

# Parameter-Modulation synthetischen Klangs, basierend auf der Idee der Schlüsselbildanimation

# Masterarbeit

# Technische Universität Berlin Fakultät I Fachgebiet Audiokommunikation

#### **Betreuer:**

Prof. Dr. Weinzierl, Leiter Fachgebiet Audiokommunikation, TU Berlin Dipl.-Ing. Athanasios Lykartsis, M. Sc., Wissenschaftlicher Mitarbeiter, TU Berlin Scott McCoid, Ableton AG Berlin

# Eidesstattliche Erklärung

Hiermit erkläre ich an Eides statt gegenüber der Fakultät I der Technischen Universität Berlin, dass die vorliegende, dieser Erklärung angefügte Arbeit selbstständig und nur unter Zuhilfenahme der im Literaturverzeichnis genannten Quellen und Hilfsmittel angefertigt wurde. Alle Stellen der Arbeit, die anderen Werken dem Wortlaut oder dem Sinn nach entnommen wurden, sind kenntlich gemacht. Ich reiche die Arbeit erstmals als Prüfungsleistung ein. Ich versichere, dass diese Arbeit oder wesentliche Teile dieser Arbeit nicht bereits dem Leistungserwerb in einer anderen Lehrveranstaltung zugrunde lagen.

Verfasser:	
Nils Jessen,	

#### **Betreuende Dozenten:**

Prof. Dr. Stefan Weinzierl

Dipl.-Ing. Athanasios Lykartsis

Mit meiner Unterschrift bestätige ich, dass ich über fachübliche Zitierregeln unterrichtet worden bin und diese verstanden habe. Die im betroffenen Fachgebiet üblichen Zitiervorschriften sind eingehalten worden. Eine Überprüfung der Arbeit auf Plagiate mithilfe elektronischer Hilfsmittel darf vorgenommen werden.

Berlin, den Nils Jessen

# Zusammenfassung

Basierend auf der Idee der Schlüsselbild-Animation ist es das Ziel dieser Masterarbeit, alternative Ansätze zur Parameter-Modulation von Synthesizern zu entwickeln und ihren Einsatz im Sound-Design zu evaluieren. Das klassische Konzept der Hüllkurve mit den distinkten Abschnitten Attack, Decay, Sustain und Release soll über eine Modularisierung der Phasen in Funktion und Anzahl flexibler gestaltet werden. Der damit einhergehenden Zunahme in der maximal erreichbaren Modulationskomplexität, soll ein vereinfachendes Bedienkonzept entgegengestellt werden. Dieses setzt neben voller Visualisierung auch auf ein auditives Feedback und soll non-lineares Sound-Design ermöglichen. Evaluiert wurde das Konzept über einen Software-Prototyp mit spezieller Anpassung an eine existierende Controller-Hardware. Die Usability-Tests mit 12 Experten haben ergeben, dass das neue Konzept sehr deutlich als hilfreich und sinnvoll wahrgenommen wurde. Bei normal komplexen Sounds gab es keine signifikanten Nachteile gegenüber dem konventionellen Ansatz, wohingegen sich die Vorteile insbesondere in den Qualitäten zeigten, die mit der Benutzerzufriedenheit assoziiert sind.

# Inhaltsverzeichnis

1	Ein	leitung	7
	1.1	Motivation	7
	1.2	Stand der Technik	8
	1.3	Eigene Vorarbeiten	10
	1.4	Traditionelle Keyframe-Animation	11
	1.5	Ein Wort zur Interpolation	12
	1.6	Das Konzept der Schlüsselbildanimation im Audiobereich	13
	1.7	Flexible Phasen	14
	1.8	Continuous Feedback	14
	1.9	Workflow-Beispiel	15
2	Met	thoden	17
	2.1	Der Prototyp	17
	2.2	Softwarebasis	17
	2.3	Implementierte Funktionen	19
	2.4	Hardwaresteuerung	20
	2.5	GUI und Mapping	21
	2.6	Evaluation des Prototyps	24
	2.7	Target-Finding	25
	2.8	AttrakDiff-Fragebogen	28
	2.9	Allgemeiner Fragebogen	29
	2.10	Test-Gruppe	31
	2.11	Test-Aufbau	32
3	Aus	swertung und Ergebnisse	33
	3.1	Statistischer Vergleich Target-Finding	33
	3.2	Aufbereitung der Daten des Target-Finding	34
	3.3	Bestimmung der Differenzen zwischen reproduzierten und Target-Sound	35
	3.4	Untersuchung der Signifikanz der Unterschiede des Target-Findings	39
	3.5	Kontrolle der statistischen Analyse	40
	3.6	Ergebnisse der Zeitmessung des Target-Finding	41
	3.7	Ergebnisse AttrakDiff	42
	3.8	Usability von Envelopes im Vergleich	44
	3.9	Ergebnisse der offenen Befragung	45

4 D	riskussion	46
4.1	Target Finding	46
4.2	Usability	
4.3	Qualitativer Teil	
4.4	Fazit	
4.5	Ausblick	50
5 L	iteraturverzeichnis	51
6 A	nhang	53
Tabo	ellenverzeichnis	
	e 1: Color-Coding	
	e 2: Aufbau Test-Session	
	e 4: Produzierte Musikgenres der Probanden	
	e 5: Vergleichbare Merkmale	
	e 6: Differenz-Kriterien pro Sound	
	e 7: Gesamtdifferenzen (normalisiert) pro Sound	
	e 8: Ergebnisse Wilcoxon Rank Test	
	e 9: Test-Sessions im Vergleich	
	le 10: Dauer Target-Finding Total	
	e 11: Dauer Target-Finding per Sound	
14001	or it. Duder Turger I maning per Sound	
A 1 1	11.1	
Abb	ildungsverzeichnis	
Abbilo	dung 1: ADSR-Envelope (Wikimedia CC, 2017)	8
Abbilo	dung 2: Yamaha DX7 Envelope (Yamaha, 1983)	9
Abbilo	dung 3: Bouncing Ball (Richardson, 2015)	11
Abbilo	dung 4: Inbetweens (Richardson, 2015)	11
Abbilo	dung 5: Code-Snippet Interpolation	12
Abbilo	dung 6: Frühe Konzeptzeichnung	13
	dung 7: Analogie Amplitude	
Abbilo	dung 8: Analogie Zeitbereich	16
	dung 9: Synister	
	dung 10: Push 2 im Betrieb mit Live (Ableton, 2016), © Ableton Ag	
Abbilo	dung 11: Frühe Version des Prototyps mit Maussteuerung	21
Abbilo	dung 12: Keyframe Envelope auf Push 2 Display	22
	dung 13: MIDI-Mapping des Push 2, © Ableton Ag (Ableton, 2017)	
	dung 14: Modulationsverlauf Sound 1	
	dung 15: Modulationsverlauf Sound 2	
	dung 16: Modulationsverlauf Sound 3	
	dung 17: AttrakDiff Model © User Interface Design GmbH	
Abbilo	dung 18: AttrakDiff Fragebogen © User Interface Design GmbH	29

Abbildung 19: Dominierende Qualitäten der Envelopes	30
Abbildung 20: Aufbau Test-Studio	32
Abbildung 21: Launch Control © Novation (Novation, 2017)	33
Abbildung 22: Spectral Flux Sound 3, ungefiltert	34
Abbildung 23: Detailansicht Session 2, Sound 2 / Target-Sound	36
Abbildung 24: Frequenzverlauf Sound 1	38
Abbildung 25: Frequenzverlauf Sound 2	38
Abbildung 26: Filterverlauf Sound 3	39
Abbildung 27: AttrakDiff – Mittelwerte	42
Abbildung 28: AttrakDiff - Wortpaare	
Abbildung 29: Envelope-Usability im Vergleich	
• • •	

# Abkürzungsverzeichnis

ADSR - Attack Decay Sustain Release

DAW – Digital Audio Workstation

GUI – Graphical User Interface

KF - Keyframe

OSC – Open Sound Control

QA – Quality Assurance

RMS - Root Mean Square

TP/LP – Tiefpass / Lowpass (Filter)

UX – User Experience

VCA – Voltage Controlled Amplifier

VFX – Visual Effects

VST – Virtual Studio Instrument

WYSIWYG – What you see is what you get

# Danksagung

Ganz besonders danke ich meinen Betreuern Thanassis Lykartsis, Scott McCoid und Jack Schaedler; Marc Halbrügge für seine Tipps zur Usability-Untersuchung, der Firma Ableton im allgemeinen und insbesondere dem Hardware-Team, sowie allen Teilnehmern meiner Studie. Auch danke ich meinen Eltern für Ihre Unterstützung, und für die Korrekturen Anne Becker und Christoph Barkowsky.

# 1 Einleitung

#### 1.1 Motivation

Die Veränderung eines Tons über die Zeit ist neben den enthaltenen Obertönen eine grundlegende Größe der Klangfarbe. Während die spektrale Komponente, also der Frequenzbereich, stark von der Synthese-Methode abhängt, kommt für die Modulation im Zeitbereich (Weinzierl, 2008) in der Regel eine Hüllkurve (Envelope) oder ein Niederfrequenzschwingkreis ("Low Frequency Oscillator" - LFO, Loy und Chowning, 2006) zum Einsatz.

Diese mehrphasigen ADSR-Envelopes (Attack, Decay, Sustain, Release, Loy und Chowning, 2006) werden hierbei konventionell durch Tastenanschläge der Klaviatur gesteuert. So wird auf ganz elementarer Ebene auch die Lautstärke eines Klangs über einen Voltage-Controlled-Amplifier (VCA, Allen, 1981), gesteuert und über einen Envelope modelliert. Hierbei ist insbesondere die Einschwingphase charakteristisch für das Timbre (z.B. die schnelle Tonhöhenänderung beim Anschlagen des Fells einer Trommel) und wird variiert mit der Form der Anregung (gezupfte bzw. gestrichene Saite). Dieses Wissen wird in der synthetischen Klangerzeugung beim Sound-Design aktiv eingesetzt.

Komplexe Modulationen sind durch die Kombination mehrerer Modulatoren (z.B. ein vom LFO gesteuerter Envelope) oder externe Kontrolldaten wie durch die Automation einer Digital Audio Workstation (DAW) bei einer Vielzahl von Synthesizern möglich. Der direkte und intuitive Zugang des "WYSIWYG" – Prinzips auf die Modulation ist damit allerdings nicht mehr gegeben.

Um dieses Problem zu lösen, soll ein Workflow aus der computergesteuerten Grafik-Animation (Lasseter, 1987) erprobt werden. Mit dem Konzept der Schlüsselbild-Animation (keyframe animation bzw. keyframing, Brotman und Netravali, 1988) können komplexe zeitliche Verläufe auf intuitive und schnelle Weise erstellt werden. Durch automatische Interpolation mehrerer Positionen zu definierten Zeitpunkten (Keyframes), sollen Parameter der Synthesizer-Engine ähnlich dem "morphing" (Bencina, 2005; Goudeseune, 2002; Pendharkar et al., 2006) mit fließenden Übergängen versehen werden.

Die Motivation hierbei ist, die seit über 50 Jahren bestehende und eher abstrakt-mathematische Herangehensweise in der Bedienung von Modulatoren neu zu überdenken. Gesucht wird ein Konzept, dass schnell, selbsterklärend und in der Bedienung für die Zielgruppe natürlicher ist. Dabei soll weniger der Typ Ingenieur als der Sound-Designer im Fokus stehen. Im besten Fall sollen dabei komplexere Modulationen ermöglicht werden sowie etwaige Einschränkungen des Standardansatzes aufgezeigt und überwunden werden. Eine empirische Untersuchung anhand eines Prototyps soll der Frage nachgehen, ob der gewählte Ansatz auf Nutzerseite zu den gewünschten Verbesserungen führt.

#### 1.2 Stand der Technik

Entgegen den Entwicklungen der Modulationsmöglichkeiten im Frequenzbereich, welcher mit der Syntheseart einhergeht, hat sich im Zeitbereich über lange Zeit wenig verändert. LFOs und ADSR-Envelopes sind in nahezu jedem Synthesizer vertreten und ein Quasi-Standard. Ob es einen alternativen und wohlmöglich besseren Weg der Steuerung gibt, der auf die Entwicklungen in Usability und User Experience Design der jüngeren Zeit aufbaut, soll im Rahmen dieser Arbeit geklärt werden.

Typischerweise wird der Envelope genutzt, um den Lautstärkeverlauf eines Tons zu steuern. Je nach Synthesizer kann er allerdings auch andere Parameter wie Tonhöhe oder Filter Cutoff Frequenz steuern. Dies ist in der Regel flexibel einstellbar. Die Funktionsweise des ADSR-Envelopes (siehe Abbildung 1) ist simpel: Durch den Tastenanschlag wird die Attack-Phase ausgelöst. Wird die Taste nicht vor dem Ende der Attack-Phase losgelassen, wird die Decay-Phase ausgelöst, welche auch eine feste Länge besitzt. Wird die Note weiter gehalten, setzt das Sustain ein. Auf dieser Amplitude (ohne Länge) verweilt der Ton so lange, wie die Note gehalten wird. Durch das Loslassen wird die Ausklingphase (Release) gestartet, welche wiederum eine feste Länge hat. In manchen Fällen wurde dieses Konzept in der Bedienung weiter vereinfacht. Decay und Release werden dann in der Regel über ein Steuerelement geregelt (wie auch beim wegweisenden Minimoog) (Moog, 1964).

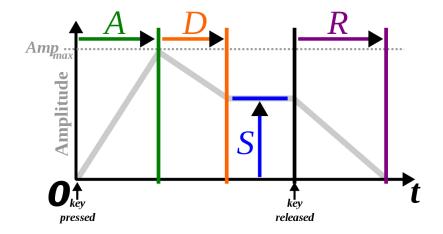


Abbildung 1: ADSR-Envelope (Wikimedia CC, 2017)

Dieses Grundkonzept der Bedienung hat sich seit dem Aufkommen der modernen Synthesizer, wie man sie heute kennt, also seit den 1960er Jahren und den Entwicklungen von Bob Moog nicht mehr grundlegend geändert.

Eine Ausnahme bilden die für Ihre schwere Handhabe bekannten "Time-Level Envelopes" des Yamaha DX7. Diese verfügen im Original über 5 Phasen, welche über "Rate" in der Länge und "Level" in der Amplitude variiert werden können (siehe Abbildung 2). Auf dieses Envelope-Konzept bauen auch einige wenige moderne Software Synthesizer wie beispielsweise Native Instruments's "Absynth" oder "FM8" auf (Native Instruments, 2017). Die maximale Menge der Phasen ist hier allerdings teilweise drastisch erhöht worden (bis zu 64 Phasen bei "Absynth"). Die Erzeugung des Envelopes erfolgt hier wie beim klassischen Manipulieren von Automationsdaten in einer DAW per Maus.

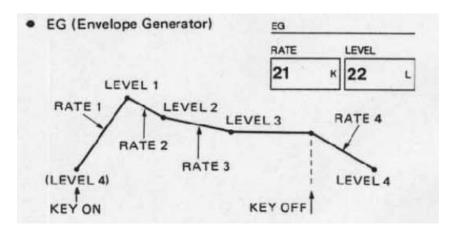


Abbildung 2: Yamaha DX7 Envelope (Yamaha, 1983)

Da es, wie bereits erwähnt, in den letzten 50 Jahren kaum eine wissenschaftlich erfasste Weiterentwicklung im Bereich der für Klangmodulation eingesetzten Hüllkurven gab, wurden artverwandte Forschungsbereiche betrachtet. In der Klangsynthese (Serra und others, 1997) und -analyse des Zeitbereichs werden beispielsweise Techniken des MIR (Music Information Retrieval) wie die Onset-Detection (Lerch, 2012) für Transientenbestimmung eingesetzt. Zur Modulations-Steuerung hingegen finden sich keine neuen Ansätze, die auf dieses Wissen aufbauen. Der momentan wohl aktivste Forschungsbereich mit hohem Deckungsgrad ist jener der elektronischen Musik-Performance (Arner, 2013; McPherson, 2015; Olowe et al., 2016). Sogenannte expressive (oder Novel-)Controller, wie das Seaboard (Lamb und Robertson, 2011), ermöglichen es, mehrere Parameter parallel in Echtzeit zu steuern, um dem Spiel so mehr Ausdruck zu verleihen. Da es abgesehen vom Kernunterschied, der Eingabemethode (vom Mensch in Echtzeit zur Performance versus Vorprogrammiert zum Sound-Design), viele ähnliche Fragestellungen gibt, sollten sich die Erkenntnisse in den folgenden Bereichen übertragen lassen.

#### **Interpolation:**

Die Interpolation von Parametern, die sich insbesondere bei Dimensionsreduktion nicht einfach intuitiv gestalten lässt (Bencina, 2005; Goudeseune, 2002; Pendharkar et al., 2006), ist nur eine der Herausforderungen. Das Ziel ist hier, eine möglichst hohe Expressivität zu erreichen, ohne jedoch die Kontrolle und somit die Reproduzierbarkeit negativ zu beeinflussen.

#### **Mapping:**

Das Mapping von Parametern (Arfib et al., 2002; Brandtsegg et al., 2011; Hunt et al., 2003; Hunt und Wanderley, 2002) ist eine weitere wichtige Fragestellung, denn auch beim Sound-und Interface-Design sind in Anbetracht von Usability und User Experience (UX) leicht erlernbare und schnelle Bedienkonzepte gefragt.

#### **Interfaces:**

Ein artverwandtes Gebiet ist jenes der Musik-Apps (insbesondere für Apple iOS), deren grundlegendes Eingabekonzept auf "Touch" (von Falkenstein, 2011; Zappi et al., 2010) und Gestensteuerung (Cadoz und Wanderley, 2000) setzt. Während der Vorarbeiten wurden diese in der Regel proprietären und wissenschaftlich nicht erfassten Produkte untersucht. Ausnahmen, wie die Untersuchung von gezeichneten Parameterverläufen (Zadel und Scavone,

2006) oder multidimensionaler Parametersteuerung per Touch-Eingabe (Tubb et al., 2012) sind aber durchaus relevant.

Überschneidungen gibt es des Weiteren auf den Gebieten GUI-Entwicklung (Gómez et al., 2007), bei denen experimentelle Ansätze mit bestehenden Paradigmen brechen. Ein breiter, produktiver Einsatz dieser neuen Techniken lässt sich hier in Ansätzen schon bei Apps für mobile Geräte wie Tablets und Smartphones erkennen. Allgemeine Erkenntnisse aus der Datenvisualisierung helfen bei der Findung intuitiverer Modelle (Momeni und Wessel, 2003; Van Nort et al., 2004) und bieten so neue Ansätze, wie hier über das Mapping auf simple geometrische Formen. Eine Idee, welche aufgrund ihrer Einfachheit auch didaktisch genutzt werden kann (Hein, 2013) und auch bis zum physikalischen Teil der Human-Computer-Interaction (HCI) weiter gedacht wird (Potidis und Spyrou, 2014). Und obwohl solche Eingabemethoden vielversprechend wirken, ist der Anwendungszweck oft eingeschränkt und die Verfügbarkeit begrenzt. Die konventionellen Eingabemethoden (Gelineck und Serafin, 2009) werden fürs erste also ihre Relevanz behalten, was aber nicht bedeutet, dass neue Paradigmen (Faria, 2011) nicht auf sie angewendet werden können.

## 1.3 Eigene Vorarbeiten

Im Vorfeld wurden zahlreiche alternative Bedienkonzepte für das Sequencing (von Falkenstein, 2011) und die Modulation betrachtet. Insbesondere im Bereich der Musik-Apps (Apple iOS) und Browser-basierten Anwendungen zeigten sich hier interessante alternative Ansätze. Da sich die Wissenschaftsgemeinde, wie die Literaturrecherche gezeigt hat, sehr auf die Untersuchung von Performance-orientierte Bedienkonzepte konzentriert und somit fast alle Neuerungen unter gewinnorientierten Aspekten entwickelt wurden, ist der damit einhergehende schlechte Verfügbarkeitsgrad von Untersuchungen alternativer Modulationsmöglichkeiten entsprechend begrenzt.

Die einzigen Referenzen sind somit oft die Handbücher der Software selbst. Es hat sich allerdings gezeigt, dass es teilweise starke Überschneidungen mit dem Bereich der modernen elektronischen Musik-Performance gibt. Insbesondere Untersuchungen zur Expressivität (Dobrian und Koppelman, 2006) von elektronischen Instrumenten sind hier relevant, da sie ein hohes Maß an Expertise beim Parameter-Mapping (Hunt et al., 2003) verlangen. Die multidimensionale Steuerung von Parametern in Echtzeit verlangt ein Höchstmaß an Usability und Reproduzierbarkeit der Ergebnisse. Auch Fragen zum "morphing" (Hunt et al., 2003) und der Dimensionsreduktion (Goudeseune, 2002) sind für das Thema Sound-Design von großer Bedeutung.

Die Vorarbeiten, Literaturrecherche und Untersuchung des State-of-the-Art hat keine mit der Idee des Continuous Feedback (siehe Kapitel 1.8) vergleichbare Technik zum Ergebnis gehabt.

# 1.4 Traditionelle Keyframe-Animation

Keyframing, Keyframe-Animation oder Schlüsselbild-Animation sind Synonyme einer Technik zur einfachen und schnellen Erstellung von Bewegungsabläufen, die ihren Ursprung in der analogen Zeichenfilmproduktion hat. Ursprünglich für die Arbeitsteilung angedacht, erstellte hier ein Hauptzeichner die Schlüsselbilder, welche die markanten Phasen eines Bewegungsablaufs markieren. Im klassischen Beispiel der "bouncing ball animation" (White, 2006) entspräche dies der Position des Balls am Scheitelpunkt und am Boden (siehe Abbildung 3).

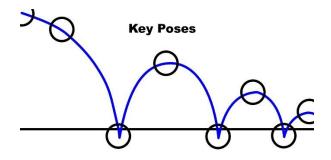


Abbildung 3: Bouncing Ball (Richardson, 2015)

Die Erstellung der Zwischenbilder bzw. Interframes oder Inbetweens konnte nun von anderen Zeichnern übernommen werden, wobei sich die Menge der Bilder neben der Geschwindigkeit der Bewegung (siehe Abbildung 4), auch an der Bildrate des Films (z.B. 24 Bilder/Sekunde) und der gewünschten Geschmeidigkeit der Animation richtet. In der analogen Welt der Animation ist die Erstellung von Zwischenbildern mit erheblichem Aufwand bzw. Kosten verbunden, weshalb in der Regel zu "doubles" (ein Bild steht für zwei Frames) gegriffen wurde. Um die Erstellung von Zeichentrickfilmen effizienter zu machen, ohne jedoch die Kontrolle an die oft nach Korea ausgelagerten Zwischenbild-Zeichner zu verlieren, war die Schlüsselbild-Animation also ein gut geeignetes Mittel.

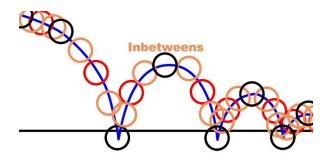


Abbildung 4: Inbetweens (Richardson, 2015)

Mit Einzug der Digitaltechnik, durch welche die Berechnung der Interframes durch den Computer bzw. Software wie Adobe's After Effects (Adobe, 2017) möglich wurde, hat sich die Arbeitsersparnis durch das Keyframing noch einmal deutlich erhöht. Die Animatoren konnten durch schnelle Manipulation der Keyframes nachträglich den Ablauf verändern, während die Menge der Interframes und das direkte Betrachten des Ergebnisses nur noch von der Rechenleistung bzw. der verhältnismäßig geringen Wartezeit abhingen. Der damit entstandene Workflow kommt in vielen Bereichen moderner Videotechnik zum Einsatz. Von 2-D über 3-D Animation und VFX bis hin zur Postproduktion zieht sich dieses Konzept durch die gesamte Schaffenskette.

## 1.5 Ein Wort zur Interpolation

Um natürliche Bewegungsabläufe zu simulieren, ist es notwendig, Beschleunigungen mit zu berücksichtigen. Völlig lineare Verläufe kommen in der Natur so nicht vor. Denkt man an das Beispiel des "Bouncing Balls" (siehe Kapitel 1.4), so würde z.B. die Verlangsamung hin zum Scheitelpunkt und die anschließende Beschleunigung Richtung Boden fehlen. Die Illusion eines physikalisch korrekten Balls wäre somit dahin. Um dieses Problem zu lösen, haben alle Keyframes die Eigenschaft des Interpolationstyps. Typisch wäre hier z.B. "ease-in" oder "ease-out" bzw. deren Kombination, welche eine positive Beschleunigung zu Beginn und eine negative zum Ende der Phase zur Folge hätte. Die Länge bzw. der Grad der Beschleunigung ist hierbei variabel.

Die Berechnung von Zwischenwerten mit variablen Interpolationsmethoden ist auch in der Welt der Synthesizer üblich. Absolut notwendig ist die lineare Interpolation. Ein ADSR-Envelope ist schließlich auch nur durch 5 distinkte Punkte definiert, die auf Basis einer Interpolationsfunktion verbunden sind und "live" berechnet werden. Aufgrund der Möglichkeit der Modulation aller Parameter in Echtzeit wäre es auch gar nicht möglich, bei einem typischen Synthesizer alle Punkte der Hüllkurve im Voraus zu berechnen. Sollen die Kurven in ihrer Steilheit variiert werden können, bietet sich die logarithmische Interpolation an (siehe Abbildung 5).

```
/**
 * interpolate logarithmically from 1.0 to 0.0f in t samples
@param c counter of the specific phase
@param t number of samples after which the specific phase should be over
@param k coeff of growth/shrink, k=1 for linear
@param slow how fast is phase applied at the start
 */
inline float interpolateLog(int c, int t, float k, bool slow)
{
    float coefficent = static_cast<float>(c) / static_cast<float>(t);
    return std::exp(std::log((slow) ? coefficent : 1.0f - coefficent) * k);
}
```

Abbildung 5: Code-Snippet Interpolation

## 1.6 Das Konzept der Schlüsselbildanimation im Audiobereich

Keyframing in der Bildbearbeitung und Animationstechnik ist ein sehr erprobtes Mittel, schnell und einfach zeitliche Verläufe zu modellieren. Alle Informationen über die Position sind für jeden Zeitpunkt offensichtlich (WYSIWYG), denn bewegt man sich in der Zeitleiste, wird der interpolierte Wert direkt angezeigt (direktes Feedback). Dank der Verkettung von vielen Keyframes können alle erdenklichen Bewegungsabläufe gestaltet werden. Der Nutzer ist hier in keiner Weise durch existierende Paradigmen eingeschränkt, welche die Menge der Schritte oder die Polarität der Bewegungsrichtung vorgeben. Diese Qualitäten sollen in den Audiobereich übertragen werden.

Der Keyframe-Envelope wurde unter der Prämisse entwickelt, alle Funktionen eines ADSR-Envelopes abzubilden und im Rahmen des Sounddesigns voll ersetzen zu können, ohne den Benutzer einzuschränken. Insbesondere das Arbeiten an längeren und komplexeren Hüllkurvenverläufen sollte vereinfacht werden, wofür insbesondere zwei Ansätze zum Einsatz kommen sollten. Alle der Phasen des Envelopes sollen Modularisiert werden, was sowohl für die Anzahl als auch die Zuweisung von MIDI Note-On bzw. -Off Events gilt. Hier durch sollen auch komplexe Attack- und Release-Phasen ermöglicht werden.

Über "Continuous Feedback" soll die Möglichkeit des Vorhörens einer Phase bzw. des modulierten Werts im Kontext des Envelopes ermöglicht werden. Denn gerade bei sehr komplexen Sounds, man denke an die Time-Level-Envelopes, ist es enorm zeitraubend, immer den gesamten Klangverlauf bei kleinen Änderungen anhören zu müssen. In der Metapher des Keyframing entspricht dies der direkten Anzeige von Zwischenbildern beim Bewegen durch die Zeitleiste einer Animation. In der Welt des Klangs entspricht dies der kontinuierlichen Rückmeldung des Sounds inklusive des aktuellen Modulationswerts und Verlaufs. Die Idee ist letztendlich, durch einen "Super-Envelope", welcher alle modulierten Parameter (inkl. den Lautstärkeverlauf) enthält, ein non-lineares Sound-Editing zu entwickeln. Losgelöst vom zeitlichen Verlauf und dem Notenanschlag ließe sich, metaphorisch gesprochen, in ein "Audio-Frame" hereinzoomen, um es im Detail zu betrachten und zu verändern.



Abbildung 6: Frühe Konzeptzeichnung

Die Steuerung des gesamten Envelopes soll ohne Computer-Maus vonstattengehen können (siehe Abbildung 6). Bisherige Lösungen moderner Time-Level-Envelopes orientieren sich hier immer stark am "Zeichnen" von Automationsdaten in der DAW. Obwohl im Ergebnis potentiell gleich, ist diese Art zu arbeiten in Ihrer Indirektheit nach Auffassung des Autors im Sinne einer musikalischen und intuitiven Bedienung sehr abstrakt und verbesserungswürdig. Für einen neuen Workflow gilt es, auch ein passendes Visualisierungskonzept und ein praktikables Mapping der Parameter zu finden, welches sich im Kern an ADSR-Envelopes orientiert, um dem Nutzer die Eingewöhnung so leicht wie möglich zu gestalten.

#### 1.7 Flexible Phasen

Typische Envelopes verfügen über drei oder vier Phasen (siehe Kapitel 1.2). Attack und Release stehen immer an erster bzw. letzter Stelle der Hüllkurve und stehen per Definition mit dem Halten bzw. Loslassen einer Taste (i.d.R. der Klaviatur) im Zusammenhang. Durch eine höchstmögliche Modularisierung der einzelnen Schritte (im Folgenden Steps genannt), soll mit diesem Paradigma gebrochen werden. Jede Phase des klassischen ADSR-Envelopes soll aus mehreren Steps bestehen können; ein "MIDI Note Off" Event kann somit beispielsweise eine 10 Step Release-Phase auslösen, die in sich komplexer ist, als es jeder ADSR-Envelope sein könnte. Modular bedeutet hier aber auch, dass die Anzahl der Steps variabel ist, was für die Darstellung sowie die Steuerung eine besondere Herausforderung bildet. Denn um eine sinnvolle Alternative darzustellen, muss die Steuerung simpler Hüllkurven vom Bedienkomfort her mit dem ADSR-Envelope vergleichbar bleiben. Ein Teil der Lösung dieses Problems ist die Nutzung von Makros, welche mehrere Bedienschritte zusammenfassen und somit schneller und einfacher gestalten. Für die Attack- und Release-Phase, die jeweils aus mehreren Teilen bestehen können, könnte ein solches Makro beispielsweise die Länge aller enthaltenen Steps modifizieren.

#### 1.8 Continuous Feedback

Im Interface-Design spricht man allgemein von Continuous Feedback, wenn beispielsweise eine Parameteränderung in Echtzeit visuell dargestellt wird. Diese Rückmeldung könnte aber z.B. auch haptisch oder auditiv geschehen und ist im Falle von multi-modalen Systemen nicht auf einen Informationskanal beschränkt. Während die Informationsdichte im visuellen Bereich dank flexibler GUIs sehr hoch erscheint, ist das auditive Feedback zumeist recht simpel gehalten. Vordefinierte Warntöne oder Sprachaufnahmen prägen das Bild. Im Zuge des technischen Fortschritts in der Sprachsynthese ist dank Text-To-Speech der Informationsgrad bereits enorm gestiegen. Man denke an das Navigationssystem im Auto, welches dynamisch die nächste Abfahrt samt Namen und Entfernung ansagt. Das Display dient hier möglicherweise nur noch zur optionalen visuellen Nachkontrolle. Das auditive Feedback scheint für diesen Einsatzzweck wie geschaffen, da der Fahrer stets die Straße im Auge behält. Der gewählte Informationskanal ist also von Einsatz abhängig. Die Situation, etwa Planung der Route, oder eine persönliche Vorliebe sind aber ebenso entscheidend.

Um den Informationsgehalt des auditiven Feedbacks auch abseits von Sprachanwendungen zu erhöhen und dynamischer zu gestalten, gibt es, verglichen mit visuellen Mitteln, wenig Bemühungen. Heraus sticht vor allem das schon lange und viel genutzte auditive Feedback in Videospielen. Man denke an dynamische Effekte wie den Hall, wenn ein Auto durch einen Tunnel fährt oder die dynamisch generierten Soundeffekte der 8-Bit Ära.

Klangsynthese ist Echtzeit ist aber auch in klassischen Interfaces zu finden und kann genutzt werden, um eine Vielzahl an verschiedenen Informationen zu transportieren (Bresin et al., 2008; Conan et al., 2014; Rocchesso et al., 2004).

Der nun hier vorgeschlagene Anwendungszweck von Continous Feedback hat zwei Besonderheiten: Obwohl es sich auch um ein multimodales System handelt, denn der Envelope wird auch komplett visualisiert, ist es das Ziel, einen Sound zu designen. Das auditive Feedback ist also keine mehr oder weniger abstrakte Repräsentation einer Information, sondern entspricht dem Ergebnis und sendet auf dem gleichen Informationskanal. Dies hat durch seine Direktheit natürlich extreme Vorteile gegenüber der sehr abstrakten Darstellung einer Grafik. Es entstehen dadurch aber auch potentielle Probleme, da ein auditives Feedback während des Sounddesigns als störend betrachtet werden könnte. Es gilt also darauf zu achten, das Feedback situationsabhängig einzusetzen und dem Nutzer eine Möglichkeit zur Steuerung zu bieten. Im Rahmen der Untersuchung des Prototyps ist das Continuous Feedback allerdings ein elementarer Bestandteil der Fragestellung, weshalb die Nutzung hier nicht optional gestaltet worden ist.

# 1.9 Workflow-Beispiel

Wie auch das Konzept des Keyframes der Animation entlehnt ist, ist der Workflow den Keyframe-Envelopes ähnlich. Der erste Keyframe bzw. Step ist durch seine Position bzw. Amplitude definiert. Die Erstellung des ersten Wertes wäre im Beispiel des "Bouncing Balls" die Y-Position eines Balls. Im Audiobereich entspräche dies dem Offset eines ADSR-Envelopes. Dank Continous Feedback ist grafisch die Position erkenntlich, die sich dynamisch allen Werten anpasst (WYSIWYG). Auditives Feedback bringt durch konstante Wiedergabe des Modulationswerts während der Änderung ein entsprechendes "What-You-Hear-Is-What-You-Get". Moduliert der Envelope beispielsweise die Tonhöhe, bekommt der Nutzer bei der Einstellung der Amplitude im KF-Envelope ein direktes Feedback über dieselbe (siehe Abbildung 7).

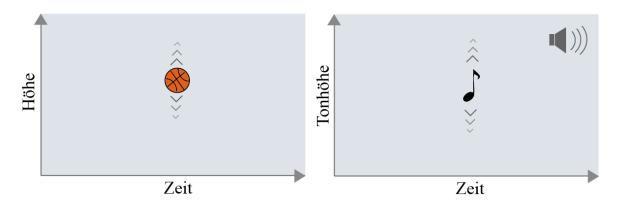


Abbildung 7: Analogie Amplitude

Jeder weitere Schritt verfügt über eine zeitliche Distanz und einen Interpolationstyp. Ob im Weiteren erst der letzte Schritt definiert wird und dann die Zwischenschritte oder komplett konsekutiv ist dem Nutzer überlassen. In der Animation erscheint letzteres bei einer Aneinanderreihung vieler Einzel-Bewegungen natürlicher, wenn man beispielsweise an den kompletten Walk-Cycle (Gehbewegung, inklusive sekundärer Animationen z.B. der Arme) denkt. Die Möglichkeit, an einer beliebigen Stelle die Bewegung um eine oder mehrere Phasen zu erweitern, wäre ein Gegenbeispiel.

Das Continuous Feedback im visuellen Bereich würde hier wieder für eine Echtzeitanzeige der interpolierten Werte sorgen. Für jeden Schritt in der Zeitleiste würde der Ball entsprechend an der korrekten Position angezeigt werden. Durch das Setzen von Loop-Points wie in einer DAW lässt sich die Animation quasi in Echtzeit durch die Manipulation des Keyframes sehr komfortabel durchführen. Im Audiobereich wird dies durch das "Loopen" des aktuellen Steps imitiert, welches automatisch bei der Änderung der Parameter Länge und Interpolation ausgelöst wird (siehe Abbildung 8). Für die Änderung der Amplitude steht wie zu Beginn die konstante Wiedergabe des Werts zur Verfügung.

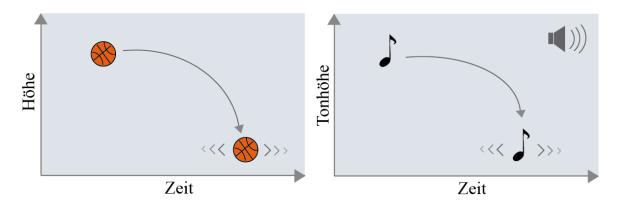


Abbildung 8: Analogie Zeitbereich

Für jeden weiteren Step wiederholt sich dieses Prozedere. Über das Hinzufügen von theoretisch unbegrenzt vielen Schritten lassen sich so extrem komplexe Verläufe erstellen. Die Zuordnung eines Steps zur Release- oder Attack-Phase ist dabei flexibel. Besteht der Sound aus beispielsweise insgesamt 10 Steps und der 3. Step ist als Release definiert, würden nach Loslassen der Taste alle Schritte von 3 bis 10 ausgelöst: Die Nutzung dieser Phasen, ist dabei optional. Ein Sound, der nicht ausklingt, muss also auch über keine Release-Phase verfügen. Für die Steuerung von mehreren Steps gemeinsam, stehen auch Makro-Funktionen zur Verfügung (z.B. Attack Time), welche alle Werte skalieren können.

# 2 Methoden

## 2.1 Der Prototyp

Ziel des Prototyps war es, die Kernelemente des Konzepts des Keyframe-Envelopes zu implementieren, visualisieren und im Rahmen eines praktischen Tests nutzbar zu machen. Primär für eine erste empirische Untersuchung entwickelt, ist der Status zum Zeitpunkt der Auswertung stabil aber experimentell. Eine spätere Veröffentlichung als Open-Source Projekt ist geplant, aber im Rahmen des eigenen Qualitätsanspruchs zeitlich noch nicht absehbar. Der Prototyp ist der Einfachheit halber als Software realisiert worden, die auf einem bestehenden Synthesizer aufbaut. Softwareseitig wurde also der neue Envelope integriert und auf einen ebenfalls bereits existierenden Hardware-Controller zugeschnitten. Denn es sollte ein allgemeingültiges Bedienungskonzept geschaffen werden, welches auch abseits von Maus und Tastatur funktionieren soll. Es bestand zudem auch die Hoffnung, Erkenntnisse aus dem Bereich der Musikperformance, insbesondere moderner, sogenannter "Expressive Controller" (Dobrian und Koppelman, 2006; Jordà, 2002) besser in die Arbeit mit einfließen lassen zu können, da sich hier einige Fragestellungen wie das Parameter-Mapping überschneiden und dies der aktivste artverwandte Forschungsbereich zu sein scheint.

#### 2.2 Softwarebasis

Als Basis für die Entwicklung wurde auf einen Software-Synthesizer zurückgegriffen, der von Studenten der Technischen Universität Berlin entwickelt wurde und als Open-Source zur Verfügung steht (Synister, 2016). "Synister" (siehe Abbildung 9) ist ein typischer subtraktiver Synthesizer mit jeweils 3 Oszillatoren, LFOs und Envelopes. Er verfügt zudem über einen 2 Multimode-Filter, einige Effekte, einen Step-Sequencer und ein relativ flexibles Modulations-System. Implementiert wurde er in C++ unter Zuhilfenahme des JUCE-Frameworks (ROLI Ltd, 2017). Der Synthesizer ist verfügbar als Stand-Alone-Anwendung und Plug-In (VST, AU) für Windows, Mac OS und Linux (experimentell).



Abbildung 9: Synister

Dank der flexiblen Architektur konnte der Keyframe-Envelope mit überschaubarem Aufwand in die Modulationsmatrix eingebunden werden. Als Modulator stand er somit direkt für eine Vielzahl an Parametern zur Verfügung. Was die GUI angeht, so wurde der neue Envelope einem neuen Panel zugeordnet. Dies hatte für den Test den Vorteil, dass der Envelope durch simples "Zuklappen" des Panels für den Nutzer auf dem Computerbildschirm verborgen blieb, und er somit gezwungen war, die Anzeige des Hardware-Controllers zu nutzen.

## 2.3 Implementierte Funktionen

Dem Software-Synthesizer wurden in einem Zeitraum von 5 Monaten diverse Features (Funktionen innerhalb der Software) hinzugefügt. Es folgt eine Liste der wichtigsten Erweiterungen.

#### **Vorbereitung:**

- Änderung der Initialisierungswerte des Synthesizers (Init-Preset) von "Hard Coded" auf das automatische Laden eines Presets (im XML-Format)
- Anpassung der Schrittweite der Knobs für eine präzisere Steuerung

#### **Keyframe-Envelope:**

- Hinzufügen eines neuen Moduls für den Keyframe-Envelope inklusive Serialisierung aller neuen Parameter und Einbindung in die Modulationsmatrix.
- Komplette Modularisierung der Envelope-Klasse in Bezug auf die Schrittmenge.
- Hinzufügen eines "Play Head" zur Darstellung der aktuellen Position innerhalb der Hüllkurve.
- Erweiterung des Envelopes um variable Attack- und Release-Phasen.
- Continuous Feedback Modes: Vorhören des ausgewählten Schritts bzw. Werts und "Scrubbing" inklusive Visualisierung.
- Monophoner Modus.

#### **Push 2 Anbindung:**

- Verbindung zum Push 2 Controller über ein spezielles Framework via libusb (libusb, 2017)
- Hinzufügen der notwendigen Architektur, um den Push 2 Controller automatisch (nur Standalone Version) zu verbinden (Midi und Display).
- Implementierung von Klassen zum Handling aller Midi-CC Events für die Steuerung durch Push 2 inklusive spezieller Anpassungen an die Push 2 API (Ableton, 2017).
- Interface-Design angepasst an das Push 2 Display inklusive "Color-Coding" aller relevanten Bedienelemente.

#### **Optimierungen:**

- Neustrukturierung der Envelope-Architektur, sodass der Code für die Visualisierung (GUI-Thread) und den Audio-Thread annähernd keine Duplikationen mehr aufweist.
- Rendering der Visualisierung des Envelopes parallel in der Software und auf dem Push 2 Display (ohne mehrfache Berechnung des Frames).
- Um den Prototyp zugänglicher zu machen und die Evaluation zu erleichtern, wurden, zusätzlich zur Standalone-Version für Windows, Plug-Ins (VST 2, AU) und ein Build für Macintosh erstellt

#### 2.4 Hardwaresteuerung

Als Hardware-Controller kam der "Push 2" von Ableton zum Einsatz, welcher üblicherweise zur Steuerung der DAW Ableton Live eingesetzt wird. Für den Prototyp wurden alle erweiterten Skripte und Funktionen für Live deaktiviert und das Gerät als reiner MIDI-Controller betrieben. Der Controller bietet mit seinen Pads (die als Klaviatur dienen können), Buttons und Knobs (bzw. Encoder oder Drehregler), alle typischen Eingabemethoden eines Hardware-Synthesizers. Besonders macht ihn aber das Farb-Display im oberen Bereich, welches hochauflösend genug ist, um auch Informationen abseits einfacher Zahlenwerte darzustellen (siehe Abbildung 10).

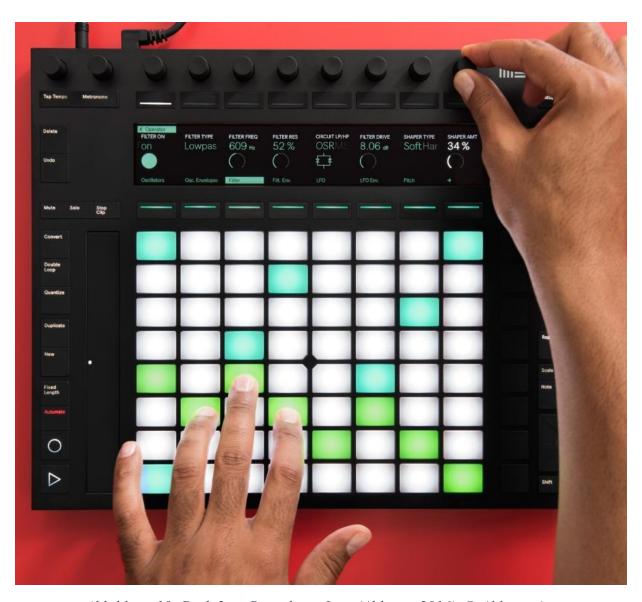


Abbildung 10: Push 2 im Betrieb mit Live (Ableton, 2016), © Ableton Ag

Die Kommunikation zwischen Soft- und Hardware, wurde für die Steuerungsaufgaben über das MIDI Protokoll gelöst. Die Kommunikation verläuft hier in der Regel unidirektional vom Controller zur Software. In Sonderfällen gibt es aber auch bidirektionale Kommunikation, wenn

es darum geht, die LEDs softwareseitig zu steuern. Dank der offen zugänglichen Dokumentation der MIDI-Implementierung (Ableton, 2017), war dies ohne weiteres möglich.

Für die Ansteuerung des Displays wurde eine bisher unveröffentlichte Bibliothek verwendet, welche eine Brücke zwischen JUCE und Push 2 schafft. Mittels libusb können hiermit komfortabel über JUCE generierte Grafiken an das Display gesendet werden. Das mit 960x160 auflösende RGB-Display mit 16 Bit Farbtiefe wurde so mit 60 Bildern / Sekunde versorgt. Dies entspricht auch der typischen Render-Geschwindigkeit am Computer-Bildschirm, was im Sinne der Performance ausgenutzt werden konnte. Die Berechnung jedes Frames (hier im Sinne von Bild) des Keyframe-Envelopes findet nur einmalig statt. Das Ergebnis wird in der GUI der Software und auf dem Display des Controllers dargestellt.

## 2.5 GUI und Mapping

Der Fokus der GUI-Entwicklung lag für den Prototyp voll auf dem Push 2 Controller. Die Steuerung während des Benutzer-Tests sollte ausschließlich über die Hardware vorgenommen werden. Für den Versuch wurden, um Bedienfehler zu vermeiden, letztendlich auch alle (hauptsächlich für interne Tests und das Debugging implementierten) per Maus steuerbaren Komponenten verborgen (siehe Abbildung 11).

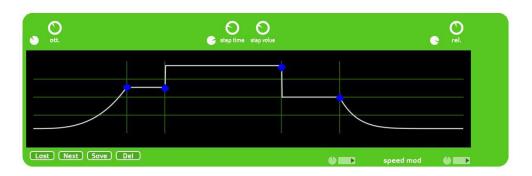


Abbildung 11: Frühe Version des Prototyps mit Maussteuerung

Die finale Version (siehe Abbildung 12) der Anzeige beinhaltet keine Knobs oder Buttons mehr, da diese Funktionen von der Hardware übernommen werden. Im oberen und unteren Bereich befinden sich die Beschriftungen der Encoder bzw. Buttons, welche auch dynamisch den aktuellen Wert anzeigen. Die Farben entsprechen dem allgemeinen Schema (siehe Tabelle 1). In der Mitte wird der Envelope dargestellt, wobei die einzelnen Steps bzw. Phasen in einem Raster angeordnet sind. Der Ausschnitt des Rasters für den aktuell ausgewählte Step ist grau hinterlegt. Der über Interpolation aktuelle manipulierbare Teil der Kurve ist orange markiert, während die über Value modifizierbare Amplitude rot eingefärbt ist.

Die aktuelle Position innerhalb der gespielten Note (Play-Head) wird mit einer gelben Raute dargestellt. Die manuelle Steuerung des Play-Heads per "Scrubbing" ist entsprechend auch gelb gehalten. Das Ende der Attack-Phase wird mit einem blauen Farbverlauf innerhalb des Rasters angedeutet. Für den Beginn der Release-Phase kommt ein grüner Gradient zum Einsatz. Die Kurve an sich ist Weiß gehalten.

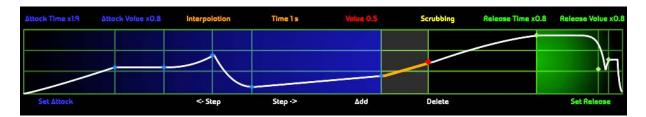


Abbildung 12: Keyframe Envelope auf Push 2 Display

Das Feedback über die Tonhöhe beim Vorhören gibt es direkt über das illuminierte Pad der Klaviatur. Dieses zeigt konstant das zuletzt gespielte Pad (nn43 bis nn92, siehe Abbildung 13) in Rot an. Wenn das Vorhören aktiv ist, pulsiert das ausgewählte Pad.

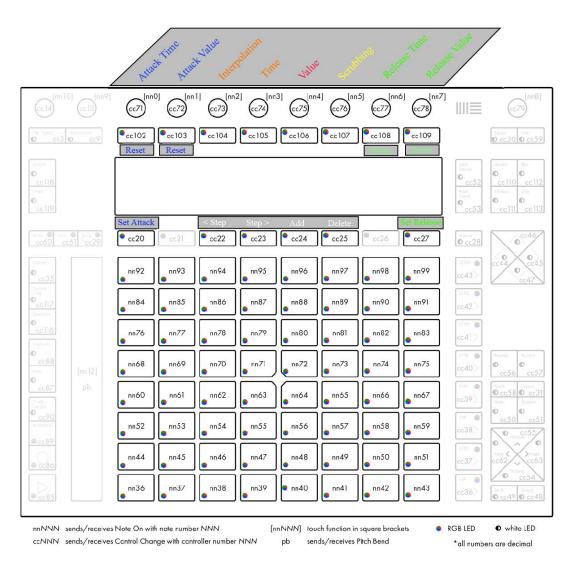


Abbildung 13: MIDI-Mapping des Push 2, © Ableton Ag (Ableton, 2017)

Der Prototyp nutzt die das Display umgebenden Bedienelemente sowie die Pads ähnlich ihrer Originalbelegung als Klaviatur. Die Drehregler cc71 bis cc78 bieten Zugriff auf die im Display oben angezeigte Funktionen. Von links nach rechts sind dies:

- Attack Time: Makro zum Einstellen der Länge aller Steps in der Attack-Phase.
- Attack Value: Makro zum Einstellen der Amplitude aller Steps in der Attack-Phase.
- **Interpolation**: Einstellung des Kurvenverlaufs des aktiven Steps, Vorhören durch Loop des ausgewählten Steps.
- **Time**: Einstellung der Länge des aktiven Steps, Vorhören durch Loop des ausgewählten Steps.
- Value: Einstellung der Amplitude des aktiven Steps, konstantes Vorhören der Amplitude.
- **Scrubbing**: Vorhören durch manuelle Bewegung des Play-Head.
- Release Time: Makro zum Einstellen der Länge aller Steps in der Release-Phase.
- Release Value: Makro zum Einstellen der Länge aller Steps in der Release-Phase.

Die Makros bestehen aus einem Multiplikator mit dem Wertebereich [0.1 - 5.0]. Die Länge eines Steps bewegt sich im Bereich [0.001 - 5.0] Sekunden. Die Amplitude ist normalisiert und bewegt sich im Intervall [0.001 - 1.0].

Die Drehregler c73, c74 und c75 nutzen zusätzlich die Berührungsempfindlichkeit aus, welche auf nn2, nn3 und nn4 senden. Für die Interpolations- und Zeiteinstellung wird hiermit durch reines Berühren des Encoders ein Vorhören im Loop-Modus des jeweils aktiven Steps ausgelöst. Durch Halten des Value-Encoders wird der aktuelle Wert des Steps (rot markiert) kontinuierlich wiedergegeben. Beim Auslösen mehrerer Vorhörmodi durch das Halten von 2 oder mehr Knobs, wird jeweils die letzte Berührung priorisiert.

Die Buttons in der Reihe unter den Push-Encodern mit der Belegung cc102 bis cc109 dienen zum einen mit der farblichen Markierung als Orientierungshilfe für die Knobs. Zum anderen sind sie zum Zurücksetzen der Werte auf den Standard (wenn vorhanden, z.B. Makro Multiplikator = 1.0) nutzbar.

In der Button-Zeile unterhalb des Displays sind die Funktionen zum Setzen des aktuellen Steps als Ende der Attack- (cc20) bzw. Beginn der Release-Phase (cc27). Gleichzeitig dienen diese Knöpfe dem An- und Ausschalten der Phasen (Toggle). Ist der aktuelle Step z.B. als Release-Step bereits ausgewählt und der Knopf wird erneut betätigt, wird die komplette Release-Phase deaktiviert. Diese Knöpfe entsprechen farblich dem allgemeinen Schema der Farbkodierung. In der Mitte zwischen c22 und c25 positioniert finden sich die weiß gehaltenen Step-Funktionen: Last, Next, Add und Delete. Hierüber kann zwischen den Steps navigiert werden und Steps hinzugefügt bzw. entfernt werden. Das Minimum für Steps liegt für den Prototyp bei 3, das Maximum bei 20.

Um eine einheitliche Benutzerführung zu gewährleisten, ist es sinnvoll, bei der Interfacegestaltung mit einer eindeutigen Farbkodierung zu arbeiten (Jones, 1962). Der nachfolgenden Tabelle 1 lässt sich das Schema entnehmen.

Farbe	Funktion
Blau	Attack
Grün	Release
Orange	Time, Interpolation
Rot	Value
Grau / Weiß	Navigation / Aktueller Step

Tabelle 1: Color-Coding

## 2.6 Evaluation des Prototyps

Im Vordergrund steht der Vergleich zwischen dem konventionellen ADSR- und dem neuen Keyframe-Envelope in Bezug auf die Usability und User Experience. Den Hypothesen, dass der KF-Envelope mindestens eine gleichwertige Effizienz, Präzision und Nutzerfreundlichkeit bietet, wird anhand einer ADSR-Envelope Kombination Untersuchungsmethoden nachgegangen. Zur Evaluation des Konzepts wurde ein Prototyp eingesetzt, dessen Einsatz die Erhebung objektiver und quantitativer Daten zu Effizienz und Präzision ermöglichte und mit den subjektiven Daten der Nutzer abgeglichen werden konnte. Denn anschließend an den praktischen Teil des Versuchs wurden Daten zur Usability und UX über Fragebögen erhoben. Über den Einsatz mehrerer Erhebungsmethoden sollte es ermöglicht werden, explorative, qualitative und quantitative Ergebnisse zu erhalten. Der Aufbau einer ca. einstündigen Test-Sessions ist der folgenden Tabelle zu entnehmen (siehe Tabelle 2).

Dauer	Teilbereich
5 min	Einführung
10 min	Freie Exploration des Prototyps
30 min	Target-Finding Sound $1-3$
5 min	Fragebogen AttrakDiff zur Bewertung der Usability
10 min	Allgemeiner Fragenbogen und Feedback

Tabelle 2: Aufbau Test-Session

In der Einführung wurden allgemeine Fragen zum Versuch beantwortet, Probanden über Ihre Rechte aufgeklärt und eine Einverständniserklärung zur Unterzeichnung vorgelegt. Anschließend wurde ihnen das Konzept der Keyframe-Animation im Allgemeinen und dann übertragen auf die Hüllkurve erklärt.

In der freien Erkundungsphase konnten sich die Teilnehmer mit der Bedienung der Hüllkurve vertraut machen und letzte Fragen stellen. Die Wahl der Synthese-Parameter und die Optionen der Modulation waren allesamt vorgegeben, sodass möglichst viele ablenkende Störfaktoren neben der reinen Bedienung der Hüllkurve vermieden wurden.

Für das sogenannte Target-Finding (Tubb, 2016) wurden den Probanden drei unterschiedliche Klänge präsentiert, welche es einmal mit ADSR- und einmal mit KF-Envelope nachzubilden galt. Anschließend wurde anhand einer Audio-Aufnahme der Session die Zeit analysiert, welche die Probanden für die Erstellung benötigten. Die Qualität des erstellten Klangs bzw. die

Ähnlichkeit zum Original wurde mit derselben Aufnahme bestimmt. Durch die statistische Auswertung der Audio-Features konnte dies quantitativ geschehen. Hierzu wurde in Matlab die "Zero-Crossing-Rate" zur Frequenzbestimmung und der "Spectral-Flux" (Lerch, 2012) zur Bestimmung des Filterverlaufs über eine Fensterfunktion verwendet. Die "Zero-Crossing-Rate", welche die Frequenz über die Nulldurchgänge der Wellenform misst, hat sich im Einsatz als sehr zuverlässig gezeigt. Der "Spectral-Flux" hingegen hat für den Filterverlauf die besten Ergebnisse gezeigt, denn durch die berechnete Änderung in der spektralen Gesamtenergie lässt sich die relative Änderung der Filterfrequenz erkennen. Ein minimalistischer Hörvergleich sollte zusätzlich zeigen, ob diese messbaren Unterschiede auch rezeptionsseitig relevant sind.

Im Anschluss an den praktischen Teil des Versuchs, wurde mittels zwei AttrakDiff (User Interface Design GmbH, 2016) - Fragebögen die Usability beider Envelope-Typen erfasst und verglichen.

Abschließend wurde den Probanden ein allgemeiner Fragebogen vorgelegt. Dieser enthielt Fragen zur Erfahrung im Sounddesign, elektronischer Musikproduktion und zur Person. Des Weiteren wurden verschiedenen Qualitätsmerkmale vergleichend zwischen beiden Envelope-Typen abgefragt. Über mehrere offen gehaltene Fragen konnten die Probanden spezielles und allgemeines Feedback geben, welches durch die Fragestellung andernfalls nicht abgedeckt gewesen wäre. Zusammen mit während des Versuchs notierten Beobachtungen bildet dies den explorativen Teil des Versuchs.

## 2.7 Target-Finding

Den Kern des praktischen Versuchs bildete die Reproduktion vorgegebener (Target-) Sounds. Insgesamt wurden den Probanden drei unterschiedliche Sounds (siehe Tabelle 3) vorgespielt, welche es nachzubilden galt. Es wurden unterschiedlich komplexe Sounds gewählt, um eine größere Bandbreite an Informationen zu gewinnen. Begonnen hat der Test immer mit dem einfachsten Sound 1, der sehr ähnlich dem Beginn des zweiten Sounds ist und eine Tonhöhenmodulation beinhaltet. Der dritte Sound ist durch eine relativ langsame Filtermodulation geprägt. Der Target-Sound konnte vor und während der gemessenen Dauer bis zur Fertigstellung des Sounds beliebig oft vom Probanden wiedergegeben werden. Den Start und das Ende der Zeitmessung wurde durch Bestätigung eines Buttons durch den Benutzer bestimmt. Es oblag also dem Probanden zu entscheiden, wann er das Ziel erreicht hat. Bei einer fehlenden Eingabe wurde die erste bzw. letzte Parametermodulation gewertet. Die Probanden wurden angewiesen, einen Kompromiss zwischen Qualität und Schnelligkeit zu finden, da eine genaue Reproduktion innerhalb des Zeitlimits nicht unbedingt möglich war, was die Ergebnisse auch bestätigen.

Sound	Länge	Modulation	Gemessen	Modulationsart	Wellenform	Verlauf
1	1,4 s	Pitch	1 Phase	Unipolar	Rechteck	
2	3,7 s	Pitch	3 Phasen, Sustain	Unipolar	Rechteck	
3	5,5 s	Filter Cutoff	3 Phasen, Sustain	Unipolar, invertiert	White Noise	

Tabelle 3: Sounds Target-Finding

Entscheidend für das Ergebnis sollte am Ende neben der Zeit, die es dauert, den Sound nachzubauen, nur die Länge und im gleichen Maße der Modulationsverlauf sein. Für diesen Zweck sollten die Nutzer von möglichst wenig abgelenkt, sich auf das Wesentliche, den Kurvenverlauf, konzentrieren können.

Bei allen drei Sounds war der maximale Wert des modulierten Parameters insofern vorgegeben, als dass er der maximalen Amplitude des Envelopes entsprach. Die initiale Einstellung des Keyframe-Envelope wurde dem ADSR-Envelope mit nachempfunden. Somit wurde der Maximalwert innerhalb der ersten Phase erreicht, was der Orientierung und Vergleichbarkeit dienen sollte. Es sollte gezielt die Frage geklärt werden, wie sich die beiden Envelopes im Vergleich verhalten, wenn es darum geht, einen Verlauf nach Gehör nachzubauen. Von daher sollte die Interaktion mit anderen Teilen der GUI soweit möglich vermieden werden.

Des Weiteren waren die Nutzer angewiesen, den gesamten Verlauf möglichst genau nachzubauen, wurden aber darüber aufgeklärt, dass beim Sustain nur die Amplitude und nicht die Länge der Phase entscheidend sei. Die Länge des Sustain wird im Falle des ADSR-Envelopes schließlich durch das Halten der Note bestimmt und ist somit nicht für die Messung relevant. Der Keyframe-Envelope verfügte zum Test-Zeitpunkt über keinen Sustain-Modus im klassischen Sinne. Für den Vergleich ist es einfach die letzte Phase vor dem Release. Die Lautstärke des Klangs spielt für das Ergebnis ebenso keine Rolle, weshalb keine Anschlagdynamik verwendet wurde und der typische Einfluss des Envelopes (im Falle von Synister sogar "hard-coded" über Envelope 1) nicht zum Tragen kam. Die Lautstärke ist somit als konstant anzusehen.

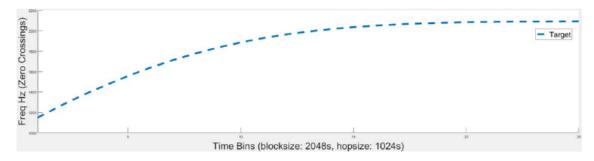


Abbildung 14: Modulationsverlauf Sound 1

Sound 1 (siehe Abbildung 14) besteht aus einer einzelnen Attack-Phase, in welcher sich die Tonhöhe verdoppelt.

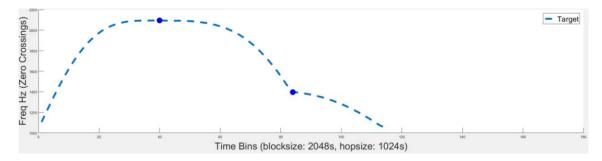


Abbildung 15: Modulationsverlauf Sound 2

Sound 2 (siehe Abbildung 15) besteht aus einer Attack-Phase ähnlich von Sound 1, einer Decay-Phase (startend bei der ersten Markierung), einem Sustain-Wert (2. Markierung) und einer anschließenden Release-Phase.

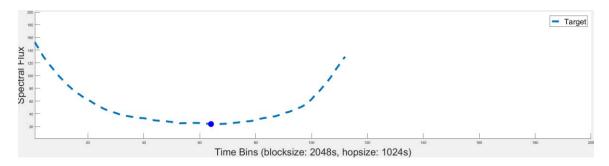


Abbildung 16: Modulationsverlauf Sound 3

Sound 3 (siehe Abbildung 16) besteht aus 2 distinkten Phasen, welche von einer längeren Sustain-Phase getrennt sind. In dieser Darstellung wurde die zeitliche Ausdehnung der Sustain-Phase entfernt und lediglich die Position markiert. Zur besseren Vergleichbarkeit zeigt diese Darstellung anhand des "Spectral Flux" bereits den Frequenzverlauf ähnlich der anderen Sounds. Die tatsächliche Kurve, die es im Envelope nachzustellen galt, ist invers. Für diesen Sound wurde nämlich beim Modulationsziel (Cutoff Frequenz) des Tiefpassfilters ein negativer Wert voreingestellt.

Es galt in einer vorgegebenen Zeitspanne von ca. 5 Minuten (Soft-Limit), einen Klang mit einem Typ Envelope nachstellen. Anschließend wurde der gleiche Sound, mit dem jeweils anderen Envelope-Typ getestet. Die Reihenfolge der Envelope-Typen wurde hierbei variiert, um Gewöhnungseffekte zu minimieren. Denn wie zu erwarten und auch von den Probanden erwähnt, ist es leichter, einen bereits zuvor erstellten Klangverlauf wiederholt nachzubilden.

Im abschließenden Vergleich des Target-Findings (Tubb, 2016), lassen sich über die Dauer und die Genauigkeit des Ergebnisses Rückschlüsse auf die Qualität (Möller et al., 2010) der Benutzungskonzepte ziehen. Während die Auswertung der Dauer noch trivial ist und durch reines Ablesen der Zeiten innerhalb der DAW ermöglicht wurde, ist die Bestimmung der Ähnlichkeit zwischen reproduziertem und Target-Sound aufwendiger.

## 2.8 AttrakDiff-Fragebogen

Der Fragebogen "AttrakDiff" (User Interface Design GmbH, 2016) wurde den Testpersonen nach Abschluss des praktischen Versuchs vorgelegt. Es handelt sich um ein Modell (siehe Abbildung 17) zur Bestimmung der Usability, welches in der Wirtschaft und Wissenschaft eingesetzt wird. Ziel ist es, die User Experience getrennt zwischen hedonistischer und pragmatischer Qualität zu untersuchen und Rückschlüsse auf die Attraktivität des Produkts zu ziehen. Es wird des Weiteren zwischen Nutzer- und Entwicklerperspektive unterschieden, um gesteckte Ziele auf ihr Erreichen hin prüfen zu können.

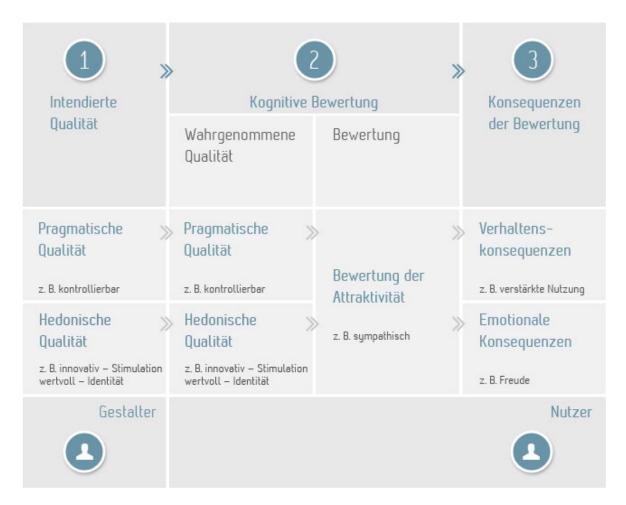


Abbildung 17: AttrakDiff Model © User Interface Design GmbH

Der Fragebogen wurde in der Variante "Vergleich A-B" eingesetzt, um einen Vergleich zwischen dem ADSR-Envelope (im Allgemeinen) und dem Keyframe-Envelope zu erzielen.

Konkret werden über ein semantisches Differenzial 10 Items abgefragt, welche aus sieben Stufen bestehen (siehe Abbildung 18). Des Weiteren werden einige demographische Daten erfasst.



#### Beurteilung des Produkts ADSR Envelope Bitte geben Sie mit Hilfe der folgenden Wortpaare Ihren Eindruck zu ADSR Envelope wieder. Bitte klicken Sie in jeder Zeile eine Position an! einfach\* ( kompliziert hässlich\* schön praktisch\* ( unpraktisch . stilvoll\* stillos voraussagbar\* • unberechenbar . minderwertig\* wertvoll phantasielos\* • kreativ gut\* • schlecht verwirrend\* übersichtlich lahm\* fesselnd \* Pflichtfeld Weiter Zurück

Abbildung 18: AttrakDiff Fragebogen © User Interface Design GmbH

# 2.9 Allgemeiner Fragebogen

Nach Durchführung des praktischen Tests, wurden die Eindrücke der Probanden neben "AttrakDiff" noch mit einem weiteren, speziell auf den Versuch zugeschnittenen Fragebogen abgefragt. Hiermit sollten noch einmal Usability-Merkmale für Envelopes im Vergleich abgefragt werden, Daten zur Versuchsgruppe erfasst und über einen offenen Frageteil neue Erkenntnisse zur Nutzung des KF-Envelopes gewonnen werden. Der Aufbau war wie folgt:

- 5 Fragen zum Erfahrungsgrad im Sound-Design.
- Ein semantisches Differenzial zum Bestimmung dominanter Qualitäten für den ADSRund KF-Envelope im Vergleich.
- 5 offene Fragen zum Einsatz des Keyframe-Envelopes, inklusive allgemeines Feedback.
- Demographische Daten.

Neben der Erfassung einiger persönlichen Daten für eine spätere Analyse von möglichen Korrelationen zwischen den Ergebnissen und Personengruppen hatte der Fragebogen zwei Ziele. Zum einen, sollte der Vergleich zwischen ADSR- und KF-Envelope anhand speziell auf den Einsatz zugeschnittener Fragestellungen (siehe Abbildung 19) weiter vorangetrieben werden und sollten die Ergebnisse aus dem Target-Finding und "AttrakDiff" damit ergänzt werden. Zum anderen sollte über den offenen bzw. explorativen Teil die Möglichkeit der Findung von neuen Fragen und Hypothesen geschaffen werden. Gleichzeitig sollte versucht werden, über quantitative Befragungsmethoden allgemeine Tendenzen in der Einschätzung des Konzepts zu finden.

	ADSR very dominant	ADSR slightly dominant	Equal	"Keyframe" slightly domi- nant	"Keyframe" very dominant
The joy of use					
Complexity of mod- ulation achievable					
The ability to cre- ate sound quickly					
Learnability					
Intuitiveness					
Hearing focus (while searching for the desired sound)					
The visual guidance (while searching for the desired sound)					
Abstractness (the opposite of tangibility)					
Musicality (the op- posite of technical- ity)					

Abbildung 19: Dominierende Qualitäten der Envelopes

Im ersten Teil wird noch einmal kurz die Eingangsvoraussetzung abgefragt, die Vertrautheit mit dem ADSR-Envelope. Die weiteren Fragen stehen mit der Musikproduktion und dem Erfahrungsgrad im Sound-Design im Zusammenhang. Ziel ist es, mögliche Zusammenhänge zwischen präferierten Genres oder Arbeitsweisen mit anderen Ergebnissen herstellen zu können und das Einsatzgebiet bzw. die Zielgruppe des KF-Envelopes besser einschätzen zu können.

- Sind Sie mit dem Konzept des ADSR-Envelopes vertraut?
- Machen Sie Musik, die Synthesizer-Sounds enthält?
  - o Wenn ja, welchem Genre kommt diese am nächsten?
- Betreiben Sie aktiv Sound-Design durch Erstellen eigener Synthesizer-Presets?
  - o Wenn ja, wie würden Sie ihren Erfahrungsgrad im Sound-Design beschreiben?

Die Erfassung demographischer Merkmale wurde an "AttrakDiff" angelehnt und bestand aus den folgenden Punkten:

- Alter
- Geschlecht
- Ausbildung
- Beruf

Der offene Fragenteil war wie folgt zusammengesetzt:

- Wenn der KF der einzige Envelope eines Synthesizers wäre, würden Sie etwas vermissen und wenn ja, was genau?
- Denken Sie, dass die Erweiterung des Envelopes um viele Schritte sinnvoll ist, oder würden Sie lieber traditionell mit mehreren Modulatoren oder DAW-Automation arbeiten?
- Würden Sie den KF-Envelope konkret im Sound-Design einsetzen wollen und wenn ja, für welchen Typ Sound im speziellen?
- Wie bewerten Sie die Idee des "Continuous Feedback"? Bitte erläutern Sie.
- Vermissen Sie spezielle Features, haben Sie allgemeine Verbesserungsvorschläge, Ideen oder Feedback?

Dieser Bereich zielte auf das Finden neuer Fragestellungen ab und sollte andernfalls nicht erfasste Probleme des Konzepts und Prototyps über ein offenes Format abdecken. Die Antwortmöglichkeiten waren allesamt nicht vorgegeben und es gab keine Pflichtangaben. Neben den allgemeinen Fragen zu fehlenden Features oder problematischen Implementierungen richteten sich die Fragen erneut auf das Einsatzgebiet und sollten auch die grundlegenden Ideen des Konzepts beleuchten.

## 2.10 Test-Gruppe

Das Bedienkonzept wurde durch Experimente mit 12 versierten Nutzern empirisch (Seifert und Kim, 2016) evaluiert. Voraussetzung war Erfahrung im Sound-Design, insbesondere mit dem ADSR-Envelope als Modulator. Die Teilnehmer (11 männlich, 1 weiblich) waren alle Mitarbeiter der Firma Ableton, welche Hard- und Software für die Musikproduktion entwickelt. Die Teilnehmer verfügten neben der musikalischen Erfahrung also über ein erweitertes Wissen in der Entwicklung von Audiotechnologie. Die Nationalität der Testpersonen wurde nicht explizit erfasst, allerdings lässt sich allgemein festhalten, dass sie sehr gemischt und europäisch dominiert war. Die Firmensprache ist Englisch.

Von den Teilnehmern, die sich geäußert haben (n=11), waren alle durch die aktive elektronische Musikproduktion mit der Funktionsweise eines ADSR-Envelopes vertraut. Die Personen waren allesamt in der Altersgruppe 20-40 Jahre angesiedelt und verfügten über eine akademische Ausbildung. Beruflich dominierte die Gruppe der Software Engineers (n=7), Sound-Designer, Designer, Quality Assurance (QA) und Student wurden jeweils einmal genannt.

8 Personen gaben an, Sounds von Grund auf zu designen. Von diesem Kreis beschrieb sich die eine Hälfte als mittelmäßig, die andere als sehr erfahren im Sound-Design. Die Maus wurde zur Steuerung von Envelopes oder Automationsdaten von 3 Personen genannt. Dem gegenüber stand die Steuerung per Drehregler (8 Personen). Die dominanten produzierten Musikgenres der Probanden sind der folgenden Tabelle 4 zu entnehmen.

Genre	Nennungen
Hip-Hop / Beats	4
Electronic / Experimental Electronic	4
Trip-Hop	2
Techno	2
Cinematic	2

Tabelle 4: Produzierte Musikgenres der Probanden

#### 2.11 Test-Aufbau

Für den Test kam ein minimales Setup zum Einsatz (siehe Abbildung 20), dessen einzige nicht austauschbare Hardware-Komponente der Push 2 Controller bildete. Neben Laptop, Maus, externer Soundkarte und einem 2. Bildschirm, kam noch ein weiterer MIDI-Controller zur Steuerung des ADSR-Envelopes und zum Abspielen der Target-Sounds zum Einsatz.

Softwareseitig wurde zur Aufnahme Ableton Live und Reaper (reaper.fm, 2017) eingesetzt, in welchem auch der Software-Prototyp als VST-Plug-In lief.



Abbildung 20: Aufbau Test-Studio

Die Steuerung des Keyframe-Envelope geschah komplett über das angepasste Interface des Push 2 Controllers. Alle relevanten Daten den Envelopes wurden über das integrierte Display angezeigt. Als Klaviatur kamen für den gesamten Versuch die Pads des Push 2 zum Einsatz. Die dem Target entsprechende Tonhöhe war entsprechend markiert.

Für die Steuerung des ADSR-Envelopes und die Wiedergabe der Target-Sounds kam ein Novation Launch Control (siehe Abbildung 21) zum Einsatz.



Abbildung 21: Launch Control © Novation (Novation, 2017)

Die Drehregler 1 - 4 bildeten hierbei Attack, Decay, Sustain und Release ab. Es konnte aber auch eine Maus für diesen Zweck eingesetzt werden, um die GUI auf dem Computer-Bildschirm zu steuern. Für die Steuerung der Form der Kurve (Shape, bzw. Interpolation) war die Maus Voraussetzung. Über die Pads 1 - 3 konnten die entsprechenden Target-Sounds abgespielt werden. Pad 8 löste einen "Clap" – Sound aus, mit welchem die Nutzer den Start und das Ende des Target-Findings markieren sollten.

# 3 Auswertung und Ergebnisse

# 3.1 Statistischer Vergleich Target-Finding

Um sich der Antwort auf die Frage zu nähern, ob das Konzept des Keyframe-Envelopes sinnvoll ist, wird in diesem Abschnitt ein statistischer Vergleich der Ergebnisse des Target-Findings angestellt. Dass bei allen 3 Sounds unterschiedliche Merkmale existieren (siehe Kapitel 2.7), sei an dieser Stelle noch einmal erwähnt. Ein Vergleich ist somit nicht direkt über alle Merkmale und Sounds hinweg möglich. In direkter Relation stehen nur die Merkmale einzelner Sounds zueinander (siehe Tabelle 5).

Sound 1	-	Fehler Amplitude	-	Hörvergleich	Erstellungsdauer
Sound 2	Zeitlicher Fehler	Fehler Amplitude	Sustain	Hörvergleich	Erstellungsdauer
Sound 3	Zeitlicher Fehler	Fehler Amplitude	Sustain	Hörvergleich	Erstellungsdauer

Tabelle 5: Vergleichbare Merkmale

Der zeitliche Fehler setzt sich aus der Summe der Länge der einzelnen Phasen zusammen. Die Länge der Sustain-Phase geht nicht in die Rechnung mit ein, da dies nicht Teil der Fragestellung war. Im Fall des ersten Sounds mit nur einer relativ kurzen Phase wird die Länge nicht betrachtet, da es keine relevanten Unterschiede gibt. Die Unterschiede sind hier zum einen nicht hörbar und auch numerisch so klein, dass sie im Rahmen der Genauigkeit der Phasenerkennung nicht aussagekräftig sind.

# 3.2 Aufbereitung der Daten des Target-Finding

Um die Unterschiedlichkeit der reproduzierten Sounds der Probanden zum originalen "Target-Sound" zu bestimmen, wurden diese mit Matlab aufbereitet. Die Samples, händisch aus der Gesamtaufnahme der Session geschnitten, wurden im ersten Schritt von führenden und anschließenden Nullen bereinigt. Somit konnte die Länge des Sounds samplegenau bestimmt werden. Für Sounds 1 und 2 (Pitch-Verlauf) wurde das Material initial tiefpassgefiltert. Die für die weitere Frequenzanalyse per "Zero Crossing Rate" unnötig und möglicherweise störenden Obertöne und Artefakte des Synthesizers konnten so minimiert werden.

Für den Sound 3 (Filterverlauf) wurde das Signal erst anhand des Features "Spectral Flux" analysiert und das Ergebnis tiefpassgefiltert. Aufgrund der stochastischen Natur des Quellmaterials (gefiltertes, weißes Rauschen), war dies notwendig, um einen "glatten", natürlichen Verlauf der realen Modulation des Filterverlaufs abzubilden. Dazu wurde ein relativ starker Filter (Moving Average, 15 bins) eingesetzt. Ohne diese Korrektur ist bereits die visuelle Analyse schwierig (siehe Abbildung 22).

Des Weiteren wurden die Werte des "Spectral Flux" auf das Maximum des Target-Sound normalisiert. Da es für den späteren Verlauf weniger um die exakten Ergebnisse des "Spectral Flux" an sich, als mehr um den Frequenzverlauf des Filters ging, wurde die unterschiedliche maximale Höhe (insbesondere Endwert) als Artefakt betrachtet. Denn wie auch bei den Sounds 1 und 2 war die Modulationshöhe vorgegeben, also hier die maximale und minimale Filterfrequenz. Erklärt werden kann diese Differenz in den Maximalwerten durch die Eigenschaft des "Spectral Flux", bei schnellen Änderungen in der Gesamtenergie des Spektrums höhere Werte zu produzieren. Für die reine Frequenzanalyse des Filters musste dieser Effekt also kompensiert werden.

Für einige wenige reproduzierte Sounds, die aufgrund technischer oder bedienungsseitiger Fehler minimale Abweichungen des vorgegebenen Maximalwerts (nicht Teil der Erhebung) um bis zu einem Halbton aufwiesen, wurde dieser in Matlab entfernt, um das Ergebnis nicht zu verfälschen. Bei Sound 3, der ursprünglich aus 3 Phasen bestand, wurde für die Analyse Attack und Decay zusammengefasst, denn der Amplitudenunterschied lag unterhalb der Wahrnehmungsschwelle und wurde von den Testpersonen nicht reproduziert (siehe Differenz Tabelle 3 und Abbildung 16).

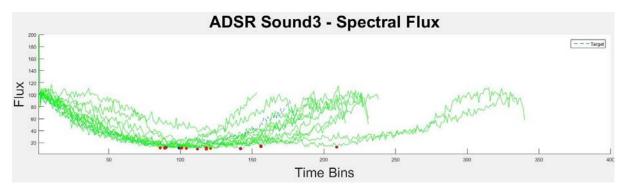


Abbildung 22: Spectral Flux Sound 3, ungefiltert

Um den zeitlichen Fehler exakt bestimmen zu können, wurden die einzelnen Phasen der Sounds einzeln betrachtet. Hierfür kam wiederum ein Matlab Script zum Einsatz, welches die erste bzw. Attack-Phase, die Sustain-Phase und die letzte bzw. Release-Phase über markante Punkte (Maxima etc.) erkennen konnte und im Graphen zur visuellen Kontrolle markierte. Dies lässt Rückschlüsse auf die Fehlerverteilung pro Phase zu. Es dient allerdings auch der Vermeidung des Folgefehlers im Frequenzbereich durch Verschiebungen. Die Sustain-Phase wurde für die weitere Berechnung der Gesamtdauer entfernt und nur der Sustain-Wert für den späteren Vergleich gespeichert.

Nach der Bestimmung der Phasen und des Sustain-Werts wurden letztendlich die Längen der unterschiedlichen Phasen bestimmt. Bei unterschiedlichen Längen der Phase wurde entsprechend der kürzere Sound durch eine Wiederholung des letzten Wertes verlängert. Diese Option birgt zwar einen Fehler in sich, für die am Ende stehende Analyse der Gesamtdifferenz wurde dieser allerdings als weniger stark eingeschätzt als das Kürzen auf die kürzere Phase, oder eine Normalisierung (Stauchung bzw. Streckung).

## 3.3 Bestimmung der Differenzen zwischen reproduzierten und Target-Sound

Für die Bestimmung der Qualität des reproduzierten Sounds wurden die aufbereiteten Daten (siehe Kapitel 3.2) in Hinblick auf deren Länge und Modulation untersucht. Um einen möglichst soliden Vergleich mit dem Target-Sound anstellen zu können, wurden an jeden Sound angepasste Kriterien (siehe Tabelle 6) betrachtet.

Sound	Gemessene Amplitude	Gemessene Länge	Modulation	Analyse Feature
1	1 Phase	Länge fix	Frequenz	Zero-Crossing-Rate
2	2 Phasen, Sustain-Wert	Länge Phase 1,2,3	Frequenz	Zero-Crossing-Rate
3	2 Phasen, Sustain-Wert	Länge Phase 1,2,3	Filterfrequenz	Spectral Flux

Tabelle 6: Differenz-Kriterien pro Sound

Im ersten Schritt wurden für alle Sounds die Audio-Features extrahiert. Um einen Zeitverlauf zu erhalten, wurde entsprechend (überlappend) gefenstert. Die Analyse erfolgte mit folgenden Einstellungen:

block-size: 2048 sampleshop-size: 1024 samples

Für die Sounds 1 und 2 kam zur die Frequenzanalyse die "Zero-Crossing-Rate" (siehe Formel 1) zum Einsatz, welche die Modulation der Tonhöhe zuverlässig abbildete.

$$\sum_{k=1}^{K} |\operatorname{sgn}[x(m)] - \operatorname{sgn}[x(m-1)]| |w(n-m)|$$

Formel 1: Zero Crossing Rate

Bei Sound 3, dem gefilterten weißen Rauschen, ergab das tiefpassgefilterte Feature "Spectral Flux" (siehe Formel 2) die besten Ergebnisse.

$$\frac{1}{k} \sum_{k=1}^{K} (|x[n-1,k]| - |x[n,k]|)^2$$

Formel 2: Spectral Flux

Die Längenbestimmung der einzelnen Phasen geschah ganz trivial anhand der gefundenen Start- bzw. Endpunkte der Phasen. Diese sind in den Graphen für den Target-Sound blau und für alle anderen Sounds rot (Ende Attack) und gelb (Beginn Release) markiert (siehe Abbildung 23). Für die Sounds 2 und 3 ergibt sich die Länge des gesamten Sounds aus der Summe der ersten und letzten Phase. Bei Sound 1 ist die Länge fix und wird daher nicht in der Analyse betrachtet.

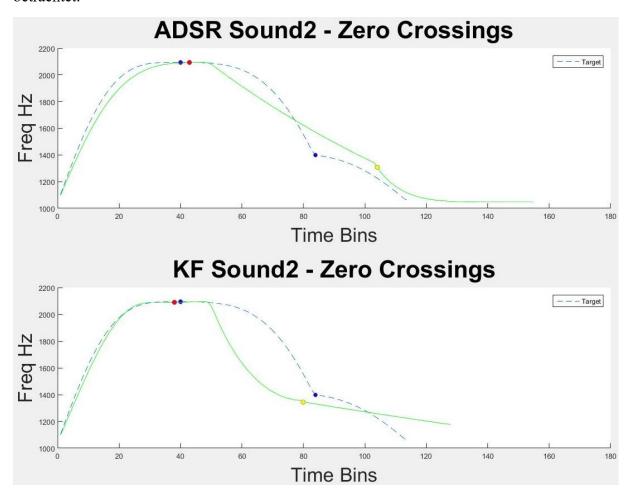


Abbildung 23: Detailansicht Session 2, Sound 2 / Target-Sound

Die Berechnung des Fehlers bzw. der Differenz des Audio-Features zur Bestimmung der Abweichung der Modulation geschah basierend auf den Kenntnissen über die Länge und Position der Phasen. Entsprechend kann direkt die jeweilige Phase des reproduzierten Sounds mit der Phase des Targets-Sounds verglichen werden. Bei unterschiedlichen Längen wurde zur Näherung der kürzere Sound in der Länge angepasst (siehe Kapitel 3.2). Anschließend wurde

der mittlere quadratische Fehler (siehe Formel 3) als Differenzmaß verwendet. Am Ende steht pro Sound, Session und Envelope-Typ ein einzelner Fehlerwert für die Amplitude und die Zeit.

$$RMS = \sqrt{\frac{1}{n} \sum_{i=1}^{n} x_i^2}$$

Formel 3: Quadratisches Mittel

Der Grand Average über alle Sessions lässt sich in Tabelle 7 ablesen. Die Werte sind hierbei auf die größte Abweichung des jeweiligen Attributs über beide Envelopes auf 1 normalisiert. Ein numerischer Vergleich ist somit nicht über alle Sounds hinweg möglich, wie eingangs erwähnt. Die Verhältnisse zwischen ADSR- und KF-Envelope lassen sich also immer nur pro Sound und Feature betrachten. Die Ergebnisse sind aber auch eindeutig genug, um sagen zu können, dass es mathematisch betrachtet einen kleineren Unterschied zwischen der Target- und der von den Probanden entwickelten Kurve beim ADSR-Envelope gibt. Die einzige Ausnahme bildet der Sustain-Wert beim Keyframe-Envelope Sound 3, der insgesamt aber nicht im hörbaren Bereich liegt.

Sound	Amplitude		Time		Sustain	
	KF	ADSR	KF	ADSR	KF	ADSR
1	0,25	0,21	-	-	-	-
2	0,54	0,36	0,30	0,29	0,51	0,04
3	0,67	0,34	0,48	0,18	0,42	0,46
Median	0,54	0,34	0,39	0,24	0,47	0,25

Tabelle 7: Gesamtdifferenzen (normalisiert) pro Sound

Es folgen Plots für die Sounds 1-3 (Abbildung 24, Abbildung 25, Abbildung 26). Diese zeigen alle Sessions des jeweiligen Envelopes (grün) im Vergleich zum Target (gestrichelt, blau).

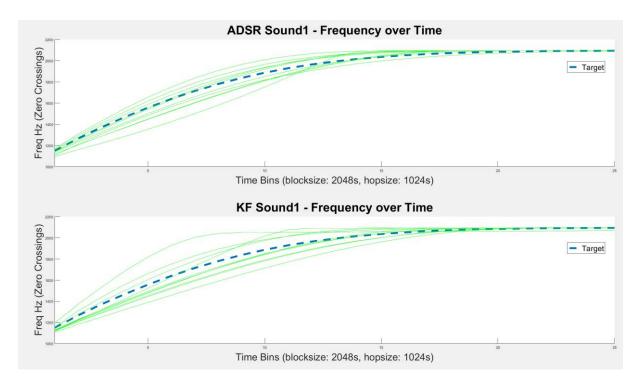


Abbildung 24: Frequenzverlauf Sound 1

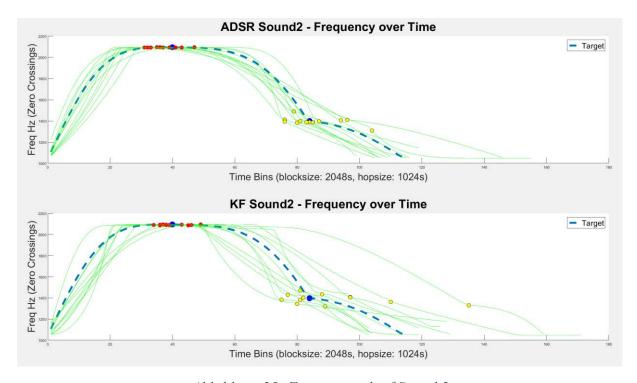


Abbildung 25: Frequenzverlauf Sound 2

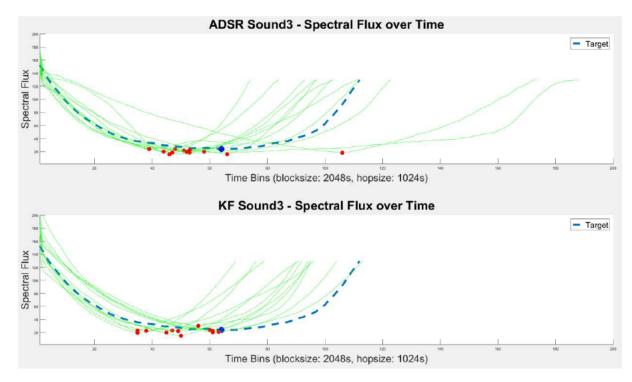


Abbildung 26: Filterverlauf Sound 3

### 3.4 Untersuchung der Signifikanz der Unterschiede des Target-Findings

Bei den messbaren und anhand der Graphen (siehe Abbildung 24, Abbildung 25, Abbildung 26) auch schon rein visuell erkennbaren Unterschieden zwischen den reproduzierten Sounds per ADSR- und Keyframe-Envelope stellt sich aus statistischer Sicht die Frage der Signifikanz.

Da beide Envelope-Typen von der gleichen Personengruppe getestet wurden, handelte es sich um verbundene Gruppen. Eine Überprüfung auf die Normalverteilung der Differenzen schlug fehl, weshalb zur Untersuchung anstelle eines gepaarten T-Tests der Wilcoxon-Signed-Rank-Test (Rey und Neuhäuser, 2011) verwendet wurde. Der Test wurde beidseitig (bei Verdoppelung des signifikanteren einseitigen p-Werts) zum Vergleich der Differenzen beider Envelope-Typen über alle 12 Sessions hinweg durchgeführt. Am Ende steht pro Sound also potentiell jeweils ein Gesamtfehler für die X- und Y-Achse bzw. die Zeit und das untersuchte Audio-Feature (Frequenz bzw. "Spectral Flux"). Hierüber lassen sich numerische Rückschlüsse auf die Hypothese der vergleichbaren Präzision gewinnen.

Wilcoxon Signed-Rank-Test (two sided), $\alpha = 5\%$					
Sound	Amplitude p	Amplitude rank	Time p	Time rank	
1	0.2334	55	-	-	
2	0.0771	62	0.9263	40.5	
3	0.0269	11	0.3296	26	

Tabelle 8: Ergebnisse Wilcoxon Rank Test

Der Wilcoxon Test zeigt, dass es beim Vergleich der Unterschiede zwischen ADSR- und KF-Envelope überwiegend keine signifikanten Unterschiede gab. Lediglich bei der Amplitude des Sound 3 ist dies nicht der Fall (siehe Tabelle 8).

### 3.5 Kontrolle der statistischen Analyse

Um die Korrektheit des Ansatzes zur Analyse der quantitativen Daten zu gewährleisten, wurde ein minimalistischer Hörversuch mit 4 Personen durchgeführt. Auf einer Skala von 1-3 (kein Unterschied, minimaler Unterschied und großer Unterschied) wurde die wahrnehmbare Differenz jedes erstellten Sounds zum Target erfasst. Dieser Test ist für sich genommen statistisch nicht belastbar und wird ausdrücklich nur als grobes Qualitätsmaß für das eigentliche Messgerät, den statistischen Fehler, angeführt. Auf psychoakustische Phänomene und Wahrnehmungsschwellen wird an dieser Stelle auch verzichtet.

Vergleicht man die Test-Sessions als Gesamtes, erhält man eine Aussage über die Verteilung zwischen den Probanden. Schaut man sich den Gesamtfehler, bestehend aus zeitlicher und frequenzseitiger Abweichung zum Target-Sound für beide Envelopes an und vergleicht diese mit den Ergebnissen des Hörtests, zeigen sich 3 Gruppen (siehe Tabelle 9).

Session Ranking - Total Error (Median, normalized)			Session Ranking – Listening Test (Median, normalized)		
4	0,23		2 0,17		
2	0,24		4	0,24	
1	0,26		1	0,29	
5	0,31		7	0,32	
6	0,33		8	0,36	
8	0,37		3	0,40	
7	0,39		6	0,42	
3	0,42		5	0,46	
11	0,51		10	0,49	
9	0,58		12	0,49	
12	0,64		11	0,54	
10	0,70		9	0,61	

Tabelle 9: Test-Sessions im Vergleich

Wie man sieht, sind die Gruppen der besten 3 sowie der 4 schlechtesten Ergebnisse deckungsgleich. Das Modell scheint also die menschliche Wahrnehmung von Unterschieden abbilden zu können. Es sei an dieser Stelle aber noch einmal darauf hingewiesen, dass messbare Differenzen generell nicht identisch mit perzeptiven Unterschieden sind. Die spektrale und temporale Auflösung des menschlichen Gehörs ist im Vergleich geringer, was zu einem natürlichen Fehler führt. Letztendlich relevant ist für das Sound-Design natürlich nur der wahrnehmbare Unterschied, weshalb diese objektiven Daten nur eine Indikatorfunktion besitzen.

# 3.6 Ergebnisse der Zeitmessung des Target-Finding

Im Vergleich der Zeit, welche die Probanden benötigten, zu einem für sie befriedigenden Ergebnis zu kommen, zeigt sich insgesamt ein klarer Vorteil beim ADSR-Envelope (siehe Tabelle 10).

Dauer in Sekunden Total (Mittelwert)				
ADSR	KF	%		
159,7	187,8	17,5		

Tabelle 10: Dauer Target-Finding Total

Die Unterschiede der einzelnen Sounds lassen sich der folgenden Tabelle 11 entnehmen.

Dauer in Sekunden (Mittelwert)				
Sound	ADSR	KF	%	
1	85,8	139,5	62,5	
2	227,1	241,4	6,3	
3	166,3	182,3	9,7	

Tabelle 11: Dauer Target-Finding per Sound

### 3.7 Ergebnisse AttrakDiff

Die Auswertung der Ergebnisse des AttrakDiff-Fragebogens zeigen insgesamt ein recht ähnliches Ergebnis für den A/B-Vergleich beider Envelope-Typen. Der Keyframe-Envelope führt in den Kategorien hedonistische Qualität und Attraktivität leicht, während die pragmatische Qualität ausgeglichen ist (siehe Abbildung 27).

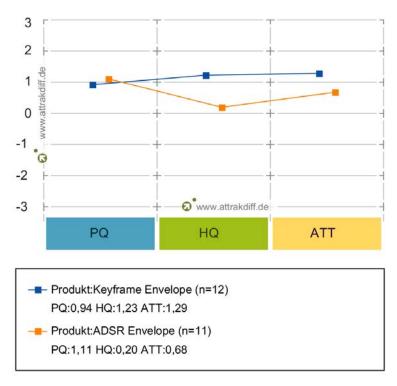


Abbildung 27: AttrakDiff – Mittelwerte

Betrachtet man die hedonistische und pragmatische Qualität gemeinsam (siehe Abbildung 27), zeigt sich wiederum ein leichter Vorsprung des Keyframe-Envelopes. Die unterschiedlichen Teilnehmerzahlen für KF- und ADSR-Envelope (n=12 und n=11) erklären sich durch das Fehlen eines Datensatzes aus unbekannten Gründen. Der maximale Unterschied beim Entfernen eines beliebigen Datensatzes des Keyframe-Envelope beträgt weniger als 4% und ist somit nicht signifikant.

Im der Detailansicht (siehe Abbildung 28) zeigt sich erneut die Ähnlichkeit über den Kurvenverlauf. Der ADSR-Envelope gewinnt bei der Einfachheit deutlich und wird als etwas übersichtlicher wahrgenommen. In den anderen Bereichen liegt der Keyframe-Envelope vorn und zeigt seine Stärken am deutlichsten bei den Attributen kreativ und fesselnd.

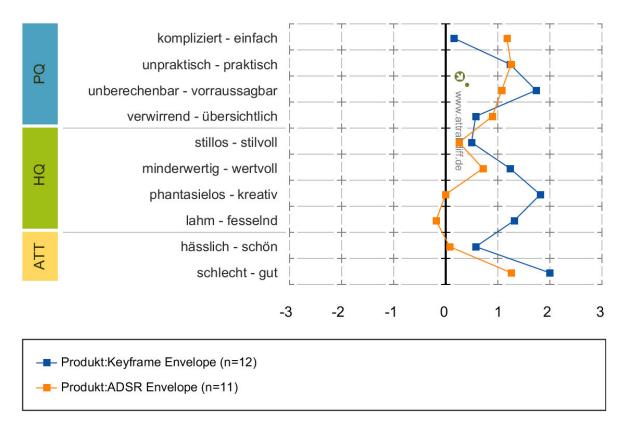


Abbildung 28: AttrakDiff - Wortpaare

### 3.8 Usability von Envelopes im Vergleich

Die Untersuchung von Usability-Merkmalen, die auf Envelopes im Speziellen zugeschnitten war, hat den Trend der allgemeinen Ergebnisse des AttrakDiff-Fragebogens insoweit fortgesetzt, als dass die Ergebnisse überwiegend nahe beieinanderliegen. Stärken zeigt der Keyframe-Envelope bei der erreichbaren Komplexität von Modulation, dem hilfreicheren visuellen Feedback und weniger deutlich beim Spaßfaktor (siehe Abbildung 29). Deutliche Schwächen existieren nicht.

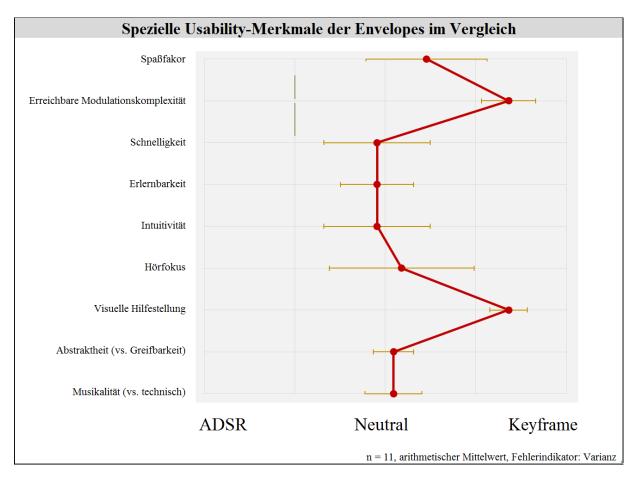


Abbildung 29: Envelope-Usability im Vergleich

### 3.9 Ergebnisse der offenen Befragung

Eine Analyse der am häufigsten genannten Schlagwörter hatte folgendes Ergebnis:

Wenn der KF, der einzige Envelope eines Synthesizers wäre, würden Sie etwas vermissen und wenn ja, was genau?

4x nein

3x direkter Parameterzugriff ohne Auswahl eines Steps

2x weniger Optionen / Komplexität

1x Möglichkeit für Live-Performance

1x Presets für schnellere Änderungen

Denken Sie, dass die Erweiterung des Envelopes um viele Schritte sinnvoll ist, oder würden Sie lieber traditionell mit mehreren Modulatoren oder DAW-Automation arbeiten?

10x ja

1x nein

Würden Sie den KF-Envelope konkret im Sound-Design einsetzen wollen und wenn ja, für welchen Typ Sound im speziellen?

4x lange und sich entwickelnde Sounds

3x pad sounds

2x ambience sounds

2x texture sounds

2x noise

Atonal, Drones, abstrakte Sounds, organische Sounds, Strings, gesturale Sounds, Rhytmus-Patterns, Reproduktion analoger Instrumente, Soundscapes

Wie bewerten Sie die Idee des "Continuous Feedback"? Bitte erläutern Sie.

8x sehr hilfreich

5x die Implementierung könnte weniger störend sein (Toggle, keine Loops)

2x manchmal Sound nicht im Kontext (sehr kurze Loops wie LFO-Modulation)

2x zeitsparend

Einladend zum Spielen, nicht gut zum performen

Vermissen Sie spezielle Features, haben Sie allgemeine Verbesserungsvorschläge, Ideen oder Feedback?

Bessere Skalierung (konstante Zeitdarstellung, Zoom), Zeichnen von Steps, Sustain-Modus, andere Positionierung der Steuerung, Option für Live-Performance, Darstellung zu ähnlich zum ADSR

## 4 Diskussion

### 4.1 Target Finding

Um einen möglichst detaillierten Vergleich der Envelopes zu ermöglichen, wurden mehrere Analysemethoden eingesetzt. Der praktische Test des Target-Findings beleuchtet die Aspekte Dauer und Genauigkeit der Reproduktion. Wie in Tabelle 7 ersichtlich ist, findet sich der größere Fehler für die Zeit und Amplitude auf Seiten des KF-Envelope, wobei der Amplitudenunterschied bei Sound 1 und die Zeitdifferenz bei Sound 2 nur marginal ist. Der Sustain-Wert ist perzeptiv nicht erfassbar und somit wenig relevant. Hierdurch ist allerdings eine Schwäche des Prototyps aufgedeckt worden, da der KF-Envelope bis dato keine Möglichkeit bietet, eine mit Sicherheit wertkonstante Sustain-Phase zu erstellen, denn die Schrittbreite des Amplitudenwerts liegt deutlich über der visualisierten Auflösung der Amplitude.

Betrachtet man die Gesamtunterschiede in Hinblick auf ihre Signifikanz (siehe Tabelle 8), wird deutlich, dass die eingangs erwähnten Fehler der Zeit und Amplitude in 4 von 5 Fällen dieses Kriterium nicht erfüllen. Lediglich die relativ großen Unterschiede in der Amplitude des Sounds 3 (p = 0,03) sprechen statistisch gesehen gegen eine zufällige Verteilung der Differenzen. Dieses Ergebnis ist allerdings aufgrund der Sonderstellung des Sounds 3 nur bedingt aussagekräftig. Denn wie auf Abbildung 22 zu erkennen ist, führt der tiefpassgefilterte "white noise" zu starken Sprüngen im zur Bestimmung der Filterfrequenz eingesetzten "Spectral Flux". Das Ergebnis musste für den Vergleich erst relativ stark gefiltert werden, was in sich einen erhöhten Fehler birgt. Wenn man sich die Ergebnisse der einzelnen Sessions ansieht (siehe Abbildung 24, Abbildung 25, Abbildung 26), fallen zwar einige deutliche Ausreißer auf. Diese finden sich allerdings bei beiden Envelope-Typen (z.B. zeitlicher Fehler KF Sound 2 und ADSR Sound 3) und im Mittel sind sie kaum noch relevant. Insgesamt kann man also festhalten, dass es zwar einen messbaren und auch hörbaren Unterschied gibt, dieser aber so gering ist, dass es sich um einen zufälligen Effekt handeln könnte.

Bei der Auswertung der Dauer für die Reproduktion des Target-Sounds (siehe Tabelle 11) wird deutlich, wie stark das Ergebnis von der kurzen Eingewöhnungszeit der Probanden abhängt. Denn während mehrjährige Erfahrung mit dem ADSR-Envelope vorlag, betrug die Eingewöhnungszeit für den KF-Envelope 10 Minuten. Diese zeigt sich am Beispiel des ersten und mit Abstand simpelsten Sounds (siehe Tabelle 6). Hier existieren noch erhebliche Differenzen in der Dauer und der KF-Envelope ist sehr deutlich unterlegen. Im Folgenden ist der Unterschied sehr viel kleiner ausgeprägt, obwohl die Komplexität der Sounds zunimmt. Dies würde auch für die These sprechen, dass der KF-Envelope für sein eigentliches Haupteinsatzgebiet sehr komplexe Modulationsverläufe besser geeignet ist. Ein Vergleich mit dem ADSR-Envelope ist aber unmöglich, da derartig vielschrittige Verläufe nicht mit dieser Technik umsetzbar sind.

Zum Target-Finding lässt sich abschließend festhalten, dass der Vorsprung des ADSR-Envelope eher gering ausfällt. Eine Untersuchung auf Korrelationen zwischen Dauer und Qualität der Reproduktion hat keine schlüssigen Ergebnisse zu Tage gefördert. Es gibt zwar einzelne sehr gute und schnelle Ergebnisse spezieller Probanden, diese sind aber nicht über alle Sounds hinweg konsistent. Durch Verbesserungen am Prototyp und mehr Erfahrung der

Testpersonen könnte sich dieses Verhältnis potentiell noch stark ändern. Für die nicht untersuchten, komplexeren Modulationsverläufe jenseits der Möglichkeiten des ADSR-Envelopes lässt sich hier nur mutmaßen. Die Ergebnisse machen aber deutlich, dass bereits die Reproduktion vom Sound mit 2 oder 3 Phasen selbst für Experten nicht einfach ist und die Teilnehmer innerhalb von 5 Minuten nicht immer zu einem sie selbst zufriedenstellenden Ergebnis kommen. Ein Vergleich zwischen KF- und modernem Time-Level-Envelopes bzw. mehreren verschalteten Modulatoren dürfte sich als sehr schwierig durchführbar herausstellen.

### 4.2 Usability

Die Usability-Untersuchungen anhand des allgemeinen AttrakDiff-Fragebogens und den speziellen Fragen zu Envelopes kommen zu dem Ergebnis, dass der KF-Envelope in keinem Fall merklich schlechter und in einigen Bereichen besser als der ADSR-Envelope abschneidet. Die größten Vorteile innerhalb des Modells von AttrakDiff finden sich in den Bereichen hedonistische Qualität und Attraktivität wieder (siehe Abbildung 27). Er wird als wertvoller, kreativer, fesselnder und besser wahrgenommen. Bei den praktischen Qualitäten sticht die Voraussagbarkeit positiv hervor, während der ADSR-Envelope als einfacher bewertet wurde (siehe Abbildung 28). In der Portfolie-Analyse zeigt sich der KF-Envelope näher dem Optimum zwischen hedonistischem und pragmatischen Maximum "begehrt". Zusammen mit einer höheren Streuung liegt der ADSR-Envelope in der pragmatischen Qualität erneut leicht vorn.

Die Untersuchung von Usability-Merkmalen, die speziell auf Envelopes abgestimmt waren, hatte ein sehr ausgeglichenes Ergebnis (siehe Abbildung 29). Nennenswerte Unterschiede gab es nur für 3 von 9 Merkmalen: Spaßfaktor, erreichbare Modulationskomplexität und visuelle Hilfestellung. Alle diese Punkte konnte der KF-Envelope dominieren. Die Hoffnungen, ein System zu entwickeln, welches insbesondere für technisch weniger versierte Musiker einfacher zu lernen, intuitiver und musikalischer ist, konnte durch die Ergebnisse hier nicht gezeigt werden. Hier wäre eine Analyse der unterschiedlichen Personen zielführend, denn während der Gespräche beim Versuch hat sich gezeigt, wie ausschlaggebend doch persönliche Faktoren wie Musikgeschmack oder Herangehensweise beim Sound-Design sind. Die Varianz der Ergebnisse ist insgesamt auch recht hoch, was dies bestätigt.

#### 4.3 Qualitativer Teil

Die offene Befragung lässt sich in drei Bereiche aufgeteilt betrachten: Das Konzept, die prototypische Implementierung des Konzepts und den praktischen Einsatz. Zur Überprüfung des Konzepts an sich wurden die beiden Kernpunkte des KF-Envelope abgefragt, welche die Modularisierung der Schritte und das Continuous Feedback sind. Die Nutzung einer variablen Anzahl von Schritten innerhalb eines Envelopes erschien 10 von 11 befragten Teilnehmern der Studie als sinnvoll. Nur eine Person würde demnach Kombination von Modulatoren oder den Umweg über die Automation innerhalb einer DAW bevorzugen. Das Continous Feedback wurde von immerhin noch 8 Personen als sehr hilfreich angesehen und zwei Mal entgegen den Messungen als zeitsparend bewertet. Das häufigste genannte Problem ist die noch nicht optimal gelöste Kontrolle über Zeitpunkt und Kontext des Feedbacks, was keine konzeptionelle Schwäche darstellt. Im Rahmen des Tests wurde allerdings auch bewusst auf die Option zur Deaktivierung des Continuous Feedback verzichtet, um möglichst viele Daten zu dieser Fragestellung zu erhalten.

Auf die Frage, ob die Nutzer spezielle Features bei der Implementierung vermissen, war die Meinung geteilt. 4 Personen vermissten nichts, während 4 Personen den für die Live-Performance notwendigen Direktzugriff auf alle Parameter vermissten. Dies ist ein durchaus wichtiger Kritikpunkt. Während der Arbeit am Prototyp hat sich aber herausgestellt, dass die Implementierung eines Performance-Modus zeitlich nicht umsetzbar ist. Von daher wurde die Fragestellung auf das Sound-Design eingeschränkt, welches auch den einzig sinnvollen Einsatzbereich für das Continuous Feedback darstellt.

Das allgemeine Feedback ergab, dass es eine kleine Personengruppe (n=2) gibt, die generell eher weniger Optionen bei einem Envelope bevorzugen. Das Sustain, eine bessere Darstellung (insbesondere der Zeitleiste) und die Steuerung per Maus (zum Zeichnen von Steps) wurden als noch fehlend empfunden.

Was den Einsatzzweck angeht, so wurden hauptsächlich lange, sich entwickelnde Sounds wie Pads, Ambiance Sounds oder Texturen genannt. Nur einmal wurde dem KF-Envelope auch ein Nutzen bei eher kurzen Sounds (Reproduktion von Attack-Phasen natürlicher Klänge) zugeschrieben. Das Potential für die Gestaltung von extrem schnellen Modulationsverläufen, wie beispielsweise Transienten müsste im Detail aber noch untersucht werden.

#### 4.4 Fazit

Insgesamt kann man sagen, dass das Konzept des Keyframe-Envelopes überwiegend positiv aufgenommen worden ist. Die Kritikpunkte standen überwiegend mit der prototypischen Natur in Verbindung und sind durch Verbesserungen in der Implementierung veränderbar. Dieser iterative Prozess des wiederholten User-Testing in der Entwicklung war durch die eingeschränkten Ressourcen im Rahmen der Masterarbeit limitiert.

Die größte Einschränkung gegenüber dem ADSR-Envelope ist die Möglichkeit, den Klang nicht in Echtzeit durch fehlendes 1-1 Mapping steuern und somit für Live-Performing nutzen zu können. Nur eine Person gab an, prinzipiell lieber mit mehreren Modulatoren arbeiten zu wollen. Gerade Personen, die weniger erfahren mit dem ADSR-Envelope oder Modulation an sich waren, fanden das Konzept aber durch Hilfen wie die Visualisierung einfacher. Die Gewöhnung an den ADSR-Envelope und die noch nicht erreichte Produktreife spielen also eine große Rolle in der Bewertung. Für einen absolut fairen Vergleich müsste der Prototyp weiter fortgeschritten sein, die Probanden deutlich mehr Eingewöhnungszeit gewährt bekommen und neben den sehr simplen getesteten Modulationsverläufen auch die eigentliche Stärke des Keyframe-Envelope einbezogen werden: komplexe Modulation (und diese im nächsten Schritt mehrdimensional), welche andernfalls nur durch Verschalten von Modulatoren oder aktuelle Implementierungen von Time-Level-Envelopes möglich wären.

Die statistische Auswertung des Target-Finding ist aufgrund der erwähnten Einschränkungen des Testumfangs und auch durch die schlechte Quantifizierbarkeit des kreativen Prozesses des Sound-Designs in ihrer Aussagekraft limitiert. Sie zeigt allerdings, dass es trotz sichtbarer Ausreißer prinzipiell in der Qualität der erreichten Genauigkeit der Reproduktion von Sounds keine signifikanten Unterschiede bei den Envelope-Typen gibt. Dies ist in Anbetracht der kurzen Eingewöhnungszeit, der Vertrautheit der Probanden mit dem ADSR-Envelope, den getesteten non-komplexen Modulationen und dem fehlenden Feinschliff eines Prototyps als positiv zu werten. Die gemessenen Zeitspannen zur Erstellung der Klänge macht vor allem deutlich, dass es eine deutliche Lernkurve gibt. Betrachtet man den vergleichsweise sehr simplen und kurzen ersten Sound mit dem Rest, fällt auf, dass der initial große zeitliche Vorsprung des ADSR-Envelopes schnell kleiner wird. Entgegen der Messung wurde der Keyframe-Envelope sogar zweifach als zeitsparend beschrieben.

Letztendlich ist es aber auch eine Frage der persönlichen klanglichen Vorlieben. Komplexe Modulationen sind auch nicht bei allen Sounds notwendig. Lange, sich stetig entwickelnde Klänge wie Flächen- oder Ambiance-Sounds wurden hier als häufigsten Einsatzzweck des Keyframe-Envelopes genannt. Um eine wirkliche Alternative zum ADSR-Envelope in allen Bereichen darstellen zu können, müsste die grundsätzliche Bedienung durch den vorher angesprochenen direkten Parameterzugriff weiter vereinfacht und die Entwicklung des Keyframe-Envelopes als Superset des ADSR-Envelopes konsequent vorangetrieben werden. Dies steht mit der Idee im Konflikt, jeden Schritt prinzipiell als im modularen Sinne gleich zu betrachten. Durch ein verbessertes Mapping und Makros ließe sich dieses Problem aber weiter minimieren. Am Ziel stände ein mindestens genauso einfaches und schnelles Bedienkonzept, welches die erweiterte Funktionalität bei Bedarf anbietet, ohne sie jedoch zu verstecken. Auch wenn es am Ende noch einigen Entwicklungsbedarf gibt, kann man jedoch sagen, dass die Ergebnisse für einen Einsatz des Bedienkonzepts der Schlüsselbildanimation im Sound-Design vielversprechend sind.

#### 4.5 Ausblick

Für die Zukunft wäre eine Veröffentlichung des Prototyps als Open-Source Projekt wünschenswert, was aufgrund von rechtlichen Gründen zum Zeitpunkt der Fertigstellung der Thesis noch nicht möglich ist. Neben der Beseitigung von kleineren Problemen sind es folgende Features, welche als nächstes geplant waren:

- Die Möglichkeit mehrere Parameter über eine Instanz des KF-Envelope zu steuern (also nicht nur 1-1, 1-n, sondern auch n-n Mapping) und zu visualisieren.
- Bipolare Modulation durch den KF-Envelope als weitere Option.
- Die Anwendung des KF-Konzepts auf einen LFO.
- Eine Portierung des Prototyps auf die iOS-Plattform als Touch-Gesteuerter OSC (Open Sound Control) Message Generator ohne eigene Klangerzeugung.

Folgende Funktionen waren bereits vor dem Test des Prototyps geplant, deren Fehlen wurde zudem auch von den Nutzern bestätigt:

- Implementierung unterschiedlicher Sustain-Modi wie: Loop Step, Random Step, Hold Step.
- Die Entwicklung eines Performance-Modus für die Echtzeitmanipulation.
- Optionen zur Kontrolle des Continuous Feedback (inaktiv, Multi-Step ...).

Das Ziel wäre hier, einen kompletten, mindestens gleichwertigen Ersatz für den ADSR-Envelope zu schaffen, der auch für die Performance bzw. die Echtzeitmanipulation ausgelegt ist. Die Implementierung des Konzepts inklusive der multidimensionalen Parametersteuerung würde das eingangs erwähnte Konzept des non-linearen Sound-Design vervollständigen. Mit einem noch weiter ausgereiften Interface und durch die Anbindung an eine höher entwickelte Synthesizer-Engine würde sich das komplette Potenzial zeigen, welches im Rahmen eines Prototyps zur Konzeptvalidierung so noch nicht möglich war.

## 5 Literaturverzeichnis

- Ableton (2017): "Ableton/push-interface." *GitHub*, Zugegriffen am 14.03.2017 von https://github.com/Ableton/push-interface.
- Ableton, Ag (2016): "Ableton Ag." *Ableton Ag*, Zugegriffen am 25.11.2016 von https://www.ableton.com/de/push/.
- Adobe (2017): "Adobe After Effects." Zugegriffen am 14.03.2017 von http://www.adobe.com/de/products/aftereffects.html.
- Allen, William A. (1981): "Applications of Voltage-Controlled Amplifiers." Vortrag auf der Audio Engineering Society Convention 70 Audio Engineering Society.
- Arfib, D.; J. M. Couturier; L. Kessous; et al. (2002): "Strategies of mapping between gesture data and synthesis model parameters using perceptual spaces." In: *Organised Sound*, 7(2), S. 127–144.
- Arner, Nicholas Franklin (2013): *Investigation of the use of Multi-Touch Gestures in Music Interaction*. mscresearch, University of York.
- Bencina, Ross (2005): "The Metasurface: Applying Natural Neighbour Interpolation to Two-to-many Mapping." *Proceedings of the 2005 Conference on New Interfaces for Musical Expression*, Vortrag auf der Singapore, Singapore National University of Singapore, S. 101–104.
- Brandtsegg, Øyvind; Sigurd Saue und Thom Johansen (2011): "A Modulation Matrix for Complex Parameter Sets." *NIME*, S. 316–319.
- Bresin, Roberto; Stefano Delle Monache; Federico Fontana; et al. (2008): "Auditory feedback through continuous control of crumpling sound synthesis." *Proc. CHI workshop on sonic interaction design*, S. 23–28.
- Brotman, Lynne Shapiro und Arun N. Netravali (1988): "Motion Interpolation by Optimal Control." *Proceedings of the 15th Annual Conference on Computer Graphics and Interactive Techniques*, Vortrag auf der New York, NY, USA ACM, (SIGGRAPH '88), S. 309–315.
- Cadoz, Claude und Marcelo M. Wanderley (2000): "Gesture-music." In: *Trends in gestural control of music*,.
- Conan, Simon; Etienne Thoret; Mitsuko Aramaki; et al. (2014): "An Intuitive Synthesizer of Continuous-Interaction Sounds: Rubbing, Scratching, and Rolling." In: *Computer Music Journal*, 38(4), S. 24–37.
- Dobrian, Christopher und Daniel Koppelman (2006): "The E in NIME: Musical Expression with New Computer Interfaces." Vortrag auf der NIME.
- von Falkenstein, Jan T. (2011): "Gliss: An Intuitive Sequencer for the iPhone and iPad." *NIME*, S. 527–528.
- Faria, Regis Rossi Alves (2011): "Profiling New Paradigms in Sound and Music Technologies." In: *Journal of New Music Research*, 40(3), S. 191–204.
- Gelineck, Steven und Stefania Serafin (2009): "A Quantitative Evaluation of the Differences between Knobs and Sliders." *NIME*, S. 13–18.
- Gómez, Daniel; Tjebbe Donner und Andrés Posada (2007): "A Look at the Design and Creation of a Graphically Controlled Digital Musical Instrument." *Proceedings of the 7th International Conference on New Interfaces for Musical Expression*, Vortrag auf der New York, NY, USA ACM, (NIME '07), S. 327–329.
- Goudeseune, Camille (2002): "Interpolated mappings for musical instruments." In: *Organised Sound*, 7(2), S. 85–96.
- Hein, Ethan (2013): A constructivist iOS rhythm tutorial system for beginners. NYU.
- Hunt, Andy und Marcelo M. Wanderley (2002): "Mapping performer parameters to synthesis engines." In: *Organised Sound*, 7(2), S. 97–108.

- Hunt, Andy; Marcelo M. Wanderley und Matthew Paradis (2003): "The importance of parameter mapping in electronic instrument design." In: *Journal of New Music Research*, 32(4), S. 429–440.
- Jones, Mam Riess (1962): "Color Coding." In: Human Factors, 4(6), S. 355–365.
- Jordà, Sergi (2002): "FMOL: Toward user-friendly, sophisticated new musical instruments." In: *Computer Music Journal*, 26(3), S. 23–39.
- Lamb, Roland und Andrew Robertson (2011): "Seaboard: a New Piano Keyboard-related Interface Combining Discrete and Continuous Control." *NIME*, S. 503–506.
- Lasseter, John (1987): "Principles of traditional animation applied to 3D computer animation." *ACM Siggraph Computer Graphics*, ACM, S. 35–44.
- Lerch, Alexander (2012): An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics. Hoboken, NJ, USA: John Wiley & Sons, Inc.
- libusb, website (2017): "libusb." Zugegriffen am 14.03.2017 von http://libusb.info/.
- Loy, R. und J. Chowning (2006): "Musimathics: The Mathematical Foundations of Music, Volumes I." In: *AMC*, 10, S. 12.
- McPherson, Andrew (2015): "Buttons, Handles, and Keys: Advances in Continuous-Control Keyboard Instruments." In: *Computer Music Journal*, 39(2), S. 28–46.
- Möller, Sebastian; Benjamin Weiss; Ina Wechsung; et al. (2010): "Qualitätsintegration bei multimodaler Interaktion." Zugegriffen am 25.11.2016 von http://www.qu.tu-berlin.de/fileadmin/fg41/publications/moeller\_2010\_qualitaetsintegration-beimultimodaler-interaktion.pdf.
- Momeni, Ali und David Wessel (2003): "Characterizing and controlling musical material intuitively with geometric models." *Proceedings of the 2003 conference on New interfaces for musical expression*, National University of Singapore, S. 54–62.
- Moog, Robert A. (1964): "Voltage-controlled electronic music modules." *Audio Engineering Society Convention 16*, Audio Engineering Society.
- Native Instruments (2017): "Native Instruments." Zugegriffen am 26.03.2017 von http://www.native-instruments.com/products/.
- Novation (2017): "Launch Control | NovationMusic.com." Zugegriffen am 02.04.2017 von https://novationmusic.de/launch/launch-control#.
- Olowe, Ireti; Giulio Moro; Mathieu Barthet; et al. (2016): "residUUm: user mapping and performance strategies for multilayered live audiovisual generation." In.:
- Pendharkar, Chinmay; Michael Gurevich und Lonce Wyse (2006): "Parameterized morphing as a mapping technique for sound synthesis." *Proc. of the Int. Conf. on Digital Audio Effects (DAFx-06), Montreal, Quebec, Canada*, Citeseer, S. 45–48.
- Potidis, Spyridon und Thomas Spyrou (2014): "Spyractable: A Tangible User Interface Modular Synthesizer." In: Masaaki Kurosu (Hrsg.) *Human-Computer Interaction. Advanced Interaction Modalities and Techniques*, Springer International Publishing, (Lecture Notes in Computer Science), S. 600–611.
- reaper.fm (2017): "REAPER | Audio Production Without Limits." Zugegriffen am 02.04.2017 von http://www.reaper.fm/.
- Rey, Denise und Markus Neuhäuser (2011): "Wilcoxon-Signed-Rank Test." In: Miodrag Lovric (Hrsg.) *International Encyclopedia of Statistical Science*, Springer Berlin Heidelberg, S. 1658–1659.
- Richardson, Jim (2015): "Animateducated.".
- Rocchesso, Davide; Federico Avanzini; Matthias Rath; et al. (2004): "Contact sounds for continuous feedback." *Proceedings of the International Workshop on Interactive Sonification*, S. 1–6.
- ROLI Ltd (2017): "JUCE." JUCE, Zugegriffen am 27.03.2017 von https://www.juce.com/.
- Seifert, Uwe und Jin Hyun Kim (2016): "Towards a Conceptual Framework and an Empirical Methodology in Research on Artistic Human-Computer and Human-Robot

- Interaction." Zugegriffen am 04.08.2016 von http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.440.3197&rep=rep1&type=pdf#page=59.
- Serra, Xavier und others (1997): "Musical sound modeling with sinusoids plus noise." In: *Musical signal processing*, S. 91–122.
- Synister (2016): "Synister | The Synthesizer made for you." Zugegriffen am 27.03.2017 von https://the-synister.github.io/.
- Tubb, Robert H. (2016): Creativity, Exploration and Control in Musical Parameter Spaces. Queen Mary University of London.
- Tubb, Robert; Anssi Klapuri und Simon Dixon (2012): "The Wablet: Scanned Synthesis on a Multi-Touch Interface dafx12 submission 18.pdf." Vortrag auf der DAFx.
- User Interface Design GmbH (2016): "AttrakDiff." *AttrakDiff*, Zugegriffen am 07.03.2017 von http://attrakdiff.de/sience.html.
- Van Nort, Doug; Marcelo M. Wanderley und Philippe Depalle (2004): "On the choice of mappings based on geometric properties." *Proceedings of the 2004 conference on New interfaces for musical expression*, National University of Singapore, S. 87–91.
- Stefan Weinzierl (Hrsg.) (2008): *Handbuch der Audiotechnik*. Berlin, Heidelberg, New York: Springer.
- White, Tony (2006): Animation from Pencils to Pixels: Classical Techniques for Digital Animators. Taylor & Francis.
- Wikimedia CC, https://commons.wikimedia.org/wiki/User:Abdull (2017): "ADSR." *Wikimedia*, Zugegriffen am 26.03.2017 von https://commons.wikimedia.org/wiki/File:ADSR parameter.svg.
- Yamaha (1983): "DX7 Manual." Zugegriffen am 14.03.2017 von https://de.yamaha.com/files/download/other assets/9/333979/DX7E1.pdf.
- Zadel, Mark und Gary Scavone (2006): "Different Strokes: A Prototype Software System for Laptop Performance and Improvisation." *Proceedings of the 2006 Conference on New Interfaces for Musical Expression*, Vortrag auf der Paris, France, France IRCAM Centre Pompidou, S. 168–171.
- Zappi, Victor; Andrea Brogni und Darwin G. Caldwell (2010): "OSC Virtual Controller." *NIME*, S. 297–302.

# 6 Anhang

Auf der CD-Rom finden sich folgende Inhalte: Dokumentation der Funktionen (Auszug), MatLab-Scripte für die Datenanalyse, Rohdaten der Umfragen, Sounds Target-Finding, Zotero Quellensammlung