Table of Contents


```
% Modulation Spectrogram using Gammatone Filterbank
%
% Author: Raymond Xia (yangyanx@andrew.cmu.edu)
%
% This script implements Kingsbury's description of the modulation
% spectrogram [1]. It requires Dan Ellis' Gammatone-like spectrogram
% package [2], specifically his rewrite of Malcolm Slaney's ERB
 filterbank
% routines.
%
% References
% [1]: Kingsbury, Brian ED, Nelson Morgan, and Steven Greenberg.
%      "Robust speech recognition using the modulation spectrogram."
%      Speech communication 25.1 (1998): 117-132.
% [2]: https://labrosa.ee.columbia.edu/matlab/gammatonegram/


TWOPI = 2*pi;
```

Global parameters

```
AUDIO_PATH = 'resources/heli_and_boat_short/heli6_short.wav';
%AUDIO_PATH = 'resources/welcome16k.wav';
SR = 16000; % sampling rat e
LONG_TERM_AVG = 1; % long-term average across time for subband
 envelope
% Gammatone filterbank parameters
N = 128; % filterbank channel number
FMIN = 0; % lowest frequency
NFFT = 2048;
% Bandpass filter parameters; Hamming window is used.
F_CENTER = 5; % center frequency in Hz
F_BW = 10; % frequency bandwidth
% Decimate?
DECIMATE = 1;
DECIMATE_FACTOR = 100; % each channel has bandwidth of approximately
 20Hz,
                       % so 100 is a safe number
```

Demo Parameters

```
DEMO_AUDIO = 1; % Put 1 to analyze audio from AUDIO_PATH, else use AM
 signal
DEMO_WGN = 0; % add white gaussian noise to signal

DEMO_CHAN = 1; % display channel number
DEMO_TSTART = 0; % time display start in seconds
DEMO_TEND = 1; % time display end in seconds
DEMO_FSTART = 0; % frequency display start in Hz
DEMO_FEND = 50; % frequency display end in Hz
```

Process input audio

```
if DEMO_AUDIO
    [x,fs] = audioread(AUDIO_PATH);
else
    % experiment AM signal
    fs = 16000;
    n =(0:2*fs-1)';
    x = cos(TWOPI*(5/fs)*n).*cos(TWOPI*(200/fs)*n);
end

if size(x,2) ~= 1 % mono processing only
    x = mean(x,2);
end
if fs ~= SR
    x = resample(x,SR,fs);
end

if DEMO_WGN
    SNR = -5; % SNR in dB
    xe = x'*x; % signal energy
    ne = xe/(10^(SNR/10)); % SNR = 10*log10(xe/ne)
    noise = randn(size(x));
    noise = noise / sqrt(noise'*noise) * sqrt(ne);
    x = x + noise;
end
```

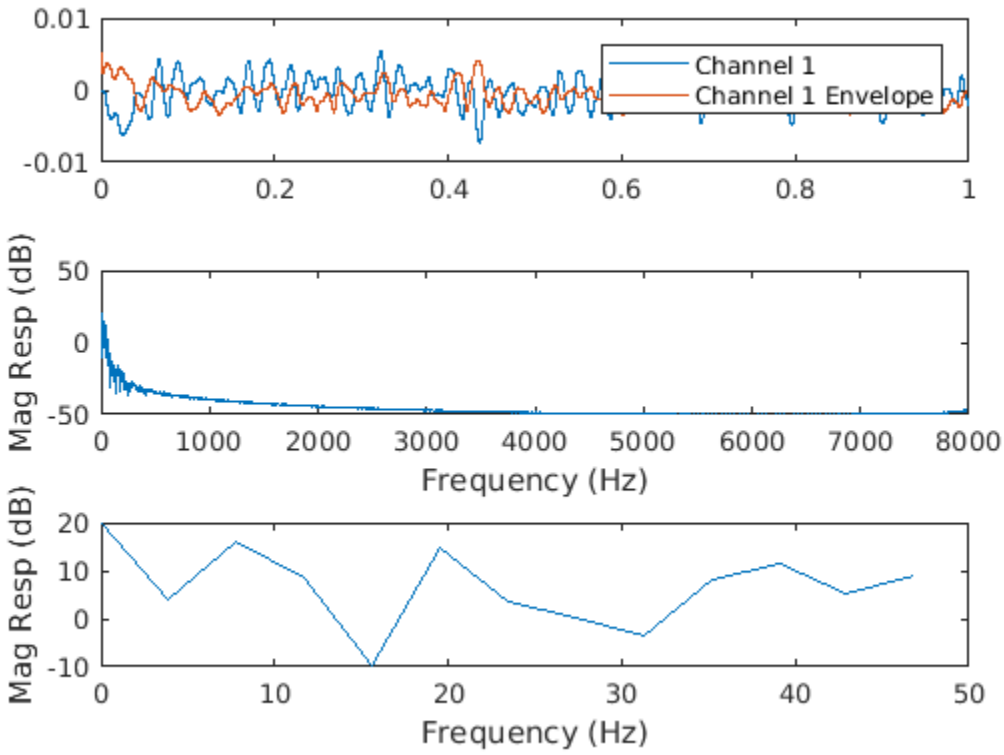
Gammatone filterbank processing

```
[fcoefs,CF] = MakeERBFilters(SR, N, FMIN);
fcoefs = flipud(fcoefs);
CF = CF(end:-1:1);
XF = ERBFilterBank(x,fcoefs);
```

Envelope extraction

```
XF_env = envelope(XF')';
if LONG_TERM_AVG % optionally, take long-term average
    XF_env = bsxfun(@minus,XF_env,mean(XF_env,2));
end
```

```
% display section
figure;
N_START = DEMO_TSTART*SR; N_END = DEMO_TEND*SR;
T = (N_START:N_END)'/SR;
subplot(311); % envelope plot
plot(T,XF(DEMO_CHAN,(N_START:N_END)+1),T,XF_env(DEMO_CHAN,
(N_START:N_END)+1));
chan_str = ['Channel ',num2str(DEMO_CHAN)];
legend(chan_str,[chan_str,' Envelope']);
subplot(312); % magnitude response of envelope
[h,w] = freqz(XF_env(DEMO_CHAN,(N_START:N_END)+1),1,NFFT);
f = w/TWOPI*SR; % convert to Hz
plot(f,20*log10(abs(h)));
xlabel('Frequency (Hz)'); ylabel('Mag Resp (dB)');
ax = gca; ax.YLim = [-50,50];
subplot(313); % zoomed-in magnitude response at low frequency
frange = (f >= DEMO_FSTART) & (f <= DEMO_FEND);
plot(f(frange),20*log10(abs(h(frange))));
ax = gca; ax.YLim = [-10,20];
xlabel('Frequency (Hz)'); ylabel('Mag Resp (dB)');
```

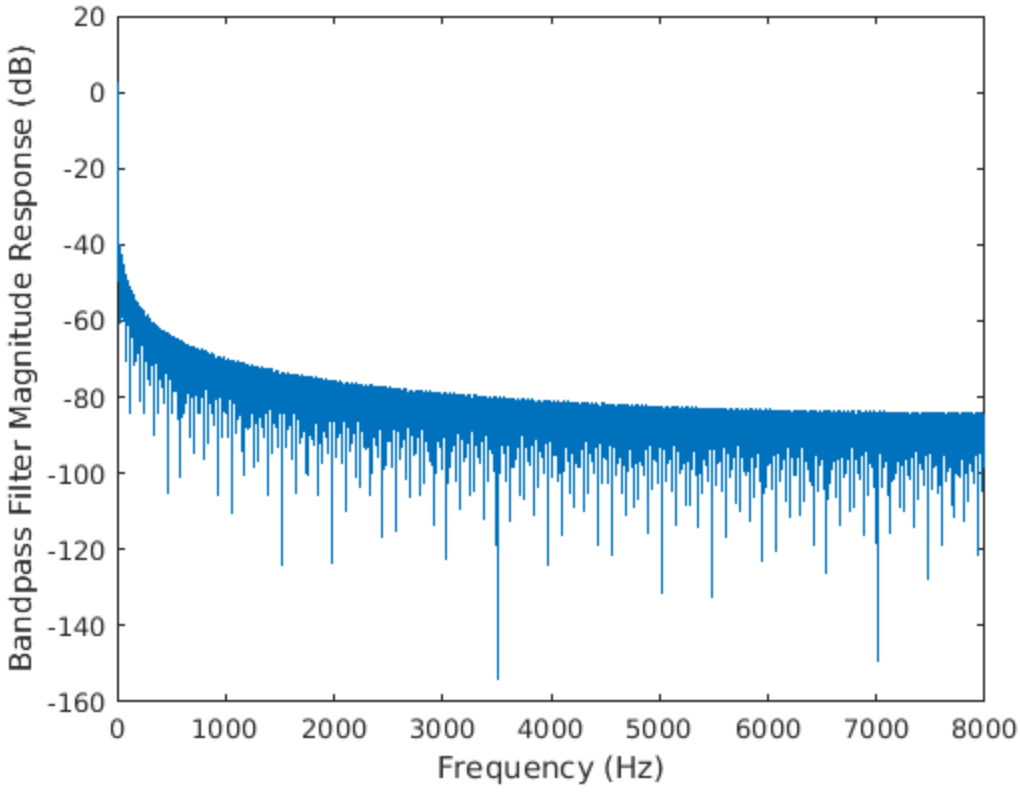


Bandpass filter using modulated hamming window

```
Nw = round(2*(SR/F_BW)+1); % hamming window length
```

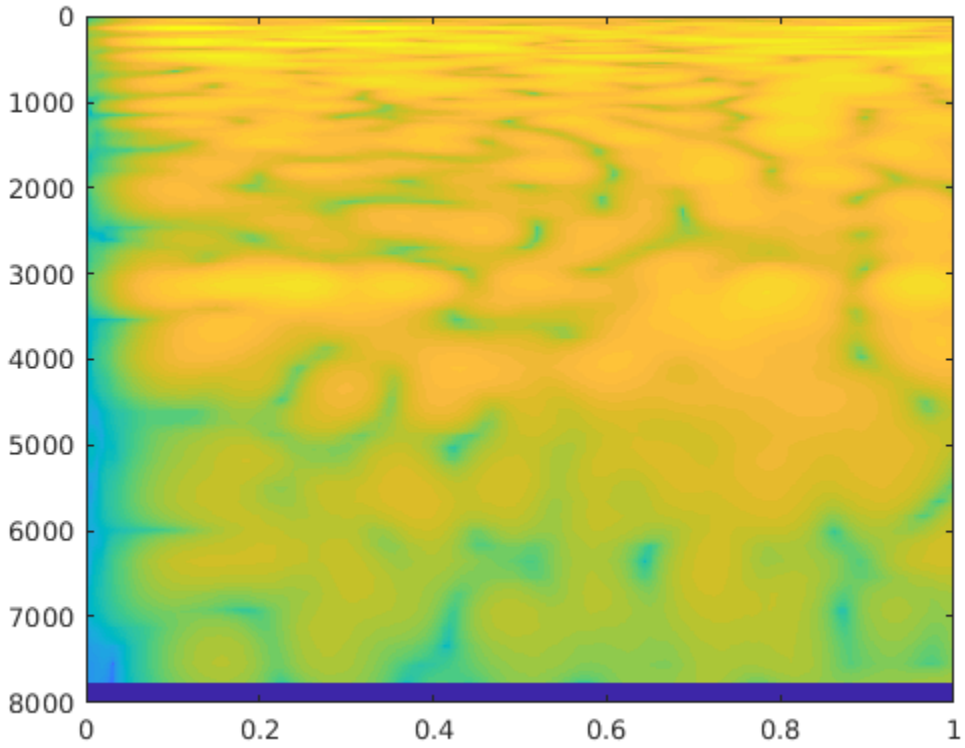
```
w = hamming(Nw); w = w/(w'*w); % normalize energy
w = w .* exp(1j*TWOPI*(F_CENTER/SR)*(0:Nw-1)'); % modulation
[hw,ww] = freqz(w,1,NFFT);
figure;
fw = ww/TWOPI*SR; % convert to Hz
plot(fw,20*log10(abs(hw)));
xlabel('Frequency (Hz)'); ylabel('Bandpass Filter Magnitude Response
 (dB)');

XF_mod = abs(filter(w,1,XF_env.').').^2;
if DECIMATE
    tmp = []; % not efficient for now
    for chan=1:size(XF_mod,1)
        tmp = [tmp;decimate(XF_mod(chan,:),DECIMATE_FACTOR)];
    end
    XF_mod = tmp;
    XF_mod_display = XF_mod(:,((N_START/DECIMATE_FACTOR):(N_END/
DECIMATE_FACTOR))+1);
else
    XF_mod_display = XF_mod(:,(N_START:N_END)+1);
end
```



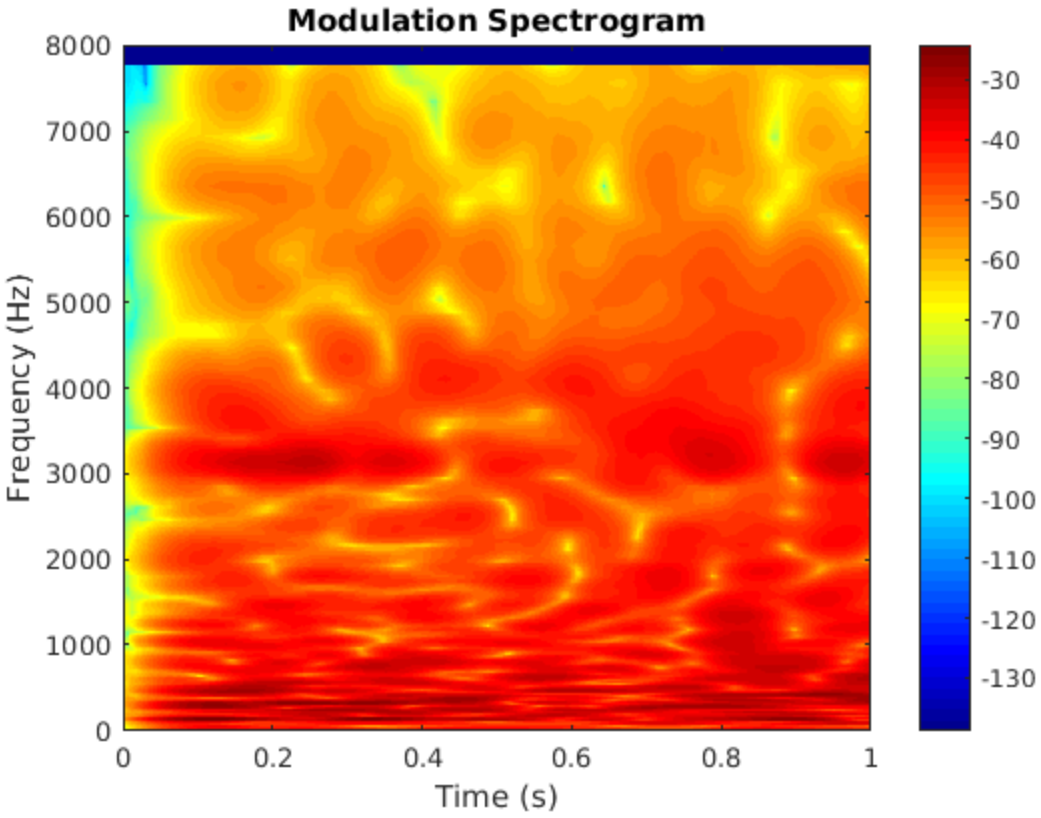
Interpolate and display modulation spectrogram

```
figure;
T_full = (N_START:N_END)/SR;
if DECIMATE
    T_mod = T_full(1:DECIMATE_FACTOR:end);
else
    T_mod = T_full;
end
F_full = linspace(0,SR/2,NFFT/2+1)';
XF_mod_interp = interp2(T_mod,CF,XF_mod_display,T_full,F_full);
imagesc(T_full,F_full,10*log10(abs(XF_mod_interp)));
```



Display output

```
%imagesc(T_full,F_full,abs(XF_mod_interp));
axis xy; colormap(jet);
xlabel('Time (s)'); ylabel('Frequency (Hz)'); colorbar
title('Modulation Spectrogram');
```



Published with MATLAB® R2017a