Kaiser Resampling

May 17, 2021

[1]:
```
import librosa
import torch, torchaudio

import numpy as np
import matplotlib.pyplot as plt
```

```
################################################################################
### WARNING, path does not exist: KALDI_ROOT=/mnt/matylda5/iveselyk/Tools/kaldi-
trunk
###          (please add 'export KALDI_ROOT=<your_path>' in your $HOME/.profile)
###          (or run as: KALDI_ROOT=<your_path> python <your_script>.py)
################################################################################
```

[2]:
```
P = 48000
Q = 44100

offset = 2000
instfreq = np.exp(np.linspace(np.log(offset+100), np.log(offset+23900),␣
 ↪96000))-offset
deltaphase = 2*np.pi*instfreq/P
cphase = np.cumsum(deltaphase)
sig = np.sin(cphase)
sig_torch = torch.from_numpy(sig)
```

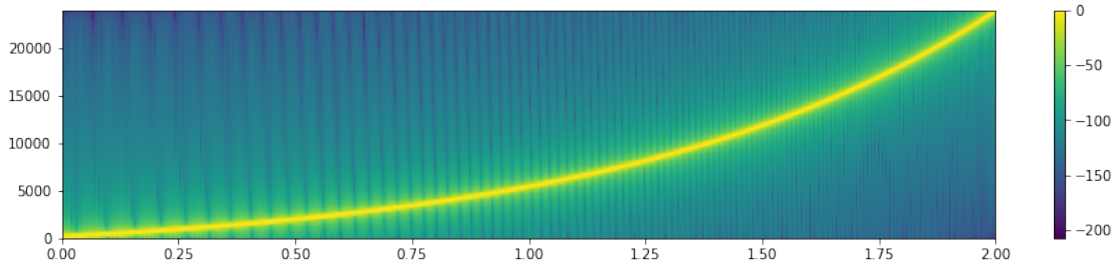
[3]:
```
zeros = 64
kaiser_best_beta = 14.769656459379492
kaiser_best_rolloff = 0.9475937167399596
```

[4]:
```
torchaudio_kaiser = torchaudio.functional.resample(torch.from_numpy(sig), P, Q,␣
 ↪lowpass_filter_width=zeros,
                                                    rolloff=kaiser_best_rolloff,␣
 ↪resampling_method="kaiser_window",
                                                    beta=kaiser_best_beta)
librosa_kaiser = librosa.resample(np.float64(sig), P, Q)
torch.max(torchaudio_kaiser - librosa_kaiser)
```

```
/Users/carolinechen/Documents/audio/torchaudio/functional/functional.py:1368:
UserWarning: To copy construct from a tensor, it is recommended to use
sourceTensor.clone().detach() or
sourceTensor.clone().detach().requires_grad_(True), rather than
torch.tensor(sourceTensor).
  beta = torch.tensor(beta, dtype=float)
```

[4]:
```
tensor(0.0236, dtype=torch.float64)
```

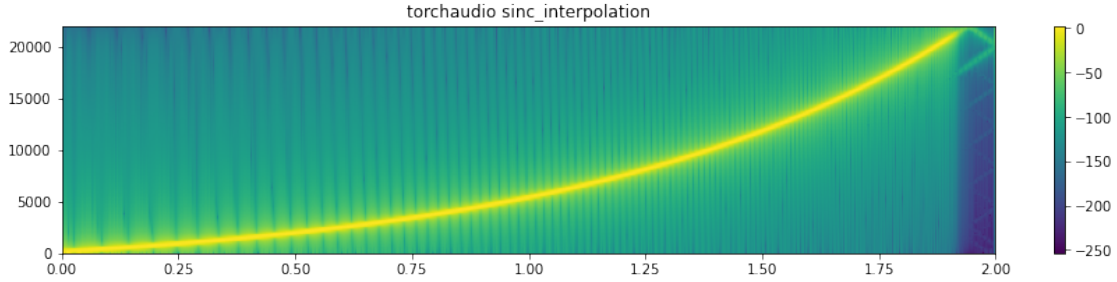
[5]:
```
plt.figure(figsize=(15,3))
plt.specgram(sig*22, scale='dB', Fs=P, NFFT=256)
plt.colorbar()
_=plt.axis((0,2,0,P/2))
```



0.0.1 torchaudio sinc interpolation

[6]:
```
# 64 zero-crossings
plt.figure(figsize=(15,3))
plt.specgram(torchaudio.functional.resample(torch.from_numpy(sig), P, Q,␣
 ↪lowpass_filter_width=zeros, rolloff=kaiser_best_rolloff)*30,
             scale='dB', Fs=Q, NFFT=256)
plt.colorbar()
_=plt.axis((0,2,0,Q/2))
plt.title("torchaudio sinc_interpolation")
```

[6]:
```
Text(0.5, 1.0, 'torchaudio sinc_interpolation')
```

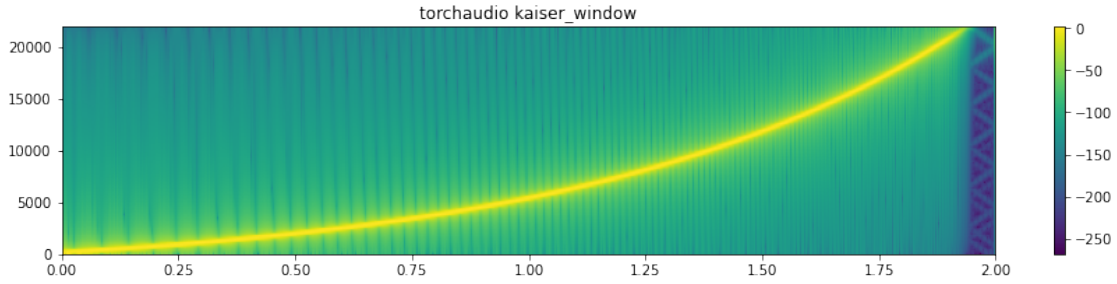


0.0.2 torchaudio kaiser_best

[7]:
```
plt.figure(figsize=(15,3))
plt.specgram(torchaudio.functional.resample(torch.from_numpy(sig),
                                            P, Q, lowpass_filter_width=zeros,
                                            rolloff=kaiser_best_rolloff,
                                            resampling_method="kaiser_window",
                                            beta=kaiser_best_beta)*30,
             scale='dB', Fs=Q, NFFT=256)
plt.colorbar()
_=plt.axis((0,2,0,Q/2))
plt.title("torchaudio kaiser_window")
```

```
/Users/carolinechen/Documents/audio/torchaudio/functional/functional.py:1368:
UserWarning: To copy construct from a tensor, it is recommended to use
sourceTensor.clone().detach() or
sourceTensor.clone().detach().requires_grad_(True), rather than
torch.tensor(sourceTensor).
  beta = torch.tensor(beta, dtype=float)
```

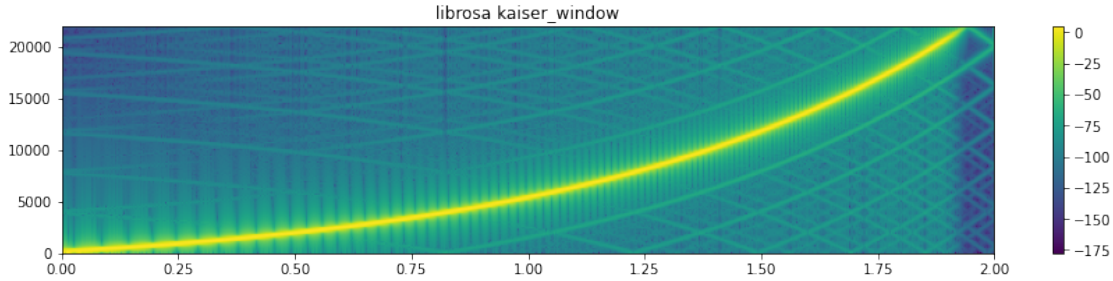
[7]:
```
Text(0.5, 1.0, 'torchaudio kaiser_window')
```



0.0.3 librosa kaiser_best

[8]:
```
plt.figure(figsize=(15,3))
plt.specgram(librosa.resample(np.float64(sig), P, Q, filter="kaiser_best")*40,␣
 ↪scale='dB', Fs=Q, NFFT=256)
plt.colorbar()
_=plt.axis((0,2,0,Q/2))
plt.title("librosa kaiser_window")
```

[8]:
```
Text(0.5, 1.0, 'librosa kaiser_window')
```



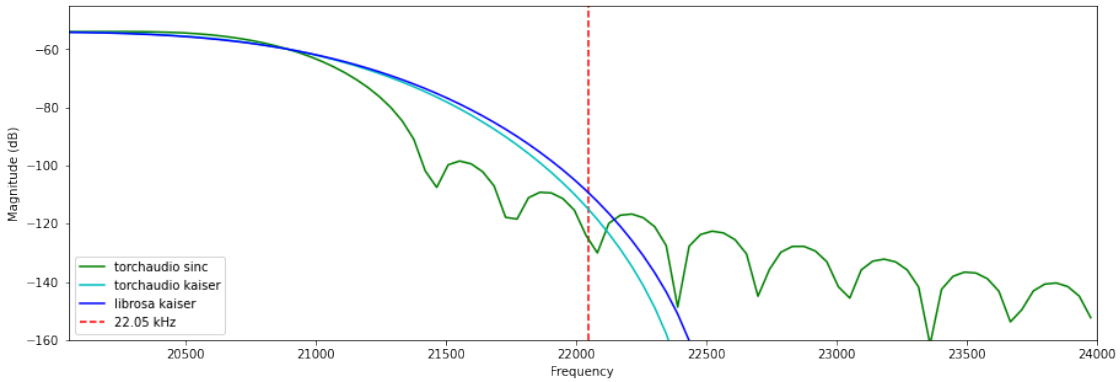
0.1 Upsampling an impulse

[9]:
```
P = 48000
Q = 44100

impulse = np.zeros(1000)
impulse[499] = 1

torch_sinc = torchaudio.functional.resample(torch.from_numpy(impulse),
                                            Q, P, lowpass_filter_width=zeros,
                                            rolloff=kaiser_best_rolloff).numpy()
torch_kaiser = torchaudio.functional.resample(torch.from_numpy(impulse),
                                              Q, P, lowpass_filter_width=zeros,
                                              rolloff=kaiser_best_rolloff,
                                              resampling_method="kaiser_window",
                                              beta=kaiser_best_beta).numpy()
librosa_kaiser = librosa.resample(impulse, Q, P)

# align to torch_sinc
torch_kaiser = np.pad(torch_kaiser,(torch_sinc.argmax() - torch_kaiser.argmax(),␣
 ↪0))
librosa_kaiser = np.pad(librosa_kaiser,(torch_sinc.argmax() - librosa_kaiser.
 ↪argmax(), 0))

print(torch_sinc.shape, torch_kaiser.shape, librosa_kaiser.shape)
```

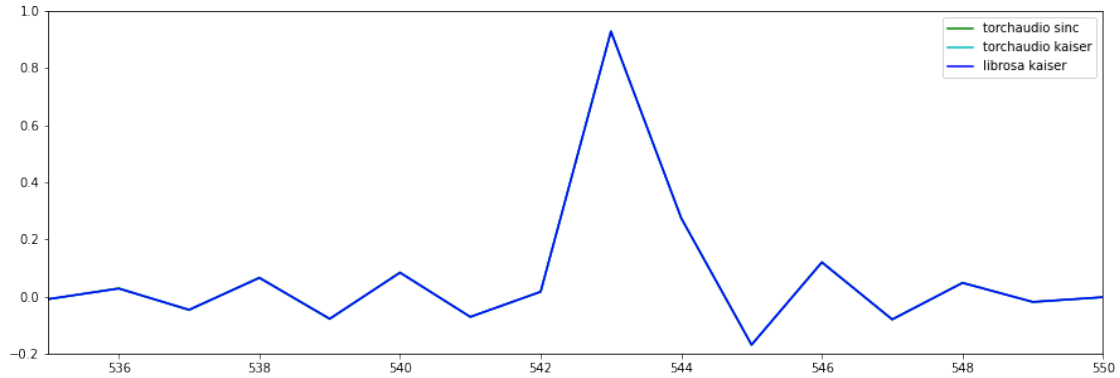
```
(1089,) (1089,) (1089,)
```

[10]:
```
plt.figure(figsize=(15,5))
plt.magnitude_spectrum(torch_sinc, c='g', Fs=P, scale='dB', label='torchaudio␣
 ↪sinc')
plt.magnitude_spectrum(torch_kaiser, c='c', Fs=P, scale='dB', label='torchaudio␣
 ↪kaiser')
plt.magnitude_spectrum(librosa_kaiser, c='b', Fs=P, scale='dB', label='librosa␣
 ↪kaiser')
```

```
plt.axis((Q/2 - 2000, P/2, -160, -45))
_=plt.vlines(Q/2, -160, 3, colors='r', linestyles='dashed', label='22.05 kHz')
_=plt.legend(loc='lower left')
```



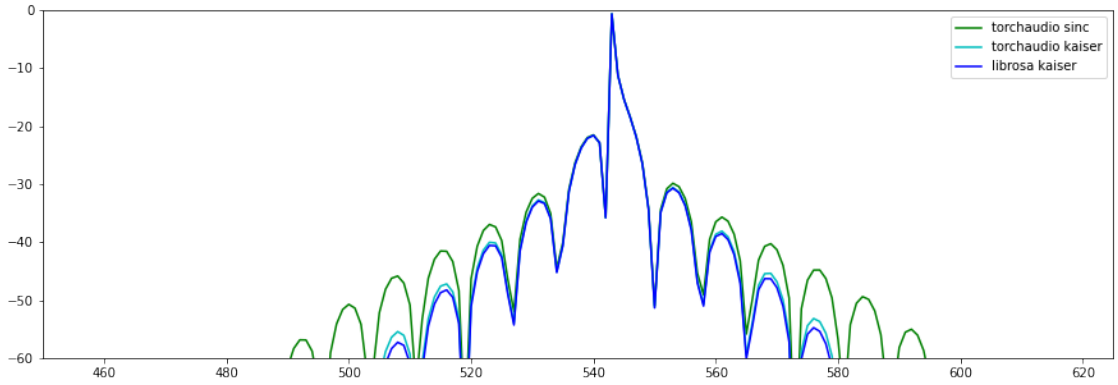
[11]:
```
plt.figure(figsize=(15,5))
plt.plot(torch_sinc, c='g', label='torchaudio sinc')
plt.plot(torch_kaiser, c='c', label='torchaudio kaiser')
plt.plot(librosa_kaiser, c='b', label='librosa kaiser')
plt.axis((535, 550, -0.2, 1))
_=plt.legend()
```



[12]:
```
plt.figure(figsize=(15,5))
plt.plot(10*np.log10(np.square(torch_sinc)), c='g', label='torchaudio sinc')
plt.plot(10*np.log10(np.square(torch_kaiser)), c='c', label='torchaudio kaiser')
plt.plot(10*np.log10(np.square(librosa_kaiser)), c='b', label='librosa kaiser')
plt.axis((450, 625, -60, 0))
_=plt.legend()
```

```
/Users/carolinechen/opt/anaconda3/envs/audio/lib/python3.7/site-
packages/ipykernel_launcher.py:2: RuntimeWarning: divide by zero encountered in
```

```
log10

/Users/carolinechen/opt/anaconda3/envs/audio/lib/python3.7/site-
packages/ipykernel_launcher.py:3: RuntimeWarning: divide by zero encountered in
log10
  This is separate from the ipykernel package so we can avoid doing imports
until
/Users/carolinechen/opt/anaconda3/envs/audio/lib/python3.7/site-
packages/ipykernel_launcher.py:4: RuntimeWarning: divide by zero encountered in
log10
  after removing the cwd from sys.path.
```



[13]:
```
%timeit torchaudio.functional.resample(sig_torch, P, Q,␣
↪lowpass_filter_width=zeros, rolloff=kaiser_best_rolloff)
```

```
11.6 ms ± 601 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

[14]:
```
%timeit torchaudio.functional.resample(sig_torch, P, Q,␣
↪lowpass_filter_width=zeros, rolloff=kaiser_best_rolloff,␣
↪resampling_method="kaiser_window", beta=kaiser_best_beta)
```

```
/Users/carolinechen/Documents/audio/torchaudio/functional/functional.py:1368:
UserWarning: To copy construct from a tensor, it is recommended to use
sourceTensor.clone().detach() or
sourceTensor.clone().detach().requires_grad_(True), rather than
torch.tensor(sourceTensor).
  beta = torch.tensor(beta, dtype=float)

15.9 ms ± 599 µs per loop (mean ± std. dev. of 7 runs, 100 loops each)
```

[15]:
```
%timeit librosa.resample(np.float64(sig), P, Q)
```

```
57.6 ms ± 1.75 ms per loop (mean ± std. dev. of 7 runs, 10 loops each)
```

[16]:
```
resampler = torchaudio.transforms.Resample(P, Q, lowpass_filter_width=zeros,␣
 ↪rolloff=kaiser_best_rolloff, resampling_method="kaiser_window",␣
 ↪beta=kaiser_best_beta)
%timeit resampler(sig_torch)
```

```
1.79 ms ± 116 µs per loop (mean ± std. dev. of 7 runs, 1000 loops each)
```

[ ]:
```

```