

# Proseminar Computer and Music – Synthesizer

Alexander Bartolomey

Hendrik Höfert

4. Februar 2018

## Zusammenfassung

Wir geben eine Einführung in die Tonsynthese und einen Überblick über einige Methoden der Klangsynthese. Nach einem Exkurs über Filter gehen wir noch kurz auf die praktische Implementation eines Software-Synthesizers ein.

## Inhaltsverzeichnis

<b>1</b>	<b>Tonsynthese</b>	<b>2</b>
1.1	Grundbaustein: Oszillator . . . . .	2
1.2	Elementare Wellenformen . . . . .	2
1.2.1	Sinus . . . . .	2
1.2.2	Triangle – Dreieck . . . . .	3
1.2.3	Saw(tooth) – Sägezahn . . . . .	3
1.2.4	Square – Rechteck . . . . .	3
<b>2</b>	<b>Synthesearten</b>	<b>4</b>
2.1	Additive Synthese . . . . .	4
2.2	Subtraktive Synthese . . . . .	4
2.3	Modulationssynthese . . . . .	6
2.3.1	Amplitudenmodulation . . . . .	7
2.3.2	Frequenzmodulation . . . . .	7
<b>3</b>	<b>Klangformung und Modulation</b>	<b>8</b>
3.1	Hüllkurven . . . . .	8
3.2	Low Frequency Oscillators (LFOs) . . . . .	8
<b>4</b>	<b>Filter</b>	<b>10</b>
<b>5</b>	<b>Implementation von Monoid</b>	<b>12</b>
5.1	Probleme bei der Implementation . . . . .	13
5.2	Einschätzung und Ausblick . . . . .	13

# 1 Tonsynthese

Die Tonsynthese, also das künstliche Erzeugen von Tönen, bildet den ersten Schritt zur Erzeugung von Musik mit Synthesizern. Dieser erste Schritt ist erstmal unabhängig von dem Ziel, Musik zu komponieren, da in diesem Schritt zuerst beliebige Schwingungen erzeugt werden, mittels analoger oder digitaler Synthese, wobei Ersterere zunehmend durch die digitale Tonsynthese ersetzt wird, da diese Synthesizer flexibler einzusetzen sind. Das Grundelement, von dem man in beiden Implementationen aber immer ausgeht, ist das folgende Bauteil:

## 1.1 Grundbaustein: Oszillator

Das Kernstück der Klangerzeugung durch einen Synthesizer sind ein oder mehrere Oszillatoren. Oszillatoren sind Schwingungserzeuger und somit in der Lage, Signale zu erzeugen. In analogen Synthesizern erzeugt ein Oszillator ein elektrisches Wechsellspannungssignal, das an Lautsprecher angelegt werden kann, um einen Ton zu erzeugen. In digitalen Software-Synthesizern wird dieses Verhalten nur nachgeahmt, da alle Werte des Signals nur diskret vorliegen. Ein digitaler Oszillator berechnet also Samplingpunkte<sup>1</sup>, die via Digital-Analog-Wandlung erst in ein analoges Spannungssignal und dann mit Lautsprechern in ein akustisches Signal übersetzt werden.

Einen Oszillator nennen wir jedes Modul, das ein durch die folgenden Kenngrößen kategorisiertes Signal erzeugt:

1. Die *Frequenz*  $f$  gibt die Geschwindigkeit an, mit der der Oszillator schwingt. Die Frequenz der Oszillation hängt direkt mit der wahrgenommenen Tonhöhe zusammen.
2. Die *Amplitude*  $A$  bezeichnet den maximalen Wert, den das Signal annimmt. Sie ist ausschlaggebend für die Lautstärke des Tons.
3. Die *Phase*  $\varphi = (2\pi f \cdot t)$  der Schwingung bezeichnet den momentanen Punkt der Oszillation auf dem Einheitskreis. Sie ist für die weitere Betrachtung nur nebensächlich, es ist aber noch anzumerken, dass es eine integrale Beziehung zwischen Phase und Frequenz einer Schwingung gibt.

## 1.2 Elementare Wellenformen

Im Folgenden werden einige der grundlegenden Wellenformen bei der Tonsynthese erläutert. Diese Wellenformen sind unabhängig von Frequenz und Amplitude, was es erlaubt, sie zu kategorisieren.

### 1.2.1 Sinus

Der grundlegendste Oszillator ist der *Harmonische Oszillator*. Seine Schwingungsform ist die einer Sinuskurve, was der Definition einer harmonischen Schwingung

---

<sup>1</sup>Samplingpunkte sind die zeitdiskreten Werte eines quantisierten Signals

entspricht. Eine solche Schwingungsmenge bezeichnen wir als Sinusoide, mit der allgemeinen Funktionsgleichung  $x(t) := A \cdot \sin(\omega \cdot t + \varphi_0)$  mit der Kreisfrequenz  $\omega := 2\pi f$  und  $\varphi_0$  der Nullphase der Schwingung. Das Spektrum eines solchen Sinusoids weist nur eine einzige Harmonik auf (s. Abbildung 1a), was akustisch als sehr gleichmäßig wahrgenommen wird. Wir nennen eine solche Schwingung auch *Elementarschwingung*.

### 1.2.2 Triangle – Dreieck

Eine dem Sinus sehr ähnliche Wellenform ist die *Dreiecksschwingung*, jedoch weist das Spektrum der Dreiecksschwingung alle ungeraden Harmonischen der Grundfrequenz der Schwingung auf, deren Amplituden aber sehr schnell (die  $n$ -te Harmonische hat die relative Amplitude  $\frac{1}{n^2}$ ) abfallen, sodass die Grundschiwingung die Dominanz über den Klang hat (s. Abbildung 1b) [2].

### 1.2.3 Saw(tooth) – Sägezahn

Die *Sägezahnschwingung*, auch Rampe, weist hingegen ein sehr reichhaltiges, harmonisches Spektrum auf, mit Maxima genau an den ganzzahligen Vielfachen der Grundfrequenz. Diese Spitzen fallen invers zur Vielfachheit der Grundfrequenz ab, d.h. die  $n$ -te Harmonische hat die relative Amplitude von  $\frac{1}{n}$  (s. Abbildung 1c) [2]. Sägezahnschwingungen werden zur Synthese von Klängen wie Streichinstrumenten oder Blechbläsern verwendet.

### 1.2.4 Square – Rechteck

Die *Rechteckschwingung*<sup>2</sup> zeichnet sich in ihrer Form durch periodische Sprünge zwischen minimaler und maximaler Amplitude aus. Im Fall der *square wave* bedeutet dies, dass eben diese periodischen Sprünge genau mit der Frequenz  $2 \cdot f_0$  auftreten, dass also die Zeit der minimalen Amplitude gleich der Zeit der maximalen Amplitude ist, wohingegen andere Varianten die Frequenz der Sprünge verändern. Zum Beispiel erhalten wir einen sogenannten *Pulse* durch eine Zeit  $t_{hi}$  auf der max. Amplitude und  $t_{lo}$  auf der min. Amplitude mit  $t_{hi} < \frac{1}{2} \cdot T$ , mit  $T$  der Periode und  $t_{lo} = T - t_{hi}$ . Das Spektrum der Rechteckschwingung besteht aus allen ungeraden Vielfachen der Grundfrequenz. (s. Abbildung 1d). Rechteckschwingungen werden in verschiedenen Situationen benutzt, meist zur Synthese von Klängen ähnlich Holzbläsern [2].

### Noise – Rauschen

Das Rauschen stellt eine besondere Art von Klang für die Tonsynthese dar, da es einerseits keine periodische Schwingungsfunktion ist und andererseits über das Spektrum und die Quelle, die es erzeugt, definiert wird. Rauschen entsteht durch Samplewerte, die, anders als bei den bisherigen Wellenformen, *unkorreliert* sind,

---

<sup>2</sup>Die Übersetzung scheitert hier, Rechteck, engl. Rectangle, bezeichnet mehr als nur die hier vorgestellte Schwingungsfunktion

d.h. im diskreten Szenario sind die Samplewerte zufällig verteilt. Rauschen wird ferner in der Frequenzverteilung im Spektrum kategorisiert, das heißt, analog zu den wahrgenommenen Farben von Licht wird Rauschen als *weißes Rauschen* bezeichnet, wenn alle Frequenzen im Spektrum des Rauschens gleichverteilt sind. Analog erfolgen die Definitionen von weiteren Kategorien, wie *Braunem Rauschen* oder *Rosa Rauschen*, die jeweils über kontinuierliche Frequenzspektren mit abgeschwächten Bereichen definiert werden [7].

## 2 Synthesarten

Als Nächstes wollen wir uns etwas genauer mit zwei grundlegend verschiedenen Arten der Klangsynthese beschäftigen. Anschließend zeigen wir, wie durch Modulation einzelner Parameter von Synthesizer-Bausteinen ebenfalls wieder Tonsynthese betrieben wird.

### 2.1 Additive Synthese

Bei der *Additive Synthese* werden Klänge durch Kombination von Elementarschwingungen erzeugt, mit dem Grundsatz, dass sich *jeder mögliche Klang* durch Kombination ebendieser Elementarschwingungen erzeugen lässt. Alle Parameter einer jeden Schwingung, welche dem zu synthetisierenden Klang hinzugefügt wird, können einzeln angepasst werden. Für komplexe Klänge ist dies auch zwangsläufig notwendig, bringt allerdings auch einen hohen Aufwand durch Feinjustierung, welche evtl. auch noch effektlos auf das Klangbild erscheint, mit sich [9, p.134].

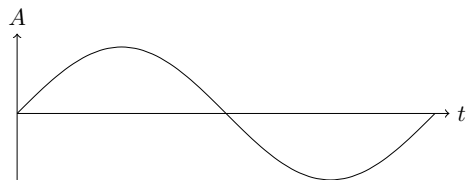
Das Konzept dieser Tonsyntheseart ist recht intuitiv, es sollte jedoch angemerkt werden, dass Kenntnisse in Musiktheorie erforderlich sind, da Interferenzeffekte und Dissonanzen - sofern diese nicht gewollt sind - schnell auftreten, denn für harmonische Klänge kommen nur bestimmte Vielfachheiten der Frequenzen infrage.

Die additive Synthese einer Sägezahnwelle ist beispielhaft in Abbildung 2, sowie das Frequenzspektrum in Abbildung 1c, dargestellt.

Jedoch kann durch diese Syntheseart ein sauberes Frequenzspektrum erreicht werden, da jede einzelne Frequenz durch die Verwendung von Sinusoiden kontrolliert ausgewählt und hinzugemischt wird. Dies wiederum führt zu weniger Interferenzeffekten zwischen Frequenzen, sobald Kompositionen mit mehr als einem Instrument respektive Synthesizer erstellt werden, in denen ein Zusammenspiel dieser gewünscht ist [2].

### 2.2 Subtraktive Synthese

Das Ausgangsmaterial dieser Syntheseart sind meist keine Elementarschwingungen, wie zuvor bei der *Additiven Synthese*, sondern eine aufgenommene oder synthetisierte Wellenform, bzw. ein - durch laufende Oszillation erzeugtes - Signal. Diese Wellenformen werden meist zu so genannten *Wavetables* zusammengefasst.

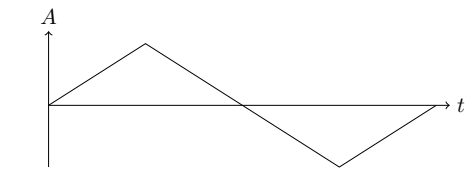


Exempl. Signalperiode einer Sinusschwingung

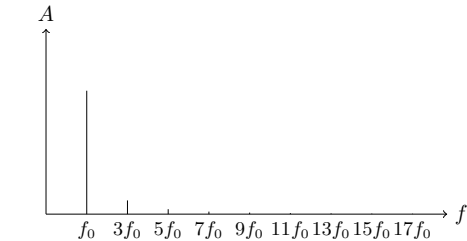


Exempl. Spektrum einer Sinusschwingung

(a) Zeit- und Frequenzrepräsentation einer Sinusschwingung

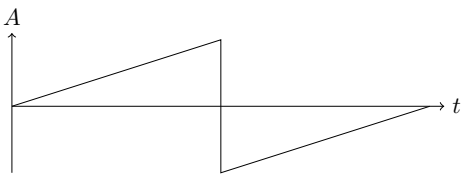


Exemplarische Signalperiode einer Dreiecksschwingung

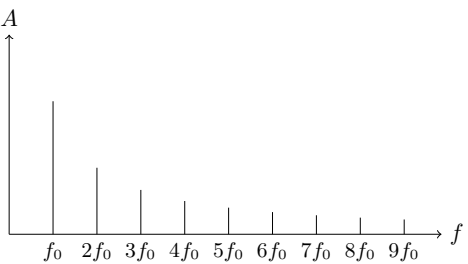


Musterhaftes Spektrum einer Dreiecksschwingung

(b) Zeit- und Frequenzrepräsentation einer Dreiecksschwingung

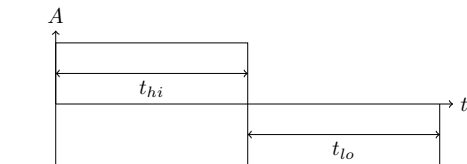


Exempl. Signalperiode einer Sägezahnschwingung

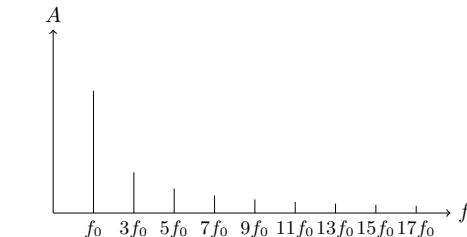


Exempl. Spektrum einer Sägezahnschwingung

(c) Zeit- und Frequenzrepräsentation einer Sägezahnschwingung



Exemplarische Signalperiode einer Rechteckschwingung



Musterhaftes Spektrum einer Rechteckschwingung

(d) Zeit- und Frequenzrepräsentation einer Rechteckschwingung

Abbildung 1: Elementare Wellenformen von Synthesizern

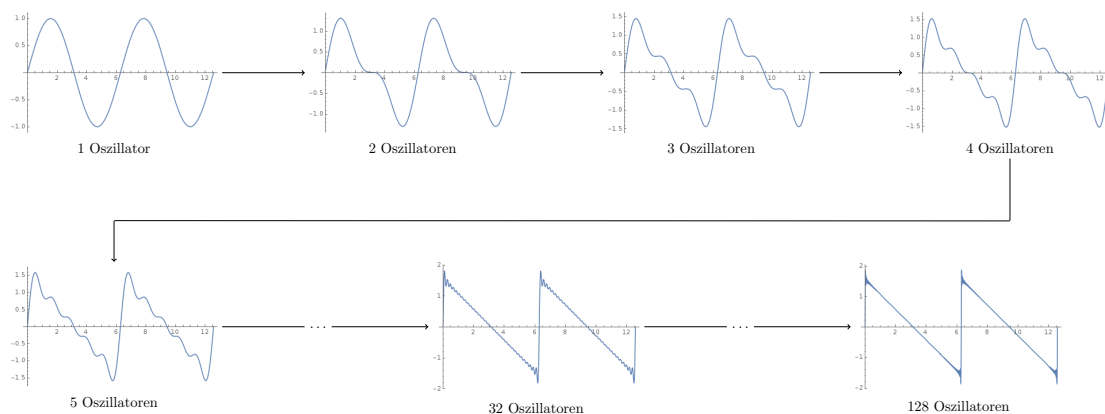


Abbildung 2: Beispiel einer Sägezahnwelle, erzeugt durch additive Synthese mit bis zu 128 Oszillatoren

Eine *Wellenform* ist für Softwaresynthesizer eine geordnete Menge von Samplingpunkten eines beliebigen Signals, meist mit Ähnlichkeiten zu oben genannten grundlegenden Wellenformen. Dieses Signal kann vorher sowohl analogen, wie auch digitalen Ursprungs gewesen sein.

Ein *Wavetable* ist dann eine Menge dieser Wellenformen. Interpolation zwischen diesen schafft dann neue, meist komplexere Wellenformen. Ein Wavetable mit Wellenformen fixer Länge kann in einer unkomprimierten Audiodatei gespeichert werden, um hohe Qualität im Prozess der Weiterverwendung zu erreichen.

Die dieser Syntheseart namensgebende Subtraktion ergibt sich aus der Verwendung von Filtern. Ein Filter ist in der Lage, Frequenzen in einem gegebenen Intervall zu isolieren, akzentuieren oder dämpfen. Die Funktionsweise ist im Abschnitt 4 genauer erläutert.

Subtraktive Synthese erhält ihren Namen also genau daher, dass Filter eben gewissermaßen Frequenzen aus einem Spektrum subtrahieren. Die subtraktive Synthese verwendet in Serie geschaltete (Wavetable-) Oszillatoren, bzw. abgespielte (und wiederholte) Samples und einen oder mehrere Filter. Durch Änderung der Parameter – selbstverständlich besteht diese Möglichkeit auch bei laufender Oszillation bzw. Synthese – kann man den erzeugten Klang sehr fein steuern und somit das gewünschte Klangbild erzeugen. Hierbei handelt es sich in der Praxis hauptsächlich um die Filterparameter und Frequenz des Eingangssignals, aber auch Phasenverschiebung und Amplitude können verändert werden [2].

## 2.3 Modulationssynthese

Unter Modulation versteht man die Klangerzeugung durch Modulation eines oder mehrerer beliebiger Parameter eines Oszillators durch ein sich veränderndes Signal. Werden die Kenngrößen eines Oszillators durch solch ein Signal verändert, so sprechen wir von Modulationssynthese, da die Modulation direkt die Klangerzeugung beeinflusst. Wir erläutern zwei populäre Modulationssynthesearten. Beide Arten finden auch außerhalb der Klangsynthese Anwendung, da sich die Modulationen

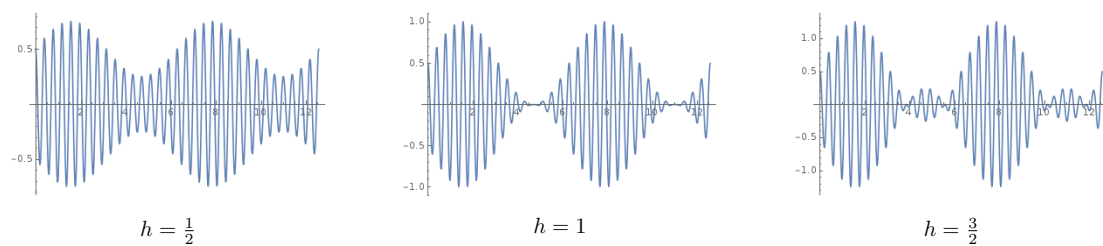


Abbildung 3: Exemplarische Amplitudenmodulation mit ver. Modulationsindizes  $h$

ideal zum Einbetten von Signalen in Basis- oder Trägersignale eignet, um so zum Beispiel von hochfrequenten Transmissionen für die Datenübertragung über lange Strecken zu profitieren. Für uns ist dies aber nur für Signale im für das menschliche Ohr hörbare Spektrum interessant, wohingegen oben genannte Vorteile von außerhalb dieses Spektrums liegenden Frequenzen ausgehen [9, p.215].

### 2.3.1 Amplitudenmodulation

Bei der Amplitudenmodulation wird die zuvor vorgestellte Kenngröße der Amplitude eines beliebigen Trägersignals durch ein Modulationssignal moduliert. Mathematisch lässt sich dies als Produkt zweier Funktionen auffassen, sei  $c(t)$  das Trägersignal über die Zeit und  $m(t)$  das Modulationssignal, dann erhalten wir das amplitudenmodulierte Ausgangssignal  $f(t) = h \cdot m(t) \cdot c(t)$ . Durch die Wahl eines Modulationssignals unterhalb der auditiven Reichweite des menschlichen Ohrs entsteht durch die Amplitudenmodulation ein Effekt, der in der Musik *Tremolo* genannt wird. Oberhalb dieser Schwellgrenze werden dem ursprünglichen Trägersignal neue Nebenfrequenzbänder beigemischt, deren Interferenzerscheinungen maßgebliche Auswirkungen auf den resultierenden Klang haben. Beispiele für Amplitudenmodulation mit verschiedenen Modulationsstärken  $h$  sind in Abbildung 3 dargestellt.

### 2.3.2 Frequenzmodulation

Für die Frequenzmodulation wird nun die Frequenz des Trägersignals moduliert. Wir betrachten also den Träger  $c(t)$  mit Frequenz  $f$  in Abhängigkeit der Zeit  $t$ . Setzen wir  $f := m(t)$  mit  $m(t)$  dem Modulator, erhalten wir ein frequenzmoduliertes Signal. Mathematisch entspricht dies der Verkettung von zwei Funktionen. Wie zuvor bei der Amplitudenmodulation erhalten wir auch bei der Frequenzmodulation mit einem Modulator unterhalb der hörbaren Frequenzen einen musikalischen Effekt, das *Vibrato*, also eine Art Flimmern der Tonhöhe. Oberhalb dieser Schwellgrenze treten, wie auch zuvor, Seiteneffekte und -beimischungen des Spektrums durch die vom Modulator induzierte Frequenz auf [1]. Gerade in moderner elektronischer Musik findet diese Art von Synthese großräumige Anwendung. Ein Beispiel für ein frequenzmoduliertes ist in Abbildung 4 dargestellt.

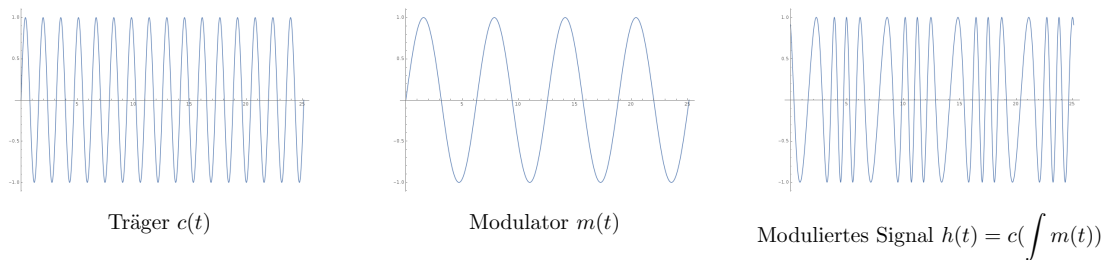


Abbildung 4: Exemplarische Frequenzmodulation zweier Sinusoide

## 3 Klangformung und Modulation

Was Modulation bedeutet, haben wir bereits in Unterabschnitt 2.3 erklärt. Nun wollen wir uns die zwei zentralen Werkzeuge zur Modulation genauer anschauen, die dafür benutzt werden, sich ändernde, realistischere Klänge zu formen.

### 3.1 Hüllkurven

Das Konzept von Hüllkurven, engl. *Envelope*, ist ein fundamentales Konzept der Tonsynthese und Nachahmung von realen Instrumenten, da die Amplitude natürlicher Klänge nicht konstant ist. Die Amplitude der meisten Instrumente folgt daher eine Hüllkurve aus Zeitsegmenten. Dabei wird ein Klang in die Segmente *Attack*, *Decay*, *Sustain* und *Release* eingeteilt, wobei *Sustain*, anders als die anderen drei Werte, die als Werte für Zeit angegeben werden, als Wert für die Amplitude angegeben wird, da *Sustain* die Lautstärke beschreibt, auf der das Signal gehalten wird. Dagegen beschreibt *Attack* die Zeit, die der Ton zum Anschwellen auf seine maximale Lautstärke braucht. *Decay* bezeichnet die Zeit, in der der Ton danach auf das *Sustain*-Level, abfällt. *Release* beschreibt schließlich den Abschnitt, in dem der Ton nach Loslassen<sup>3</sup> der Taste abklingt. Für ein schematisches Beispiel einer Hüllkurve siehe Abbildung 5.

Der beschriebene Fall der Anwendung einer Hüllkurve auf ein Signal wurde bisher nur anhand dessen Amplitude erklärt, da dies ein intuitives Verständnis für die Funktionsweise der Hüllkurve vermittelt. Es ist aber ebensogut möglich, mit einer Hüllkurve einen beliebigen anderen (modulierbaren) Parameter eines Synthesizers zu kontrollieren und modulieren.

Auch ist das oben vorgestellte Konzept, namentlich *ADSR*-Modell für die Initialen der Segmente, um beliebige Abschnitte erweiterbar (oder verkürzbar), um damit mehr Variation in die Modulation einzubringen.

### 3.2 Low Frequency Oscillators (LFOs)

*Low-Frequency-Oscillators* oder *LFOs* sind Oszillatoren, deren Frequenz im Regelbetrieb unterhalb der auditiven Schwellgrenze liegt. Die Amplitude des LFOs wird

<sup>3</sup>Für Instrumente mit Anschlag, für andere Gruppen erfolgt die Definition von Loslassen der Note analog.



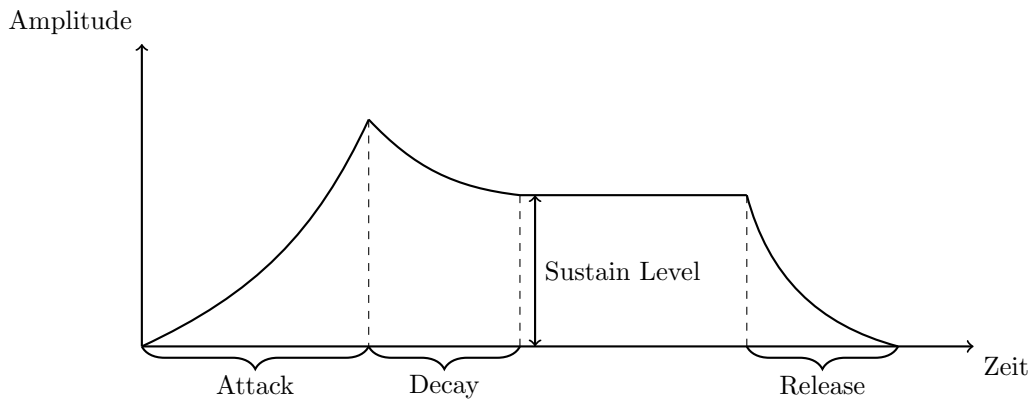


Abbildung 5: Schematische Hüllkurve nach *ADSR-Modell*

als Wert benutzt, um einen beliebigen (modulierbaren) Parameter des Synthesizers mit einem Faktor, der Modulationsstärke oder -tiefe zu modulieren. Typischerweise modulieren LFOs die Amplitude des Ausgangssignals oder die Filterfrequenz eines Filters, jedoch kann mit der Modulation anderer Parameter ebenfalls das Klangbild auf teils exotische Art verändert werden [5].

Offensichtlich ist die Wellenform des LFOs maßgeblich verantwortlich für den resultierenden Klang. Insbesondere ist die Komplexität dieser Wellenform verantwortlich für die Änderungsrate des beeinflussten Parameters, somit wählt man in der Praxis meist *elementare Wellenformen*, da diese dafür sorgen, dass der resultierende Ton, durch Frequenz und Amplitude des LFOs, halbwegs berechenbar für den den Synthesizer bedienenden Benutzer und kontrollierbar bleibt.

Wichtig ist auch, dass der LFO in seiner Schwingungsfrequenz unterhalb der auditiven Schranke bleibt, d.h. in der Regel um die 20–50 Hz, da sonst beispielsweise bei der Modulation der Frequenz durch den LFO Effekte aus der Frequenzmodulation auftreten, d.h. dem Spektrum werden mitunter unerwünschte Nebenfrequenzen beigemischt. Ebenso verhält es sich mit der Modulation der Amplitude durch einen LFO und auch durch die Modulation anderer Parameter können ähnliche ungewollte Seiteneffekte auftreten.

Beide hier vorgestellten Modulationsarten sind bereits Teil der ersten analogen Synthesizer gewesen. Sie stellen damit fundamentale und vor allem alteingesessene Konzepte der Klangformung dar. Für die Erzeugung mancher Klänge mögen diese begrenzten Möglichkeiten nicht ausreichen.

Jedoch hat sich in der jüngsten Zeit in der Entwicklung von vornehmlich Software-Synthesizern durch die neuen technischen Möglichkeiten Einiges in dem Bereich der Modulation und der dafür zur Verfügung stehenden Mittel getan. Insbesondere sind die Grenzen zwischen Hüllkurve und LFO immer mehr verschwommen: Wo LFOs vorher einzig in der Lage waren, ein kontinuierliches Signal zu erzeugen, das periodisch wiederholt wurde, sind in vielen kommerziellen Implementationen auch Hüllkurven in der Lage, repetitiv zu laufen. Auf der anderen Seite haben sich LFOs von der kleinen Menge vorgeschriebener Wellenformen gelöst und sind

immer variabler geworden, bis zu dem Punkt, an dem die Wellenform durch einen anderen Oszillator oder LFO erzeugt werden, oder gar mit Benutzereingaben gemalt werden kann. Letzteres Feature hat auch auf die Erzeugung von Hüllkurven Einfluss gehabt.

Schließlich sind auch exotischere Modulationsarten implementiert worden, die entweder die beiden Grundkonzepte fortführen und erweitern, oder aber komplett neue Konzepte präsentieren. Nennenswert sind hierbei sicherlich das Hinzufügen von Zufall, d.h. Parameter werden auf (pseudo-)zufällige Werte innerhalb eines (wieder modulierbaren Intervalls) gesetzt, ebenso wie Step-Sequencer und Arpeggiatoren. Erstere sind klassische Werkzeuge der Beat-Produktion für Perkussion in Musik, zweite dienen dem rhythmischen Variieren der Tonhöhe in (harmonischen) Abständen.

Die Parametrisierung und Abstrahierung dieser Möglichkeiten, mithilfe von modularen Software-Implementation der Synthesizer, erlaubt es heute, nahezu jeden beliebigen Parameter durch jeden beliebigen Parameter zu modulieren, was das klassische Mittel zur Notation der Modulation, die *Modulationsmatrix*, in der Modulationstiefe, Quelle und Ziel der Modulation sowie die Richtung der Modulation eingebettet sind, an die Grenzen der Visualisierung bringt [3].

## 4 Filter

*Filter* spielen bei der Klangsynthese eine wichtige Rolle. Neben der Tatsache, dass sie bei der subtraktiven Synthese unersetzlich sind, sorgen sie auch dafür, Klänge, die mittels anderer Synthesarten erzeugt wurden, abzurunden bzw. zu optimieren oder helfen bei der Entfernung von Rauschen oder Störgeräuschen.

*Filter* sind dazu in der Lage, Frequenzen in einem gegebenen Intervall zu Isolieren, Akzentuieren oder Dämpfen. Welche Bereiche wie stark verändert werden, hängt vom Filtertyp ab. So dämpft beispielsweise eine Tiefpass-Filter (engl.: Lowpass-Filter) alle Frequenzen oberhalb einer Frequenz, die die obere Schranke des sogenannten “pass bands” darstellt, immer stärker, bis zu einem gewissen Punkt, an dem keine weiteren Frequenzen mehr durchgelassen werden (s. Abbildung 6).

Die Frequenzen im “pass band” werden durch den Filter (weitestgehend, siehe *Resonanz*) nicht beeinflusst. Den Bereich, in dem die Frequenzen zunehmend gedämpft werden, nennt man “transition band” und das Intervall, in dem keine Frequenzen mehr durchgelassen werden nennt man “stop band”. Derartige Bereiche gibt es für jedes Filtermodell und werden bei der Analyse des Frequenzgangs meist angegeben, um ein Modell zu kategorisieren. Um Filter in ihrer Funktionsweise und ihren Eigenschaften besser zu verstehen, ist es wichtig die Kenngrößen zu verstehen:

1. Die (*Grenz-*)*Frequenz* (engl.: cutoff frequency) ist die Frequenz, ab welcher die Dämpfung der Frequenzen beginnt. In welche Richtung der Frequenzdomäne gedämpft wird, unterscheidet sich je nach Filtermodell.

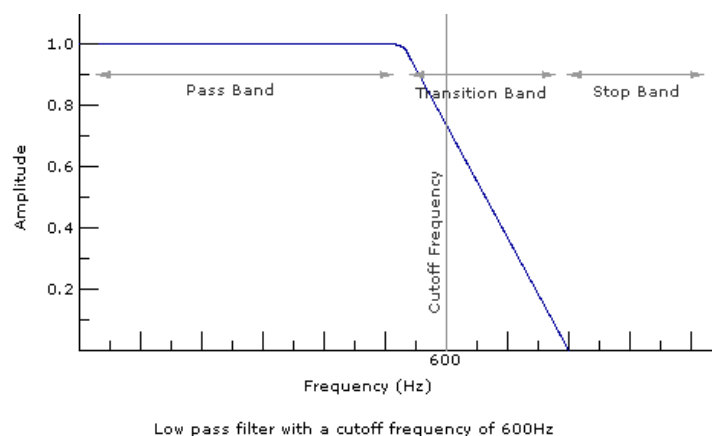


Abbildung 6: Beispiel eines Tiefpassfilters mit Grenzfrequenz 600 Hz und ohne Resonanzverstärkung

2. Bei der *Resonanz* handelt es sich um eine Größe für Verstärkung der Frequenzen dicht um die Grenzfrequenz herum. Ein hoher Resonanzwert sorgt häufig für eine Art Pfeifen im resultierenden Ton.
3. Schließlich gibt es noch den *Gütefaktor* (eng.: quality factor)  $Q$ , der die Schärfe der Resonanz angibt [10]. Teils fälschlicherweise wird der Gütefaktor meist mit Resonanz gleichgesetzt, dadurch bedingt, dass beide Parameter direkt miteinander zusammenhängen und einzeln das Frequenzspektrum ähnlich beeinflussen.

Vor den ersten digitalen Filterimplementationen gab es analoge Filter, welche dementsprechend auch nur auf analogen Signalen operieren konnten. An diesen orientieren sich digitale Filter, somit ist die Terminologie die Selbe. Analoge Filter, die Wechselstromeigenschaften analoger Bauteile ausnutzen, werden mit Spulen, Kondensatoren, Widerständen und teilweise auch Transistoren realisiert, um das analoge Eingangssignal zu filtern.

Digitale Filter können hingegen, da es sich nur um einfache mathematische Funktionen aus Summen und Produkten handelt, deutlich variabler angepasst werden, bieten aber auch die Möglichkeit ganz neuer, speziellerer Filter. Im Vergleich zu Analogen können digitale Filter jedoch nur diskrete Signalwerte bestimmen, was wiederum bei der Wiedergabe zu einem leichten Rauschen führen kann. Bei ausreichender Bittiefe ist dies jedoch nicht mehr hörbar [4].

In Applikationen, etwa Software-Synthesizern, aber auch vielen anderen Anwendungsbereichen, die sich mitunter nicht nur mit Audiosignalen beschäftigen, werden digitale Filter durch das mathematische Modell des Infinite-Impulse-Response-Filter, kurz IIR-Filter, realisiert. Ein IIR-Filter wird durch die Rekursionsformel

$$y_n = \frac{1}{a_0} \cdot \left( \sum_{k=0}^M b_k \cdot x_{n-k} - \sum_{l=1}^N a_l \cdot y_{n-l} \right)$$

berechnet, wobei es sich bei  $x_k$  um die zeitdiskrete Folge der Samplingpunkte des Eingangss- und bei  $y_k$  um die zeitdiskrete Folge der Samples des Ausgangssignals

handelt,  $n$  die Anzahl der Samples bezeichnet,  $b_k$  die Koeffizienten des Eingangssignals mit Filterordnung  $M$  und  $a_l$  die Koeffizienten für das rückgekoppelte Ausgangssignal mit Filterordnung  $N$  sind.

Die Rückkopplung sorgt dafür, dass für einen einzigen Eingangsimpuls ein unendlich langes Ausgangssignal entstehen kann, die sogenannte Impulsantwort (engl. impulse response). Die Berechnung der Koeffizienten hängt einerseits vom gewünschten Filtertyp als auch von Phase und Spektrum des Eingangssignals ab. Schließlich kann man die Filterfunktion und die Anordnung der Rechenschritte der Rekursionsformel für jeden Filtertyp optimieren, was in der Praxis aber nicht gemacht wird, da häufig Implementationen von digitalen Filtern eine große Auswahl verschiedener Filtertypen bieten und man keine universelle Optimierung für alle Filtermodelle findet [4, 8]. Durch die Verwendung von Datenstrukturen mit wahlfreiem Zugriff kann man in der Praxis die Rekursion in eine einzige Iteration über die Folge vereinfachen, da die Rückkopplung einfach durch Verwendung der bereits berechneten Werte realisiert werden kann. Auch kann im Voraus entschieden werden, ob Koeffizienten mit Wert 0 vorhanden sind und Berechnungen, die diese verwenden eingespart werden, was zwar nichts an der asymptotischen Laufzeit ändert, in der Praxis jedoch viele arithmetische Operationen sparen kann.

## 5 Implementation von Monoid<sup>4</sup>

**Monoid** ist der für dieses Seminar in Java entwickelte, auf der Bibliothek **Beads** [6] basierende Software-Synthesizer, anhand dessen wir verschiedene Synthesarten und -techniken implementiert haben, um dabei in der Praxis benutzte Konzepte sowie auftretende Herausforderungen und Probleme aus direkter Sicht darstellen zu können.

Die Signalgenerierung erfolgt durch einen DSP (Digital Sound Processor), der das Signal in Blöcken von Samples generiert, der Buffergröße, die nach der Generierung mit der Samplingfrequenz auf die analoge Ausgabe angelegt wird.

Dabei folgt der Synthesizer dem hier vorgestellten Konzept für die Tonsynthese, d.h. es wird mittels Oszillatoren ein Signal erzeugt, das entweder als Modulationsquelle für andere Oszillatoren genutzt wird oder in Filter mit verschiedenen Filtermodellen geleitet wird. Die Kenngrößen, d.h. Frequenz, Amplitude, Grenzfrequenz und weitere, jedes Moduls können durch die vorgestellten Werkzeuge moduliert werden.

Die modulare Implementation der Signalkette ist gerade das, was **Monoid** aus Entwicklungsperspektive interessant macht, da programmatisch durch die von der Bibliothek bereitgestellte Pipeline für Module jedes beliebige Modul durch eigens entwickelte Schnittstellen als neue Signalkette zusammengekettet werden können.

Wir wollen an dieser Stelle jedoch mehr auf die Besonderheiten und Schwierigkeiten als auf die Entwicklung selbst eingehen.

---

<sup>4</sup><https://github.com/occlonium/Monoid/>

## 5.1 Probleme bei der Implementation

Bisher ist *Monoid* ein monophoner Synthesizer, was bedeutet, dass wir nur eine klangerzeugende Instanz gleichzeitig verwenden können. Dies war bisher auch notwendig, da die Berechnung des Ausgangssignals noch zu zeitintensiv für mehrere Stimmen ist, weshalb wir Polyphonie, also die Mehrstimmigkeit, nur in eigens dafür optimierten Modulen, z.B. der Sektion “additiven Syntese”, verwenden konnten.

Die Berechnung der Signalkette bedarf also noch einiger Optimierung und Anpassung. Für eine sinnvolle Verwendung der Oszillatoren mit nachgeschalteten Modulen, also Filtern, Effekten usw., ist ein Routing-Konzept zwingend nötig. Dies würde es erlauben grafisch besser darzustellen, wie sich mehrere Parameter beeinflussen, deren Quellen mitunter aus komplett verschiedenen Modulen stammen. Ein derartiges Konzept befindet sich derzeit im Entwurf, dessen Spezifizierung allerdings Probleme bereitet, da ein derart nicht-linearer Signalfluss an die Grenzen der momentanen grafischen Darstellbarkeit geht. Die Implementation eines solchen Konzeptes setzt allerdings schon eine effiziente und präzise, zuverlässige Berechnung eines jeden Signals voraus, welche zu diesem Zeitpunkt der Implementation noch nicht gegeben ist.

Auch die manuelle Kontrolle des Synthesizers ist anspruchsvoll. So sollten Input-Events mit möglichst geringer Latenz umgesetzt werden. Eine direkte Umsetzung erfordert aber hochfrequentes Abfragen solcher Events, was wiederum die Berechnung der Signalkette verlangsamt. Neben direkten Input-Events war auch die eigene Implementierung der MIDI-Schnittstelle für die Benutzung des Synthesizers mit einer Tastatur als Controller mit Schwierigkeiten verbunden. Eine eigene Implementation würde die benötigte Geschwindigkeit und Umsetzung von MIDI-Signalen ermöglichen, jedoch wurde die Implementation selbiger, wie auch das Verwenden und Anpassen von internen Möglichkeiten respektive Bibliotheken vorerst zurückgestellt.

## 5.2 Einschätzung und Ausblick

Die verwendete Bibliothek *Beads* [6] bietet Modularität und *Monoid* durch den direkten Aufbau auf Selbiger ebenfalls. Somit kann der Synthesizer nach Paradigmen der objektorientierten Programmierung leicht um neue Module erweitert werden, ohne die bisher implementierten Funktionalitäten zu beeinflussen.

So können zum Beispiel, wie in etlichen professionellen Synthesizer vorhanden, ein Effekt- oder Mixer-Modul hinzugefügt werden. Außerdem ist auch Potential für die bereits beschriebene Polyphonie und eine standardisierte MIDI-Schnittstelle vorhanden.

Weiterhin gilt es, das bisher nur programmatisch mögliche Patching von Komponenten grafisch für die Komponenten zugänglich zu machen, um dadurch die Möglichkeiten für den Benutzer zu erweitern.

## Literatur

- [1] John M. Chowning. The synthesis of complex audio spectra by means of frequency modulation. *Journal of the audio engineering society*, 21(7):526–534, 1973.
- [2] David Crombie. *The Complete Synthesizer: A Comprehensive Guide*. 1984.
- [3] Steve Duda. *xFer Serum - User Manual*, 2014.
- [4] Karl-Dirk Kammeyer and Kristian Kroschel. *Digitale Signalverarbeitung*. Teubner B.G. GmbH, 2006.
- [5] Jari Kleimola. Design and implementation of a software sound synthesizer. Master's thesis, 2005.
- [6] Evan X Merz. *Sonifying Processing: The Beads Tutorial*. 2011.
- [7] Rudolf Müller. *Rauschen (Halbleiter-Elektronik) (German Edition)*. Springer, 1979.
- [8] John G. Proakis and Dimitris K. Manolakis. *Digital Signal Processing: Pearson New International Edition*. Pearson Education Limited, 2013.
- [9] Curtis Roads. *The Computer Music Tutorial*. The MIT Press, Cambridge, Mass. u.a., 1996. ISBN in Vorlage falsch.
- [10] Lutz von Wangenheim. *Aktive Filter und Oszillatoren*. Springer-Verlag GmbH, 2007.