

Toningenieur-Projekt

Neues Multi-Touch-Instrument mit visuellem Feedback

Hannes Bradl

Betreuung: Dr. Franz Zotter

Graz, 4. März 2021



institut für elektronische musik und akustik



Zusammenfassung

Ziel dieser Arbeit ist die Entwicklung eines neuartigen polyphonen Multi-Touch-Instruments, welches, im Gegensatz zu den meisten traditionellen Instrumenten, eine Tonskala frei einstellbarer Auflösung zulässt. Bei der Komposition oder Improvisation in alternativen Stimmungssystemen ist es oft schwer möglich die Konsonanz bzw. Dissonanz verschiedener Akkorde einzuschätzen, da die klassische Harmonielehre nur einen kleinen Teil des möglichen Skalenvorrats abdecken kann. Aus diesem Grund sollen während des Spielens, auf Basis eines mathematischen Konsonanzmodells, optische Hinweise auf denkbare Griffkombinationen in Echtzeit angezeigt werden. Als Spieloberfläche bietet sich, wegen seiner Flexibilität und des geringen Preises, ein Multi-Touch-Bildschirm an.

Inhaltsverzeichnis

1	Einleitung	4
2	Theoretischer Hintergrund	4
2.1	Was ist Dissonanz und wie kann man sie modellieren?	5
2.2	Dissonanz eines Klanges bzw. Akkords	7
2.3	Dissonanzkurven	8
2.4	Optimierte Spektren	10
3	Implementierung	13
3.1	Plattform und Framework	13
3.2	Implementierung der Dissonanzkurven	14
3.3	Benutzeroberfläche und Features der App	15
4	Hörversuch	17
4.1	Versuchsdesign	18
4.2	Ergebnisse	20
5	Schlussfolgerung und Ausblick	25

1 Einleitung

Die gleichstufig temperierte Stimmung, in der englischsprachigen Literatur „twelve-tone equal temperament“ oder 12-TET genannt, ist seit Jahrhunderten üblich. Für Tasteninstrumente, wie das Klavier oder die Orgel, gibt es ein umfangreiches Repertoire an Kompositionen. Vertreter der zeitgenössischen Musik begnügen sich jedoch oft nicht mit diesem herkömmlichen Stimmungssystem. Dies ist auch verständlich wenn man sich die klanglichen Kompromisse vor Augen führt, die bei der Konstruktion von 12-TET eingegangen werden. Ein weiteres Argument, das für die Verwendung alternativer Skalen spricht, ist die Erkundung neuer Intervalle und damit auch neuer Klänge. Als treffende Analogie sehe ich das Experimentieren mit neuen Farben in der Malerei.

Für beliebige Tonskalen existiert jedoch keine Harmonielehre, die das Einschätzen der Dissonanz von bestimmten Akkorden während der Komposition oder Improvisation erleichtert.

Der Musiktheoretiker William A. Sethares beschreibt in seinem Buch „*Tuning, Timbre, Spectrum, Scale*“ [Set05] ein Modell zur Bestimmung der Dissonanz von Klängen und Akkorden. Auf Basis seines Algorithmus und unter Verwendung eines Multi-Touch-Displays ist es möglich eine optische Spielhilfe zu implementieren, die beim Musizieren Hinweise zur Dissonanz verschiedener Griffkombinationen anhand einer Graustufenanzeige gibt.

Es sind zwar bereits polyphone Keyboards, wie z.B. das *Lumatone* [Hor20], mit frei einstellbarer Tonskala auf dem Markt, diese haben jedoch, neben dem hohen Preis, weitere Nachteile. Zum einen verwenden sie fixe Tasten und sind daher nicht sehr flexibel. Zum anderen bieten die Tasten kein visuelles Feedback. Das in dieser Arbeit vorgestellte Multi-Touch-Display als Benutzeroberfläche soll zeigen, wie diese Nachteile überwunden werden können.

Im Rahmen dieser Arbeit wird zuerst ein Konsonanzmodell näher betrachtet und anschließend daraus die prototypische Multi-Touch-Anwendung entwickelt. Mit dem Ziel herauszufinden wie gut die implementierte Spielhilfe funktioniert, wird nachfolgend ein Hörversuch durchgeführt.

2 Theoretischer Hintergrund

Eine eindeutige Definition des Begriffs *Dissonanz* und dessen Gegenstücks der *Konsonanz*, ist im musikalischen Kontext durch die Wandlungsfähigkeit von Stilmitteln schwer zu definieren. Es gibt dennoch verschiedene Erklärungsversuche, warum Klänge allgemein als eher dissonant oder eher konsonant wahrgenommen werden. In diesem Abschnitt wird ein mathematisches Modell vorgestellt, welches die Grundlage der Spielhilfe des Instruments darstellt. Als Hauptquelle dieses Abschnitts dient das Buch „*Tuning, Timbre, Spectrum, Scale*“ [Set05] von William A. Sethares.

2.1 Was ist Dissonanz und wie kann man sie modellieren?

Ernst Terhardt beschreibt die Musikalische Konsonanz als das Zusammenwirken von *Harmonie* und *sensorischer Konsonanz* [Ter00] (siehe Abb. 1). Die linke Hälfte des Baumdiagramms ist für mein Projekt noch nicht hilfreich, da Harmonie vor allem in der dur-moll-tonalen Musik eine Rolle spielt. Im Rahmen dieser Arbeit möchte ich jedoch, losgelöst von der klassischen zwölfstufigen Skala, einzelne Akkorde bzw. Klänge für sich betrachten und nicht im Verhältnis zu anderen Akkorden bzw. Klängen oder einem Grundton. Außerdem soll der implementierte Algorithmus möglichst simpel sein, um in Echtzeit als Spielhilfe eingesetzt werden zu können. Aus diesen Gründen wird die Harmonie-Komponente vernachlässigt.

Die rechte Hälfte des Baumdiagramms beschreibt die sensorische Konsonanz und beinhaltet laut Terhardt die psychoakustischen Empfindungsgrößen Rauigkeit, Klanghaftigkeit und Schärfe. Diese drei Größen können nicht nur im gewohnten westlichen musikalischen Kontext, sondern auch zur Beschreibung beliebiger Klänge und Geräusche verwendet werden. Im Zusammenhang mit der Wahrnehmung von Dissonanz kann Rauigkeit laut Terhardt als wichtigster Parameter angesehen werden. Diese ist auch die Basis des Dissonanzalgorithmus, der in meiner Applikation eingesetzt wird. Denn obwohl Rauigkeit in erster Linie ein Sinneseindruck ist, lässt sie sich quantitativ durch Amplitudenschwankungen, die durch Schwebungen nahe beieinander liegender Teiltöne eines Klanges entstehen modellieren.

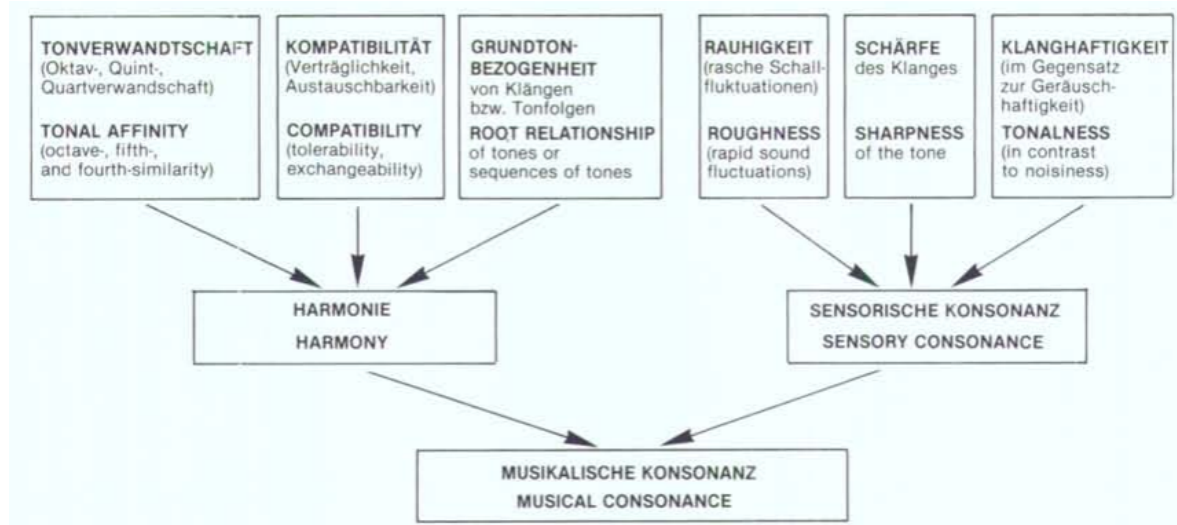


Abbildung 1 – Ein Modell der Musikalischen Konsonanz nach E. Terhardt [Ter83]

Plomp und Levelt beschreiben dazu einen Hörversuch [PL65]. Die Versuchspersonen sollten den Wohlklang (sensorische Konsonanz) verschiedener Intervalle von Sinustönen beurteilen. Herausgekommen ist die Kurve in Abb. 2.

Die wahrgenommene Dissonanz wurde am geringsten bewertet, wenn beide Töne die selbe Frequenz besitzen. Wird die Frequenz eines Tons leicht erhöht, entstehen langsame

Schwebungen mit der Schwebungsfrequenz $f_S = |f_1 - f_2|$, die noch als angenehm empfunden wurden. Bei einer größeren Distanz zwischen den beiden Tönen, ist die Schwebungsfrequenz f_S höher. Die Rauigkeit wird damit größer und führt zu einem Anstieg der Dissonanz. Oberhalb einer bestimmten Schwebungsfrequenz sind beide Sinustöne ohne Rauigkeit getrennt voneinander hörbar; das resultierende Spektrum kann wieder angenehmer klingen. Interessant ist, dass die Dissonanz den Beobachtungen nach mit dem Abstand der beiden Töne stetig sinkt, d.h. eine None wurde z.B. konsonanter als die Oktave bewertet. Dies trifft jedoch nur bei der Betrachtung von reinen Sinustönen zu, die in akustischen Musikinstrumenten kaum in Reinform auftreten.

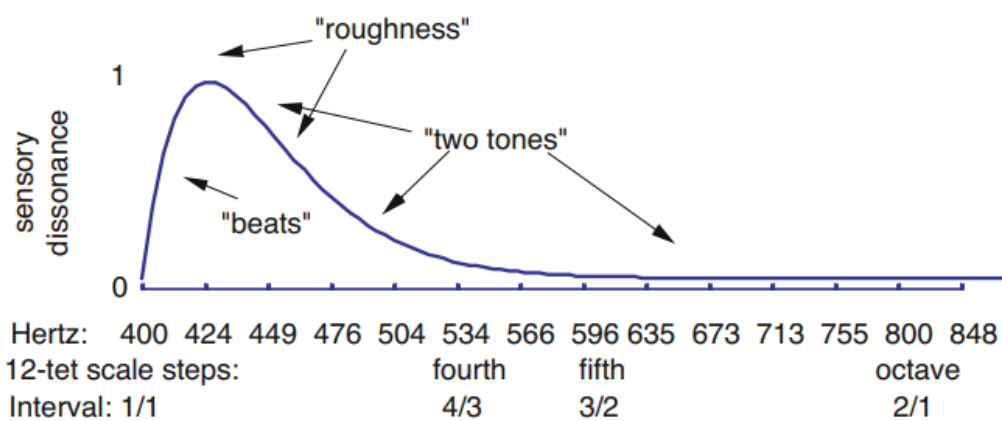


Abbildung 2 – Das Ergebnis aus dem Hörversuch von Levelt und Plomp. Die wahrgenommene Dissonanz zweier gleichzeitig klingender Sinustöne wurde für unterschiedliche Intervalle untersucht. Die Frequenz des tieferen Tons entspricht 400Hz , der höhere Ton wurde entsprechend der x-Achse variiert. [Set05]

Die Kurve aus Abb. 2 wird breiter bzw. schmaler, wenn der Grundton verändert wird (siehe Abb. 3).

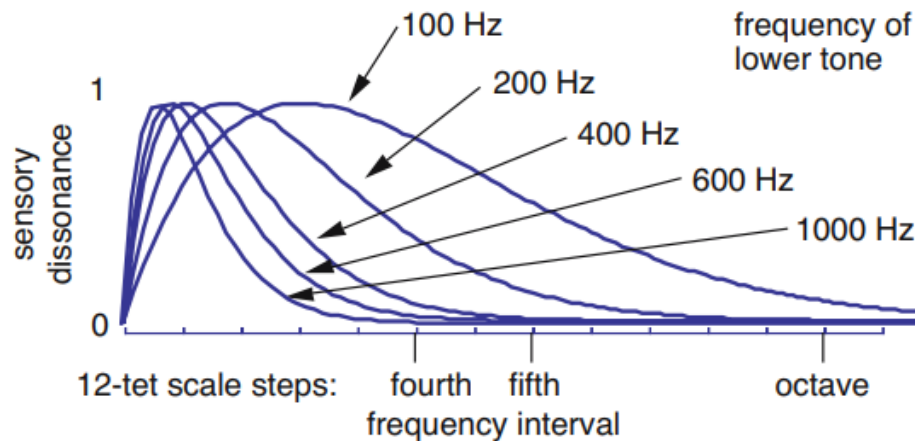


Abbildung 3 – Abhängigkeit der Dissonanzwahrnehmung zweier Sinustöne von der Frequenz des tieferen Tons. [Set05]

Bis jetzt wurden reine Sinustöne betrachtet. Die meisten klassischen westlichen Instrumente erzeugen jedoch ein mehr oder weniger harmonisches Spektrum, neben dem Grundton f erklingt nämlich auch eine Reihe an Obertönen mit den Frequenzen $2f$, $3f$, $4f$, In diesem Fall müssen alle in einem Klang oder Akkord vorkommenden Teiltöne (Sinustöne) paarweise betrachtet und für jedes Paar ein Dissonanzwert berechnet werden. Anschließend können diese Skalare summiert werden. Bei einem Dreiklang, dessen einzelne Töne aus jeweils 20 Teiltönen bestehen, sind $(3 \cdot 20)^2 = 3600$ Kombinationen zu berücksichtigen. An diesem Punkt wird ersichtlich, dass solch ein Algorithmus einen großen Rechenaufwand für ein Echtzeitsystem darstellen kann.

2.2 Dissonanz eines Klanges bzw. Akkords

In diesem Abschnitt wird ein Algorithmus von Sethares [Set05] vorgestellt, der als Eingang alle Teiltonfrequenzen inklusive deren Amplituden eines (statischen) Spektrums erhält. Bei diesem Spektrum kann es sich entweder um einen einzelnen Klang oder einen ganzen Akkord handeln. Auf Basis dieser Information wird ein Dissonanzwert errechnet. Zunächst wird die Kurve von Plomp und Levelt aus Abb. 2 wie folgt parametrisiert

$$d(x) = e^{-b_1 x} - e^{-b_2 x}. \quad (1)$$

Dabei ist x der Frequenzabstand zweier Sinustöne, also $x = |f_1 - f_2|$. Die Parameter $b_1 = 3.5$ und $b_2 = 5.75$ wurden durch Abgleich der Kurve mit dem Hörversuch von Plomp und Levelt ermittelt. Da $d(x)$ in Abhängigkeit von $f_{min} = \min(f_1, f_2)$, wie in Abb. 3 ersichtlich, gestaucht bzw. gestreckt wird, führt das Modell von Sethares noch einen Faktor $s = \frac{x^*}{s_1 f_{min} + s_2}$ mit den Konstanten $x^* = 0.24$, $s_1 = 0.021$ und $s_2 = 19$ ein. Die Berücksichtigung der Teiltonamplituden geschieht mittels Gewichtung mit $l_{min} = \min(l_1, l_2)$. Nach diesem Modell können lautere Teiltöne einen höheren Dissonanzwert

erzeugen als leisere. Zu beachten ist, dass es sich bei l_1 und l_2 nicht um Amplituden, sondern Lautheiten (in Sone) handeln soll. Die Dissonanz zweier Sinustöne errechnet sich nun wie folgt:

$$d(f_1, f_2, l_1, l_2) = l_{\min}[e^{-b_1 s|f_1-f_2|} - e^{-b_2 s|f_1-f_2|}]. \quad (2)$$

Um aus diesem Modell für ein Sinustonpaar die Dissonanz D_K eines Klages mit den Teiltonfrequenzen $\{f_1, f_2, \dots, f_n\}$ und -lautheiten $\{l_1, l_2, \dots, l_n\}$ zu bestimmen, sind die Dissonanzwerte aller Teiltonkombinationen laut Sethares aufzusummieren:

$$D_K = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n d(f_i, f_j, l_i, l_j). \quad (3)$$

Wenn m Noten in den Intervallen $\{\alpha_1, \dots, \alpha_{m-1}\}$ mit gleichartiger Teiltonzusammensetzung gleichzeitig erklingen, berechnet sich der Dissonanzwert des Gesamtklages identisch wie der eines Klages mit allen Teiltönen der Akkordbestandteile $\{f_1, \dots, f_n, \alpha_1 f_1, \dots, \alpha_1 f_n, \alpha_{m-1} f_1, \dots, \alpha_{m-1} f_n\}$. Die um die jeweiligen Intervalle verschobenen Teiltöne des Grundklages werden somit als Gesamtspektrum interpretiert und dann die Dissonanz der Menge aller vorhandenen Teiltöne berechnet.

Teiltonamplituden sind in Lautheiten umzurechnen. Für eine Sinuswelle $p(t) = A \sin(2\pi f_0 t + \phi)$ entspricht der Effektivwert des Schalldrucks $P_e = A/\sqrt{2}$. Daraus kann der Schalldruckpegel $SPL = 20 \log(P_e/P_{ref})$ mit $P_{ref} = 20 \mu\text{Pa}$ bestimmt werden. Die Lautheit wird mit der einfachen Schätzformel

$$l = \frac{1}{16} 2^{\frac{SPL}{10}} \quad (4)$$

approximiert. Die Grundlage dieser Formel ist die Tatsache, dass eine Pegeländerung um +10 dB als circa doppelt so laut wahrgenommen wird. Damit 40 dB einem Sone entsprechen, kommt der Faktor 1/16 hinzu.

2.3 Dissonanzkurven

Im vorherigen Abschnitt wurde mit Gl. 3 eine Formel präsentiert, mit deren Hilfe die Dissonanz eines Akkords parametrisch, quantitativ modelliert werden kann. Die Bewertung des aktuell klingenden Akkords kommt für vorausschauende Spielbarkeit aber zu spät. Als Spielhilfe wird ein Algorithmus benötigt, der „in die Zukunft blicken lässt“ und anhand der bereits liegenden Töne jedem potenziellen Zusatzton auf der verallgemeinerten Klaviatur einen Dissonanzwert zuordnet und anzeigt. Dazu fügt man zum bestehenden Akkord einfach einen weiteren virtuellen Ton im Intervall α_v zum Grundton hinzu. Entsprechend den Intervallen der verwendeten Skala wird α_v verändert und jeweils ein Dissonanzwert des Gesamtakkords (inklusive dem virtuellen Ton) berechnet.

Abb. 4 zeigt die entsprechende Dissonanzkurve entlang der denkbaren Intervalle für einen Klang mit *harmonischem* Spektrum mit 10 gleich lauten Teiltönen, der zusätzlich zu einem bereits liegenden Klang mit dem selben Spektrum gespielt werden würde. Diese

Kurve erhält man, indem man zum Grundklang $\{f_1, f_2, \dots, f_n\}$ einen weiteren Klang mit den Frequenzen $\{\alpha f_1, \alpha f_2, \dots, \alpha f_n\}$ hinzufügt. Nun wird α im (mathematischen) Intervall $[1, 2.1]$ schrittweise erhöht und jeweils die Dissonanz des Gesamtspektrums $\{f_1, f_2, \dots, f_n, \alpha f_1, \alpha f_2, \dots, \alpha f_n\}$ mit dem in Abschnitt 2.2 vorgestellten Modell berechnet. Abb. 4 legt nun Dissonanzminima nahe, die bei Intervallen der reinen Stimmung liegen. Bei Verwendung eines harmonischen Spektrums klingt, laut diesem Modell erwartungsgemäß, die Oktave am konsonantesten, gefolgt von der Quinte.

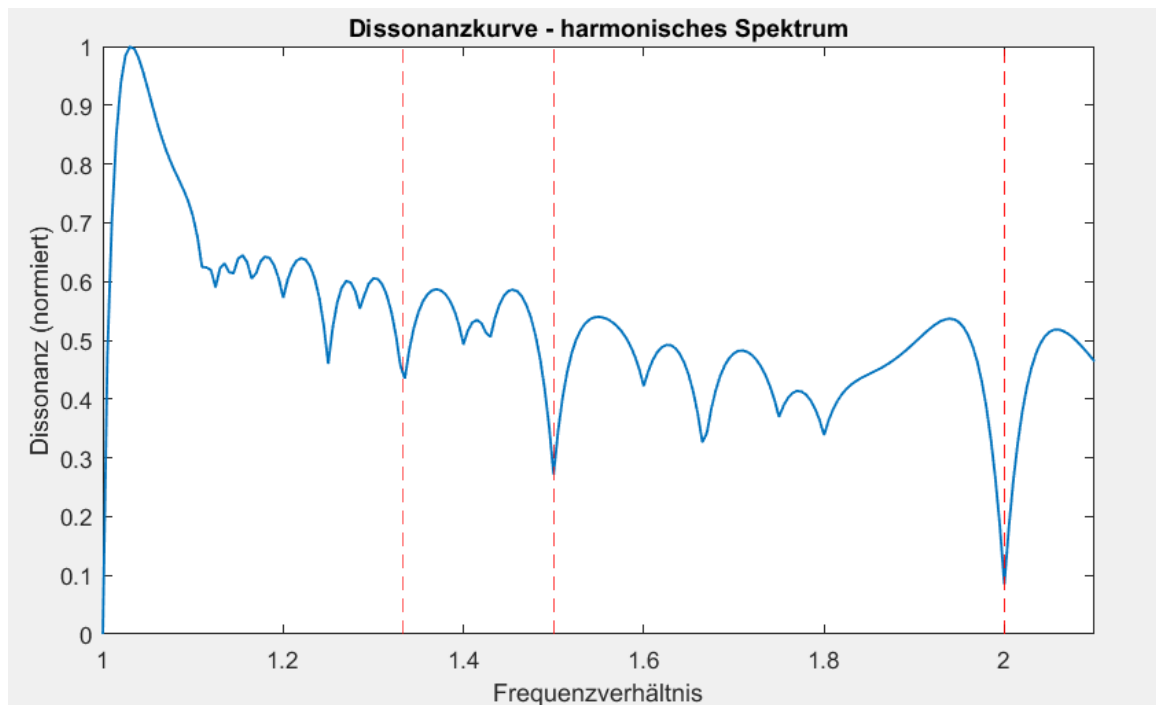


Abbildung 4 – Dissonanzkurve eines Spektrums mit 10 harmonischen Teiltönen gleicher Amplitude. Der Grundton liegt bei 400 Hz. Strichliert eingezeichnet sind die Frequenzverhältnisse 2 (Oktave), $3/2$ (reine Quinte) und $4/3$ (reine Quarte).

Abb. 5 zeigt die Dissonanzkurve für ein *inharmonisches* Spektrum mit 10 Teiltönen gleicher Lautstärke. Im Unterschied zu Abb. 4 liegen die Minima nicht mehr bei den Intervallen der reinen Stimmung.

Wie man sieht, besteht ein Zusammenhang zwischen dem Klang eines Instruments und den Minima der modellierten Dissonanzkurve. Daraus kann man laut Sethares zur Hypothese gelangen, dass für jedes Spektrum eine ideale Skala existiert, deren Intervalle in den „Tälern“ der Dissonanzkurve liegen. Ein Beispiel dazu sind die klassischen Gamelan-Instrumente Indonesiens: Aus den inharmonischen, glockenähnlichen Klängen dieser Instrumente könnten sich deren unregelmäßige fünf- und siebenstufige Tonleitern entwickelt haben.

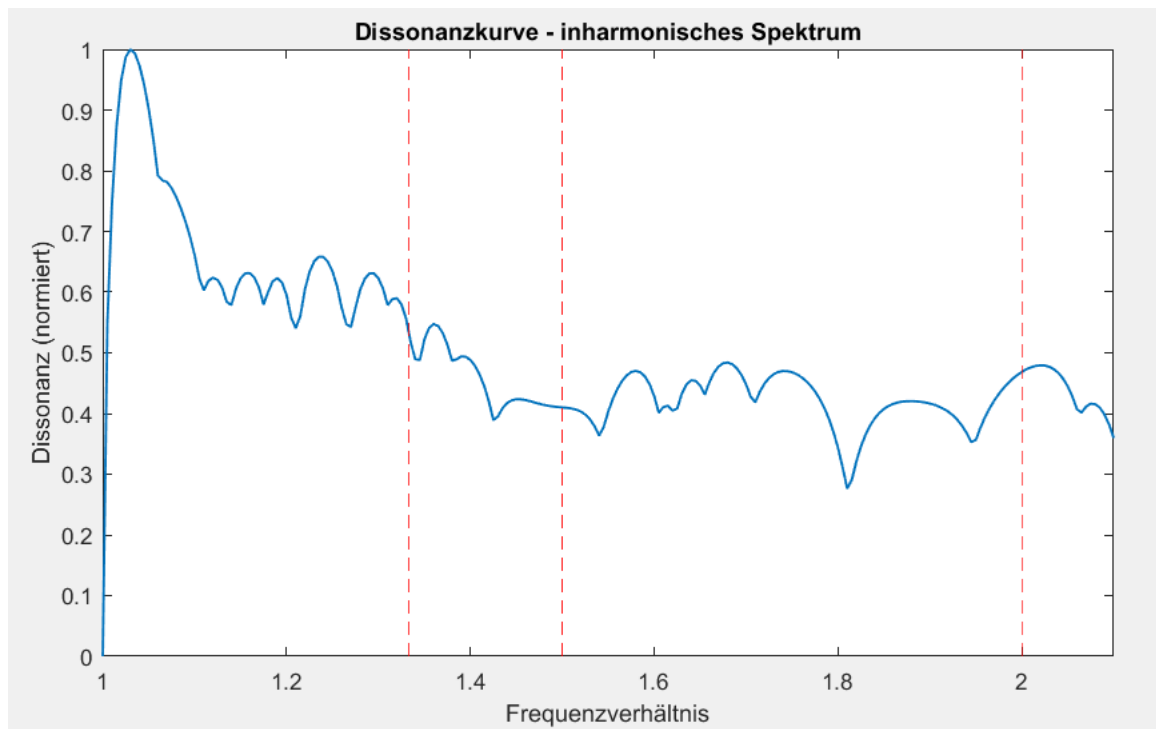


Abbildung 5 – Dissonanzkurve eines Spektrums mit 10 inharmonischen Teiltönen gleicher Amplitude. Der Grundton liegt bei 400 Hz. Strichliert eingezeichnet sind die Frequenzverhältnisse 2 (Oktave), $3/2$ (reine Quinte) und $4/3$ (reine Quarte).

2.4 Optimierte Spektren

Im vorherigen Abschnitt wurde erörtert, dass sich für jedes Spektrum eine ideale Stimmung finden lassen könnte. Genauso gut gilt auch der Umkehrschluss: Für jede Tonskala lässt sich ein ideales, für jeden Skalenton ähnliches Teiltönspektrum konstruieren, um die Konsonanz der in der Tonleiter auftretenden Intervalle zu maximieren.

Im Rahmen dieser Arbeit werden nur gleichstufige Stimmungen (engl. Equal Temperaments) behandelt, die die Oktave in eine geometrische Folge gleich großer Stufen unterteilen. Das Frequenzverhältnis r zweier aufeinanderfolgender Skalentöne ist $\sqrt[n]{2}$, wobei n für die Anzahl der Stufen pro Oktave steht. Für $n = 12$ erhält man z.B. $r = \sqrt[12]{2} = 1.05946$ für ein Halbtonintervall. Die Intervalle einer Skala mit beliebigem n sind $\{1, r, r^2, r^3, \dots, r^n\}$. In Abb. 6 sind die Intervalle 5- bis 31-stufiger Stimmungen in Cent abgebildet. Außerdem wurden die reinen Frequenzverhältnisse Quint, Quart und große Terz eingezeichnet.

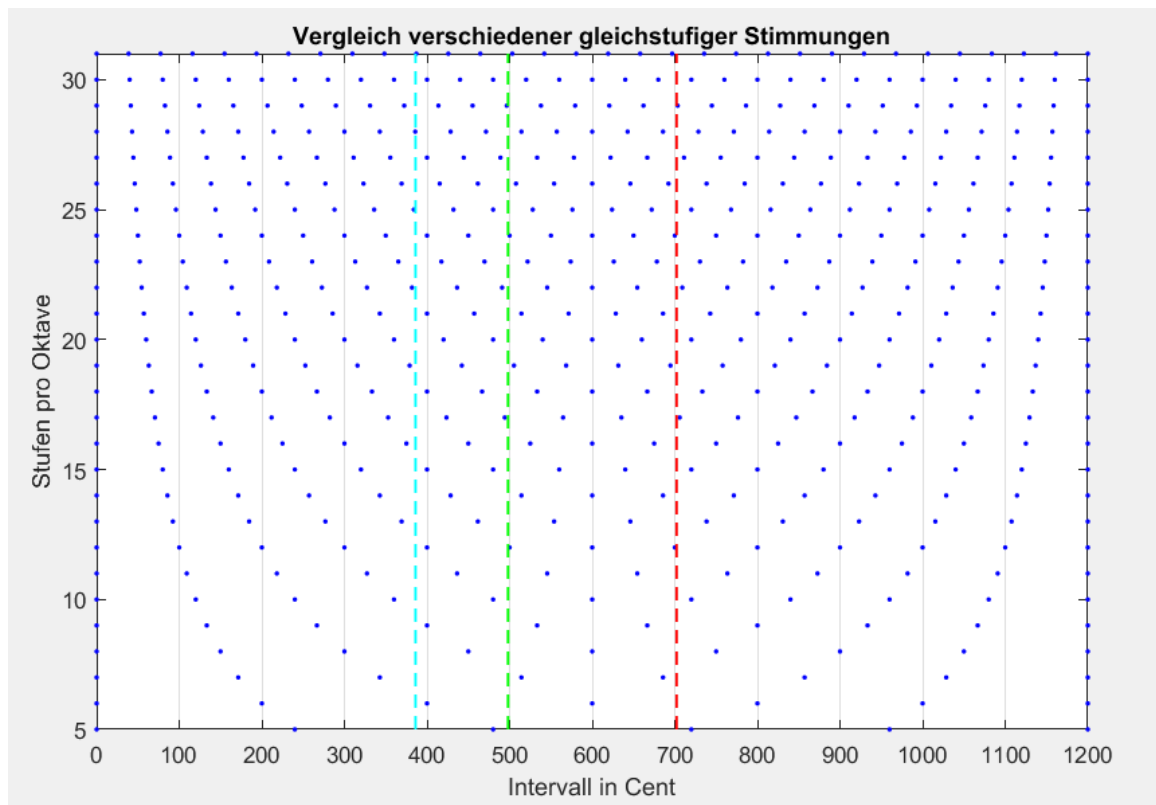


Abbildung 6 – Gleichstufige Stimmungen mit 5 bis 31 Tönen pro Oktave. Die farbigen vertikalen Linien repräsentieren die reine Quint (rot), reine Quart (grün) und reine große Terz (cyan).

Das Ziel ist nun ein Spektrum zu finden, dessen Dissonanzkurve Minima an den Stellen der Tonskala hat. Dies gelingt durch die Anpassung der Teiltonverhältnisse im Spektrum an die Intervalle der verwendeten Skala, d.h. sie müssen in der Menge $\{r^k \mid k \in \mathbb{N}\}$ enthalten sein. Da der Parameter k frei wählbar ist, gibt es nicht nur eine Lösung dieses Problems, sondern unendlich viele [Set05, S.245]. Als eine Möglichkeit ein gutes Spektrum zu finden, kann man von einer harmonischen Teiltonreihe mit den Verhältnissen $\{1, 2, 3, \dots, n\}$ ausgehen und jedes Intervall zur nächstgelegenen ganzzahligen Potenz von r quantisieren.

Beispiel 1: Für eine gleichstufige Stimmung mit 7 Tönen pro Oktave (7-TET) wird ein geeigneter Klang mit 6 Teiltönen gesucht.

Lösung:

$$\{1, r^7, r^{11}, r^{14}, r^{16}, r^{18}\} = \{1, 2, 2.97, 4, 4.88, 5.94\}$$

In Abb. 7 sind Darstellungen der in der Benutzeroberfläche des Instruments integrierten Dissonanzkurven abgebildet, einmal für ein harmonisches und einmal für ein für 7-TET optimiertes Spektrum. Die Teiltonamplituden sind jeweils $\{1, 1/2, 1/3, 1/4, 1/5, 1/6\}$, wie sie auch in der Sägezahnschwingung auftreten. Abb. 7(b) zeigt die Verschiebung der

Dissonanzminima hin zu den Stufen der Skala.

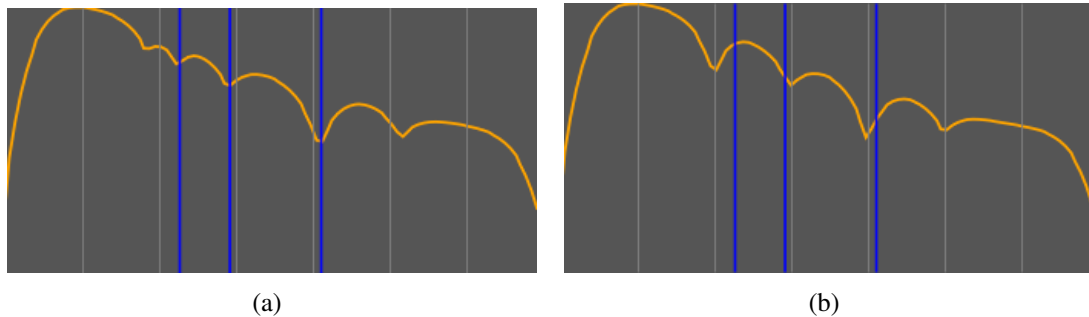


Abbildung 7 – Dissonanzkurven eines harmonischen Spektrums (a) und eines für 7-TET optimierten Spektrums (b). Die blauen vertikalen Linien repräsentieren die reinen Intervalle Quinte, Quarte und große Terz. Die 7 Stufen der Tonleiter sind in grau eingezeichnet. Die Teiltonamplituden sind jeweils $\{1, 1/2, 1/3, 1/4, 1/5, 1/6\}$.

Beispiel 2: Für eine gleichstufige Stimmung mit 12 Tönen pro Oktave (12-TET) wird ein geeigneter Klang mit 6 Teiltönen gesucht.

Lösung:

$$\{1, r^{12}, r^{19}, r^{24}, r^{28}, r^{31}\} = \{1, 2, 2.997, 4, 5.04, 5.99\}$$

Im Falle dieser 12-TET-Skala fallen die Abweichungen zu jenen eines harmonischen Spektrums offenbar gering aus. Abb. 8 rechts zeigt diese Verbesserung auch im treffsicheren Zusammenfallen der Dissonanzminima mit den Skalenstufen (grau vertikal). Theoretisch sollte der obige Klang zu konsonanteren Akkorden in dieser altbewährten Stimmung führen. Ob das mit der Wahrnehmung von Testpersonen übereinstimmt, wurde in einem Hörversuch (Abschnitt 4) überprüft.

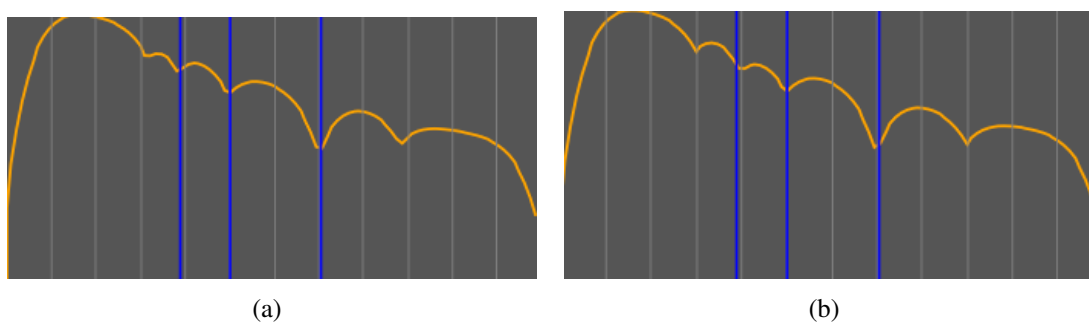


Abbildung 8 – Dissonanzkurven eines harmonischen Spektrums (a) und eines für 12-TET optimierten Spektrums (b). Die blauen vertikalen Linien markieren die reinen Intervalle Quinte, Quarte und große Terz. Die 12 Stufen der Tonleiter sind als graue Linien eingezeichnet. Die Teiltonamplituden sind jeweils $\{1, 1/2, 1/3, 1/4, 1/5, 1/6\}$.

3 Implementierung

3.1 Plattform und Framework

Als Plattform sollte ursprünglich ein iPad dienen, daher wurden die ersten Tests mit Pure Data (Pd) und MobMuPlat [III16] durchgeführt. MobMuPlat erlaubt es, Pd-Patches auf mobilen Geräten auszuführen und mithilfe einfacher GUIs zu steuern. Diese Form der Implementierung ist zwar einfach, aber aktuell nicht effizient genug für die Echtzeitfähigkeit. Den Hauptgrund vermute ich in der Ineffizienz der verschachtelten Schleife in Pd, die zur Berechnung der Teiltonkombinationen benötigt wird (siehe Abb. 9).

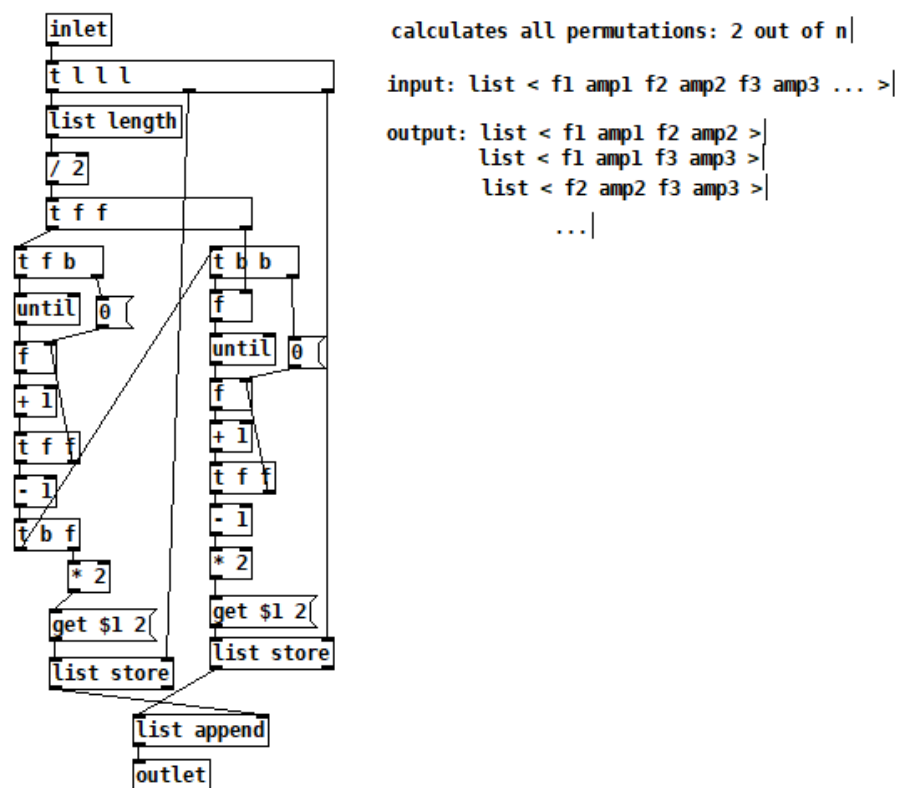


Abbildung 9 – Berechnung aller möglicher Teiltonpaare aus einer Liste mit allen Teiltönen in Pure Data.

Um die Effizienz zu steigern wurde auf C++ in Kombination mit JUCE [Rob13] gewechselt. JUCE ist ein Framework für C++ zur Entwicklung von plattformunabhängiger Software. Der damit erstellte Quellcode kann relativ einfach für verschiedene Betriebssysteme wie Windows, macOS, Linux, iOS und Android kompiliert werden. Aufgrund der zahlreichen Klassen für GUIs und Audiosignalverarbeitung ist JUCE gut zur Programmierung von Audioplugins und -apps geeignet. Während der Entwicklung des hier vorgestellten Instruments habe ich mit Windows 10 und einem externen Multi-Touch-Bildschirm gear-

beitet. Der entstandene Sourcecode¹ sollte durch die JUCE-Umgebung für all die oben genannten Systeme kompilierbar sein. Voraussetzung zur Nutzung des Instruments ist somit lediglich ein Touchscreen.

3.2 Implementierung der Dissonanzkurven

In Abschnitt 2.3 wurde bereits ein Algorithmus zur Berechnung der Dissonanzkurven vorgestellt. Nun wird die Implementierung in C++ und die Erstellung einer Graustufenkala zur Darstellung dieser Kurven behandelt. Die Berechnung dieser Graustufenanzeige findet in der Klasse `BackgroundVisualisation` statt. Deren zwei Methoden `update()` und `dissmeasure()` werden im Folgenden näher betrachtet.

update():

Diese Funktion wird alle 50 ms aufgerufen, d.h. die Anzeige wird mit 20 Hz aktualisiert. In der unten abgebildeten verschachtelten Schleife werden alle Stufen der Skala und ihre Teiltöne ausgewertet, um dazu einen Dissonanzwert zu berechnen und in `dissvector[i]` zu speichern: Der Vektor `newPartials[]` enthält die Teiltonfrequenzen für die neue (virtuelle) Note, welche zu den Teiltönen der bereits liegenden Noten hinzugefügt wird. Die Funktion `dissmeasure()` wird für jede „Taste auf der Klaviatur“ aufgerufen und gibt jeweils einen Dissonanzwert zurück.

```
for (float i = 0; i < numberOfNotes; i++) {
    for (int j = 0; j < numberOfPartials; j++) {
        newPartials[j] =
            root * partialRatios[j] * std::pow(2, i / notesPerOctave);
    }
    std::copy(newPartials.begin(), newPartials.end(),
        allPartials.begin() + (numberOfIntervals * numberOfPartials));
    dissvector[i] = dissmeasure(allPartials, allAmplitudes);
}
```

Im nächsten Schritt werden die Einträge des Vektors `dissvector[]` normiert, damit sie im Wertebereich 0 bis 1 liegen und den Alpha-Wert der Graustufenanzeige bestimmen können.

```
float dissvector_min =
    *min_element(dissvector.begin(), dissvector.begin() + numberOfNotes);
for (int i = 0; i < numberOfNotes; i++)
    dissvector[i] = dissvector[i] - dissvector_min;

float dissvector_max =
    *max_element(dissvector.begin(), dissvector.begin() + numberOfNotes);
for (int i = 0; i < numberOfNotes; i++)
    dissvector[i] = dissvector[i] / dissvector_max;
```

dissmeasure():

Hier erfolgt die eigentliche Berechnung des Dissonanzmaßes: Die Argumente der Funkti-

1. <https://git.iem.at/s1331075/multi-touch-instrument>

on sind zwei Vektoren deren Einträge alle Teiltonfrequenzen bzw. alle zugehörigen Teiltonamplituden des Akkords sind. Zurückgegeben wird der Dissonanzwert des Spektrums. Nach der Initialisierung der Konstanten werden die Amplituden in Grobschätzungen der Lautheit umgerechnet.

```
const float x_star = 0.24f;
const float s1 = 0.0207f;
const float s2 = 18.96f;
const float b1 = 3.51f;
const float b2 = 5.75f;

float d = 0.0f;

const int N = freq.size();
jassert(freq.size() == amp.size());

std::vector<float> loudness(N);
for (int j = 0; j < N; j++) {
    float SPL = 2 * std::log10(
        (amp[j] / juce::MathConstants<float>::sqrt2) / 0.00002);
    loudness[j] = 0.0625 * std::pow(2, SPL);
}
```

Darauf folgt die verschachtelte Hauptschleife, die Dissonanzwerte aller Teiltonpaare mithilfe des in Abschnitt 2.2 beschriebenen Modells berechnet und anschließend summiert. Der resultierende Wert ist der Rückgabeparameter.

```
for (int i = 0; i < N; i++) {
    if (freq[i] >= 0) {
        for (int j = 0; j < N; j++) {
            if (freq[j] >= 0) {
                float l_ij = std::min(loudness[i], loudness[j]);
                float s =
                    x_star / (s1 * std::min(freq[i], freq[j]) + s2);
                float f_dif = std::abs(freq[i] - freq[j]);
                d += l_ij * (std::exp(-b1 * s * f_dif) -
                    std::exp(-b2 * s * f_dif));
            }
        }
    }
}
return d;
```

3.3 Benutzeroberfläche und Features der App

In Abb. 10 ist die GUI des Instruments erkennbar. Auf die einzelnen Teile wird anschließend kurz eingegangen. Ein Demovideo der App ist auch auf Youtube zu finden.²

2. <https://www.youtube.com/watch?v=zsW0xHFn5No>

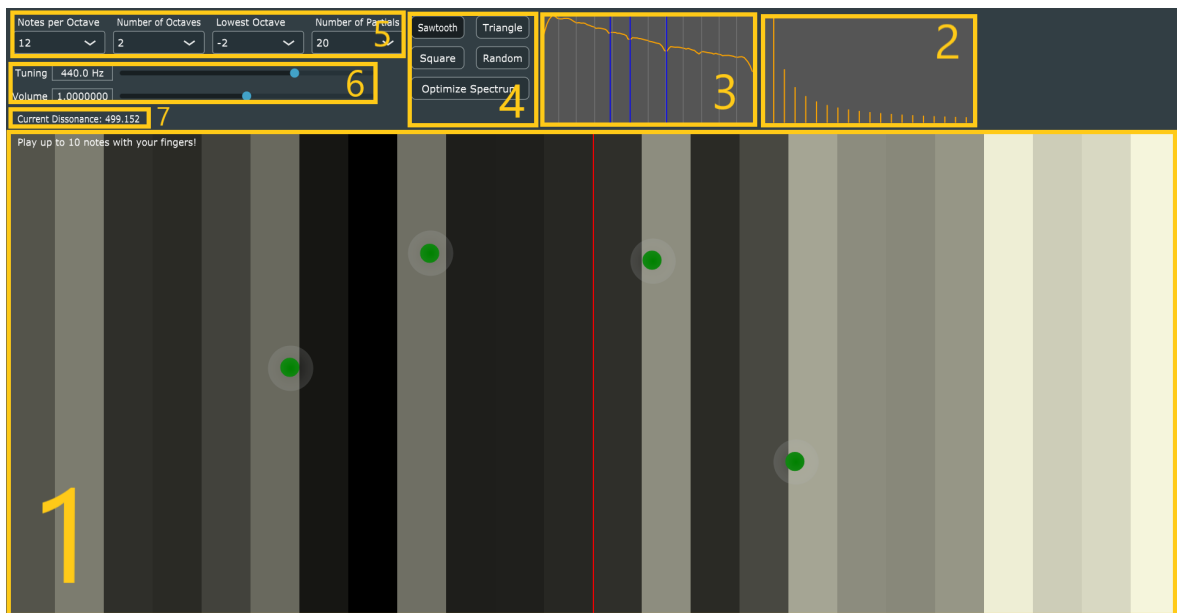


Abbildung 10 – Benutzeroberfläche der App

(1) Spieloberfläche mit Graustufenanzeige:

Die „Klaviatur“ des Instruments. Anhand der momentan liegenden Noten wird für jede Stufe der Skala ein Dissonanzwert in Echtzeit bestimmt und in Form von Grautönen (je heller, umso konsonanter) graphisch dargestellt. Die Darstellung liefert auf diese Weise eine Spielhilfe bei der Wahl weiterer gewollt konsonanter oder dissonanter Akkordtöne. Die einzelnen Oktaven werden durch vertikale rote Linien voneinander abgegrenzt. Abb. 11 zeigt ein Beispiel mit sehr feiner Frequenzauflösung.

(2) Spektrumsanzeige:

Hier werden die Teiltonamplituden und -frequenzverhältnisse angezeigt.

(3) Dissonanzkurve:

In Abhängigkeit des Klangspektrums, der Anzahl an Teiltönen und der Lage der tiefsten Oktave wird eine Dissonanzkurve geplottet. Die Stufen der aktuellen Skala sowie die reinen Intervalle Quinte, Quarte und große Terz werden auch abgebildet.

(4) Spektrumsauswahl:

Folgende Spektren stehen zur Verfügung:

- Sägezahn
- Dreieck
- Rechteck
- Random
- Optimiertes Spektrum

Bei den ersten drei genannten handelt es sich um harmonische Spektren mit unterschied-

lichen Gewichtungen der Teiltöne. Random generiert jedes mal ein zufälliges Spektrum. Die Spektrumoptimierung verschiebt die Teiltonverhältnisse hin zu den Intervallen der aktuellen Skala, wie in Abschnitt 2.4 erläutert. Die Teiltonamplituden entsprechen denen der Sägezahnschwingung.

(5) Wahl der Tonskala und der Anzahl der Teiltöne:

- Tonstufenzahl: 2 bis 120 Stufen pro Oktave
- Anzahl an Oktaven
- Tiefste Oktave: Transposition der Skala in Oktavschritten
- Anzahl an Teiltönen: bis zu 20

Zu beachten ist, dass der Rechenaufwand mit der Teiltonanzahl, der Tonstufenzahl und der Anzahl an Oktaven stark ansteigt. Falls Synthese oder Bedienbarkeit zu stocken beginnen, ist es ratsam, einen dieser Parameter zu senken.

(6) Tuning- und Lautstärkeregler:

Stufenlose Feinstimmung des Instrumentengrundtons und logarithmische Lautstärkeregelung.

(7) Dissonanzwertanzeige:

Anzeige des Dissonanzwertes des momentanen Akkords.

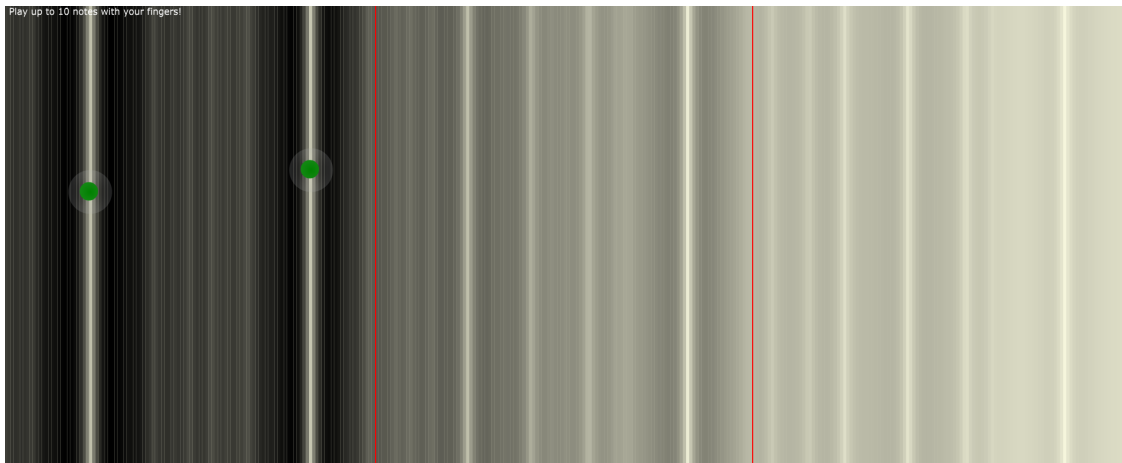


Abbildung 11 – Beispiel der Graustufenanzeige: Drei Oktaven mit jeweils 120 Tonstufen (120-TET). Zwei Noten liegen auf der Skala.

4 Hörversuch

Um die angezeigten Graustufen der Spielhilfe bzw. das Modell aus Abschnitt 2.2 zu evaluieren, wurde ein Hörversuch durchgeführt. Der Versuch soll, neben der Anwendbarkeit des Dissonanzmodells, auch untersuchen, ob die dem Modell entspringenden optimierten

Teiltonspektren den erwarteten konsonanten Vorsprung bringen.

4.1 Versuchsdesign

Testumgebung:

Zur Vereinfachung in der aktuellen COVID-19-Pandemie wurde das Experiment von den Teilnehmer_innen zuhause über Kopfhörer oder Studiomonitore durchgeführt. Als Software diente die MUSHRA-App [Rud19] des IEM. Ein MUSHRA-Hörversuch („Multi Stimulus with Hidden Reference and Anchor“) findet häufig bei der Untersuchung der Qualität von Audiosignalen Anwendung; dabei wird unter die Stimuli ein Referenzsignal und ein bandbegrenzter „Anker“ gemischt [Mue13]. Beides wurde im Rahmen dieses Hörversuchs aber nicht verwendet.

Versuchsteilnehmer_innen:

Bei den 14 Proband_innen handelte es sich entweder um Toningenieur-Studierende oder Musiker_innen mit mehr als 10 Jahren musikalischer Erfahrung.

Aufbau und Ablauf des Experiments:

Der Hörversuch bestand aus drei vergleichenden Multi-Stimulus-Wertungsaufgaben (*Trials*) zu je sechs Akkorden, wobei sowohl die Reihenfolge der *Trials* als auch die der darin enthaltenen Stimuli randomisiert wurden. Die Versuchsteilnehmer_innen wurden angewiesen, die Akkorde auf einer Skala von *sehr dissonant* bis *sehr konsonant* zu bewerten. Abb. 12 zeigt mit Beispielwerten die Benutzeroberfläche der zweiten Vergleichsaufgabe (*Trial 2*). Im Durchschnitt haben die Proband_innen ca. 15 Minuten für die gestellten Aufgaben benötigt.

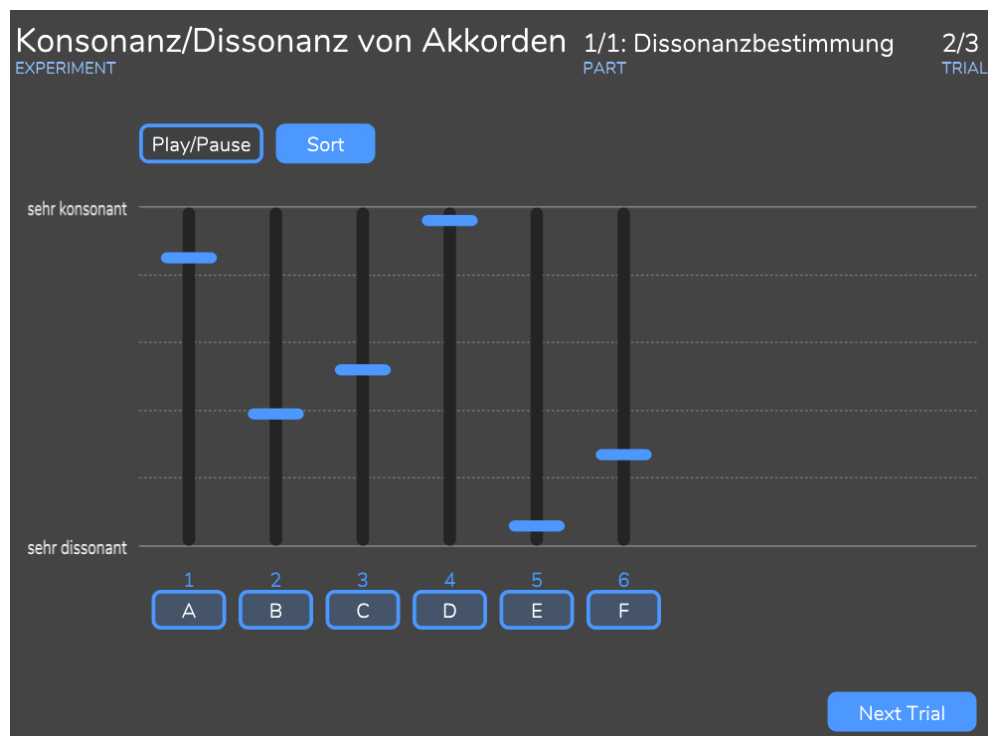


Abbildung 12 – Benutzeroberfläche der MUSHRA-Hörversuchssapp [Rud19]

Stimuli:

Verwendet wurden Akkorde bestehend aus drei bis vier Einzeltönen. Für jeden *Trial* wurde ein anderes Stimmungssystem benutzt (12-TET, 10-TET, 19-TET). In den folgenden drei Tabellen findet man den Aufbau der Akkorde. Die Spalte *Akkordtöne* enthält die Stufen der Tonskala, die gleichzeitig erklingen. Man beachte, dass sich die einzelnen Stimuli pro *Trial* jeweils nur durch einen Ton unterscheiden.

In allen Fällen waren die Einzeltöne aus 20 Teiltönen zusammengesetzt und der tiefste Ton jedes Akkords hatte eine Frequenz von 220 Hz. Jeweils drei der sechs Akkorde in jedem *Trial* verwendeten eine harmonische Sägezahnschwingung. Bei den anderen drei Stimuli handelte es sich um die gleichen Akkorde, jedoch mit einem für die jeweilige Skala optimiertem Spektrum (siehe Abschnitt 2.4). Die Teiltongewichtungen wurden nicht verändert.

Stimulus	Akkordtöne	Spektrum	Dissonanz
1	1, 6, 9, 13	harmon.	357.5
2	1, 6, 9, 11	harmon.	419.0
3	1, 6, 9, 10	harmon.	448.4
4	1, 6, 9, 13	opt.	285.3
5	1, 6, 9, 11	opt.	326.9
6	1, 6, 9, 10	opt.	355.9

Tabelle 1 – Stimuli für *Trial 1* (12-TET)

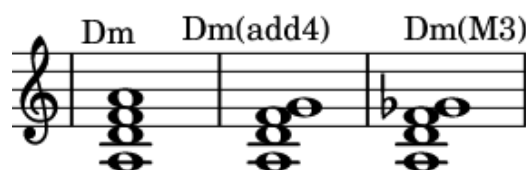
Stimulus	Akkordtöne	Spektrum	Dissonanz
1	1, 8, 5	harmon.	234.7
2	1, 8, 3	harmon.	245.4
3	1, 8, 2	harmon.	249.7
4	1, 8, 5	opt.	151.7
5	1, 8, 3	opt.	166.8
6	1, 8, 2	opt.	181.7

Tabelle 2 – Stimuli für *Trial 2* (10-TET)

Stimulus	Akkordtöne	Spektrum	Dissonanz
1	1, 9, 15, 4	harmon.	438.2
2	1, 9, 15, 5	harmon.	455.8
3	1, 9, 15, 8	harmon.	471.5
4	1, 9, 15, 4	opt.	398.7
5	1, 9, 15, 5	opt.	392.3
6	1, 9, 15, 8	opt.	435.7

Tabelle 3 – Stimuli für *Trial 3* (19-TET)

Die Akkorde aus den *Trials 2* und *3* kann man leider nicht so einfach in ein herkömmliches Notensystem eintragen, für *Trial 1* gelingt dies jedoch ohne weiteres und daher ist auch eine harmonische Analyse möglich. Das Ergebnis kann Abb. 13 entnommen werden. Stimuli 1 bzw. 4 entsprechen eindeutig einem D-Moll-Akkord mit der Quinte im Bass. Die Stimuli 2 bzw. 5 kann man als D-Moll mit zusätzlicher Quarte interpretieren und 3 bzw. 6 könnte man als D-Moll mit hinzugefügter großer Terz auffassen.

Abbildung 13 – Notation der Akkorde aus *Trial 1*

4.2 Ergebnisse

Statistische Auswertung der Daten:

Aufgrund der kleinen Stichprobe von 14 Teilnehmer_innen und wegen des möglichen Einflusses von Skalenrändern auf die Symmetrie, muss angenommen werden, dass ein auf Gaußverteilungen basierender, nur bei größerer Stichprobenanzahl robuster, t-Test nicht sinnvoll ist. Stattdessen wird auf einen nichtparametrischen Test ausgewichen. Der Wilcoxon-Vorzeichen-Rang-Test kann in diesem Fall verwendet werden, um zu überprüfen, ob sich

die Mediane der Testergebnisse zweier Stimuli signifikant voneinander unterscheiden [EGS17, 308]. Die Nullhypothese lautet: $H_0 : M_i = M_j$, d.h. die Mediane der Verteilungen, aus denen die beiden Stichproben stammen, unterscheiden sich nicht.

Für jeden *Trial* werden die Stimuli $i, j \in \{1, 2, \dots, 6\}$ nun paarweise verglichen. Die Matlab-Funktion `signrank()` gibt jeweils einen p-Wert zurück. Pro *Trial* erhält man eine 6x6-Matrix von p-Werten. Anschließend wurde eine Bonferroni-Holm-Korrektur durchgeführt. Wenn $p_{i,j} < 0.05$ (bei einem Signifikanzniveau $\alpha = 5\%$), dann kann die Nullhypothese für dieses Stimulipaar verworfen werden, was bedeutet, dass sich die Ergebnisse signifikant voneinander unterscheiden. Diese Paare wurden in den Abb. 14 bis 16 mit einem Sternchen markiert. Die drei Abbildungen zeigen darüber hinaus die Mediane und Konfidenzintervalle eines Binomialtests. Für jeden der folgenden drei Plots wurde eine lineare Regression der Mediane aller sechs Akkorde durchgeführt. Außerdem gruppieren sich die Antwortdaten nach den verwendeten Spektren, daher wurden zwei getrennte Regressionsgeraden beider Gruppen ermittelt. Bei der linken Dreiergruppe handelt es sich jeweils um Akkorde mit harmonischem Spektrum, die anderen Akkorde sind jene mit optimiertem Spektrum.

Um den Gruppenunterschied der wahrgenommenen Dissonanz zwischen harmonischen und optimierten Spektren zu untersuchen, wurden die Hörversuchsdaten innerhalb der Dreiergruppen zusammengefasst ausgewertet. Wieder wurde ein paarweiser Wilcoxon-Vorzeichen-Rang-Test verwendet, um zu überprüfen, ob sich die Gruppen signifikant unterscheiden. Die Mediane und 95%-Konfidenzintervalle dieser Gruppen sieht man in Abb. 17.

Interpretation der Ergebnisse:

Abb. 14 bis 16 bestätigen das Modell von Sethares innerhalb der Akkordgruppen mit gleichem Spektrum, jedoch versagt das Modell beim Vergleich von Akkorden mit unterschiedlichen Spektren. Innerhalb der Gruppen wurden hohe Bestimmtheitsmaße R^2 erzielt, lineare Regression über den gesamten Datensatz scheint das Modell zu widerlegen. Die Ursache dafür ist, dass die Akkorde mit optimierten (inharmonischen) Spektren von den Versuchspersonen entgegen ihres Entwurfsziels als dissonanter bewertet wurden, als jene mit harmonischem Spektrum. Abb. 17 verdeutlicht diesen Sachverhalt. Ich vermute den Grund für diese Ergebnisse in der musikalischen Prägung der Teilnehmer_innen mit den westlichen, harmonischen Klängen und dadurch auch mit den resultierenden Schwebungen von Akkorden in gleichstufig temperierten Stimmungssystemen. Es ist gut möglich, dass ein Hörversuch mit Proband_innen aus anderen Kulturen (die z.B. Gamelan Indonesiens gewohnt sind) einen völlig anderen Ausgang haben könnte. Um das Dissonanzmodell besser evaluieren und gegebenenfalls formal falsifizieren zu können, wäre ein umfangreicherer Hörversuch mit vielen verschiedenen Akkorden in einigen Skalen nötig. Auch eine größere Anzahl an Versuchspersonen wäre wünschenswert.

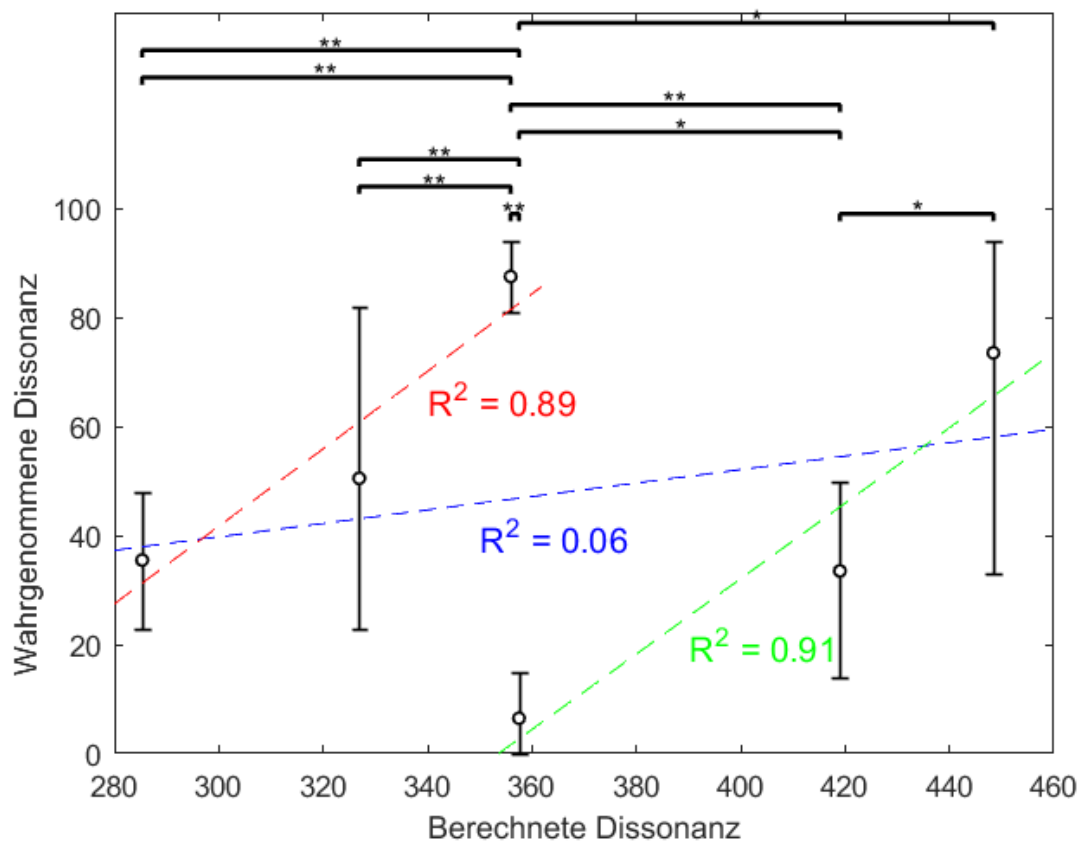


Abbildung 14 – Mediane und 95%-Konfidenzintervalle eines Binomialtests für die Akkorde aus *Trial 1* (12-TET). Die Klammern mit Sternchen markieren signifikante Unterschiede zwischen den Daten (* für $p \leq 5\%$, ** für $p \leq 1\%$). Regressionsgeraden inkl. Bestimmtheitsmaß R^2 wurden für den gesamten Datensatz (blau), sowie für die beiden Dreiergruppen „harmonische Spektren“ (rot) und „optimierte Spektren“ (grün) ermittelt.

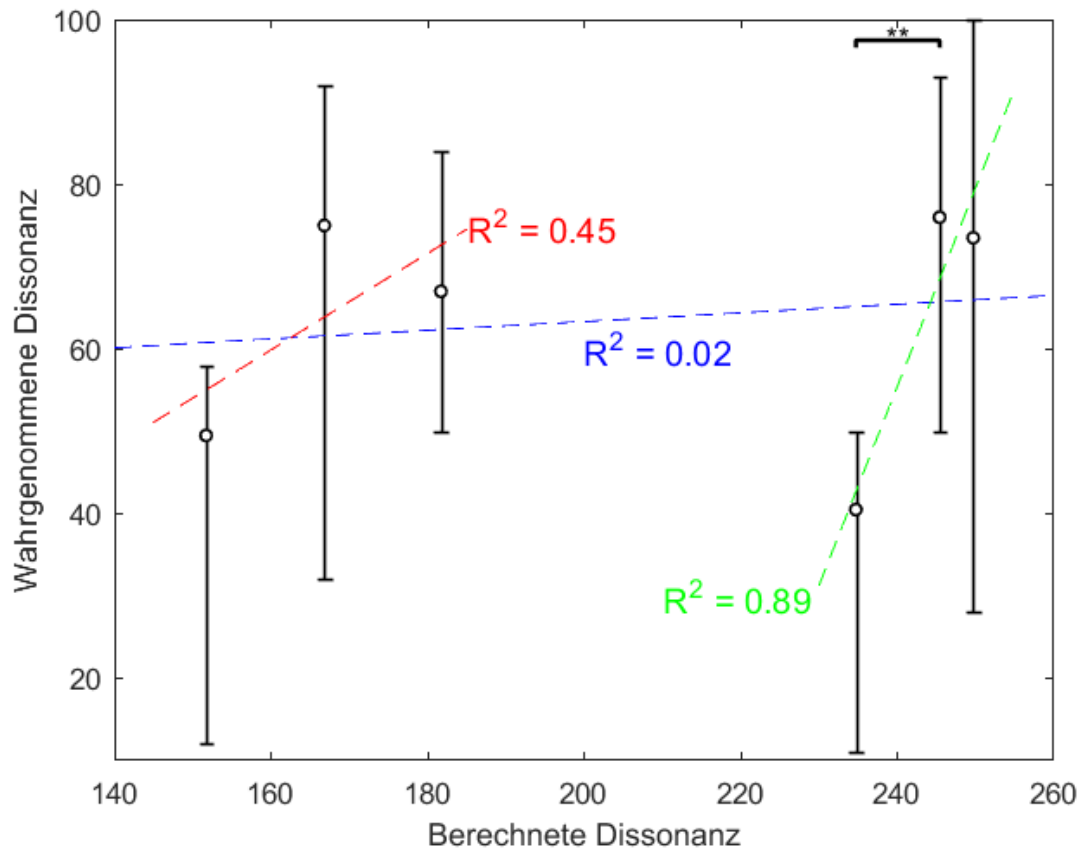


Abbildung 15 – Mediane und 95%-Konfidenzintervalle eines Binomialtests für die Akkorde aus *Trial 2* (10-TET). Die Klammern mit Sternchen markieren signifikante Unterschiede zwischen den Daten (* für $p \leq 5\%$, ** für $p \leq 1\%$). Regressionsgeraden inkl. Bestimmtheitsmaß R^2 wurden für den gesamten Datensatz (blau), sowie für die beiden Dreiergruppen „harmonische Spektren“ (rot) und „optimierte Spektren“ (grün) ermittelt.

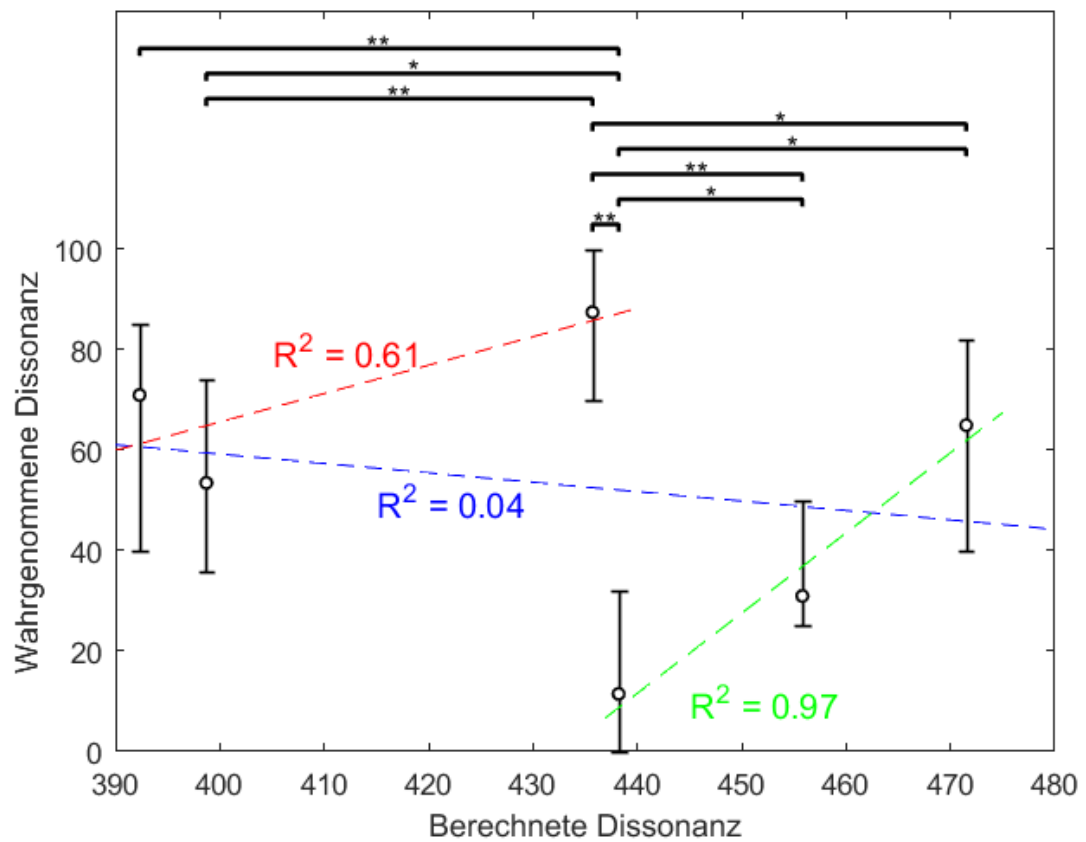


Abbildung 16 – Mediane und 95%-Konfidenzintervalle eines Binomialtests für die Akkorde aus *Trial 3* (19-TET). Die Klammern mit Sternchen markieren signifikante Unterschiede zwischen den Daten (* für $p \leq 5\%$, ** für $p \leq 1\%$). Regressionsgeraden inkl. Bestimmtheitsmaß R^2 wurden für den gesamten Datensatz (blau), sowie für die beiden Dreiergruppen „harmonische Spektren“ (rot) und „optimierte Spektren“ (grün) ermittelt.

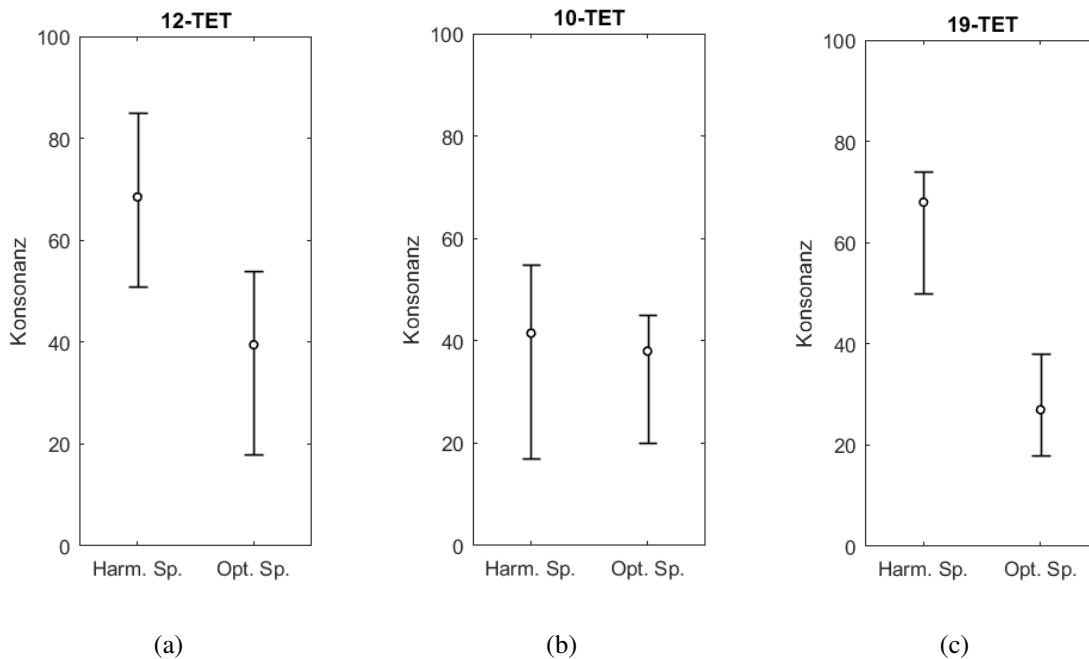


Abbildung 17 – Vergleich der zusammengefassten harmonischen bzw. optimierten Spektren jedes *Trials*. Die Grafik zeigt die Mediane und 95%-Konfidenzintervalle eines Binomialtests. Für die *Trials* 1 und 3 (12-TET, 19-TET) besteht ein statistisch signifikanter Unterschied zwischen harmonischen und optimierten Spektren, wie aus einem paarweisen Wilcoxon-Vorzeichen-Rang-Test hervorgeht. Innerhalb von *Trial* 2 (10-TET) gibt es keinen signifikanten Unterschied zwischen den beiden Gruppen.

5 Schlussfolgerung und Ausblick

In meiner Projektarbeit habe ich einen Prototypen eines polyphonen Multi-Touch-Instruments mit einer Spielhilfe in Form einer Dissonanzanzeige entworfen und umgesetzt. Der Sourcecode sowie ein Build für Windows sind auf Gitlab (<https://git.iem.at/s1331075/multi-touch-instrument>) verfügbar. Zur Kompilierung wird lediglich eine IDE und das JUCE-Framework benötigt. Als Werkzeug zur Dissonanzanzeige wurde das Modell von Sethares beschrieben und angewendet. Um dieses Modell zu evaluieren, wurde ein Hörversuch durchgeführt. Dieser Versuch zeigt auf, dass weitere Arbeiten zur Modellierung polyphoner Konsonanz bzw. Dissonanz nötig wären. Das bisherige Modell „bestraft“ polyphone Schwebungen offenbar zu stark, denn häufig schätzen Versuchspersonen gewohnte harmonische Teiltonspektren mit mehrstimmigen Schwebungen als konsonanter ein, verglichen mit weniger schwebenden inharmonischen, weil auf Konsonanz angepasste Spektren.

Abschließend habe ich einige Verbesserungsvorschläge für den im Rahmen diese Projekts entstandenen Prototypen:

Bei Verwendung einer Touch-Oberfläche ist es naheliegend, dass auch die vertikale Achse der Klaviatur einen Parameter steuern könnte. Vorstellbar wären z.B. verschiedene Fil-

ter oder Effekte. Auch eine ADSR-Hüllkurve wäre für die musikalische Praxis hilfreich. Außerdem könnte man neben den gleichstufigen Skalen beliebige, frei konfigurierbare Stimmungen implementieren. Nützlich wäre zudem etwa eine Anzeige mit Pfeilen, um mögliche Veränderungsrichtungen gegriffener Noten bzgl. höherer Konsonanz oder Dissonanz zu signalisieren. Zu guter Letzt kann natürlich auch das Dissonanzmodell von Sethares optimiert werden, das nur auf dem Rauigkeitseindruck basiert. Ein detaillierteres Modell, wie das von Terhardt [Ter84], berücksichtigt mehrere Komponenten; die Rauigkeit ist nur ein Teil davon.

Literatur

- [EGS17] M. Eid, M. Gollwitzer, and M. Schmitt, Statistik und forschungsmethoden, 2017.
- [Hor20] D. Horvath. (2020) Lumatone. [Online]. Available: <https://www.lumatone.io/>
- [III16] D. Iglesia and I. Intermedia, “The mobility is the message: The development and uses of mobmuplat,” in Pure Data Conference (PdCon16). New York, 2016.
- [Mue13] S. Y. Mueller, “Handbuch für versuchsdesign in der psychoakustik,” Master’s thesis, TU Graz, 11 2013.
- [PL65] R. Plomp and W. J. M. Levelt, “Tonal consonance and critical bandwidth,” The journal of the Acoustical Society of America, vol. 38, no. 4, pp. 548–560, 1965.
- [Rob13] M. Robinson, Getting started with JUCE. Packt Publishing Ltd, 2013.
- [Rud19] D. Rudrich, “MUSHRA Hörversuchapp,” 2019. [Online]. Available: <https://git.iem.at/rudrich/mushra>
- [Set05] W. A. Sethares, Tuning, timbre, spectrum, scale. Springer Science & Business Media, 2005.
- [Ter83] E. Terhardt, “Musikwahrnehmung und elementare hörempfindungen,” Audiologische Akustik-Audiological Acoustics, vol. 22, pp. 52–56, 1983.
- [Ter84] ———, “The concept of musical consonance: A link between music and psychoacoustics,” Music perception, vol. 1, no. 3, pp. 276–295, 1984.
- [Ter00] ———. (2000) Musical consonance. [Online]. Available: <https://www.ei.tum.de/fileadmin/tueifei/mmk/Personen/Terhardt/ter/top/muscons.html>