Team 1: 3-D Audio Optimization Tool

Albi Bregu, Arjun Chandrasekhar, Ravynne Jenkins,
Stephen Lee, Jonathan Wong

Motivation

- Audio is mastered in a “perfect” setting
- Audio is configured by professional audio engineering and technicians
- Home audio systems have imperfections in their room’s frequency response
- Simple and straightforward implementation to fix this issue
- Should work with all speaker systems

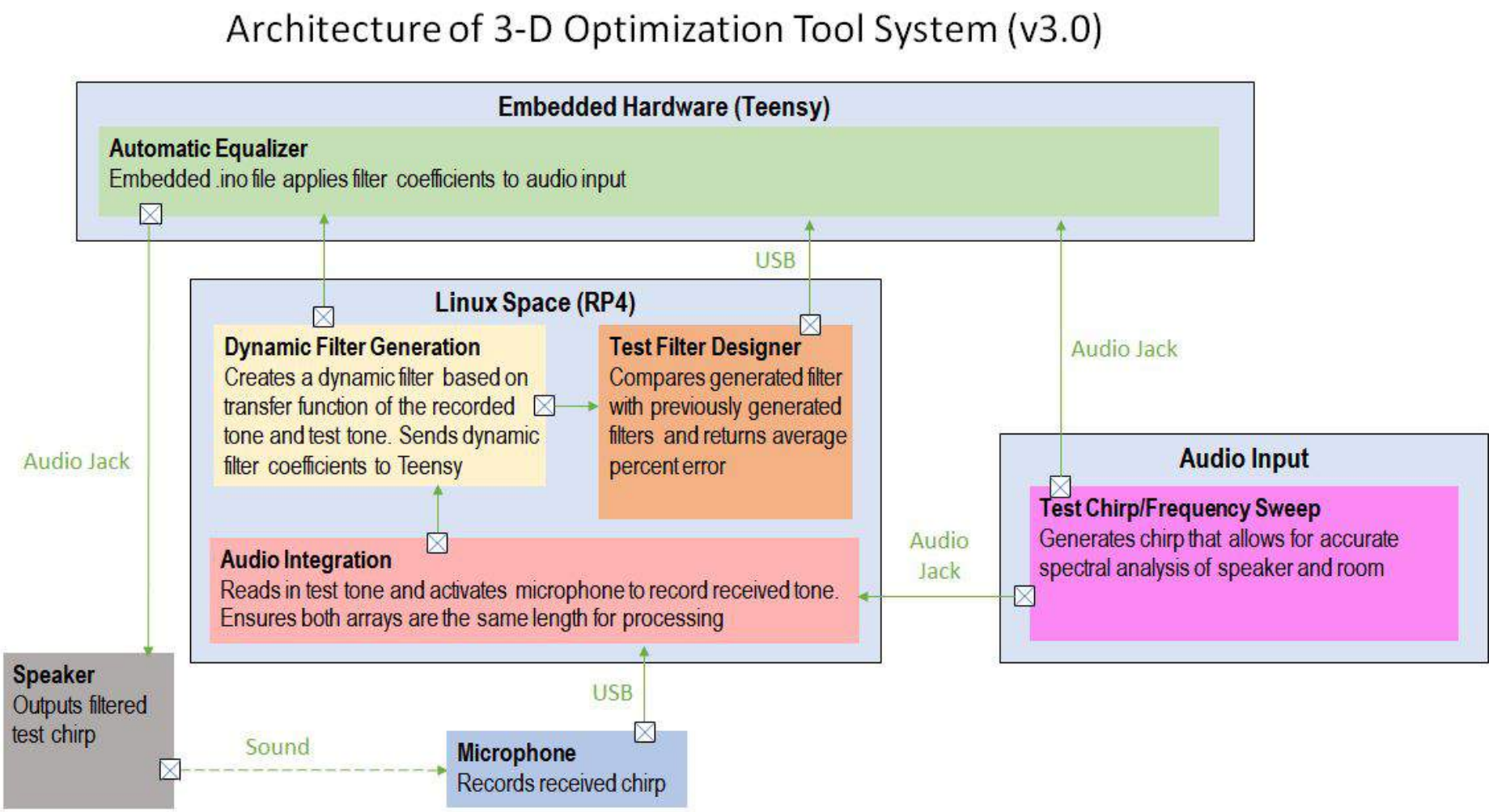
Previous Work

- Audyssey
- THX Tune Up
- Multichannel Acoustic Calibration (MCACC)
- Digital Cinema Auto Calibration (DCAC)
- Yamaha Parametric Room Acoustic Optimizer (YPAO)

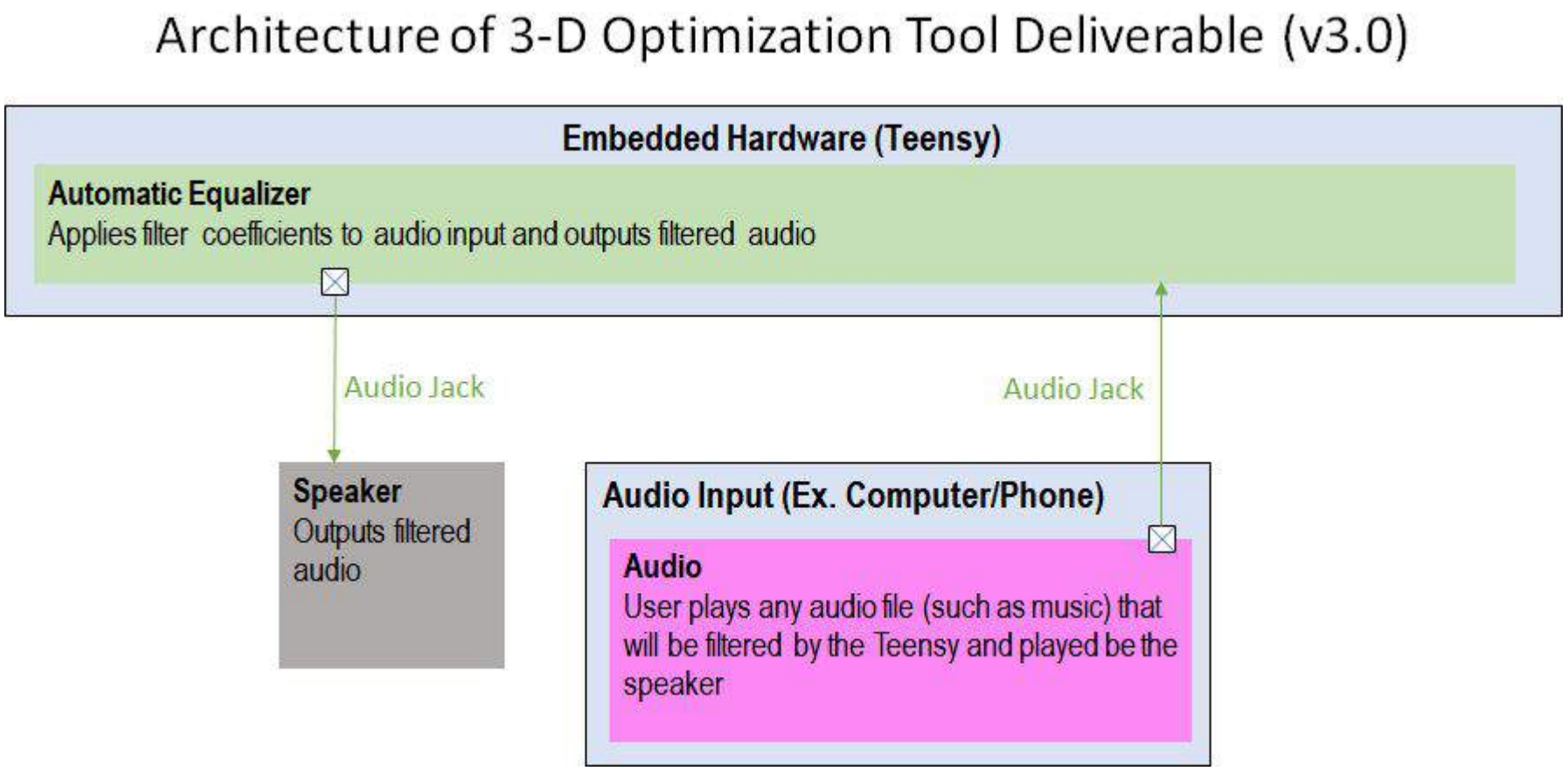
Project Description

- Dynamic digital filter creation
- Filters speaker output to match input
- Corrects variations due to environment and speaker characteristics
- Creates an approximation of the original signal
- Tested with 5.1 Surround Sound Audio System
    - 5 speakers, 1 subwoofer

Testing Block Diagram



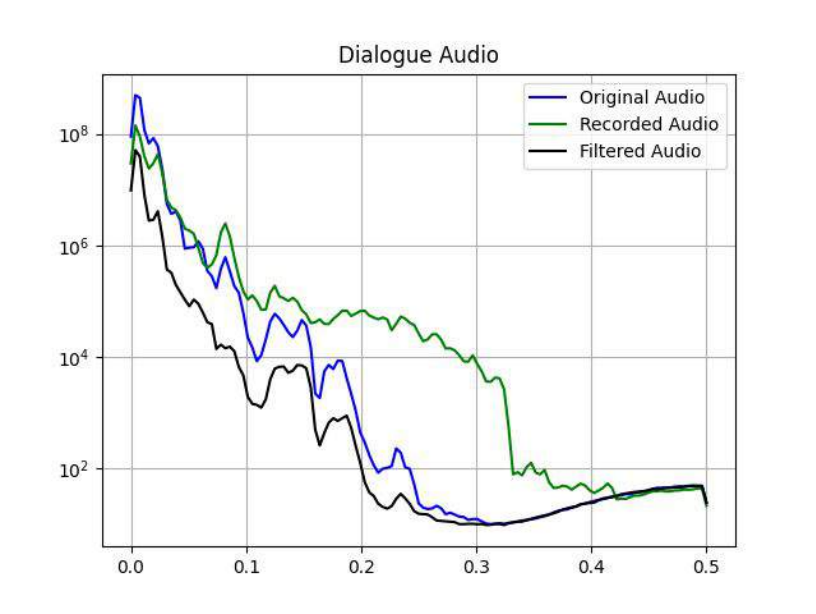
Deliverable Block Diagram



Filter Generation Signals & Test Tones

- Key component is developing proper test signals
- Need to cover the right frequency ranges
  - Three types of speakers in 5.1 system: center, left/right, subwoofers
- Different types of signals used and combined
  - White noise
  - Pink noise
  - Chirps
  - More complex signals
- Shorter filter generation signals vs. Longer speaker test tone signals

Iterative Dynamic Filter Generation



- Building a filter based on a received input and a known desired output
- Shape of signal is most important factor
- $y(t) = x(t) \oplus h(t)$
- Convert signals into frequency domain
- $Y(\omega) = X(\omega)H(\omega)$
- $H(\omega) = Y(\omega) / X(\omega)$
  - $X(\omega)$: recorded signal
  - $Y(\omega)$: original signal
  - $H(\omega)$: generated filter
- Apply filter to signal and record new signal
- Process repeated until original and recorded are within tolerance

Teensy and Raspberry Pi 4 Integration

- Sending filter coefficients from Raspberry Pi 4 to Teensy
- Teensy receives coefficients and creates a new filter using them
- This filter is applied to the audio signal on the Teensy
- The filtered audio is then played out onto the speaker
- Teensy is now portable digital filter

Class Concepts Used

- FIR filter generation
  - Coefficient generation through transfer functions
- Microcontroller Implementation
  - Live audio signal filtering
- Serial Communication
  - UART Byte Transfer

Final Product

- 3D Audio Optimization Tool
  - Filter generation software
  - Run with .wav files
- Portable digital audio filter (Teensy)
  - Works with any analog audio source
    - Phone, laptop, TV, etc...

Filtering Demo

Portable Teensy Demo

Reflections

- Syncing up audio signal is a non-trivial challenge
- Filtering creates significant delay
  - Current implementation is audio only
- Smaller filters would allow audio/visual sync
  - Very short filter generation tones (10-20 ms)
  - Filter truncation (order of 4800-9600)
- Read serial inputs was not easy to format
- High quality microphone is important

Q/A

3-D Audio Optimization Tool

Albi Bregu, Arjun Chandrasekhar, Ravynne Jenkins, Stephen Lee, Jonathan Wong
[1]Electrical and Computer Engineering, University of Michigan, Ann Arbor, Michigan.

Motivation

- Audio is tested in a "perfect" setting
- Audio is configured by professional audio engineering and technicians
- Home audio systems have imperfections in their room's frequency response
- Straightforward implementation to fix this issue
- Should work with all speaker systems

Deliverable

- 3-D Audio Optimization Tool
- Generates dynamic FIR filter
- Portable digital audio filter (Teensy)
- Works with any analog audio source

Materials and Methods

- RP4 plays audio and generates filter
- Microphone records speaker output
- Teensy applies filter to live audio signal
- Tested with 5.1 Surround Sound Audio System
  - 5 speakers, 1 subwoofer

Reflection

- Syncing up audio signal a non-trivial challenge
- Filtering creates significant delay
  - Current implementation is audio only
- Faster filtering would allow audio/visual sync
  - Very short filter generation tones (10-20 ms)
  - Filter truncation (order of 4800-9600)
- High quality microphone is important

High-Level Block Diagram

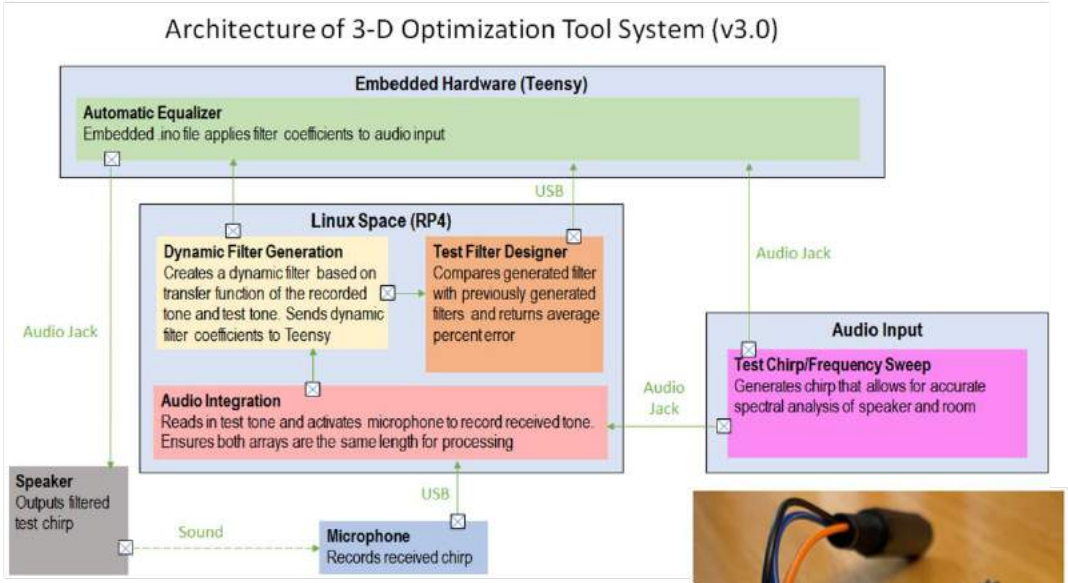


Figure 1: Block diagram of the system

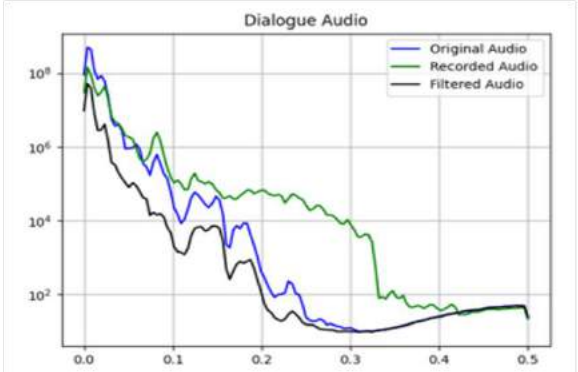


Figure 2: Dialogue Audio

Figure 3: Final Deliverable

ECE ELECTRICAL & COMPUTER ENGINEERING UNIVERSITY OF MICHIGAN