Table of Contents

Creation of RIRs, application to signals, parameter calculation

```
clear
clc
close
```

Define parameters

```
room = [10.2 7.1 3.2]; % Room dimensions
rt60 = [1.0 0.8 0.7 0.6 0.5 0.4]; % Reverberation time
rec = [ 4.5 3.4 1.5]; % Receiver positions
fs = 44100;

src = [ 6.2 2.0 1.8; % source positions
        7.9 3.3 1.75;
        5.8 5.0 1.9;
        2.1 2.5 1.5];
rec_orders = 1; % First order ambisonics
```

Generate RIRs

TEST_SCRIPT_SH;

```
load RIRs&Bs.mat % Much faster than re-running every time
```

Prepare audio

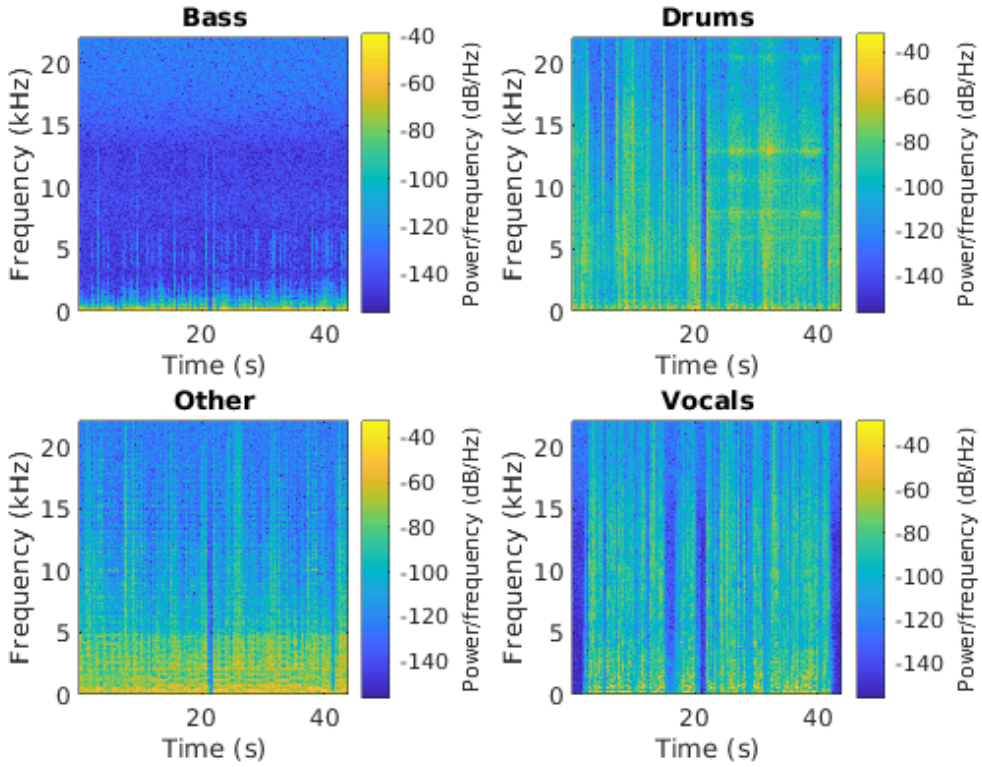
Load, convert to mono

```
[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];
```

Check out isolated spectrograms

```
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')
```



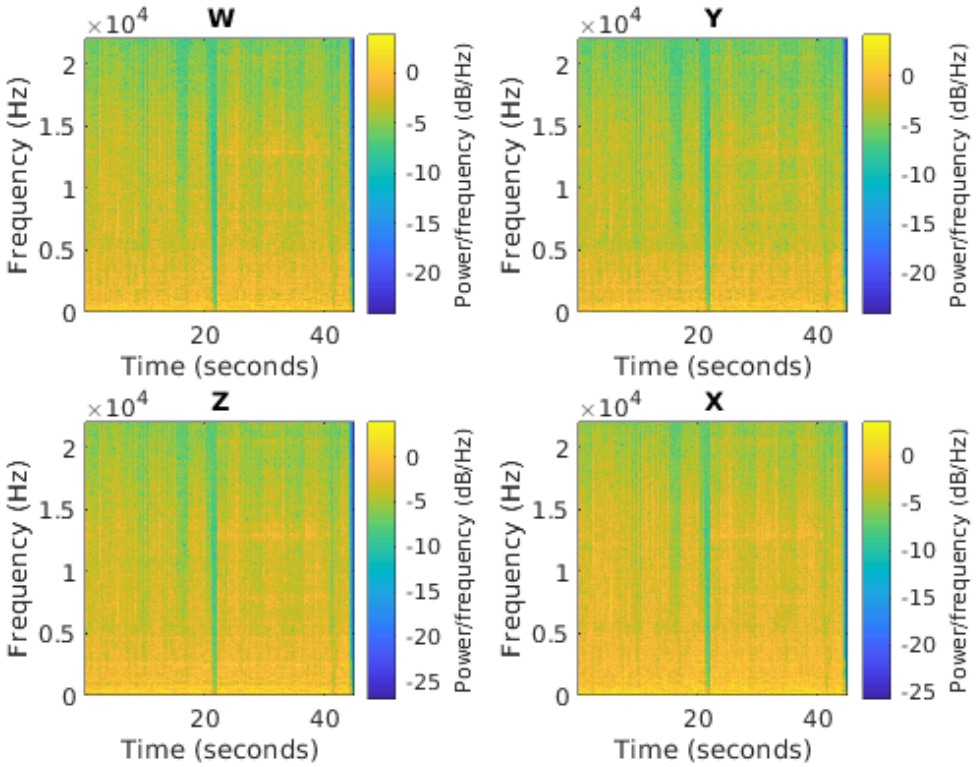
Generate sound scenes

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);
load 'sh_sigs'
```

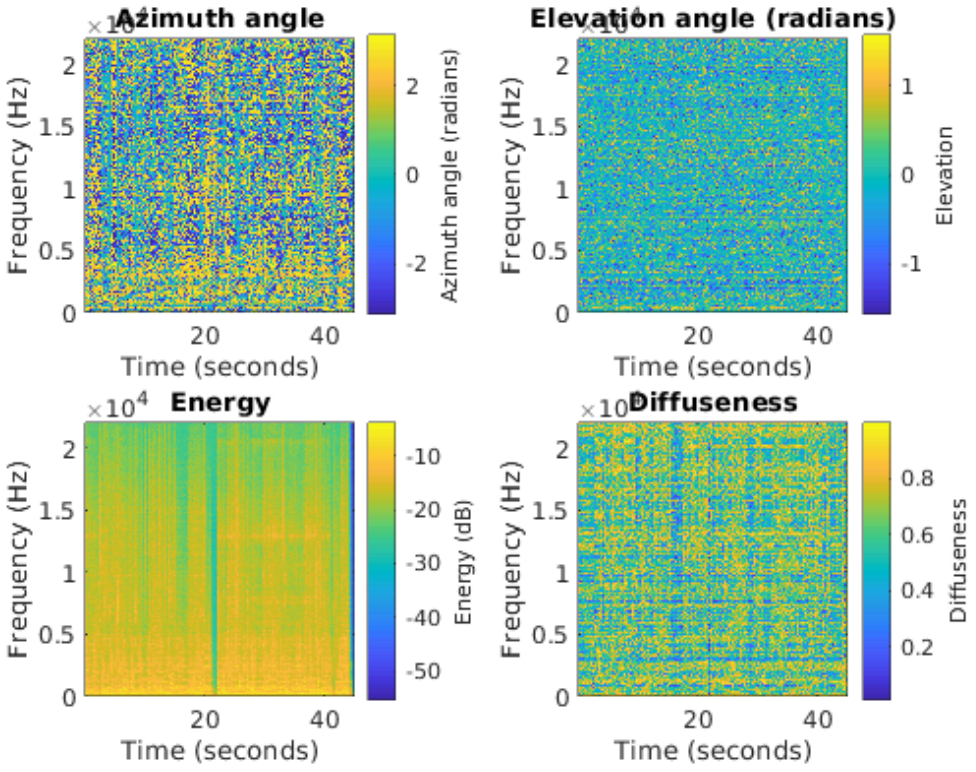
Check out ambisonic spectrograms

```
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;
```



Published with MATLAB® R2019b