

# Themenvorschlag für Masterarbeit

In vielen spektralen Modellen wird ein Klang als Summe von einzelnen Sinustönen modelliert, die jeweils eine ganz bestimmte Frequenz besitzen. Das daraus resultierende Spektrum ist dann ein Linienspektrum – d.h. eine Superposition von Delta-Distributionen. Viele natürliche Klänge jedoch sind nicht exakt periodisch sondern nur noch quasi-periodisch und besitzen ein Spektrum, bei dem die einzelnen Teiltöne eine gewisse Bandbreite besitzen. Eine Möglichkeit solche Klänge zu modellieren, wäre, wenn man im Frequenzbereich nicht mehr Linienspektren annimmt, sondern jeden Teilton als eine bestimmte Verteilung modelliert (z.B. als Gauss-Verteilung). Man könnte das (Amplituden-)Spektrum also beispielsweise in Form eines Gaussian-Mixture Modells darstellen:

$$A(f) = \sum_{n=1}^N w_n \cdot \exp\left(-\frac{(f - f_n)^2}{\sigma_n^2}\right)$$

Die Teiltonfrequenz entspricht dann dem Erwartungswert, die Teiltonbandbreite der Varianz und die Teiltonamplitude dem Gewichtungsfaktor. Linienspektren sind in diesem Modell als Spezialfall für  $\sigma \rightarrow 0$  enthalten – in diesem Sinne kann es als Verallgemeinerung angesehen werden. Dieser Gedanke wurde bereits in dem Freeware Software Synthesizer „ZynAddSubFX“ implementiert, jedoch ist dieser Synthesizer recht umständlich in der Bedienung und es existiert keine zufriedenstellende Implementation für die weit verbreitete VST-Schnittstelle. In diesem Modell wird nur das Amplitudenspektrum modelliert und es gibt auch keine Zeit-Frequenz-Zerlegung – es handelt sich also um eine reine Frequenzbereichsdarstellung ohne Phaseninformation. Daher ist dieses Modell nicht in der Lage, irgendwelche zeitlichen Strukturen im Signal zu erfassen - es eignet sich daher in erster Linie für relativ statische Flächenklänge. Deshalb bezeichnet Paul Nasca, der Autor des „ZynAddSubFX“, dies auch als PadSynth – Algorithmus. In diesem Algorithmus wird ein sehr großes diskretes Amplituden-Spektrum gemäß obiger Formel errechnet und die Phasen der einzelnen Frequenzen mit Zufallswerten belegt (sehr groß soll heißen: mit sehr vielen bins, in der Größenordnung  $2^{18} - 2^{20}$ ). Dieses wird dann per iFFT in den Zeitbereich transformiert – das Ergebnis ist ein Zeitsignal, das sich perfekt loopen lässt. Im Synthese-Algorithmus werden die Werte für die Mittenfrequenzen, Gewichtungsfaktoren und Bandbreiten durch mathematische Formeln beschrieben, z.B.  $f_n = n \cdot f_1$  für die Mittenfrequenzen,  $w_n = 1/n$  für die Gewichtungen und  $\sigma_n = \text{const.}$  für ein einfaches sägezahn-artiges Spektrum. Ich habe selbst mit diesem Algorithmus in MatLab herumexperimentiert und auch schon angefangen, ein VST-Instrument auf Basis dieses Algorithmus zu programmieren und muss sagen, dass dieser Algorithmus wirklich herrliche (Flächen-)Klänge erzeugen kann – mit anschließender Filter- und Amplitudenhüllkurve sind auch andere Klänge möglich. Ich könnte mir sehr gut vorstellen, dieses VST Instrument im Rahmen meiner Masterarbeit zu einem vorzeigbaren, benutzerfreundlichen Instrument weiterzuentwickeln. Da eine Masterarbeit aber auch möglichst neue Gedanken enthalten sollte und die Idee zu diesem Algorithmus leider nicht von mir stammt, habe ich mir gedacht, das Konzept um einen Parameter-Schätz-Algorithmus zu erweitern, der Mittenfrequenzen, Gewichtungsfaktoren und Bandbreiten für obiges Mixture-Model aus einem Audiofile extrahiert. Damit würde der PadSynth-Algorithmus von einem reinen Synthese-Algorithmus zu einer Art Analyse-Resynthese-Algorithmus erweitert. Ich habe auch schon einen Algorithmus in MatLab implementiert der auf k-means Clustering basiert und recht gut funktioniert. Ein kleines C-Programm könnte die von MatLab geschätzten Parameter dann von einer .mat - Datei in eine .fxp – Datei umwandeln, die dann als Preset von dem VSTi geladen werden kann. Rein algorithmische Klangsynthese ohne reales Vorbild wäre bei meinem VSTi natürlich weiterhin möglich.

Weitere Infos zum ZynAddSubFX:

<http://zynaddsubfx.sourceforge.net/doc/PADsynth/PADsynth.htm>