ECE 398 Design Review

Mobile Music Composition

1. Introduction

The aim of this project is to build a device that can capture motion and tactile data from a musician and generate MIDI signals, music notation, and sounds on a mobile platform.

Unlike many other types of artists, musicians are unable to record their ideas for a new piece of music when they are away from their recording equipment. Unlike a poet or a mathematician, there is no easy way to get an idea with layers musical notation onto a sheet of paper for later consumption. This is a unique problem faced by musicians. To prevent this large creative loss, we propose to build a mobile platform by which musicians can convert the motion of their hands and fingers into samples that can be used to compose music while mobile. Many current solutions exist, such as Air Beats\(^1\), a glove that allows musicians to compose music digitally without interacting with a screen. However, there is yet to be a commercially successful product as there is a high reliance on either expensive pre-existing technology with questionable reliability (digital paper technique), or a lack of throughput capability due to inexpressive language during pattern recognition. Our solution aims to solve both of these problems with current solutions by using a minimal set of viable hardware and an expressive language that can be used to quickly translate motion into synthesizable sound.

i. Benefits

1. Able to compose music while on the move, or when other recording equipments are not available.
2. Ideal for a quick sketch to save creative ideas
3. Able to quickly demo the idea in their mind to other people
ii. Features
   1. An easy-to-learn and expressive gesture language that uses hand and fingers motion to express music.
   2. A wearable hardware that can capture hand and fingers movement and generate MIDI data.
   3. Adaptable recognition algorithm based on each user’s movement
   4. Personalizable shortcuts and gestures can extend the language.
   5. Interface APP runs on user’s’ smartphone to communicate with the wearable hardware
   6. Wireless bluetooth headphones connection
   7. Ability to draw from a sound sample library and attach
   8. Generate music notation in real-time
   9. Instant music playback during the composition
   10. Two or more devices can be connected and work together.
   11. Battery lasts 4 hours after each full charge

iii. Goals
   1. Demonstrate what makes a "language" expressive, easy to learn, and powerful to use. Possible examples: vim, stenotyping, Cornell note taking, Dvorak keyboard layout, American Sign Language, etc.
   2. Evaluate the kinesthetics of the hand to make it easy to keep a rapid pace without damage to your hand health (eg carpal tunnel)
   3. Propose alternatives to the hand for impaired individuals
   4. Develop a prototype language
2. Design

Figure 0: Block Diagram for implementation of system
Figure 1: Motion pattern recognition algorithm

Figure 2: Motion recognition state machine
There are three main subsystems in the system: power, wearable hardware and interface APP.

1. The **power subsystem** is responsible for the power supply of the entire system. The power of our system is supplied by a Lithium-ion battery, and a power board with voltage feedback regulation to maintain a stable 5V+/-.1V with a current supply capability of 0mA to 2500mA.

2. The **wearable hardware** is a hardware subsystem, which contains a vibration sensor unit, memory unit, bluetooth data transmission unit, and a microcontroller.
   - The **sensor unit** includes a flex sensor unit which can be used to detect the motion of fingers. It also includes an inertial measurement unit (IMU), which contains an accelerometer, a gyroscope and a magnetometer. The IMU can be used to detect more complex hand movements and gestures. Angles necessary for gesture detection are measured relative to a gyroscope at the base of the glove.
   - The **memory unit** is used to store all the data needed for the hardware subsystem, including program code and data generated during operation. The memory unit also stores the necessary data for executing instructions. It has an on-chip memory for fast data access and also a DRAM for large data storage.
   - The **Bluetooth unit** allows users to connect their wireless speaker and headphone, and it also enables the communication between the wearable hardware and the smart device that runs the interface APP. The microcontroller is the CPU of the hardware subsystem. It executes instructions and orchestrates communication among different units within the hardware subsystem.
   - The **microcontroller** will be used to coordinate communication between the various hardware subsystems and the user. It will accomplish this by collecting sensor data over a serial connection in a raw format, processing that sensor data into an acceptable format for Bluetooth transmission, and parsing any response over Bluetooth into the appropriate hardware commands.

3. The **interface APP** is a software subsystem that serves as the interface between user and hardware. The APP runs on user’s smart devices. It communicates with the hardware via Bluetooth. It allows the user to replay the music just composed and generates the sheet music for the user. It also runs the pattern recognition algorithm that can recognize user’s motions and generate the music accordingly.
Language Gestures

The language is currently not something we have fully fleshed out, but we are taking cues from American Sign Language to create the details.

Figure 3: Change current instrument

Figure 4: Rewind current track. Reverse direction of right finger for fast forward.

Figure 5: Change volume of current track

Figure 6: Define base note (combination of thumb/finger uniquely defines note)

Figure 7: Change active track

Figure 8: Define relative note transition (thumb position is sharpness/flatness)
3. Requirements and Verification

<table>
<thead>
<tr>
<th>Requirements</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Battery lasts at least 4 hours (5V +/- .1V, 2500mA)</td>
<td>Fully charge a battery 100 times; record the lifetime for each time and then compute the average lifetime</td>
</tr>
<tr>
<td>Gyroscope within each fingertip needs to be able to distinguish zenith angle to within +/- .1 radians</td>
<td>Take three measurements from the gyroscope under z-axis, y-axis, and x-axis bias using known angle.</td>
</tr>
<tr>
<td>Gyroscope within each fingertip needs to be able to distinguish azimuth angle to within +/- .1 radians</td>
<td>Take three measurements from the gyroscope under z-axis, y-axis, and x-axis bias using known angle.</td>
</tr>
<tr>
<td>Sensor unit can capture the motion with greater than 95% accuracy</td>
<td>Run the motion capture algorithm with all the possible movements in the language 20 times by five different people, and then confirm the accuracy</td>
</tr>
<tr>
<td>Motion recognition algorithm has 95% accuracy</td>
<td>Use the motion data from the above step as testing dataset to test the recognition algorithm and check if the accuracy is above 95%. The detailed testing procedure is described in the tolerance analysis</td>
</tr>
<tr>
<td>Motion recognition allows user defined new gestures/movements with greater than 90% accuracy.</td>
<td>Translate novel observation vectors to internal memory</td>
</tr>
<tr>
<td>Bluetooth connection allows communication between the hardware and the interface APP within 0 to 5.5m</td>
<td>Test the data transmission between the APP and hardware by transmitting testing audio and motion data in the 5.5m range and confirm the transmissions have an packet error rate smaller than 1% successful by examining the packet error rate</td>
</tr>
</tbody>
</table>

4. Tolerance Analysis

**Critical components:**

- Sensor unit is the most critical component of our system. It is essential for capturing the user’s gestures and it is the bottleneck for the performance of
recognition algorithm. The accuracy of the entire system depends on the accuracy of the sensor unit.

- There are two subsystems within the sensor unit: inertial measurement unit (IMU) and flex sensor unit. The accuracy of the sensor unit is a function of the accuracies of the IMU and flex sensor unit.

**Acceptable Tolerance:**
- The acceptable tolerance for the sensor unit should be within 10%, since this is the maximum error rate for the recognition algorithm to be able to function. If the accuracy for the sensor unit is less than 90%, then the recognition algorithm will not be able to distinguish different gestures, and thus greatly impair overall performance of the system.
- The accuracy of the sensor unit can be calculated as a function of the accuracies of IMU and flex sensor:
  \[
  \text{Accuracy(Sensor Unit)} = \frac{3}{4}\text{Accuracy(IMU)} + \frac{1}{4}\text{Accuracy(Flex Sensor)}
  \]
  The accuracy of the IMU is weighted more heavily than the flex sensor as the measurements it takes are more critical to normal function.
- The IMU has more impact on the accuracy of the sensor unit because the gestures are mainly determined by the hand movements, whereas the finger gestures only provide a supplementary information about the user’s gesture. Also, flex sensors in general are not known for their accuracy.

**Test Procedure:**
To test accuracy of the overall system, we need to first test the accuracy of the IMU and the accuracy of the flex sensor unit separately.

**Testing flex sensor unit:**
The flex sensor only has to distinguish whether the bend is between 0° to 45° (closed fist) or between 45° to 90° (open palm). The following is the test procedure for testing the flex sensors:

1. Mount all flex sensors on the hardware.
2. Bend the each flex sensor in the range of 0° to 45° 100 times and record the output angles from the flex sensor in a binary representation. We will test to 5° of granularity, giving 10 total tests for each range. If the output angle falls within the range, then record as 1; otherwise, record as 0.
3. Repeat the same experiment for the range of 45° to 90°.
4. Compute the accuracy by summing up the total all the results and divided it by the total number of results.
Testing IMU:
The main components from the IMU that are useful for our system is the accelerometer and gyroscope. Both units should have an accuracy higher than 90%. The overall accuracy of IMU is the average accuracy of the accelerometer and the gyroscope.

1. Testing Accelerometer:
   - Test the accuracy of the gravitational acceleration
   - Use measurements of another accelerometer with known superior performance (with accuracy within 1%) as the ground truth and compare it with the measurements with our accelerometer readings to obtain the accuracy for our accelerometer

2. Testing Gyroscope
   - Make sure that in the duration of 4 four hours, the drifting effect of the gyroscope will not make the accuracy of the gyroscope unacceptable:
   - Run the gyroscope for four hours.
   - Set the gyroscope on a constant rotation rate of 60°/s around all three axises in the phone coordinate system, and record the readings from the gyroscope.
   - Compute the accuracy of the gyroscope by comparing the measured value with the ground truth.

Derive the accuracy of the sensor unit:
After testing IMU and gyroscope individually, we can derive the overall accuracy of the sensor unit by using the accuracy function.

5. Cost
   Our estimate is that in total each group member will contribute 100 hours to the project, averaging 12.5 hours per week. This estimate comes from a conservative aggregation of the amount of time it will take to implement the simpler features and the the extended testing that will take place in the last three weeks of the project.

   Labor:
   - ($40/hr) * (2.5) * 100 hours = $10,000/partner
   - ($10,000) * (3 partners) = $30,000

Parts:
<table>
<thead>
<tr>
<th>Part</th>
<th>Cost (per unit)</th>
<th>Quantity (units)</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>MSP430 uC</td>
<td>$1.54</td>
<td>10</td>
<td>$15.40</td>
</tr>
<tr>
<td>785060 2500MAh</td>
<td>$14.95</td>
<td>10</td>
<td>$149.50</td>
</tr>
<tr>
<td>LSM303DLHC (IMU)</td>
<td>$13.46</td>
<td>10</td>
<td>$134.60</td>
</tr>
<tr>
<td>239-BT800 (Bluetooth unit)</td>
<td>$10.60</td>
<td>10</td>
<td>$106</td>
</tr>
<tr>
<td>Elastic Bands</td>
<td>$1</td>
<td>20</td>
<td>$20</td>
</tr>
<tr>
<td>Other (resistors, capacitors, etc.)</td>
<td>$50</td>
<td>1</td>
<td>$50</td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td><strong>$475.50</strong></td>
</tr>
</tbody>
</table>

Grand Total: $30,475.50
6. Schedule

Week 1:
Research and Preparation
- Adam: Sensor Research and Purchase
- Jeremy: Microcontroller and Memory Subsystem Research and Purchase
- Ziheng: Powerboard Design and Printing (or Purchase)

Week 2:
Sensor and Power System Validation
- Adam: Complete IMU Validation
- Jeremy: Complete Bluetooth Validation
- Ziheng: Complete Powerboard Validation

Week 3:
Subsystem and Architecture Design
- Adam: Microcontroller State Machine Framework
- Jeremy: Control Word Specification and Serial Interfaces
- Ziheng: State Machine / Interface input

Week 4:
Architecture Implementation (Units)
- Adam: Flex Sensor Module
- Jeremy: Bluetooth Transmitter Module
- Ziheng: Accelerometer Module

Week 5:
Architecture Implementation (Integration)
- Adam: Flex Sensor to Microcontroller
- Jeremy: Microcontroller to Bluetooth
- Ziheng: Accelerometer to Microcontroller

Week 6:
Mobile Application Implementation (Units)
- Adam: Pattern Recognition Algorithm (acquisition and processing)
- Jeremy: Sheet Music Generation
- Ziheng: Audio Processing

Week 7:
Mobile Application Implementation (Integration)
  ● Adam: State Machine Framework
  ● Jeremy: State Machine Framework/User Interface
  ● Ziheng: User Interface

Week 8:
Full System Integration Testing
  ● Adam: Serialization and Deserialization Testing
  ● Jeremy: Bluetooth Communication Testing
  ● Ziheng: User Experience Testing

7. Ethics and Safety:

7.1 IEEE Code of Ethics:
Our design and product will comply with the IEEE Code of Ethics. The following two points from the code are especially relevant for our design:

1. “to accept responsibility in making decisions consistent with the safety, health, and welfare of the public, and to disclose promptly factors that might endanger the public or the environment”

   We considered the safety and health risks of our design and will provide a list of cautions to take when using our hardware. We also designed our hardware to reduce any potential risks to the users.

2. “seek, accept, and offer honest criticism of technical work, to acknowledge and correct errors, and to credit properly the contributions of others;”

   We will properly cite the contributions from previous research and similar products.

7.2 Safety Statement:
The hardware is operating at a low voltage, so there is no prominent risks electric shock to humans when using the hardware. However, the sensors unit contains fragile components that can be broken when the hardware is not used properly. For example, the flex sensors can be damaged when overbent and the IMU can be damaged if the hardware is dropped to the ground from high places.

List of caution to take when using the wearable hardware:
- Avoid using the hardware with wet hands or near water.
- Avoid dropping or squeezing the hardware.
- Using the hardware within the safe temperature range: 0°C to 45°C (32°F to 113°F)
- Be cautious for the potential electrostatic discharge.
- Before using, check that there is enough space around you. Also, make sure the hardware cannot slip out of your hand. If the hardware hits a person or object, this may cause accidental, injury or damage.
- Be cautious for the hardware overheat. Remove the hardware immediately when the hardware is overheating.

References


