Table of Contents

Creation of RIRs, application to signals, parameter calculation

```
clear
clc
close all
```

Define parameters

```
room = [10, 10, 10]; % Room dimensions
rt60 = [1]; % Reverberation time
rec = [ 5, 5, 5]; % Receiver positions
fs = 44100;

% source positions (one at +1 meters from source in all 3 dimensions)
src = [6,6,6];
% src = [6, 5, 5;
%     5, 6, 5;
%     5, 5, 6];
rec_orders = 1; % First order ambisonics
```

Generate RIRs

```
TEST_SCRIPT_SH;
```

```
Room dimensions (m)          10x10x10
Room volume (m^3)            1000
Mean absorption coeff        0.27
Sabine Rev. Time 60dB (sec)  1
Critical distance (m)        1.8
Mean free path (m)           6.7
Compute echogram: Source 1 - Receiver 1
Apply SH directivites
Apply absorption: Source 1 - Receiver 1
Rendering echogram: Source 1 - Receiver 1
     Filtering and combining bands
Elapsed time is 0.189291 seconds.
```

Prepare audio

```
tt = (1:1/fs:20)';
src_sigs = [cos(2*pi*1000*tt) cos(2*pi*5000*tt) cos(2*pi*10000*tt)];

if false
    % Load, convert to mono
    [bass, ~] = audioread('bass.wav');
    bass = sum(bass, 2)/2;
    [drums, ~] = audioread('drums.wav');
    drums = sum(drums, 2)/2;
    [other, ~] = audioread('other.wav');
    other = sum(other, 2)/2;
    [vocals, ~] = audioread('vocals.wav');
    vocals = sum(vocals, 2)/2;

    src_sigs = [bass, drums, other, vocals];
end
```

Check out isolated spectrograms

```
%figure
%[s, w, t] = spectrogram(src_sigs, hann(2048), 1024, 2048, fs,
 'yaxis');
%
% imagesc( t, w, log(abs(s)) ); %spectrogram
% set(gca,'YDir', 'normal');
% col = colorbar;
% col.Label.String = 'Power/frequency (dB/Hz)';
% title('Original sound');
% xlabel('Time (seconds)')
% ylabel('Frequency (Hz)')

if false
    figure
    subplot(221)
    spectrogram(bass, hann(2048), 1024, 2048, fs, 'yaxis');
    title('Bass')
    subplot(222)
    spectrogram(drums, hann(2048), 1024, 2048, fs, 'yaxis')
    title('Drums')
    subplot(223)
    spectrogram(other, hann(2048), 1024, 2048, fs, 'yaxis')
    title('Other')
    subplot(224)
    spectrogram(vocals, hann(2048), 1024, 2048, fs, 'yaxis')
    title('Vocals')
end
```

Generate sound scenes

Each source is convolved with the respective mic IR, and summed with the rest of the sources to create the microphone mixed signals

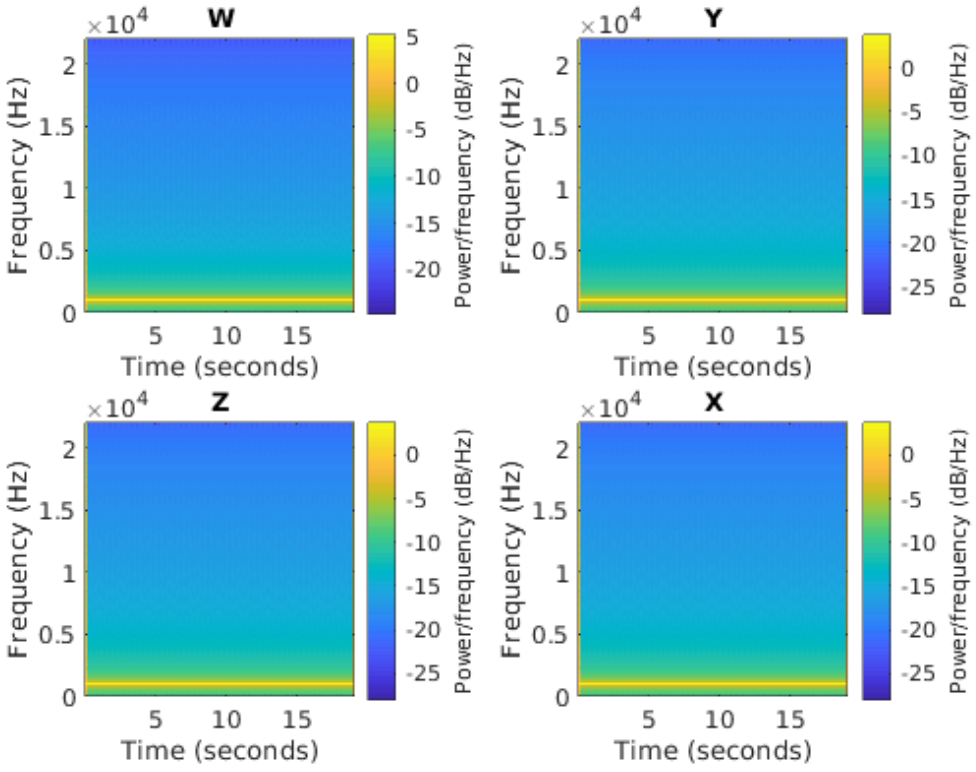
```
% This should be the signals received at each microphone
sh_sigs = apply_source_signals_sh(sh_rirs, src_sigs);

% Normalize
sh_sigs(:, 2:4) = sh_sigs(:, 2:4)/sqrt(3);
```

```
Convolving with source signal: Source 1 - Receiver 1
```

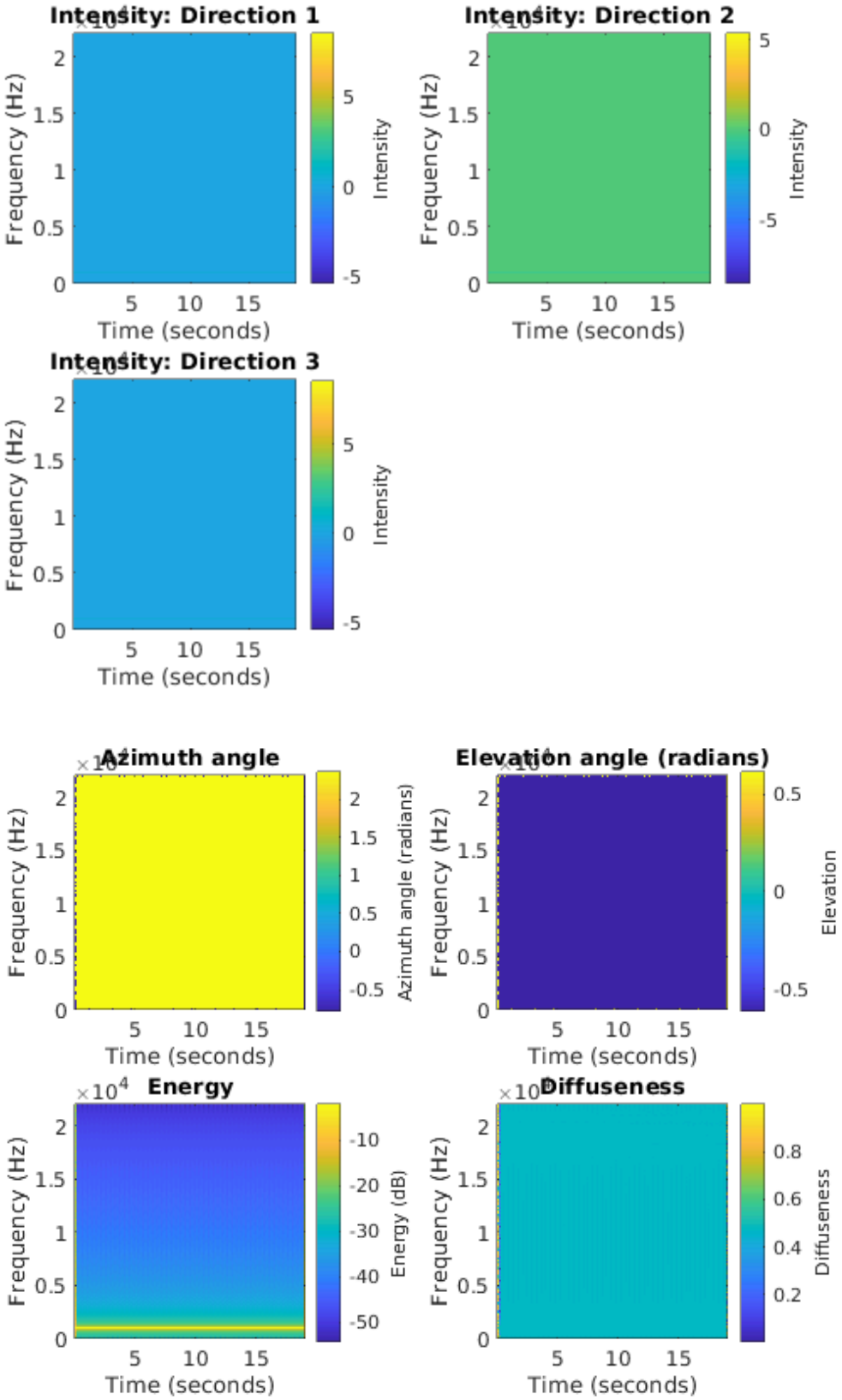
Check out ambisonic spectrograms

```
figure
mix(1).title = 'W';
mix(2).title = 'Y';
mix(3).title = 'Z';
mix(4).title = 'X';
for idx = 1:4
    subplot(2,2,idx)
    % B: spectrogram for each HOA
    [B(idx).spec, w, t] = spectrogram(sh_sigs(:, idx), hann(2048), ...
        1024, 2048, fs, 'yaxis');
    [m,n] = size(B(idx).spec);
    imagesc( t, w, log(abs(B(idx).spec)) ); %spectrogram
    set(gca,'YDir', 'normal');
    col = colorbar;
    col.Label.String = 'Power/frequency (dB/Hz)';
    title(mix(idx).title);
    xlabel('Time (seconds)')
    ylabel('Frequency (Hz)')
end
```



Calculate and plot DirAC params

```
dirAC_calculation;
```


Intensity: Direction 1
Intensity: Direction 2
Intensity: Direction 3
Azimuth angle
Elevation angle (radians)
Energy
Diffuseness
Frequency (Hz)
Time (seconds)
Intensity
Azimuth angle (radians)
Elevation
Energy (dB)
Diffuseness

Published with MATLAB® R2019b