Entwicklung eines virtuellen Musikinstruments auf Basis der Impulse Pattern Formulation

Bachelor-Thesis

zur Erlangung des akademischen Grades B.Sc.

Nadja Rutsch

[redacted]

Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Design, Medien und Information
Department Medientechnik

Erstprüfer: Prof. Dr. Robert Mores

Zweitprüfer: Simon Linke

Hamburg, 28.02.2019

Inhaltsverzeichnis

Abstract

The Impulse Pattern Formulation by Bader (2013) is a systematic approach for the description of musical instruments. This approach is used as basis for the realisation of a virtual musical instrument. In order to use the IPF as basis of the instrument, its meaning in terms of sound design parameters must be formulated. This is done by looking at sound generation in the violin.

The formulated sound design parameters should be controlled by the IPF. For this purpose a program in the development enviroment Max/MSP is built. The program can be used as a virtual musical instrument. An Artificial Neural Network is implemented as part of the instrument, in order to raise its playability.

The IPF is able to design overtones, noise components and transients of sound. The virtual musical instrument can be described as real-time capable. It can be used as basis for further developments.

Zusammenfassung

Die Impulse Pattern Formulation ist ein von Bader (2013) vorgestellter Ansatz zur systematischen Beschreibung eines Musikinstruments. Dieser systematische Ansatz wird als Basis in einem virtuellen Musikinstrument umgesetzt. Eine Interpretation des Ansatzes erfolgt am Beispiel der Geige.

Aus der Interpretation ergibt sich, wie Klangparameter in Abhängigkeit von der IPF gesteuert werden können. Auf dieser Grundlage wird ein Programm in der Entwicklungsumgebung Max/MSP erstellt, das als virtuelles Musikinstrument funktioniert. Teil des Instruments ist ein Künstliches Neuronales Netzwerk, das für eine verbesserte Spielbarkeit sorgt.

Die IPF leistet als Synthesebasis eine Gestaltung von Obertönen, Rauschanteilen und Einschwingverhalten von Klängen. Das virtuelle Musikinstrument ist echtzeitfähig und kann als Basis für weitere Entwicklungen verwendet werden.

1 Einleitung

Mit Beginn des digitalen Zeitalters ist auch die analoge Klangerzeugung ins Digitale überführt worden. Mit digitaler Klangerzeugung wird oftmals ein steriler Charakter des Klangs assoziiert, da jeder Prozess, der vorher von Transistoren, Kondensatoren und Widerständen mit jeweils individuellen Toleranzen ausgeführt wurde, in der digitalen Welt mit höchster Präzision nachmodelliert werden kann. Hierbei gehen diese Toleranzen der einzelnen Bauteile verloren und werden durch ein perfektes System ersetzt, dessen Verhalten vollständig vorhersagbar ist (Vail 2014: S. 234).

Wird durch einen digitalen Prozess versucht, die mechanische Klangerzeugung eines Musikinstruments nachzubilden, ergibt sich ein weiteres Sterilitätsproblem, dem bereits analoge Synthesizer unterliegen. Bei der Modellierung von Musikinstrumenten wird meist von einer optimalen Anregung des Schwingungssystems ausgegangen, die sich in der realen Welt derart nicht wiederfindet. So gehört zum Klang einer Geige nicht nur der perfekt gestrichene Ton, sondern gleichermaßen das Kratzen und Quietschen, das entsteht, wenn etwa ein Amateur die Geige spielt (Roads u. Strawn 1996: S. 266).

Der Versuch, solche Imperfektionen und Zufälligkeiten in der Klangentstehung nachzubilden, ist oftmals Bestandteil der Entwicklung eines virtuellen Musikinstruments. Klassische Lösungsansätze hierfür sind z.B. das Hinzufügen von Rauschen oder Noise (Serra u. Smith 1990: S. 12-13). In der vorliegenden Arbeit wurde ein Ansatz als Basis in einem virtuellen Musikinstruments realisiert, der für die Modellierung von nicht-optimaler Anregung eine Lösung liefert. Es handelt sich bei diesem Ansatz um die von Rolf Bader vorgestellte „Impulse Pattern Formulation“.

Die Impulse Pattern Formulation (IPF) leistet nicht nur eine Modellierung der Anforderungen, die ein Instrument an seine Anregung stellt, um einen stabilen Ton zu erzeugen. Sie eignet sich darüber hinaus noch dazu, das Einschwingverhalten vor dem Zustandekommen eines stabilen Tons nachzustellen. Dieses Einschwingen trägt einen Großteil zum Klang bei, der für ein Musikinstrument jeweils als charakteristisch

empfunden wird.

Mit der Beschreibung von Einschwingverhalten und instabilen Systemzuständen bietet die IPF ein interessantes Potential zur Modellierung von reellen Musikinstrumenten. Demnach ist ein Ziel der Verwendung der IPF im virtuellen Musikinstrument eine Annäherung an das systematische Verhalten reeller Musikinstrumente. Allerdings ist der große Vorteil eines virtuellen Musikinstruments gegenüber einem reellen Musikinstrument, dass es keinen physikalischen Limitationen unterliegt. Daher sollen auch weitere Möglichkeiten zur Klangformung erkundet werden, die die IPF im erweiterten Sinne liefern kann.

In der vorliegenden Arbeit wird ein kurzer Überblick über verbreitete Syntheseverfahren gegeben und die IPF als systematischer Ansatz für ein neues Syntheseverfahren vorgestellt. Auf Grundlage des Leistungsumfangs der IPF und vor dem Hintergrund anderer Syntheseverfahren wird ein Anforderungsprofil des virtuellen Musikinstruments erstellt. Zur Erreichung dieses Profils wird ein passendes Umsetzungskonzept erarbeitet. Dieses Konzept beinhaltet die Integration eines Künstlichen Neuronalen Netzwerkes, das einer Steigerung der Spielbarkeit des virtuellen Musikinstruments dient. Die Umsetzung des Instruments in der Entwicklungsumgebung Max/MSP, unter Einbeziehung des darin integrierten Neuronalen Netzwerks, wird im Hauptteil der Arbeit skizziert, und die erzielten Ergebnisse schließlich mit dem Anforderungsprofil abgeglichen.

2 Theoretische Grundlagen

2.1 Klangsynthese-Verfahren

Eine Klangsynthese auf Basis der Impulse Pattern Formulation bietet einige Vorteile gegenüber herkömmlichen Verfahren der Klangsynthese. Im Folgenden wird ein kurzer Überblick über verbreitete Klangsynthese-Verfahren gegeben.

Wavetable-Synthese

Die Wavetable-Synthese macht sich die repititiven Eigenschaften von Schallwellen in der Musik zunutze, indem sie lediglich die Amplitudenwerte für eine einzelne Schwingungsperiode in einer Liste speichert. Eine solche Liste wird „Wavetable“ genannt. Um eine periodische Schwingung zu erhalten, wird diese Liste in einer Schleife immer wieder ausgelesen. Diese Form der Synthese ist der Kern in digitalen sowie manchen analogen Oszillatoren (Roads u. Strawn 1996: S. 90).

Additive Synthese

Die additive Synthese arbeitet auf Basis der Erkenntnis von Jean Baptiste Joseph Fourier, dass sich jede periodische Schwingung als Summe von verschiedenen Sinusschwingungen ausdrücken lässt. Die Obertöne, die zur Erzielung eines gewünschten Klangs beitragen, lassen sich also durch eine Addition von Sinusschwingungen erzeugen. Eine weitere Formung des Klanges ergibt sich durch die Anwendung von Hüllkurven auf die jeweiligen Obertöne. Die additive Synthese ermöglicht somit eine präzise Gestaltung des Klangs, allein durch das Hinzufügen weniger Obertöne (Ruschkowski 1998: S. 305-307).

Synthese durch Frequenzmodulation

Die Obertöne, die bei der additiven Synthese durch Addition erzeugt werden, entstehen bei der FM-Synthese durch eine Modulation der Grundfrequenz. Die Grundfrequenz wird durch die Trägerfrequenz der Modulation gegeben. Die Obertöne entstehen schließlich im unteren und oberen Seitenband des frequenzmodulierten Trägersignals. Eine vielfältige Formung des gewünschten Klanges geschieht allein durch die Wahl von Verhältnis der Träger- zur Modulationsfrequenz sowie der Modulationsstärke. Eine Verkettung von Modulatoren bietet erweiterte Möglichkeiten der Klangformung. Die Reduzierung der klangformenden Parameter auf Frequenzverhältnis und Modulationsstärke erleichtert die Handhabung der Klanggestaltung enorm, bedeutet allerdings gegenüber der additiven Synthese auch ein Verlust an Gestaltungspräzision (Ruschkowski 1998: S. 308-318).

Synthese durch Amplitudenmodulation

Durch eine Kombination von Oszillatoren und Amplituden-Modulatoren lassen sich vielfältige Klänge synthetisieren. Je nach Kombination können z.B. Tremolo-Effekte, Klangfärbungen stationärer Töne oder Hüllkurven erzeugt werden. Das Ergebnis ist hierbei nicht nur vom Modulationssignal und dessen Verhältnis zur Grundfrequenz abhängig, sondern wird auch wesentlich bestimmt von der Anordnung der oszillierenden und modulierenden Elemente. Durch eine entsprechende Planung der zu kombinierenden Elemente ergibt sich eine schnell zu erreichende Vielfalt in der Klanggestaltung (Ruschkowski 1998: S. 319-322).

Waveshaping-Synthese

Bei der Waveshaping-Synthese wird ein Eingangssignal mit Obertönen angereichert, indem sein Verlauf durch eine nichtlineare Übertragungsfunktion umgeformt wird. Die Klangformung geschieht hier also durch die Formung der Übertragungsfunktion. Sie ist allerdings in ihrer Reinform auf eine Erweiterung um harmonische Obertöne begrenzt. Bei komplexen Kombinationen von Eingangssignal und Übertragungsfunktion verringert sich die Handhabbarkeit der Waveshaping-Synthese, da eine Einschätzung des Effekts und somit eine Umsetzung der gewünschten Klangformung schwierig wird (Ruschkowski 1998: S. 323-327).

Granularsynthese

In der Granularsynthese entsteht ein Klang aus der Zusammensetzung einzelner kurzer Impulse. Zur Klanggestaltung können die einzelnen Impulse in Inhalt, Frequenz, Amplitude und Reihenfolge variiert werden. Als inhaltliche Basis solcher Impulse kann auch ein bereits aufgenommener Klang dienen, der dann im Rahmen einer Granulierung in Klangimpulse zerlegt wird. Werden diese Klangimpulse dann in der Granularsynthese wiedergegeben, lässt sich neben dem ursprünglichen Klang der Aufnahme auch eine Vielzahl weiterer Klänge generieren. Um ein kontinuierliches Klangerlebnis zu schaffen, sollten die Klangimpulse etwa eine Länge von 10 ms bis 60 ms haben (Ruschkowski 1998: S. 327-332).

Subtraktive Synthese

Als Grundlage der subtraktiven Synthese wird üblicherweise ein Oberton-reicher Klang verwendet, wie z.B. eine Sägezahn- oder Rechteckschwingung. Zur Gestaltung des Klangs werden die unerwünschten Obertöne entsprechend gedämpft. Hierzu können diverse Filter verwendet werden. Der Klang kann also durch die Wahl des Rohmaterials sowie der klangverändernden Elemente bestimmt werden (Roads u. Strawn 1996: S. 184-185).

Physical Modeling

Das Physical Modeling ist eine Syntheseart, die sich darauf spezialisiert, reelle physikalische Prozesse bei der Klangentstehung durch mathematische Prozesse im Computer nachzubilden. Hierbei soll nicht nur das Klangergebnis möglichst nah am Klang des reellen Musikinstruments sein, sondern auch die einzelnen Komponenten, die zur Entstehung dieses Klangs führen, mathematisch beschrieben und simuliert werden. Der zu erzielende Klang soll hierbei von Grund auf neu synthetisiert werden, also gänzlich ohne Verwertung aufgenommener Samples (Ruschkowski 1998: S. 345-349).

2.2 Die Impulse Pattern Formulation

Die von Bader (2013: S. 285-315) vorgestellte Impulse Pattern Formulation beschreibt die Anregung eines Musikinstruments durch Impulse. Hierbei wird allgemein angenommen, dass ein Musikinstrument ein System ist, das auf sich selbst einwirkt. Das

System wird von externen Impulsen, wie z.B. dem Zupfen einer Saite, zum Schwingen angeregt. Diese Impulse werden durch den Klangkörper getragen, bis sie schließlich an den Entstehungsort rückgekoppelt werden. Die Entstehung von Impulsmustern wird in einem solchen System also sowohl durch die externe Impuls-Anregung bestimmt, als auch von der internen verzögerten Rückwirkung ebendieser Impulse.

Nach Bader (2013: S. 287) lässt sich das System vereinfacht zunächst durch folgende Formel beschreiben:

$$g_+ = g - ln\left(\frac{g}{\alpha}\right) \tag{2.1}$$

Hierbei bezeichnet g_+ den Folgezustand des Systems, d.h. zeitlich einen Impuls später als g (Bader 2013: S. 288). α bezeichnet einen Parameter der externen Anregung, wie etwa die Stärke des Bogendrucks bei einer gestrichenen Saite. Die Stärke der Rückwirkung von g auf den Folgezustand g_+ unterliegt einer Dämpfung. Um eine Nähe zur reellen Physik von Musikinstrumenten zu erhalten, wird für die Rückwirkung eine exponentielle Dämpfung angenommen (Bader 2013: S. 286).

In Abhängigkeit von α und dem vorherigen Systemzustand g kann sich aus der IPF sowohl stabiles als auch chaotisches Verhalten ergeben. In Abbildung 2.1 ist der Systemzustand g in Abhängigkeit von $\frac{1}{\alpha}$ aufgetragen. Für kleine $\frac{1}{\alpha}$ ist ein stabiler Bereich sichtbar, bei dem der Systemzustand einer einzelnen Kurve folgt. Für größere Werte von $\frac{1}{\alpha}$ treten Bifurkationen im System auf, d.h. das System springt zwischen zwei oder mehr Zuständen hin und her. Der Bereich der Bifurkationen geht schließlich für noch größere Werte von $\frac{1}{\alpha}$ in einen chaotischen Bereich über, bei dem g bei jedem Iterationsschritt sprunghaft einen neuen Wert annimmt, ohne dass sich die Systemzustände wiederholen.

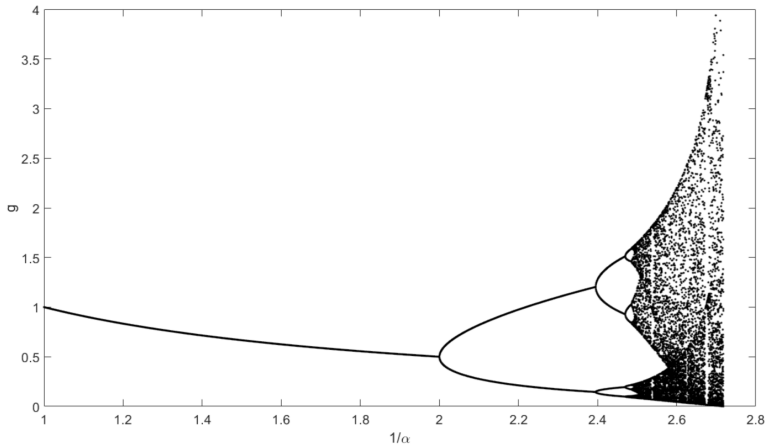

Abb. 2.1: Verhalten des Systemzustands g in Abhängigkeit von $\frac{1}{\alpha}$ bei einem Startwert von $g_0 = 1$. Für niedrige Werte von $\frac{1}{\alpha}$ ergibt sich ein stabiles Verhalten, bei höheren Werten treten Bifurkationen auf. Quelle: Linke u. a. (2019: S. 3)

Ein stabiler Systemzustand ergibt sich aus der IPF also nur unter bestimmten Anforderungen an α. Bei kleinen Werten von α ist die gedämpfte Rückwirkung der Impulse nicht stark genug, um einen stabilen Systemzustand zu bewirken (Bader 2013: S. 290). Es ergeben sich Bifurkationen erster oder höherer Ordnung, bis die Werte für g ab einem bestimmten Wert von α schließlich komplex werden und im Realteil divergieren (Linke u. a. 2019: S. 3-4).

Formel 2.1 beschreibt ein System, bei dem nur ein Impuls pro Periode auf das System rückwirkt. Es wird also von einem einzigen Reflektionspunkt ausgegangen, der eine Rückwirkung von Impulsen bewirkt. Die IPF kann nach Bader (2013: S. 290-293) auch Systeme mit beliebig vielen Reflektionspunkten beschreiben. Pro Reflektionspunkt wird die Formel demnach um eine gedämpfte Rückwirkung auf das System erweitert. Diese mehrfache Rückwirkung wird mathematisch wie folgt beschrieben (Bader 2013: S. 291):

$$g_+ = g - ln\left(\frac{g}{\alpha} - \sum_{k=1}^{n} \frac{\beta_k}{\alpha} \cdot e^{g-g_{-k}}\right) \tag{2.2}$$

In Formel 2.2 bezeichnet β_k den k-ten Reflektionskoeffizienten, d.h. die Stärke der Rückwirkung des um k Schritte zurückliegenden Impulses. Je weiter der Punkt, an dem der Impuls reflektiert wird, vom betrachteten System zurückliegt, desto länger benötigt der Impuls für den Weg seiner Rückkopplung und desto mehr Dämpfung erfährt dieser demnach (Bader 2013: S. 291). Die Rückwirkung auf den Folgezustand

und somit auch der Koeffizient des Reflektionspunktes richtet sich also nach zeitlichem und örtlichem Verzug zwischen dem betrachteten System und dem jeweiligen Reflektionspunkt. Für die Reflektionskoeffizienten gilt daher die Bedingung:

$$\alpha > \beta_1 > \beta_2 > ... > \beta_k \tag{2.3}$$

Des Weiteren argumentiert Bader (2013: S. 291), dass die Rückwirkung eines Impulses niemals seine ursprüngliche Stärke erreichen wird, und leitet daraus folgende Bedingung ab:

$$\alpha + \sum_k \beta_k < 1 \tag{2.4}$$

Je nach Wahl von α und den Reflektionskoeffizienten, ergeben sich unterschiedliche Regionen für stabiles, chaotisches und divergierendes Verhalten der IPF. Nach Bader (2013: S. 293) führt eine Erhöhung der Anzahl schwacher Impulsrückwirkungen zu einer Erhöhung der Stabilität des Systems.

Linke u. a. (2019: S. 11) konnten anhand von numerischen Simulationen zeigen, dass die Bereiche stabiler und instabiler Systemzustände nicht nur von der Höhe und Anzahl der Reflektionskoeffizienten abhängen, sondern auch durch den Anfangszustand g_0 des Systems bestimmt werden. Die Wahl der Reflektionskoeffzienten beeinflusst in jedem Fall das Verhältnis, in dem die verschiedenen Bereiche auftreten. So wird bei Erhöhung von β_1 der Bereich von α, in dem sich chaotisches Verhalten ergibt, verschwindend gering. Bei entsprechenden Reflektionskoeffizienten ist bei einer Änderung von auch ein direkter Übergang vom stabilen ins divergierende Verhalten möglich, ohne dass Bifurkationen im reellen Zahlenbereich auftreten (Bader 2013: S. 293).

Einschwingverhalten

In Abhängigkeit von α, g_0 und β_k ergibt sich aus der IPF ein bestimmtes Einschwingverhalten des Systems. Je höher der Abstand zwischen dem Startwert g_0 und dem Fixpunkt g_s ist, gegen den das System im stabilen Bereich konvergiert, desto mehr Schritte braucht das System, um sich dem Fixpunkt g_s zu nähern. Der Fixpunkt ist nach Linke u. a. (2019: S. 7) wie folgt definiert:

$$g_s = \alpha + \sum_{k=1}^{n} \beta_k \tag{2.5}$$

Die IPF ist also in der Lage, das Einschwingverhalten eines Systems anhand von einigen wenigen Parametern zu beschreiben. In der von Bader (2013: S. 295) übernommenen Abbildung 2.2 ist dargestellt, wie ein solches Einschwingverhalten von g aussehen kann, wenn ein Startwert von $g_0 = 1$ gegeben ist und α konstant gehalten wird. Das System konvergiert gegen einen Fixpunkt von $g_s = 0,5$.

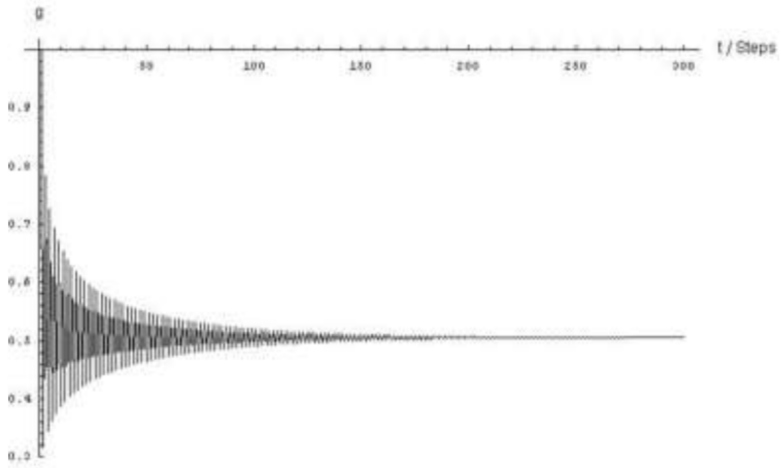

Abb. 2.2: Einschwingverhalten von g innerhalb von 300 Iterationen der IPF bei einer konstanten Anregungsstärke von $\alpha = 0,5$ und einem Startwert von $g_0 = 1$. Quelle: Bader (2013: S. 295)

Damit aus der Beschreibung des reinen Impulsmusters auch eine Beschreibung des Klangs resultieren kann, muss eine Impulsform in das Muster injiziert werden. Das der Klangentstehung zugrunde liegende Impulsmuster besteht jedoch unabhängig von der Impulsform (Bader 2013: S. 288). Die IPF kann daher als Top-Down Ansatz verstanden werden, der für alle Instrumente gültig ist. Die Wahl einer konkreten Impulsform stellt dann eine Spezifikation in Richtung eines bestimmten Instruments dar.

2.3 Interpretation der IPF am Beispiel der Geige

Die IPF leistet eine abstrakte Modellierung des Systemzustands eines Musikinstruments. Um diese abstrakte Formulierung in konkrete Parameter zur Klanggestaltung im Synthesizer umsetzen zu können, ist eine physikalische Interpretation der IPF nötig.

Bader (2013: S. 286-287) spricht im Allgemeinen bei seinen Ausführungen zur IPF über α als Parameter der Anregungsstärke. In seinem Anwendungsbeispiel, in dem er die IPF zur Modellierung der Physik einer Geige verwendet, bezeichnet er α als Stärke des Bogendrucks (Bader 2013: S. 305). Diese Definition findet insofern Anschluss an die reelle Physik, als für das Zustandekommen einer Helmholtz-Bewegung beim Geigenspiel ein bestimmter minimaler Bogendruck vorausgesetzt ist. Eine Helmholtz-Bewegung ist die Auslenkung einer gestrichenen Saite, die sich bei einem exakt periodischen Wechsel von Haften und Gleiten der Saite am Bogen ergibt (Cremer 1981: S. 45). Durch die Abwechslung von Haften und Gleiten entsteht an der Saite eine Sägezahn-ähnliche Impulsfolge.

Liegt der Bogendruck unterhalb des Minimums, kann keine Helmholtz-Bewegung mehr etabliert werden und es kommt zu zusätzlichen „Gleitphasen während eines Impulsumlaufs“ (Mores 2014: S. 337). Die Auslenkung einer Geigensaite bei Auftreten solcher zusätzlichen Gleitphasen ist in Abbildung 2.3 zu sehen.

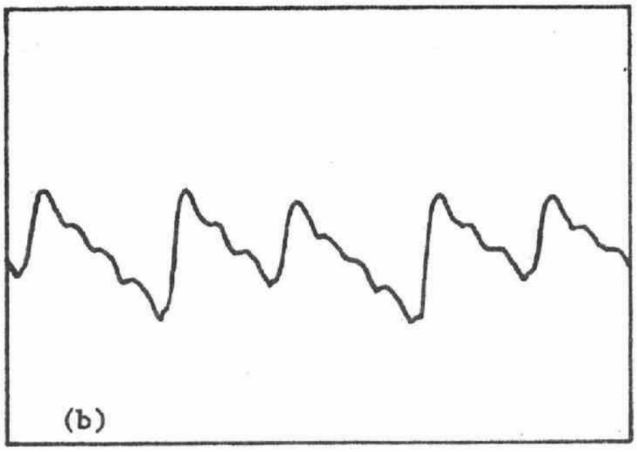

Abb. 2.3: Zeitlicher Verlauf der Kraftwirkung einer G-Saite am Steg einer Geige. Der Mindest-Bogendruck zur Etablierung einer Helmholtz-Bewegung ist nicht gegeben, es kommt zu zusätzlichen Gleitphasen. Quelle: McIntyre u. Woodhouse (1984: S. 20)

Das Verhalten einer gestrichenen Saite bei Nichterreichen des Mindestbogendrucks kann als Hilfe zur Interpretation des Systemzustands g der IPF dienen. Bader (2013: S. 286) spricht von einem Zusammenhang zwischen g und der Amplitude sowie der Periodizität der Schwingung, ohne diesen Zusammenhang zu konkretisieren. Linke u. a. (2019: S. 16) schlagen vor, g als Modulationsfaktor für die Amplitude und Δg für die Frequenz oder Phase der Impulse zu verwenden.

Um bei Bifurkationen in der IPF eine Klangveränderung zu erhalten, die an die zusätzlichen Gleitphasen bei Nichterreichen des Helmholtz-Regimes angelehnt ist, scheint es geeignet, mithilfe von Δg die Phase zu modulieren. Bei einer Bifurkation erster Ordnung treten dann zwei zueinander phasenverschobene Impulse pro Periode der Grundfrequenz auf.

Der Schwingungsverlauf bei einer phasenverschobenen Überlagerung von zwei Sägezahn-Schwingungen ist in 2.4 dargestellt. Es ergibt sich, unter Nichtberücksichtigung der Auslenkungsrichtung, ein ähnlicher Verlauf wie in Abbildung 2.3. Die Phasenverschiebung der Sägezahn-Schwingungen in Abbildung 2.4 beträgt 144°. Ein solcher Wert kann als Orientierung für die Skalierung von Δg dienen.

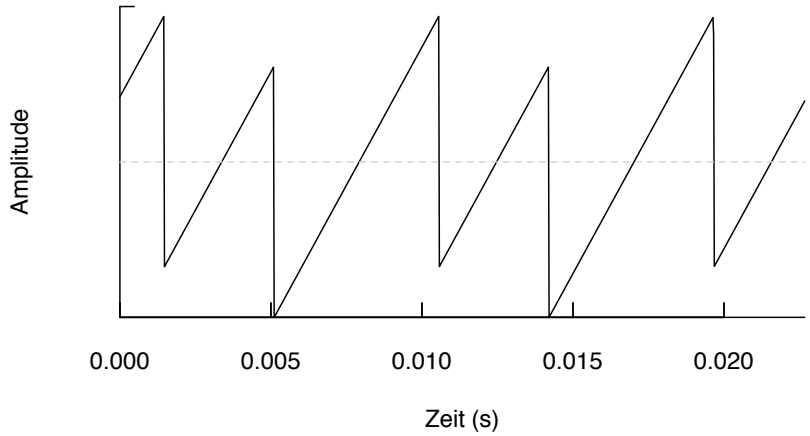

Abb. 2.4: Normalisierter zeitlicher Verlauf der Überlagerung zweier Sägezahn-Schwingungen gleicher Amplitude mit einer Phasenverschiebung von 144°.

Bei einer Interpretation von Δg als Phasenverschiebung ergeben sich die folgenden Klangergebnisse für die unterschiedlichen Systemzustände:

1. g ist stabil: Ein stabiler Ton in der Grundfrequenz entsteht.
2. g springt zwischen zwei oder mehr Zuständen: Es entstehen zusätzliche Obertöne.
3. g ist chaotisch: Der Ton ist instabil, es entsteht Noise.
4. g divergiert: Es kommt kein Ton zustande.

Die Interpretation von imaginären und divergierenden Zuständen von g als Nichtzustandekommen eines Tons wurde von Bader übernommen (Bader 2013: S. 293).

Dämpfung

Damit die Interpretation von Δg als Phasenverschiebung im Fall von Bifurkationen zu einer Überlagerung zweier oder mehr Schwingungen und somit zu einer Entstehung von Obertönen führt, müssen sich vorausgegangene Werte von Δg noch im System befinden. Es wird daher davon ausgegangen, dass jeder Systemzustand nach seinem initialen Auftreten einer Dämpfung unterliegt und im Rahmen dieser im System erhalten bleibt. Die Beschreibung der IPF von Bader umfasst keine Konkretisierung einer solchen Dämpfung. Um die aufgelisteten Klangergebnisse zu erzielen, ist die Implementierung einer Dämpfung dennoch nötig.

3 Anforderungsprofil

Die grundsätzliche Aufgabe des virtuellen Musikinstruments ist es, die IPF als zentrales Synthese-Element zu verwenden. Es muss also möglich sein, die Kontrollparameter der IPF zu steuern. Die Werte, die sich aus Berechnung der IPF ergeben, müssen zur Klangformung verwendet werden.

Die Vor- und Nachteile der in Kapitel 2.1 vorgestellten herkömmlichen Syntheseverfahren ergeben sich zumeist aus der Handhabbarkeit der Klangformung sowie der Vielfältigkeit und Präzision möglicher Klangergebnisse. Eine einfache Handhabbarkeit geht oft mit einem Verlust von Vielfältigkeit oder Präzision einher. Um die IPF optimal als Synthese-Basis nutzen zu können, sollen die im Folgenden aufgeführten Anforderungen an Funktionsumfang und Nutzerfreundlichkeit gestellt werden. Des Weiteren soll das virtuelle Musikinstrument echtzeitfähig und erweiterbar sein.

3.1 Funktionsumfang

Die Klanggestaltung durch die IPF soll möglichst vielfältig sein. Möglichkeiten zur Klangformung ergeben sich in der IPF durch Wahl der Impulsform, der Reflektionskoeffizienten sowie durch die Anregungsstärke α. Eine Anwendung von Hüllkurven, um z.B. den Einschwingvorgang eines Tons zu beschreiben, ist nicht nötig. Das Einschwingverhalten wird bereits durch die IPF vorgegeben.

Die IPF gilt für eine beliebig hohe Anzahl von Reflektionskoeffizienten. Das virtuelle Musikinstrument soll zunächst die Verwendung von bis zu zwei solcher Reflektionskoeffizienten ermöglichen. Eine Verwendung von zwei Reflektionskoeffizienten würde bedeuten, dass in der IPF von drei aneinander gekoppelten Untersystemen ausgegangen wird. Als klangerzeugende Komponente soll lediglich ein Untersystem betrachtet und durch die IPF berechnet werden.

Der Synthesizer soll mit einem MIDI-Controller gespielt werden können. Demnach müssen die Kontrollparameter der IPF sowie die Tonhöhe durch MIDI-Befehle

steuerbar sein. Die im MIDI-Signal enthaltene Anschlagsstärke soll als Gate für die Tonsynthese dienen.

3.2 Nutzerfreundlichkeit

Ein großer Vorteil der IPF ist, dass die Kontrolle von wenigen Parametern bereits eine Klangformung zulässt. Das virtuelle Musikinstrument soll sich somit durch leichte Handhabbarkeit und hohe Nutzerfreundlichkeit auszeichnen. Als potentielle Nutzer des virtuellen Musikinstruments werden Musiker/-innen und Musikproduzent/-innen angenommen. Die Handhabung des Synthesizers soll demnach auf diese Nutzergruppe ausgerichtet sein.

Als richtungsweisend für die Erreichung einer hohen Nutzerfreundlichkeit können die „Grundsätze der Dialoggestaltung“ gemäß DIN EN ISO 9241-110 dienen (Deutsches Institut für Normung 2006). Die Grundsätze bestehen u.a. aus Aufgabenangemessenheit, Erwartungskonformität, Individualisierbarkeit und Fehlertoleranz des Programms.

Im Sinne der Aufgabenangemessenheit soll der Synthesizer eine Bedienoberfläche enthalten, deren Elemente auf ein für die Praxis nötiges Minimum reduziert sind. Unnötige Interaktionen sollen auf dieser Oberfläche vermieden werden. Um den Nutzer zu entlasten, sollen zudem sinnvolle Startparameter voreingestellt sein (Deutsches Institut für Normung 2006).

Damit der Synthesizer sich möglichst erwartungskonform verhält, soll die Bedienoberfläche mit Elementen arbeiten, die an allgemein anerkannte Konventionen zur Kontrolle eines Synthesizers angelehnt sind. Musiker/-innen und Produzent/-innen müssen die Steuerung des Synthesizers somit nicht von Grund auf neu erlernen, sondern können auf bereits gelernte Umgangsformen zurückgreifen (Deutsches Institut für Normung 2006). Der Nutzer soll zudem visuelles Feedback zum Status seiner Steuerung erhalten. Das Feedback bildet das Resultat der Steuerung ab und soll dem Nutzer zeigen, dass seine Interaktionen bearbeitet werden (Norman 2002: S. 23).

Eine Individualisierbarkeit kann erreicht werden, indem die Bedienoberfläche alternative Darstellungsformen anbietet (Deutsches Institut für Normung 2006). Es soll eine Bedienoberfläche angeboten werden, die auf die steuerbaren Kontrollelemente reduziert wird, sowie eine erweiterte Bedienoberfläche, die die gesamte Struktur des Programms darlegt und somit eine Anpassung der Funktionsweise erlaubt.

Zur Erreichung einer Toleranz gegenüber Fehlern muss verhindert werden, dass Benutzereingaben zu undefinierten Systemzuständen führen (Deutsches Institut für Normung 2006). Diese Anforderung ist insbesondere wichtig hinsichtlich der divergierenden Ergebnisse der IPF. Da diese Ergebnisse zu undefinierten Systemzuständen führen, sollen sie möglichst verhindert werden.

3.3 Echtzeitfähigkeit

Die Benutzung eines Systems in Echtzeit stellt Anforderungen an die Latenz des Systems. Der Synthesizer lässt sich als „weiches Echtzeitsystem“ beschreiben. D.h. ein Überschreiten der geforderten Maximal-Latenz führt ausschließlich zu Einbußen in der Qualität, nicht etwa zu einer Beschädigung des Synthesizers oder gar zur Gefährdung des Nutzers. Es reicht demnach, wenn die geforderte Latenz durch ein statistisches Kriterium erreicht wird (Wörn u. Brinkschulte 2006: S. 321).

Als maximaler Latenzwert für interaktive Audio-Anwendungen kann ein Wert von 30 ms angenommen werden (Lago u. Kon 2004). Für das virtuelle Musikinstrument soll sowohl im Betrieb per Computer als auch im Betrieb per MIDI-Controller im Median eine Latenz von maximal 30 ms erreicht werden.

3.4 Erweiterbarkeit

Bei der Entwicklung des virtuellen Musikinstruments auf Basis der IPF handelt es sich um eine Neuentwicklung, deren Funktionsumfang erstmal nur einen wesentlichen Kern abdecken soll. Um noch weitere Funktionen implementieren zu können, ist es wichtig, dass das virtuelle Musikinstrument leicht zu erweitern ist.

Eine Software verliert in der Regel mit steigender Komplexität an Erweiterbarkeit. Die Komplexität des Programms wird zwar von der gewünschten Funktionalität maßgeblich festgelegt, lässt sich aber durch bestimmte Methoden beherrschen. Maßnahmen zur Beherrschung von Komplexität sind etwa Abstraktion, Hierarchisierung und Trennung von Funktionseinheiten nach Zuständigkeiten (Brcina u. a. 2009). Das virtuelle Musikinstrument soll daher möglichst nach diesen Prinzipien entwickelt werden.

4 Konzeption

Im Folgenden wird ein Konzept des IPF-basierten Synthesizer formuliert, das dem Anforderungsprofil möglichst gerecht werden soll. Hierzu wird eine geeignete Entwicklungsumgebung gewählt und die Umsetzung der Skalierung der IPF-Parameter und Dämpfung der Impulse konkretisiert.

4.1 Wahl der Entwicklungsumgebung

Als Entwicklungsumgebung wurde die von Cycling '74 hergestellte grafische, objektbasierte Umgebung „Max/MSP“ gewählt. Die Software ist bei Künstlern, Musikern und Entwicklern zur Herstellung interaktiver Systeme verbreitet, vor allem im audiovisuellen Bereich (Cycling '74 b).

Max/MSP ermöglicht eine zeitgleiche Implementierung und Anwendung, d.h. konzeptionelle, inhaltliche Anpassungen des Synthesizers können auch noch während des Entwicklungsprozesses umgesetzt werden. Diese Eigenschaft ist insbesondere für die Umsetzung der IPF nützlich, da sie eine Erforschung von klanglich interessanten Möglichkeiten zur Nutzung der IPF während des Entwicklungsprozesses erlaubt.

Während bestimmte Objekte den Nutzer direkt mit dem Programm interagieren lassen, können andere Objekte Informationen von externer Hardware empfangen. Eine Kommunikation mit MIDI-Geräten ist daher direkt möglich (Cycling '74 f).

Max/MSP ist unter anderem deshalb bei Musikern beliebt, weil eine Schnittstelle zur Musikproduktions-Software Ableton Live existiert. Mit der Erweiterung eines Max/MSP-Programms um sog. „Max for Live“-Objekte kann ein Max/MSP-Programm in Ableton Live als Plugin verwendet werden (Cycling '74 d).

4.2 Skalierung der IPF

Die Wertebereiche von α, β, γ und Δg sollen so gewählt werden, dass sich eine leichte Handhabung und eine gute Spielbarkeit des Instruments ergeben sowie eine möglichst hohe klangliche Vielfalt und Präzision.

Reflektionskoeffizienten β und γ

Je nach Wahl von β und γ verändert sich der Anteil der Werte für α, bei denen die IPF chaotisch wird oder konvergiert bzw. divergiert. Zur Erreichung einer hohen klanglichen Vielfalt soll der Wertebereich von β sowohl ein Verhalten ohne chaotische Bereiche abdecken, als auch für chaotische Bereiche verschieden hohen Anteils. β wird anhand der numerischen Simulationen von Linke u. a. (2019: S. 12) auf einen Bereich von 0 bis 0,3 eingegrenzt. γ soll so skaliert werden, dass die Bedingung aus Formel 2.3 stets erfüllt ist. Als Maximalwert für γ wird deshalb β gesetzt.

Anregungsstärke α

Falls die IPF ein Ergebnis mit Imaginärteil berechnet, soll die Klangsynthese abgebrochen werden, um zu verhindern, dass die IPF im Realteil divergiert. Für eine gute Spielbarkeit des Instruments muss der Wertebereich von α möglichst so skaliert werden, dass keine imaginären Ergebnisse auftreten, da sonst die Klangsynthese abgebrochen wird. Die Skalierung von α muss in Abhängigkeit von β und γ geschehen, da β und γ bestimmen, in welchen Bereichen von α es zu einem divergierenden Verhalten der IPF kommt.

Über numerische Simulationen lässt sich feststellen, ab welchem Wert von α es zu divergierendem Verhalten kommt (Linke u. a. 2019: S. 9-15). Dieser Mindestwert von α zur Vermeidung divergierenden Verhaltens gilt jedoch pro Simulation nur für eine einzige Kombination von β und γ. Es müsste also für jede mögliche Kombination von β und γ bereits im Voraus ein entsprechender Mindestwert für α bekannt sein, der dann bei der Steuerung von β und γ im Synthesizerbetrieb ausgelesen werden müsste. Wird nur eine überschaubare Anzahl von Kombinationen numerisch simuliert, ließen sich Fälle außerhalb des abgedeckten Bereichs interpolieren. Eine solche Interpolation ist nicht zwangsläufig akkurat.

Aufgrund dieser Nachteile einer Skalierung von α durch numerische Simulationen, wird das zu erwartende Verhalten der IPF von einem Künstlichen Neuronalen

Netzwerk (KNN) modelliert. Das Netzwerk wird anhand von Daten aus numerischen Simulationen trainiert. Das daraus resultierende Modell ist aber auch für Kombinationen außerhalb des zugrunde liegenden Datensatzes gültig. Nach einer erfolgreichen Entwicklung des KNN-Modells wird der Datensatz nicht mehr benötigt und kann somit im virtuellen Musikinstrument eingespart werden. Der einzustellende Mindestwert von α zur Vermeidung divergierenden Verhaltens kann dann anhand des KNN-Modells während des Synthesizerbetriebs berechnet werden.

Phasenverschiebung Δg

Die Skalierung von Δg wird anhand der Phasenverschiebung zusätzlicher Gleitbewegungen bei der Geige vorgenommen. Wie in Kapitel 2.3 dargestellt, bewegt sich die Phasenverschiebung einer einfachen zusätzlichen Gleitbewegung bei der Geige im Bereich um 180°. Die IPF soll bei einer Bifurkation erster Ordnung also eine Phasenverschiebung im Bereich um 180° bewirken.

Wie sich Δg der Bifurkation erster Ordnung in Abhängigkeit von β und γ verhält, ist noch nicht untersucht. Es scheint wahrscheinlich, dass sich Δg der Bifurkation erster Ordnung durch die Wahl von β und γ verändert. Also wäre auch hier eine Skalierung in Abhängigkeit von β und γ sinnvoll. Da die Spielbarkeit des Instruments von dieser Skalierung allerdings unbeeinflusst bleibt, wird der Einfachheit halber nur eine ungefähre Skalierung anhand des an Abbildung 2.1 ablesbaren Wertes von Δg vorgenommen und für alle Kombinationen von β und γ verwendet.

Δg der Bifurkation erster Ordnung bewegt sich in Abbildung 2.1 ungefähr um den Wert 1. Es wird daher festgelegt, dass ein Δg von 1 eine Phasenverschiebung um 180° bewirken soll.

4.3 Dämpfung

Für die Dämpfung der Impulse wird ein Karplus-Strong Algorithmus in das virtuelle Musikinstrument integriert. Die generelle Funktionsweise des Algorithmus ist in Abbildung 4.1 dargestellt. Das Eingangssignal erfährt eine Tiefpass-Filterung, indem der Durchschnitt von aufeinander folgenden Samplewerten berechnet wird. Nach der Tiefpass-Filterung wird das Signal bereits am Ausgang abgegriffen. Zugleich wird das gefilterte Signal in eine Verzögerungseinheit gegeben, die das Signal um die Anzahl der Samples verzögert, die eine zeitliche Verzögerung von einer Periode der gewünsch-

ten Tonhöhe ergibt. Dieses verzögerte Signal wird dann zum Eingangssignal addiert (Roads u. Strawn 1996: S. 293-294).

Durch das Delay-Element bleibt ein Impuls, der am Eingang anliegt, also noch länger im System. Die Tiefpass-Filterung bewirkt eine Dämpfung der Impulse. Neben der Tiefpass-Filterung können auch weitere Modifikationen des Signals vorgenommen werden (Roads u. Strawn 1996: S. 294).

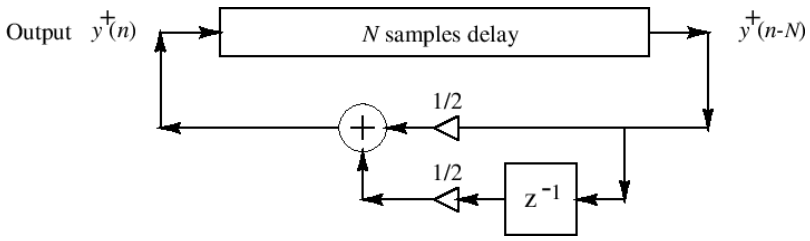

Abb. 4.1: Funktionsweise des Karplus-Strong Algorithmus. Das Eingangssignal wird durch die Berechnung des Durchschnitts von aufeinander folgenden Samplewerten Tiefpass-gefiltert. Das gefilterte Signal wird um die Länge der gewünschten Tonhöhe verzögert und erneut zum Eingangssignal addiert. Quelle: (Smith 2019: The Karplus-Strong Algorithm)

Die Stärke der Dämpfung bei Anwendung des Karplus-Strong Algorithmus ist von der Länge der Verzögerung, also von der Periodenlänge des gewünschten Tons abhängig. Um eine Dämpfung unabhängig von der Tonhöhe zu erhalten, kann der Algorithmus um einen Faktor zur frequenzabhängigen Streckung der Dämpfung erweitert werden (Roads u. Strawn 1996: S. 295-296).

5 Modellierung durch ein Künstliches Neuronales Netzwerk (KNN)

Das Netzwerk soll eine Klassifizierung des zu erwartenden Verhaltens der IPF leisten. Eine Unterteilung des Verhaltens der IPF in folgende drei Klassen erweist sich als sinnvoll (Linke u. a. 2019: S. 9):

1. Der Systemzustand g ist stabil.
2. Der Systemzustand g ist instabil, Bifurkationen und Chaos treten auf.
3. Die IPF liefert imaginäre Werte für g, der Realteil von g divergiert.

Um die Spielbarkeit des Synthesizers zu erhöhen, soll α anhand des Modells, das durch das Neuronale Netzwerk gebildet wird, so skaliert werden, dass die IPF möglichst keine imaginären bzw. divergierenden Werte liefert.

5.1 Das Multilayer-Perzeptron (MLP)

Die Grundversion eines Perzeptrons ist eine einfache Nachbildung eines Neurons. Ein Perzeptron, kann wie ein Neuron auch einen Erregungszustand aufweisen. Dieser Erregungszustand wird im Perzeptron durch einen binären Ausgabewert wiedergespiegelt. Der Ausgabewert wird 1, wenn eine bestimmte Aktivierungsschwelle erreicht wird. Wann diese Aktivierungsschwelle erreicht wird, ist von einer Aktivierungsfunktion abhängig. Als Aktivierungsfunktion wird z.B. eine lineare, eine Sigmoid- oder eine ReLU-Funktion verwendet (Gurney 1997: S. 19-20). Eine Sigmoidfunktion ist eine Stufenfunktion mit weichen Übergängen. Der Ausgabewert der Sigmoidfunktion bewegt sich abhängig vom Eingabewert zwischen 0 und 1 (Rashid u. Langenau 2017: S. 33)

Welchen Wert die Aktivierungsfunktion erreicht, ist abhängig von den Eingabewerten, also den Input-Werten der Datenvektoren, sowie von der Gewichtung dieser Eingabewerte. Die Eingabewerte werden mit einer bestimmten Gewichtung multipliziert, anschließend aufsummiert und schließlich in die Aktivierungsfunktion gegeben (Rashid u. Langenau 2017: S. 41-42).

Der Trainingsprozess eines Perzeptrons wird durchgeführt, indem Schritt für Schritt die Gewichtung der Eingabewerte verändert wird. Der Ausgabewert des Perzeptrons wird nach jeder Veränderung der Gewichtung mit dem tatsächlichen, gewünschten Ausgabewert verglichen. Je nachdem, ob der Ausgabewert größer, kleiner oder gleich dem gewünschten Wert ist, wird die vorherige Gewichtung inkrementiert, dekrementiert bzw. beibehalten (Gurney 1997: S. 41-43). Der gewünschte Ausgangswert muss also beim Training bekannt sein, weshalb es sich um ein sog. „überwachtes Lernen“ handelt (Reed u. Marks II 1999: S. 7).

Ein solches Perzeptron besitzt mit der Eingabe- und Ausgabeschicht zwei Schichten. Das Multilayer-Perzeptron erweitert diese Schichten um eine weitere verborgene Schicht (engl. „hidden layer“). Diese verborgene Schicht ermöglicht die Klassifizierung nicht-linearer Systeme, indem mehrere Perzeptronen darin miteinander logisch verknüpft werden. In Anlehnung an Rashid u. Langenau (2017) wurden die in Tabelle 5.1 dargestellten Spezifikationen für das MLP gewählt.

Spezifikation	**Wert**
Anzahl Ausgabewerte	3
Anzahl verborgener Schichten	1
Anzahl Perzeptronen pro Schicht	4
Min. Anzahl Trainingsdurchläufe	10
Max. Anzahl Trainingsdurchläufe	100
Gewichtsveränderung pro Durchlauf	0.1
Min. Verbesserung pro Durchlauf	0.00001
Null Rejection	Keine
Anzahl der Trainingsgruppen pro Durchlauf	10
Aktivierungsfunktion	Sigmoid
Größe der Validierungsgruppen	20

Tabelle 5.1: MLP-Spezifikationen zur Klassifizierung des erwarteten Verhaltens der IPF

5.2 Max/MSP-Bibliotheken

Für Max/MSP gibt es zwei extern entwickelte Bibliotheken um den Anwendungsbereich des maschinellen Lernens: die ml.star- sowie die ml.lib-Bibliothek (Bullock u. Momeni 2015, Smith u. Garnett 2012). Das Objekt aus der ml.lib-Bibliothek, das ein mehrschichtiges Perzeptron realisiert, bietet im Vergleich zum ml.star-Pendant deutlich mehr Möglichkeiten zur Kontrolle von bestimmenden Parametern für die Modellbildung und den Trainingsprozess. Da unter Kontrolle dieser Parameter deutlich bessere Ergebnisse erzielt werden können, wird für die Modellierung der IPF die ml.lib-Bibliothek verwendet.

ml.lib bedient sich einer bereits für C++ entwickelten Bibliothek, dem „Gesture Recognition Toolkit“ (GRT) von Nick Gillian (Bullock u. Momeni 2015: S. 266). Das GRT ist speziell auf die Analyse von Gestik in Echtzeit ausgelegt, aber auch für andere Einsatzbereiche des maschinellen Lernens geeignet.

5.3 Training

Datensatz

Das KNN soll ein Modell liefern, das für alle denkbaren Kombinationen von α, β und γ innerhalb der bereits eingegrenzten Wertebereiche funktioniert. Der Trainingsdatensatz umfasst $1/\alpha$-Werte im Bereich von 1 bis 4 in 101 Abstufungen. Die Werte für β bewegen sich im Bereich von 0.0 bis 0.3 mit 31 Abstufungen und die Werte für γ im Bereich von 0.0 bis 0.2 mit 21 Abstufungen. Die Schrittgröße für die Reflektionsstärken wurden verhältnismäßig grob gewählt, um den Trainingsaufwand einzugrenzen.

Für jede Abstufung von α werden 12 Iterationsschritte berechnet, bevor die Klassifizierung der Wertekombination dem Datensatz hinzugefügt wird. So soll verhindert werden, dass das Einschwingverhalten der IPF als Bifurkation oder Chaos missklassifiziert wird. Aus der Kombination der Werte für α, β und γ ergibt sich eine Größe von

$$n = 101 \cdot 31 \cdot 21 = 65\,751$$

für den Trainingsdatensatz. Bei der Modellierung soll der Einfluss des ursprünglichen Systemzustands g_0 vorerst nicht berücksichtigt werden. Daher wird g_0 konstant

gehalten. In Anlehnung an Linke u. a. (2019: S. 9-10) wurde für das Training $g_0 = g_s$ festgelegt.

Klassifizierung

In Abbildung 5.1 ist dargestellt, welche Logik in Max/MSP implementiert ist, um eine Klassifizierung des Outputs der IPF vorzunehmen. Zur Klassifizierung wird der Systemzustand g nach 150 Iterationsschritten betrachtet. Zunächst wird überprüft, ob sich aus der IPF ein imaginärer Wert für den Systemzustand g_{150} ergibt. In diesem Fall wird der Output der IPF in Max zu „NaN“ (engl. not a number). NaN ist ein numerischer Datentyp, der ungleich sich selbst ist. Auf dieses Kriterium wird wie in Abbildung 5.1 dargestellt getestet. Wird diese Bedingung erfüllt, erhält der Output, der dem Datenvektor zugeordnet wird, den Wert 3.

Wird diese Bedingung nicht erfüllt, wird g_{150} auf Klasse 1 (Stabilität) geprüft. Bei stabilem Verhalten konvergiert der Systemzustand g gegen g_s. Falls g_{150} im Konfidenzintervall von $g_s \pm 0,1\,\% \cdot g_s$ liegt, wird Klasse 1 gesetzt. Ist schließlich weder die Bedingung für Klasse 3, noch die Bedingung für Klasse 1 erfüllt, wird angenommen, dass der Zustand der IPF chaotisch ist oder Bifurkationen auftreten. Es wird somit Klasse 2 gesetzt.

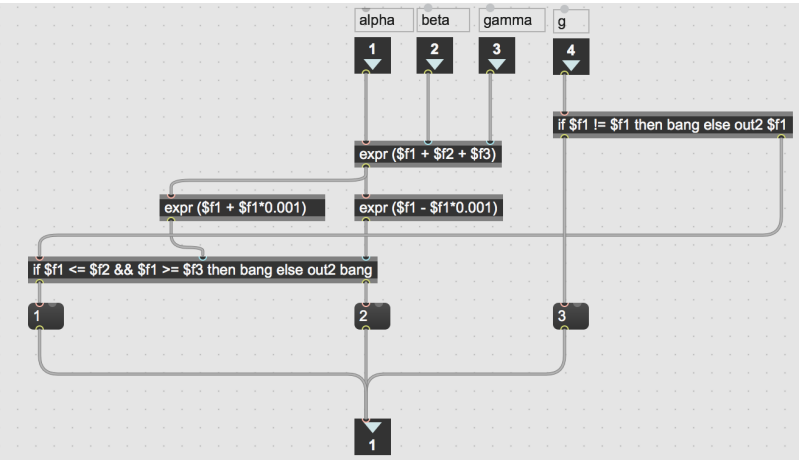

Abb. 5.1: Screenshot der Implementierung der Logik zur Klassifizierung der Datenvektoren in Max/MSP. Der Output der IPF wird erst auf Klasse 3 (rechts), anschließend auf Klasse 1 (links) geprüft und zuletzt auf Klasse 2 gesetzt (mittig).

5.4 Modell

Das KNN-Modell berechnet, welcher Klasse mögliche Kombinationen von α, β und γ zugehörig sind. Ein aus α, β und γ bestehender Datenvektor dient also als Eingabewert. Ausgabewert des KNN-Modells ist die Klasse des erwarteten Verhaltens, also Klasse 1 (stabil), 2 (Bifurkationen/Chaos) oder 3 (imaginär/divergierend).

In Abbildung 5.2 (rechts) ist für zwei Reflektionspunkte ($\gamma = 0$) das erwartete Verhalten der IPF dargestellt, das durch das MLP modelliert wird. Zur Gegenüberstellung ist links das tatsächliche Verhalten, das sich bei der Sammlung der Daten ergeben hat, abgebildet.

Das Max-Objekt gibt leider keinerlei Angaben darüber, wie hoch der Fehler des MLP-Modells ist, wenn dieses gegen den Datensatz getestet wird. Anhand von Abbildung 5.2 lässt sich zumindest für $\gamma = 0$ eine Ähnlichkeit zwischen tatsächlichem Datensatz und dem Modell erkennen. Das Modell befindet sich unter dem Dateinamen „mlp_ipf.model“ auf dem Datenträger im Anhang.

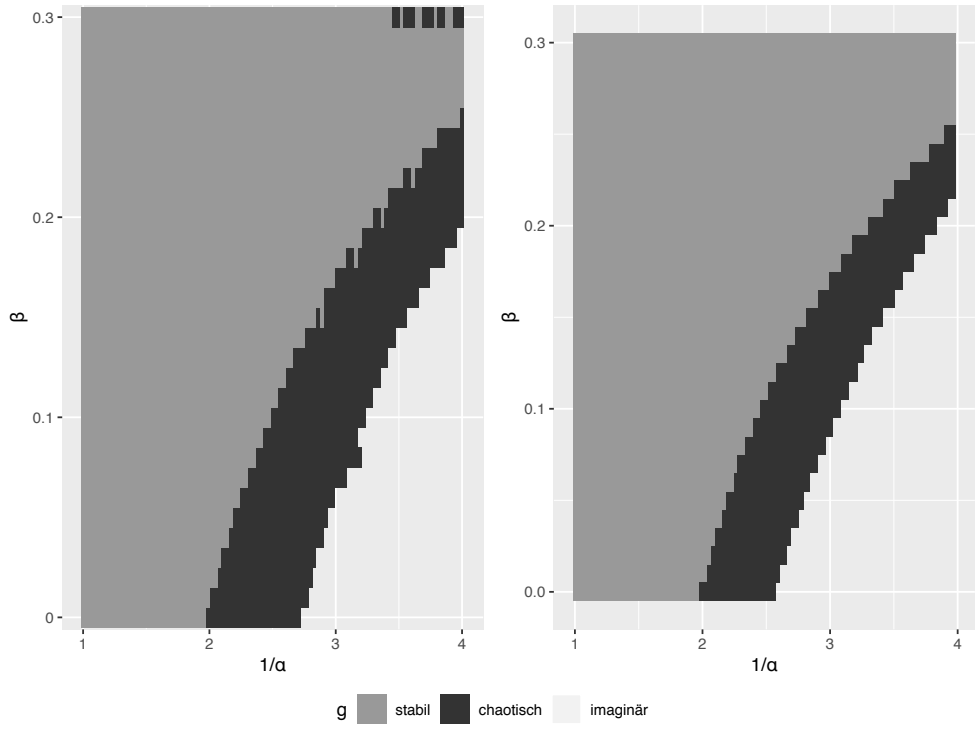

Abb. 5.2: Reales Verhalten der IPF (links) sowie KNN-Modell des erwarteten Verhaltens (rechts) bei zwei Reflektionspunkten ($\gamma = 0$). Bei einem Startwert von $g_0 = 1$ ergeben sich in Abhängigkeit von α und β die abgebildeten stabilen, chaotischen und imaginären Bereiche für g.

5.5 Limitationen

Der Datensatz beinhaltet für die einzelnen Input-Parameter recht wenige Abstufung. Pro Wertekombination wurden nur 150 Iterationen der IPF durchgeführt, was zu einer fehlerhaften Klassifizierung führen kann. Es ist damit zu rechnen, dass diese Ungenauigkeiten in einer gewissen Fehlerhaftigkeit des Modells resultieren. Dieser Fehler befindet sich bereits im zugrunde liegenden Datensatz und würde daher nicht einmal durch die Fehlerrate des Modells erfasst werden.

Selbst bei einer perfekten Abbildung der Realität, würde das Modell ausschließlich für Startwerte von $g_0 = g_s$ gelten. Da das Verhalten der IPF je nach Startwert hiervon abweichen kann (Linke u. a. 2019: S. 10), ist das KNN-Modell als Tendenz des erwarteten Verhaltens zu verstehen und nicht als allgemein gültig zu betrachten.

Ein Problem in der Anwendung des Modells liegt darin, dass das verwendete Max-Objekt der Bibliothek ml.lib gespeicherte Modelle offenbar fehlerhaft ausliest (Fred-Voisin 2017). Damit das MLP-Modell die Klassifizierung wie gedacht durchführt, muss das Training zu Beginn jeder Session neu durchgeführt werden, also immer wenn das Instrument neu geladen wird. Dieser Trainingsaufwand verlängert die Initialisierung des Synthesizers um eine Zeit von etwa einer Minute. Der Datensatz hingegen wird korrekt gespeichert und ausgelesen. Er bleibt somit erhalten.

6 Umsetzung

Eine Übersicht der Funktionsweise des virtuellen Musikinstruments ist in Abbildung 6.1 dargestellt. Das Instrument reagiert auf MIDI-Befehle, mit denen die Kontrollparameter der IPF sowie die Frequenz des Sägezahnsignals und das Gate gesteuert werden. In Abbildung 6.1 ist lediglich der Syntheseweg einer einzigen Stimme dargestellt. Im Synthesizer sind acht Stimmen integriert. Verschiedene MIDI-Befehle für Tonhöhe und Anschlagsstärke können also an bis zu acht Stimmen verteilt werden. Die Kontrolle von α geschieht durch einen MIDI-Parameter, der für alle Stimmen gilt.

Wenn das Gate offen ist, werden Berechnungen der IPF im Takt des Sägezahnsignals getriggert. Die IPF moduliert je nach Einstellung entweder die Frequenz des Phasors oder die Phase. Der Phasor liest die beiden Impulsformen aus dem Wavetable aus. Die aus dem Wavetable ausgelesenen Impulse können über einen Filter bearbeitet werden, bevor sie in den Karplus-Strong Algorithmus zur Dämpfung und schließlich auf den Audio Ausgang gegeben werden. Das Modell des Künstlichen Neuronalen Netzwerks skaliert in Abhängigkeit von β und γ den Wertebereich von α.

Alle rot gekennzeichneten Elemente befinden sich auf der Oberfläche des virtuellen Musikinstruments und sind somit durch den Nutzer steuerbar. Die Umsetzung des Syntheseweges, inklusive der Berechnung der IPF und des Triggersystems sowie der Modulation, die durch die IPF erfolgt, wird im Folgenden dargestellt.

6.1 Klangsynthese

Die Klangsynthese in Max/MSP geschieht durch das Abspielen eines Wavetables über ein Sägezahnsignal. Die Frequenz der Klangsynthese lässt sich über die Frequenz des Sägezahnsignals manipulieren. Der Wertebereich des Sägezahnsignals bewegt sich zwischen 0 und 1 und dient als Index für den dahinter geschalteten Wavetable.

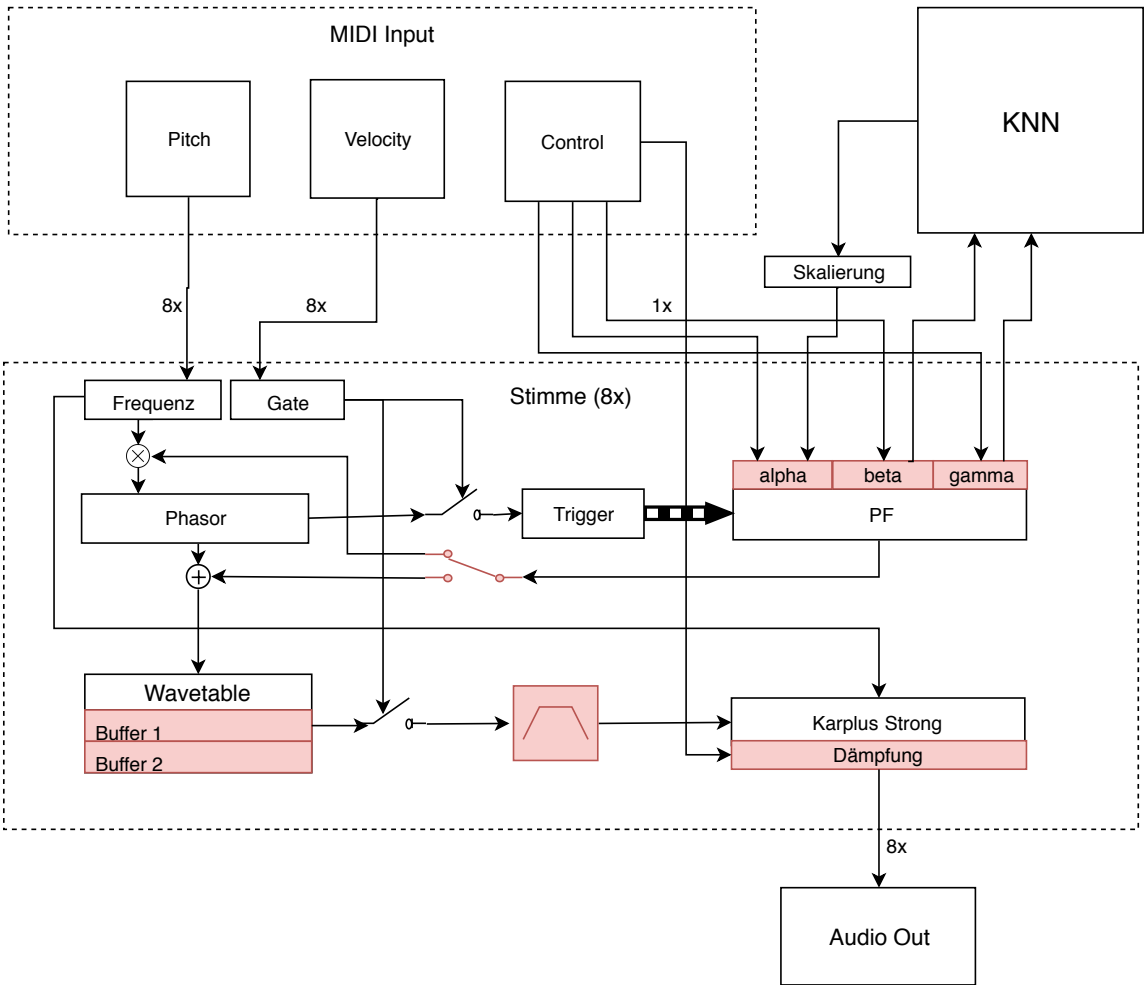

Abb. 6.1: Funktionelle Darstellung des virtuellen Musikinstruments. Rot gekennzeichnete Bereiche sind auf der Bedienoberfläche steuerbar.

Der Wavetable greift auf ein „buffer“-Objekt zurück, dessen Inhalt durch den Benutzer beliebig einstellbar ist. Unabhängig von Inhalt und Länge dieses Wavetables bewegen sich die Indizes seiner Amplitudenwerte im Bereich zwischen 0 und 1, d.h. der erste Amplitudenwert wird durch den Index 0 abgerufen und der letzte durch den Index 1. Dank dieser Skalierung ist der Wavetable direkt kompatibel mit den Werten des Sägezahnsignals und lässt sich hierdurch abspielen. Über die Kombination von Sägezahn, Wavetable und Buffer lassen sich Wellenform, Phasenverschiebung, Amplitude und Frequenz der Schwingung einstellen.

Die Phasenverschiebung kann manipuliert werden, indem zur Amplitude des Sägezahnsignals ein Wert zwischen -1 und +1 addiert wird, wobei -1 einer Phasenver-

schiebung von $-360°$ entspricht bzw. $+1$ einer Phasenverschiebung von $+360°$. Um sicherzugehen, dass sich der Index nach einer solchen Phasenverschiebung nicht außerhalb des Bereichs von 0 und 1 befindet, wird der Ausgang nach der Addition nochmals mithilfe eines Modulos kontrolliert. So wird eine Phase von $370°$ umgerechnet in die identische Phase von $10°$.

Kombination verschiedener Impulsformen

Die im buffer-Objekt gespeicherte Wellenform des synthetisierten Klangs ist die Impulsform in der IPF. Damit zwei verschiedene Impulsformen miteinander kombiniert werden können, wurde der o.g. Klangsynthese-Weg für zwei parallele Audiokanäle implementiert. Es lässt sich z.B. eine Sägezahn- mit einer Sinusschwingung kombinieren. Es überlagern sich dann beide Schwingungen zu einer Misch-Wellenform.

Interessant wird die Kombination zweier Impulsformen auch dann, wenn die IPF in einen instabilen Zustand gerät, bei dem Bifurkationen auftreten. Die Ergebnisse der IPF werden abwechselnd auf die zwei parallelen Audiokanäle verteilt. Im Fall von einer einzigen Bifurkation, die in zwei verschiedene Phasenzustände übersetzt wird, werden die unterschiedlichen Impulsformen mit unterschiedlichen Phasen ausgelesen. Ein Beispiel für eine solche Kombination von Sägezahn- und Sinusschwingung ist in Abbildung 6.2 dargestellt.

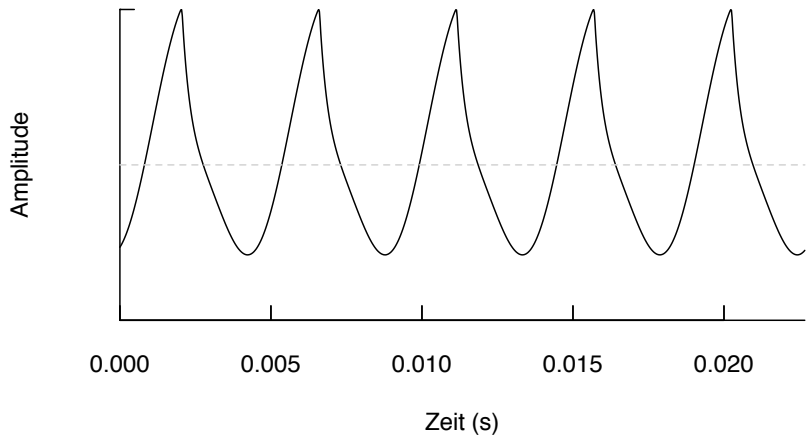

Abb. 6.2: Normalisierter Amplitudenverlauf des Ausgangssignals des Synthesizers bei Überlagerung zwei verschiedener Wellenformen. Als Impulsformen wurden eine Sägezahn- und eine Sinusschwingung gewählt, die durch die IPF um $169°$ phasenverschoben zueinander sind.

Hier wurde der erste Buffer mit einer Sägezahn-Wellenform gefüllt und der zweite

Buffer mit einer Sinus-Wellenform. Zur Berechnung der IPF wurden keine Reflektionspunkte ($\beta = 0, \gamma = 0$) sowie ein α von 0,42 verwendet. Die Sägezahnschwingung hat eine Phase von 194° und die Sinusschwingung eine Phase von 25°. Die beiden Schwingungen sind also um 169° phasenverschoben zueinander. Die Frequenz der Schwingungen beträgt 440 Hz.

Polyphonie

Der Synthesizer besitzt acht Stimmen, die gleichzeitig gespielt werden können. Das verwendete Objekt „mc.poly“ beinhaltet acht Instanzen eines darin gesondert implementierten Max/MSP-Programms, in welchem die Synthese geschieht. Über sog. „midi events“ werden Informationen zu Frequenz und dazugehöriger Anschlagsstärke an die verschiedenen Stimmen verteilt, die diese Informationen dann parallel verarbeiten. Die Anschlagsstärke wird als Gate für die jeweilige Stimme verwendet. Bei einer Anschlagsstärke von 0 wird das Gate geschlossen, bei einer Anschlagsstärke größer 0 geöffnet.

Nach erfolgter Klangsynthese im Objekt mc.poly werden die acht Audiokanäle auf zwei Kanäle bei mittigem Panning runtergemischt. Bevor das Stereo-Signal in den Audio-Ausgang des Max/MSP-Programms gegeben wird und schließlich als Klang hörbar ist, wird das Signal noch von einem Gain-Slider kontrolliert. Mit diesem kann die Gesamt-Lautstärke angepasst werden. Die Verarbeitungskette zwischen MIDI Input und Audio Output ist in Abbildung 6.3 dargestellt.

Kein Bestandteil der Polyphonie ist die Steuerung des Anregungsparameters α. Grund hierfür ist in erster Linie, dass die Implementierung einer polyphonen Steuerung von α in Max/MSP sich als schwierig gestaltet. α kann im Synthesizer vom „Aftertouch“-Parameter oder vom Modulationsrad des MIDI-Controllers gesteuert werden. Die Steuerung des Modulationsrads bestimmt lediglich einen einzelnen Parameter und kann daher nicht polyphon verwendet werden, sondern muss für alle Stimmen gelten.

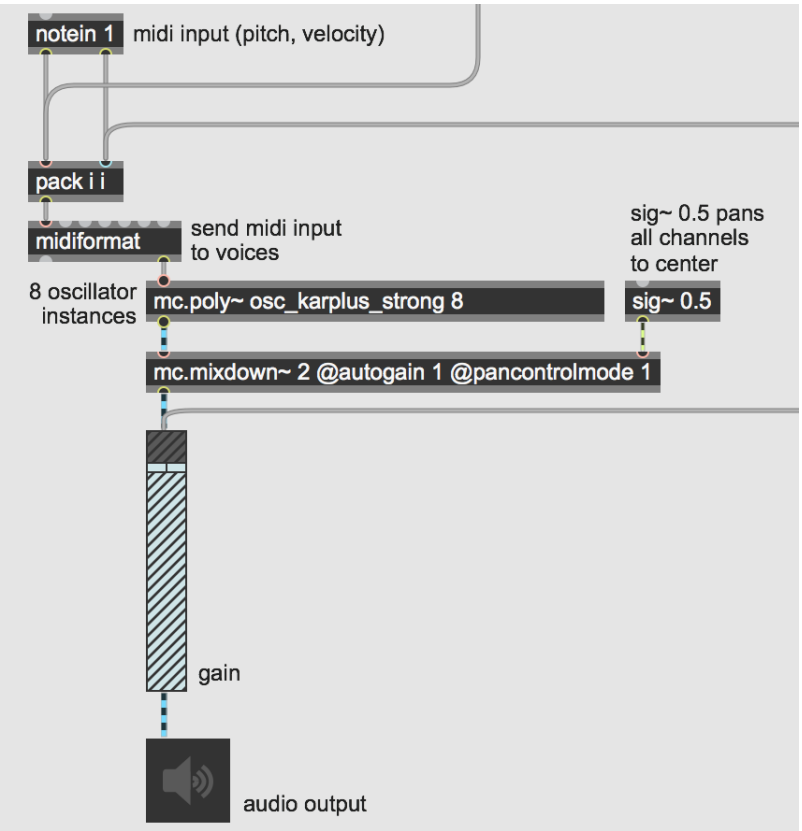

Abb. 6.3: Screenshot der in Max/MSP implementierten Verarbeitungskette zwischen MIDI-Input und Audio Output. Die MIDI-Informationen zu Tonhöhe und Velocity werden über das Objekt „midiformat“ an die 8 Stimmen des „mc.poly“ Objektes verteilt. Nach der Verarbeitung in mc.poly geschieht ein Mixdown auf 2 Kanäle, gefolgt von einer Lautstärkekontrolle.

6.1.1 Phasenmodulation

Abbildung 6.4 zeigt einen Screenshot der in Max/MSP implementierten Verarbeitungsschritte zur Phasenmodulation durch die IPF. Die Werte der Phasenmodulation, die sich aus der IPF ergeben, werden abwechselnd auf zwei Audiokanäle verteilt, damit eine Synthese von zwei verschiedenen, zueinander phasenverschobenen Impulsformen ermöglicht werden kann.

Zur Phasenmodulation wird das Sägezahnsignal, das den Wavetable ausliest, in seiner Amplitude moduliert. Zum Amplitudenwert des Sägezahnsignals wird Δg aus der IPF hinzugefügt. Δg wird hierzu aus g_+ und g kalkuliert und anschließend zur Skalierung durch den Faktor 2 geteilt. Der Faktor wurde so gewählt, dass sich die Phasenverschiebung von zwei aufeinander folgenden Impulsen bei einer Bifurkation

erster Ordnung für den Fall $\beta = 0$ und $\gamma = 0$ im Bereich von 180° bewegt.

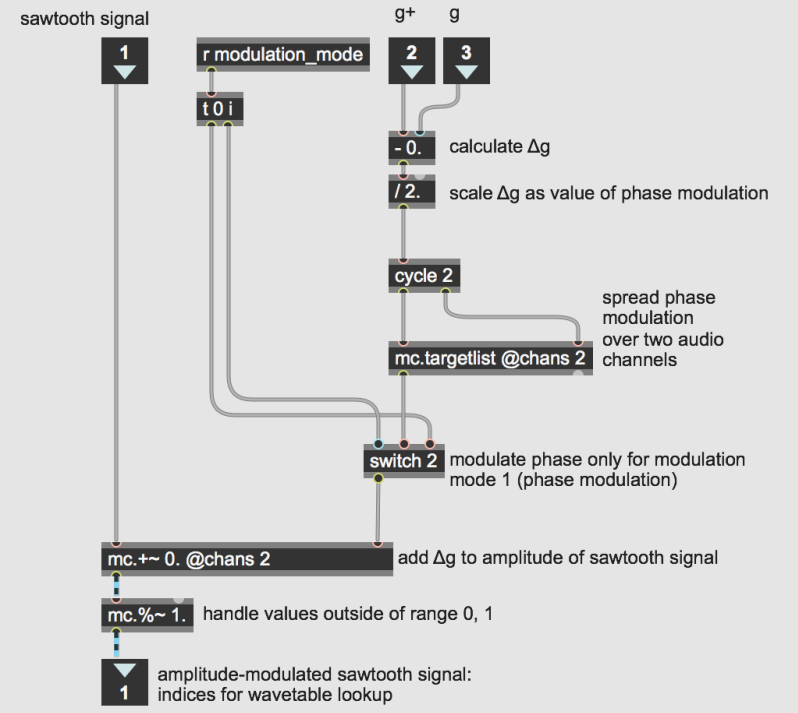

Abb. 6.4: Screenshot der in Max/MSP implementierten Verarbeitungskette zur Phasenmodulation durch die IPF. Δg wird kalkuliert, skaliert und auf zwei Audiokanäle verteilt. Die modulierten Werte werden anschließend durch einen Modulo 1-Operator auf einen Wertebereich von 0 bis 1 begrenzt.

Für eine Bifurkation erster Ordnung ergibt sich für die aufeinander folgenden Werte Δg_+ und Δg ein Verhältnis von

$$\Delta g_+ = -\Delta g \tag{6.1}$$

Die Phasenverschiebung der Impulse zueinander beträgt:

$$\Delta\phi = \left| \frac{\Delta g_+}{2} \% 1 - \frac{-\Delta g}{2} \% 1 \right| \tag{6.2}$$

Die Phasenverschiebung wird somit 0, wenn $\Delta g_+ = 1$ ist:

$$\begin{aligned}\Delta\phi &= \left|\frac{1}{2}\%1 - \frac{-1}{2}\%1\right| \\ &= |0,5 - 0,5| \\ &= 0\end{aligned}$$

Durch den Modulo 1-Operator wird die Phasenverschiebung bei einer Bifurkation erster Ordnung immer 0, wenn Δg ein ganzzahliges Vielfaches von 1 ist. In diesem Fall ergibt sich also ein stabiler Ton ohne zusätzliche Obertöne, obwohl die IPF verschiedene Systemzustände errechnet. Die Umsetzung der IPF in eine Steuerung von Klangparametern ist in diesen Fällen also fehlerhaft.

Für andere Werte von Δg ergibt sich eine Phasenverschiebung ungleich 0, z.B.:

$$\begin{aligned}\Delta\phi &= \left|\frac{0,5}{2}\%1 - \frac{-0,5}{2}\%1\right| \\ &= |0,25 - 0,75| \\ &= 0,5 \\ &\hat{=} 180°\end{aligned}$$

6.1.2 Frequenzmodulation

Zusätzlich zur Phasenmodulation ist eine Frequenzmodulation in den Synthesizer implementiert worden, die eine weitere klangliche Steuerung durch die IPF ermöglicht. Befindet sich das Instrument im Frequenzmodulations-Modus, wird der in Abbildung 6.5 dargestellte Prozess durchgeführt.

Die Frequenz des Sägezahnsignals, das den Wavetable ansteuert, wird in diesem Modus durch die IPF moduliert. Als Modulationsfaktor wird der Wert des Systemzustands g verwendet. Dieser wird in Bezug zum Fixpunkt $g_s = \alpha + \beta + \gamma$ (Linke u. a. 2019: S. 7) gesetzt und zur Skalierung durch diesen Fixpunkt dividiert.

Die modulierte Frequenz wird, wie in Abbildung 6.1 dargestellt, nur für die Steuerung des Sägezahnsignals verwendet. Für die Verzögerung der gedämpften Impulse im Karplus-Strong Algorithmus wird weiterhin die unmodulierte Grundfrequenz verwendet.

Die Frequenzmodulation entfernt sich deutlich von der Interpretation der IPF am Beispiel der Geige. Sie wurde während des Entwicklungsprozesses erprobt und auf-

grund der interessanten klanglichen Ergebnisse implementiert.

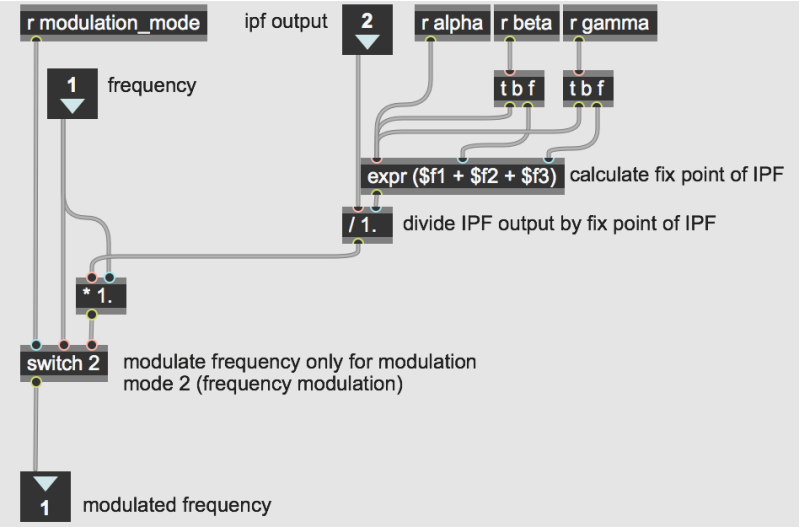

Abb. 6.5: Screenshot der in Max/MSP implementierten Verarbeitungskette zur Frequenzmodulation. Der Systemzustand g wird durch den Fixpunkt g_s geteilt und anschließend mit der Grundfrequenz multipliziert.

6.2 Berechnung der IPF

Da der zeitliche Abstand zwischen g_+ und dem vorherigen Systemzustand g einer Periodenlänge entspricht, muss pro Periode eine iterative Berechnung der IPF durchgeführt werden. Hierzu wurde ein entsprechendes Triggersystem in das virtuelle Musikinstrument implementiert. Hinter dem Triggersystem findet die Berechnung der IPF statt. Diese wurde einmal, wie von Bader vorgesehen, mit reellen Reflektionskoeffizienten umgesetzt. Zusätzlich wurde die IPF für eine komplexe Berechnung transformiert, inklusive komplexer Reflektionskoeffizienten.

6.2.1 Triggersystem

In Abbildung 6.6 ist ein Screenshot des in Max/MSP implementierten Triggersystems zu sehen. Als Signal, das den Trigger auslöst, wird das Sägezahn-Signal verwendet, das auch den Wavetable ansteuert. Der Trigger zur Berechnung der IPF wird ausgelöst, wenn die Steigung des Sägezahnsignals von einem positiven Wert auf einen negativen Wert wechselt. Dieser Wechsel geschieht exakt zum Ende jeder Periode und ist daher als Trigger für die IPF geeignet.

Da eine Trigger-Auslösung immer im Abstand der Grundfrequenz geschehen soll, muss sie unabhängig von der Phase des Impulses geschehen. Das Sägezahnsignal, das den Trigger auslöst, wird daher im Signalfluss abgegriffen, bevor es zur Phasenverschiebung des Impulses verändert wird.

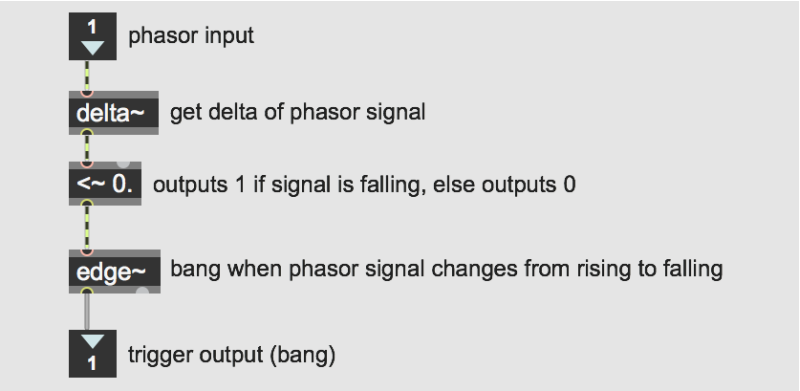

Abb. 6.6: Screenshot des in Max/MSP implementierten Triggersystems. Wechselt das phasor-Signal von einer positiven Steigung auf eine negative Steigung, wird ein Trigger ausgelöst, der wiederum eine Iterationsberechnung der IPF auslöst.

Für die Genauigkeit des Triggersystems der IPF sind die Audio-Einstellungen in Max/MSP von enormer Wichtigkeit. Zur Messung der Genauigkeit wurde das phasor-Signal, das den Trigger auslöst, aufgenommen. Zeitgleich wurde ein Impulssignal aufgenommen, das durch den Trigger ausgelöst wird. Anhand der Summe beider Signale lässt sich eine zeitliche Verzögerung zwischen gewünschter Trigger-Auslösung und tatsächlichem Trigger feststellen.

In Abbildung 6.7 und 6.8 ist jeweils die Summe dieser beiden Signale dargestellt mit dem Unterschied, dass in Max/MSP verschiedene Audio-Einstellungen während der Aufnahmen gewählt wurden. Die Aufnahme in Abbildung 6.7 ist entstanden, als Max/MSP im sog. „Audio Interrupt“-Modus war. Während der Aufnahme in Abbildung 6.8 hat Max/MSP ohne Audio Interrupt operiert. Abbildung 6.7 zeigt einen ziemlich akkuraten Trigger, der als dicke Linie sichtbar ist. In Abbildung 6.8 ist eine deutliche Latenz zwischen abfallendem Sägezahnsignal, also der gewünschten Trigger-Auslösung, und dem tatsächlichen Trigger zu sehen. Die restlichen Audio-Einstellungen wurden für den abgebildeten Vergleich konstant gehalten.

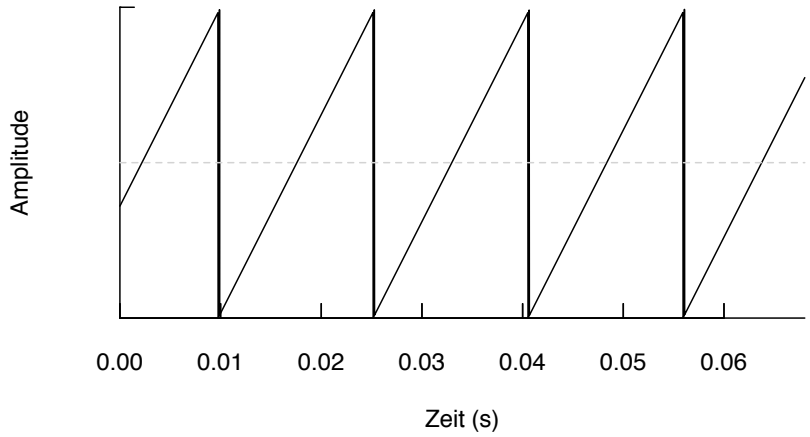

Abb. 6.7: Normalisierter Amplitudenverlauf der Summe des phasor-Signals und des Impulssignals, das zeitgleich mit der IPF getriggert wird. Max/MSP operiert im „Audio Interrupt"-Modus.

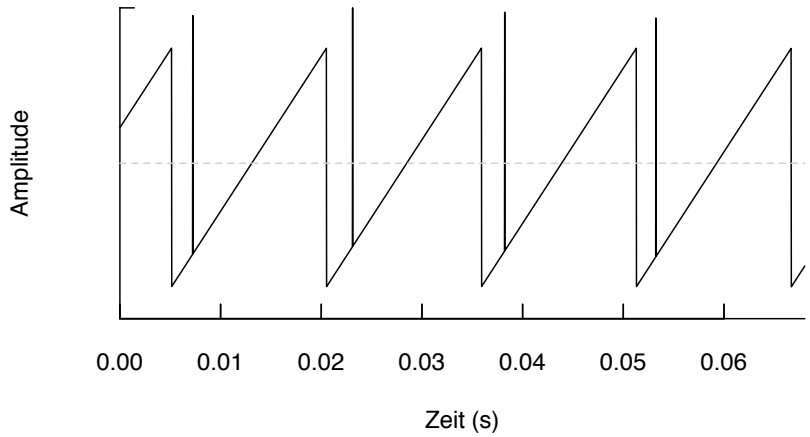

Abb. 6.8: Normalisierter Amplitudenverlauf ohne Audio Interrupt. Eine deutliche Latenz zwischen Abfallen des Sägezahnsignals und Impulssignal ist erkennbar.

Für die Genauigkeit des Triggersystems sind die im Folgenden aufgeführten Audio-Einstellungen von Relevanz. Die Audio-Einstellungen werden im Synthesizer automatisch über das Objekt „adstatus" auf die benötigten Einstellungen gesetzt.

Signal Vector Size

Die Größe der Signal-Vektoren bestimmt, wie viele Samples in der Audio-Domäne von MSP-Objekten gleichzeitig verarbeitet werden (Cycling '74 e). Für die zuvor abgebildeten Verläufe wurde eine Vektorgröße von 2 gesetzt. Bei größeren Vektoren

konnte eine deutliche Latenz festgestellt werden. Daher beträgt die Vektorgröße im Patch 2.

Scheduler in Overdrive

Diese Einstellung legt fest, mit welcher Priorität der Event Scheduler in Max/MSP operiert. Der Event Scheduler bestimmt die zeitliche Abfolge von allen Ereignissen außerhalb der Audio-Domäne. Die Aktivierung von „Scheduler in Overdrive“ erhöht im Allgemeinen die zeitliche Exaktheit eines Patches (Cycling ’74 e) und muss daher für den IPF-Synthesizer gesetzt sein.

Audio Interrupt

Der Event Scheduler kann bei einer Aktivierung von „Scheduler in Overdrive“ zusätzlich noch in den „Audio Interrupt“-Modus gesetzt werden. Der Event Scheduler setzt dann zeitlich exakt vor der Verarbeitung eines Signalvektors in der Audio-Domäne ein. Durch diese Einstellung kann die zeitliche Genauigkeit enorm erhöht werden, wenn Ereignisse in der Audio-Domäne vom Event Scheduler kontrolliert werden (Cycling ’74 e). Bei der Steuerung von Klangparametern durch die IPF tritt dieser Fall ein. So erklärt sich auch der in Abbildung 6.7 und 6.8 erkennbare Unterschied in der zeitlichen Exaktheit des Triggers. Der Event Scheduler muss sich also im Audio Interrupt Modus befinden.

Letztendlich kann eine ausreichende Genauigkeit nur durch die Kombination von geringer Vektorgröße und Audio Interrupt erzielt werden. So operiert der Event Scheduler bei einer Vektorgröße von 2 mit einer Frequenz, die bis zur Hälfte der Samplingfrequenz beträgt. Standardmäßig operiert der Scheduler in einer deutlich geringeren Frequenz von 1 000 Ereignissen pro Sekunde (Cycling ’74 e).

6.2.2 Schrittweise Berechnung

Formel 2.1 der IPF ist im virtuellen Musikinstrument in mehrere einzelne Berechnungsschritte unterteilt worden. Schritt 1 der Berechnung ist der Term im Logarithmus:

$$out_1 = \frac{g}{\alpha} - \frac{\beta}{\alpha} e^{g-g_{-1}} - \frac{\gamma}{\alpha} e^{g-g_{-2}} \tag{6.3}$$

In Schritt 2 wird das Ergebnis des Terms aus Formel 6.3 in die Polarschreibweise komplexer Zahlen umgewandelt. Für diese Umwandlung gibt es in Max/MSP ein Objekt namens „cartopol“. Hierbei wird kein Imaginärteil, sondern lediglich der Realteil übergeben. Diese Umwandlung führt dazu, dass mit dem Betrag des Ergebnisses weitergerechnet wird. Sollte der Term also ein negatives Ergebnis haben, wird verhindert, dass Max versucht, den Logarithmus einer negativen Zahl zu berechnen. Der Logarithmus einer negativen Zahl würde zum Output NaN führen. Es ergibt sich somit für den zweiten Schritt:

$$out_2 = |out_1| \tag{6.4}$$

Zugleich wird der Winkel aus der Polardarstellung auf Ungleichheit mit 0 geprüft. Der Winkel wird bei einem negativen Ergebnis von Formel 6.3 zu π. Wenn der Winkel ungleich 0 ist, wird die Klangsynthese abgebrochen. In der Berechnung der IPF ist somit eine Abbruchbedingung implementiert, um zu verhindern, dass die IPF komplexe Zustände errechnet und im Realteil divergiert.

Im dritten Schritt wird schließlich die Berechnung des Folgezustands g_+ durchgeführt:

$$out_3 = g - ln(out_2) \tag{6.5}$$

Die Ergebnisse der zwei vorhergegangen Iterationsschritte werden im Objekt „bucket“ zwischengespeichert. Das Ergebnis für g_+ sowie die zwei gespeicherten Ergebnisse werden über Inputs in Formel 6.3 gegeben und ersetzen dort Variablen für g, g_{-1} und g_{-2}. Das Ergebnis für g_+ wird zusätzlich als g auch wieder in Formel 6.5 eingespeist.

α, β und γ sind in Schritt 1 ebenfalls als Variablen implementiert und werden entsprechend der Steuerung durch den Nutzer angepasst. Bei Initialisierung des Synthesizers werden g_0, g_1, g_2 und g_+ auf 1 gesetzt.

6.2.3 Skalierung von α durch das KNN

Anhand des KNN-Modells wird berechnet, welcher der höchste Wert für α ist, der bei der Kombination von β und γ nach dem Modell zu divergierendem Verhalten der IPF führt. Dieser Wert wird dann als unterer Grenzwert zur Skalierung der Werte verwendet, die der Nutzer für α wählen kann. Eine Neuskalierung von α findet immer bei einer Änderung von β oder γ statt.

6.3 Karplus-Strong Algorithmus

Der Karplus-Strong Algorithmus zur Dämpfung der Impulse ist mithilfe der von Max/MSP zur Verfügung gestellten Technik „Gen“ umgesetzt. Gen dient in Max/MSP als weitere Schnittstelle zwischen grafischer Entwicklung mittels Patching und dem Schreiben von Code. Ein Gen Objekt beinhaltet ein eigenes Patch, dessen grafische Oberfläche an normale Max/MSP-Patches angelehnt ist. Gleichzeitig lässt sich ein Gen-Patch aber auch als Code darstellen und bearbeiten (Cycling ’74 a).

Vorteil dieser Technik ist vor allem eine effiziente, samplegenaue Verarbeitung von Signalen (Cycling ’74 a). Diese samplegenaue Verarbeitung wird für die Implementierung des Karplus-Strong Algorithmus benötigt. Um das gewünschte Verhalten des Algorithmus zu erzielen, muss der darin enthaltene Delay exakt auf die Periodenlänge der Grundfrequenz und somit auf die Länge eines Impulses eingestellt sein.

Max/MSP stellt in seinen fertigen Beispiel-Patches ein solches, in Gen implementiertes Karplus-Strong Patch zur Verfügung. Dieses wurde für den Synthesizer übernommen. Die Dämpfung wird dabei so angepasst, dass sie für alle Frequenzen etwa gleich lang ist. Die Stärke der Dämpfung lässt sich über den Parameter „decay“ einstellen.

6.4 Bedienoberfläche

Abbildung 6.9 zeigt die fertige Bedienoberfläche des virtuellen Musikinstruments. Sie umfasst drei Drehregler, zwei Slider, vier Menüs sowie eine grafische Oberfläche zur Filtereinstellung. Hilfsweise kann der Nutzer auch die abgebildete Klaviatur verwenden, falls kein MIDI-Controller zur Verfügung steht. Das „mode“-Menü und der „flush“-Button beziehen sich auf diese Klaviatur.

alpha

Der alpha-Slider ist zuständig für die Steuerung des Anregungsparameters α. Ein Slider in Max/MSP gibt Werte zwischen 0 und 128 aus. Diese Werte werden umgerechnet in entsprechende Werte für α. Nach dieser Umrechnung und der Skalierung durch das KNN wird α als Parameter in die IPF zur Berechnung des Systemzustands g gegeben.

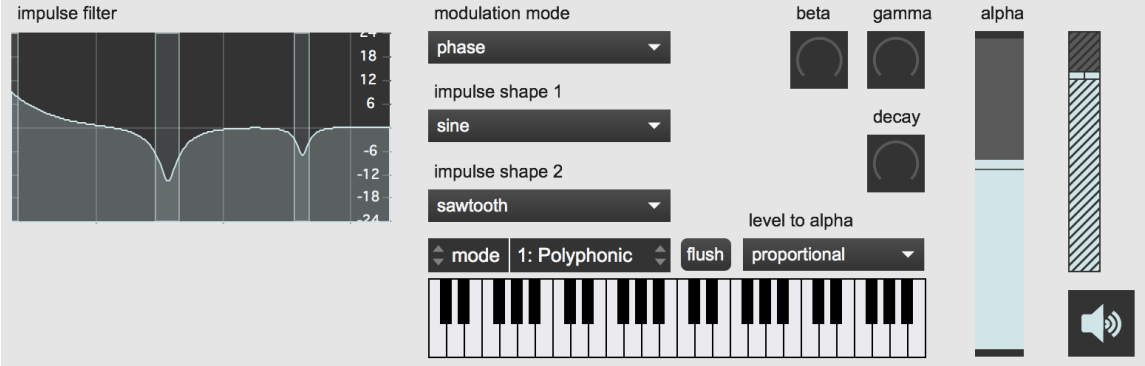

Abb. 6.9: Screenshot der fertigen Bedienoberfläche des virtuellen Musikinstruments

beta, gamma, decay

Die Einstellung der Reflektionskoeffizienten lässt sich mithilfe von Drehreglern vornehmen. Für den Drehregler von β wurde eine Untergrenze von 0 und eine Obergrenze von 0,3 voreingestellt. Der Maximalwert von γ Als Default wurde für beide Reflektionskoeffizienten ein Wert von 0 festgelegt, so dass die IPF mit nur einem Reflektionspunkt berechnet wird ($\beta = 0$, $\gamma = 0$).

Der „decay“-Regler bestimmt die Stärke, mit der ein verzögerter Impuls im Karplus-Strong Algorithmus als Feedback wieder in das Delay gespeist wird. Je stärker das Feedback ist, desto schwächer ist die Dämpfung und desto länger klingt ein Impuls nach.

level to alpha

In diesem Menü lässt sich festlegen, in welchem Verhältnis zu α die Lautstärke geregelt werden soll. Standardmäßig steigt die Lautstärke des erzeugten Tons proportional zu α. Zusätzlich besteht die Möglichkeit eines antiproportionalen Verhältnisses.

modulation mode

Im „modulation“-Menü kann zwischen den beiden Modulations-Modi umgeschaltet werden. Initial ist hier die Phasenmodulation eingestellt.

impulse shape 1, impulse shape 2

Darunter ist die Auswahl der beiden Impulsformen zu sehen. Es stehen jeweils Dreiecks-, Sägezahn- und Sinusschwingung zur Verfügung sowie eine Hamming-Fensterfunktion. Die Auswahl wurde anhand einer Auswahl von Funktionen getroffen, die Max/MSP direkt zur Verfügung stellt. Es besteht zusätzlich unter dem Punkt „upload“ die Möglichkeit, eine eigene wave-Datei als Impulsform hochzuladen. Standardmäßig ist in beiden Menüs eine Sägezahnschwingung gesetzt.

impulse filter

Auf der grafischen Filteroberfläche lässt sich das Spektrum des Impulses filtern. Der Filter ist im Signalfluss nach Wavetable-Auslesung und vor Karplus-Strong Dämpfung implementiert. Der Filter besteht aus drei kaskadierten Peak-Notch Filtern, deren Frequenz, Gain und Güte sich einstellen lässt.

7 Ergebnisse

Die Umsetzung des virtuellen Musikinstruments auf Grundlage der IPF hat die hier vorgestellten Ergebnisse hinsichtlich Funktionsumfang, Echtzeitfähigkeit, Nutzerfreundlichkeit und Erweiterungsfähigkeit erzielen können. Im Ausblick werden denkbare Verbesserungsvorschläge und Erweiterungsmöglichkeiten des Instruments genannt. Das fertige Max/MSP-Programm befindet sich auf der CD im Anhang unter dem Dateinamen „ipf_synthesizer“.

7.1 Funktionsumfang

Die grundsätzliche Funktion des virtuellen Musikinstruments konnte realisiert werden. Der Synthesizer ist in der Lage, Befehle vom Nutzer im Sinne der IPF als Klangparameter umzusetzen. Es lassen sich Werte für α und für die zwei Reflektionskoeffizienten β und γ mithilfe eines MIDI-Controllers steuern. Die Impulsform lässt sich durch den Nutzer festlegen und kann zusätzlich durch einen Filter bearbeitet werden. Die Stärke der Dämpfung ist einstellbar.

Der Synthesizer enthält weitere Modi, die mehr Möglichkeiten zur Klanggestaltung schaffen. So kann mit der IPF nicht nur die Phase der Impulse moduliert werden, sondern auch ihre Frequenz. Das Verhältnis von der Anregungsstärke α zur Lautstärke lässt sich von proportional auf antiproportional umkehren, so dass nach Belieben die obertonreichen und geräuschartigen Töne des Bifurkations- bzw. Chaosregimes lauter hörbar gemacht werden können.

In Abbildung 7.1 ist das normalisierte Ausgangssignal des virtuellen Musikinstruments für die drei Regimes dargestellt. Über den Ausgangssignalen ist jeweils der normalisierte Verlauf von Δg in blau aufgetragen. Als Impulsform wurden Sägezahnwellen gewählt sowie eine Frequenz von 65,41 Hz. Oben befindet sich die IPF im stabilen Zustand. Das Ausgangssignal entspricht dem unmodulierten Sägezahnsignal, Δg ist 0.

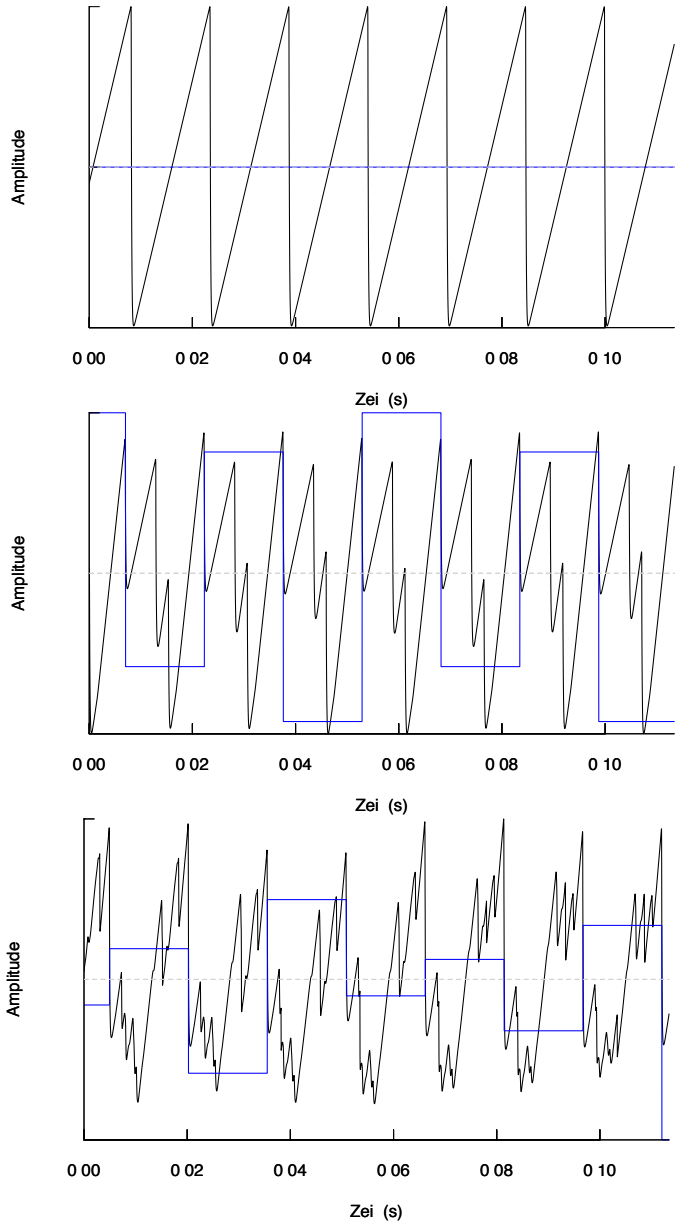

Abb. 7.1: Normalisierter Verlauf des Ausgangssignals des Synthesizers im stabilen Regime (oben), bei Bifurkationen zweiter Ordnung (mittig) und bei Chaos (unten). Über dem Ausgangssignal ist in blau der zeitgleiche normalisierte Verlauf von Δg aufgetragen. Bei der Aufzeichnung wurde für β und γ ein Wert von 0 gewählt sowie ein Sägezahnsignal mit einer Frequenz von 65,41 Hz und ein Decay-Wert von 1,17.

In der Mitte befindet sich die IPF im Regime von Bifurkationen. Es treten Bifurkationen zweiter Ordnung auf, Δg springt zwischen vier Zuständen hin und her. Es treten zusätzliche Sägezahn-Impulse während des Umlaufs einer Periode der Grund-

frequenz auf, welche als Obertöne hörbar sind. Im chaotischen Regime wechselt Δg zwischen verschiedenen Zuständen, ohne dass ein sich wiederholender Verlauf zu erkennen ist. Es ergibt sich somit ein Ausgangssignal mit einem Rauschanteil. Die Frequenz des Grundtons ist aber noch erkennbar und hörbar.

In Abbildung 7.1 ist zudem erkennbar, dass das virtuelle Musikinstrument bei der Umsetzung der IPF in Klangparameter zeitlich exakt arbeitet. Es ist zu sehen, dass Δg sich exakt im Takt der Grundfrequenz verändert und ein Zustandssprung zeitgleich mit dem Steigungswechsel des größten Sägezahnpeaks geschieht.

Eine Veränderung des Anregungsparameters α führt wie eingangs vorgestellt zu einer Bildung von Transienten. Das Ausgangssignal des virtuellen Musikinstruments bei einer Erhöhung von α durch den Nutzer ist in Abbildung 7.2 aufgetragen. Amplitudenverlauf des Ausgangssignals (schwarz) und Verlauf von α (blau) sind normalisiert dargestellt. α wurde für die Aufzeichnung schrittweise von 0,4 auf 0,6 erhöht. Es ist erkennbar, dass erst nach einem anfänglichen Einschwingvorgang ein stabiler Sägezahn-Ton etabliert werden kann.

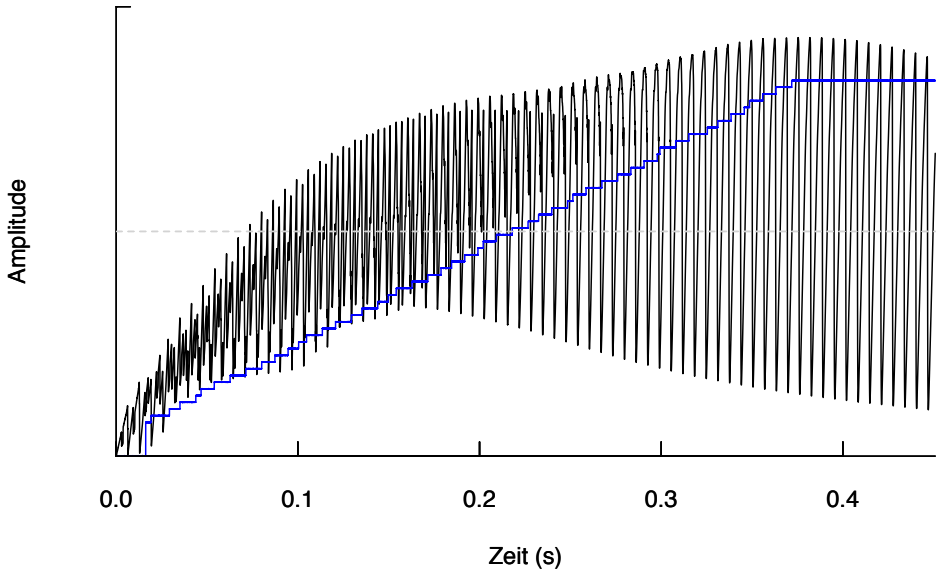

Abb. 7.2: Normalisierter Verlauf des Ausgangssignals bei einer schrittweisen Erhöhung von α von 0,4 auf 0,6. Über dem Ausgangssignal ist in blau der normalisierte Verlauf von α aufgetragen. Nach einem anfänglichen Einschwingvorgang entsteht ein stabiles Sägezahnsignal. Für die Aufzeichnung wurde $\beta = 0$ und $\gamma = 0$ gewählt sowie ein Dämpfungswert von 1,17.

7.2 Latenz

Durchführung der Messung

Die Messung der Latenz wurde mit dem in Abbildung 7.5 dargestellten Aufbau durchgeführt. Als Triggersignal der Klangsynthese wurde in einem Durchlauf die Betätigung der Leertaste der Computertastatur verwendet und in einem zweiten Durchlauf der Anschlag einer externen MIDI-Klaviatur. Pro Durchlauf wurden 20 Messungen durchgeführt.

Das Triggersignal wurde jeweils von einem Mikrofon in einem vernachlässigbar geringen räumlichen Abstand zur Triggerbetätigung aufgenommen. Das Triggersignal wird über ein Interface in den Computer gegeben und im ersten Kanal einer Aufnahme gespeichert.

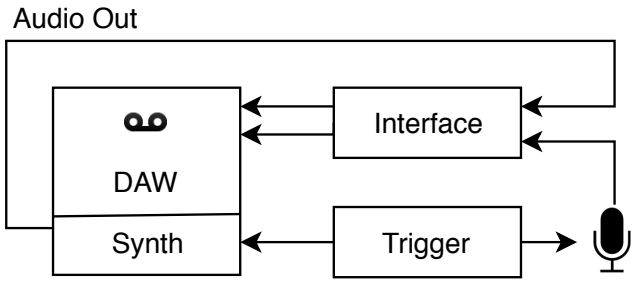

Abb. 7.3: Funktionelle Darstellung des Messaufbaus. Das Triggersignal wird mit einem Mikrofon über ein Interface in den Computer gegeben und dort im ersten Kanal einer Audiodatei aufgenommen. Das Audio-Ausgangssignal des Computers wird über das Interface zurück in den Computer gespeist und im zweiten Kanal der Audiodatei aufgenommen.

Zugleich wird der Audio-Ausgang des Computers über das Interface wieder in den Computer gespeist und in einem zweiten Audiokanal aufgenommen. Aus der Aufnahme der zwei Signale lässt sich direkt die Latenz ablesen. Hierzu wurde in Audacity manuell die zeitliche Differenz zwischen der Spitze des Triggersignals und dem Beginn des synthetisierten Tons ausgemessen. Ein Beispiel für den manuellen Messvorgang in Audacity ist in Abbildung 7.4 zu sehen. Da die Latenz des Audio-Interfaces sowohl im Trigger- als auch im Ausgangssignal vorhanden ist, rechnet sie sich in der Differenz raus und muss somit nicht extra berücksichtigt werden.

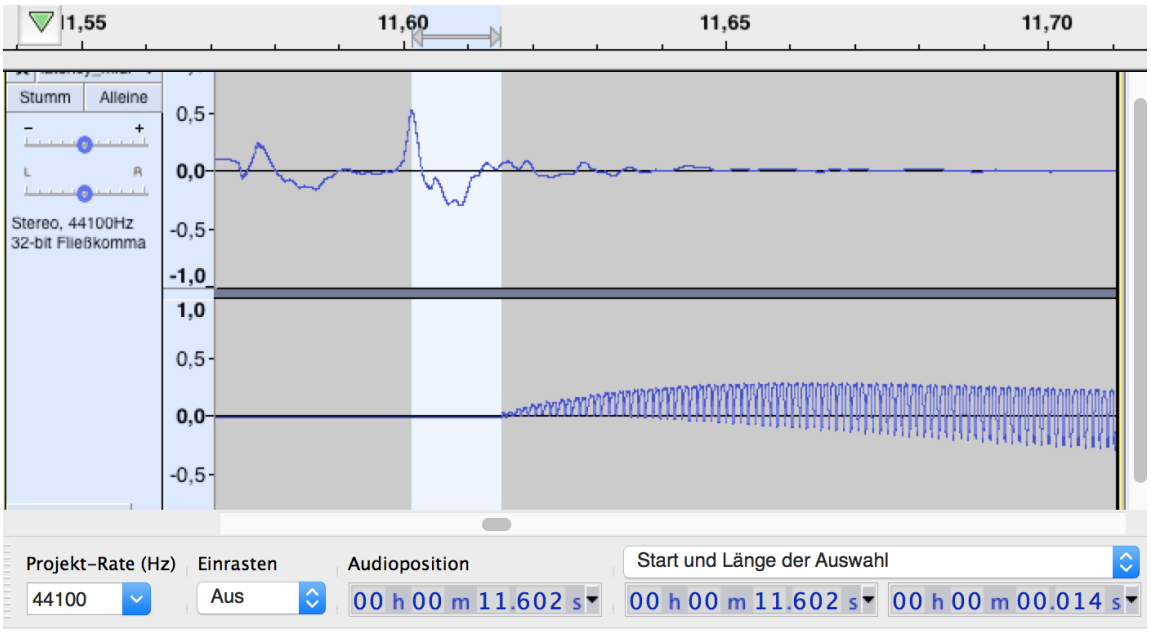

Abb. 7.4: Screenshot des manuellen Messvorgangs in Audacity. Der Beginn der Auswahl wird auf die Spitze des Triggersignals in Kanal 1 gesetzt, das Ende auf den Beginn des Ausgangssignals in Kanal 2. Rechts unten im Fenster wird die Länge der Auswahl und somit die Latenz angezeigt.

Messergebnisse

Die Ergebnisse der beiden Messdurchläufe sind in Abbildung 7.5 dargestellt. Für die Computer interne Steuerung hat sich ein Median von 15,5 ms für die Latenz ergeben. Für die externe Steuerung mit einem MIDI Controller konnte im Median eine Latenz von 14 ms gemessen werden. Die anfangs gestellten Anforderungen an die Latenz können also erfüllt werden. Das virtuelle Musikinstrument lässt sich als echtzeitfähig bezeichnen.

Es zeigt sich, dass die Steuerung mit einem MIDI Controller eine deutlich niedrigere Latenz ergibt als die Steuerung mit der Maus. Eine wahrscheinliche Erklärung hierfür ist, dass das grafische Schnittstellen-Objekt zur Steuerung mit der Maus mit einer deutlich höheren Latenz auf einen Mausklick reagiert als der MIDI Controller auf einen Tastendruck. Als MIDI Controller wurde ein Yamaha YDP-143 B Arius verwendet. Der Hersteller hat keine Latenz für das Gerät angegeben.

Wie hoch die Latenz des virtuellen Musikinstruments allein ist, lässt sich also nicht genau beziffern. Es kann aber dennoch festgestellt werden, dass die Latenz den

Anforderungen an ein Echtzeitsystem im Audiobereich gerecht wird.

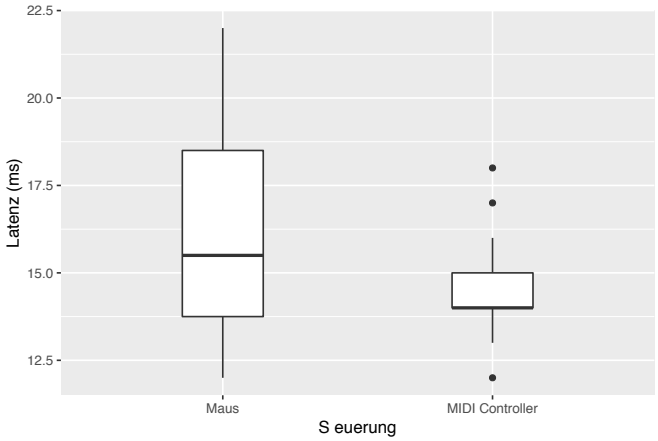

Abb. 7.5: Gemessene Latenz bei Computer interner Steuerung (links) sowie bei Verwendung eines MIDI-Controllers (rechts). Es ergibt sich ein Median von 15,5 ms bzw. 14 ms.

7.3 Weitere Anforderungen

Neben der Echtzeitfähigkeit und dem geforderten Funktionsumfang konnten auch die meisten Anforderungen an die Nutzerfreundlichkeit und Erweiterbarkeit des Synthesizers erfüllt werden.

Nutzerfreundlichkeit

Bei Initialisierung des virtuellen Musikinstruments wird dieses im sog. „presentation mode“ geöffnet. In diesem Modus ist die Bedienoberfläche auf ein übersichtliches Minimum von Elementen reduziert, die für die Bedienung benötigt werden. Die Elemente erinnern in ihrer Gestalt größtenteils an verbreitete Steuerungselemente von Synthesizern. Es wurden Drehknöpfe, eine Filterkurve und Slider verwendet.

Wenn der Nutzer ein Bedienelement steuert, wird das Ergebnis seiner Interaktion auf der Bedienoberfläche sichtbar. Für die Bedienelemente wurden jeweils sinnvolle Startwerte voreingestellt, so dass der Synthesizer sofort in Betrieb genommen werden kann.

Die grafischen Schnittstellenelemente zeigen den aktuellen Status der Bearbeitung an. Falls der Nutzer den Synthesizer individuell erweitern möchte, kann eine erweiterte Darstellungsform gewählt werden. Hierzu muss lediglich der „presentation mo-

de“ deaktiviert werden. Der Nutzer erhält dann einen Einblick auf die komplette Struktur und Funktionsweise des Programms und kann diese beliebig anpassen.

Um zu verhindern, dass das Programm beim Auftreten von komplexen Werten aus der IPF abstürzt, wird dieser Fall im Synthesizer abgefangen und führt lediglich zu einem Verstummen des zuvor gespielten Tons. Zusätzlich wird die Wahrscheinlichkeit des Auftretens solcher Werte durch die Skalierung des Künstlichen Neuronalen Netzwerks erheblich reduziert. Diese Maßnahmen sorgen für eine gute Spielbarkeit des Instruments.

Allerdings sind Einbußen in der Robustheit des Programms im Laufe des Entwicklungsprozesses entstanden. Mit steigender Komplexität, Strukturierung und Exaktheit des Programms, ist es zu einem leicht erhöhten Aufkommen von Abstürzen gekommen. Eine mögliche Erklärung für die Abstürze ist die Verwendung der achten Version von Max/MSP. Diese wurde im September 2018 veröffentlicht (Cycling ’74 c). Es gab also noch nicht allzu viel Zeit zur Beseitung von möglichen, durch die Nutzer beobachteten Fehlerquellen. Der Synthesizer kann trotz gelegentlicher Abstürze dennoch im Regelfall zur Klangsynthese verwendet werden.

Erweiterbarkeit

Um die Hürden zur Erweiterung des Max/MSP-Programms möglichst gering zu gestalten, wurde versucht, das Programm nach den Grundsätzen der Abstraktion, Hierarchisierung und Trennung von Funktionseinheiten nach Zuständigkeiten zu strukturieren. Eine solche Strukturisierung ist für einen Großteil des Programms gelungen.

Hierarchisch wurde das Programm in mehrere Ebenen unterteilt. Auf oberster Ebene befinden sich die Bedienelemente. Auf darunter liegenden parallelen Ebenen befinden sich der Klangsyntheseweg und die Skalierung durch das KNN. Die Unterprogramme auf diesen Ebenen sind größtenteils nach Funktionalitäten getrennt.

In Abbildung 7.6 ist die Struktur des Klangsynthesewegs einer einzelnen Stimme abgebildet. Die Struktur ist an das funktionelle Diagramm in 6.1 angelehnt. Eine klare Gliederung nach Funktionseinheiten ist somit im Programm erkennbar. Innerhalb der einzelnen Funktionseinheiten wurden die Verarbeitungsschritte möglichst auskommentiert.

Eine Erweiterbarkeit nach den genannten Kriterien konnte zumindest für einen Teil des Programms erreicht werden.

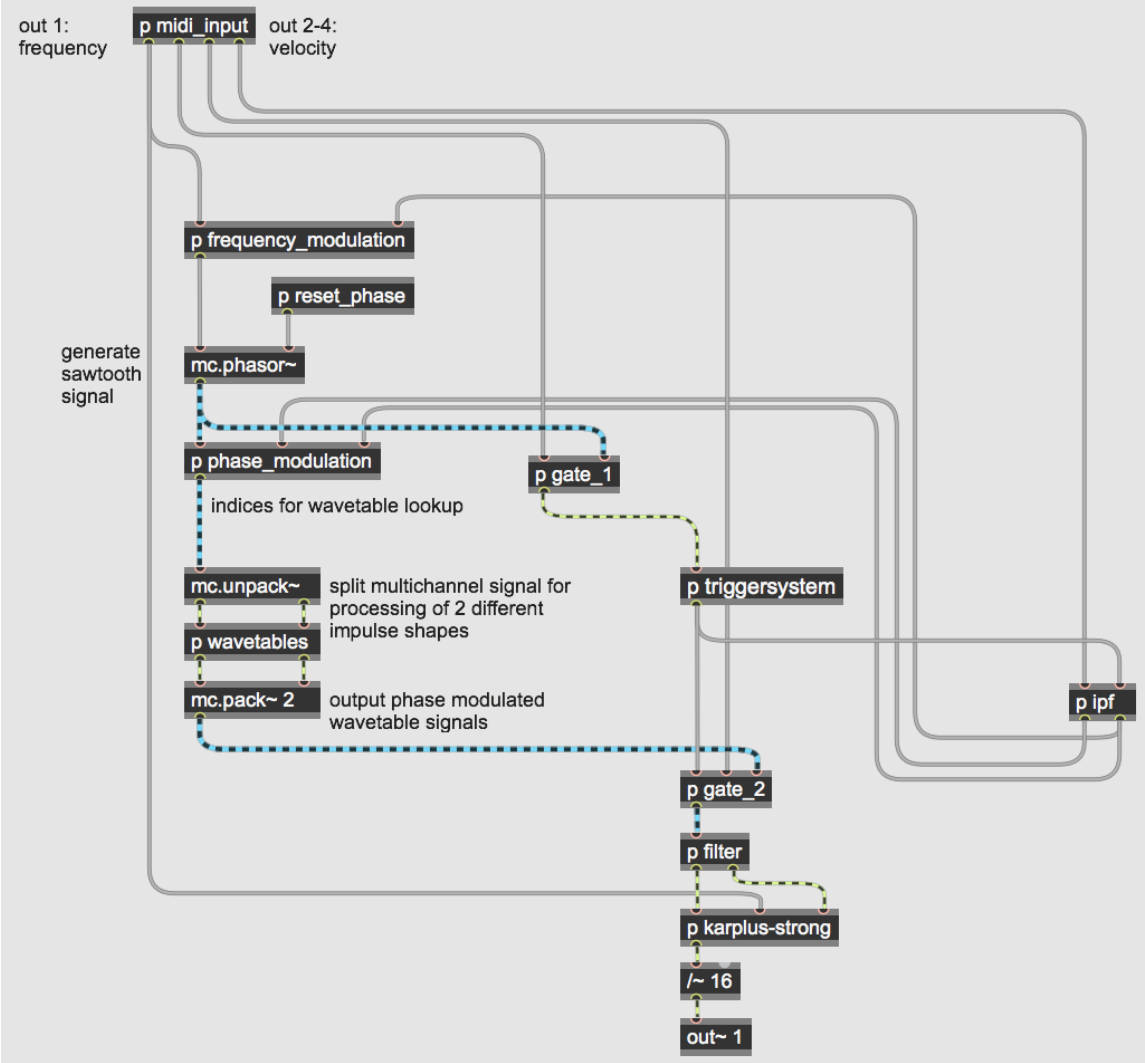

Abb. 7.6: Screenshot der Programmstruktur auf der Ebene des Klangsynthesewegs einer einzelnen Stimme. Das Programm ist in Funktionseinheiten gegliedert und stellenweise zur Erklärung auskommentiert.

7.4 Ausblick

Im virtuellen Musikinstrument ist die Steuerung von wesentlichen Parametern der IPF umgesetzt. Systemzustände, die sich aus der IPF ergeben, werden zur Klanggestaltung verwendet. Um den Funktionsumfang des virtuellen Musikinstruments über diesen Kern hinaus zu erweitern, könnte eine Steuerung von weiteren Parametern der IPF implementiert werden. Denkbar wäre eine Umsetzung von mehr als zwei Re-

flektionskoeffizienten sowie eine Miteinbeziehung mehrerer Perspektiven, anstatt der Betrachtung von nur einem schwingenden Untersystem des Instruments.

Die Verwendung der Systemzustände zur Klanggestaltung könnte um eine Amplitudenmodulation erweitert werden. Ein Einfluss des Systemzustands auf die Amplitude wurde bereits von Bader (2013: S.) postuliert. Diese wurde im Synthesizer allerdings noch nicht umgesetzt.

Zur Dämpfung der Impulse wurde ein Karplus-Strong Algorithmus verwendet. Denkbar wäre auch eine andere Umsetzung der Dämpfung. In der Vorstellung der IPF von Bader (2013: S.) ist eine Dämpfung nicht konkretisiert, ihre Notwendigkeit ergibt sich erst aus den gewünschten Klangergebnissen. Daher wäre eine weitere Untersuchung von Dämpfungsmöglichkeiten sinnvoll.

Um die Anbindung des virtuellen Musikinstruments an verbreitete Software zur Musikproduktion zu ermöglichen, könnte das Programm in der „Max for Live“ Umgebung implementiert werden. Es ließe sich so an Ableton Live anbinden. Auch eine komplette Umsetzung in der Max-Technik Gen ist denkbar. Damit könnte ermöglicht werden, das Programm in eine virtuelle Studiotechnologie (VST) zu kompilieren, die als Plug-in ohne die Verwendung der Entwicklungsumgebung Max/MSP funktioniert.

Eine Schwachstelle des Synthesizers ist die Umsetzung des Künstlichen Neuronalen Netzwerkes. Da die Auslesung des Modells nicht korrekt ausgeführt wird, muss das KNN bei Initialisierung des Programms stets neu trainiert werden. Zur Vermeidung dieser Initialisierungszeit wäre zu empfehlen, die Umsetzung des KNNs auszulagern, z.B. mithilfe der Tensorflow API von Google. Eine Anbindung des Programms an die Tensorflow API von Google ist mithilfe der von Max/MSP zur Verfügung gestellten Techniken möglich.

8 Fazit

Die Realisierung des virtuellen Musikinstruments auf Basis der Impulse Pattern Formulation war erfolgreich. Der Systemzustand, der sich aus der IPF ergibt, musste zur Umsetzung in Klangparameter erst interpretiert werden. Die IPF bildet somit einen abstrakten, systematischen Ansatz für die Grundlage einer Klangsynthese.

Eine Konkretisierung des systematischen Ansatzes konnte anhand der Geige vorgenommen werden. Als problematisch bei der Umsetzung stellte sich vor allem das Auftreten von komplexen Systemzuständen heraus, deren Bedeutung unbekannt ist und nicht interpretiert wird. Um solche Zustände zu vermeiden, wurde ein Künstliches Neuronales Netzwerk entwickelt. Das KNN-Modell hilft dabei, diese unbekannten Zustände zu vermeiden. Das Problem kann dennoch nicht komplett umgangen werden. Eine kompetente Entwicklung des KNN-Modells gestaltet sich als relativ aufwendig. Der Aufwand hierfür steht in einem recht hohen Verhältnis zum Nutzen.

Interessant am virtuellen Musikinstrument ist in erster Linie das Ergebnis des Klangs. Die IPF leistet eine Gestaltung des Klangs, die sich von herkömmlichen Syntheseverfahren unterscheidet. Da die IPF ein iterativer Berechnungsansatz zur Gestaltung von Klangparametern ist und ihre Ergebnisse nicht nur von der aktuellen Steuerung sondern auch von deren Verlauf abhängig sind, sind die Klangergebnisse nicht komplett vorhersehbar und können sich bei scheinbar gleicher Ansteuerung unterscheiden. Eine Nutzung des Synthesizers wird vor allem durch diese Eigenschaft interessant. Der Klang verliert an Sterilität, die normalerweise mit digitalen Synthesizern assoziiert wird.

Eingeschränkt wird die Gestaltung des Klangs unter anderem durch die fixe Nutzung des Karplus-Strong Algorithmus zur Dämpfung der Impulse. Lediglich die Stärke der Dämpfung lässt sich durch einen einzigen Parameter einstellen. Das führt zu einer für den Algorithmus charakteristischen Klangfärbung aller Töne.

Insgesamt bildet der entwickelte Kern eine in sich gut abgeschlossene Funktionseinheit, die gute Möglichkeiten zur Erweiterung bietet. Neben den im Ausblick bereits

genannten Erweiterungsmöglichkeiten könnten auch experimentellere Interpretationen der IPF interessant zur Klanggestaltung sein. Als Beispiel einer experimentellen Interpretation wurde eine Frequenzmodulation durch die IPF umgesetzt, die lediglich die Grundfrequenz moduliert, nicht aber die im Karplus-Strong Algorithmus verwendete Frequenz, in der die Impulse gedämpft werden.

A Anhang

Auf der CD im Anhang befinden sich folgende Dateien:

1. Max-Patches: Das Hauptprogramm „ipf_synthesizer.maxpat“ startet das virtuelle Musikinstrument. Es greift auf die Unterprogramme „karplus-strong-ba.gendsp“, „knn.maxpat “ und „osc_karplus_strong.maxpat“ zu.

2. „ml.mlp.mxo“/„ml.mlp.mxe“: Mac- bzw. Windowsversion des externen Objekts zur Ausführung des Multilayer-Perzeptrons.

3. „mlp_ipf.data“: Datensatz, der beim Training des KNN verwendet wird.

4. „mlp_ipf.model“: Modell, das vom KNN gebildet wird.

Das virtuelle Musikinstrument ist in zwei Versionen ausführbar, in einer Mac- sowie in einer Windowsversion.

Abbildungsverzeichnis

Tabellenverzeichnis

Literaturverzeichnis

[Bader 2013] Bader, R.: *Nonlinearities and Synchronization in Musical Acoustics and Music Psychology*. Springer Berlin Heidelberg, 2013 (Current Research in Systematic Musicology). https://books.google.de/books?id=xQhHAAAAQBAJ. ISBN 9783642360985

[Brcina u. a. 2009] Brcina, R. ; Bode, S. ; Riebisch, M.: Optimisation process for maintaining evolvability during software evolution. In: *Engineering of Computer Based Systems, 2009. ECBS 2009. 16th Annual IEEE International Conference and Workshop on the* IEEE, 2009, S. 196 205

[Bullock u. Momeni 2015] Bullock, J. ; Momeni, A.: Ml. lib: robust, cross-platform, open-source machine learning for max and pure data. In: *NIME*, 2015, S. 265 270

[Cremer 1981] Cremer, L.: *Physik der Geige*. Hirzel, 1981 https://books.google.de/books?id=2LxGAAAAMAAJ. ISBN 9783777603728

[Cycling '74 a] Cycling '74: *Gen*. https://docs.cycling74.com/max7/vignettes/gen_overview, . [Online; abgerufen am 19.01.2019]

[Cycling '74 b] Cycling '74: *Information and contact*. https://cycling74.com/company/, . [Online; abgerufen am 13.01.2019]

[Cycling '74 c] Cycling '74: *Max 8 is here*. https://cycling74.com/articles/max-8-is-here, . [Online; abgerufen am 04.02.2019]

[Cycling '74 d] Cycling '74: *Max For Live*. https://docs.cycling74.com/max7/vignettes/max_for_live_topic, . [Online; abgerufen am 25.01.2019]

[Cycling '74 e] Cycling '74: *MSP: Audio Input and Output*. https://docs.cycling74.com/max7/tutorials/04_mspaudioio, . [Online; abgerufen am 19.01.2019]

[Cycling '74 f] Cycling '74: *Using Max with Other Applications*. https://docs.cycling74.com/max7/vignettes/max_and_other_apps, . [Online; abgerufen am 25.01.2019]

[Deutsches Institut für Normung 2006] Deutsches Institut für Normung: *DIN EN ISO 9241: Ergonomie der Mensch-System-Interaktion: Teil 110: Grundsätze der Dialoggestaltung*. 2006

[FredVoisin 2017] FredVoisin: *mlp read message load data but not model*. https://github.com/irllabs/ml-lib/issues/136, 2017. [Online; abgerufen am 16.01.2019]

[Gurney 1997] Gurney, K.: *An Introduction to Neural Networks*. Taylor & Francis, 1997 https://books.google.de/books?id=HOsvllRMMP8C. ISBN 9781857285031

[Lago u. Kon 2004] Lago, N. ; Kon, F.: The Quest for Low Latency. In: *Proceedings of the International Computer Music Confernence* Bd. 30, ICMA, 2004, S. 33 36

[Linke u. a. 2019] Linke, S. ; Bader, R. ; Mores, R.: *The Impulse Pattern Formulation (IPF) as a model of musical instruments - investigation of stability and limits*. 2019. Manuskript

[McIntyre u. Woodhouse 1984] McIntyre, M. E. ; Woodhouse, J.: A parametric study of the bowed string: the violinist's menagerie. In: *Journal of the Catgut Acoustical Society* 42 (1984), S. 18 21

[Mores 2014] Mores, R.: Saiteninstrumente, Laaber, 2014 (Handbuch der systematischen Musikwissenschaft). ISBN 9783890076997, S. 319 368

[Norman 2002] Norman, Donald A.: *The Design of Everyday Things*. New York, NY, USA : Basic Books, Inc., 2002. ISBN 9780465067107

[Rashid u. Langenau 2017] Rashid, T. ; Langenau, F.: *Neuronale Netze selbst programmieren: Ein verständlicher Einstieg mit Python*. O'Reilly, 2017 (Animals). https://books.google.de/books?id=b9N3DwAAQBAJ. ISBN 9783960101031

[Reed u. Marks II 1999] Reed, R. ; Marks II, R.J.: *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press, 1999 (A

Bradford Book). https://books.google.de/books?id=7C4TDgAAQBAJ. ISBN 9780262181907

[Roads u. Strawn 1996] Roads, C. ; Strawn, J.: *The Computer Music Tutorial*. MIT Press, 1996 (The Computer Music Tutorial). https://books.google.de/books?id=nZ-TetwzVcIC. ISBN 9780262680820

[Ruschkowski 1998] Ruschkowski, A.: *Elektronische Klänge und musikalische Entdeckungen*. Reclam, 1998 (Reclams Universal-Bibliothek). https://books.google.de/books?id=Y6rPAAAACAAJ. ISBN 9783150096635

[Serra u. Smith 1990] Serra, Xavier ; Smith, J.: Spectral Modeling Synthesis: A Sound Analysis/Synthesis Based on a Deterministic plus Stochastic Decomposition. In: *Computer Music Journal* 14 (1990), 12-24. http://dx.doi.org/http://doi.org/10.2307/3680788. DOI http://doi.org/10.2307/3680788. SMS

[Smith u. Garnett 2012] Smith, B. D. ; Garnett, G. E.: Unsupervised Play: Machine Learning Toolkit for Max. In: *NIME*, 2012

[Smith 2019] Smith, Julius O.: *Physical Audio Signal Processing*. https://ccrma.stanford.edu/~jos/pasp/Karplus_Strong_Algorithm.html, 2019. online book, 2010 edition

[Vail 2014] Vail, M.: *The Synthesizer: A Comprehensive Guide to Understanding, Programming, Playing, and Recording the Ultimate Electronic Music Instrument*. Oxford University Press, 2014 https://books.google.de/books?id=fFapAgAAQBAJ. ISBN 9780199334858

[Wörn u. Brinkschulte 2006] Wörn, H. ; Brinkschulte, U.: *Echtzeitsysteme*. Springer Berlin Heidelberg, 2006 (EXamen. press Series). https://books.google.de/books?id=tNcAyjeaASkC. ISBN 9783540274162

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben eindeutig kenntlich gemacht.

Ort, Datum Nadja Rutsch