Contents

1 Exercise 6

We want to to reproduce the tonotopie of a humean cochla by using filters.

Implement a filterbank with 3, 6, 12 and 22 channels and show how the recorded signal is beeing influenced by the different amount of channels. As the electrodes of a Cochlear Implant are evently spaced along the the cochlear, we will calculate the center frequencies of the filters using the Greenwood function. The bandwidth of the filter is calculated using Equivalent Rectangular Bandwidth (ERB)

a) Implment the Filterbank using Butterworth IIR filter and plot the frequency response for the filterbanks with 3 and 22 Channels

b) Record a arbitrary word using your soundcard or download a wav file from the internet and pass the signal through your filterbank. Plot the resulting signals for a filterbank with 6 and 12 channels.

c) reassemble the signal after filtering (by summation). Plot the long term spectrum as well as the spectrogram for the original and for the filterd signal. Do so for both a filterbank with 3 and one with 12 channels.

1.1 Recording a signal and processing

```
% Recording a signal
%% Record a audio signal using the microphone port of your soundcard
AufnahmeZeit=2; % / s        Length of the recorded signal
fs=48000;       % / Hz  frequency of the signal
bit=16;         % (bit)       Resolution in bits

% Generate a audio object
recObj = audiorecorder(fs, bit, 1);
get(recObj)
% start the recording
disp(
'Start speaking.'
)
pause(2);
```

```
% two seconds time to take a breath
recordblocking(recObj, AufnahmeZeit);
disp(
'End of Recording.'
);
% Play the recorded signal
play(recObj);
% Save the data in a double-precision array
myRecording = getaudiodata(recObj,
'double'
);
Filter Bank:
% Greenwood Parameters (1990)
      EarQ = 7.23824;
minBW = 22.8509;
order = 1;
% Calculate the ERB and center frequencies
% numChannels: Number of channels

loFreq: Niedrigste Frequenz (zb. 100 Hz)-> Erster Filter

hiFreq: Höchste Frequenz (zb. 8000 Hz)  -> Letzter Filter
ERBlo = ((loFreq/EarQ)^order + minBW^order) ^ (1/order);
ERBhi = ((hiFreq/EarQ)^order + minBW^order) ^ (1/order);
overlap = (ERBhi/ERBlo)^(1/(numChannels-1));

Equivalent Rectangular Bandwidth
ERB = ERBlo * (overlap.^(0:numChannels-1));

Mittenfrequenzen
cf = EarQ*(((ERB.^order) - (minBW.^order)).^(1/order));
  .
  .
  .

% Create the bandpass filters
ftype = 'bandpass';
f_Ny=fs/2;

% Nyquist frequency
```

```
.
.

for
 i=1:numChannels
% Order of the filter
% N_l_Cf : Lower cut off frequency (has to me normalized to the nyquist freq.)
[b,a]=butter(2,[N_l_Cf(i) N_u_Cf(i)],ftype);

% Filter the signal
      Filterausgang(:,i)=filter(b,a,x);
% Frequency responses and group delays
[H(:,i),w]        = freqz     (b, a, n_ff, fs);
      [Gr(:,i), w_gd]   = grpdelay  (b, a, n_ff, fs);
end
```

1.2 Plots

```
% Plot the amplitudespectrum
% FFT of the signal
    n = floor(length(x)/2);
%FFT normalization
    FFTtmp = fft(x');
    FFT = abs(FFTtmp(:,1:n+1)./n);
% Real Part and scaling
    f=(0:n)/n*fs/2;
semilogx(f, FFT)
% Plot with logarithmic axis

axis([50 10000 min(FFT) max(FFT)]);
xlabel('\fontsize{16}Frequency (Hz)');
ylabel('\fontsize{16}\fontsize{16}p^{2} [pa^{2}]')
grid on

% Plot the frequency response in KHz and dB
plot((w/1e3),20*log10(abs(H)),
'LineWidth'
,2.0)
% Plot the spectrogram
% fs: Sampling RAte
```

```
% Lenght of the Hamming window in samples
N_Fenster=10e-3*fs;
% Lenght of the window overlap in samples
      Ueberlapp=5e-3*fs;
% Lenth of the return values B and f
      n_out=1024;
% Calculate the spectrogram for the signal
[B,f,t]=specgram(signal, n_ou, fs, N_Fenster, Ueberlapp);

bmin= max(max(abs(B))) *10^-(60/20);
imagesc(t, f, 20*log10( max(abs(B), bmin)/bmin ) );
```