EPRO/IAS

Bachelor Studiengang Medientechnik
Fachhochschule St.Pölten

Patrik Lechner

Wien, December 15, 2016

Introduction

Contents

Introduction II

I. Semester 3

1. Sampling, Waveshaping, und nicht Linearität **2**

1.1. Notizen 2

1.2. Uebersicht 3

1.3. Waveshaping 3

1.3.1. The simples case: a linear Transfer function. 4

1.3.2. Simple non-linearity: X^2 5

1.3.3. Why is Waveshaping useful? 7

1.4. Sampler 8

1.5. Hausübung 12

1.5.1. Testmodul 12

1.5.2. Distortion 12

II. Semester 4

Part I.

Semester 3

1. Sampling, Waveshaping, und nicht Linearität

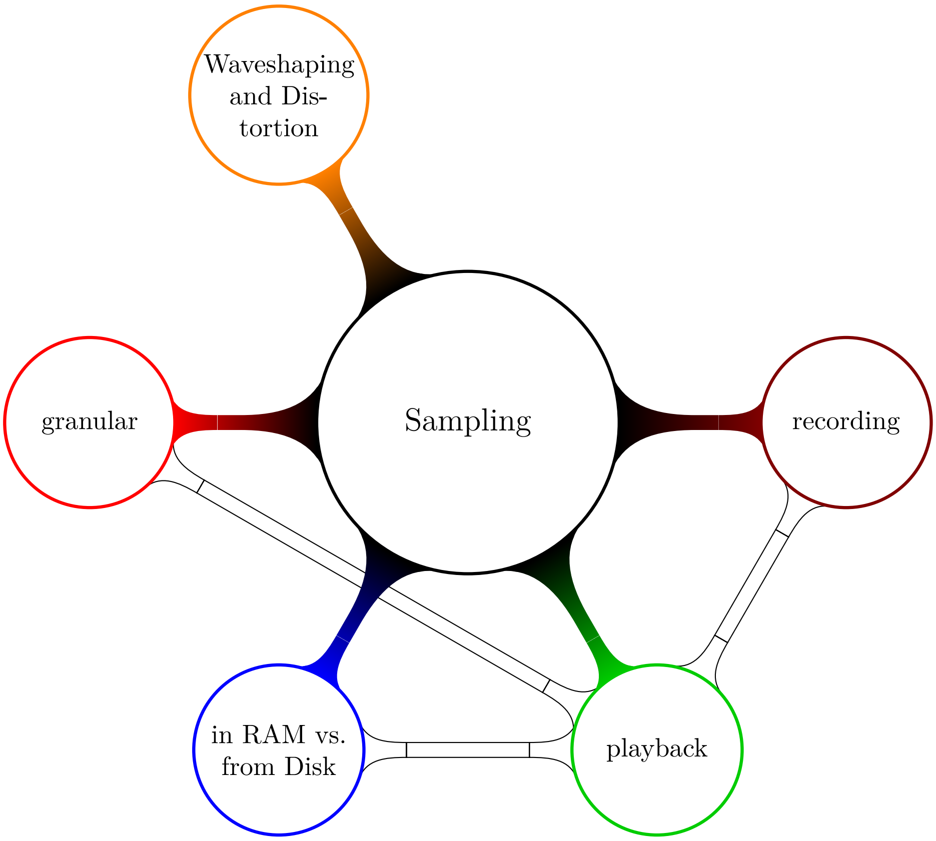

Figure 1.1.: Lecture Contents

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 quote, page “wavesahper”:
„The mathematics of non-linear operations on audio signals is difficult, and not well understood.“ Waveshaping means distortion. It add overtones.

1.3.1. The simples case: a linear Transfer function.

See 1.2. A linear transfer function is used as a lookup table for a sinosoidal input.

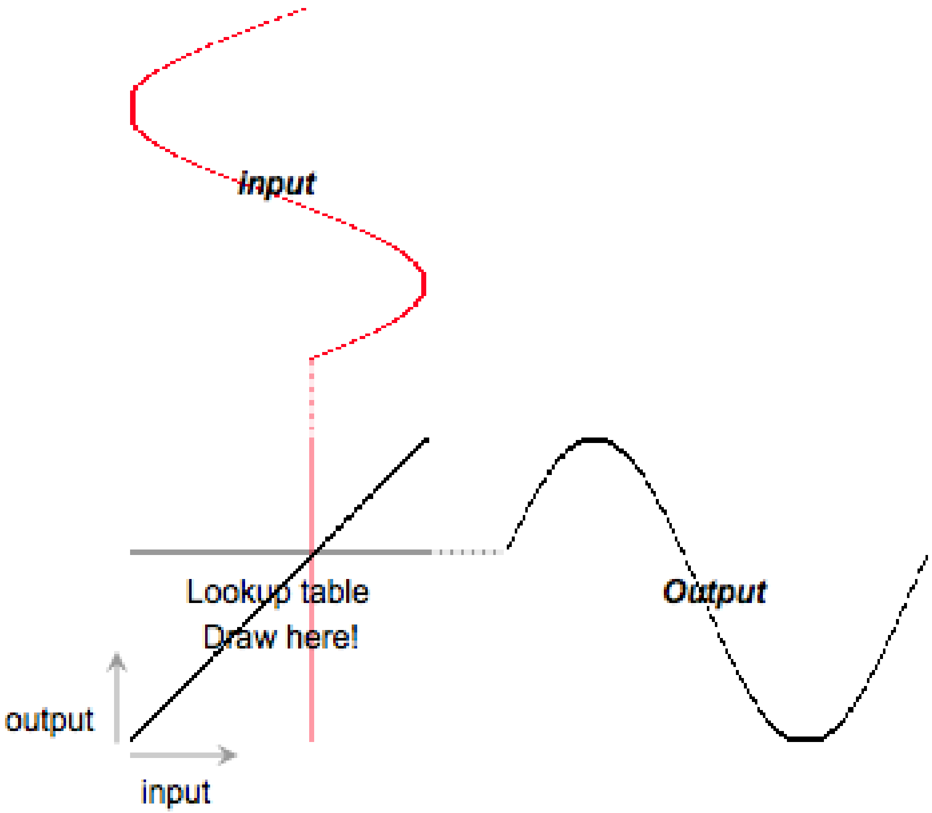

Figure 1.2.: Linear Transfer function

A transfer function in the sense of a waveshaper (a "transfer function" might also mean frequency response in other contexts) is a simple look-up function. Waveshaping means to use an input wave *look up* values in a table or function. A linear transfer function, let's call it l, results in no change, since, by definition, it returns $l(x) = x$. This means, that whatever value we pass in, we get the same value out. Non-liner transfer functions behave differently. They map they input to other values, such as $f(x) = x^2$. It may seem trivial, but if we put 2 into f we get 4 as an output. If we now plot this function we get:

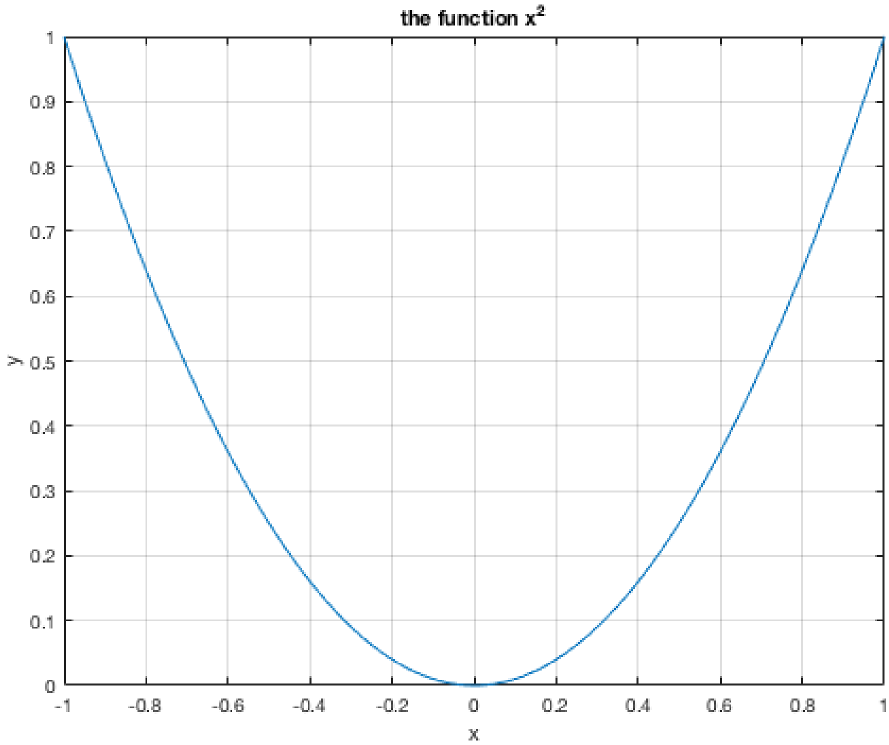

Figure 1.3.: The function $f(x) = x^2$

1.3.2. Simple non-linearity: X^2

A 2nd order polynomial shall be analyzed. The function

$$f(x) = x^2 \tag{1.1}$$

is used and also plotted in figure 1.3.

We can simply plot what happened is we apply the function before trying to understand analytically:

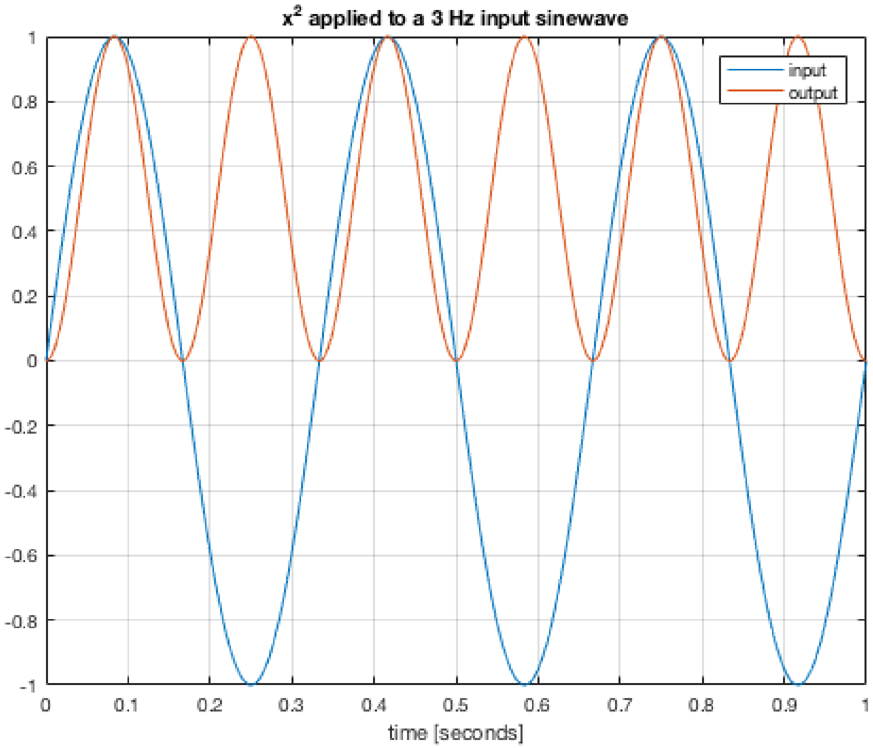

Figure 1.4.: Applying the square function to an input sine wave.

Weird, the input seems to double in frequency. Let's try to understand what's happening.
So we calculate what happens if we send a cosine through this function, so let's take:

$$x = cos(\omega) \tag{1.2}$$

wit arbitrary ω. We can just ignore ω here for a while. Usually, there should be some indexing variable in the cosine function if we want to describe an oscillator that moves over time, but let's also skip that.
So applying our square function we of course get:

$$y = cos(\omega)^2 \tag{1.3}$$

This again results in:

$$y = cos(\omega) \cdot cos(\omega) \tag{1.4}$$

So far so trivial. Note that a multiplication of two oscillators is called *Amplitude Modulation* (actually, in this case we encounter "Ring Modulation", but let's ignore that also), and we know things about Amplitude modulation, namely:

When multiplying two oscillators, we get sum and difference of the two input frequencies. (And the whole output is attenuated by 6dB)

The above statement in equation form:

$$cos(a) \cdot cos(b) = \frac{cos(a+b) + cos(a-b)}{2} \tag{1.5}$$

We could also have looked up this *trigonometric identity*. This means for our experiment with our cosine squared:

$$y = \frac{cos(\omega + \omega) + cos(\omega - \omega)}{2} \tag{1.6}$$

So:

$$y = \frac{cos(2 \cdot \omega) + cos(0)}{2} = \frac{cos(2 \cdot \omega)}{2} + \frac{1}{2} \tag{1.7}$$

We arrive at the same result! **But is this true for every input? That would mean we just built a frequency shofter, did we? No.** Waveshaping is much more complicated, which is immediately obvious when we try to do the same t two oscillators:

$$x = cos(\omega_1) + cos(\omega_2) \tag{1.8}$$

then

$$y = (cos(\omega_1) + cos(\omega_2))^2 \tag{1.9}$$

$$y = cos(\omega_1)^2 + cos(\omega_2)^2 + 2 \cdot cos(\omega_1) \cdot cos(\omega_2) \tag{1.10}$$

And finally:

$$y = \frac{cos(2 \cdot \omega_1)}{2} + \frac{1}{2} + \frac{cos(2 \cdot \omega_2)}{2} + \frac{1}{2} + 2 \cdot (\frac{cos(\omega_1 + \omega_2) + cos(\omega_1 - \omega_2)}{2}) \tag{1.11}$$

1.3.3. Why is Waveshaping useful?

The output spectrum is dependent on the input amplitude. This makes it easy to create complex involving spectra.

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 transferkuntion 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.

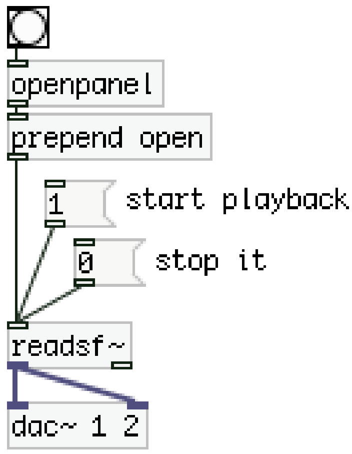

Figure 1.5.: simpleSampler

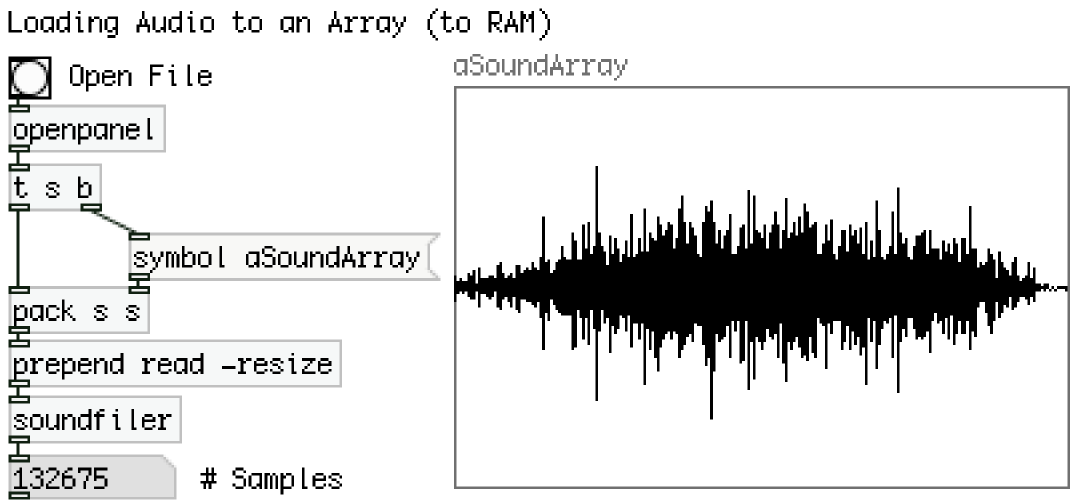

Figure 1.6.: sound in Ram

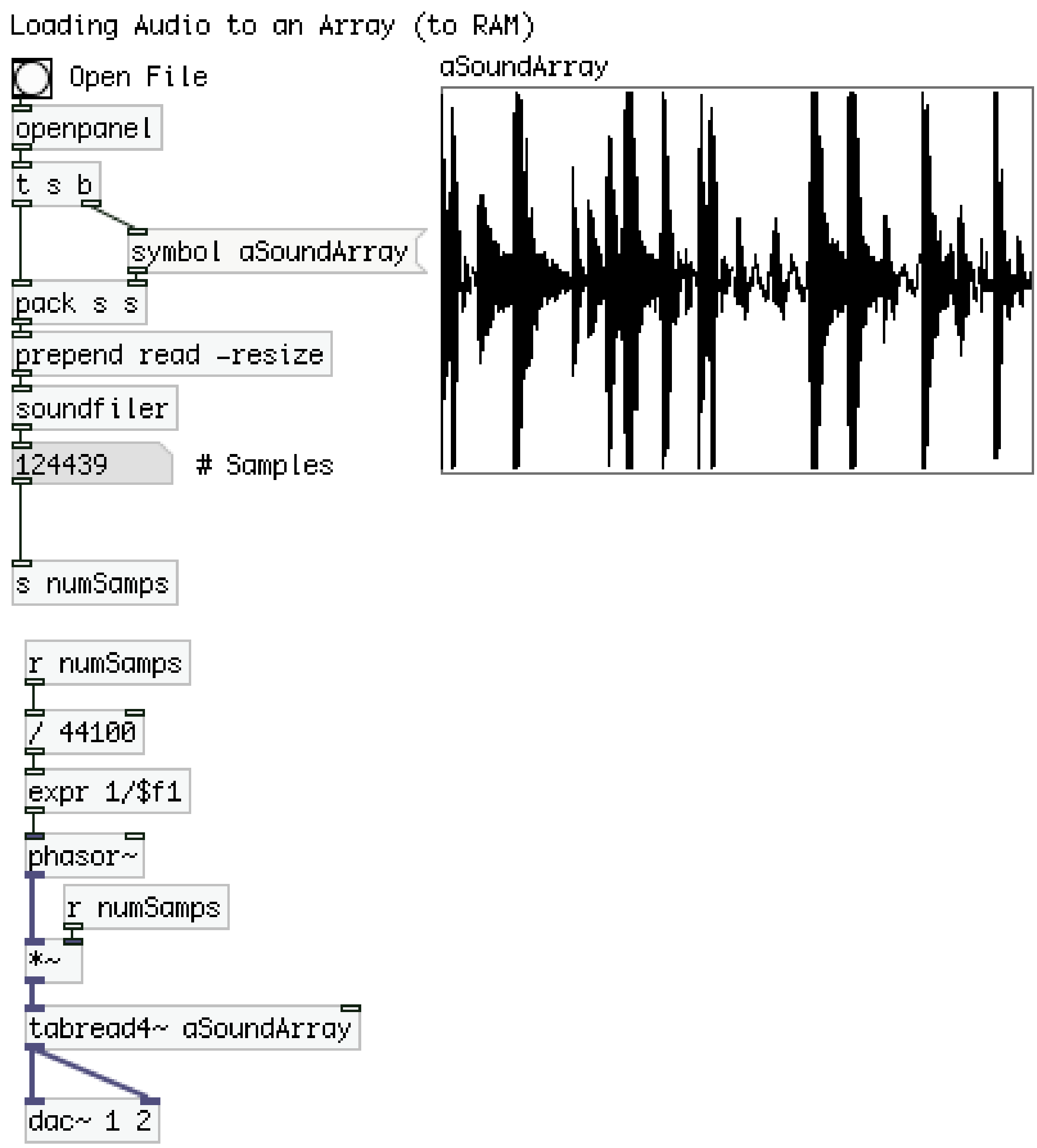

Figure 1.7.: RamFilePlayback

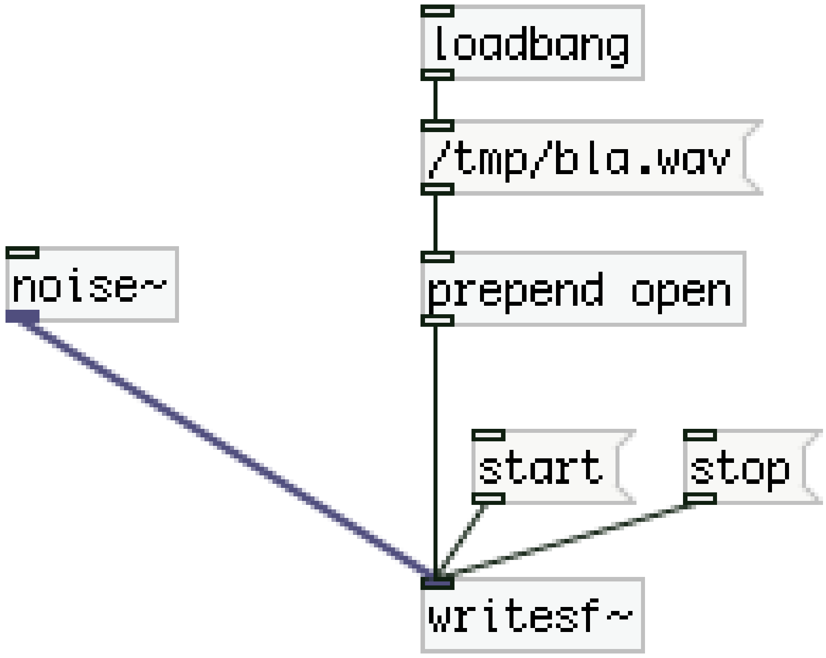

Figure 1.8.: writing Audio to disk

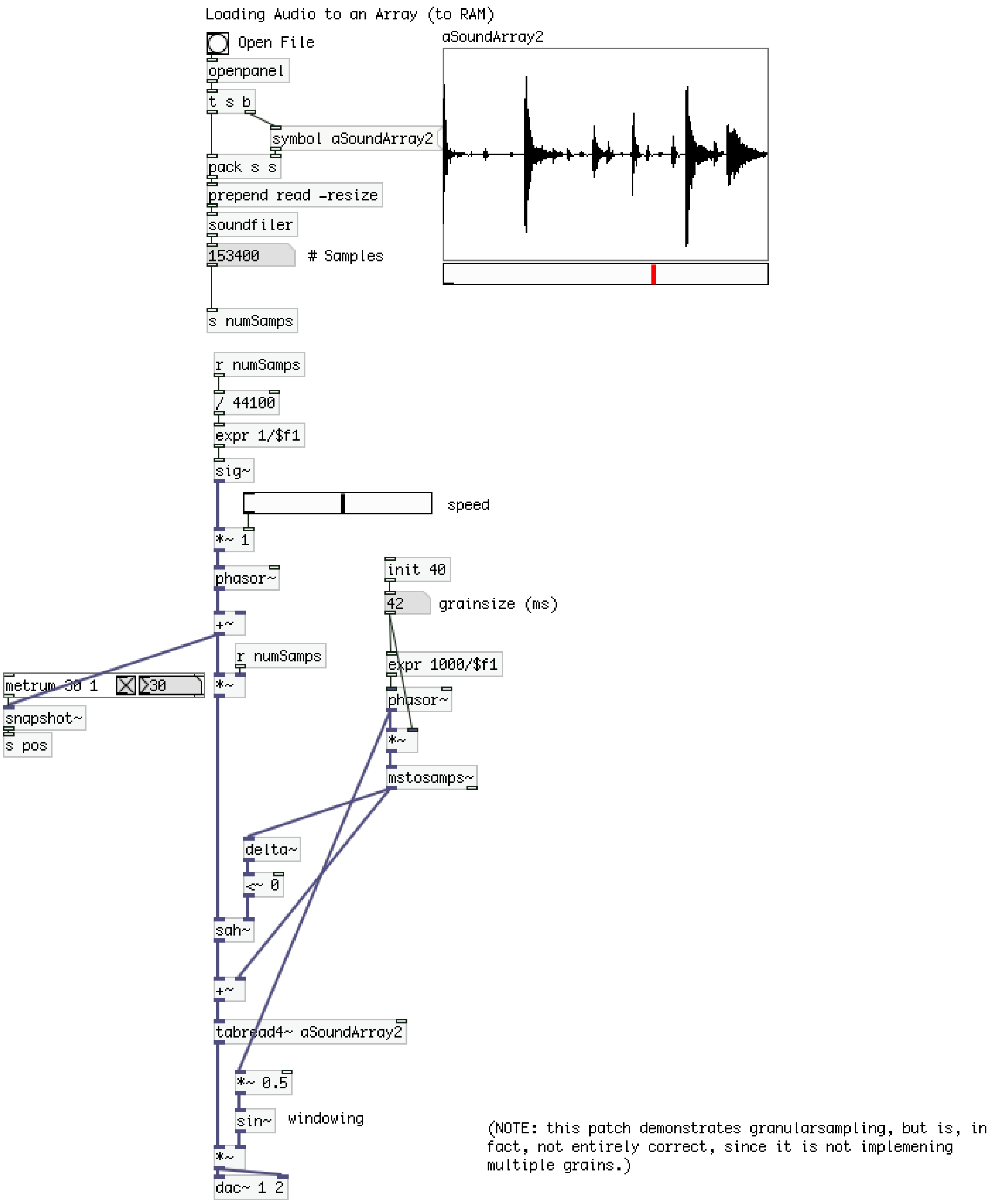

Figure 1.9.: moreSamplimg.pd, a simplified version of granular sampling

1.5. Hausübung

1.5.1. Testmodul

baue ein audio Testmodul mit folgener 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)

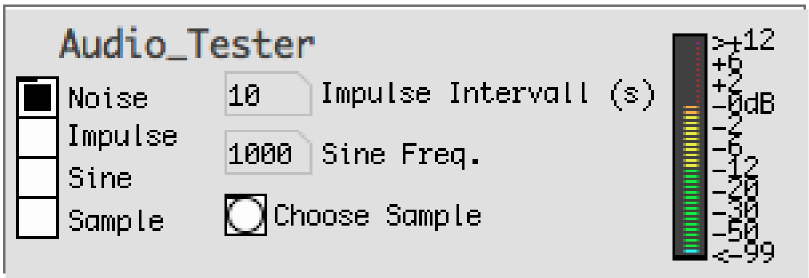

Figure 1.10.: audioTester.pd, zu bauen als Hausübung

1.5.2. Distortion

Baue eine besonders wohlklingende Distortion, inkl. User interface.

Part II.

Semester 4

List of Figures

1.1. Lecture Contents 2
1.2. Linear Transfer function 4
1.3. The function $f(x) = x^2$ 5
1.4. Applying the square function to an input sine wave. 6
1.5. simpleSampler 8
1.6. sound in Ram 8
1.7. RamFilePlayback 9
1.8. writing Audio to disk 10
1.9. moreSamplimg.pd, a simplified version of granular sampling 11
1.10. audioTester.pd, zu bauen als Hausübung 12

Bibliography

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