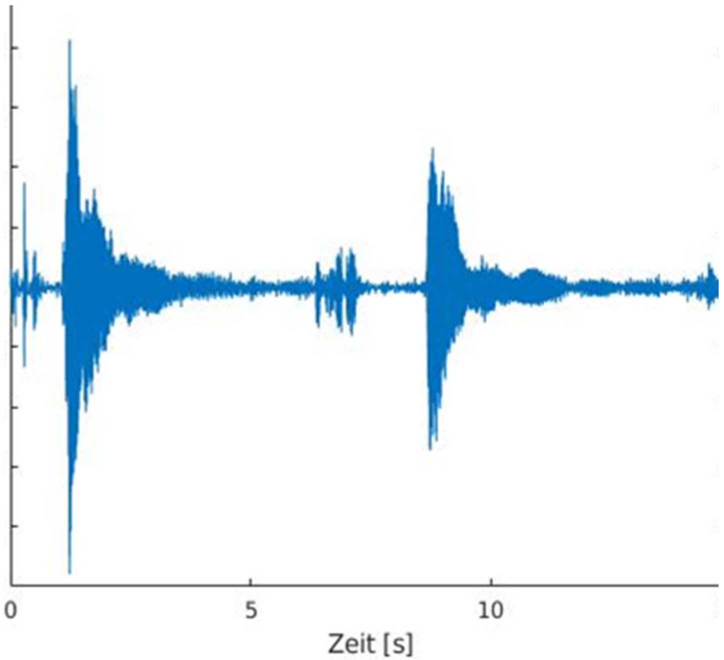
@ waves

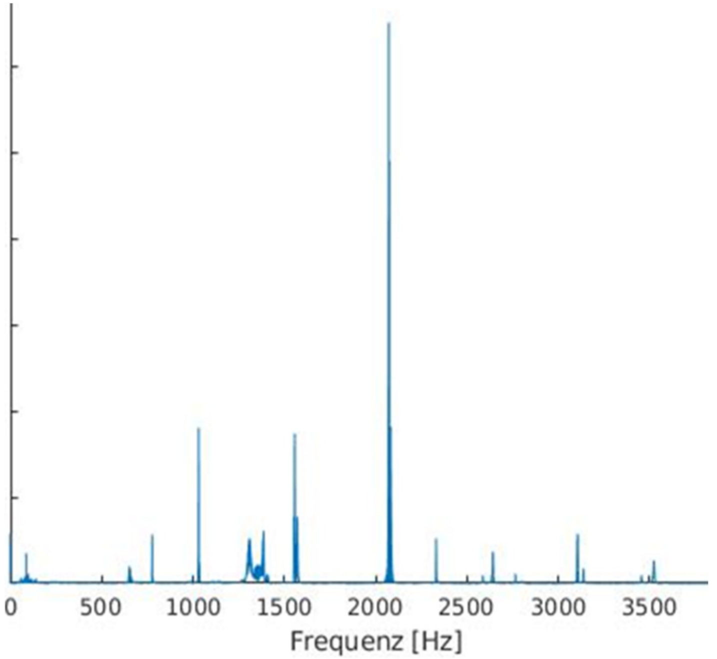
von **Auzinger Günter,**
Donnermair Maximilian, Eckertsberger Gabriel, Eybl Julian, Haimböck Steffi,
Heiligenbrunner Lukas, Meindl Emil, Mottas Maximilian, Penzinger Johanna,
Salfer Michael, Wollersberger Julian

Trumpet Waves

von Musik zur Fourier-Transformation



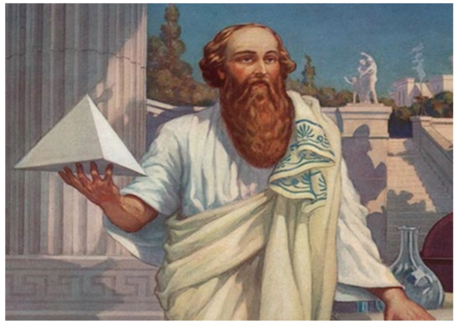
⇒



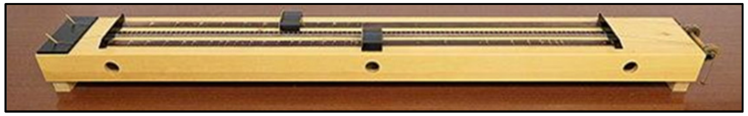
Übersicht

- Pythagoras
- Töne
- Abtastung
- Formeltheorie
- Fourier Transformation
- Beispiele

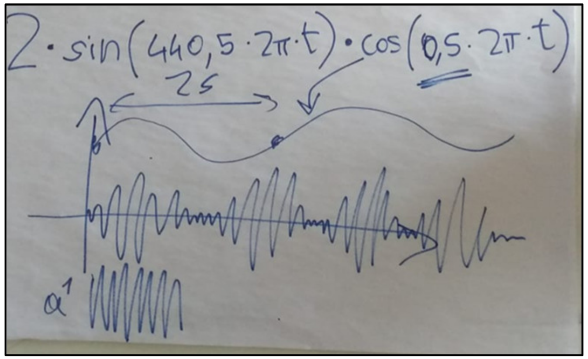
Pythagoras: Vater der Musiktheorie



- Erste mathematische Darstellung der Musik
- Verhältnisse: kleine Zahlen = Götter
- Monochord konstruiert: verschiebbarer Steg



- Stimmung: Schwebungen
- ungünstige Zahlenverhältnisse



- Sinüsse & Cousinen
- Additionstheoreme

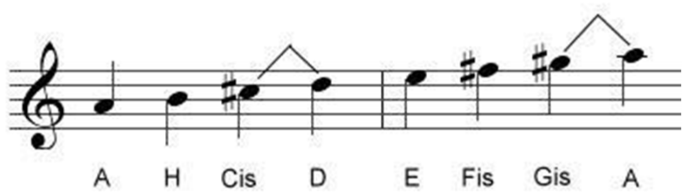
$$\sin(\alpha+\beta) = \sin\alpha \cdot \cos\beta + \cos\alpha \cdot \sin\beta$$
$$\cos(\alpha+\beta) = \cos\alpha \cdot \cos\beta - \sin\alpha \cdot \sin\beta$$
$$\sin(\alpha-\beta) = \sin\alpha \cdot \cos\beta - \cos\alpha \cdot \sin\beta$$
$$\cos(\alpha-\beta) = \cos\alpha \cdot \cos\beta + \sin\alpha \cdot \sin\beta$$

Tonentstehung

- Instrumente bestehen grundsätzlich aus einem Generator und einem Resonator
- nur eine transiente Strömung für Musik z.B.: Flöte
- Trompete: Obertöne, überblasen, Naturtonreihe
- spezielle Instrumente (Zink, Serpent)

Töne auf der Gitarre

Abmessen der Saitenlängen und
Rückschluss auf Tonfrequenz:

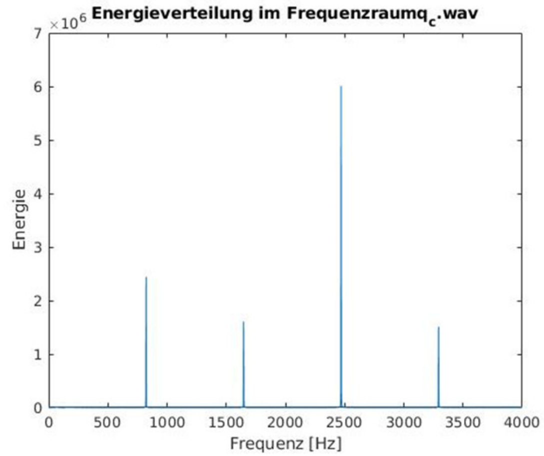


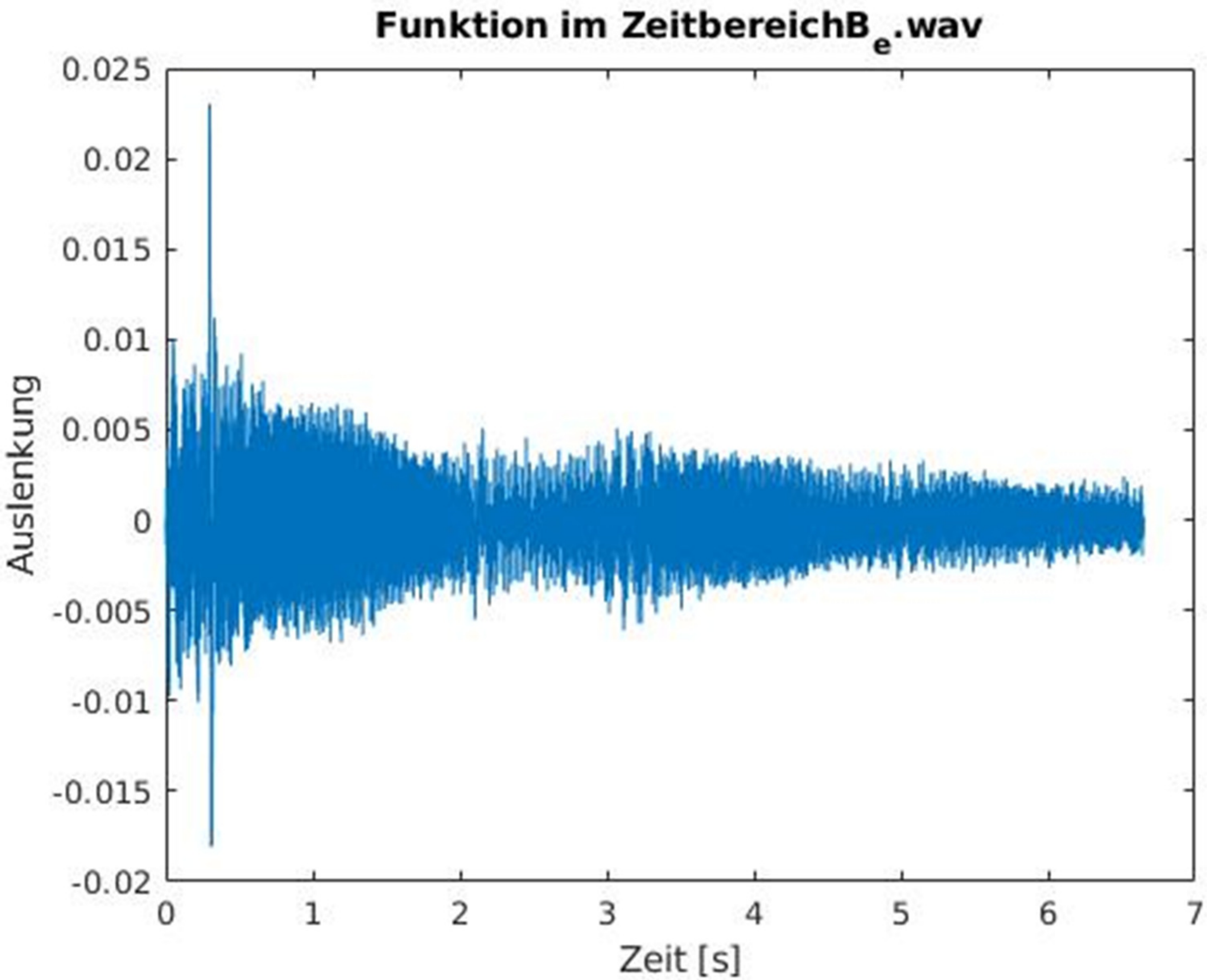
a = 650mm
h = 580mm
cis1 = 519mm
d1 = 487mm
e1 = 433mm
fis1 = 386mm
gis1 =

Konsonanz:
a → a' = 650:325 = 2:1 Oktav
a → e' = 650:433 = 3:2 Quint
a → d' = 650:487 = 4:3 Quart

Dissonanz:
a → h = 650 : 580 = 1,12:1 gr. Sekun
→ klingt nicht gut

Flageolett

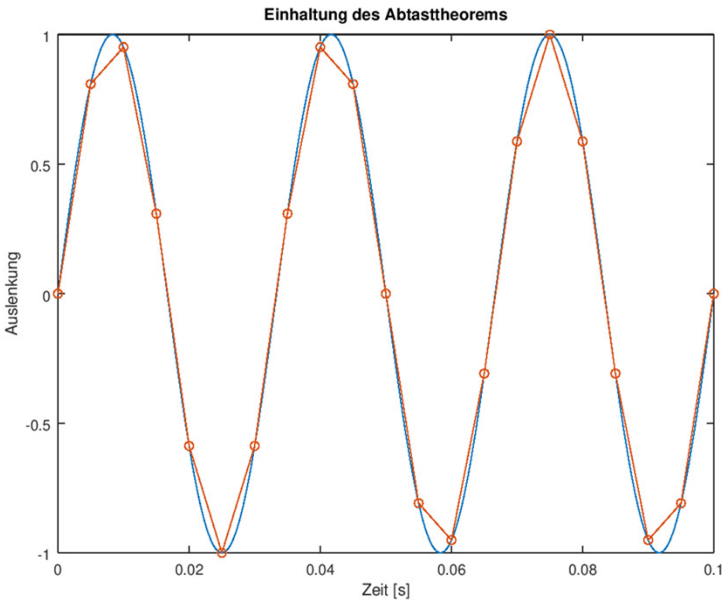


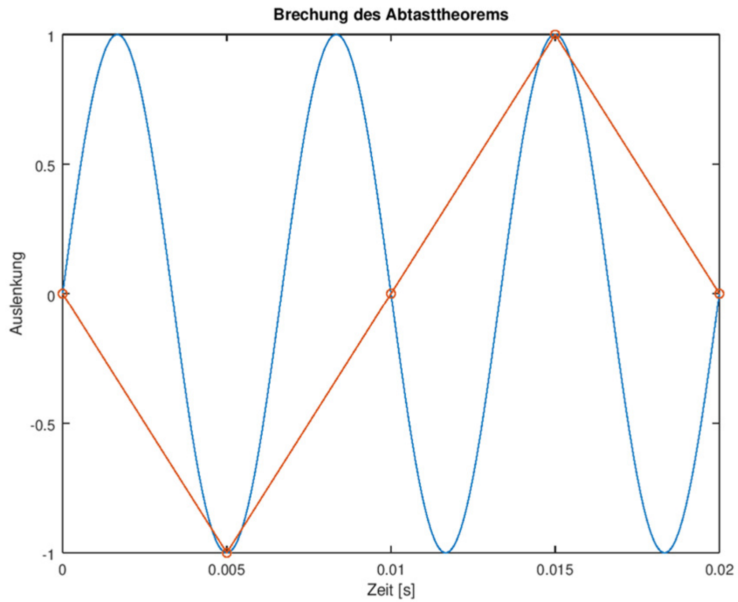


Funktion im ZeitbereichB$_e$.wav
0.025
0.02
0.015
0.01
0.005
0
-0.005
-0.01
-0.015
-0.02
Auslenkung
0
1
2
3
4
5
6
7
Zeit [s]

Abtastung - Sampling

- Signal muss abgetastet werden
  - Allgemeines Abtasttheorem:
  - $f_{abtast} \geq 2 \cdot f_{max}$





$$f(t) = s_1 \cdot sin(t) + s_2 \cdot sin(2t) + s_3 \cdot sin(3t) + \ldots +$$
$$+c_0 + c_1 \cdot cos(t) + c_2 \cdot cos(2t) + c_3 \cdot cos(3t) + \ldots$$

$$\vec{S_1} = \begin{pmatrix} sin(t_0) \\ sin(t_1) \\ sin(t_2) \\ \ldots \end{pmatrix}$$

$$\vec{F} = s_1 \cdot \vec{S_1} + s_2 \cdot \vec{S_2} + s_3 \cdot \vec{S_3} + \ldots +$$
$$+c_0 \cdot \vec{C_0} + c_1 \cdot \vec{C_1} + c_2 \cdot \vec{C_2} + c_3 \cdot \vec{C_3} + \ldots$$

$$s_k = \frac{2}{n} \cdot \vec{F} \cdot \vec{S_k}$$

$$c_k = \frac{2}{n} \cdot \vec{F} \cdot \vec{C_k}$$

Diskrete Fourier Transformation

- Grundform der Transformation
- Rechnet für einzelne Frequenzen

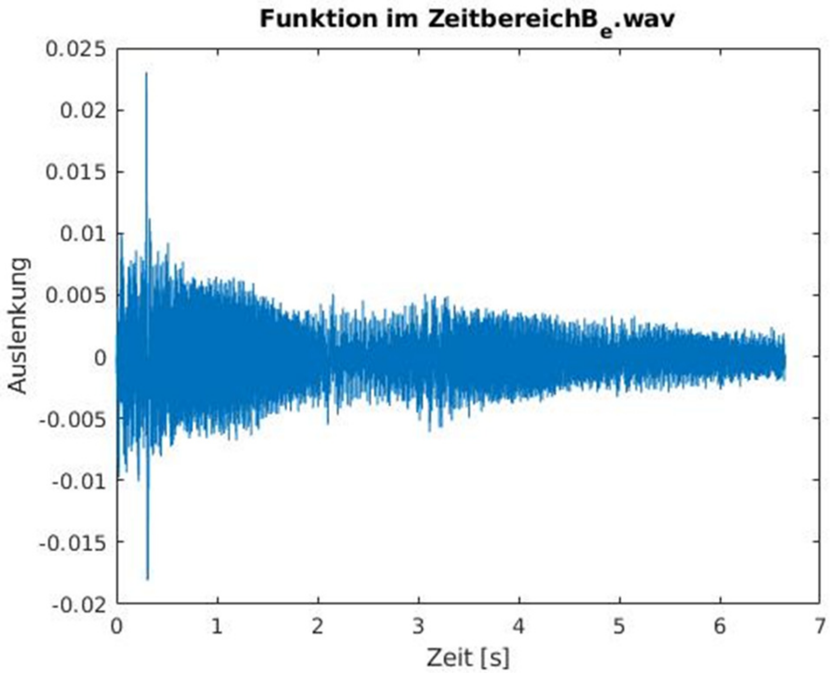
$$\mathbf{F} = \begin{bmatrix} \omega_N^{0\cdot 0} & \omega_N^{0\cdot 1} & \cdots & \omega_N^{0\cdot (N-1)} \\ \omega_N^{1\cdot 0} & \omega_N^{1\cdot 1} & \cdots & \omega_N^{1\cdot (N-1)} \\ \vdots & \vdots & \ddots & \vdots \\ \omega_N^{(N-1)\cdot 0} & \omega_N^{(N-1)\cdot 1} & \cdots & \omega_N^{(N-1)\cdot (N-1)} \end{bmatrix}$$

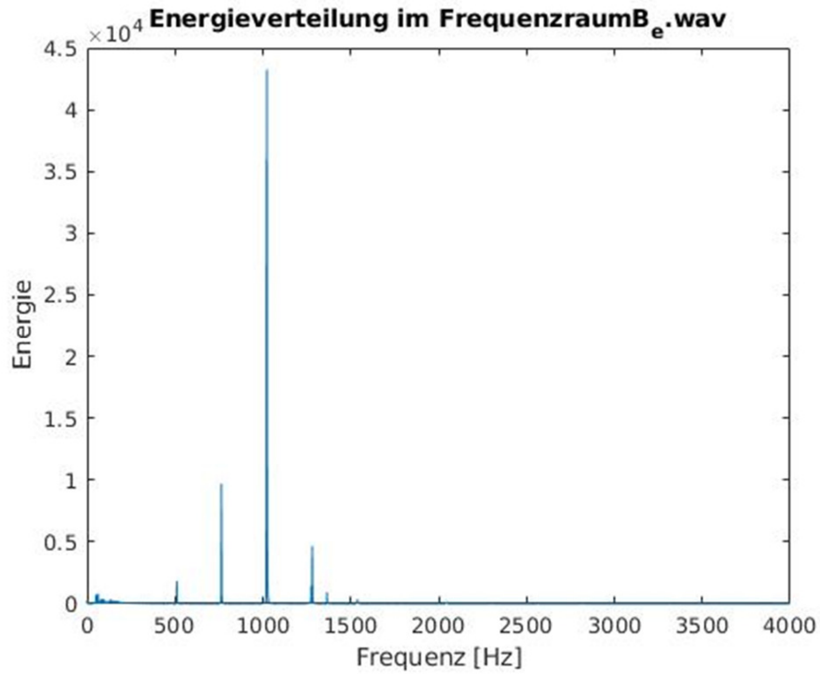
```
n = length(fTime);
dTime = 1/samplingRate; % [s]
maxTime = dTime*n;
dOmega = 2*pi/(n*dTime);
maxFrequIndex = floor(maxFrequ/dOmega);
timeValues = (0:n-1)*dTime;
timeValsScaled2pi = timeValues*2*pi/maxTime;
omegaValues = (0:maxFrequIndex-1)*dOmega;
fFrequSin = zeros(1,maxFrequIndex);
fFrequCos = zeros(1,maxFrequIndex);

for k = 0:(maxFrequIndex-1)
  fFrequSin(k+1) =
dot(fTime,sin(k*timeValsScaled2pi));
  fFrequCos(k+1) =
dot(fTime,cos(k*timeValsScaled2pi));
end
```

Fast Fourier Transformation

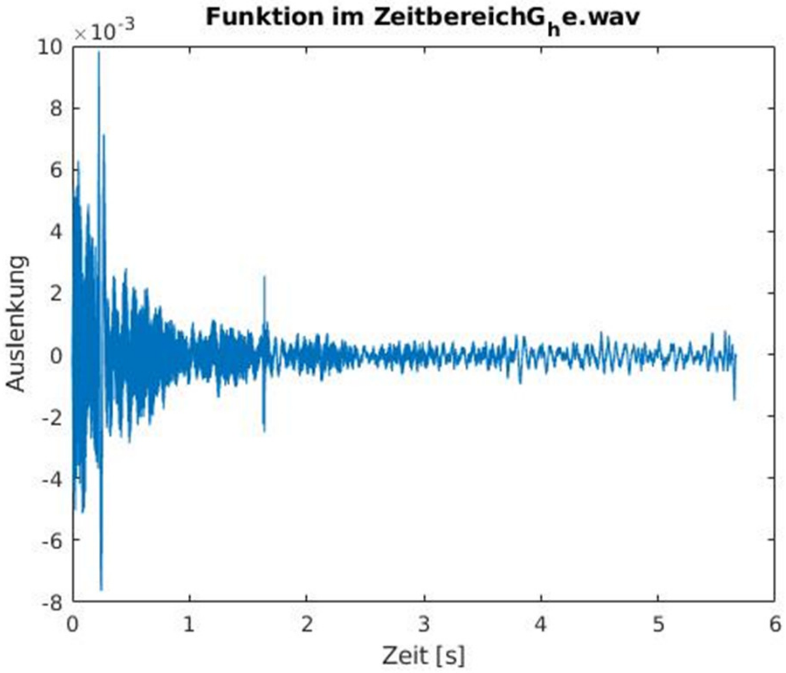
- DFT langsam zum Berechnen: $O(n^2)$
- FFT viel schneller: $O(n * \log(n))$
  Bsp: bei 1 Sekunde 4000-fach schneller
- Dafür viel komplexer:
  - Komplexe Zahlen
  - Komplexe Exponenten
  - Matrix-Darstellung
  - Primzahlen
- Jemand anders hat’s schon programmiert
  -> einfach verwenden

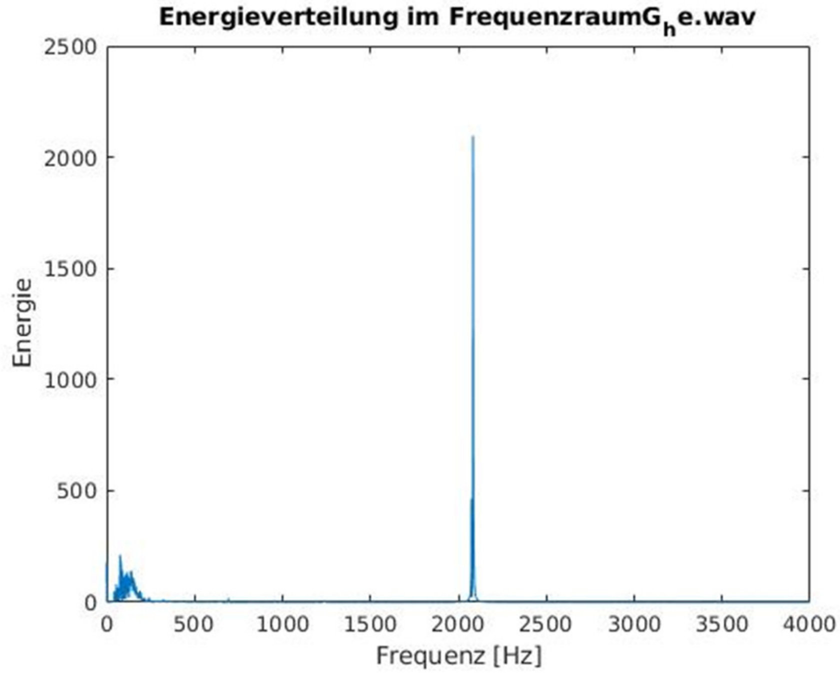
Lösung des Beispiels: Basston E



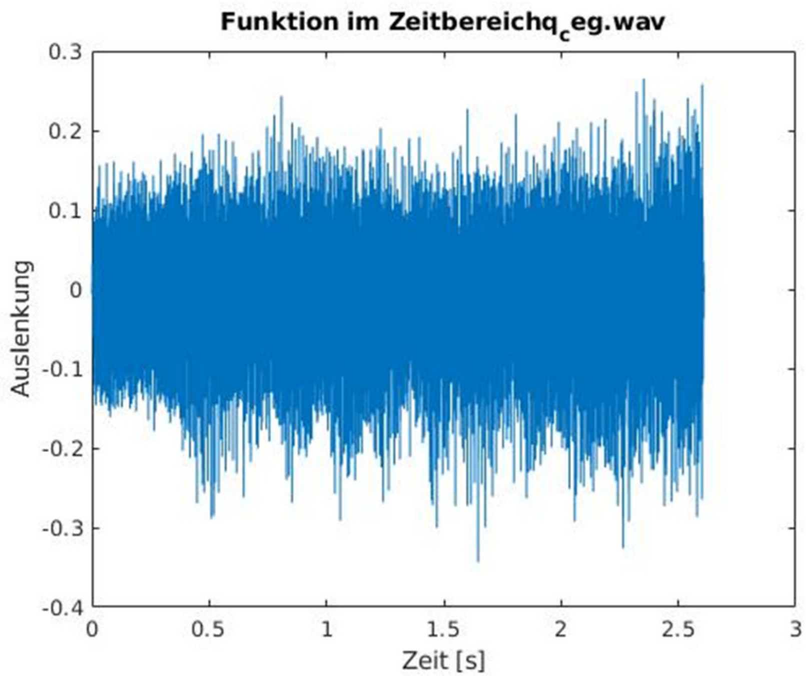


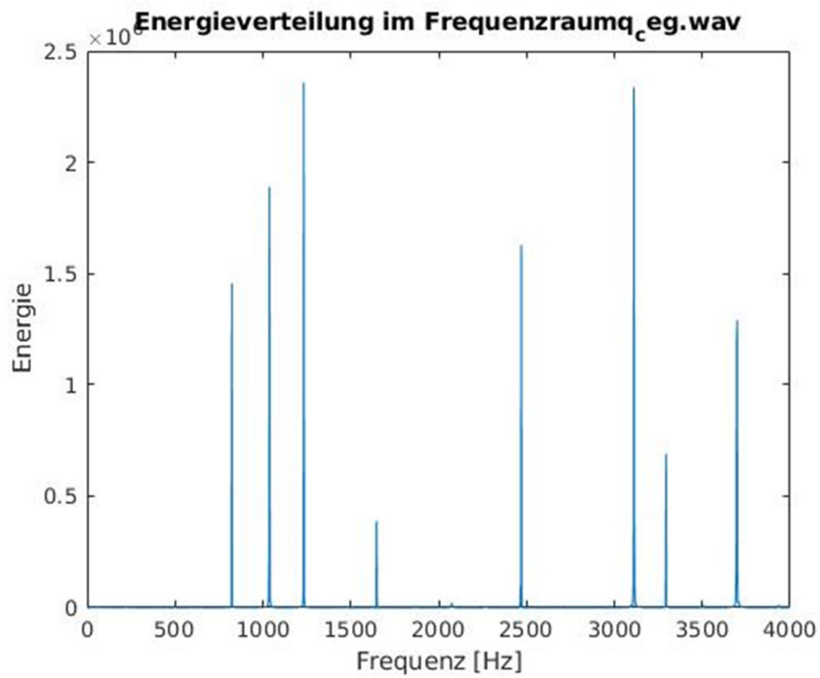
Gitarrenton hohes e



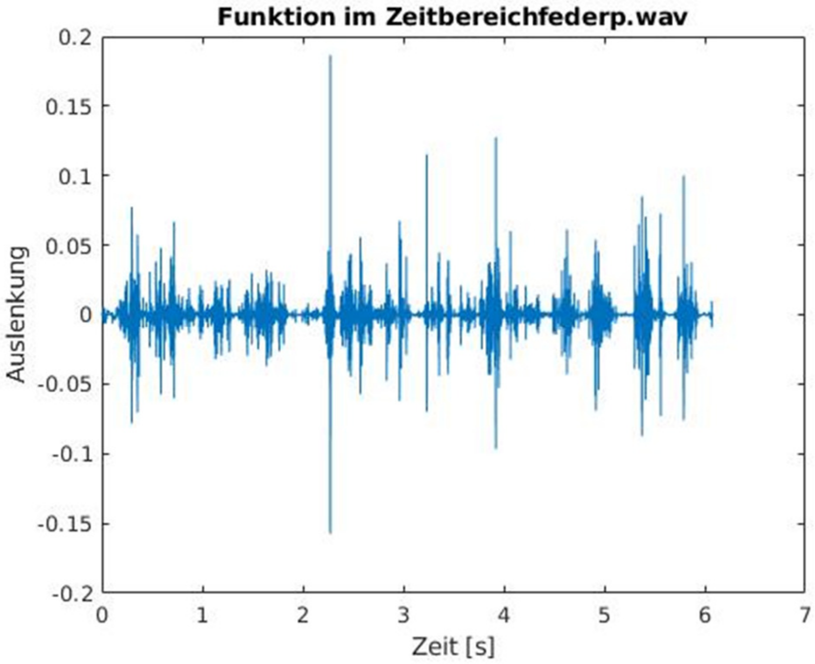


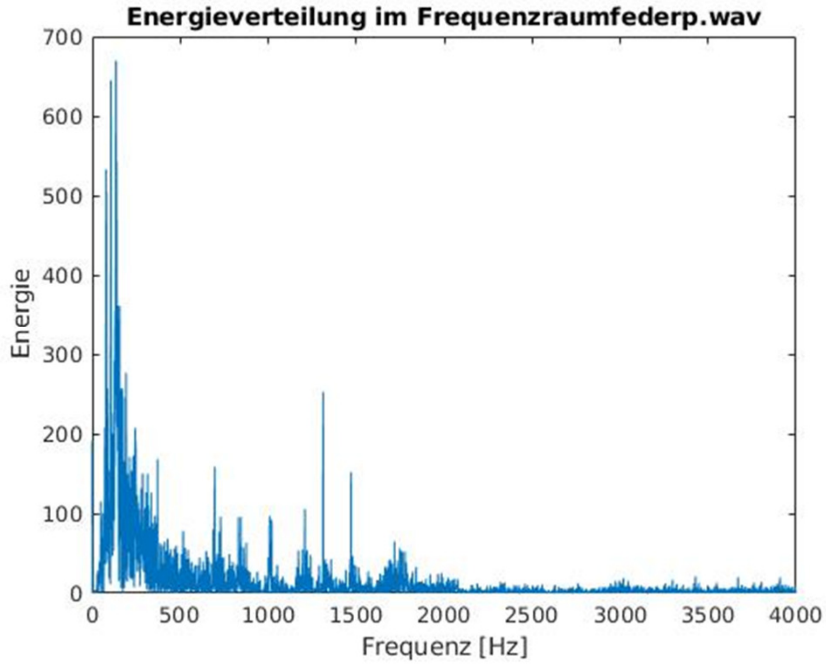
Akkordeon-Dreiklang: C-Dur (c-e-g)





Federschachtelrascheln





Kugelschreiber fällt auf Tischplatte