werden, welches einen eigenen Hebel bereitstellte, um die Zuordnung der Tasten zu den Tönen, also die Stimmung, während des Spielens zu verändern.

Im Jahr 1979 stellte der Komponist und Musiktheoretiker Harry Partch die These auf, dass einmal ein Instrument existieren könnte, welches in der Lage wäre seine Stimmung frei zu verändern und seinem gesamten Stimmungsumfang zu nutzen. Partch beschäftigte sich, im 20. Jahrhundert, als einer der ersten mit mikrotonaler Musik und schrieb Stücke für seine selbstgebauten Instrumente, die in seinen eigenen Tonsystem gestimmt waren und über eine reine Stimmung verfügten, die auf ganzzahligen Intervallen beruhte. Sein Ansatz für adaptives Tuning ist die Messung von Konsonanz und ihre Darstellung in Form einer Metrik bzw. Abstandsfunktion, um die Tonhöhen während einer Darbietung in Echtzeit zu verändern [29].

¹³ Ein andere Ursache für die Abweichungen ist die Schwierigkeit einen Ton immer wieder Cent-genau zu treffen.

$$D_F(\alpha)$$

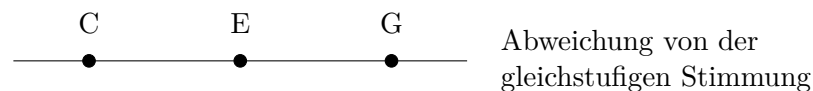
beschreibt so eine Funktion, wobei D die Entfernung der Tonhöhe (eines Tones) zur Konsonanz und F das Spektrum der Intervalle α darstellt. Auf der Abstandsfunktion existieren lokale Extremstellen, ihre Minima sind Stellen, an denen Intervalle auftreten, die maximale Konsonanz besitzen. Beim adaptiven Tuning gilt es nun den Gradienten dieser Dissonanz-Stellen zu berechnen und die Tonstufen bis zum nächstgelegenen Minimum der Funktion zu verstimmen [9]. Zwei mögliche Implementationen, die Partchs Idee weiter entwickelt haben, sind das Hermode-Tuning sowie das Spring-Tuning.

3.4.2 Hermode-Tuning

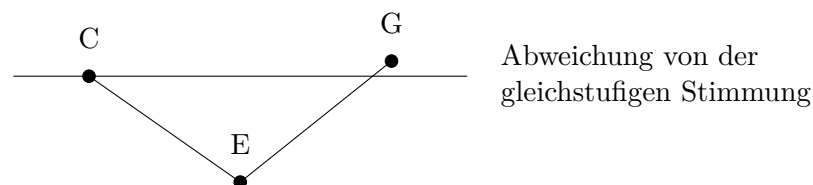
Das Hermode-Tuning, oder auch HMT, wird seit 1988 von Werner Mohrlök mit dem Ziel die Fehler der gleichstufigen Stimmung auszugleichen und Instrumente in Echtzeit zu verstimmen entwickelt.

Um mit herkömmlichen Instrumenten, also Instrumenten ohne dynamischem Tuning, spielbar zu bleiben, verändert das HMT die absoluten Tonhöhen, so dass die Summe der Verstimmung innerhalb einer Oktave immer null ergibt. Nachfolgendes Beispiel erläutert die Funktionsweise des HMT-Algorithmus:

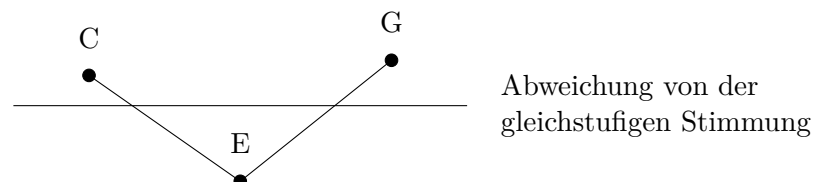
- (i) Der Akkord C-E-G wird angeschlagen.



- (ii) Der Algorithmus analysiert die angeschlagenen Tasten und berechnet, welche Verstimmung nötig ist, um das Intervall rein erklingen zu lassen. Dabei wird von der gleichstufigen Stimmung ausgegangen. Im Beispiel wäre eine Verstimmung des E um -14 Cent sowie des G um $+2$ Cent notwendig.



- (iii) In einem dritten Schritt wird berechnet, wie die Verstimmung ausgeglichen werden muss, damit die Summe innerhalb einer Oktave null ergibt. Im konkreten Beispiel wären dies 4 Cent.



Das Hermode-Tuning kann in verschiedenen Modi aktiviert werden:

- Anpassung von Terzen und Quinten.

- Anpassung von Terzen, Quinten und Quartan.
- Anpassung der Stimmung durch Analyse des harmonischen Mittelpunktes eines Stückes. Dies erfolgt basierend auf den letzten zehn angeschlagenen Akkorden, die weiteren Stimmungsanpassungen nehmen nicht mehr die gleichstufige Stimmung als Ausgangspunkt, sondern verschieben die Stimmung um den berechneten Mittelpunkt.
- Anpassen der Stimmung als Tiefenparameter zwischen zwei definierten Stimmungsextremen.

Dabei kann für alle Modi auch die „Aggressivität“ festgelegt werden, diese bestimmt, wie schnell und wie stark der Algorithmus in die Stimmung eingreift. Das Hermode-Tuning kann auch als Referenztemperierung für Orchester verwendet werden [34]. Anwendung findet das Hermode-Tuning in Programmen wie Apples Logic Pro, Capella, Steinbergs Cubase und anderen. Zum Einsatz in Instrumenten kommt es bei elektronischen Orgeln der Firma Content Organs und in der bereits beschriebenen Pfeifenorgel in Bad Liebenwerda (vgl. Abschnitt 3.2.11).

3.4.3 Spring-Tuning

Auf den ersten Blick scheint den Ansätzen zum adaptiven Tuning ein einfaches Prinzip zugrunde zu liegen, tatsächlich ergeben sich aber auch mit diesen Probleme. Angenommen der Akkord C-D-G-A soll gespielt werden und rein gestimmt sein. Die Stimmung der Quinten kann einfach realisiert werden mit 702 Cent je Quinte (C mit 0 Cent, D mit 204 Cent, G mit 702 Cent und A mit 906 Cent). Die große Sexte C-A sollte, um rein zu klingen, eine Größe von 884 Cent aufweisen: genau hier tritt das Problem als pythagoreisches Komma auf denn vier Quinten ergeben keine große Terz ($884 \text{ Cent} \neq 906 \text{ Cent}$). Reine Intervalle sind also selbst bei adaptivem Tuning unmöglich zu realisieren.

Um hier einen Kompromiss zu finden, der die Verstimmung insgesamt möglichst gering hält und sich an den jeweils gespielten Akkord anpasst, wurde von John de Laubenfels um 2000 das Spring-Tuning entwickelt. Das Spring (engl. *Feder*) Tuning beruht auf einem Modell mit so genannten Stimmfedern für alle reinen Intervalle. Abbildung 3.12 zeigt, wie immer eine dieser Federn zwei Noten miteinander verbindet.

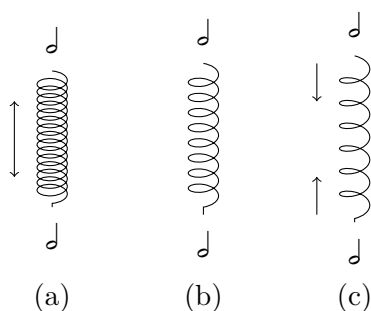


Abbildung 3.12: (a) Komprimierte Feder drückt sich auseinander, (b) ruhende Feder, (c) gestreckte Feder zieht sich zusammen.

Die Federn befinden sich in Ruhelage, wenn das Intervall rein gestimmt ist. Wenn das Intervall größer ist als seine reine Stimmung, zieht es die Feder zusammen, ist es kleiner, drückt es die Feder auseinander. Wird ein Akkord gespielt, werden alle betroffenen Noten mit diesen Stimmfedern verbunden und der Algorithmus berechnet das Verhalten des Federnsystems. Sobald das System einen eingeschwungenen Zustand erreicht hat und alle Federn zum Stillstand gekommen sind, wurde eine Kompromisstimmung gefunden. Die verbundenen Federn bilden ein *chaotisches System*. Das bedeutet, dass die zeitliche Entwicklung des Systems unvorhersehbar ist, obwohl seine Komponenten auf deterministischen Gleichungen beruhen. Jeder Übergang

zwischen Tönen fordert vom System ein anderes Verhalten und wird vom vorhergehenden Verhalten beeinflusst. Abbildung 3.13 zeigt das System von Federn für das anfängliche Beispiel des Akkords C-D-G-A.

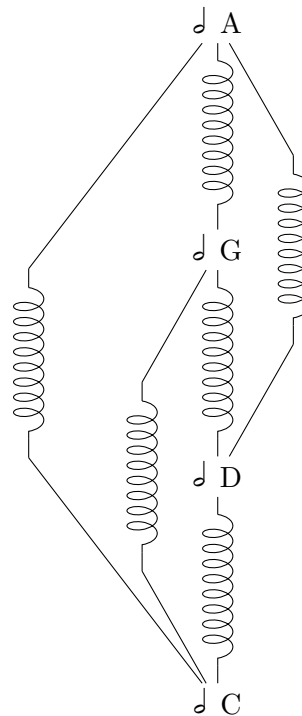


Abbildung 3.13: Durch Federn verbundener Akkord. Die Federn ziehen und drücken. Ein Stimmungskommiss stellt sich ein.

Das Spring-Tuning berechnet die Verstimmung der Quinten und Terzen im angeführten Beispiel. Welche Intervallgröße diese genau annehmen, lässt sich nicht voraussagen, sondern ist abhängig von den einzelnen Federkonstanten – also der Kennzahl, wie leicht die Feder ihre Größe verändert. Diese Federkonstante ändert sich während des Spielens. De Laubenfels geht davon aus, dass das Verstimmen einen bestimmten *Schmerz* verursacht, dieser Schmerz in Cent ist proportional zum Quadrat der Stimmungsänderung einer Note. Ziel ist es nun während des Spielens diesen Schmerz möglichst gering zu halten. Insgesamt werden drei Arten von Schmerz unterschieden:

- (i) Die Verstimmung von zugleich erklingenden Noten (während des Spielens).
- (ii) Die unterschiedliche Stimmung einer Note zu verschiedenen Zeitpunkten. (Für den Fall, dass eine Note in verschiedenen Intervallen, also musikalischen Kontexten auftritt.)
- (iii) Die Verstimmung des gesamten Systems nach oben oder unten.

Das Spring-Tuning versucht diesen Schmerz auf die gleiche Weise zu verarbeiten wie die Adaptierung der Stimmung. Für all diese Arten von Schmerz wird eine neue Feder eingeführt, die wiederum eine spezifische Federkonstante besitzt. So werden über die einzelnen Adaptierungen hin alle drei Arten von Schmerz möglichst gering gehalten. [9]

3.4.4 Zusammenfassung

Adaptives Tuning bringt eine Reihe von Lösungsvorschlägen und Möglichkeiten Stimmungsprobleme zu lösen; dennoch ist es keine Universallösung, es bleiben Schwierigkeiten bestehen, die

im Aufbau des westlichen (chromatischen) Systems beheimatet sind und auch nicht endgültig gelöst werden können.

Der Vorteil adaptiven Tunings ist klar die Möglichkeit über viele Tonartarten modulieren zu können und trotzdem überwiegend reine, also schwebungsfreie, Klänge zu erhalten. In digitalen Instrumentenschnittstellen wie MIDI sind bereits Befehle verankert Stimmungsinformationen auszutauschen und viele Instrumente sind in der Lage diese Information auszuwerten und ihre Stimmung entsprechend anzupassen. Geht es um die Realisation adaptiven Tunings, sind meistens die klassischen (nicht elektronischen) Instrumente ein Hindernis, sie legen die Schwierigkeit an den Tag nicht mit einfachen Mitteln ihre Stimmung ändern zu können. Dennoch existieren auch verschiedene Vorreiterinstrumente, die sich dieses Problems annehmen, eines davon ist die Hermode-Orgel in Liebenwerda (vgl. Abschnitt 3.2.11).

Die hier vorgestellten Varianten adaptiven Tunings bauen alle darauf auf möglichst reine Intervalle zu bilden; damit werfen sie aber auch die Frage auf, ob reine Intervalle auch wirklich das geforderte Ziel sind. Schon Abschnitt 2.2.5 verweist auf den gezielten Einsatz von Konsonanz als musikalischen Gestaltungsparameter. Zudem würde ein System, dessen Ziel es ist stets reine Intervalle wiederzugeben, zu Stimmungsänderungen führen, die vom Spieler nicht mehr kontrolliert werden können. Wie es beim Spring-Tuning, einem chaotischen System, der Fall ist, sind Stimmungsänderungen auch nicht eindeutig reproduzierbar, was ebenso ein Nachteil ist wie der Verlust der direkten Kontrolle.

In der vorliegenden Arbeit werden einige ausgewählte Temperierungen beschrieben, die das Problem der reinen Intervalle zu lösen versuchen. Da die Interpretation eines Stückes, gerade aus der Perspektive der historisch informierten Aufführungspraxis, nach einer zeitgetreuen Temperierung streben kann, stellt sich die Frage: Wieso soll ein Computer also an einer Optimierung arbeiten, die vielleicht nicht gewünscht ist? Der Organist selbst kann ebenso eine Entscheidung darüber treffen, welche Stimmung er gedenkt einzusetzen und wann er eine dynamische Stimmungsänderung als Gestaltungsparameter gebraucht.

Das Zentrum für Orgelforschung am Institut für Kirchenmusik und Orgel der KUG¹⁴ beschäftigt sich mit der Steuerung von Klangparametern, wie es auch die Stimmung einer ist, über Interfaces wie Schweller, Aftertouch oder Körpertracking. Die Kontrolle über ein dynamisches Tuning dem Organisten zu übergeben und ihm damit die Möglichkeit zu verschaffen es wie einen Gestaltungsparameter einzusetzen, ist ein dem hier beschriebenen adaptiven Tuning-System entgegen gerichteter Aspekt. Das nachfolgende Kapitel beschreibt ein solches System.

¹⁴ Universität für Musik und darstellende Kunst Graz

4

Dynamische Intonation digitaler Konzertorgeln

Während Kapitel 2 die musikalischen und psychoakustischen Grundlagen der Tonhöhenwahrnehmung umriss, wurden in Kapitel 3 die Probleme von reinen Intervallen an Tasteninstrumenten verdeutlicht. Dasselbe Kapitel enthält auch Lösungsansätze wie die Erweiterung der Klaviatur auf mehr denn 12 Tasten oder computergestützte Systeme zur Stimmungsnachführung – adaptives Tuning.

Kapitel 4 beschreibt nun ausgehend von den vorgangegangenen ein im Zuge dieser Arbeit entwickeltes Laborsystem, welches die dynamische Intonation von digitalen Konzertorgeln ermöglicht und zur Erforschung der daraus resultierenden Möglichkeiten dient. Zudem bringt das System einen pädagogischen Mehrwert mit sich: Stimmungssysteme und ihre Abweichungen von reinen Intervallen können im Unterricht so leicht veranschaulicht werden. Dabei musste zuerst festgestellt werden, welche dynamische Änderung der verschiedenen Intonationsparameter an digitalen Konzertorgeln möglich ist und wie diese Änderung umgesetzt werden kann. Ziel war es die dynamische Intonation während des Spiels als, künstlerischen Gestaltungsparameter zu ermöglichen.

Man unterscheide nochmals die Begriffe Intonation, Stimmung und Temperierung. Eine Temperierung versucht die Probleme reiner Intervalle möglichst gut zu kompensieren und führt durch die Anpassung der einzelnen Tonhöhen zu einer spezifischen Stimmung (vgl. Abschnitt 3.1). Stimmung, Lautstärken- und Klangfarbenanpassungen ergeben zusammen die Intonation.

Der Lautstärkenregulierung als Intonationsparameter liegt die Anpassung der Orgel an einen Aufstellungsort zugrunde, mit dem Ziel dass jede Taste innerhalb eines Registers einen Ton adäquater Lautstärke hervorbringt und Besonderheiten eines Raumes wie Moden ausgeglichen werden. Auch die Anpassung der Lautstärke der einzelnen Register wird bei der Intonation vorgenommen, dabei werden Zusammenklang und Kontrast der Register zueinander festgelegt.

Die Lautstärkenregulierung bei der Intonation steht im Unterschied zu der Lautstärkenregulierung, die der Organist durch die Bewegung der Schweller durchführt.

Die Klangfarbe ist ein Charakteristikum jedes Instrumentes. Wie auch in Abschnitt 2.2.4 angeführt, ist die Klangfarbe kein eindeutiger Parameter, sondern eine Summe von Klangeigenschaften, die von uns psychoakustisch verarbeitet werden. Sie ist für eine dynamische Änderung, wie sie in dieser Arbeit gefordert wird, ebenfalls nicht einfach realisierbar.

Die Stimmung als dynamisch veränderbarer Parameter bietet sich an, allerdings in einem mikrotonalen Ausmaß, wie sie bei der Temperierung von Instrumenten zum Einsatz kommt. Das Problem der reinen Intervalle soll und kann damit nicht gelöst werden. Wie im Abschnitt 2.2.5 (Seite 21 dargelegt, kann nicht davon ausgegangen werden, dass eine durchgehend reine Stimmung Ziel jeder Interpretation ist.). Durch eine einfach zu handhabende und schnelle Variation von Temperierungen wird es möglich die Stimmung des Instrumentes an einzelne Passagen eines Stückes anzupassen. Ausgehend davon entstand die Idee einen Stimmungsvektor zwischen zwei definierbaren Temperierungen aufzuspinnen und eine Bewegung auf diesem Vektor während des Spiels zu ermöglichen.

4.1 Laborsystem

Dem Beginn der eigentlichen Systementwicklung ging die Evaluierung voraus, welche Instrumente sich für eine dynamische Änderung der Stimmung eignen und wie ein System zur dynamischen Intonierung aussehen könnte. Untersucht wurden dabei sowohl die am Institut für Kirchenmusik und Orgel verfügbaren Instrumente als auch eventuelle Neuanschaffungen.

4.1.1 Die digitalen Orgeln am Institut für Kirchenmusik und Orgel

Das Institut für Kirchenmusik und Orgel wurde seit 2012 mit zwei digitalen Orgeln ausgestattet, die leicht unterschiedliche Konzepte verfolgen. Den Instrumenten gemein ist zum einen das Ziel eine mobile Orgel zu schaffen, zum andern die Möglichkeiten digitaler Klangerzeugung zu erforschen und Vorteile, die sich gegenüber konventionellen Pfeifenorgeln ergeben, bei der Wiedergabe auszunutzen.

Rodgers-Orgel

Die digitale Konzertorgel am Institut für Kirchenmusik und Orgel der Kunstuniversität Graz wurde 2012 von der Firma Rodgers als Spezialanfertigung gebaut und sticht durch Besonderheiten wie einen höhenverstellbaren Spieltisch und vieles mehr hervor. Institutsvorstand Gunther Rost beschreibt das damit verbundene Projekt wie folgt:

[...] we've tried to transfer the idea of the organ into the 21st century. Our goal is to empower a new generation of organists and composers to discover and develop new ways of seeing, playing and using the instrument in the future. [35]

Der Gesamtidee zugrunde liegt auch der Versuch einer räumlichen Transportierbarkeit. So ist das Instrument vollkommen mobil aufgebaut, kann innerhalb weniger Stunden in Einzelteile zerlegt, verpackt, transportiert und an anderer Stelle – beispielsweise in einem Aufführungsraum – wieder aufgebaut werden.

Das Herzstück dieser digitalen Konzertorgel ist der so genannte *Digital Cage*, ein Rechner im Spieltisch, welcher die Signalgenerierung anhand der gegebenen Parameter (Stellung der Registerzüge, Schwellerposition, Tastenanschläge, Temperierung, . . .) übernimmt. Dabei werden Samples von Pfeifen realer Orgeln herangezogen, Windbewegungen simuliert und vordefinierte Raumeinstellungen verwendet. Trotz des hohen Rechenaufwandes ist der Prozess echtzeitfähig und der resultierende Klang, bei richtiger Beschallung, kaum noch von einer realen Orgel zu unterscheiden [36].

Die erste digitale Orgel der Firma Rodgers wurde bereits 1993 vorgestellt und arbeitete mit digitalen Signalprozessoren von Roland. Über den tatsächlichen Aufbau des Rechenwerkes im Inneren ist jedoch aus Gründen des Produktschutzes und der Irrelevanz für die meisten Endanwender wenig bekannt. Der physikalische Klang entsteht dabei erst an einem Lautsprecherarray, welches sich in beliebiger Entfernung zum Spieltisch befinden kann. Durch die digitale Klangerzeugung entsteht die Möglichkeit verschiedene Signale unabhängig voneinander abzugreifen und so vom realen Aufstellungsraum für eine Aufnahme vollständig zu trennen. Damit ist die Orgel sowohl in ihrer wahrgenommenen physikalischen Größe als auch in dem mit ihr generierten Klang komplett auf verschiedene Orgelliteratur zum einen sowie verschiedene Raumsituationen zum anderen adaptierbar und gibt einen Blick auf neue Einsatzfelder und Aufnahmevarianten frei. Außerdem sind im System mehrere historische Stimmungstabellen hinterlegt, welche sich durch ein Menü an der Orgel bzw. ein Web-Interface auswählen lassen.

Viele Einstellungen der Orgel, aber auch Informationen wie Anschläge, derzeitige Registrierung usw. lassen sich per MIDI übertragen [36]. Obwohl das MIDI-Protokoll umfangreich implementiert ist, wird das MIDI-Tuning-System, ein Teil des MIDI-Protokolls, welches es erlauben



Abbildung 4.1: Die „Rodgers-Orgel“ bei einer Aufführung im Haus für Musik und Musiktheater (MUMUTH) der Universität für Musik und darstellende Kunst Graz.

würde Stimmton und Temperierung per MIDI festzulegen, nicht unterstützt. Ein Umschalten der Temperierung über das Web-Interface ist möglich, allerdings nicht unterbrechungsfrei. Aufgrund dieser Tatsachen aber auch der Unveränderlichkeit der implementierten Temperierungen kommt das Instrument für eine dynamische Änderung der Stimmung nicht in Frage.

Hauptwerk-Orgel

Als ultramobiles Instrument wurde 2014 eine weitere Orgel konzipiert. Diese besteht aus einem leicht zerlegbaren Spieltisch mit vier Manualen. Der Spieltisch ist (technisch betrachtet) lediglich ein MIDI-Interface, welches Informationen über Tastenanschläge und Betätigung der Steuerungsknöpfe bzw. Schweller an einen Rechner weiterleitet.

Der Kern des Systems ist ein leistungsstarker Rechner mit einem Windows-Betriebssystem, auf diesem Rechner läuft ein Programm namens *Hauptwerk*¹⁵, welches aus Samples von Pfeifenorgeln den Orgelklang an einem Audio-Interface bereitstellt. Die dabei verwendeten Samples müssen als separate *Software*-Pakete erworben werden und stammen von diversen Orgeln auf der ganzen Welt. Im Gegensatz zu Rodgers werden die einzelnen Samples aber nicht trocken aufgenommen, sondern von einer bestimmten Hörposition innerhalb des Aufstellungsraumes aus. Diese nassen Samples bringen also einen spezifischen Hall mit sich.

Auch Hauptwerk versucht bei seiner Klangerzeugung Eigenschaften von Pfeifenorgeln nachzuahmen, dazu gehören Windschwankungen aber auch Details wie die Geräusche, die beim Ziehen der Register oder Betätigen der Klaviatur auftreten.

In der aktuellen Version der Hauptwerk-Software kommen Release-Trigger-Samples zum Einsatz; in Abhängigkeit zum vorangegangenen Ton sowie der Dynamik des Tastenanschlags werden hier unterschiedliche End-Samples verwendet, die nach Erklingen des Tones unterschiedliche

¹⁵ Um Verwechslungen mit dem Orgelwerk Hauptwerk zu vermeiden wird Hauptwerk im folgenden als Hauptwerk-Software bezeichnet.



Abbildung 4.2: Die „Hauptwerk-Orgel“ des Instituts für Orgel und Kirchenmusik bei einem Konzert im Sternenturm Judenburg.

Photo von Wolfgang Spekner für das Institut für Kirchenmusik und Orgel.

Originalakustiken des Raumes einfügen. Diese Technologie erzeugt zusätzlich eine hohe Authentizität des Klanges¹⁶.

Die Bedienung der Hauptwerk-Software kann zu einem Großteil über MIDI erfolgen, viele der Anweisungen sowie Schaltflächen können mit einem beliebigen MIDI-Kommando hinterlegt werden. Außerdem unterstützt Hauptwerk das MIDI-Tuning-System zum Teil, so kann beispielsweise der Stimmton (Normalton) der Orgel¹⁷ über MIDI geändert werden.

Hauptwerk ermöglicht es dem Benutzer weitere Temperierungen im XML-Format zu erzeugen. Eine beliebige Anzahl dieser Temperierungen kann in einen eigenen Setzer geladen und während des Spiels vom Organisten abgerufen werden. Das Umschalten zwischen den einzelnen Temperierungen erfolgt dabei unterbrechungsfrei. Dadurch eignet sich Hauptwerk besonders für dynamische Intonation, bringt allerdings den Nachteil mit sich, dass die Schnittstelle dieser Stimmungsbearbeitung auf Dateiebene basiert und keine Rückmeldung bereitstellt.

4.1.2 Nicht verfügbare Systeme

Weiters wurden andere Softwaresampler und Expander betrachtet. Synthesizer-Systeme wurden weggelassen, da sie am Institut für Kirchenmusik und Orgel der KUG¹⁸ nicht in Verwendung sind und sich das zu entwickelnde System in die Instrumentenlandschaft des Instituts integrieren lassen sollte.

Grand Orgue

Grand Orgue ist ein Software-Sampler, ein eigenständiges Programm, das im Gegensatz zur Hauptwerk-Software auf allen gängigen Betriebssystemen (Windows, Linux, OSX) läuft und un-

¹⁶ Weitere Informationen auf www.hauptwerk.com.

¹⁷ Bei Hauptwerk ist dies der Ton A.

¹⁸ Universität für Musik und darstellende Kunst Graz.

ter der GPLv2¹⁹-Lizenz erhältlich ist. Grand Orgue kann ebenfalls von einem beliebigen MIDI-Interface aus gesteuert werden, unterstützt aber nicht das MIDI-Tuning-System. Über ein Menü lassen sich verschiedene Temperierungen laden. Das Umschalten zwischen den einzelnen Temperierungen ist jedoch nicht unterbrechungsfrei möglich.

Da es sich um freie Software handelt, wäre es denkbar eine Erweiterung oder Patch zu realisieren, die es erlaubt zu einem beliebigen Zeitpunkt eine Temperierung zu injizieren. Da man aber auch das Problem des natlosen Umschaltens lösen müsste, wäre dies ein großer Eingriff in die Software. Eine Anfrage an die Entwickler von Grand Orgue ob sie bereit wären die Weichen für dynamisches Tuning zu stellen, blieb unbeantwortet. Grand Orgue, ist somit also ebenfalls keine Option zur Realisierung der Arbeit.

Expander

Eine weitere Möglichkeit Orgelklang mithilfe von existierender Hardware wie MIDI-Interfaces oder Orgeln mit MIDI-Ausgang wiederzugeben ist die Verwendung von s. g. Expandern. Expander sind kleine Hardwaresampler, die als Erweiterungsmodule für bestehende Instrumente gedacht sind.

Verschiedene Firmen wie Viscount, Hoffrichter, Rodgers bieten Expander an, welche es ebenfalls erlauben aus einer Liste an Temperierungen zu wählen. Aus den Dokumentationen der einzelnen Geräte ist aber nicht ersichtlich, wie zwischen den einzelnen Stimmungen gewechselt werden kann bzw. ob es eine Möglichkeit gibt diese selbst zu definieren. Auch in einschlägigen Orgelforen (sakral-orgel.de u. ä.) kann man dazu keine genauen Auskünfte finden. Da die Investition in einen Expander auch mit genügend Informationen nicht in Frage kommt, da kein Bedarf besteht, kommt diese Variante für die Arbeit ebenfalls nicht in Betracht.

4.1.3 Beschreibung des Laborsystems

Von den im vorangegangenen Abschnitt beschriebenen Orgeln und Orgel-Systemen eignet sich besonders die Hauptwerk-Orgel für einen Einsatz im Zuge eines Laboraufbaus, da sie eine Schnittstelle zur dynamischen Intonation bietet. Auch die musikalische Praxis am Institut für Orgel und Kirchenmusik (die Hauptwerk-Orgel wird dort aktiv im Konzertbetrieb verwendet) spricht für die Wahl des Instrumentes.

Zusammengefasst sind die Vorteile:

- Bestehende Hardware;
- Sampler läuft auf Rechner mit Maus, Tastatur auf einer Windows-Plattform.

Gegen die Hauptwerk-Orgel sprechen:

- Das Datei-Interface;
- Hauptwerk ist Closed-Source und proprietär.

Das Arbeiten der Hauptwerk-Software auf einem Windows-Rechner bietet den Vorteil, dass für eine relativ weit verbreitete Plattform eine Software entwickelt werden kann, die sich sowohl über Maus und Tastatur als auch über angeschlossene Interfaces bedienen lässt.

¹⁹ GNU General Public License, „freie Software“ Lizenz die Ausführen, Ändern und Verbreiten erlaubt. www.gnu.org/licenses/gpl-2.0.html

Funktionsweise des Temperierungswechsels in Hauptwerk

Die Hauptwerk-Software stellt ein separates Fenster bereit, welches alle Informationen zur derzeitigen Stimmung des Instrumentes auflistet. Abbildung 4.3 zeigt dieses Fenster, welches über das Menü (*View* → *Large floating control panels (for this organ)* → *Pitch*) aufgerufen werden kann.



Abbildung 4.3: Hauptwerk Pitch Window.
Bildschirmfoto der Hauptwerk-Software.

Durch Betätigen der Schaltfläche *Load* kann eine existierende Stimmung in Form einer Datei aufgerufen, geladen und aktiviert werden. Anschließend kann diese existierende Stimmung einer Position im Setzer durch Drücken der Schaltfläche *Assign* zugewiesen werden. Bei Betätigung der Schaltfläche *Load* wird die Datei erneut ausgelesen und die darin enthaltene Stimmung aktiviert. Für die Beschreibung der Temperierung kommt dabei das XML-Format zum Einsatz, es verknüpft 133 Töne mit ihren Frequenzen. Eine genaue Beschreibung des XML-Formats der Temperierungen befindet sich im Anhang B.

Allen beschriebenen Schaltflächen lässt sich eine MIDI-Nachricht zuweisen, wodurch die Hauptwerk-Software über MIDI steuerbar wird. Wurde eine Temperierung in den Setzer geladen, kann die ihr zugrunde liegende Datei verändert werden. Durch Senden einer wählbaren MIDI-Nachricht kann ein *Load* ausgelöst, die Temperierung also neu ausgelesen und aktiviert werden.

Die Temperierungen, die über dieses Menü geladen werden können, legen lediglich die Größen der Intervalle, genauer gesagt den Abstand der Töne zueinander fest. Die tatsächlichen Tonhöhen ergeben sich durch den *Base pitch*, den Stimmton (Normalton), der ebenfalls im *Pitch*-Fenster festgelegt werden kann und dem mittleren A eine fixe Frequenz zuweist.

Systementwurf

In der Hauptwerk-Software ist es möglich selbst gestaltete Temperierungen abzurufen und während des Spielens per MIDI zu aktivieren. Dieses Eigenschaften schaffen die Schnittstelle zu einer dynamischen Stimmungsänderung.

Die Bedienung der Hauptwerk-Software selbst erfolgt über Maus und Tastatur, Ein- und Ausgaben des Programms an einem Monitor. Es bietet sich an ein System zur dynamischen Stimmungsänderung als eigenständiges Programm zu entwickeln und auf die gleiche Art zu bedienen. Zusätzlich soll es die Möglichkeit einer Bedienung auch während des Spielens, also vom Instrument aus, geben. Dies wird über MIDI als Schnittstelle möglich, auf diese Weise kann ein beliebiges Gerät bzw. Interface an das System angeschlossen werden.

Abbildung 4.4 zeigt einen Entwurf des Labor-Systems. In der Mitte befindet sich eine Software, die die dynamische Stimmungsänderung ermöglicht und mit dem Benutzer in Kontakt tritt. Diese Software wird im nachfolgenden als *Dynamic Tuning System* oder kurz *DynTune* bezeichnet. Sie erstellt die Temperierung für das Hauptwerk-System und benachrichtigt es über eine MIDI Schnittstelle. Die Steuerung kann wahlweise per Hand über Maus, Tastatur oder über ein MIDI-Interface erfolgen.

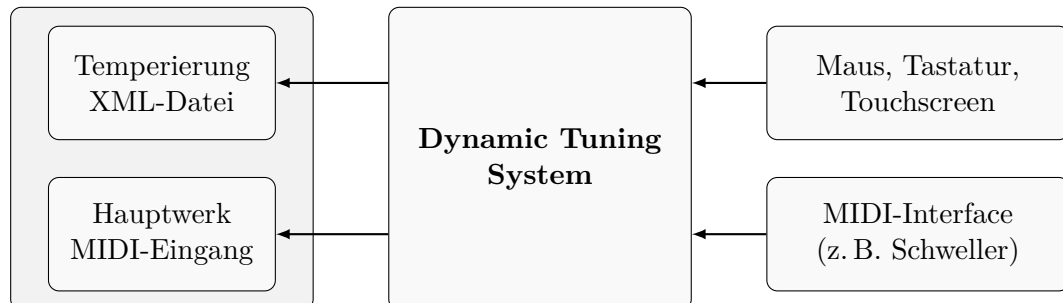


Abbildung 4.4: Schematische Darstellung des Laborsystems. Im Zentrum steht das Dynamic Tuning System, welches per Hand oder über Interfaces bedient wird. Die Steuerung von Hauptwerk erfolgt über eine Datei- und eine MIDI-Schnittstelle.

Um diesen Entwurf umzusetzen, wurde ein System mit dem Namen *DynTune* (Dynamische Stimmung) programmiert. Das Programm *DynTuneGUI* bietet eine grafische Benutzeroberfläche zur Verwendung der Software.

4.2 DynTune-Software

DynTune ist eine Software zur dynamischen Stimmungsänderung. Ihr Ziel ist die dynamische Manipulation der von der Hauptwerk-Software verwendeten Temperierung. Dafür können zwei bestehende Temperierungen geladen und *gemischt* werden. Diese werden als *Basisstimmung* (engl. „Base Tuning“) und *Zielstimmung* (engl. „Target Tuning“) bezeichnet. Die Software spannt einen Stimmvektor zwischen den beiden Stimmungen auf, mit einem Regler kann bestimmt werden, wie groß der Anteil welcher der beiden geladenen Stimmungen sein soll. Außerdem wird die Möglichkeit zur Verfügung gestellt die Basis der Temperierung zu verändern, also den Ton, auf dem die Stimmung aufgebaut ist. Daraus resultierend entsteht die *aktuelle Stimmung* (engl. „Current Tuning“), die an die Hauptwerk-Software gesendet wird.

4.2.1 Benutzeroberfläche

Die Benutzeroberfläche *DynTuneGUI* stellt wichtige Informationen der momentanen Temperierung grafisch dar und bietet Möglichkeiten diese zu manipulieren. Abbildung 4.6 zeigt die Benutzeroberfläche nach dem Programmstart.

Im rechten Teil des Fenster werden zwei Grafiken erzeugt, die sich beide auf die aktuelle Stimmung beziehen. Die obere Grafik zeigt die Verstimmung der großen Terzen, die untere die Verstimmung der Quinten, beide in Cent. Angezeigt wird jeweils der erste (untere) Ton des Intervalls. Als Referenz dienen die in Abschnitt 2.1.2 beschriebenen Frequenzverhältnisse, daraus ergibt sich für die Quinten eine Intervallgröße von 701,96 Cent und für die großen Terzen eine Intervallgröße von 386,31 Cent. D. h. in der Grafik, die die Verstimmung der Terzen darstellt, bedeutet ein Balken in der Höhe von 20 Cent bei C, dass die Terz C/E um 20 Cent größer ist als eine reine große Terz, also eine Intervallgröße von 406,31 Cent aufweist.

Im linken Teil der Oberfläche werden in der obersten Zeile der Name der Basisstimmung, der aktuellen Stimmung und der Zielstimmung dargestellt, wie sie auch in der Hauptwerk-Software angezeigt werden würden²⁰.

Die zweite Zeile (als engl. „Panning (Base/Target Tuning)“ gekennzeichnet) bestimmt die Position am Stimmvektor. Der dort gezeigte *Vektor-Slider* besitzt 100 Stellungen, diese 100 Stellungen bestimmen, wie sich die aktuelle Stimmung aus Basis- und Zielstimmung zusammensetzt.

Die dritte Zeile (als engl. „Current Tuning (Cent)“ gekennzeichnet) ist Ein- und Ausgabeoberfläche in einem. Zwölf Tuning-Slider zeigen die Verstimmung der einzelnen Oktaventöne in Bezug auf die gleichstufige Stimmung (vgl. Abschnitt 3.2.7). Die gleichstufige Stimmung als Bezug wurde gewählt, um eine verständliche und leicht vergleichbare Oberfläche zu schaffen. Alle Slider können in Schritten von -25 bis 25 Cent bewegt werden. Steht z. B. der zweite Slider, der Slider über C# auf einem Wert von 5 Cent bedeutet das, dass das Intervall C/Cis nicht die Größe von 100 Cent hat, wie es in gleichstufiger Stimmung der Fall wäre, sondern um 5 Cent, größer ist, also eine Größe von 105 Cent aufweist. Die Slider werden dynamisch aktualisiert und beziehen sich immer auf die aktuelle Stimmung. Ist der Vektor-Slider auf 0 %, entspricht die aktuelle Stimmung genau der Basisstimmung, ist der Vektor-Slider auf 100 % entspricht die aktuelle Stimmung genau der Zielstimmung. In beiden Fällen werden die zwölf Tuning-Slider für Eingaben aktiviert und ermöglichen ein Anpassen der Basis- bzw. Zielstimmung. Für alle anderen Werte des Vektor-Sliders können die zwölf Tuning-Slider nicht manuell bewegt werden. Die aktuelle Stimmung setzt sich immer aus Basis- und Zielstimmung zusammen. Eine direkte Änderung der einzelnen Intervallgrößen zwischen diesen Stimmungen ist nicht vorgesehen.

Die untersten zwei Zeilen (als engl. mit „Tuning Base [Base Tuning]“ bzw. „Tuning Base [Target Tuning]“ beschriftet) ermöglichen es die Note zu setzen, auf der Basis- bzw. Zielstimmung aufgebaut sind. In den meisten Fällen wird die Stimmung auf C aufgebaut. Betrachtet man z. B. die pythagoreische Stimmung auf Seite 31, besitzt diese eine stark (um das pythagoreische

²⁰ Dieser Name ist in der Temperierungsdatei als *ShortName* gekennzeichnet (vgl. Anhang B).

Komma) verstimmte Quinte. Ist die Basis der Stimmung C, handelt es sich dabei um die Quinte Cis/Gis. Ändert man die Basis von C auf Cis, verschiebt sich diese Quinte von Cis/Gis auf D/A. Abbildung 4.5 stellt dieses Verhalten in gleicher Form grafisch dar, wie es auch die Software tun würde.

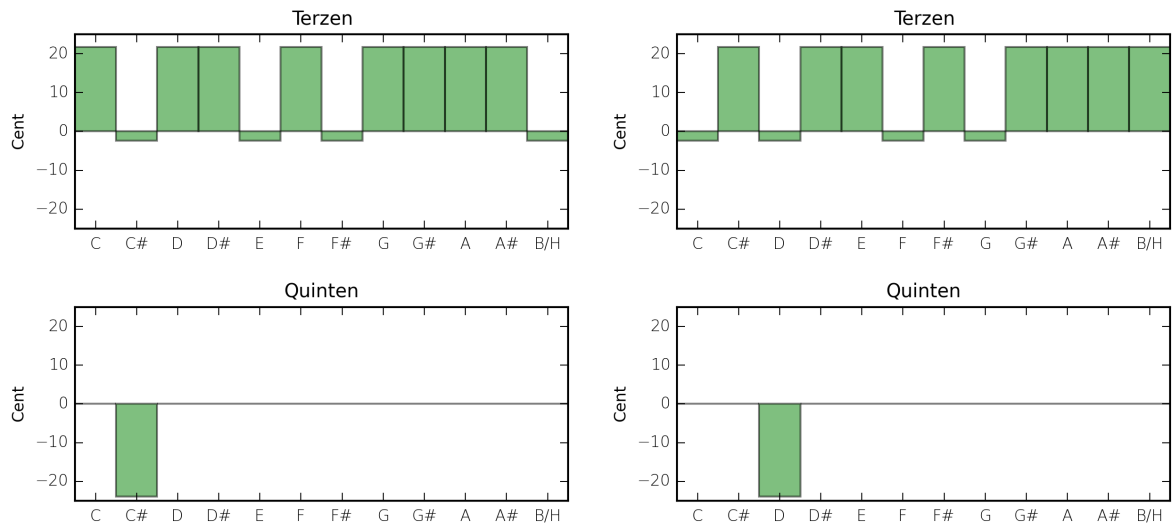


Abbildung 4.5: Darstellung der Verstimmung von Terzen und Quinten wie in der DynTune Software.
Linke Spalte: Stimmung auf Basis C, rechte Spalte: Stimmung auf Basis Cis.

Die Änderung der Basis der Stimmung ermöglicht es, nicht nur in C-Dur, sondern wahlweise auch in anderen Tonarten, z. B. *pythagoreisch rein* (vgl. Abschnitt 3.2.3) zu spielen oder die Basis der Stimmung kontinuierlich zu verändern. Diese Funktion steht in keinem der eingangs beschriebenen Orgelsystemen (Hauptwerk-Software, Grand-Orgue, Rodgers Instruments, ...) zur Verfügung.

Das *Datei*-Menü bietet weiters folgende Möglichkeiten:

Open Base Tuning zeigt einen Dateidialog an, um eine neue Temperierung als Basisstimmung festzulegen. Nachdem diese ausgewählt wurde, wird sie sofort in das Programm geladen.

Open Target Tuning zeigt einen Dateidialog an, um eine neue Temperierung als Zielstimmung festzulegen. Nachdem diese ausgewählt wurde, wird sie sofort in das Programm geladen.

Save current tuning speichert die aktuelle Stimmung als Datei mit dem zuletzt verwendeten Dateinamen. Wurde zuvor noch keine Stimmung gespeichert, wird ein Dateidialog zur Auswahl des Dateinamens und Speicherorts angezeigt.

Save current tuning as zeigt einen Dateidialog an, um die aktuelle Stimmung als Datei zu speichern.

Reset setzt alle Änderungen, die vom Benutzer durchgeführt wurden, zurück. Basis- und Zielstimmung entsprechen wieder dem Ausgangszustand.

Exit beendet das Programm.

Das Menü *Engine* bietet die Möglichkeit die Kommunikation zwischen dem Programm und der Hauptwerk-Software zu starten oder zu stoppen. Über das Menü *MIDI* kann die gesamte MIDI-Kommunikation gestartet oder gestoppt werden, dies betrifft sowohl das Auslesen ankommender Steuer-Nachrichten als auch das Senden von Signalen an die Hauptwerk-Software. Außerdem kann über das Menü eine Test-Nachricht an die Hauptwerk-Software gesendet werden, um z. B. dort die Konfiguration der Schaltfläche *Load* zu vereinfachen.

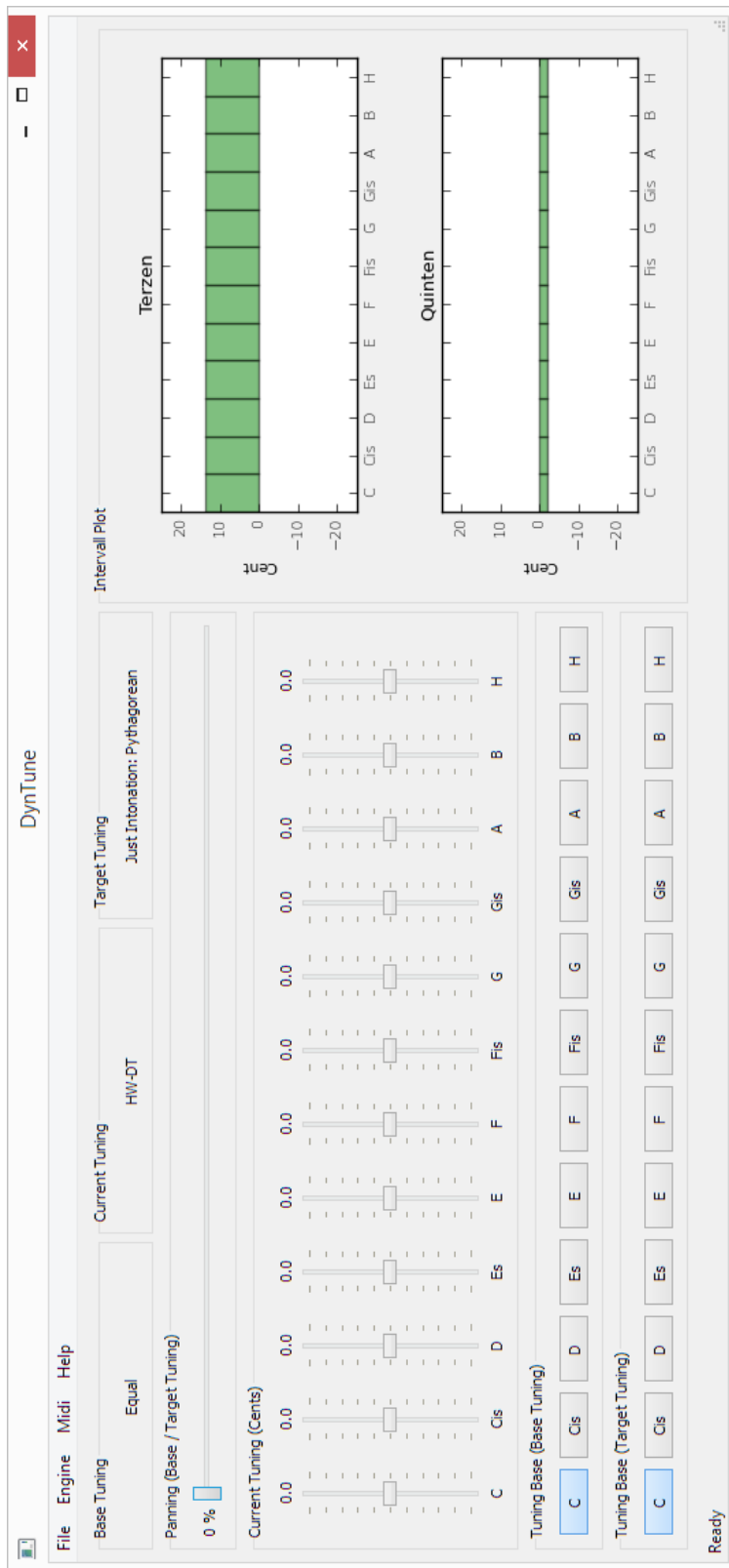


Abbildung 4.6: Grafische Benutzeroberfläche der DynTune Software.

4.2.2 Konfigurationsdatei

Die grafische Benutzeroberfläche DynTuneGUI kann eine Konfigurationsdatei beim Programmstart laden. In ihr können verschiedene Parameter des Programms festgelegt werden. Für die Verwendung von MIDI, also der Kommunikation mit der Hauptwerk-Software oder einem Interface zur Steuerung, ist die Konfigurationsdatei unbedingt notwendig, denn sind keine MIDI-Parameter definiert, wird der dafür notwendige Teil in der Software deaktiviert.

Beim Start von DynTune kann eine Konfigurationsdatei explizit angegeben werden. Dazu muss das Programm mit dem Parameter `-c` aufgerufen werden.

```
DynTuneGUI.exe -c <config.ini>
```

Wird keine Datei angegeben, sucht die Software im aktuellen Verzeichnis nach dieser Datei `hwdts.ini`. Wird keine existierende Konfigurationsdatei gefunden, erzeugt das Programm nach dem Start eine Standardkonfiguration. Nachfolgend werden die einzelnen Teile der Konfigurationsdatei näher beschrieben. Für diese einzelnen Teile gilt das gleiche wie für die Datei an sich: Fehlt einer der Teile, wird auf eine Standardkonfiguration zurückgegriffen.

Startparameter

In der Sketion `hwdts` kann festgelegt werden, ob nach dem Programmstart die MIDI-Kommunikation mit der Hauptwerk-Software und Interfaces zur Steuerung automatisch gestartet wird.

```
1 [hwdts]
2 autostart = on
```

autostart Mögliche Werte sind: on/off, True/False und yes/no. Standardwert ist off.

Hauptwerk-Temperierungen

In der Sektion `hw` wird das Programm über den Speicherort der Temperierungen informiert.

```
3 [hw]
4 dir = C:\Hauptwerk
5 base = Equal
6 target = JustIntonation-Pythagorean
```

dir gibt den absoluten Pfad zum Hauptverzeichnis der Hauptwerk-Software an. Standardwert ist `C:\Hauptwerk`. Falls in der Hauptwerkinstallation Programm und Datenverzeichnis getrennt wurden, so ist das Verzeichnis anzugeben, in dessen Unterverzeichnis sich die Temperierungen (`HauptwerkSampleSetsAndComponents/Temperaments`) befinden.

base Der Parameter gibt an, welche Temperierung bei Programmstart als *Basis* geladen wird. Der Name entspricht dem Dateinamen der Stimmung, die sich im Temperierungs-Verzeichnis der Hauptwerk-Software befindet. Als Standardwert wird die gleichstufige Stimmung verwendet. Diese ist im Programm selbst hinterlegt.

target Der Parameter gibt an, welche Temperierung bei Programmstart als *Target* geladen wird. Der Name entspricht dem Dateinamen der Stimmung, die sich im Temperierungs-Verzeichnis der Hauptwerk-Software befindet. Als Standardwert wird die gleichstufige Stimmung verwendet. Diese ist im Programm selbst hinterlegt.

Midi-Kommunikation

In der Sektion `midi` werden die Parameter für die Midi-Kommunikation festgelegt. Das betrifft Ein- und Ausgangsport sowie die gesendeten und empfangenen Nachrichten. Bei den gesendeten und empfangenen Nachrichten handelt es sich immer um den Typ *ControlChange*. Ist die Sektion nicht vorhanden, wird die Midi-Kommunikation im Programm deaktiviert.

```

7 [midi]
8 inport = loopport-pd
9 outport = loopport-hw
10 out-channel = 0
11 out-control = 1
12 out-value = 100
13 in-channel = 0
14 in-control = 0
15 in-value-min = 0
16 in-value-max = 100

```

inport legt den Eingangsport fest, an dem das Programm auf Steuernachrichten wartet. Diese Nachrichten legen die Position am Stimmvektor fest.

outport legt den Ausgangsport fest, auf welchem das Programm Signale an die Hauptwerk-Software sendet.

out-channel gibt den MIDI-Kanal an, über den das Programm Signale an die Hauptwerk-Software sendet. Mögliche Werte sind 0-15. Standardwert ist 0.

out-control gibt den Controller an von welchem die MIDI-Nachricht gesendet wird. Mögliche Werte sind 0-127. Standardwert ist 1.

out-value gibt den Wert an, der in der MIDI-Nachricht gesendet und von der Hauptwerk-Software als Trigger benutzt wird. Mögliche Werte sind 0-127. Standardwert ist 100.

in-channel gibt den MIDI-Kanal an, auf welchem das Programm auf Steuernachrichten wartet. Mögliche Werte sind 0-15. Standardwert ist 0.

in-control gibt den Controller an, auf dessen Steuernachrichten das Programm wartet. Mögliche Werte sind 0-15. Standardwert ist 0.

in-value-min gibt den unteren Grenzwert an, der an das Programm gesendet wird. Dieser Wert wird vom Programm als 0 % interpretiert. Die Position am Stimmvektor entspricht der Basisstimmung. Mögliche Werte sind 0-127. Standardwert ist 0.

in-value-max gibt den oberen Grenzwert an, der an das Programm gesendet wird. Dieser Wert wird vom Programm als 100 % interpretiert. Die Position am Stimmvektor entspricht der Zielstimmung. Mögliche Werte sind 0-127. Standardwert ist 127.

Normalton der Stimmung

Die Sektion legt den Stimmtton (Normalton) der Stimmung fest, also den fixen Ton, um welchen herum die dynamische Stimmung aufgebaut wird. Ist die Sektion nicht vorhanden, wird $A = 440$ Hz als Stimmtton (Normalton) verwendet.

```

17 [base-pitch]
18 frequency = 440
19 note-name = A

```

frequency legt die Frequenz des Stimmtons (Normaltons) fest. Der Wertebereich sind Gleitkommazahlen größer 0. Der Standardwert ist 440.

note-name gibt die Note an, die dem Normalton entspricht. Mögliche Werte sind C, C#, D, D#, E, F, F#, G, G#, A, A#, B und H. Standardwert ist A.

4.3 Inbetriebnahme der Software

DynTune muss über MIDI mit der Hauptwerk-Software verbunden werden. Dazu gibt es verschiedene Möglichkeiten, z B. über einen physikalisch vorhandenen MIDI-Port oder einen virtuellen Loop-Interface wie loopMIDI²¹. Eine verfügbare MIDI-Verbindung sei für die weiteren Schritte vorausgesetzt.

Nachdem eine Konfigurationsdatei erstellt wurde (vgl. dazu Abschnitt 4.2.2), sollte diese als

```
hwtdts.ini
```

im gleichen Verzeichnis abgelegt werden, in dem sich auch DynTune befindet. Anschließend kann DynTune per Doppelklick auf `DynTuneGUI.exe` gestartet werden.

In der Hauptwerk-Software ist über das Menü

View → *Large floating control panels (for this organ)* → *Pitch*

das Stimmungs-Menü zu öffnen. Über die Schaltfläche *Load...* muss nun im Temperierungs-Verzeichnis der Hauptwerk-Software die Datei

```
hwtdts.Temperament_Hauptwerk.xml
```

ausgewählt und durch Betätigung der Schaltfläche *Assign* in den Setzer geladen werden. Per Rechtsklick auf die Schaltfläche *Load* wird nun ein MIDI-Befehl hinterlegt. Es genügt dafür *Auto-detect* auszuwählen und in der DynTune Software eine Testnachricht zu senden (dies erfolgt über das MIDI-Menü. vgl. Abschnitt 4.2.1.). In einem letzten Schritt werden die Einstellungen der MIDI-Kommunikation für die Schaltfläche *Load* der Hauptwerk-Software bearbeitet. Per Rechtsklick auf diese wird nun *Adjust-MIDI* ausgewählt. Ein neues Fenster wird geöffnet, dieses Fenster ist in Abbildung 4.7 dargestellt. Der in der Abbildung rot markierte Parameter

Prevent rapid piston re-triggering

muss deaktiviert werden, um eine Aktualisierung der Stimmung mit hoher Rate zu ermöglichen²².

Falls DynTune nicht über den Autostart-Parameter in der Konfiguration gestartet wurde, kann nun über das Menü *Engine* die dynamische Aktualisierung von Hauptwerk gestartet werden. DynTune ist nun einsatzbereit.

²¹ www.tobias-erichsen.de/software/loopmidi.html.

²² Es ist nicht genauer bekannt, ab welcher Betätigungsrate der Schaltfläche über MIDI die Hauptwerk-Software keine Kommandos mehr befolgt.

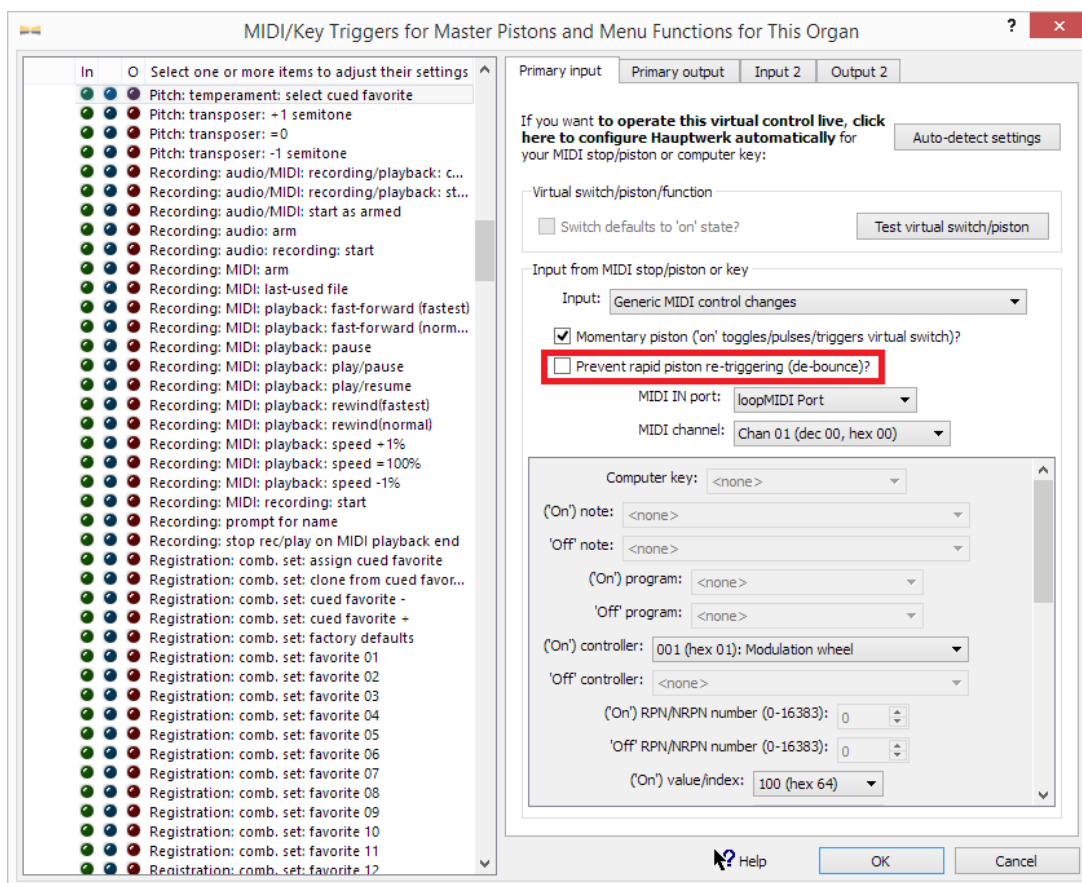


Abbildung 4.7: Midi/Key-Einstellungen in der Hauptwerk-Software. Der markierte Parameter („Prevent rapid piston re-triggering“) ist zu deaktivieren. Bildschirmfoto der Hauptwerk-Software.

4.3.1 Steuerung über externe Interfaces

Falls in der Konfigurationsdatei (Abschnitt 4.2.2) MIDI-Eingangs-Parameter festgelegt wurden, wartet DynTun nach dem Start der MIDI-Kommunikation automatisch auf Steuer-Nachrichten. Momentan wird bei allen Steuer-Nachrichten davon ausgegangen, dass es sich um den Typ „Control Change“ handelt, deshalb ist die Angabe von Kanal und Controller erforderlich. Werte von 0-127 werden automatisch auf einen Bereich von 0-100 % am Stimmungs-Vektor umgelegt. Ist das Interface in seinem Bereich eingeschränkt können über die Konfigurationsdatei Minimal- und Maximal-Wert gestgesetzt werden. DynTune, wird die Skalierung anpassen.

4.4 Technische Implementierung

DynTune wurde vollständig in Python²³ (Version 3.4) erstellt. Für die grafische Benutzeroberfläche wurde die QT²⁴-Bibliothek gewählt. Durch diese Kombination ist die Software Plattform unabhängig und unterstützt mit Windows sowie OSX die beiden Plattformen, auf denen auch die Hauptwerk-Software derzeit verfügbar ist. (Eine Linux Version ist ebenso möglich, wurde aber noch nicht getestet.) Außerdem kommen folgende weitere Bibliotheken (Python Module) zum Einsatz:

²³ www.python.org.

²⁴ www.qt.io.

matplotlib²⁵ ist eine Python Bibliothek zum zweidimensionalen Plotten verschiedenster Daten. Die DynTune Software verwendet sie zur Darstellung der Terzen- und Quintenverstimmung.

mido²⁶ ist eine Bibliothek, die das Arbeiten mit MIDI-Objekten in Python ermöglicht. Sie stellt Funktionen bereit, um MIDI-Nachrichten zu erstellen und zu manipulieren. Für das Senden und Empfangen von Nachrichten wird ein Backend benötigt.

pygame²⁷ wird von mido als Backend zum Senden und Empfangen von MIDI-Nachrichten verwendet. Derzeit hat es die Einschränkung, dass keine Nachrichten vom Typ Sysex unterstützt werden.

numpy²⁸ ist nach eigenen Angaben das fundamentale Paket für wissenschaftliches Rechnen mit Python. Die Bibliothek unterstützt das Rechnen mit N-dimensionalen Arrays (Matritzen) und wird in DynTune u.a. zum Berechnen und Manipulieren des Stimmungsvektors eingesetzt.

DynTune realisiert verschiedene Entwurfsmuster, darunter das Model-View-Presenter-Muster. Das bedeutet, dass Daten, Verarbeitungslogik und Präsentationsoberfläche voneinander getrennt sind. Die Stimmungsdaten, das Model, sind im Hauptwerk XML-Format gespeichert (siehe Anhang B) und werden mit einem XML-Parser (HWXml) ausgelesen sowie geschrieben.

Die Anzeige dieser Daten sowie die Interaktion mit dem Benutzer geschieht in der DynTuneGUI Anwendung, dem View. Die Brücke zwischen diesen beiden bildet DynTune, das für die Verarbeitung und Manipulation der Daten zuständig ist, es nimmt die Rolle des Presenters ein.

Der Vorteil dieses Entwurfsmusters ist, dass sowohl verschiedene Präsentationsoberflächen gewählt werden können — denkbar ist z. B. auch die Steuerung über eine Weboberfläche oder eine reine Kommandozeilen-Anwendung —, aber auch die Art der Datenspeicherung variabel ist. Sollte eine Möglichkeit gefunden werden, andere Orgeln für die Verwendung von dynamischen Tuning zu erschließen, muss nur dieser Teil ausgetauscht werden, nicht aber DynTune oder DynTuneGUI. Abbildung 4.8 verdeutlicht die Implementation des Entwurfsmusters.



Abbildung 4.8: Implementation des Model-View-Presenter-Musters in der DynTune-Software.

DynTune, DynTuneGUI und HWXml bilden die elementaren Bestandteile der Software. Die Darstellung der Verstimmung von Terzen und Quinten übernimmt die beschriebene matplotlib. Diese ist in einen separaten Programmteil, dem **IntervallPlotter**, ausgelagert und könnte auch zum Abspeichern von Grafiken genutzt werden.

Mido mit dem Pygame-Backend bietet keine Möglichkeit bei Empfang von Nachrichten ein Event auszulösen. Das Programm selbst muss überprüfen, ob neue Nachrichten vorhanden sind und diese abholen (*Poll*). Dieser Vorgang wurde in einen parallel laufenden Teil des Programms (*Thread*), den **DynTuneMidiReceiver**, ausgelagert, um den restlichen Programmablauf nicht zu stören.

Ebenfalls in einen separaten und parallel laufenden Programmteil, dem **DynTuneHWUpdate**, ist die Kommunikation mit der Hauptwerk-Software ausgelagert, dies aus zwei Gründen.

²⁵ www.matplotlib.org.

²⁶ www.github.com/olemb/mido.

²⁷ www.pygame.org.

²⁸ www.numpy.org.

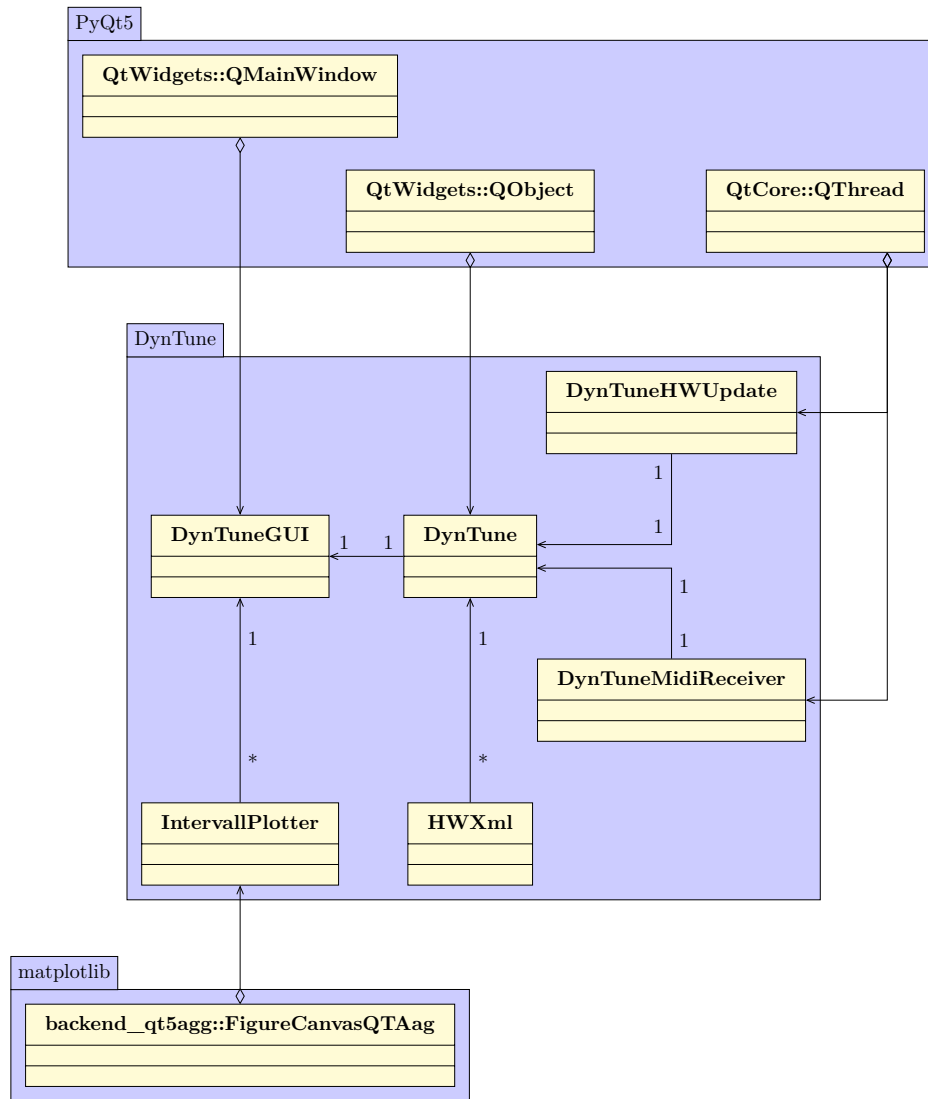


Abbildung 4.9: Klassendiagramm der DynTune-Software.

Erstens: Da nicht vorausgesagt werden kann wie lange das Schreiben der Temperierung dauert, soll der restliche Programmablauf nicht gestört werden. Zweitens: Da der Vorgang des Dateischreibens sowie der Benachrichtigung der Hauptwerk-Software und das Auslesen der Datei seitens der Hauptwerk-Software nicht in beliebig kurzer Zeit erfolgen kann und es auch keine Rückmeldung gibt, ob der Vorgang erfolgreich war, muss ihm genügend Zeit eingeräumt werden. Die Aktualisierungsrate der Stimmung in der Hauptwerk-Software wird beschränkt.

Die Verwendung von QT hat den Nachteil, dass eine sehr große Bibliothek, die ursprünglich rein auf die Erstellung von grafischen Benutzeroberflächen abzielte, auf dem Endsystem vorhanden sein muss. Die Stärke von QT eine umfangreiche und einfache Lösung zur Gestaltung der Oberfläche zu bieten sowie der Vorteil der Plattformunabhängigkeit überwiegen jedoch. Zusätzlich bietet QT darauf aufbauenden Anwendungen eine gemeinsame Verwaltung von Threads (QThreads) und Locks (z. B. QMutex) und sorgt so für eine störungsfreie Kommunikation zwischen den einzelnen Teilen einer Anwendung. Um dieses Zusammenspiel zu nutzen, baut DynTune auf QT auf. Der Presenter `DynTune` wird von `QObject` abgeleitet und kann so den QThreads `DynTuneHWUpdate` und `DynTuneMidiReceiver` als Parent (Verwalter – wörtlich eigentlich „Elternteil“) dienen. Abbildung 4.9 zeigt das Klassendiagramm von DynTune und verdeutlicht den Zusammenhang der einzelnen Komponenten.

4.4.1 Programmablauf

DynTune wird über `DynTuneGUI` gestartet. `DynTuneGUI` liest die Konfigurationsdatei und instanziiert mit den darin enthaltenen Parametern `DynTune`. `DynTune` liest sofort die gewünschten Temperierungen aus den angegebenen Dateien und initialisiert Basis-, Ziel- und aktuelle Stimmung. Anschließend baut `DynTuneGUI` die grafische Benutzeroberfläche auf und beendet den Startprozess um das Hauptfenster der Anwendung zu zeigen.

Wurde der Parameter *Autostart* in der Konfiguration gesetzt (siehe Abschnitt 4.2.2) teilt `DynTuneGUI` `DynTune` mit die beiden Threads (`DynTuneHWUpdate` und `DynTuneMidiReceiver`) zu starten. Ist dies nicht der Fall können diese nun manuell über das Menü *Engine* bzw. *Midi* gestartet werden.

Sobald Teile der Stimmung über die Benutzeroberfläche verändert werden, werden diese Änderungen `DynTune` mitgeteilt und die aktuelle Stimmung wird unmittelbar neu berechnet. Das gleiche gilt für den Fall, dass von der Komponente `DynTuneMidiReceiver` eine Steuer-Nachricht empfangen wurde. Nachdem eine Neuberechnung der aktuellen Stimmung abgeschlossen wurde, löst `DynTune` ein Signal aus. Sowohl `DynTuneGUI` als auch `DynTuneHWUpdate` empfangen dieses Signal. `DynTuneGUI` wird daraufhin die Benutzeroberfläche mit den Informationen der neuberechneten aktuellen Stimmung aktualisieren. `DynTuneHWUpdate` wird die Temperierungsdatei der Hauptwerk-Software aktualisieren und dieser über eine MIDI-Nachricht mitteilen dass diese die Temperierungs-Datei neu laden kann.

Wie bereits beschrieben wird `DynTuneHWUpdate` den Prozess des Schreibens der Temperierungsdatei und die Verständigung der Hauptwerk-Software so kontrollieren, dass dieser nicht mehrmals unmittelbar hintereinander ausgelöst werden kann, um der Hauptwerk-Software immer genügend Zeit zur Verarbeitung einzuräumen. Die Mindestpause, die dabei eingehalten wird, ist in `DynTuneHWUpdate` festgelegt und beträgt in der derzeitigen Implementierung 100 ms, daraus ergibt sich eine Aktualisierungsrate der Stimmung von circa 10 Hz. (Zu den 100 ms kommt noch die Zeit hinzu, die die Software für den Rechen- und Schreibvorgang benötigt.)

Eine genaue Dokumentation der Software findet sich in Anhang C.

5

Epilog

Das dynamische Tuning, wie es von DynTune realisiert wird, ist in dieser Form erstmals für Orgel verfügbar. Um an eine systematische Beurteilung des Systems heranzugehen, müssen geeignete Fragestellungen und Experimente erst entwickelt und die zu erwartenden Einsatzmöglichkeiten in der Literatur gefunden werden. Diese zum Teil musikalisch-analytische Aufgabe ist nicht mehr Teil der Arbeit.

Nachfolgende Abschnitte bilden die Ergebnisse erster explorativer Anwendungen ab. Diese beinhalten Anwendungsbeschreibung, ein exemplarisches Interview, theoretische Reflexionen und Untersuchungen. Zur Evaluation stehen Hörbeispiele im Anhang A zur Verfügung.

5.1 Ergebnisse erster Anwendung

Das DynTune-System wurden im November 2015, dem Institutsvorstand am Institut für Orgel und Kirchenmusik, Gunther Rost, vorgestellt. Am 19. November 2015 fand es bei einem Konzert im Sternenturm der Stadtgemeinde Judenburg ersten Einsatz. Im Rahmen dieses Konzerts wurden die Goldberg-Variationen (BWV 988) von Johann Sebastian Bach aufgeführt. Im Vorfeld wurde Herr Rost befragt:

Warum haben Sie Interesse an dynamischem Tuning?

„Alle Tasteninstrumente müssen bisher mit einem globalen Stimmungskompromiss leben. Das Wohltemperierte Klavier J. S. Bachs thematisiert Temperaturvarianten, die erstmals ein ausgeglichenes Spiel in allen Tonarten durch die geschickte Verteilung der ‚Stimmungsfehler‘ ermöglichen. Für jeden Moment einer Partitur die Zusammenklänge gezielt ausstimmen zu können, bietet einen ganzen Kosmos neuer Möglichkeiten.“

Wie würden Sie es bei der Darbietung der Goldberg-Variationen einsetzen?

„Zunächst sehr vorsichtig, da das Tool großes musikalisches Gewicht hat – ich muss also erstmal üben und ein Gefühl dafür entwickeln.“

DynTune wurde zuvor an der Hauptwerk-Orgel in Betrieb genommen. Vor und während des Konzerts traten keine Probleme mit der Software auf. Herr Rost wurde anschließend nochmals nach seiner Einschätzung befragt:

Wie haben Sie die Software bei den Goldberg-Variationen eingesetzt?

„Ausschließlich in der großen, langsamen Variation 25 – diese thematisiert extreme harmonische Spannung. Hier habe ich versucht die Kontraste zwischen entspannten Durakkorden und den dazwischen liegenden harmonischen Labyrinthen etwas zu intensivieren.“

Wie gut eignet sich der Schwelltritt als Steuerung?

„Der Schwelltritt eignet sich in manualiter Zusammenhängen sehr gut, wenn das Pedal virtuos eingesetzt wird, entsprechend weniger gut. Man könnte also noch andere Möglichkeiten für die Livesteuerung probieren wie z. B. Aftertouch oder Körpertracking.“

Welche Veränderungen oder Erweiterungen würden Sie sich für spätere Versionen der Software wünschen?

„Sicher wäre hilfreich, wenn sich die jeweilige Temperatur in die gewählte Registrierungsdatei speichern ließe und per Sequenzer abrufbar wäre. Eine Variante wäre auch, die Tonhöhe Stimme für Stimme über die Tastatur steuern zu können.“

Vom Konzert wurden ein Stereo-Master sowie die MIDI-Daten der Hauptwerk-Software aufgezeichnet. Im Stereo-Master ist die angesprochene Variation 25 mit der von DynTune dynamisch generierten Stimmungen enthalten. Diese Variation findet sich als Klangbeispiel 10 im Anhang A. DynTune wurde auf die Verwendung der Silbermann-Stimmung als Basisstimmung und der Kirnberger-Stimmung als Zielstimmung konfiguriert.

Mit den gewonnenen MIDI-Daten konnte die Variation nochmals in der Hauptwerk-Software reproduziert werden, diesmal ohne den Einsatz von DynTune. Klangbeispiel 8 enthält die Variation in Silbermann-Stimmung, Klangbeispiel 9 in Kirnberger-Stimmung und Klangbeispiel 11 in gleichstufiger Stimmung.

Der Anhang enthält außerdem die aus Abschnitt 3.2 bekannte Kadenz, als Klangbeispiel 7, unter Zuhilfenahme einer dynamischen Stimmung. DynTune wurde dazu auf die Verwendung der reinen Stimmung als Basisstimmung und der gleichstufigen Stimmung als Zielstimmung konfiguriert. Zu diesem Beispiel wurde ein Video aufgezeichnet, das die Schwellerbewegung des Organisten, die DynTune-Software sowie die Klangänderung, also den Einsatz der DynTune Software, zeigt: dies findet sich im Anhang A als Beispiel 12.

5.2 Diskussion

5.2.1 Approximationsfunktion

In der derzeitigen Version der Software werden die Stimmungen im Cent-Bereich manipuliert. Basis- und Zielstimmung beinhalten so wie der Stimmungsvektor Intervallgrößen in Cent. Die Bewegung auf dem Stimmungsvektor ist linear, die Veränderung der Frequenz daher logarithmisch (vgl. Abschnitt 2.1.3). Es stellt sich nun die Frage, welchen Einfluss diese logarithmische Bewegung hat. Abbildung 5.1 zeigt den Unterschied zwischen einer linearen Bewegung im Frequenz-Maß und einer linearen Bewegung im Cent-Maß für eine Verstimmung um 25, 50, 100 Cent zwischen einer Basis- und Zielstimmung. Dies würde in gleichstufiger Stimmung einer Verstimmung um einen Achtel-, Viertel- bzw. Halbton entsprechen.

Zu erkennen ist, dass der Unterschied zwischen den beiden Verfahren sogar bei der Verstimmung um einen Halbton lediglich 0,72 Cent beträgt und deshalb vernachlässigbar ist (vgl. Abschnitt 2.2.1).

Die Änderung der Tonhöhe um einen Halbton zwischen zwei Stimmungen ist eher fiktiv und kommt bei den in Abschnitt 3.2 beschriebenen Stimmungen nicht vor. Die in der Regel maximale Änderung liegt im Bereich von ungefähr 15 Cent. Abbildung 5.2 zeigt den Übergang von gleichstufiger Stimmung zu pythagoreischer Stimmung und stellt dabei die Abweichung zwischen linearer und logarithmischer Annäherung dar. In diesem Beispiel werden die Noten **Cis**, **D** und **Gis** betrachtet mit den Frequenzen, wie sie auch in der Temperierungsdatei der Hauptwerk-Software festgelegt sind. Die maximale Abweichung zwischen den beiden Funktionen beträgt in diesem Fall 0,026 Hz.

Diese Untersuchung zeigt, dass es keinen merklichen Unterschied zwischen linearer und logarithmischer Bewegung auf dem Stimmungsvektor gibt. Zudem kann auch keine der beiden Methoden als ideal angesehen werden, ebenso wenig wie eine Aussage darüber getroffen werden kann, ob nicht eine dritte Funktion besser geeignet wäre.

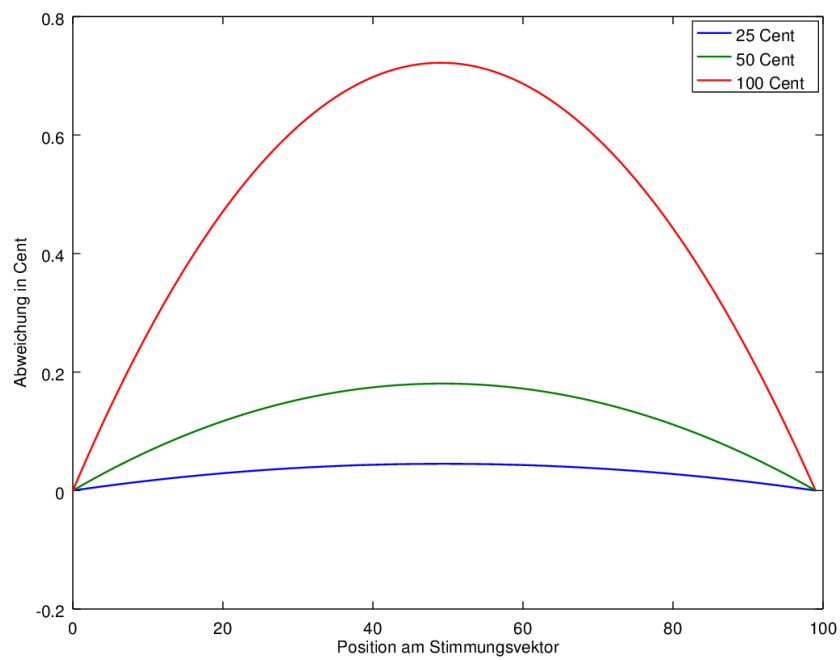


Abbildung 5.1: Abweichung zwischen linearer und logarithmischer Bewegung am Stimmungsvektor.

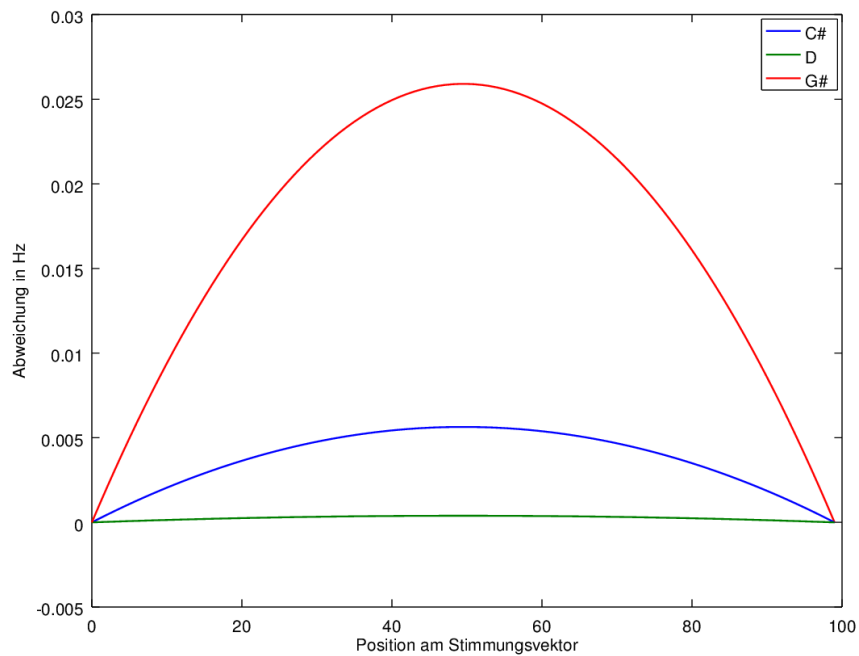


Abbildung 5.2: Abweichung zwischen linearer und logarithmischer Bewegung am Stimmungsvektor. Gewählt wurde die gleichstufige Stimmung als Basisstimmung sowie die pythagoreische Stimmung als Zielstimmung.

Intelligente Approximation

Die Bewegung auf dem Stimmungsvektor verläuft für alle Noten gleich. Das Intervall, um den ein Ton umgestimmt werden muss, wird in Schritte geteilt und durchlaufen. Bei den in Abschnitt 3.2 beschriebenen Stimmungen kommt es dabei zu Änderungen der Intervallgröße von 0 bis maximal ca. 15 Cent. Die Frage, die sich dabei eröffnet, ist, ob für alle Noten die gleiche Annäherung erfolgen soll oder ob z. B. für kleine Änderungen eine lineare, für große Änderungen der Intervallgröße eine logarithmische Annäherung zu erfolgen hat.

Zudem bleibt offen, ob Intervalle besonderer Wichtigkeit existieren und ob diesen Intervallen eine besondere Beachtung geschenkt werden muss, z. B. in Form einer Fixierung, die ihre Größe für alle Stimmvorgaben festlegt.

Die Verantwortung für die Bewegung am Stimmungsvektor selbst liegt momentan beim Organisten. Dieser muss je nach Basis- und Zielstimmung selbst entscheiden, wie schnell Sprünge der aktuellen Stimmung durchgeführt werden.

5.2.2 Aktualisierungsrate

Wie in Abschnitt 4.4.1 beschrieben verlaufen das Berechnen der Stimmung sowie das Schreiben der Temperierungsdatei asynchron. Die derzeit implementierte Aktualisierungsrate beträgt ungefähr 10 Hz.

Betrachtet man eine durchschnittliche Bewegung des Schwellertritts (siehe dazu auch das Video in Anhang A), dauert diese einige Sekunden. Es kommt während des Verlaufs von einer Basis- zu einer Zielstimmung also mindestens zu 15–20 Aktualisierungen der Stimmung. Geht man davon aus, dass Basis- und Zielstimmung keine extremen Unterschiede (z. B. von Halbtonschritten) aufweisen, sind die einzelnen Aktualisierungen nicht wahrnehmbar.

5.2.3 Normalton

Die DynTune-Software erlaubt es in ihrer Konfiguration einen Stimmtton (Normalton) festzulegen. Dieser Stimmtton dient auch als Basis zur Erstellung der Temperierungsdatei für die Hauptwerk-Software (vgl. Abschnitt 4.2.2). In der aktuellen Version der Hauptwerk-Software wird nach der Analyse der Temperierungsdatei diese normalisiert. Dem Ton A (MIDI-Note Nummer 69) wird immer die in der Hauptwerk-Software angegebene Frequenz zugewiesen. Eine Konfiguration in DynTune ist deshalb, zumindest für die Nutzung mit der Hauptwerk-Software, von keiner Relevanz.

Ausgehend von diesem Stimmtton wird DynTune auch die Umrechnung in das Cent-Maß und damit die Bestimmung der Intervallgrößen durchführen. Anschließend kann jede Intervallgröße in der DynTune-Software nachbearbeitet werden. Zu überlegen bleibt, was passieren soll, wenn der Slider über der Note bewegt wird, die als Stimmtton festgelegt ist. Das derzeitige Verhalten der Software führt zu einer Änderung aller Intervalle.

Ein Beispiel: Ist A der Stimmtton und wird der Slider über C um +10 Cent verschoben, wird das Intervall C/A um 10 Cent größer. Die Tonhöhe von C steigt, die von A bleibt unverändert. Wird der Slider über A um +10 Cent verschoben, wird das Intervall C/A um 10 Cent größer. Die Tonhöhe von C ändert sich, die von A bleibt unverändert (bedingt durch die Normalisierung auf den Stimmtton). Zusätzlich ändert sich aber auch die Tonhöhe aller anderen 10 Noten.

5.2.4 Ideale Stimmung

Im Gegensatz zu anderen Systemen, die eine dynamische Stimmung ermöglichen (Hermoder, Spring u.a., vgl. Abschnitt 3.4) versucht DynTune nicht eine ideale Stimmung aufgrund des Stückverlaufs oder anderer Parameter zu postulieren. Aus den Kapiteln 2 und 3 der Arbeit

geht hervor, dass ein Weg zu einer idealen Stimmung noch nicht gefunden ist. Mit seiner Funktion stellt DynTune aber Möglichkeiten bereit einzelne Teile eines Stückes zu unterstreichen oder zu entspannen. Dies geht aus den ersten Anwendungen im Konzertbetrieb hervor (vgl. Abschnitt 5.1).

5.3 Zusammenfassung

Kapitel 3 der Arbeit gibt einen Einblick in die Probleme, die beim Stimmen von Tasteninstrumenten auftreten: Es wird deutlich dass keine Stimmung gefunden werden kann, in der alle Intervalle aller Tonarten rein erklingen. Adaptive Systeme versuchen, durch Analyse der Intervalle bereits vor der Klangerzeugung das Instrument umzustimmen. Diese Systeme haben eine möglichst reine Stimmung zum Ziel. In den in Abschnitt 3.2 beschriebenen Stimmungen wird versucht, möglichst viele Intervalle rein wiederzugeben. Das bedeutet aber nicht, dass z. B. bei der Interpretation von Stücken in reiner Stimmung, das Spiel mit Dissonanzen ausgeschlossen ist. Abschnitt 2.2.5 zeigt, wie sich der Begriff *Dissonanz* im Laufe der Zeit gewandelt hat. Dissonanz wird in den dort aufgelisteten Definitionen als Teil der Musik betrachtet. Wenn Dissonanz aber ein Teil der Musik ist, kann das Ziel einer dynamischen Stimmung nicht darin bestehen, alle Intervalle rein zu stimmen.

Bereits in Abschnitt 3.2.13 wird darauf eingegangen, wie sehr unser Gehör bereits auf die Verstimmung von Intervallen, der gleichstufigen Stimmung, konditioniert ist. Durch den vereinfachten Wechsel und Übergang zwischen verschiedenen Stimmungen kann vielleicht (wieder) ein Gefühl dafür entwickelt werden Reinheit und Unreinheit gezielt einzusetzen.

DynTune eröffnet dem Organisten den Einsatz verschiedener neuer Gestaltungsmöglichkeiten darunter die Änderung der Basis einer Stimmung (vgl. Abschnitt 4.2.1). Keines der adaptiven Stimmungssysteme, die bei der Literaturrecherche zu finden waren, bietet bisher diese Möglichkeit, deswegen lassen sich die genauen Anwendungsfelder noch nicht eingrenzen. Organisten müssen erst ein Gefühl für die Software entwickeln, bevor diese intensiv eingesetzt werden kann. Die Software bietet jedenfalls einen „Kosmos neuer Möglichkeiten“.

DynTune kann auch als *Stand-Alone*-Anwendung ohne MIDI-Eingabe oder das Schreiben von Temperierungen verwendet werden. Dieser Modus erlaubt es DynTune auch didaktisch einzusetzen, um Stimmungssysteme, Temperierungen oder die Änderung von Stimmbasen zu veranschaulichen.

5.4 Ausblick

Die Steuerung von Klangparametern, wie es auch die Stimmung einer ist, über Interfaces wie Schweller, Aftertouch oder Körpertracking sind Teil des Bereichs der Orgelforschung am Institut für Kirchenmusik und Orgel. Welche Eingabemöglichkeit sich für Stimmungsänderungen am besten eignet, muss noch erforscht werden. Erste Einsätze der Software haben aber gezeigt, dass die Verwendung des Schwelltritts sich vom Organisten leicht angeeignet werden kann und dieser relativ schnell ein Gefühl für den Umgang mit DynTune entwickeln kann.

DynTune wird am Institut für Kirchenmusik und Orgel weiter untersucht und auch weiter entwickelt werden. Das Programm wird über die Funktionalitäten erweitert werden: z. B. Die Funktionalität mit einem zusätzlichen Slider und einem Algorithmus auch den Stimmtone der Hauptwerk-Software festzulegen. Zudem soll es möglich werden, wie von Herrn Rost in Abschnitt 5.1 angestoßen, Stimmungen in einen Setzer zu laden und diese per Tastendruck abzurufen.

Eine weitere Idee beinhaltet eine Verstimmungsanzeige, die die Verstimmung des gerade gespielten Akkords anzeigt. Dazu sollen die gedrückten Tasten über das MIDI-Interface des Spielti-

sches erkannt werden. Daraus können, auf der Basis von Quinten und Terzen, Intervalle erkannt und die Verstimmung dieser Intervalle gegenüber reinen aufsummiert und angezeigt werden. Dieses Werkzeug soll dabei helfen für einzelne Teile eines Stücks den Punkt am Stimmungsvektor zu finden, an dem die Verstimmung bzw. Unreinheit minimal ist.



Klangbeispiele

Im Anhang der gebundenen Arbeit findet sich eine CD, die Klangbeispiele sowie ein Video der DynTune-Verwendung enthält. Die nachfolgende Tabelle enthält eine Beschreibung der einzelnen Dateien. Außerdem finden sich ausgewählte Beispiele auf der Webseite des Instituts für Kirchenmusik und Orgel im Bereich Orgelforschung: <http://orgel-kirchenmusik.kug.ac.at>.

Alle Klangbeispiele wurden von Gunther Rost auf der Hauptwerk-Orgel unter Verwendung eines Samplesets der Silbermann-Orgel in der Petrikirche in Freiberg gespielt und von Jan Ročnik aufgezeichnet. Die Beispiele 1–7 entstanden am 07.12.2015 am Institut für Kirchenmusik und Orgel. Die Beispiele 8–11 entstanden anlässlich eines Konzerts im Sternenturm Judenburg (vgl. Abschnitt 5.1) am 19.11.2015.

	Dateiname	Beschreibung
Bsp. 1	<code>pythagoreisch.wav</code>	C-Dur/a-Moll-Kadenz in pythagoreischer Stimmung.
Bsp. 2	<code>mitteltoenig.wav</code>	C-Dur/a-Moll-Kadenz in mitteltöniger Stimmung.
Bsp. 3	<code>silbermann.wav</code>	C-Dur/a-Moll-Kadenz in Silbermann-Stimmung.
Bsp. 4	<code>werckmeister.wav</code>	C-Dur/a-Moll-Kadenz in Werckmeister-Stimmung.
Bsp. 5	<code>gleichstufig.wav</code>	C-Dur/a-Moll-Kadenz in gleichstufige Stimmung.
Bsp. 6	<code>neidhardt.wav</code>	C-Dur/a-Moll-Kadenz in Neidhardt-Stimmung.
Bsp. 7	<code>dyntune.wav</code>	C-Dur/a-Moll-Kadenz in dynamischer Stimmung mit DynTune. Basisstimmung: reine Stimmung, Zielstimmung: gleichstufige Stimmung.
Bsp. 8	<code>goldb-silbermann.wav</code>	Johann Sebastian Bach BWV988 „Goldberg-Variationen“, Variation 25 Silbermann-Stimmung.
Bsp. 9	<code>goldb-kirnberger.wav</code>	Johann Sebastian Bach BWV988 „Goldberg-Variationen“, Variation 25 Kirnberger-Stimmung.
Bsp. 10	<code>goldb-dyntune.wav</code>	Johann Sebastian Bach BWV988 „Goldberg-Variationen“, Variation 25 dynamische Stimmung mit DynTune. Basisstimmung: Silbermann-Stimmung Zielstimmung: Kirnberger-Stimmung.
Bsp. 11	<code>goldberg-dyntune.wav</code>	Johann Sebastian Bach BWV988 „Goldberg-Variationen“, Variation 25 gleichstufige Stimmung.
Bsp. 12	<code>dyntune.mp4</code>	Video mit Schwellerbewegung und DynTune-Bedienung zu Beispiel 7.

B

Hauptwerk XML-Format

Temperierungen für die Hauptwerk-Software, werden im XML-Format abgespeichert. Die von der Hauptwerk-Software verwendete Dateiendung ist auf *.Temperament_Hauptwerk_xml festgelegt und nicht *.xml wie es von XML-Dateien zu erwarten wäre. Diese XML-Dateien enthalten neben einigen allgemeinen Informationen über die Temperierung eine Liste von 133 Noten und den dazugehörigen Frequenzen. Diese 133 Noten entsprechen grundsätzlich den „MIDI Note Numbers“, reichen aber nicht bis 127, wie es bei MIDI der Fall ist, sondern bis 133, was einem Umfang von 11 Oktaven und einer Note entspricht. Tabelle B.1 zeigt den Zusammenhang zwischen „MIDI Note Numbers“ und Notennamen.

Zwar werden in der Datei Noten mit Frequenzen in Verbindung gebracht, tatsächlich extrahiert Hauptwerk daraus aber lediglich die Intervallgrößen, also den Abstand der Töne zueinander. In einem weiteren Schritt wird von der Hauptwerk-Software intern auf das mittlere A normalisiert, dessen Frequenz über das User-Interface festgelegt wird.

Eine Dokumentation des Formats ist nicht öffentlich zugänglich, nachfolgender Ausschnitt zeigt ein XML-Schema, das aus dem gesammelten Wissen²⁹ erstellt wurde.

```

1 <?xml version="1.0" ?>
2 <xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
   attributeFormDefault="unqualified"
   elementFormDefault="qualified">
3   <xs:element name="Hauptwerk" type="HauptwerkType"/>
4   <xs:complexType name="_GeneralType">
5     <xs:sequence>
6       <xs:element type="xs:integer" name="Sys_ObjectID"/>
7       <xs:element type="xs:integer" name="UniqueTemperamentID"/>
8       <xs:element type="xs:string" name="Name"/>
9       <xs:element type="xs:string" name="ShortName"/>
10      <xs:element type="xs:integer" name="SupplierID"/>
11      <xs:element type="xs:string" name="SupplierName"/>
12      <xs:element type="xs:string" name="Comments"/>
13      <xs:element type="xs:string" name="TemperamentVersion"/>
14    </xs:sequence>
15  </xs:complexType>
16  <xs:complexType name="ObjectListType">
17    <xs:sequence>
18      <xs:element type="_GeneralType" name="_General"
19        minOccurs="0"/>
20      <xs:element type="noteType" name="note" maxOccurs="unbounded"
21        minOccurs="0"/>
22    </xs:sequence>
23    <xs:attribute type="xs:string" name="ObjectType"
24      use="optional"/>
25  </xs:complexType>
26  <xs:complexType name="noteType">
27    <xs:sequence>

```

²⁹ Recherche an den verfügbaren Temperierungen am Institut und im Internet.

```

25     <xs:element type="xs:integer"
        name="MIDINoteNumberOnEightFootStop" />
26     <xs:element type="xs:decimal" name="PitchHz" />
27   </xs:sequence>
28 </xs:complexType>
29 <xs:complexType name="HauptwerkType">
30   <xs:sequence>
31     <xs:element type="ObjectListType" name="ObjectList"
        maxOccurs="unbounded" minOccurs="0" />
32   </xs:sequence>
33   <xs:attribute type="xs:string" name="FileFormat" />
34   <xs:attribute type="xs:decimal" name="FileFormatVersion" />
35 </xs:complexType>
36 </xs:schema>

```

Beschreibung der wichtigsten Tags:

<UniqueTemperamentID > wird vermutlich von der Hauptwerk-Software intern verwendet, um die Temperierung zu identifizieren. Diese sollte sich von den anderen im System verwendeten Temperierungen unterscheiden. In der Arbeit hat sich eine ID von 999999 als hilfreich erwiesen.

<Name> gibt den Namen der Temperierung an, der in der Hauptwerk-Software in Menüs verwendet wird.

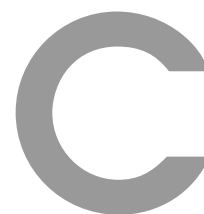
<ShortName> gibt die Kurzbezeichnung der Temperierung an, die in der Hauptwerk-Software in der Pitch-Anzeige sowie im Setzer für Temperierungen auftaucht.

<MIDINoteNumberOnEightFootStop> ist die „Midi Note Number“ des Tones, die anhand der Zuweisung aus Tabelle B.1 erfolgt. Die Bezeichnung MIDI trifft nur bedingt zu, da der Tonumfang von 128 Tönen bei MIDI auf 133 erweitert wird.

<PitchHz> ist die Frequenz des Tones in Herz. Diese kann als Dezimalzahl mit „.“ als Dezimaltrennzeichen angegeben werden.

Oktave	Midi Note Number											
	C	C#	D	D#	E	F	F#	G	G#	A	A#	B
0	0	1	2	3	4	5	6	7	8	9	10	11
1	12	13	14	15	16	17	18	19	20	21	22	23
2	24	25	26	27	28	29	30	31	32	33	34	35
3	36	37	38	39	40	41	42	43	44	45	46	47
4	48	49	50	51	52	53	54	55	56	57	58	59
5	60	61	62	63	64	65	66	67	68	69	70	71
6	72	73	74	75	76	77	78	79	80	81	82	83
7	84	85	86	87	88	89	90	91	92	93	94	95
8	96	97	98	99	100	101	102	103	104	105	106	107
9	108	109	110	111	112	113	114	115	116	117	118	119
10	120	121	122	123	124	125	126	127	128	129	130	131
11	132											

Tabelle B.1: „MIDI Note Numbers“ für die in der Hauptwerk-Software verwendeten 133 Töne einer Temperierung. Das mittlere A, der Ton, der Hauptwerk intern zur Normalisierung dient, ist hervorgehoben.



DynTune-Dokumentation

C.1 DynTuneGUI

DynTuneGUI is a graphical user interface for DynTune, a dynamic tuning system. It is inherited by QMainWindow.

C.1.1 Static Class Members

NOTE_NAMES is an array containing 12 elements which are used as labels for the notes of an octave.

TEMPERAMENT_SUBDIR is the directory where Hauptwerk puts its temperaments, relative to the installation directory. Stays the same in every installation.

TEMPERAMENT_EXT is the extension of the Hauptwerk temperament files.

C.1.2 Class Variables

dt is an instance of the DynTune backed.

tplotter is an instance of IntervallPlotter used for thirds.

qplotter is an instance of IntervallPlotter used for fifths.

sliders is an array containing 12 QSliders for per note tuning.

vector_slider is the QSlider for the panning between the two temperaments.

slider_labels is the QLabel of the sliders for per note tuning.

base_tb_group is the QButtonGroup for selecting the tuning base of the base temperament.

target_tb_group is the QButtonGroup for selecting the tuning base of the target temperament.

bt_label is the QLabel giving information for the selected base tuning.

ct_label is the QLabel giving information for the current tuning.

tt_label is the QLabel giving information for the selected target tuning.

vs_label is the QLabel providing current position of the vector_slider.

last_filename is a String containing the path where a file should be saved.

hw_temperament_dir stores the full path of the Hauptwerk temperament directory.

bt_mod is a Boolean, True if the base temperament was modified by the user.

tt_mod is a Boolean, True if the target temperament was modified by the user.

C.1.3 Class Functions

__init__(conf=None)

The initialization function for the DynTuneGUI. Constructs a new `QTMainWindow` and connects to a new instance of the DynTune system. If `conf` is the path to an existing configuration file, this file will be used. Else it will check for a configuration file in its directory or create a default configuration, this is done using `read_config` or `get_default_conf`. Also the `DynTune.tuning_changed` signal will be connected to `update_gui`.

initUI()

The initialization of the user interface is called by the `__init__` function. It creates all QT-elements needed, initializes them and shows up the graphical user interface when finished.

createUI_menubar()

Should be called by called by `initUI` and creates the menu bar showing up in the user interface.

createUI_vector_slider_box(p)

Should be called by `initUI` and creates a widget containing `vector_slider` which is used to set the panning between base and target tuning. `p` will be used as parent for the widget.

createUI_tuning_slider_box(p)

Should be called by `initUI` and creates a widget containing `sliders` and `slider_labels` which are used to set the per note tuning. `p` will be used as parent for the widget.

createUI_base_tb_button_box(p)

Should be called by `initUI` and creates a widget containing `base_tb_group` which is used to set the tuning base of the base tuning. `p` will be used as parent for the widget.

createUI_target_tb_button_box(p)

Should be called by `initUI` and creates a widget containing `target_tb_group` which is used to set the tuning base of the target tuning. `p` will be used as parent for the widget.

createUI_intervall_plotter_box(p)

Should be called by `initUI` and creates a widget containing `tplotter` and `qplotter` which are used to show the detuning of the thirds and fifths. `p` will be used as parent for the widget.

createUI_info_box(p)

Should be called by `initUI` and creates a widget containing `bt_label`, `ct_label` and `tt_label` which are used to show informations about the base, current and target tuning. `p` will be used as parent for the widget.

closeEvent(event)

Overwrites the `closeEvent` function of `QMainWindow`. When the application is closed this function will be called. It stops all running threads to ensure a safe shut down (no file open or incomplete files).

revert_tunings()

Calls `DynTune.revert_tunings()` and so triggers a revert of changes made by the user in the base and target temperament. Afterwards it updates the user interface accordingly.

update_temperament_labels()

Calls `DynTune.get_temperament_names()` to get the names of the base and target tuning to update `bt_label` and `tt_label`. This function is called to initialize the two labels and every time a new temperament is loaded through `open_base_tuning()` or `open_target_tuning()`.

open_base_tuning()

Shows a `QFileDialog` to choose a new base tuning. If a file has been selected it calls `DynTune.set_base_temperament_path` which will update the backed.

open_target_tuning()

Shows a `QFileDialog` to choose a new target tuning. If a file has been selected it calls `DynTune.set_target_temperament_path` which will update the backed.

save_as()

Shows a `QFileDialog` to save the current tuning and sets `last_filename` to this path. Then `save()` will be called.

save()

Checks if `last_filename` is set. If it is `None` `save_as` will be called otherwise `DynTune.save_current_tuning` to save the current tuning to the given path.

show_about_dialog()

Shows a small pop up with informations about the program version and contact information.

vector_slider_changed(value)

The function is connected to the SLIDER_CHANGED signal of `vector_slider`. It is called every time the slider is moved by the user. It calls `DynTune.set_tuning_vector_position`. Afterwards it checks if the slider is fully moved to the current or target tuning, if so it enables the per note tuning sliders through calling `set_tuning_sliders_enable` otherwise it disables them.

set_tuning_sliders_enable(en)

Sets the state of the per note tuning sliders in `sliders` to `en`.

tuning_slider_changed(v)

The function is connected to the SLIDER_CHANGED signal of all sliders in `sliders`. It is called every time one of the sliders is moved by the user. It checks if `vector_slider` is fully moved to the base or target tuning and calls `DynTune.set_all_base_tuning` respectively `DynTune.set_all_target_tuning` to set the per note tuning.

base_tb_group_clicked(id)

The function is connected to the BUTTON_CLICKED signal of all buttons in `base_tb_group`. If a button is pressed the method is called with `id` as an argument for the index of the clicked button. It calls `DynTune.set_bt_tuning_base` to set the new tuning base for the base tuning.

target_tb_group_clicked(id)

The function is connected to the BUTTON_CLICKED signal of all buttons in `target_tb_group`. If a button is pressed the method is called with `id` as an argument for the index of the clicked button. It calls `DynTune.set_tt_tuning_base` to set the new tuning base for the target tuning.

set_status_text(text)

Displays the string given in `text` in the status bar.

set_bt_modified(mod)

Sets the `bt_mod` to `mod`. This will indicate if the base tuning was modified by the user; if so, the `bt_label` will be appended by a red „(M)“.

set_tt_modified(mod)

Sets the `tt_mod` to `mod`. This will indicate if the target tuning was modified by the user; if so, the `tt_label` will be appended by a red „(M)“.

update_gui()

Updates the user interface by getting the current per note tuning through calling `DynTune.get_current_tuning` and sets the sliders in `sliders` accordingly. Sets the current panning of the vector slider through calling `DynTune.get_tuning_vector_position`. Sends the values received by `DynTune.get_fifths_detune` to `qplotter`. Sends the values received by `DynTune.get_thirds_detune` to `tplotter`. Afterwards calls `QApplication.processEvents` to

give the application time to process all Events occurs in the meantime. The major part of the events should have something to do with the user interface update.

read_config(config_filename)

Uses the python configparser module to read all configuration values given in `config_filename` file and returns a dictionary containing these values.

get_default_conf()

Generates a dictionary that contains the minimum needed configuration values and returns it.

C.2 IntervallPlotter

IntervallPlotter is a QT-widget used to plot the size of 12 musical intervals, in cent, as a bar chart. It is inherited by `matplotlib.backends.backend_qt5agg.FigureCanvasQTAgg`. The Matplotlib FigureCanvas inherits `PyQt5.QtWidgets.QWidget` so it can be used as Matplot and QT-Widget.

C.2.1 Static Class Members

FONT declares which font should be used in matplotlib.

NOTE_NAMES is an array with 12 elements, containing the names of the twelve notes of an octave. It's used for the X-axes labels of the plot.

C.2.2 Class Variables

tuning is an array of size 12, containing the tuning in cent which should be displayed.

dirty is used as a flag to indicate if **tuning** has changed.

lock is a `QMutex` used for controlling the access of **tuning** and **dirty**.

timer is a `QTimer` which is started immediately and periodically calls `update_plot`.

C.2.3 Class Functions

__init__(parent=None, title=None, width=5, height=4, dpi=100, iv_size=None)

The initialization function for the IntervallPlotter. Constructs a new widget with the given **width** and **height**. **parent** will be used as the QT-parent element, **title** is the title of the plotted figure which will be displayed above of it. **dpi** is the resolution of the plot; according to some examples in the internet, 100 should be a good value for most of the screen applications. **iv_size** can be used to tell IntervallPlotter the size of the intervals that will be plotted. If **iv_size** is given and an integer value the X-axes labels of the plot will display both notes forming the interval.

compute_initial_figure(title,iv_size=None)

The function will compute and draw the initial figure - a plot where all intervals have the size of zero. It draws labels, axes and the title. This function should be called by `__init__` once the `IntervallPlotter` is created. Later on `update_plot` will only redraw the bars and leave the rest as it is.

set_tuning(tuning)

Acquires the internal lock and sets the class variable `tuning` to the given `tuning`. The `dirty` flag will be set to `True` and the lock released.

update_plot()

This function should only be called by `timer`, which has been setted up in `__init__`. It acquires the internal lock only to make a copy of the `tuning`, sets the `dirty` flag to `false` and releases the lock. Afterwards it updates values of the bars in the plot and redraws them.

C.3 HWXml

HWXml is a XML parser with only small functionality. It is designed to read and write temperaments in the Hauptwerk XML format.

C.3.1 Static Class Members

None

C.3.2 Class Variables

None

C.3.3 Class Functions

__init__()

The initialization function of HWXml. Due all functions can be accessed static, no initialization is required and this function stays empty.

parse_tp(file_path)

Parses an existing temperament file located at the given path as `file_path` as the absolute path. The function returns an array with the temperament short name at the first entry and the frequency to note mapping at the second entry. The frequency to note mapping is an array containing 133 frequencies for the notes described in chapter B.

`create_tp(mapping, path)`

Writes a frequency to note mapping to a XML-file. `mapping` is an array containing 133 frequencies for the notes described in chapter B. `path` is the absolute path where the file will be written to, including its file name extension. The given temperament name in the created file will always be *HW-DTS*.

C.4 DynTune

DynTune is a dynamic tuning system, currently fitted only for the Hauptwerk software. It requires an interface such as DynTuneGUI to run. For design reasons DynTune is inherited by QObject.

C.4.1 Static Class Members

`default_organ_base_pitch` is a dictionary containing `note` and `frequency` which are used as base pitch if no other is given at initialization.

`tuning_change` is a pyqtSignal which signals (in e.g. DynTuneGUI) that the tuning has been changed.

C.4.2 Class Variables

`hw_temperament_dir` is the directory where Hauptwerk stores its temperaments.

`base_temperament_path` is the path to the temperament which is used for the base tuning.

`target_temperament_path` is the path to the temperament which is used as target tuning.

`base_tuning_name` is the name of the base tuning, extracted from the temperament file.

`target_tuning_name` is the name of the target tuning, extracted from the temperament file.

`equal_tuning` is an internal reference; numpy array of 12 elements, representing the equal tuning in cent.

`base_tuning` is a numpy array of 12 elements, containing the base tuning in cent.

`bt_tuning_base` is the note number of the tuning base note for the base tuning.

`target_tuning` is a numpy array of 12 elements, containing the target tuning in cent.

`tt_tuning_base` is the note number of the tuning base for the target tuning.

`current_tuning` is a numpy array of 12 elements, containing the current tuning in cent.

`tuning_vector` is a numpy array of 12 elements, representing the vector from base tuning to target tuning.

`tuning_vector_position` stores the current position on the tuning vektor.

`hw_thread` is the thread for Hauptwerk communication. Instance of DynTuneHWUpdate.

`vector_lock` is a QMutex to control the access of `tuning_vector` and `current_tuning`

vector_position_lock is a QMutex to control the access of **tuning_vector_position**.

midi is a configuration dictionary containing **inport** (input port name), **outport** (output port name), **out-channel**, **out-control**, **out-value**, **in-channel**, **in-value**, **in-value-min** and **in-value-max**.

midi_inport is the MIDO input port.

midi_outport is the MIDO output port.

midi_thread is the thread for incoming MIDI communication. Instance of DynTuneMidiReceiver.

C.4.3 Class Functions

__init__(tdir,base,target,organ_base_pitch,midi=None,qtparent=None)

Initialization function for DynTune, sets up the midi ports, reads temperament files and initializes all class members. **tdir** is the absolute path of the Hauptwerk temperament directory, **base** is the absolute path to the base temperament file, if it is **None** the internal equal temperament will be used. **target** is the absolute path to the target temperament file, if it is **None** the internal equal temperament will be used. **organ_base_pitch** is a dictionary containing the organ base pitch as described in C.4.1. **midi** is a dictionary containing all midi parameters as described in C.4.2. **qtparent** is the QT parent application, this parameter is necessary to let all sub-threads to be scheduled by the QT parent application.

__del__()

The function is called by Python when the Object is destroyed, it guarantees a save shut down - in e. g. opened files are closed and hardware locks are released.

init_hwmts()

The initialization function is called by **__init__** and **revert_tunings()** it sets up **base_tuning** and **target_tuning** and triggers a recalculation (reinitialization) of the class members **tuning_vector** and **current_tuning**.

load_tuning_file(filename)

Loads a Hauptwerk temperament file given by an absolute path in **filename** using HWXml (follow up section C.3) and calculates the tuning values for one octave in cent. The therefore used octave is always that one which is assigned to the midi note numbers 60 - 72 (compare section B). The tuning in cent is calculated relative to the note given in **organ_base_pitch** for the reason that this note is the only one that keeps a fixed frequency on all operations. The tuning is assigned to a numpy array with size of 12. The function returns an array with the tuning vector at position zero and the name that was given by the temperament file at position one.

recalculate_tuning()

The function uses `base_tuning` and `target_tuning` to calculate the `tuning_vector` (a position vector), between them. Afterwards it calculates the `current_tuning`, which is at `tuning_vector_position` on `tuning_vector`. For the calculation the `vector_lock` is acquired. After the calculation the `tuning_changed` signal is emitted.

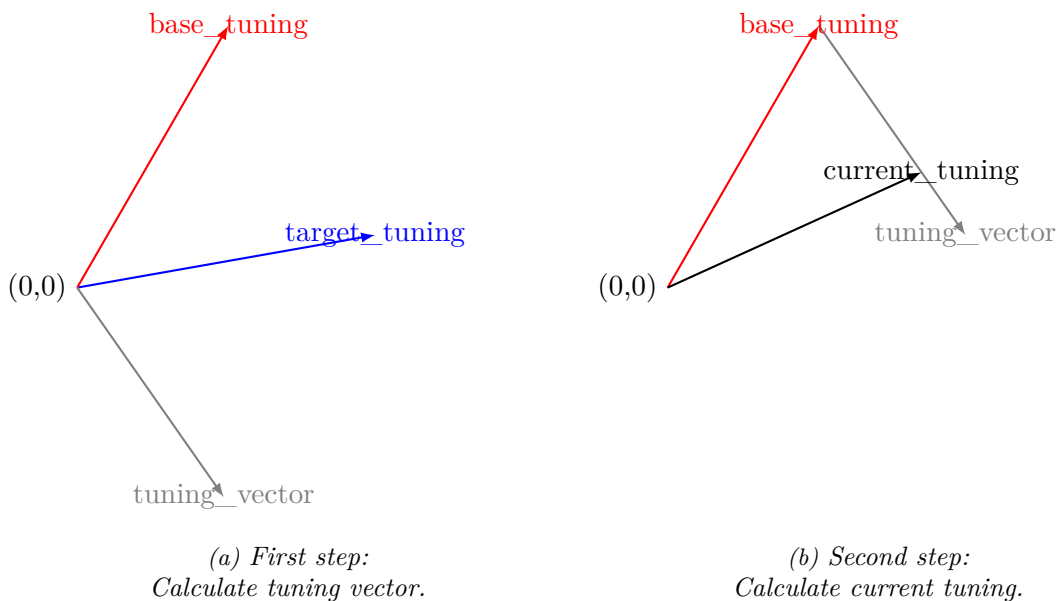
```

1 def recalculate_tuning(self):
2     self.vector_lock.lock()
3     # First step: Calculate tuning_vector
4     self.tuning_vector = self.target_tuning - self.base_tuning
5     # Second step: Calculate current_tuning
6     self.current_tuning = self.base_tuning + ( \
7         (self.tuning_vector_position/100) * \
8         self.tuning_vector)
9     self.vector_lock.unlock()
10    self.tuning_change.emit()

```

The following Figure shows the calculation of the current tuning for a 2-dimensional (2-note) system. Note that in DynTune 12-dimensions are required.

In a first step the tuning vector is calculated as subtraction between base and target tuning. In the second step a designated length of this tuning vector is taken and added to the base tuning. The resulting vector is the current tuning.

**update_hauptwerk()**

The function calls `get_frequency_mapping` to retrieve the current tuning as an array containing 133 frequencies according to the 133 midi note numbers described in Appendix B, then `HWXml` (follow up section C.3) is used to generate a temporary tuning file. If the file was successfully created it is renamed to `hwmts.Temperament_Hauptwerk_xml` in the Hauptwerk directory. Finally `notify_hauptwerk_midi` is called.

notify_hauptwerk_midi()

Sends a midi message on `midi_outport` with the parameters given in `midi` dictionary (see section C.4.2), only if `midi` exist.

save_current_tuning(filename)

Similar to `update_hauptwerk` but saves the tuning to the in `filename` given path.

start_hw_updates()

Creates and starts the `hw_thread` if it not exists.

stop_hw_updates()

Stops the `hw_thread` if it exists.

start_midi()

Opens the midi ports for input and output, then creates and starts the `midi_thread` if it not exists and midi parameters were given in the `midi` (see section C.4.2).

stop_midi()

Stops the `midi_thread` if it exists, closes the midi input and output port.

get_fifths_detune()

Calculates the detune of the fifths within the `current_tuning`. For the time the functions accesses the `current_tuning` the `vector_lock` is acquired. As reference a perfect fifth with a ratio of $3/2$ is used. The function returns an array with 12 elements each representing, the detune of one of the fifth in an octave, in cent.

```
1 def get_fifths_detune(self):
2     # Calculate the size of a fifth in cent
3     clean_fifths = 1200*np.log(3/2)/np.log(2)
4     self.vector_lock.lock()
5     # Create a vector containing the upper note of all fifths
6     # within an octave. Therefor take the tuning of two octaves:
7     # current_tuning is the first
8     # current_tuning + 1200 is the second
9     # From this vector take the elements 7 to 19 which are
10    # the upper notes of the fifths to the notes in current_tuning
11    shift_vector = np.append(self.current_tuning,
12                             self.current_tuning + 1200)[7:19]
13    # Calculate the size of all fifths at once.
14    fifths = deepcopy( shift_vector - self.current_tuning )
15    self.vector_lock.unlock()
16    # For the fifths detune simply subtract the size of a perfect
17    fith.
18    fifths_detune = fifths - clean_fifths
19    return fifths_detune
```

get_thirds_detune()

Calculates the detuning of the thirds within the `current_tuning`. For the time the functions accesses the `current_tuning` the `vector_lock` is acquired. As reference a perfect third with a ration of $5/4$ is used. The function returns an array with 12 elements each representing, the

detune of one of the third in an octave, in cent. Similar to `get_thirds_detune`. See sec. C.4.3 for proper explanation.

`get_temperament_names()`

Returns an array containing `base_tuning_name` and `target_tuning_name` in this order.

`get_current_tuning(eq_ref=False)`

Returns an array of size 12 representing the current tuning in cent for the 12 notes of an octave. If `eq_ref` is `True`, `equal_tuning` will be subtracted and the function will return the detune of the 12 notes from their values in an equal tuning. The function acquires `vector_lock` makes a copy of `tuning_vector` and releases it.

`get_tuning_vector_position()`

Returns the current value of `tuning_vector_position`. The function acquires the `vector_position_lock` makes a copy of `tuning_vector_position` and releases it.

`set_tuning_vector_position(position)`

The function acquires `vector_position_lock`, sets `tuning_vector_position` to `position` and releases it.

`get_frequency_mapping()`

Acquires the `vector_lock` and makes a copy of `current_tuning` then releases it again. The function converts the given cent values into frequencies. The in `organ_base_pitch` (see section C.4.1) defined frequency is assigned to the note, also given there. The resulting array of 12 frequencies is recalculated for eleven octaves and one note, resulting in 133 notes for the 133 midi notes described in B. The function returns an array containing 133 frequencies.

```

1 def get_frequency_mapping(self):
2     # Get the current tuning in cent.
3     self.vector_lock.lock()
4     shift =
5         self.current_tuning[self.organ_base_pitch.get('note_index')]
6     tuning_cent = deepcopy( self.current_tuning - shift )
7     self.vector_lock.unlock()
8     # tuning_hz is the tuning for one octave in hz
9     # with the tone number note_index set to
10    # the frequency given in organ_base_pitch
11    tuning_hz = np.power(2,np.divide( tuning_cent , 1200 ) ) *
12        self.organ_base_pitch.get('frequency')
13    # we want 11 octaves, currently we have the tuning
14    # for the middle one, so we need additionally
15    # 5 octaves before and 5 octaves after
16    # music is a vector like [...,1/4,1/2,1,2,4,...]
17    music = np.array([np.logspace(-5,5, num=11,base=2)])
18    # we multiply this with our tuning for one octave
19    # and get a matrix with 11 rows
20    # each representing one octave
21    # as we want a simple array
22    # we shrink it back to no dimension (-1)
23    mapping = (music.T*tuning_hz).reshape(-1)

```

```

22 # somehow hauptwerk wants to get 133 notes...
23 # ... which refers to 11 octaves + 1 note
24 # so the last note is the first tone of the
25 # 12th octave in whole, or with other words:
26 # the middle c + 6 octaves
27 mapping = np.append(mapping, tuning_hz[0]*np.power(2,6))
28 return mapping

```

Equation C.1 shows the lines 15 and 21 from the source code. Equation C.2 shows the line 27, the complete mapping. In between the dimension is reduced to 1.

$$\underbrace{\begin{pmatrix} \vdots \\ \frac{1}{4} \\ \frac{1}{2} \\ 1 \\ 2 \\ 4 \\ \vdots \end{pmatrix}^T}_{\text{music}} \cdot \underbrace{\begin{pmatrix} f_c & \cdots & f_a & \cdots & f_b \end{pmatrix}}_{\text{tuning_hz}} = \underbrace{\begin{bmatrix} \cdots \\ \frac{1}{4} \cdot \text{tuning_hz} \\ \frac{1}{2} \cdot \text{tuning_hz} \\ 1 \cdot \text{tuning_hz} \\ 2 \cdot \text{tuning_hz} \\ 4 \cdot \text{tuning_hz} \\ \cdots \end{bmatrix}}_{\text{mapping}} \quad (\text{C.1})$$

$$\text{mapping} = \left[\cdots \frac{1}{2} \cdot \text{tuning_hz} \quad 1 \cdot \text{tuning_hz} \quad 2 \cdot \text{tuning_hz} \quad \cdots \quad f_c \cdot 2^6 \right] \quad (\text{C.2})$$

set_tt_tuning_base(index)

Sets the tuning base of the target tuning to the note number **index**. The note number is an integer between 0 and 11 corresponding to the notes from C to B. The function acquires the **vector_lock** sets the target tuning accordingly and releases the lock. Afterwards **recalculate_tuning** is called. The current tuning base note number is assigned to **tt_tuning_base**.

```

1 def set_tt_tuning_base(self, index):
2     # compared to the actual tuning base
3     # assigned to self.tt_tuning_base:
4     # how many notes the tuning bases moves
5     # forward or backward
6     rot = index - self.tt_tuning_base
7     # rotate the tuning vector based on the equal tuning
8     # roll does a rotation where the last element gets
9     # the first and visa vers.
10    # equal_tuning is subtracted to get relative values
11    new_tuning = np.roll(self.target_tuning - self.equal_tuning, rot)
12    # shift the tuning so that the base pitch doesn't change
13    new_tuning = new_tuning -
        new_tuning[self.organ_base_pitch.get('note_index')]
14    # add the equal_tuning to get again absolute values
15    self.target_tuning = new_tuning + self.equal_tuning
16    self.tt_tuning_base = index
17    # trigger recalculation
18    self.recalculate_tuning()

```

set_bt_tuning_base(index)

Sets the tuning base of the base tuning to the note number `index`. The note number is an integer between 0 and 11 corresponding to the notes from C to B. The function acquires the `vector_lock` sets the base tuning accordingly and releases the lock. Afterwards `recalculate_tuning` is called. The current tuning base note number is assigned to `bt_tuning_base`. The function is similar to `set_tt_tuning_base` a proper explanation can be found in sec. C.4.3.

set_note_base_tuning(note_index,value,eq_ref=False)

Sets the note with the note number given as `note_index` in the `base_tuning` to `value`. The note number is an integer between 0 and 11 corresponding to the notes C to B. If `eq_ref` is `True` the equal tuning will be used as reference, so `value` indicates the detuning of the note from the equal tuning and not the absolute value. (Remark: All values in cent.) Afterwards `recalculate_tuning` is called.

set_note_target_tuning(note_index,value,eq_ref=False)

Sets the note with the note number given as `note_index` in the `target_tuning` to `value`. The note number is an integer between 0 and 11 corresponding to the notes C to B. If `eq_ref` is `True` the equal tuning will be used as reference, so `value` indicates the detuning of the note from the equal tuning and not the absolute value. (Remark: All values in cent.) Afterwards `recalculate_tuning` is called.

set_all_base_tuning(values,eq_ref=False)

Similar to `set_note_base_tuning` but sets all 12 notes at once to the array given as `values`. If `eq_ref` is `True` the equal tuning will be used as reference, so `values` indicates the detuning of the notes from the equal tuning and not the absolute values. (Remark: All values in cent.) Afterwards `recalculate_tuning` is called.

set_all_target_tuning(values,eq_ref=False)

Similar to `set_note_target_tuning` but sets all 12 notes at once to the array given as `values`. If `eq_ref` is `True` the equal tuning will be used as reference, so `values` indicates the detuning of the notes from the equal tuning and not the absolute values. (Remark: All values in cent.) Afterwards `recalculate_tuning` is called.

set_base_temperament_path(filename)

Sets `base_temperament_path` to `filename` and calls `init_hwdts`.

set_target_temperament_path(filename)

Sets `target_temperament_path` to `filename` and calls `init_hwdts`.

revert_tunings()

Calls `init_hwdts`.

C.5 DynTuneHWUpdate

DynTuneHWUpdate is a thread to control cyclic updates of the DynTune temperament file in the Hauptwerk directory. Due the fact that there is no interface to Hauptwerk which provides feedback, this thread ensures that Hauptwerk gets a useful amount of time to read the temperament file. The Thread waits for a tuning change but triggers updates of the temperament file with a maximum update time. It is inherited by QThread.

C.5.1 Static Class Members

MAX_UPDATE_INTERVAL The minimum time to wait before between two updates of the Hauptwerk temperament file in seconds.

C.5.2 Class Variables

ready is a boolean that indicates if the thread has stopped working. By default it is set to **True**.

stopped is a boolean that tells the thread to stop, by default it is set to **False**.

dirty is a boolean that indicates that the tuning has changed and Hauptwerk needs an update.

lock is a QMutex to control the access of **dirty**. Used as internal lock.

condition is a QWaitCondition to control control waiting and running state of the thread.

dyntune is the reference to the DynTune system used to communicate with Hauptwerk.

C.5.3 Class Functions

__init__(dyntune=None, parent=None)

Initialize the thread, **dyntune** is an instance of the DynTune system used for the communication with Hauptwerk, **parent** is a QObject and needed to add the thread to be scheduled by the QT main application.

trigger()

The function acquires the internal lock is acquired to set **dirty** to **True**. After the lock has been released the thread will be woken if it is waiting for **condition**.

```
1 def trigger(self):
2     self.lock.lock()
3     self.dirty = True
4     self.lock.unlock()
5     # wake all threads waiting for this condition
6     # this should only be this one itself
7     self.condition.wakeAll()
```

stop()

Sets the variable **stopped** to **True** which will have as result that the thread stops running and finishes. After the variable has been set the thread will be woken if it is waiting for **condition**.

is_ready()

Returns the state of `ready`. `True` indicates that the thread has been finished his work or it has not been started.

run()

Work routine of the thread. Should not be called from outside. Calls `DynTune.update_hauptwerk` if `trigger` was called with a maximum of one call in `MAX_UPDATE_INTERVAL`.

```

1 def run(self):
2     # set ready to false – the thread starts working
3     self.ready = False
4     # the thread stops if stopped is set to True
5     while not self.stopped:
6         # Call the update_hauptwerk function
7         # this will create the new temperament file
8         # and send a message to hauptwerk
9         self.dyntune.update_hauptwerk()
10        # wait for some time before checking again
11        # so that hauptwerk has enough time to read
12        # the file
13        self.msleep(MAX_UPDATE_INTERVAL)
14        # acquire the lock and wait for the
15        # tuning to get dirty (changed) again
16        self.lock.lock()
17        if not self.dirty:
18            self.condition.wait(self.lock)
19            self.dirty = False
20            self.lock.unlock()
21        # after the threads last run, ready is set to True
22        self.ready = True

```

C.6 DynTuneMidiReceiver

DynTuneMidiReceiver is a thread listening to midi messages on a specified port. If messages arrive it sets the tuning vector position of DynTune. It is inherited by QThread.

C.6.1 Static Class Members

None

C.6.2 Class Variables

`stopped` is a boolean that tells the thread to stop, by default it is set to `False`.

`dyntune` is the reference to the DynTune system used to communicate with Hauptwerk.

C.6.3 Class Functions

`__init__(dyntune=None, parent=None)`

Initialize the thread, `dyntune` is an instance of the DynTune system. The thread uses the midi port defined in DynTune. `parent` is a `QObject` and needed to add the thread to be scheduled by the QT main application.

`stop()`

Sets the variable `stopped` to `True` which will have as result that the thread stops running and finishes.

`run()`

Work routine of the thread. Should not be called from outside. Waits for midi messages to be received on `DynTune.midi_inport`, reads the message and checks if fits to the parameters given in `DynTune.midi` (follow up C.4.2) and calls `DynTune.set_tuning_vector_position` if so.

```
1 def run(self):
2     # extract the midi parameters from
3     # the DynTune.midi dictionary
4     in_channel = self.dyntune.midi.get('in-channel')
5     in_control = self.dyntune.midi.get('in-control')
6     in_value_min = self.dyntune.midi.get('in-value-min')
7     in_value_max = self.dyntune.midi.get('in-value-max')
8     in_value_range = in_value_max - in_value_min
9     # the thread runs until stopped is False
10    while not self.stopped:
11        # midi_inport.receive is a blocking function
12        # it will wake up the thread when a messages is received
13        msg = self.dyntune.midi_inport.receive()
14        # check if the messages fits the given parameters
15        if(msg.type == 'control_change' and \
16           msg.channel == in_channel and \
17           msg.control == in_control):
18            # fit the value to the designated range
19            if msg.value > in_value_max:
20                value = in_value_max
21            elif msg.value < in_value_min:
22                value = in_value_min
23            else:
24                value = msg.value
25        # set the new value in DynTune
26        self.dyntune.set_tuning_vector_position( \
27            np.around( \
28                ((value-in_value_min) /in_value_range) * 100 \
29            ))
```

Literaturverzeichnis

- [1] H. Grabner, *Allgemeine Musiklehre*, 24th ed. Bärenreiter, 1959.
- [2] M. Pflüger, „Einleitung (Psychoakustik),“ 2014, Unterlagen zur Vorlesung: Psychoakustik 1, Universität für Musik und darstellende Kunst Graz.
- [3] S. Bleeck, „Psychophysikalische Untersuchung von spektralen und zeitlichen Mechanismen des auditorischen systems,“ Diplomarbeit, Technische Hochschule Darmstadt, 1996, letzter Zugriff: 12.12.2015. [Online]. Via: www.tonhoehe.de
- [4] D. Muzzolini, *Genealogie der Klangfarbe*. Peter Lang AG, 2006.
- [5] A. Seebeck, „Beobachtungen über einige Bedingungen der Entstehung von Tönen,“ *Annalen der Physik*, vol. 129, no. 7, pp. 417–436, 1841.
- [6] C. Lichtblau, „Tonhöhenwahrnehmung – Zeitlicher Abriss der Forschung,“ Seminararbeit, Universität Ulm, 2004, letzter Zugriff: 12.12.2015. [Online]. Via: www.informatik.uni-ulm.de/ni/Lehre/SS04/HSSH/pdfs/TonhoehenI.pdf
- [7] A. J. M. Houtsma and J. L. Goldstein, „The central origin of the pitch of complex tones: Evidence from the musical interval recognition.“ *The Journal of the Acoustical Society of America*, vol. 51, no. 2 (Part 2), 1972.
- [8] E. Zwicker, *Psychoakustik*. Springer-Verlag, 1982.
- [9] W. A. Sethares, *Tuning, timbre, spectrum, scale*, 2nd ed. Springer, 2005.
- [10] E. Körner, „Ein universelles Stimmgerät,“ Diplomarbeit, Technische Universität Graz, 1987.
- [11] J. Tenney, *A History of: Consonance and Dissonance*, 1st ed. Excelsior Music Publishing Company, 1988.
- [12] R. Plomp and M. Levelt, „Tonal consonance and critical bandwidth,“ *Journal of the Acoustical Society of America*, vol. 38, pp. 548–560, 1965.
- [13] A. Schönberg, *Harmonielehre*, 3rd ed. Universal Edition, 1922.
- [14] R. Janke, „Grundzüge Intonation,“ 2008, letzter Zugriff: 12.12.2015. [Online]. Via: www.orgel-info.de
- [15] R. Schröter, „Die Stimmung von Tasteninstrumenten,“ Diplomarbeit, Hochschule für Musik und Theater München, 2002, letzter Zugriff: 12.12.2015. [Online]. Via: www.robertschroeter.de/diplom.pdf
- [16] H. J. Schugk, *Praxis barocker Stimmungen und ihre theoretischen Grundlagen*, 4th ed. Selbstverlag Rolf Drescher, 1981.
- [17] K. L. Paech, „Vergleich der Centzahlen verschiedener Stimmungen,“ 2015, Unterlagen zur Vorlesung: Geschichte der Orgel und ihrer Musik, Universität für Musik und darstellende Kunst Graz.
- [18] A. Werckmeister, *Musicalische Paradoxal-Discourse*. Quedlinburg (Verlagsort), 1707.
- [19] A. D. Fokker, *Neue Musik mit 31 Tönen*. Verlag der Gesellschaft zur Förderung des systematischen Musikwissens e.V., 1966.

- [20] J. Huber and M. Mayrock, „Wohltemperiert in guter stimmung,“ in *Plauderei zu Mathematik und Musik*. Friedrich-Alexander-Universität Erlangen-Nürnberg, 2010, p. 34f.
- [21] M. Praetorius, *Syntagma musicum. Bd. 2: De Organographia*. Bärenreiter, 1619, Nachdruck Kassel 2001.
- [22] H. W. Kaufmann, „More on the tuning of the archicembalo,“ *Journal of the American Musicological Society*, vol. 23, no. 1, 1970.
- [23] C. Stembridge, „The cimbalò cromatico and other italian keyboard instruments with nineteen or more division to the octave (surviving specimens and documentary evidence),“ *Performance Practice Review*, vol. 6, no. 1, 1993.
- [24] W. Pole, „A new keyed musical instrument for just intonation,“ *Nature*, vol. 44, no. 1141, pp. 446–448, September 1891.
- [25] M. Kornemann, *Portrait Klavier*, 1841st ed. Bärenreiter, 2010.
- [26] R. Harlow, „Recent organ design innovations and the 21st-century „Hyperorgan“,“ 2011, letzter Zugriff: 12.12.2015. [Online]. Via: www.huygens-fokker.org/docs/Harlow-RecentOrganDesignInnovationsandthe21stCenturyHyperorgan.pdf
- [27] R. Wille, „Mutabor – ein medium für musikalische erfahrung,“ in *HyperKult. Geschichte, Theorie und Kontext digitaler Medien*, M. Warnke, W. Coy, and G. C. Tholen, Eds. Stroemfeld, 1997, pp. 383–389.
- [28] C. Drösser, „Die sauberste Orgel der Welt,“ *Zeit Online*, Dezember 2013, letzter Zugriff: 12.12.2015. [Online]. Via: www.zeit.de/2013/52/orgel-stimmung-computer
- [29] H. Partch, *Genesis Of A Music: An Account Of A Creative Work, Its Roots, And Its Fulfillments*, 2nd ed. Da Capo Press, 1979.
- [30] R. W. Duffin, *How Equal Temperament Ruined Harmony (And Why You should Care)*. W. W. Norton & Company, 2007.
- [31] J. Gedan, „Inharmonizität: Das inharmonische Teiltonspektrum von Saiten und seine Konsequenzen für Klang und Klavierstimmung,“ *pian-e-forte Fachwissen*, 2008, letzter Zugriff: 12.12.2015. [Online]. Via: www.pian-e-forte.de/texte/pdf/inharm.pdf
- [32] H. Hinrichsen, „Entropiebasiertes Stimmen von Musikinstrumenten,“ *Rev. Bras. Ens. Fis.*, no. 34, 2012, Übersetzung des engl. Originalartikels, Letzter Zugriff: 12.12.2015. [Online]. Via: www.physik.uni-wuerzburg.de/~hinrichsen/research/entropy/tuning/deutsch.pdf
- [33] F. Goisl, „Die Orgel: Aufbau, Funktionsweise, Klang – eine Annäherung von musikalischer und tontechnischer Seite.“ Diplomarbeit, SAE-München, 2003, letzter Zugriff: 12.12.2015. [Online]. Via: downloads.audio-community.de/sae/facharbeiten/florian-dieorgel.pdf
- [34] W. Mohrlök, „The hermode tuning system,“ Trossingen, Deutschland, 2003, letzter Zugriff: 12.12.2015. [Online]. Via: sethahres.engr.wisc.edu/paperspdf/hermode.pdf
- [35] G. Rost, „Music in motion,“ *International Arts Manager*, 2013.
- [36] J. Rocnik, „Mehrkanalige Aufnahme und Postproduktion digitaler Konzertorgeln,“ Toningenieur-Projekt, Universität für Musik und darstellende Kunst Graz, 2014, letzter Zugriff: 12.12.2015. [Online]. Via: orgel-kirchenmusik.kug.ac.at/fileadmin/media/institut-06/Dokumente/Downloads/Orgelforschung/rocnik_konzertorgeln_2014.pdf