Hand Gesture-Controlled DJ Board

EECS 452: Digital Signal Processing Design Lab – Fall 2025

Durrah Azdi, Cordy Wettstein, Pirachat Ittiravivong, Anjani Malli Reddi

MICHIGAN ENGINEERING
UNIVERSITY OF MICHIGAN

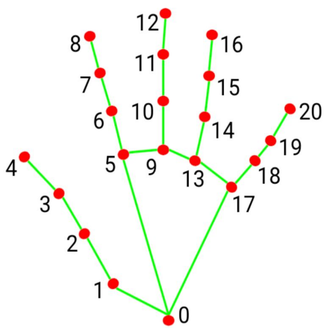
Introduction and Motivation

Current research shows that gestural interfaces boost expression in live music by mapping body motion directly to sound, enhancing stage presence and audience engagement. Our project is a real-time audio processor that allows users to control sound parameters, such as volume, pitch, echo, and filters using hand gestures instead of physical knobs or sliders. Our system fuses IMU on Teensy + OpenCV on RaspberryPi to demonstrate a robust and scalable embedded audio system.

Algorithm and Hardware Breakdown

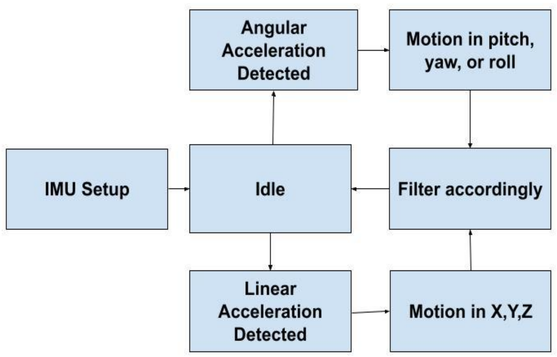
- Hand Gesture Recognition: Captures live frames from the Raspberry Pi camera and classifies gestures in real time using MediaPipe.
- Motion Analysis: Uses Adafruit BNO055 IMU data to prioritize and execute the appropriate motion based on angular and linear acceleration.

Hand Gesture Classification



0. WRIST
1. THUMB_CMC
2. THUMB_MCP
3. THUMB_IP
4. THUMB_TIP
5. INDEX_FINGER_MCP
6. INDEX_FINGER_PIP
7. INDEX_FINGER_DIP
8. INDEX_FINGER_TIP
9. MIDDLE_FINGER_MCP
10. MIDDLE_FINGER_PIP
11. MIDDLE_FINGER_DIP
12. MIDDLE_FINGER_TIP
13. RING_FINGER_MCP
14. RING_FINGER_PIP
15. RING_FINGER_DIP
16. RING_FINGER_TIP
17. PINKY_MCP
18. PINKY_PIP
19. PINKY_DIP
20. PINKY_TIP

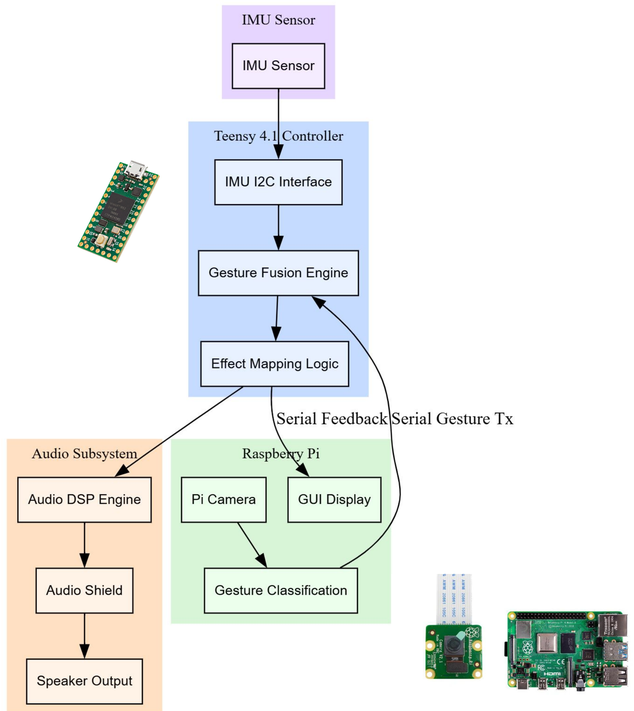
IMU Algorithm Diagram



Signal Processing Techniques

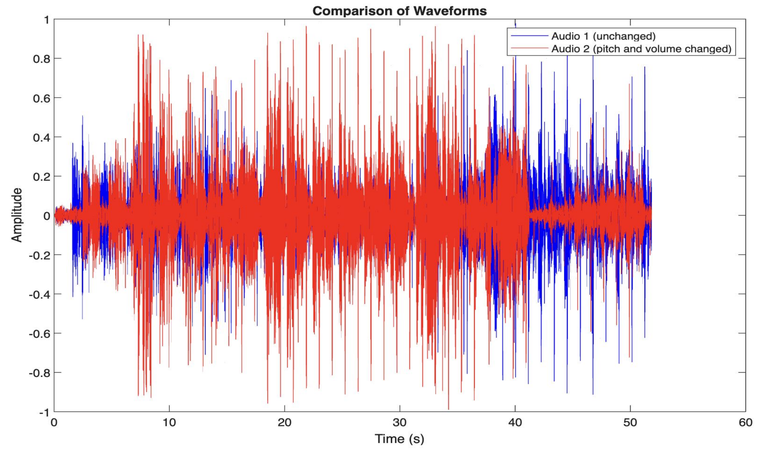
Real-time Filtering: Low-pass and high-pass processing for effects.
Pitch & Effects: Real-time pitch shift and audio effects.
Gain Control: Dynamic gain and volume adjustment.
Smoothing: Moving-average and smoothing filters for clean transitions.
Motion Tracking: Point-to-point camera tracking for gesture control.
Control Modes: Motion-based start/stop, overlay, and EQ switching.

System Architecture and Specifications

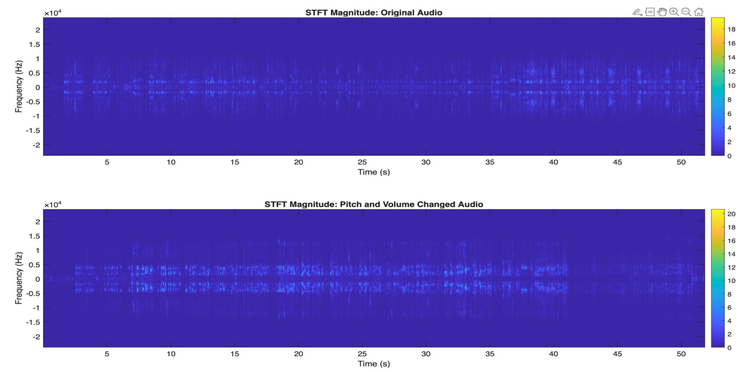


- The Raspberry Pi is connected to the camera and Teensy 4.0.
- Adafruit BNO055 IMU is wired to Teensy, connecting all hardware systems.
- Camera with 30 FPS, 1080p; IMU with 100 Hz sampling rate.
- Gesture control in the Python file sends appropriate information to the serial monitor of the Teensy.
- Teensy uses its serial monitor information to turn music on and off, change volume, and update pitch.

Testing Validation and Results



Waveform plot of original audio and audio with pitch and volume adjustments via hand gestures.



Spectrogram plot of original audio and audio with pitch and volume adjustments via hand gestures.

Challenges

- Merging the camera and IMU outputs proved difficult, but was solved by condensing everything into two files.
- Finding appropriate thresholding for the IMU was difficult, but through rigorous testing we found working numbers.
- Implementing a GUI proved to be difficult as it increased latency.