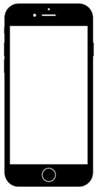
SpeakerPhone

Real-Time Device Emulation

By Jesca Rachelle Chengula, Beau Hanson, Aaron Messina

Abstract

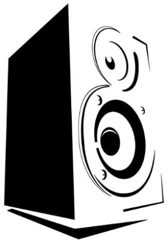
I. **Goal** - create a library of low-quality speaker emulators
II. **Approach** - quantify/define each device's unique sonic features → design a number of signal processing stages
    A. Design of filters
    B. Listening (trial & error)
III. **Processing** - real time (Simulink)
IV. **Result** - processed audio should have similar features to those outputted through the devices



iPhone 5

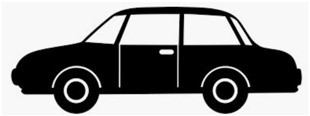
iPhone 7s+

Galaxy S9



Stereo

Subaru Forester



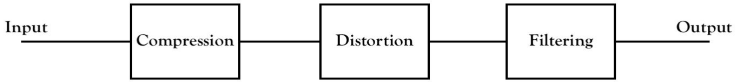
Introduction

Recording:

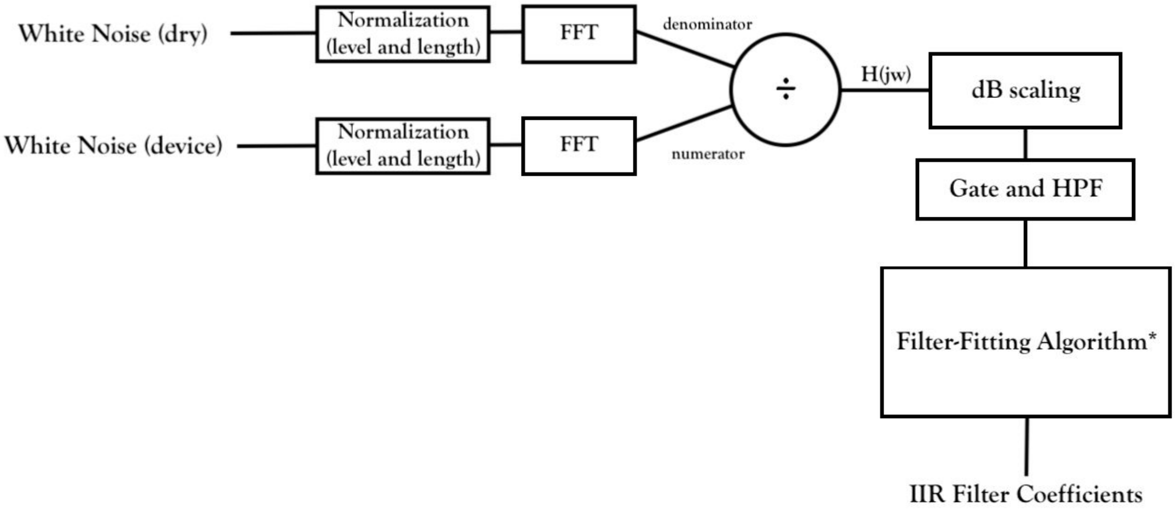
- Use measurement microphones
- Record devices playing white noise, sine sweep, pure sine tones, and music

Three steps for processing:

- Compression
- Distortion
- Filtering



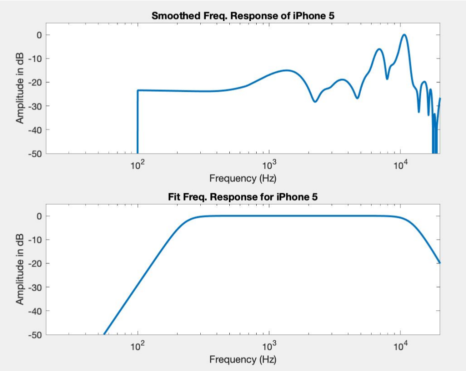
Filtering (Beau)

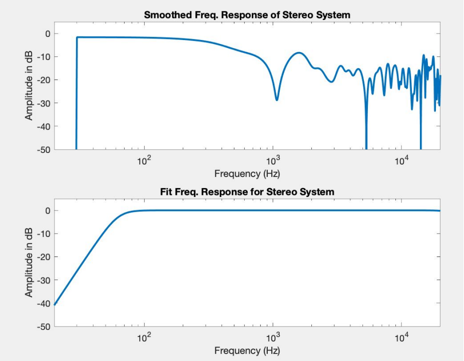


**This algorithm is a triple-nested "for" loop*

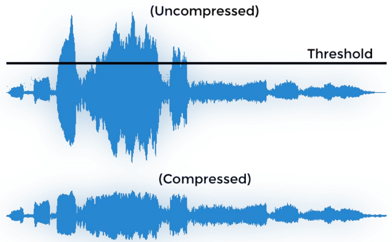
Filtering (Beau)

1. Define device filter with a smoothed version of $H(j\omega) = \frac{Y(j\omega)}{X(j\omega)}$
2. Algorithmically choose IIR filter based on **lowest Mean Square Error.**
   The device's filter is determined given the set of the following variables
   a. Low-Frequency Cutoff $f_c \in [100Hz, 1500Hz]$
   b. High-Frequency Cutoff $f_c \in [3kHz, 18kHz]$
   c. Filter order $n \in [10, 40]$
3. Create and save final filter coefficients for application

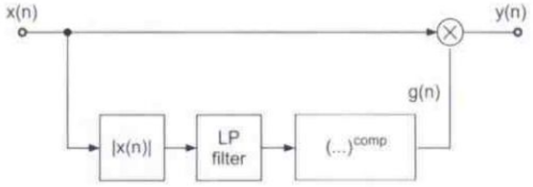




Compression(Jesca)



1. For reduction of the dynamic range of the signal
2. Pass the uncompressed signal over a compressor function
3. Input parameters: Release, attack, compression ratio and the filter parameter
4. Make up gain to boost the compressed signal



$R = \frac{XdB - CT}{YdB - CT}$, R>1 is compression

Harmonic Distortion (Aaron)

Approach 1:

- Use fft to estimate f0 and a few odd harmonics
- Add harmonics directly to frame

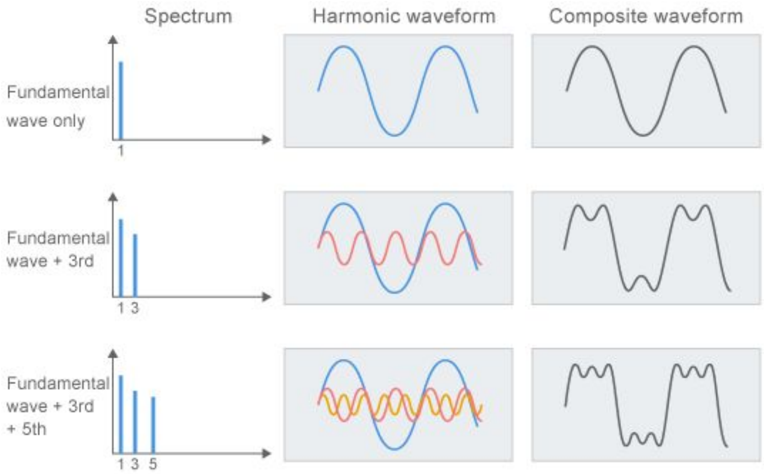
Approach 2:

- Estimate f0 and harmonics (same as App. 1)
- Generate high Q parametric filters boosting the harmonic frequencies
- Apply filters to frame

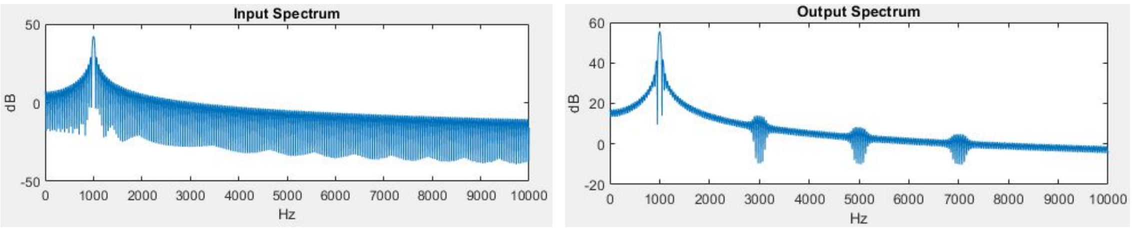
Parameters for varying devices:

- Number of harmonics amplified
- Gain of the parametric filters

Adding Harmonics -- Theory:



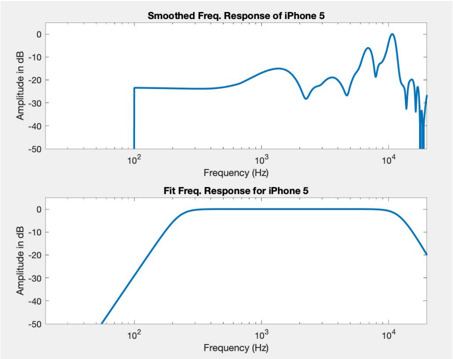
Sample -- 1kHz sine wave:



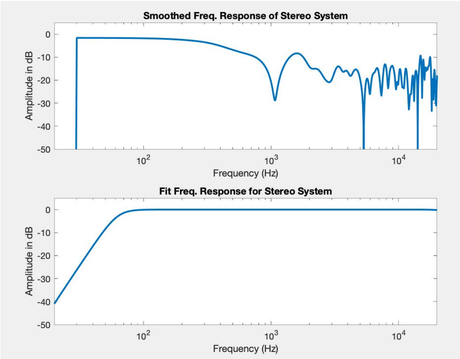
Evaluation (Beau)

- Evaluate FFTs forv MSE of different devices filters
- Can weight frequency cutoff variables for better aural correlation
- Which is better: lower MSE? Or aural similarity test?

Acceptable MSE values:
< 3000



MSE ~2000



MSE ~1600

Examples(Jesca)

Peg through iPhone 5

Peg dry

Peg through Samsung S9

Peg through Subaru Forester

Future Implementations (Aaron)

- Conversion to plug-in format
- Using adjustable parameters that are initially set based on device type
- Past Simulink: using overlap-add to accomplish cleaner signal processing

Questions?