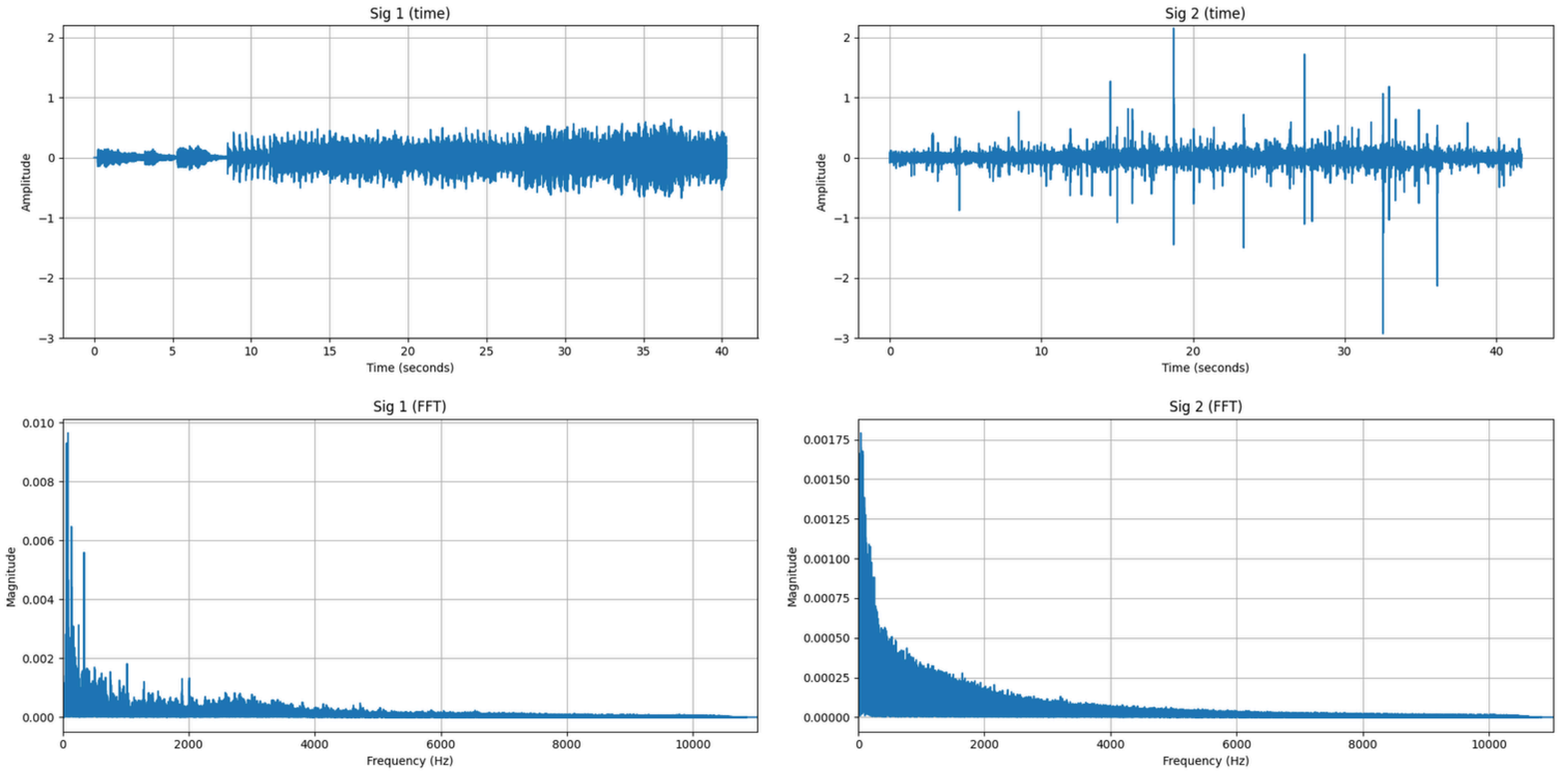
Tonverbesserung für alte Schallplatten

Erster Ansatz:

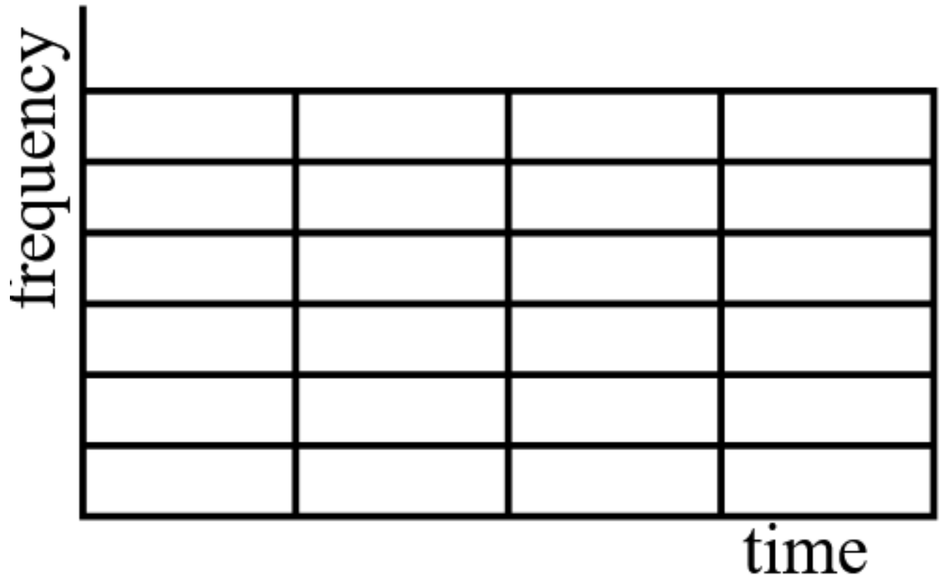
Threshold für Frequenzen festlegen

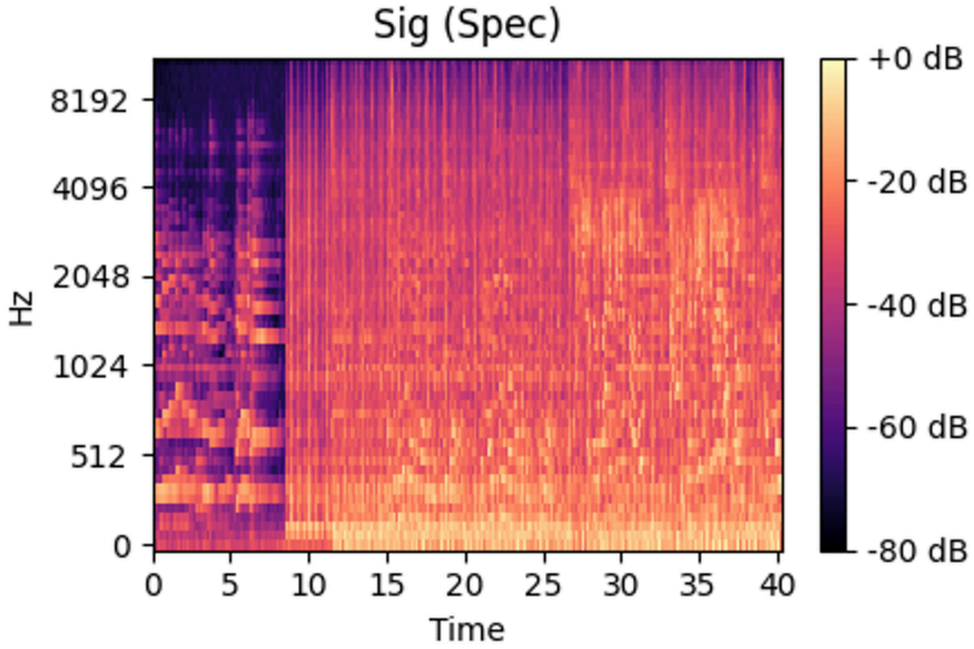
→ kleiner als Threshold ist Rauschen



Specral Gating

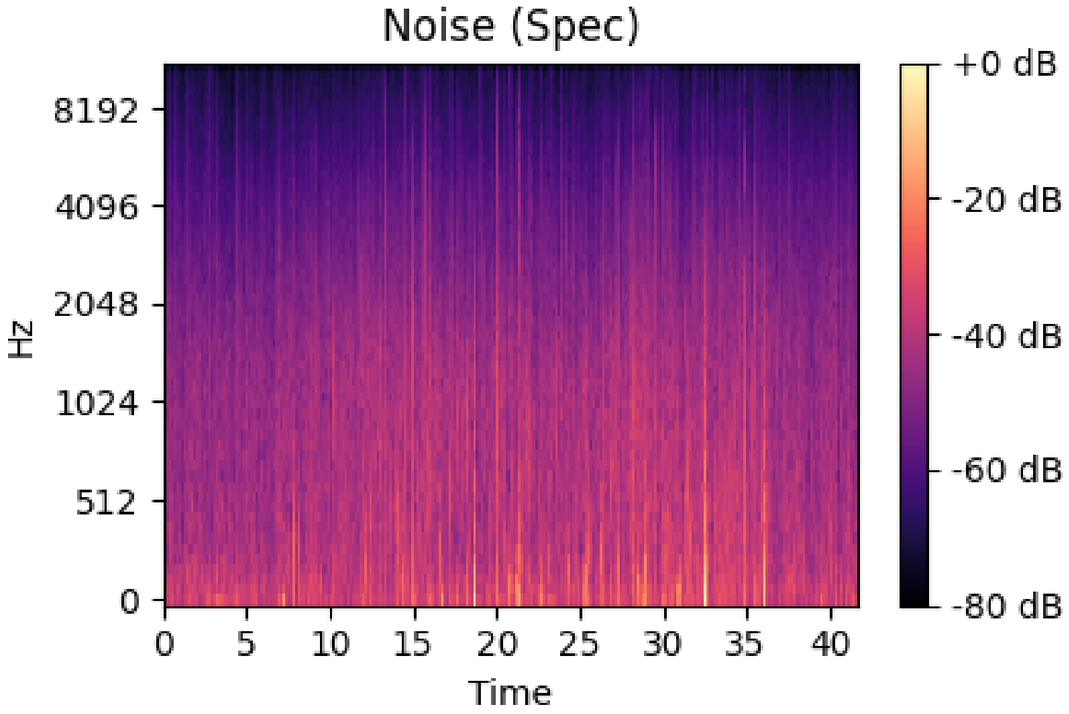
Verwendung von **STFT**
(Short-Time Fourier Transform)

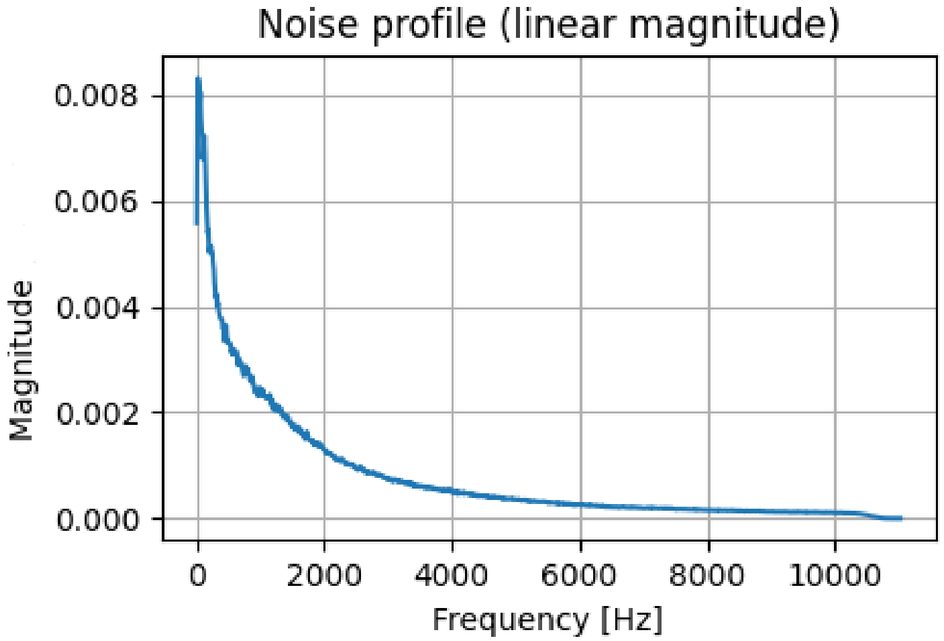




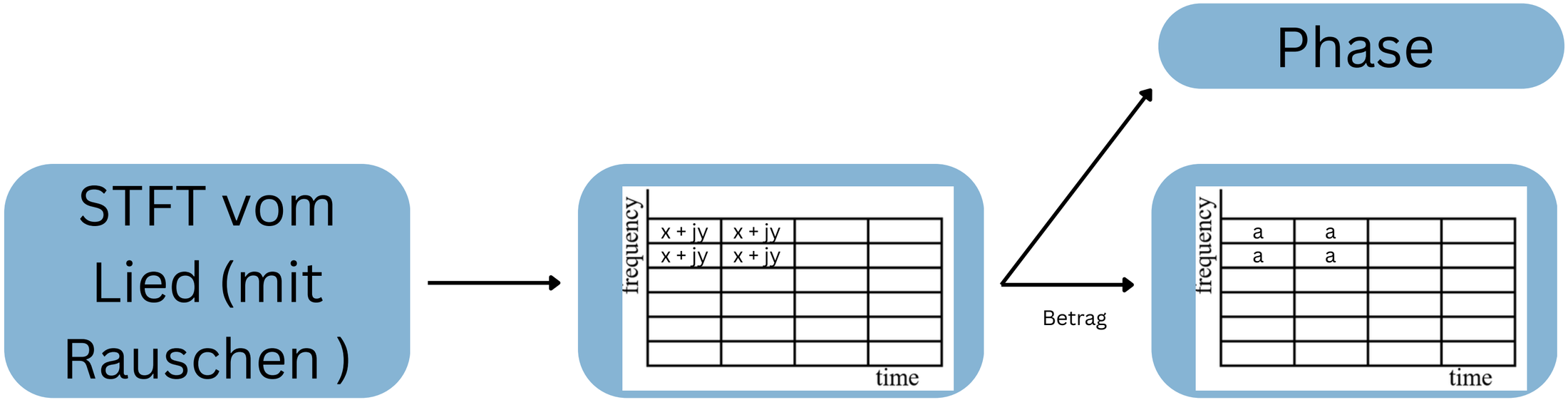
Specral Gating

Erstellen einer Maske des Rauschens mit Quantil





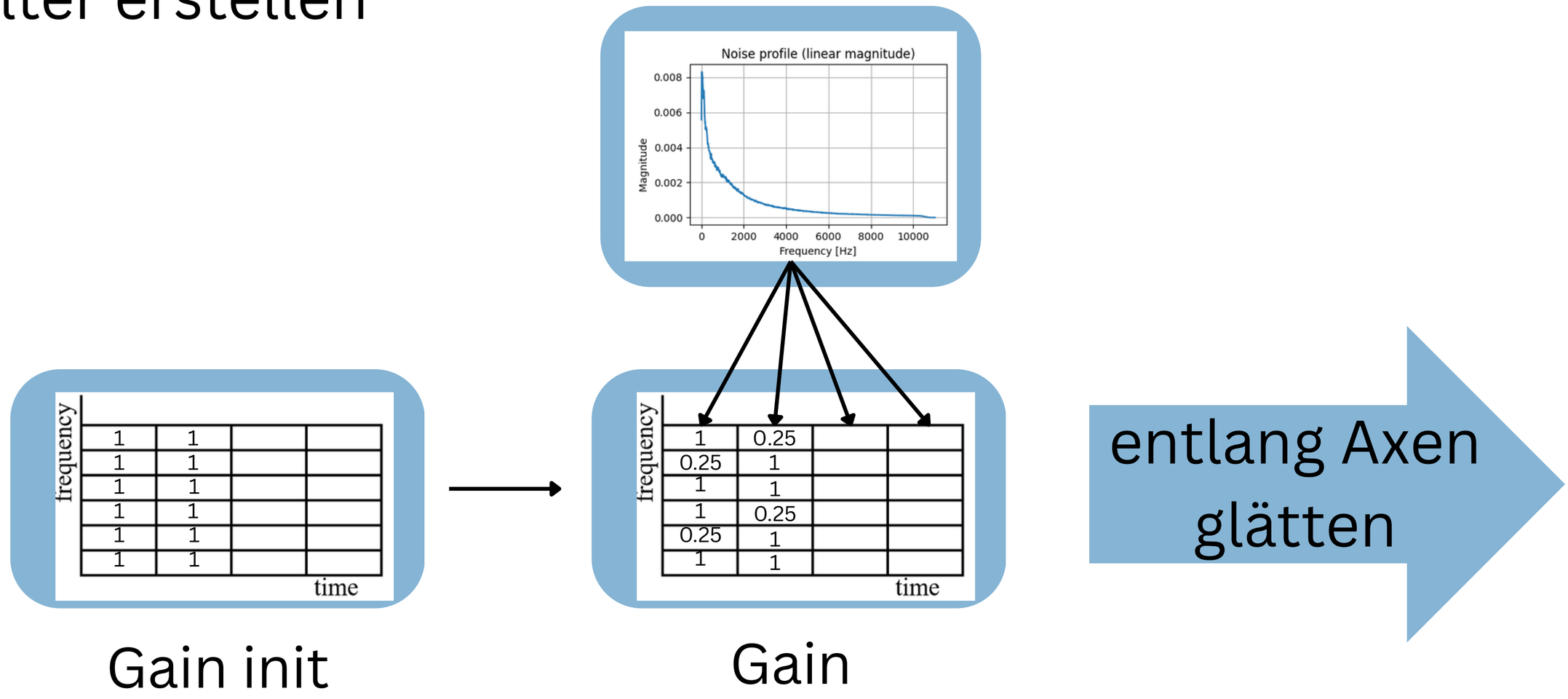
Specral Gating



Mag:
“Stärke” der einzelnen freq über die Zeit

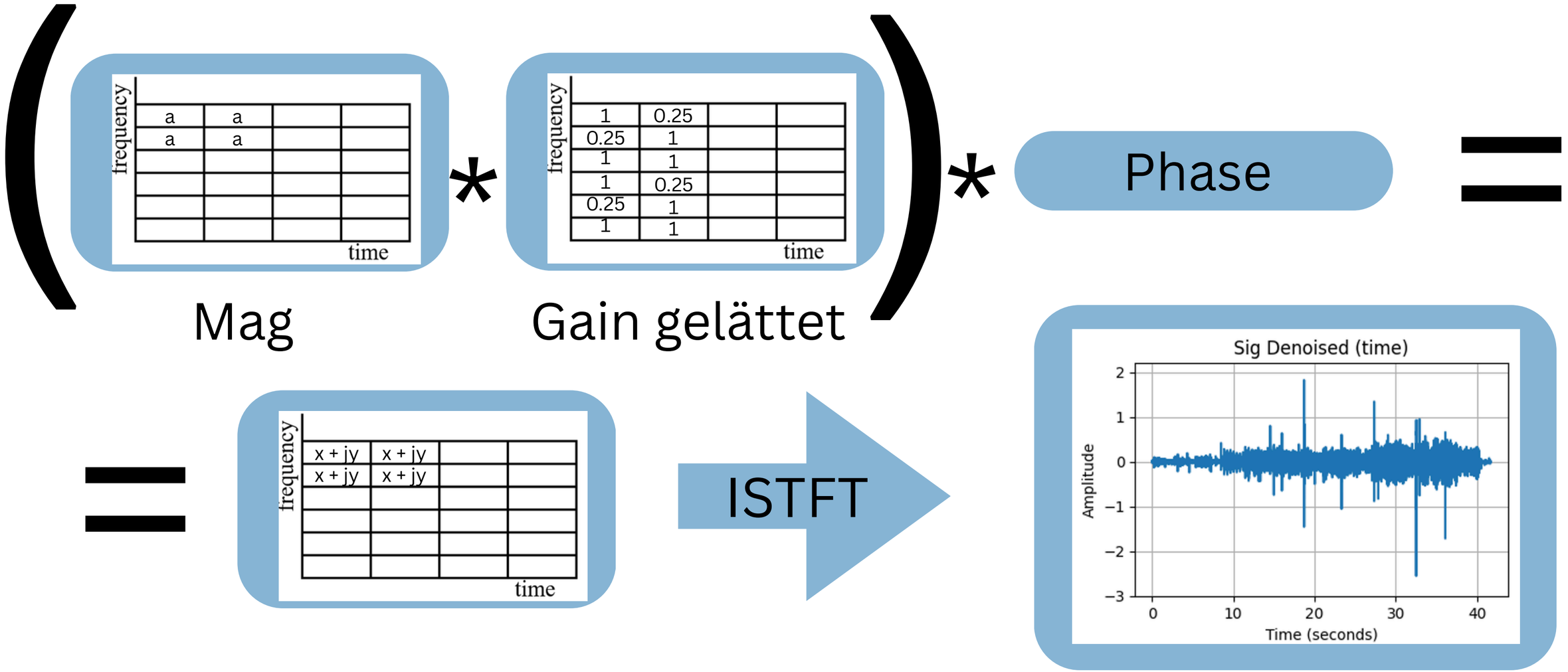
Specral Gating

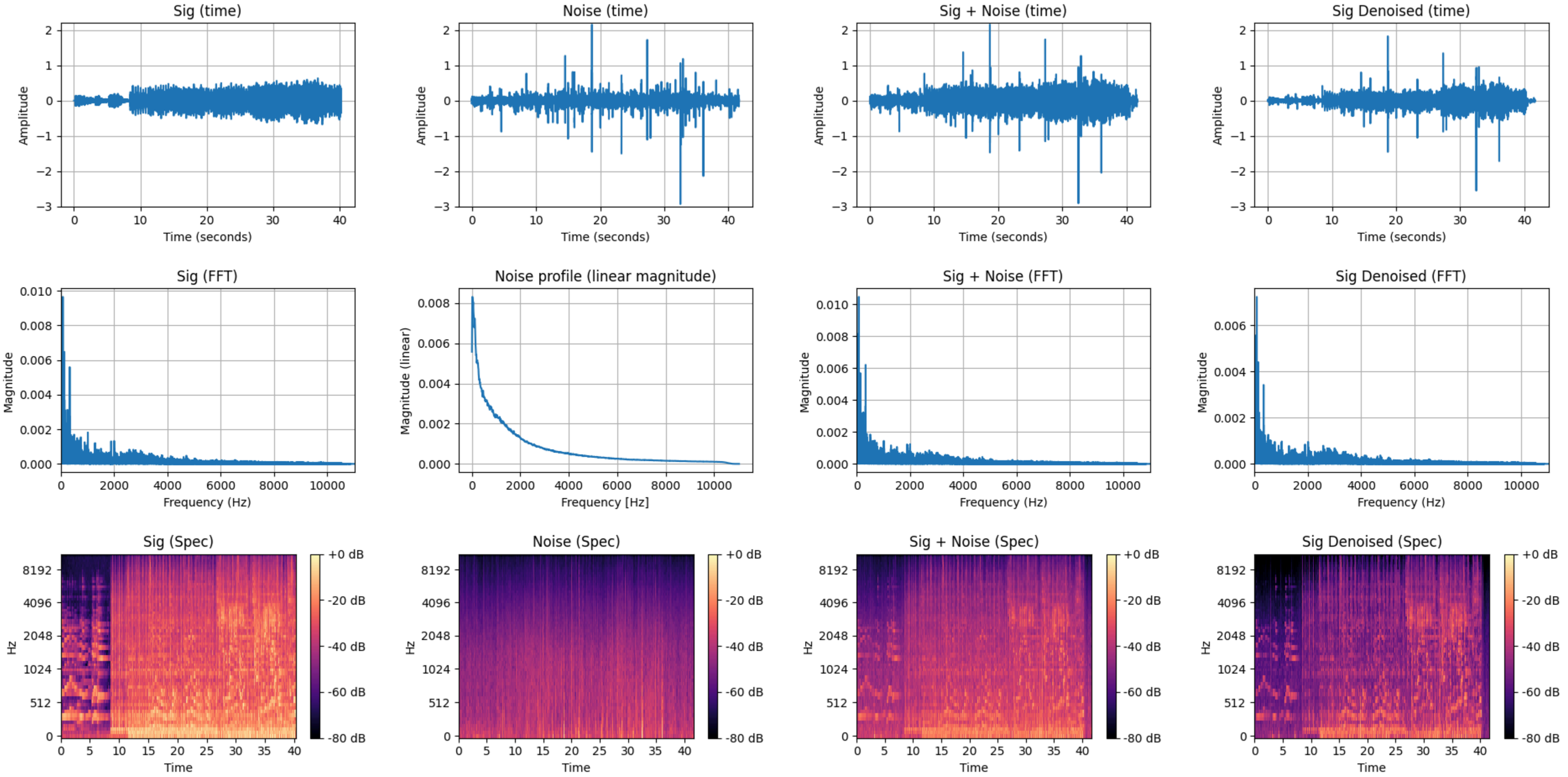
Filter erstellen



Specral Gating

Rauschen “entfernen”




Sig (time)
Noise (time)
Sig + Noise (time)
Sig Denoised (time)
Amplitude
Time (seconds)
Sig (FFT)
Noise profile (linear magnitude)
Sig + Noise (FFT)
Sig Denoised (FFT)
Magnitude
Magnitude (linear)
Frequency (Hz)
Frequency [Hz]
Sig (Spec)
Noise (Spec)
Sig + Noise (Spec)
Sig Denoised (Spec)
Hz
Time
+0 dB
-20 dB
-40 dB
-60 dB
-80 dB

Graphical user interface - Gui

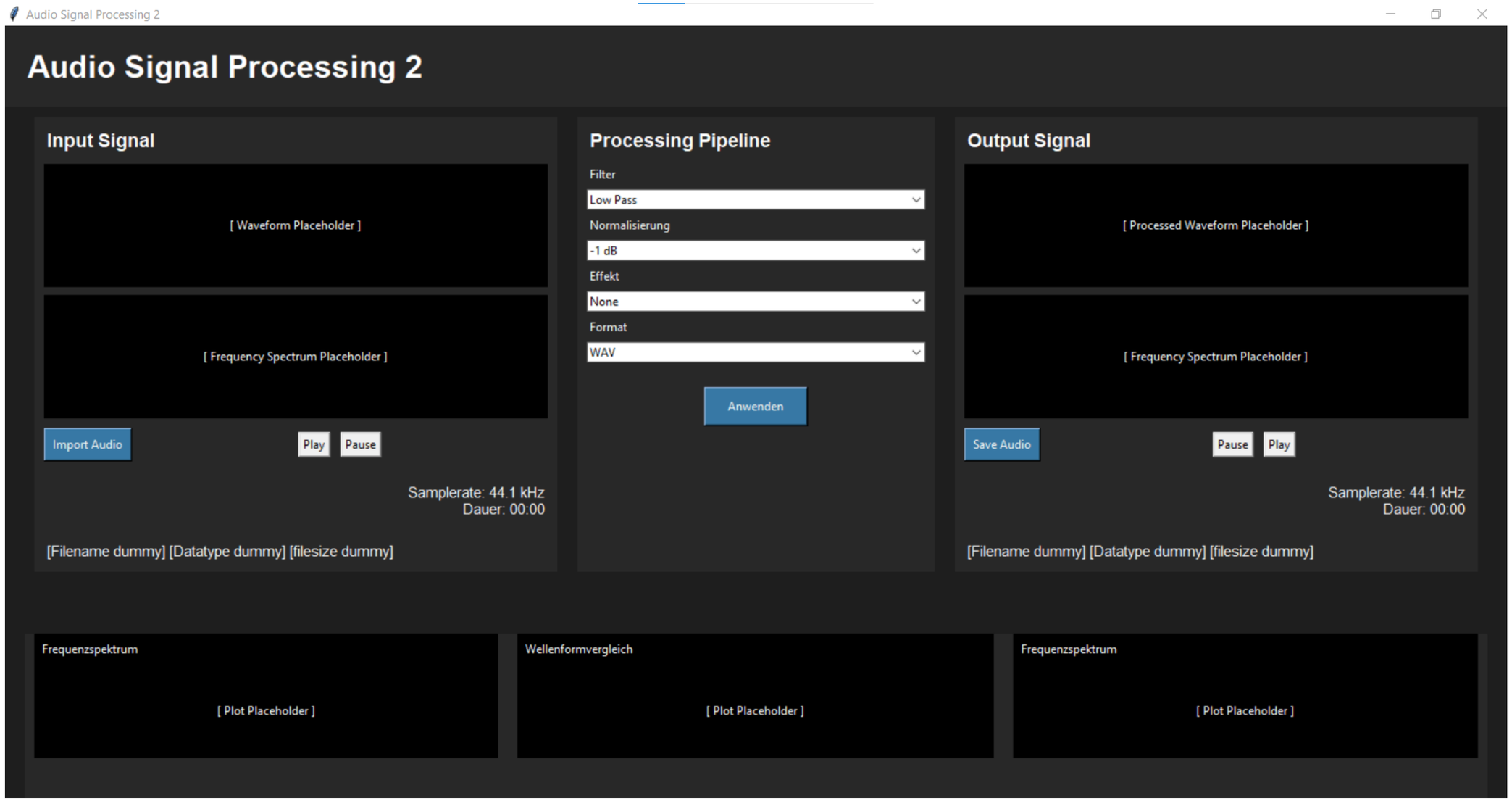
- Grafische Oberfläche
- Schnittstelle Benutzer-Logikprogramm

- Ziel:
  - Intuitiv / selbsterklärend
  - Übersichtlich
  - Unterhaltend

Graphical user interface - Gui

- Erstellt mit Tkinter (Stadard Gui Bibliothek von Python)
- Kleine bis mittlere Desktopanwendungen
- Grundprinzip:
  - Frames werden erstellt
  - Widgets werden in den Frames plaziert und mit Events kombiniert

Graphical user interface - Gui



Graphical user interface - Gui

```
src_gui/
├── main_gui.py
│
├── window.py
├── header.py
│
├── input_view.py
├── analysis_view.py
├── output_view.py
│
├── pipeline_view.py
│
├── config.py
│
├── __init__.py
└── __pycache__/
```

- In der main_gui wird das Hauptframe erstellt + Aufruf der Unterframes
- In der config wird das allgemeine Erscheinungsbild definiert

Ausblick

- Diagramme einbetten
- Schnittstellen zur Logik verknüpfen
- Weitere Widgets ergänzen (Scrollbar, etc.,)