UMassAmherst

Team Turn Up

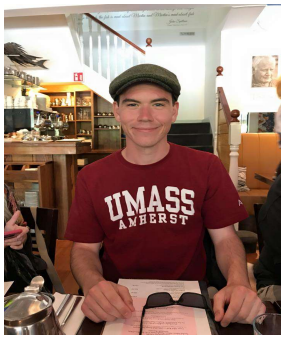
October 16, 2018

Department of Electrical and Computer Engineering

Advisor: Professor Pishro-Nik

Team Members

Nicholas Kafasis
CSE



Harold Healy
EE

Ryan Walsh
CSE

Rahaun Perkins
CSE

Problem Statement

- In environments with dynamic noise levels, frequent volume adjustments for Speakers and TVs are a nuisance. Current speakers and stereo systems need to be manually adjusted.
- **Solution:** Design a dynamic volume controller would automatically regulate the volume of these devices based on a ratio of ambient noise to audio input signals.

Automation in Todays Market

- Todays market is being taken over by automated services and devices
  - Temperature Control
  - Screen Adjustments

Design Alternatives

- TOA Electronics Digital Ambient Noise Controller
- Very Expensive($2300)
- Complicated for normal user
- Meant for commercial environments
- Significant power consumption

Design Alternatives (con't)

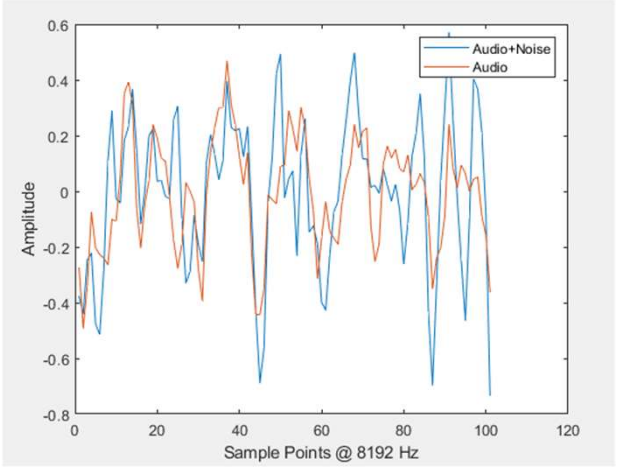
- International Control Systems - Automatic TV Sound Regulator
  - Exclusively for TVs
  - Does not factor in background noise

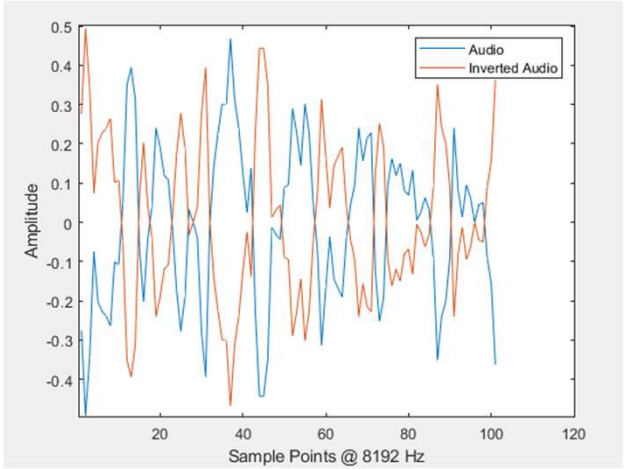
Requirements and System Specifications

- Easy to use phone app will allow user to set fixed level of volume above/below ambient noise.
- System will maintain SNR within a range based on initial calibration
- System will not exceed max volume setting
- System will not react suddenly to isolated loud noises
- System will work within a 20 ft. radius of microphone

Solutions

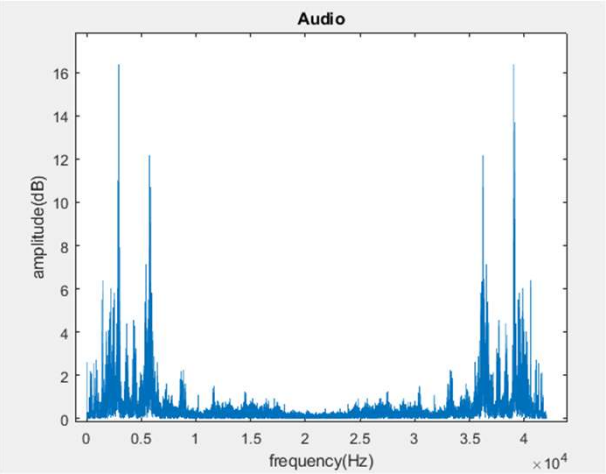
- Noise Cancellation
- Invert known signal to remove sound from speaker (left only with background noise)
- Issues with delay and timing
- Could lose parts of ambient noise

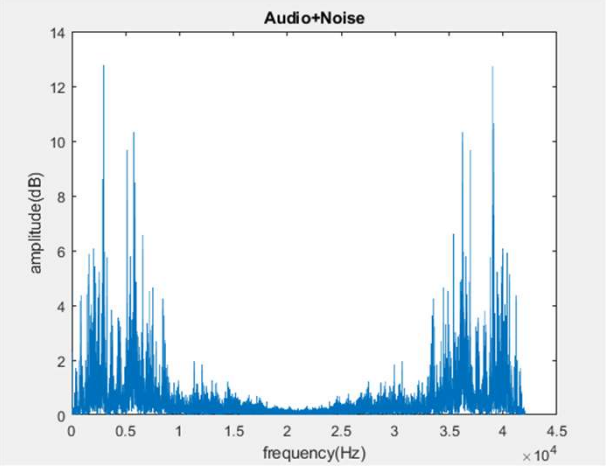




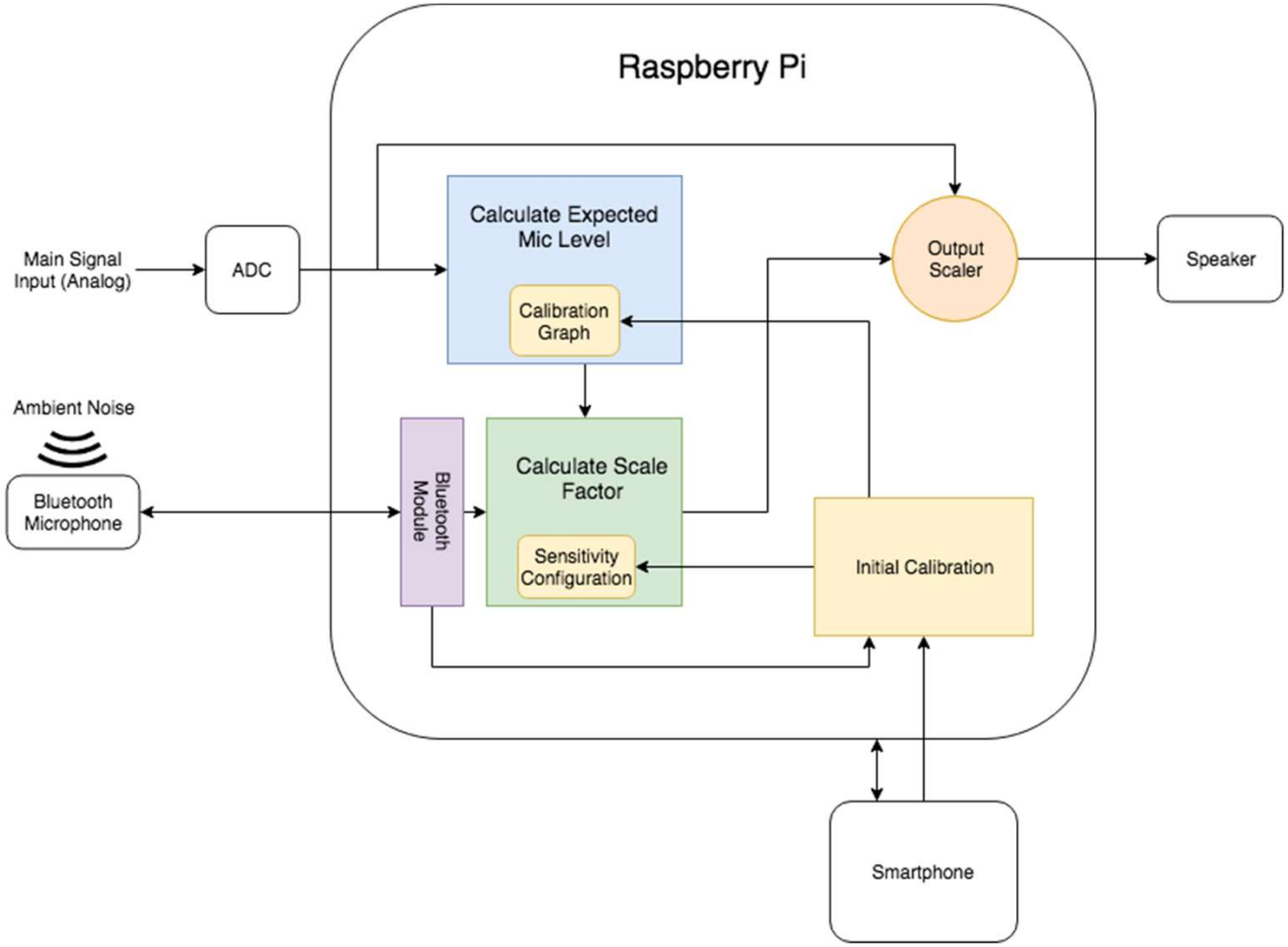
Solutions (con't)

- Frequency Analysis
- Compare normalized spectra of input signal and signal at microphone with added noise
- Issues with calculation time
- Simpler solution available





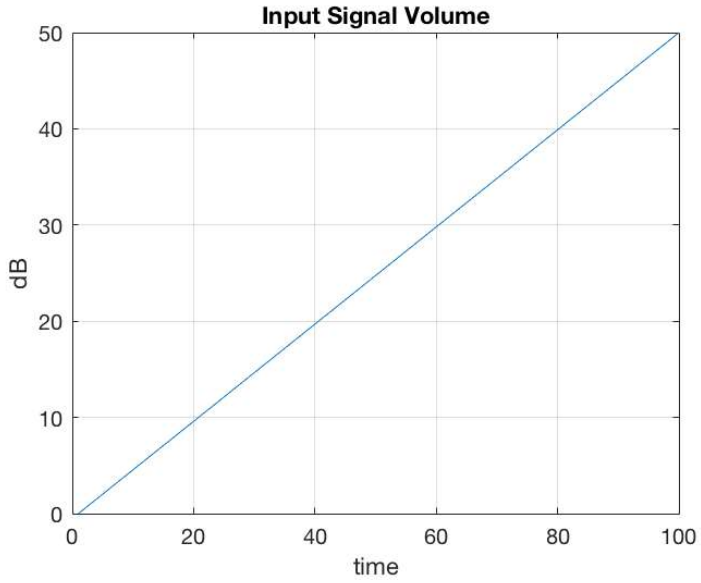
Design System (with Block Diagram)

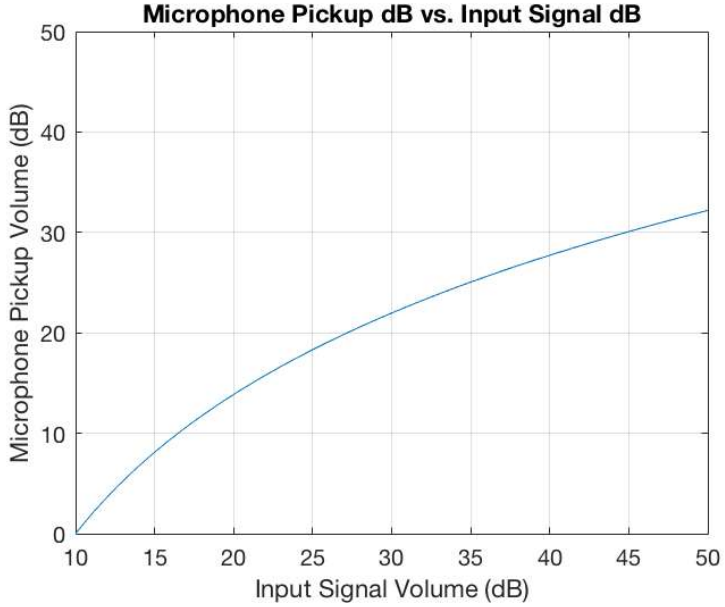


Initial Calibration

- User will be able to configure system with a smartphone app.
- In the app, user will have the following abilities:
  - Modify the system's sensitivity
  - Modify the system's noise tolerance
  - Begin the calibration process
- Calibration Process:
  1. Play a predetermined sound signal from the minimum to maximum volume level.
  2. Microphone will pick up the sound from the speaker, send it to the Pi.
  3. System will record the relationship between the volume of the input signal and the volume of the signal that is picked up by the microphone.
  - Thus, this will give a model of the expected microphone pick-up volume as a function of the input signal volume.
    - Store this model internally as a look-up table for precision

Input Volume to Microphone Pick-up Volume





Noise Tolerance Threshold and Sensitivity

- Noise tolerance: maximum ratio of expected-to-actual microphone pickup volume that system will tolerate.
  - User will select from a set of ratios we define.
  - Will require our analysis to define appropriate ratios to choose from
- Sensitivity: how quickly the system will respond to detected noise
  - User-selected sensitivity setting will factor into determining the noise averaging intervals
    - Scale from 1-10 for user to select from.
  - Requires our analysis and refining to ensure smooth volume adjust

Calculating the Scale Factor

- Take averages of input signal volume and microphone pickup volume over small amounts of time.
- Determine expected pickup volume from input signal volume averages; compare to actual microphone pickup volume averages.
  - Also average the difference of expected and actual pickup volume over time in order to avoid sudden, rapid changes in volume.
  - If average difference rises above the noise tolerance threshold, output a scale factor that will rise volume gradually until the difference comes back into the threshold.
- Continuously work to drive down the volume level to the desired volume level while keeping the ambient noise to signal noise ratio within the calculated threshold.

Team Member Roles

Harold Healy
- PCB Design
- Team Manager

Ryan Walsh
- Calibration Design
- App Development

Nicholas Kafasis
- Core Software Design

Rahaun Perkins
- Core Software Design

Cost Analysis

- Raspberry Pi: $35
- Enclosure: $20
- Microphone w/ Bluetooth: $30
- Power Supply: $15
- USB Cables: $10
- SD Card: $7

- Total: $117.00

MDR Deliverables

- Show proof of captured signals and computed calibration graph
- Device cable of adjusting volume in reasonable time
- Single microphone system
- Wired connections (to start)
- No App, Variables will be set manually in code

Thank you

Questions?