



EPRO Uebung

Bachelor Studiengang Medientechnik
Fachhochschule St.Pölten

Wien, November 30, 2015

Introduction/Notes

Umstrukturieren! Pd muss auch vermittelt werden, nicht direkt audio/DSP. Im nächsten Semester aufgaben von hier mit reinnehmen! <http://www.soundonsound.com/sos/allsynthsecrets.htm> AM/FM fehlt? vorbereiten:

- FFT händisch rechnen.
- umstrukturierung: sampling an den anfang. Audio überblicks-übung.

Form

Inverted Classroom

Nicht erledigen d. Hausübung bzw. nicht anschauen d. videos macht das konzept kaputt. Macht unsere stunden zum teil sinnlos. Konzept soll vorteile für studenten und vortragende bringen!

Sinnfrage d. Inhalts/unterrichts klären

Unterscheidung: Tool lernen, technik/theorie lernen

Contents

Introduction/Notes	II
Form	II
Inverted Classroom	II
Sinnfrage d. Inhalts/unterrichts klären	II
Unterscheidung: Tool lernen, technik/theorie lernen	II

I. Semester 3

1. Sampling, Waveshaping, und nicht Linearität	2
1.1. Notizen	2
1.2. Uebersicht	2
1.3. Waveshaping	3
1.3.1. Einfachster fall, linear transfer function.	3
1.3.2. Einfache nicht linearität: X^2	4
1.4. Sampler	5
1.5. Hausübung	9
1.5.1. Testmodul	9
1.5.2. Distortion	9
2. Filter	10
2.1. Notizen	11
2.2. einfuehrung	12
2.2.1. Raetsel	13
2.3. kammfilter	17
2.4. Moving Average	20
2.5. convolution, Faltung	21
2.6. IIR/FIR	21
2.6.1. Onepole	22
2.7. Hausübung	22
3. Polyphonie/Message domain	23
3.1. Notizen	23
3.2. Einführung	23
3.3. HUE kontrolle	23

3.4. Filter wiederholen	24
3.5. Signale betrachten	24
3.6. Synth auffetten	24
3.6.1. Polyphonie	25
3.7. Unison	25
3.8. Hall	25
3.9. Sequencer	25
3.10. drumpad/envelope following	25
4. Karplus	27
4.1. Notizen	27
4.2. Vorbereitung	27
4.3. Uebersicht	27
4.4. Theorie, Schwingungsfähige systeme	27
4.5. Parktical Karplus-Strong	28
4.6. Schulübung, Sounddesign challenge	28
5. FFT	30
5.1. Notizen	30
5.2. Fourier Transformation	30
 II. Semester 4	
1. Jitter	35
1.1. Notizen	35
1.2. Überblick	35
1.3. Andere Software Pakete für visuelle Echtzeit Arbeit	35
1.4. Introduction	36

Part I.

Semester 3

1. Sampling, Waveshaping, und nicht Linearität

1.1. Notizen

Waveshaping generell.

Evt. Subtraktiver Synth am ende.

Zusammenhang waveshaping, distortion. lookup table, sampling, wavescanning.

Waveshaping = modulation (Farnell, 2010, p. 257)

chebyshev

„intermod. distortion“ , vgl. miller puckette,

<http://msp.ucsd.edu/techniques/latest/book-html/node78.html>

middle term! ($(a+b)^2 = a^2 + 2ab + b^2$)

beispiele: sinus vllt:

dyn. non-linear functions

1.2. Uebersicht

Zunächst: Sampling, dann lineare transfer function.

1. übersicht: fragen wies geht, wie ist es mit Max/MSP?
2. HÜ ankündigen
3. neue abstraction: z-1
4. Raetsel
5. Wo letztes mal stehngeblieben? Poly syth fertig?
6. linearität erklären. Nun Nichtlinearität.
7. wavetable/lookup wiederholung
8. Lookup als waveshaping/distortion
9. waveshaping durch processing vs. durch lookuptable
10. durchrechnen

11. chebyshev
12. Sampling (Ram nicht Ram)
13. send and receive von GUI f. Hausübung

1.3. Waveshaping

Wikipedia zitat, seite wavesahper:

„The mathematics of non-linear operations on audio signals is difficult, and not well understood.“

1.3.1. Einfachster fall, linear transfer function.

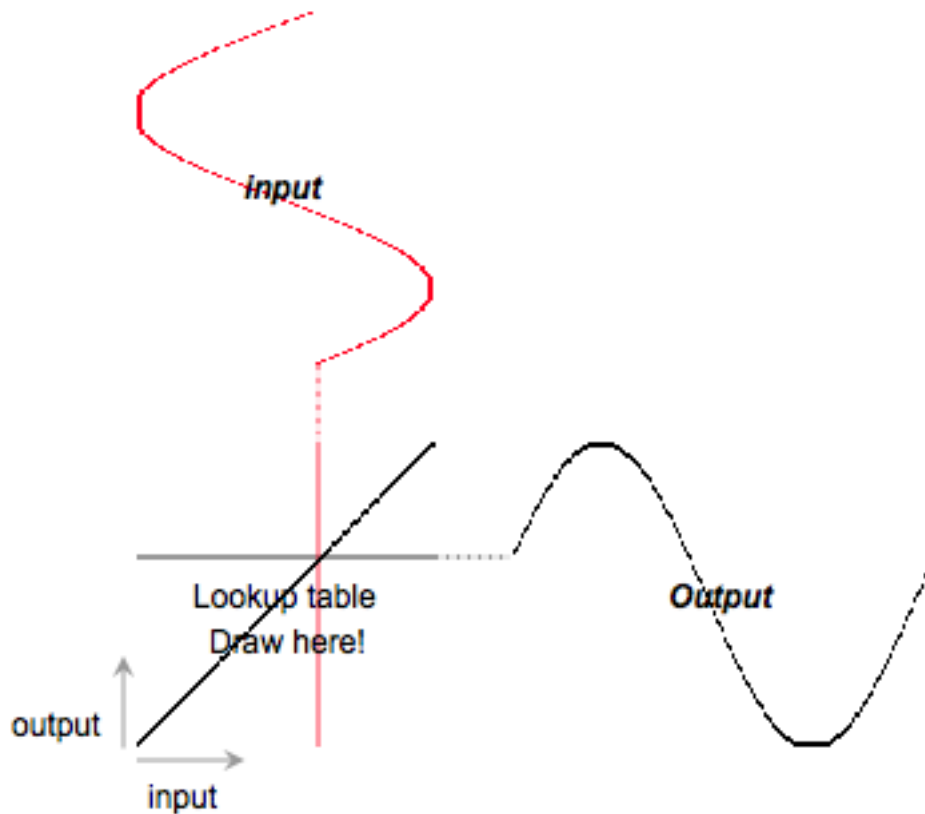


Figure 1.1.: Linear Transfer function

1.3.2. Einfache nicht linearität: X^2

Eine einfacher Fall: ein polynom 2. ordnung soll nun analysiert werden. Die funktion

$$f(x) = x^2 \quad (1.1)$$

wird benutzt.

Ein polynom geradzahligter ordnung n als Transferfunktion produziert immer alle geradzahligten obertöne von n bis 0, exklusive der grundfrequenz (da ja auch ungerade, 1 ist eine ungerade zahl).

Ein polynom ungeradzahligter ordnung n als Transferfunktion produziert immer alle ungeradzahligten obertöne von n bis 1, also der grundfrequenz.

Wir errechnen was mit einem cosinus signal passiert wenn es durch diese funktion geschickt wird. Also:

$$x = \cos(\omega) \quad (1.2)$$

mit beliebigen omega. (die laufzeit-/index-variable t bzw. n wird hier ausgepart, sie ist hier irrelevant).

Demnach ist

$$y = \cos(\omega)^2 \quad (1.3)$$

Das wiederum ergibt

$$y = \cos(\omega) \cdot \cos(\omega) \quad (1.4)$$

eine simple amplituden modulation eigentlich *Ringmodulation*. Über Ringmodulation wiederum wissen wir dass summe und differenz der eingangsfrequenzen ausgegeben werden. Also:

$$y = \frac{\cos(\omega + \omega) + \cos(\omega - \omega)}{2} \quad (1.5)$$

Also:

$$y = \frac{\cos(2 \cdot \omega) + \cos(0)}{2} = \frac{\cos(2 \cdot \omega)}{2} + \frac{1}{2} \quad (1.6)$$

Aber gilt die auch für jeden Input? das würde Bedeuten wir haben einen freq. shifter gebaut? Nein. Waveshaping ist weit komplexer, was sofort ersichtlich ist wenn man zwei oszillatoren in die funktion schickt:

$$x = \cos(\omega_1) + \cos(\omega_2) \quad (1.7)$$

dann

$$y = (\cos(\omega_1) + \cos(\omega_2))^2 \quad (1.8)$$

$$y = \cos(\omega_1)^2 + \cos(\omega_2)^2 + 2 \cdot \cos(\omega_1) \cdot \cos(\omega_2) \quad (1.9)$$

letztendlich kommt man zu:

$$y = \frac{\cos(2 \cdot \omega_1)}{2} + \frac{1}{2} + \frac{\cos(2 \cdot \omega_2)}{2} + \frac{1}{2} + 2 \cdot \left(\frac{\cos(\omega_1 + \omega_2) + \cos(\omega_1 - \omega_2)}{2} \right) \quad (1.10)$$

Wieso ist waveshaping praktisch? Amplitudenabhängigkeit d. spectrums. Wieso ist waveshaping verwandt mit modulation, sowohl FM, PM als auch AM? Am: siehe x^2 . FM: siehe $f(x) = \cos(x)$. Auch kann letztendlich eine transferfunktion in cos/sin bestandteile zerlegt werden um zu einer menge an frequenzmodulationen anzukommen, bzw kann der wavetable als polynom angenähert(o. Taylor entwicklung) werden um bei AM anzukommen.

1.4. Sampler

Sampling von d. Festplatte vs. vom RAM.

Zusammen bauen. Geschwindigkeit modulieren etc.

Evtl. `moresampling.pd`, granular sampling kurz erklären.



Figure 1.2.: simpleSampler

Loading Audio to an Array (to RAM)

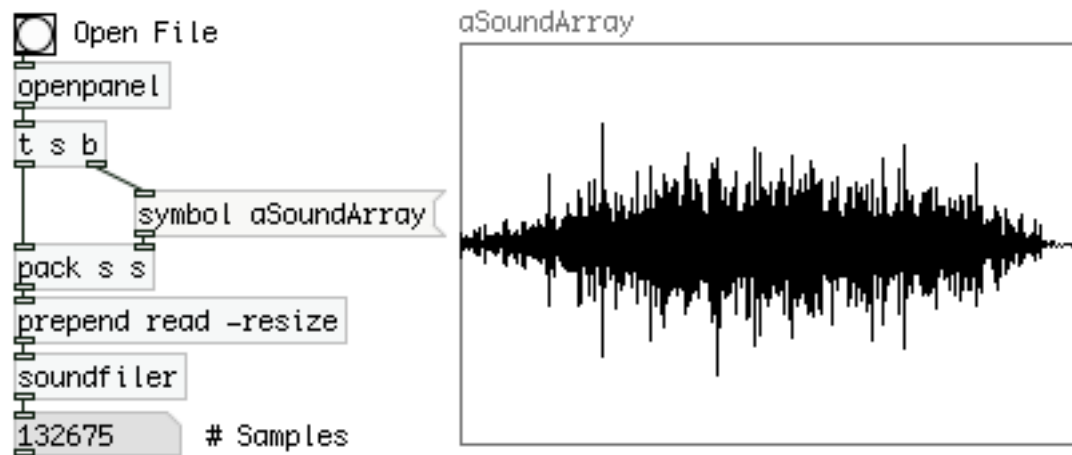


Figure 1.3.: sound in Ram

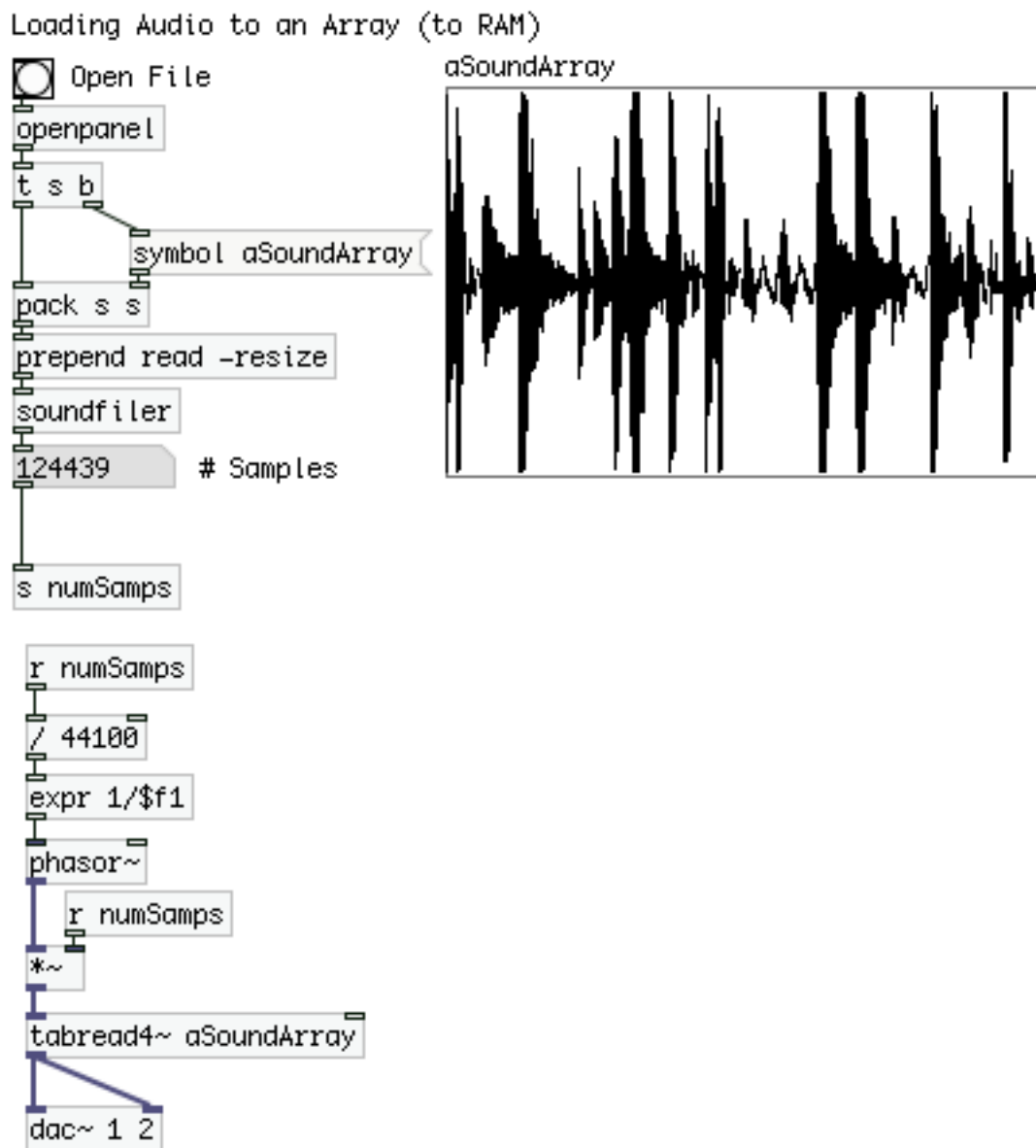


Figure 1.4.: RamFilePlayback

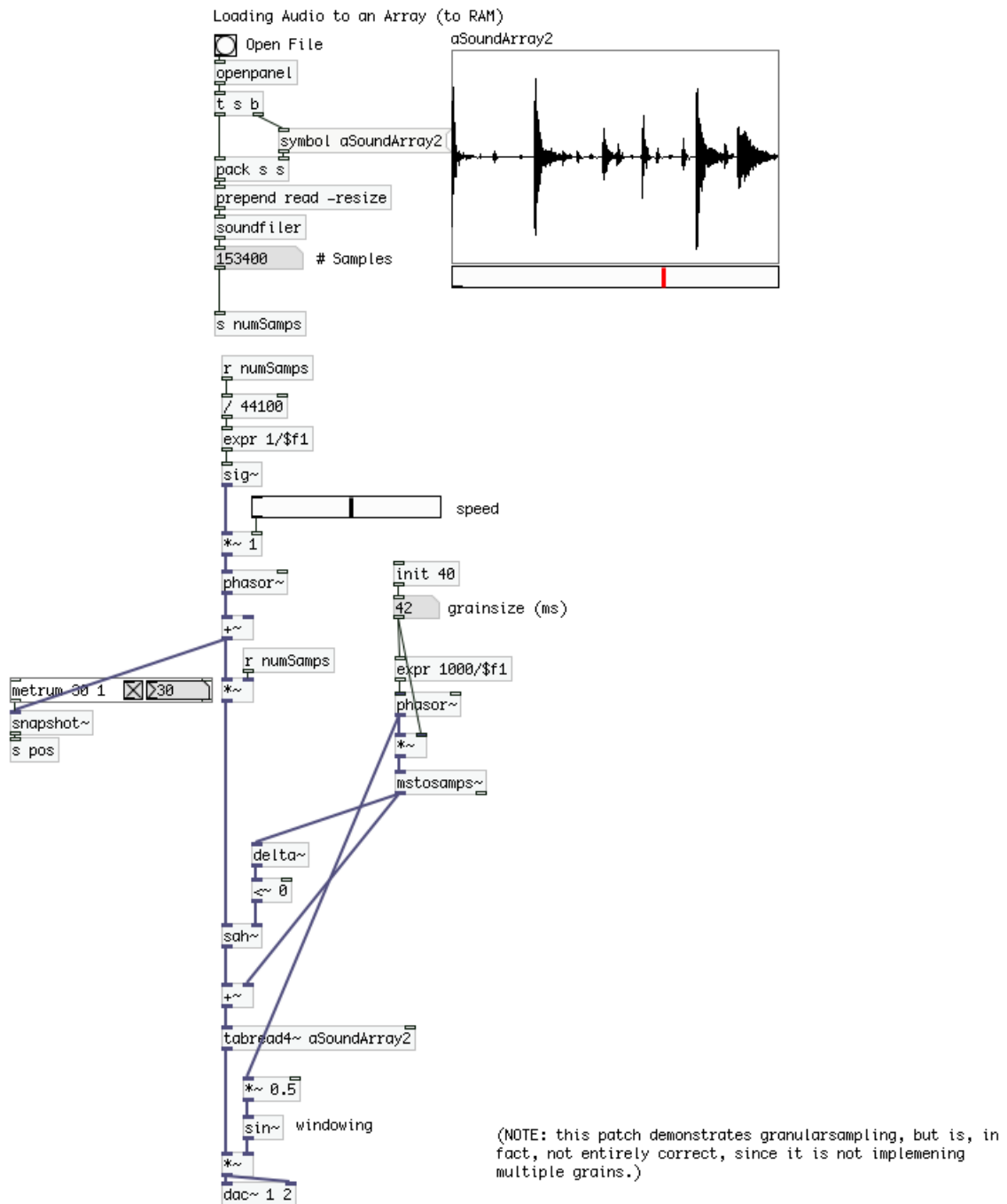


Figure 1.5.: moreSampling.pd, a simplified version of granular sampling

1.5. Hausübung

1.5.1. Testmodul

baue ein audio Testmodul mit folgender spezifikation:

- Ein audio output
- verschiedene klangquellen wählbar:
 1. White Noise
 2. Sinus (freq. einstellbar)
 3. soundfile (file wählbar)
- GUI
- verfügbar(in eurem pfad, und jederzeit abrufbar als abstraction)
- output pegel sichtbar (level meter)

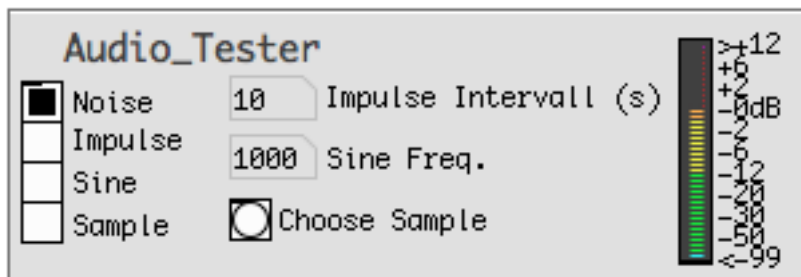


Figure 1.6.: audioTester.pd, zu bauen als Hausübung

1.5.2. Distortion

Baue eine besonders wohlklingende Distortion, inkl. User interface.

2. Filter

2.1. Notizen



Figure 2.1.: Caravaggio, Narziss. ad Feedback: Echo liebt Narziss etc.

Notizen aus erster UE:

Viellt. mit IR anfangen? Vielleicht davor noch: das beispiel mit noise + sinus.

IR, bypass system, hall system.

garnicht faltung. Moving average zu long FIR. Impuls antworten anhand beispielen besprechen.

1. FIR!

kammfilter(FF)

dann FIR kammfilter

chorus flanger

LFO

->index/tiefe, offset, frequenz wdh.

-frequenzgang auslöschungen, rechnen..

2. FIR one sample delay

FIR bauen, vllt. an tafel blockdiagramm.

FIR hipass

3. mehrere Samples delay IR differenzen gleichung

eventuell Faltung(faltungs hall connection herstellen)

faltung als multiplikation d. spektra multiplikation als faltung d spektra.

M4L faltungs hall herzeigen

LTI?

differentiator, akkumulator?

4. IIR

notizen POlyphonie was separat, was einmal

2.2. einfuehrung

Ziel der LV:

1. Gefühl dafür bekommen was die häufigsten Operationen der DSV eigentlich tun.

2. Gefühl dafür bekommen, dass es günstig ist, zwischen verschiedenen repräsentationen hin und her zu wechseln (time und frequency domain, aber auch system repräsentationen)
3. Sich in professioneller literatur(DSP lehrbücher) nicht völlig verloren fühlen.
4. schönheit zu sehen dass alles das „gleiche“ ist.(Delay, filter, reverb, chorus, flanger etc)

Wozu?:

1. Wir arbeiten ständig mit Operationen der DSV. Filter sind nicht nur in synthesesizern. Ein tieferes verständnis erleichtert die arbeit. Wie funktioniert zum beispiel ein blur?
2. Was ist ein FeedForward compressor?

Nachdenken über filter allgemein. Was ist ein filter im gespür d. studenten.
Bekannt was ein LTI system ist?

2.2.1. Raetsel

Woraus könnte das signal in Bild 2.2 bestehen? **Kann jemand das spektrum aufmalen?**

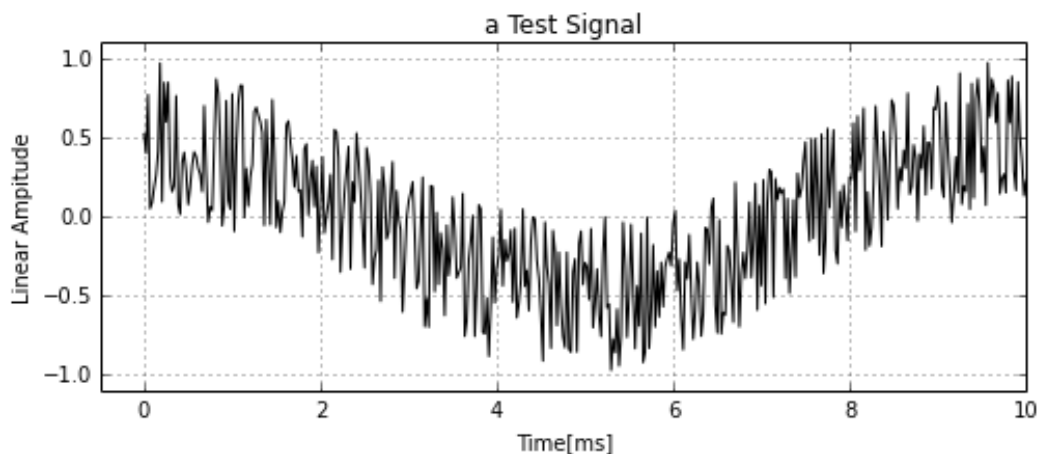


Figure 2.2.: Original Signal

Das Signal in Bild 2.2 wurde gefiltert und das Signal in Bild 2.3 ist entstanden. Was für ein Filter wurde möglicherweise verwendet?

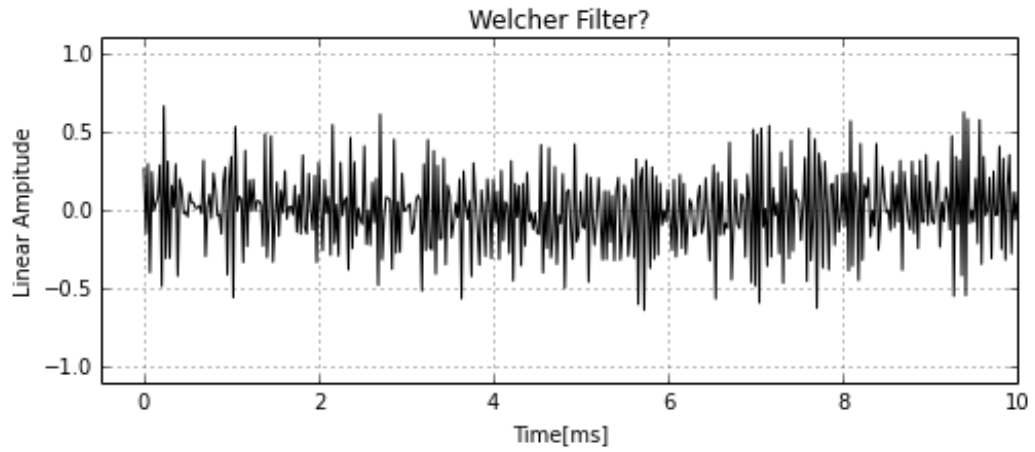


Figure 2.3.: raetsel 1

Das Signal in Bild 2.2 wurde gefiltert und das Signal in Bild 2.4 ist entstanden. Was für ein Filter wurde möglicherweise verwendet?

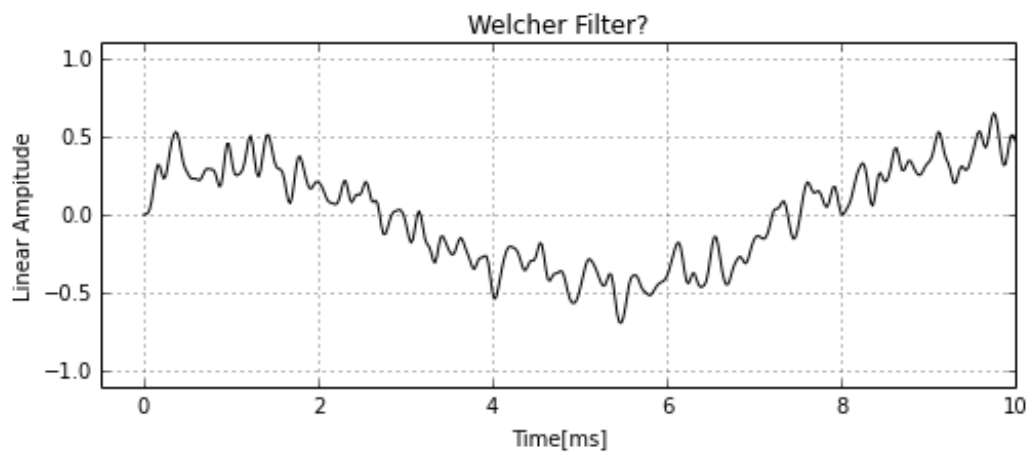


Figure 2.4.: raetsel 2

Lösung

Original signal = White noise + cosinus bei 100 Hz.

Raetsel 1 : ein Highpass.

Raetsel 1 : ein Lowpass.

(many FIR filters in series.)

-Wie könnte ein filter realisiert werden?

-Ein digitales signal ist eine reihe von zahlen, eine sequenz von werten. (über bildliche darstellungen sprechen)

-Echtzeit fall: ich bekomme in ein system ein signal(also ständig werte) und soll ein zb. lowpass gefilteretes signal ausgeben.

Was kann ich mit dem signal machen? Zum beispiel addieren, multiplizieren, verzögern.

Einfacher moving average. Zunächst nur an der tafel! Impulsantwort ausrechnen! Was ist ein impuls.

Was ist eine impulsantwort? Beispiele.. unit impulse (= dirac impuls, dirac funktion, delta function)

Diagramm aufmalen:

impuls \rightarrow Raum \rightarrow Impulsantwort

oderr

impuls \rightarrow beliebiges system \rightarrow Impulsantwort

anwendungsbeispiel: tontechnik bei okto; mischpult ausmessen, dbmax ausmessen(fragwürdige sache! Wieso?)

FRAGE

Ein system(zB. ein Filter) hat folgende Eigenschaft:

Wenn zwei signale getrennt (d.h. unabhängig voneinander, zB. nacheinander.) in das system geschickt werden, und die Ergebnisse summiert werden, entsteht das signal P . Wenn andererseits die signale zuvor summiert werden und dann in das system geschickt werden entsteht ebenfalls P . Welche gleichung entspricht einer beschreibung dieser eigenschaft?

1. $f(x) + f(y) = f(x * y) = P$

2. $f(x) + f(y) = f(x + y) = P$

3. $f(x) + f(x) = f(x + y) = P$

4. $f(x) + f(y) = f(x + x) = P$

5. nichts von alledem.

richtig : nr.2

Erklärung eines LTI systems:

- Linear:
 - Superpositionsprinzip $f(x) + f(y) = f(x + y)$
 - Homogenität $a * f(x) = f(a * x)$
- Time Invariant (zeit-invariant)

Zurück zu filter: was glauben sie wie er funktioniert.

2.3. kammfilter

Wiederum fragen was sich die studenten vorstellen.

„Wir bauen ein delay.“ nicht vergessen: block , subpatcher.

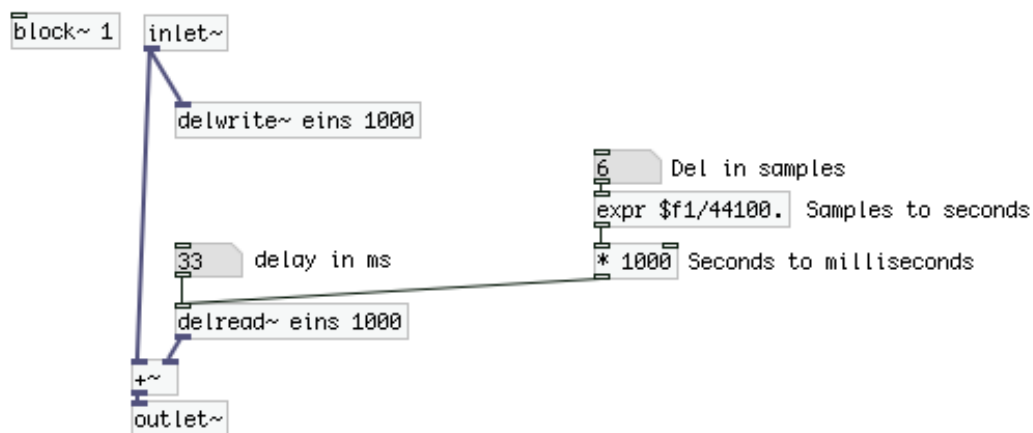


Figure 2.5.: SimpleDelay

Studenten sollen ausprobieren. Verschiedene inputs f. kammfilter.

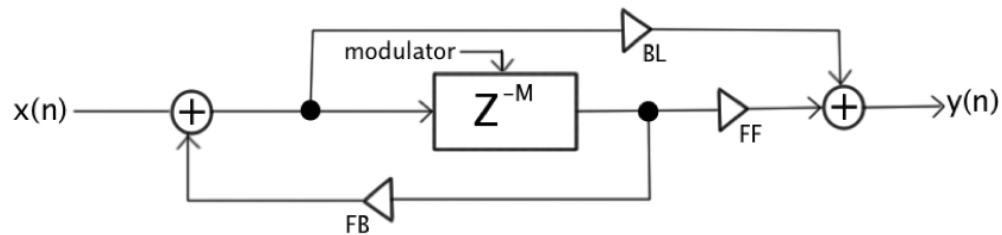
Modulation des kammfilters. Studenten sollen LFO an den kammfilter dran bauen.

→ Flager

→ Chorus

index/tiefe, offset, frequenz wiederholen.

Generalized Effects Structure proposed by Dattoro



modulator is either an LFO or lowpass noise

Effect:	BL:	FF:	FB:	Delay (ms):	Depth (ms):	Modulator:
Vibrato	0	1	0	0	0-3	0.1-5 Hz sine
Flanger	.7	.7	.7	0	0-2	0.1-1 Hz sine
Chorus	.7	1	-.7	1-30	1-30	lowpass noise
Doubling	.7	.7	0	10-100	1-100	lowpass noise

Figure 2.6.: generalized Effect structure

über name sprechen, sowohl spectrum als auch Impulsantwort(IIR variante) sind kammförmig.

Eigentlich einfach ein delay, ein delay mit feedback im falle von IIR.

Zunächst nicht wichtig die versch. representationen zu verstehen.

Patch representation kurz durchbesprechen. Sollte verständlich sein.

Unterschiedliche representationen durchbesprechen. Fragen, abstimmen?, was die studenten für die angenehmste representation halten.

- Patch: vorteile: interaktiv, allgemein, reifolge der events nachvollziehbar. Nachteil: nicht sehr übersichtlich. Nicht sehr konzis, nicht kompakt.
- Block Diagramm: übersichtlich, allgemein. Reifolge der events klar, pfeile! Fluss, richtung, eindeutig. Nachteile: Nicht kompakt.
- differenzen gleichung: Vorteil: kompakt, konzis. Nachteil: reifolge der events nicht klar ersichtlich: kein rezept, eher eine beschreibung. (nicht imperativ sondern deklarativ, va. im Fall von Feedback)
- Impulsantwort: **Frage: was ist ein Impuls?** Vor/Nachteil: Nicht allgemein, nur ein spezieller zustand des systems beschrieben. Nicht kompakt, aber kann

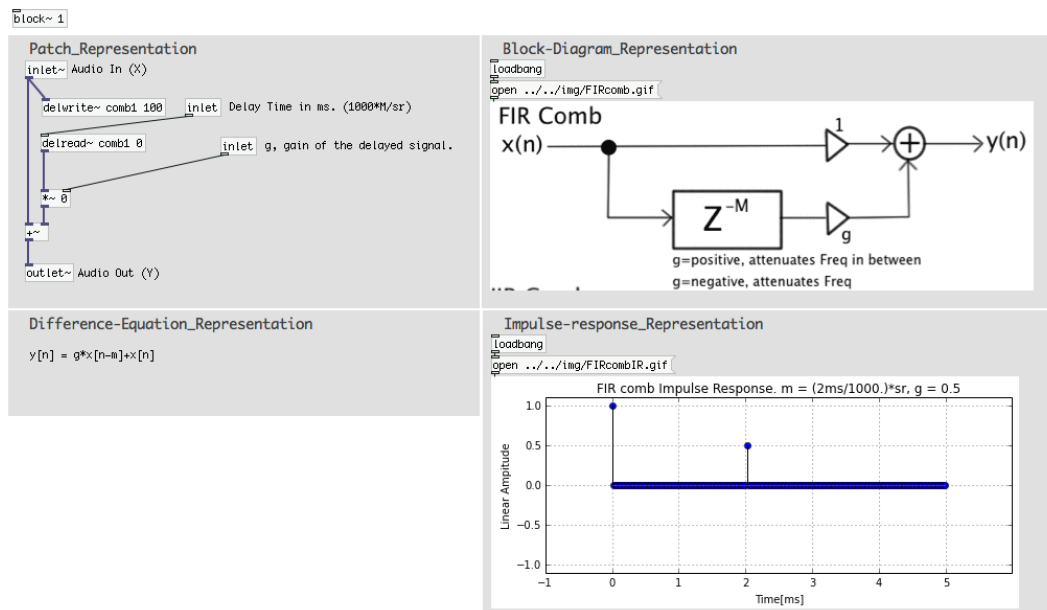


Figure 2.7.: Der Patch 01_combFilter.pd

sehr intuitiv sein. Beispiele an tafel: IR von bypass-system, IR mit viel hall, IR von bandpass filter mit hoher resonanz, eventuell: differentiator, accumulator.

- andere representationen: text-code, transfer funktion, frequency response(magnitude + phase response/group delay)

2.4. Moving Average

Wo hat der kammfilter immer sein erstes Tal im spectrum? Sinus an tafeln. Antwort: bei

$\lambda = (2 * \Delta t)$ (wobei Δt die delay zweit in sekunden. und λ die periodendauer der gesuchten frequenz.) Daher:

$$f_c = 1/(2 * \Delta t)$$

Nun wird das delay auf ein sample reduziert. Ein lowpass entsteht der seine grenzfrequenz bei $sr/2$ hat.

Frage: was macht ein lowpass filter in der timedomain?

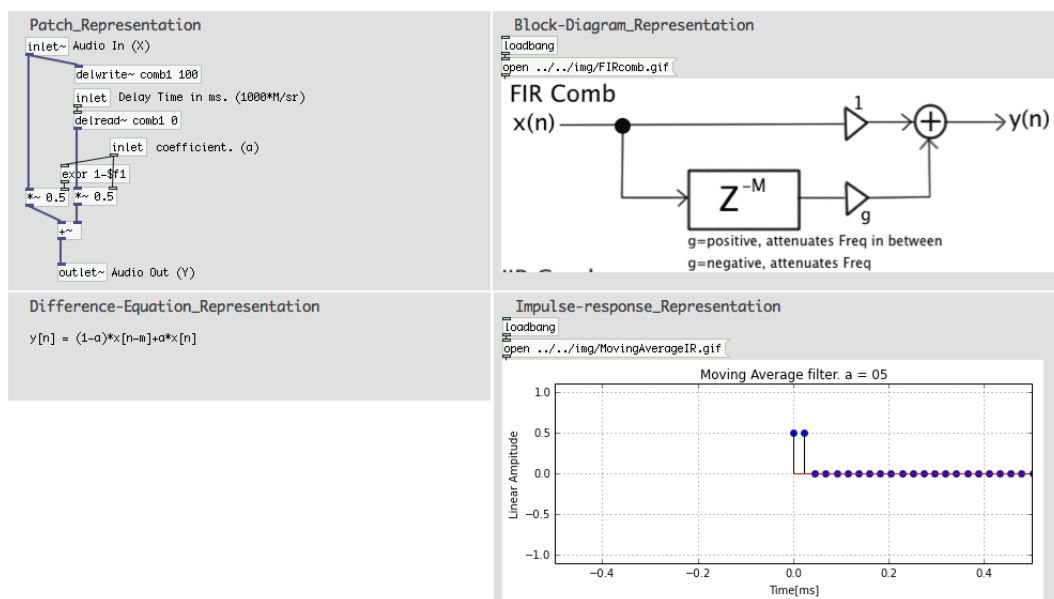


Figure 2.8.: movingAverage

EXKURS: VIDEO FILTER, lowpass, time domain erkennen.

Frage: Vergleiche original und die gefilterte variante (01.mov) . Was für ein filter könnte angewandt worden sein:

1. lowpass
2. bandpass
3. hipass
4. notch
5. nichts von alledem

wie könnte ein highpass realisiert werden? Wie schaut der effekt eines highpass filters in der frequenz- und wie in der timedomain aus?

2.5. convolution, Faltung

Video (unten) zeigen, vorher erklären:

conv theorem.

http://youtu.be/_vyke3vF4Nk?t=25m16s

start bei min 25.

maxpatch

2.6. IIR/FIR

Je steiler die Filterflanke sein, je komplexer der filter sein soll, desto mehr delays werden benötigt.

Wieso: einfache Erklärung: Um aus einem signal, das alle frequenzen enthält (zB dirac impuls) ein signal zu machen, das hauptsächlich sehr tiefe frequenzen enthält wird ein system benötigt das „lange wellen“ zu produzieren im stande ist.

An tafel zeichnen: dirac impuls und unit step/ DC. Frage nach spectrum.

frage nach FIR der diesen IR haben könnte.

Patch: lomgSimpleFir

Grenzfall: integrator macht aus unit impulse, $\delta[n]$, unit step signal $u[n]$ (DC).

Integrator an tafel malen.

Daher: Es kann gezeigt werden dass ein feedback pfad einer unendlichen menge an delays gleichkommt, siehe IR.

Patch: 02_combFilterIIR

nebenbei: linear phase = symmetrischer IR, immer FIR, daher weniger performant. Linear phase filter sind notwendig wenn die timedomain wellenform möglichst unbeeinträchtigt bleiben soll.

2.6.1. Onepole

onepole an tafel beschreiben. Block diagramm, malen, nach differenzengleichung fragen. Patch herzeigen.

patch: IIRtest

danach:

patch: IIRtest2

Fragwürdig aber falls interesse: onepole coeff berechnung zB:

$$a_0 = \frac{2\pi f_c}{sr} \quad (2.1)$$

oder

$$a_0 = \sin\left(\frac{2\pi f_c}{sr}\right) \quad (2.2)$$

2.7. Hausübung

hausübung: Baue einen einen chorus. TESTSIGNAL AUSBESSERN! nicht noise!!

3. Polyphonie/Message domain

3.1. Notizen

Vor der übung buffersize raufstellen!

Bug: Buffersize verstellen: pd funktioniert nicht mehr. Lösung: prefs löschen.

Zu Löschen: `/Library/Preferences/org.puredata.*`

Polyphonie, synthesizer aus vorletzter uebung erweitern. polysynt bauen?

SEQ

Eventuell sampler als aufheiterung. Sampler + edrumkit? Sequencer!

eventuell hall als aufheiterung

Durchgehen: signale beurteilen. (spectrogram , aussteuerungsanzeige, scope)

hue -> signale beurteilen -> abstraction -> message domain -> sequencer.

3.2. Einführung

Ziele d. UE:

1. HUE kontrolle
2. kurz Filter wiederholen.
3. Signale betrachten.
4. Synth auffetten.
5. daher **message domain**. -polyphonie, sequencer. zur motivation als erstes Sequencer herzeigen.
6. Einen pd Patch bauen der dinge ermöglicht die in einer DAW nicht realisierbar sind.

3.3. HUE kontrolle

Chorus war zu bauen. Erklären des blockschaltbilds, erklären des pd patches. Womit wurde der chorus getestet?

3.4. Filter wiederholen

Was ist ein FIR was ein IIR filter? Vor/nachteile? Beziehung zwischen kammfilter und lowpass?

3.5. Signale betrachten

BlackBox rätsel!

siehe pd patch `backBox1.pd` Was für ein signal haben wir? Welche mittel stehen zur verfügung? um das signal zu vermessen?

- snapshot
- print
- aussteuerungsanzeige
- peakamp
- sprectrogram
- scope
- tabwrite

siehe pd patch `signalMetering.pd` **Bedeutung von filtern bei der betrachtung von signalen!**

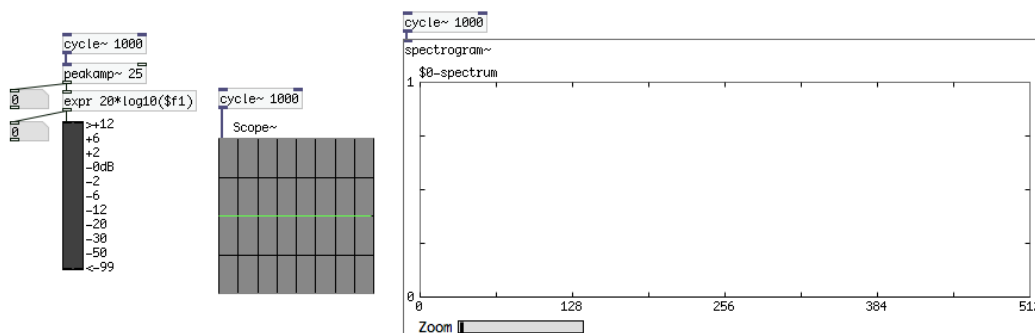


Figure 3.1.: metering techniques

3.6. Synth auffetten

Was fällt den studenten ein was man machen könnte um den synth fetter zu machen?
zB: Polyphonie, unison, Chorus, hall.

3.6.1. Polyphonie

Simplen poly synth from scratch.

3.7. Unison

Supersaw

3.8. Hall

gemeinsam Schroeder paper anschauen. Nachbauen.

3.9. Sequencer

Sequencer + drumpad = wir sparen uns zeit und geld.

3.10. drumpad/envelope following

Vor der übung buffersize raufstellen!

HUE: eventuell sequence bauen.

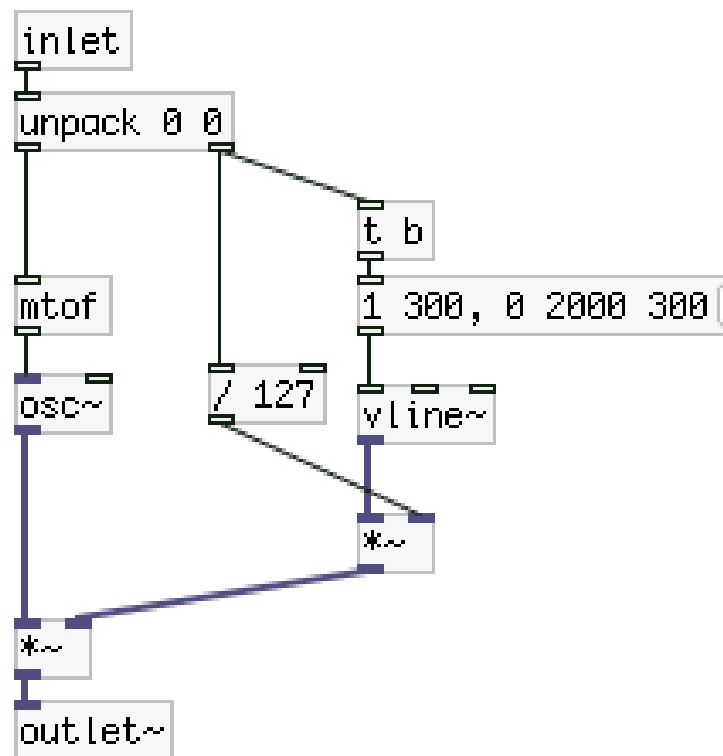


Figure 3.2.: Simple Synth Voice, SynthVoice.pd

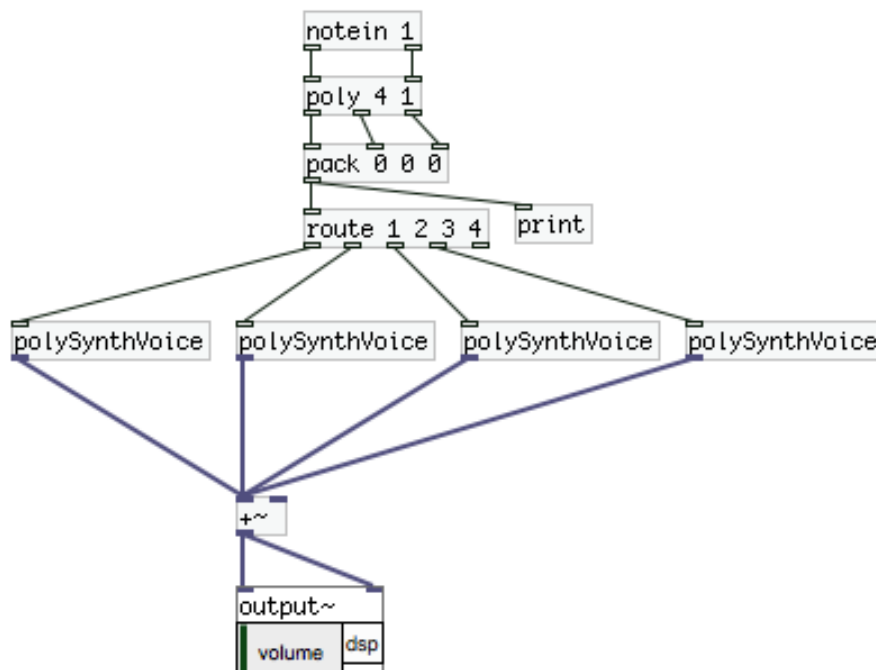


Figure 3.3.: Example Poly Host, patch 01_polySynthHost.pd

4. Karplus

4.1. Notizen

Karplus strong und PM,
schwingungssysteme,
simulation..

Ableton Live: donner,
rechtfertigung: Computerspiele, etc.
(Wellen-)Experiment.

Geschichte: Bernoulli und D'Alembert

Wellengleichung Karplus Strong (bauen, geschichte erzählen) (Pause) Waveguide, Scattering Junctions erwähnen und je nach interesse behandeln. Kelly und Lochbaum modell. Kaffeehägerl synthetisieren. (Praat, reson)

4.2. Vorbereitung

Konzeptuelle Gedanken(imaginary number, complex numbers):

https://www.youtube.com/watch?v=-IJuqR6nz_Q

<https://www.youtube.com/watch?v=EIstpPXKWng> wellengleichung:

<https://www.youtube.com/watch?v=Li6OEMCtQ9k>

4.3. Uebersicht

1. motivation: gewitter im ableton live
2. PM: erklärung
3. Karplus strong
4. Schulübung: sounddesign challenge in pd.

4.4. Theorie, Schwingungsfähige systeme

Wellengleichung, eindimensionale saite:

$$c^2 \cdot \frac{\delta^2 f}{\delta x^2} = \frac{\delta^2 f}{\delta t^2} \quad (4.1)$$

Zwei lösungs ansätze:

- Bernoulli = ca. fourier, spectral, summe von sinus schwingungen.
- D'Alembert = ca. Physical modeling, zwei funktionen die entlang der saite reisen (in entgegengesetzte richtung).

$$y(x, t) = y^+ \left(t - \frac{x}{c} \right) + y^- \left(t + \frac{x}{c} \right) \quad (4.2)$$

D'Alembert sagt also, es gäbe zwei wellen, die in zwei richtungen entlang der saite reisen, die lässt sich einfach simulieren, mittels zwei in einander geschalteten delays.

4.5. Parktical Karplus-Strong

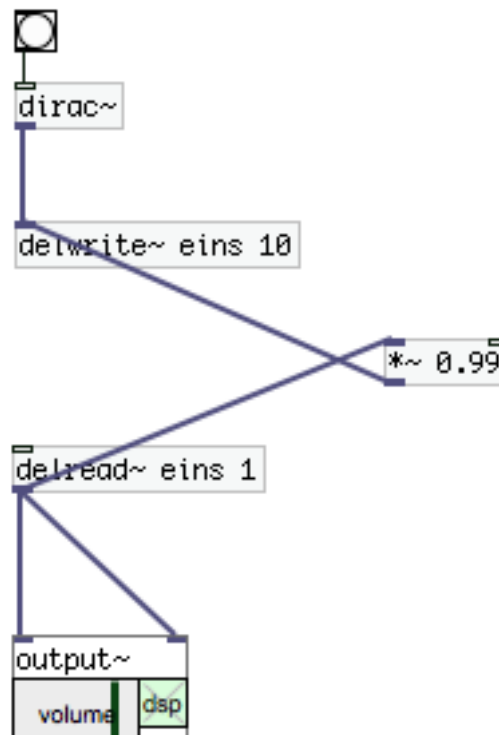


Figure 4.1.: Basic setup

4.6. Schulübung, Sounddesign challenge

zB:

- Vokal

Figure 4.2.: grundsätzliches Waveguide/Karplus Strong Modell.

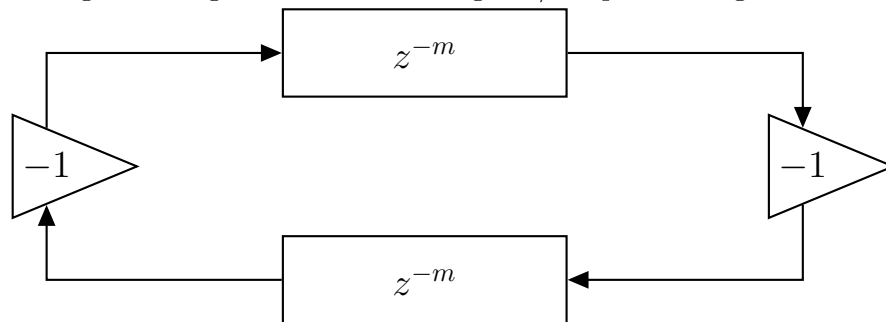
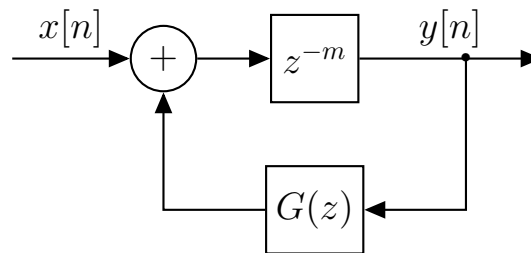


Figure 4.3.: einzelne Saite, $G(z)$ ein lowpass, „Loss Filter“



- E-gitarre
- beliebiges Physikalisches object

5. FFT

5.1. Notizen

vorbereitung:

Eulers identity, complex numbers.

Initialisierung Fourier transformation. Spectral filter Spectral Reverb, delay. Windowing Convolution, evtl. cross correlation freq. crossover spectral synyth, spectraum display.

evtl auch:

- send / receive, send / receive bei gui objekten.
- initialisierung
-

5.2. Fourier Transformation

Geschichte: Bernoulli, Euler, Gauß, Fourier

Eulers identität:

$$e^{ix} = \cos(x) + i \cdot \sin(x) \quad (5.1)$$

Fourier Transformation:

$$X(f) = \mathcal{F}\{x(t)\} = \int_{-\infty}^{\infty} x(t) e^{-j2\pi ft} dt \quad (5.2)$$

Inverse Fourier Transformation:

$$x(t) = \mathcal{F}^{-1}\{X(f)\} = \int_{-\infty}^{\infty} X(f) e^{j2\pi ft} df \quad (5.3)$$

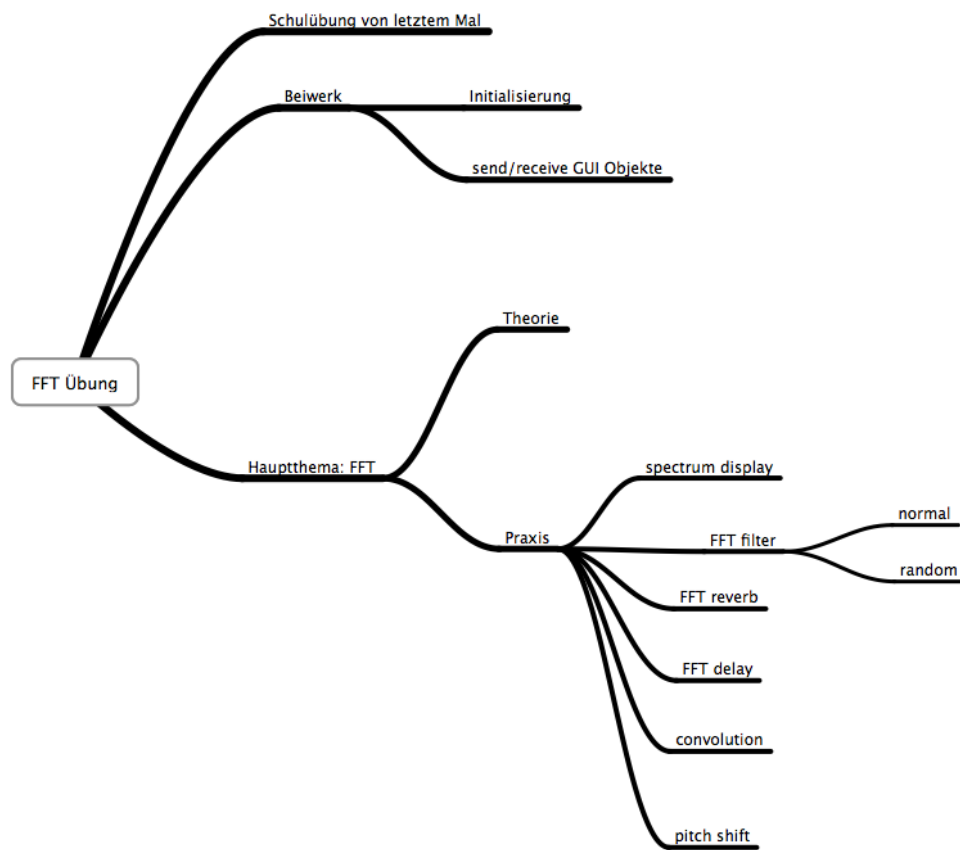


Figure 5.1.: caption

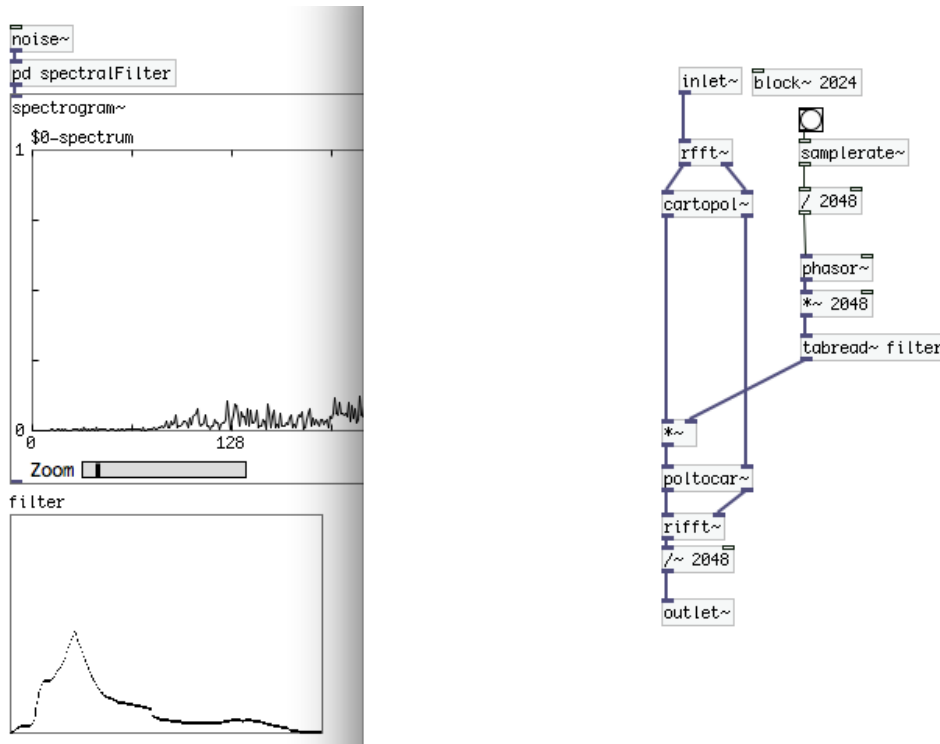


Figure 5.2.: spectralFilter.pd

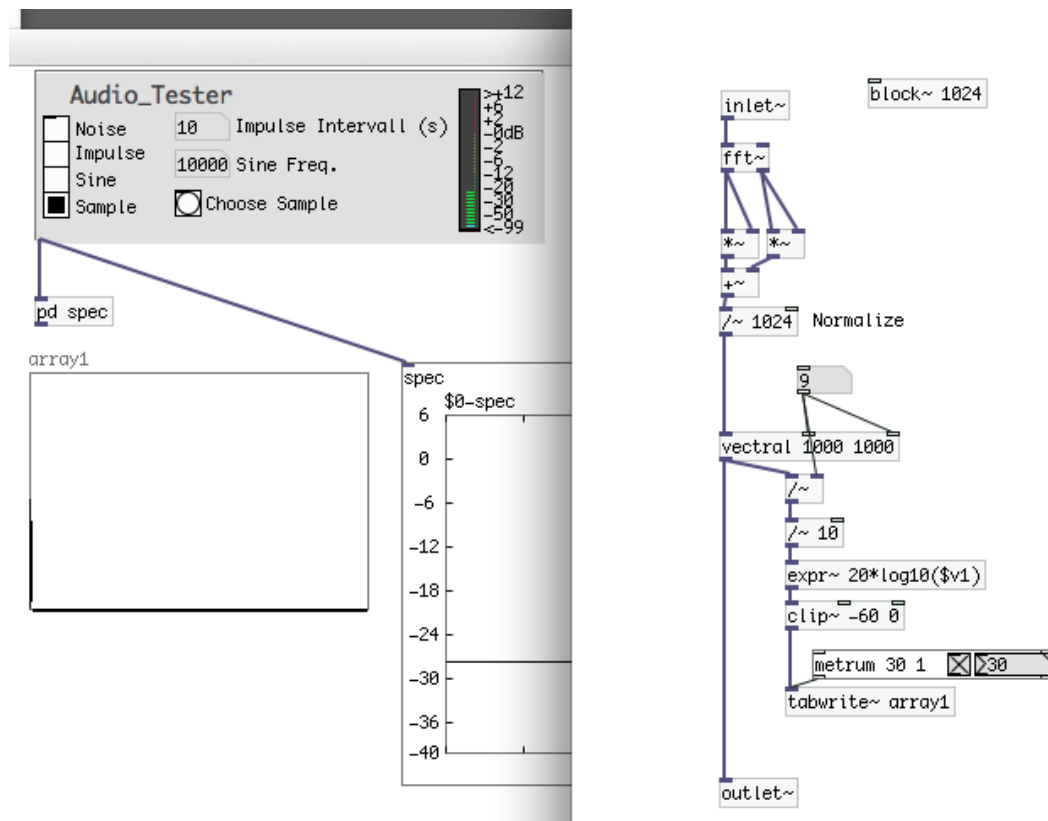


Figure 5.3.: showspectrum

Part II.

Semester 4

1. Jitter

1.1. Notizen

1.2. Überblick

- Stundenwiederholung
- Überblick über Software Pakete
- Vizzie
- kommunikation pd/max?
- Aufgabe: Video Laden, Abspielen, Audio analyse, modulation eines Effektes, Recording.
- Jitter Matrix (Wiederholung?)
- Audio analogien: jit.convolve, anti-aliasing(Open GL), jit.charmap(lookup table)
- Open GL, anti aliasing,

1.3. Andere Software Pakete für visuelle Echtzeit Arbeit

Graphik Programmierungsumgebungen

- Jitter
- Gem (pure Data)
- vvvv
- TouchDesigner
- processing
- Quarz Composer

„Professionelle“ weniger offene Systeme(weniger programmier-orientiert)

- Watchout

- Vizrt
- Ventuz

Game Engines

- Unreal 4
- Unity
- Cry Engine
- ...

1.4. Introduction

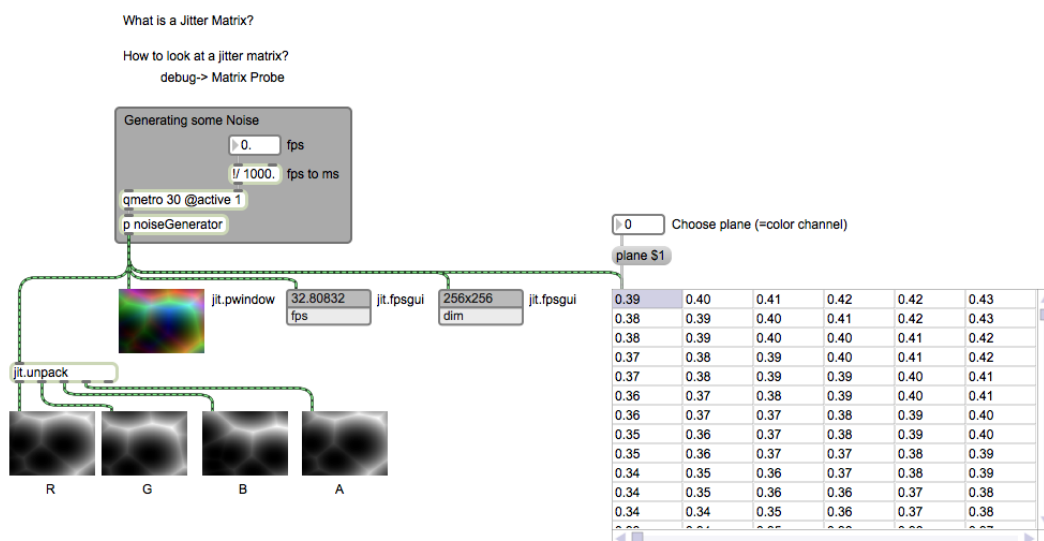


Figure 1.1.: How to look at a jitter matrix

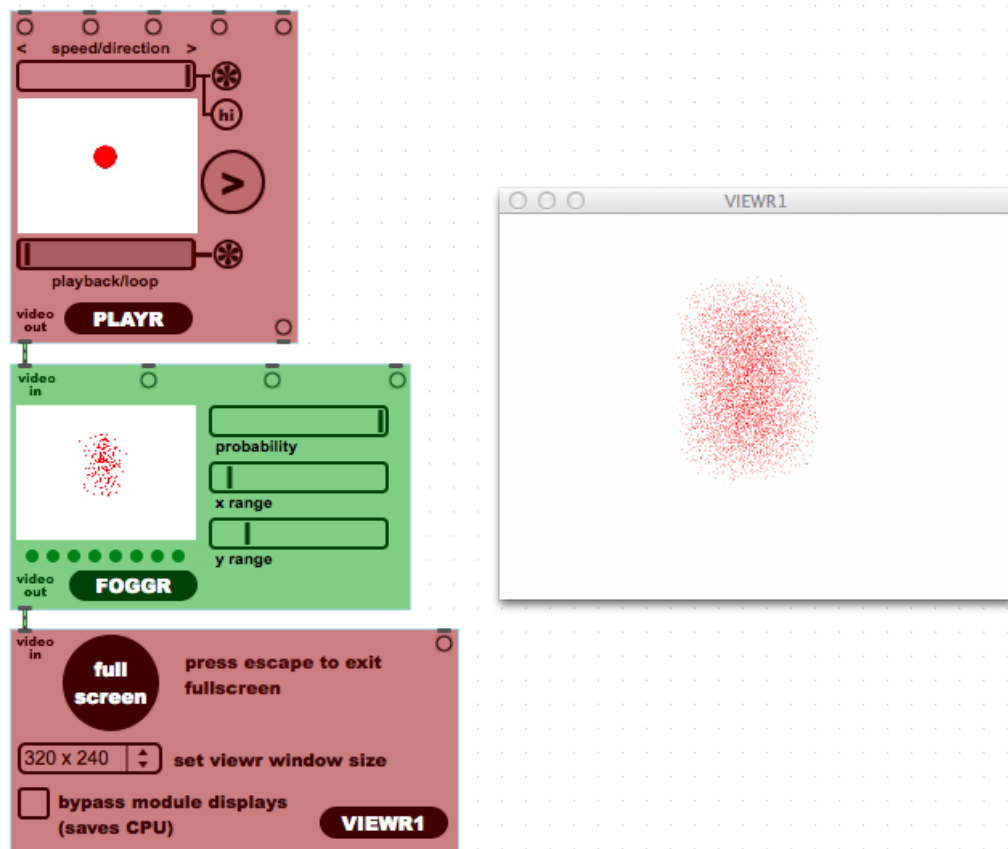


Figure 1.2.: hi Level Jitter

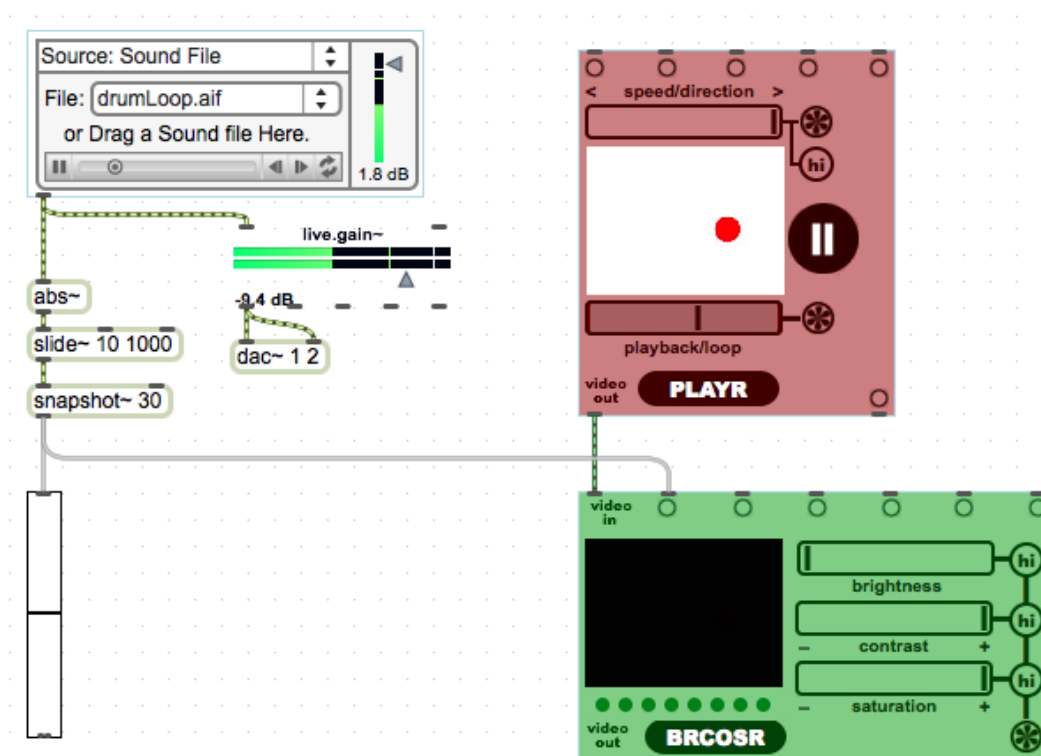


Figure 1.3.: VizzieAudioModulated

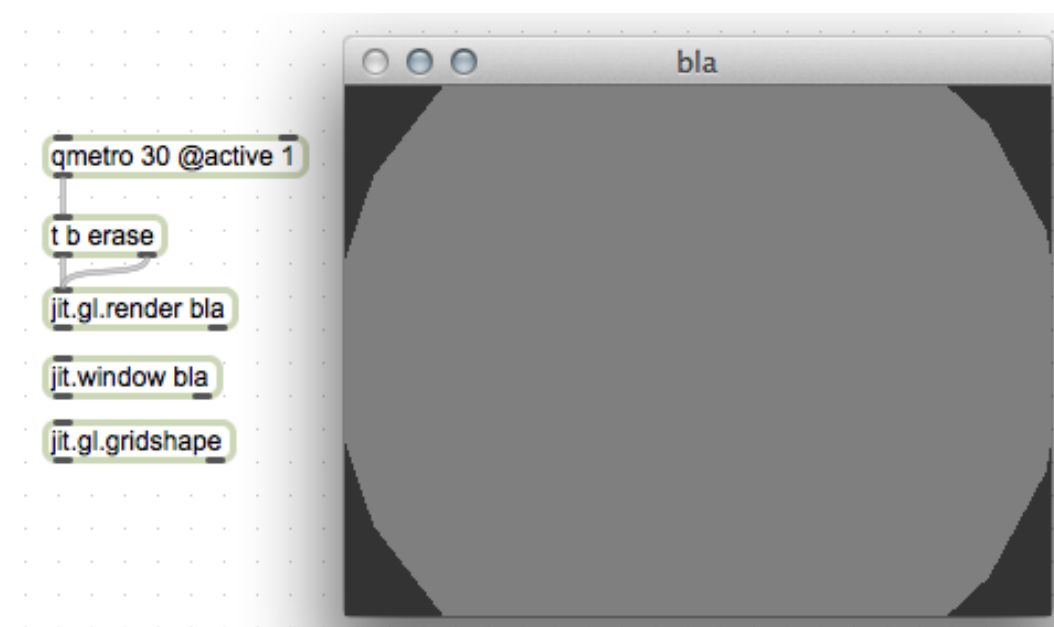


Figure 1.4.: basicGL

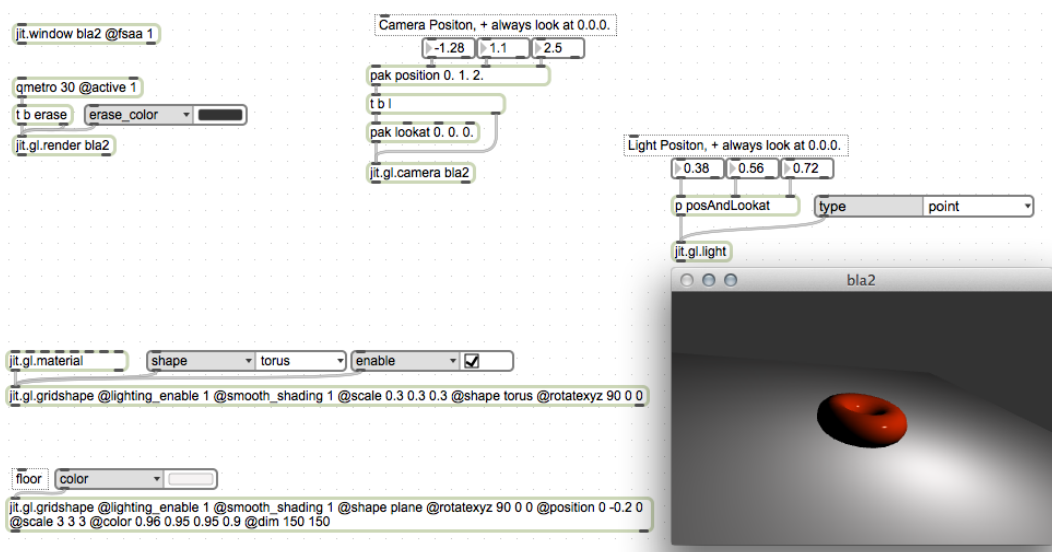


Figure 1.5.: openGLAdvanced

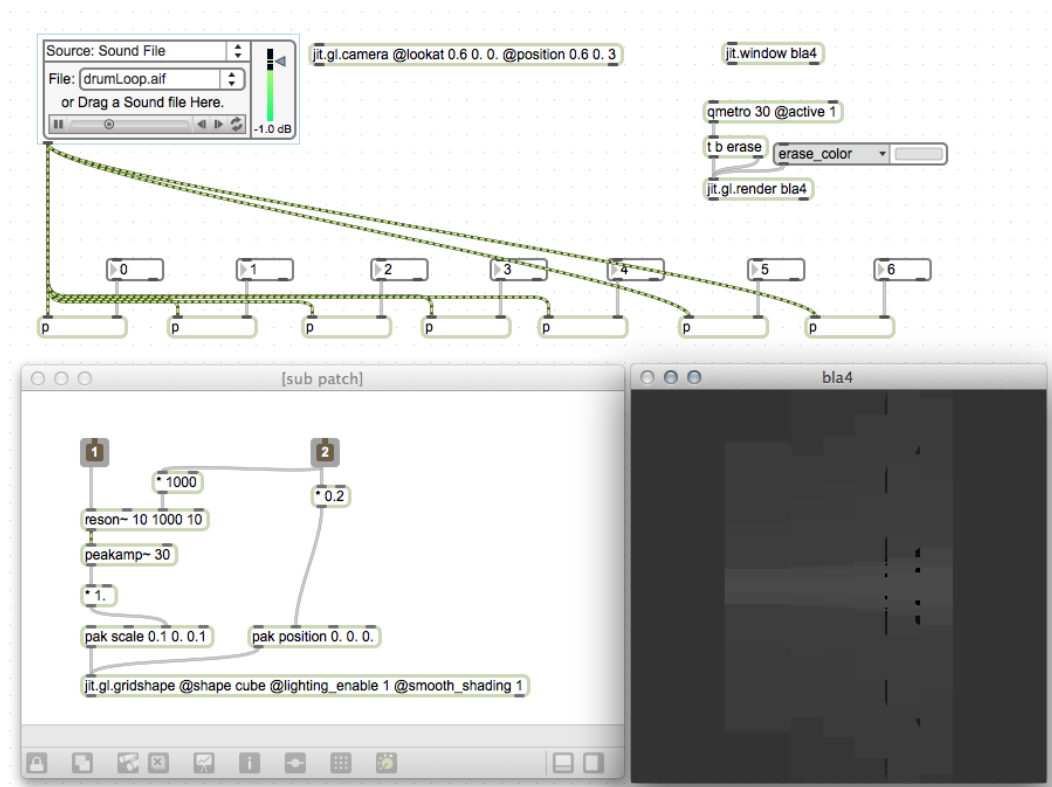


Figure 1.6.: specViz

List of Figures

1.1.	Linear Transfer function	3
1.2.	simpleSampler	5
1.3.	sound in Ram	6
1.4.	RamFilePlayback	7
1.5.	moreSampling.pd, a simplified version of granular sampling	8
1.6.	audioTester.pd, zu bauen als Hausübung	9
2.1.	Caravaggio, Narziss. ad Feedback: Echo liebt Narziss etc.	11
2.2.	Original Signal	13
2.3.	raetsel 1	14
2.4.	raetsel 2	14
2.5.	SimpleDelay	17
2.6.	generalized Effect structure	18
2.7.	Der Patch 01_combFilter.pd	19
2.8.	movingAverage	20
3.1.	metering techniques	24
3.2.	Simple Synth Voice, SynthVoice.pd	26
3.3.	Example Poly Host, patch 01_polySynthHost.pd	26
4.1.	Basic setup	28
4.2.	grundsätzliches Waveguide/Karplus Strong Modell.	29
4.3.	einzelne Saite, $G(z)$ ein lowpass, „Loss Filter“	29
5.1.	caption	31
5.2.	spectralFilter.pd	32
5.3.	showspectrum	33
1.1.	How to look at a jitter matrix	36
1.2.	hi Level Jitter	37
1.3.	VizzieAudioModulated	38
1.4.	basicGL	38
1.5.	openGLAdvanced	39
1.6.	specViz	39

Bibliography

Farnell, A. (2010). *Designing sound*. MIT Press, Cambridge, Mass.