Dancing Hexapod
Group ID: 12

Manas Chaudhari
Ayesha Mudassir
Siddharth Sarangdhar
Kedar Tatwawadi

Problem Statement

- Making an autonomous dancing that will choose dance moves based on the beats and the tempo of the song.

Requirements

- Functional
  - Liveness: Perform dance moves in response to the data received.
  - Safety: Hexapod shouldn’t lose balance, or cause damage to itself.
  - Timeliness: The delay between beat and the response should not be noticeable.
  - Hexapod should perform at least 5 different moves.
  - A UI for interaction with the Hexapod

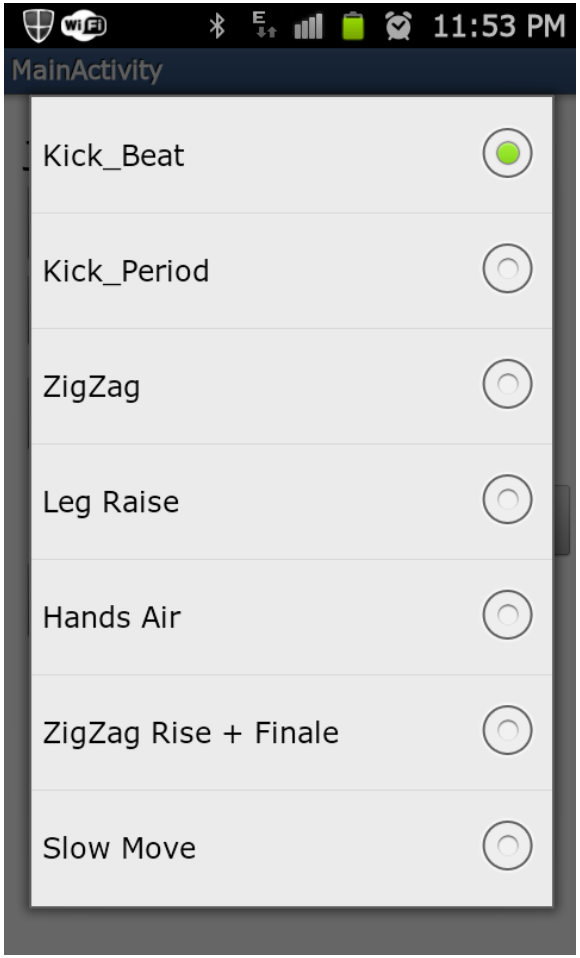
Requirements

- Non-Functional
  - Wireless Communication
  - Movements shouldn’t be jerky.
  - The user must be alerted in case of failure by the Hexapod.

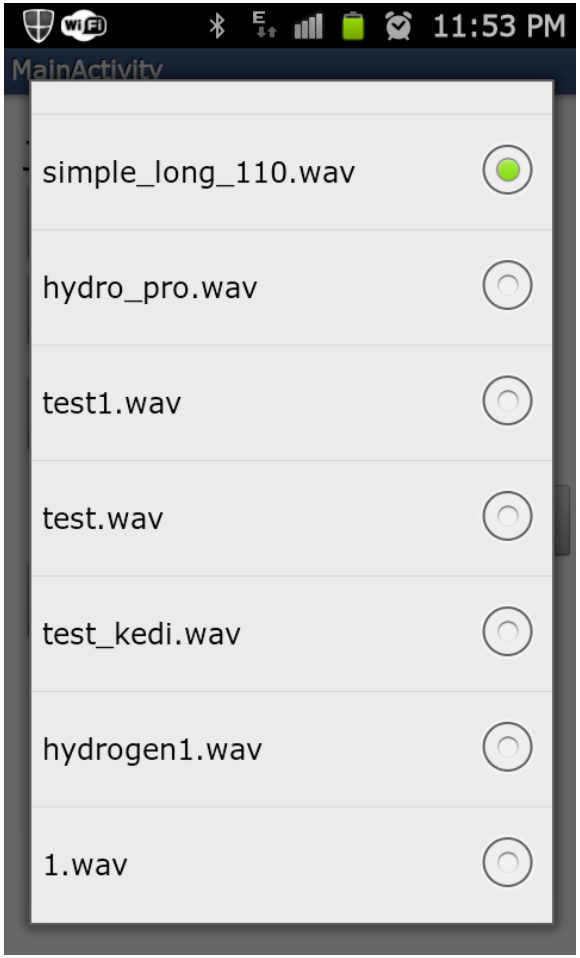
Implementation Details

UI on Android

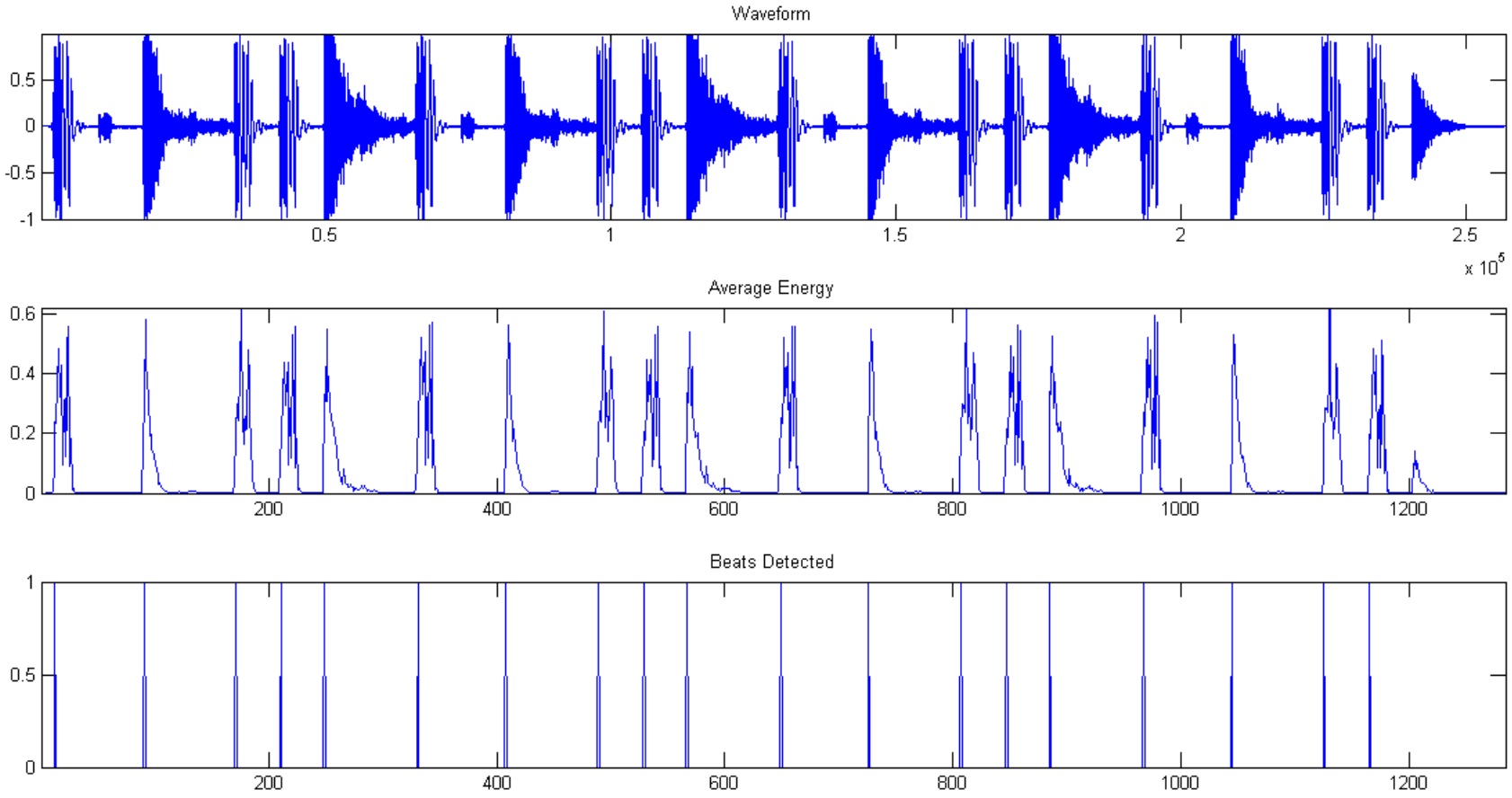
Dance Moves


11:53 PM
MainActivity
Kick_Beat
Kick_Period
ZigZag
Leg Raise
Hands Air
ZigZag Rise + Finale
Slow Move

Beat Patterns



Beat Detection



Communication

- Hexapod processes around 1600 interrupts per second.
- Minimal decision making on Hexapod for Timeliness.
- Designed the communication protocol with minimum payload size.
- Overwrote Xbee Port UART0 for Bluetooth Operation, as Hexapod doesn’t have a spare UART port.

Levels of beat detection

- Audio with only single periodic beat
- Audio having single beat but with variable periods
- Audio with multiple beats with different periods with same intensities
- Audio with multiple beats with different periods with different intensities
- Audio with lyrics or other instruments having rhythmic elements which are not beats.

Deviation from problem statement

- Static processing vs Streamed processing
- Autonomous dance move selection

Energy Calculations

- Due to very high current requirement of hexapod, experimental energy profiling was difficult.
- Battery rating: 5000 mAh, 7.4V
- Average current when servo motors switched on <= 10A (30 mins time for discharging)
- Average power requirement <= 74 W
- <= because of full discharge assumption.

Problems Faced

Hardware Limitations

- Hexapod
  - Bulky Design. Not suitable for Dance moves.
    - Lifting the bot is difficult due to its weight.
  - Tips of the legs are rubber padded which results in jerks.
  - Non-symmetric servo motor placement. Stability Issues.

Hardware Limitations

- Android Phone
  - Implementation of ~50 order High Pass Filter for beat detection on vocal songs is too intensive for Android device
  - Limited buffer size for raw audio playback (30 seconds)
    - Single dance move per beat pattern played

Angle Calculations

- Calculating min inter step delay for a move:
- NRS-993 Motor specifications: 150 ms /60˚.
- Example:
  - Hands air:
  - Legs 1,3,5: [90 60 20] to [90 0 100]
  - Legs 2,4,6: [90 60,20] to [90 40 0]
  - Max angle difference for any motor = 80˚.
  - Expected min delay: 200 ms.
  - We have taken a safety factor of 2 .

Problems Faced

- Different hexapod than what was used in the past project.
  - Code from past project could not be used.
  - Hexapod GUI was not working
- Bluetooth communication was not working for unclear reasons.
- Extremely high current rating (15 A)
  - Experimental energy profiling was difficult.
- We received the hexapod battery after the first demo which resulted in large delay.

Project Reusability and Deliverables

- Use of Existing Open Source Libraries as far as possible
  - Musicg : Reading PCM data from a .wav file
- Creation of Android compatible Beat-Detection module in JAVA.
- Beat detection code in MATLAB.
- Generation of Documentation (ongoing)
  - Doxygen for code documentation.

Future Improvements

Streamed processing

- Support for long songs
- This would unlock more extensions
  - Multiple moves per song
  - Autonomous selection of dance moves based on decision tree constructed using spectral features.

Other extensions

- Dedicated processor
  - Work on vocal songs by implementing high order filters
- Mounting artificial head on Camera pod to add more flavor to dancing.
- Designing more complex dance moves.

THANK YOU