Electronic Piano Tutor: An Autonomous Alternative to Private Piano Lessons with Haptic Stimulation

Joseph Breda, CSE, Matthew Caswell, CSE, Aleksa Deric, CSE, and Cassius Peter, EE

Abstract— **Learning how to play the piano can be a very daunting but also expensive experience. For those just starting to learn how to play or unsure whether they are willing to spend money on a professional teacher yet, we offer a cheaper alternative. Electronic Piano Tutor (EPT) both teaches a person how to play the piano and provides them with feedback on their performance. An LED strip that can be laid down on almost any keyboard tells the user which notes to press while two gloves provide the user with haptic stimulus, teaching them to use the right fingers when playing. Lastly, a microphone records the sound of the note played, allowing the system to determine if the note played was correct, and provide feedback if the player hit the wrong note, all in real time. All that for a price of three or four regular piano lessons.**

I. INTRODUCTION

The piano is a beautiful instrument, but learning how to play it is a very complicated task. It requires the user to be able to keep track of their hand placement relative to the keys, the timing and velocity in their playing, as well as reading sheet music to know what notes to play next, all in virtually real time. The complexity of this task makes the piano a very difficult instrument to teach yourself. Piano tutors are readily available, but are often very expensive, costing an average of eighty-two dollars an hour in the Franklin region [1]. Hiring a tutor has several complications besides cost, such as what happens when the student is too nervous to play in front of someone else, or if the tutor's teaching style doesn't work well with the learner's learning style.

There have been three major attempts to solve this problem. The first is using piano teaching software, such as Synthesia [2]. Synthesia displays a virtual keyboard, and the notes to be played as blocks falling down, past the keys. A block overlapping with a key is an instruction to press the key in question. This negates the need for the player to read sheet music, and Synthesia specifically is able to suggest which finger to use on which key, as well as provide feedback on overall note accuracy if the user is playing a connected keyboard [2]. While this provides an effective teaching method, the program uses its own virtual keyboard, meaning that users still need to mentally translate the keys of the virtual keyboard to the keys on the real keyboard. Additionally, feedback requires a midi keyboard, so this function doesn't work with analog pianos.

J. Breda from Needham, MA (e-mail: jbreda@umass.edu).
M. Caswell from Needham, MA (e-mail: mcaswell@umass.edu).
A. Deric from Mendon, MA (e-mail: aderic@umass.edu).
C. Peter from Franklin, MA (e-mail: cpeter@umass.edu).

The second major solution is what's known as a smart piano. These pianos have velocity sensors and LEDs on each key, allowing you to see what key to press, and detects if you pressed the wrong key. The feedback is beneficial, and the ability to light up keys removes the need to interpret the music being read. While this is an effective way of learning piano, this method has no way of incorporating finger placement. Additionally, the total cost of using one of these pianos is high. The pianos themselves are expensive, especially for a high-quality one, and only work with songs bought through the smart piano's producer requiring the user to spend a few dollars on every song they wish to play.

The final method comes from an academic study into haptic, or touch, stimulation [3]. By wearing a special glove that vibrated the wearers fingers in the same pattern as if they were playing, participants were able to learn the fingerings for a simple song while focusing on doing something else, like studying. Since this method doesn't require actively playing the piano, it allows for more practice throughout the day, and the study proves it is highly effective [3]. While effective, since this was developed for a study, it is not available to consumers. It also only works on really simple songs, that can be played without moving the hands positions. Finally, the rig itself is unwieldy, with the main processor and battery mounted directly on the back of the hand.

Requirement	Specification
Instruction	System should show the user sheet music, what key on the piano that corresponds to, which fingers to use to hit those keys, and detect errors
Synchronization	All signals to the user for pressing a key should arrive within 100ms of each other
Accuracy of note detection	Correctly detect single note melodies as often as it can, with priority on most commonly used notes in middle octaves
Latency of note detection	Note detection should provide live feedback. No more than 1 second of latency
Non-obstructive	System should not obstruct normal finger mobility
Battery life	at least 2 hours
Range	10 feet
Retrofit-ability	Any standard width (48 inch) 88 key keyboard
Cost	The final system should be producible for less than 250 dollars (Equivalent to 3 to 5 piano lessons)

TABLE I
SYSTEM SPECIFICATIONS

The system has to be able to teach a person how to play a piano, meaning it has to display music sheets, highlight the note that is supposed to be played both on the screen and the piano itself and vibrate the right finger on the gloves so that the person knows which finger to use to push down the key and through muscle memory develop that skill. In order to ensure that, all of these outputs need to happen in sync within 100ms. The system should likewise be accurate and timely when recognizing notes and report few false positives if any, as that would interfere with the learning process. More than anything though, the system should be unobtrusive, light and allow some freedom of movement around the piano. Piano practice sessions typically last 30 minutes to an hour, therefore our system has to be able to meet this spec as well and ideally last longer than that. Finally, the system needs to be cheap and capable of fitting to most keyboards, so it is a better value than getting a teacher. The system requirements are summarized in Table I.

II. DESIGN

A. Overview

Our design consists of three main components as shown in figure 1. The first is the Android tablet, which displays sheet music on screen, sends note and finger information to the controller board, and recognizes the played notes using its on-board microphone. The controller board receives note information, displays it on the LED strip, and sends the finger information off to the glove boards. The two glove boards each receive the finger information and vibrate the corresponding fingers. This system does most of the processing on the android tablet, allowing the controller board and glove boards to use simple low power micro-controllers. This allows for smaller batteries on the glove boards which will allow for an overall smaller and less obtrusive design.

B. Note detection/Microphone

The goal of this subsystem is to detect if the user is playing a note and determine which note is being played. We design this system to work in real-time with 1 second latency tolerance.

For note detection we used PyAudio, a audio streaming library for Python to parse input audio information. We used Numpy and Scipy - common scientific computing packages in Python to parse the audio and compute the Fast Fourier Transform (FFT) to obtain frequency information of each audio time snippet. When running this on a computer we were capable of calculating this operation over 10 times per second.

Once the audio signal was in the frequency spectrum, we locate the 3 maximum magnitude frequency peaks of the audio signal. Naively, we can expect that the maximum peak is the fundamental frequency of the note being played. However, this is not often the case in the real world. Even within a constrained environment with little to no background noise, the reverberations of the room or the timbre of the note being played can cause the upper harmonics to surpass the fundamental in magnitude.

However, we found in in our tests that 3 largest harmonics were often at the frequencies of the fundamental or 5^{th} interval frequencies. These lie at the first, second, and third harmonic in the harmonic series. This reflects our understanding of music theory as the fundamental and 5^{th} interval of a note are the characteristic frequencies that make a note sound like that note. If the magnitude of any other interval was greater than these frequencies, then they would no longer qualify as that note.

We designed a small method to check the possible permutations of the 3 frequencies of maximum magnitudes, those most prominent in the Fourier transform of the signal, and to attempt to fit them into the positions of the first, second, and third harmonic positions. That is, are two of the max frequencies the same note name (octaves of each other) and is one of the frequencies 7 semi-tones (a 5^{th} interval) above the others. If so, then our system is confident it has detected a note at the fundamental frequency (one note that appears twice in this set of 3). While this system is still rather naive, it significantly outperforms the alternative of assuming the frequency of maximum magnitude at each time step is a note with some arbitrary confidence check.

We would like to further the development of this subsystem by implementing the McLeod Pitch Method (MPM) [4] or some smart alternative. We would also like to have our system return the note number $[0, 87]$ as opposed to note name $A, A\#, B, B\#, etc.$. This is a more complex task because the second harmonic in the harmonic series can easily be mistaken for the fundamental in detection and this would result in the same note name, but a note number 12 semitones above ground truth. To do this we will need to further examine the contours of the different piano note's Fourier transform. We have determined that the characteristics of a low note on the piano differ from the high notes in shape of Fourier transform. The lower notes in the frequency spectrum get - what is often referred to in the music community as - "muddy". This means that the individual harmonic frequency peak get get wider spread and bleed into each other, forming rolling hills. Whereas in the upper end of the frequency spectrum where the higher keys lie, the harmonic peaks are very narrow and easily distinguishable. This means a system that checks the different positions on the piano may need to be implemented.

We will consider the note recognition complete when it is capable of correctly classifying all notes in twinkle twinkle little star without exceeding 1 error (false positive or false negative) per play through. We will call this the $AleksaTest$ in honor of our musically inept team member who will serve as the test subject for this metric.

C. Android tablet

The Android tablet is one of the main components of our system its purpose is to provide an interface for the user to interact with all the other components. The tablet has several jobs including displaying sheet music and highlighting the

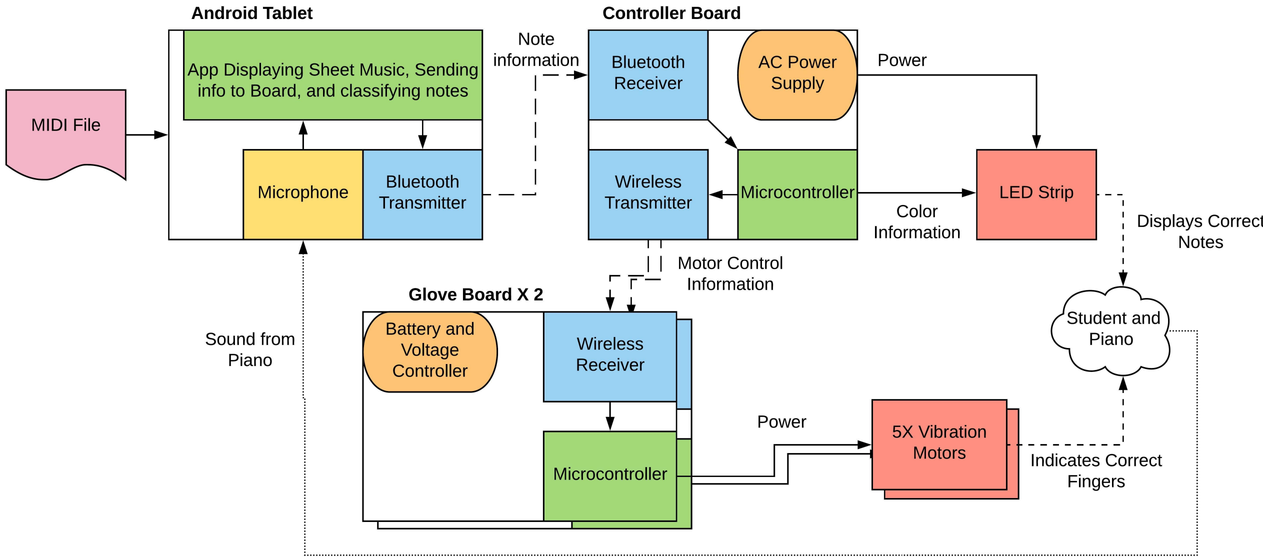

Fig. 1. Block diagram of the system

notes that have to be played, checking which note and finger should be used to play said notes and sending that information to the main controller board via Bluetooth, and most importantly, listening to the user playing and checking for errors. Currently we are developing this interface on a phone, however eventually we plan to purchase a bigger tablet that will used in the final design. Because both devices are Android based we expect there to be no issues in going from the phone to the tablet. While it is still early to decide which tablet to purchase, several from the Amazon's Fire family look very appealing to us.

This functionality will come in the form of an app that will implement the things described above. Most of our development up to this point has been done using Android Studio and we expect to continue using the same program. Aleksa on our team already has some app development experience and has therefore been the member focusing on developing this system. Currently our app is able to communicate to the control board via Bluetooth, which was one of our MDR goals. Google provides a detailed guide on how to use and interact with the Bluetooth API on its Android developers website [5] along with examples, thus serving as the best and primary resources as we are going through the process. As described the app also has to be able to read a MIDI file and convert it to sheet music to display. As of now we are still in the process of researching what kind of structures and APIs are available for us to do this, though we believe this function will not be hard to implement and is one of our least priorities.

To listen to the piano we plan to use the microphone that comes integrated with the tablet that we will purchase. Likewise, this is another part of our project that we are currently researching how to do. To run the classifier written in Python on an Android device which uses Java we are planning to use one of many libraries available to achieve just this task such as Pyqtdeploy or QPython.

Finally, if required, we have many backup ideas. For example, if it turns out that the integrated microphone is not high quality enough, we plan to purchase an external microphone. Likewise another major requirement for this system is to be able to do audio processing in near real-time, however if it turns out that whatever device we get is not capable enough we can always switch to another more powerful platform such as a laptop, though given the hardware that most modern mobile devices pack we do not anticipate either to be a problem.

D. Controller Board and LEDs

The controller board is responsible for receiving note and finger information from the Android tablet, lighting up LEDs on an LED strip placed over the keys, and passing the finger information off to the glove boards. The control board uses an AT-09 Bluetooth chip to communicate with the tablet over Bluetooth. The ATmega328P micro-controller sees this as a serial link from the tablet to the controller board that is used to receive strings of ASCII characters.

The syntax of this communication is broken up into commands separated by semi-colons. Each command starts with either a y, n, or f character followed by a comma separated list of numbers. For commands starting with y, the list of numbers will be interpreted as a list of keys that will have their LED's turned on. The keys are numbered from zero for the leftmost and lowest key, to 87 for the rightmost and highest key. Commands starting with n will turn off the corresponding note LED's. Commands starting with f indicate fingers to vibrate. Each finger is numbered with 0 being the left pinky finger and 9 being the right pinky finger.

The controller board reads in all of these commands, and turns on or off the corresponding LEDs for the y or n commands. For the f commands, the board re-formats the command from a comma separated list of fingers into just a string of those finger numbers with no separators, and sends the new command off to the glove board.

For the LED light strip we are using the W2812 LED strip. This provides us with individually addressable RGB LEDs at a density of 144 LED's per meter. When laid over a keyboard the individual LED's do not line up one to one with the piano keys and there are multiple LEDs per key. We plan on creating a static mapping of note number to LED's to allow us to light up individual keys. To mount the LED's, we are planning on fixing the strip to a rigid wooden backing that can be placed on the back of the piano keys. The LED's are very thin and we envision the wooden bar only being as wide as the LED strip so the bar will not interfere with playing.

The operation of the controller board is a major factor in meeting many of our system specifications. The most difficult spec for the controller board is synchronization of all the signals delivered to the user. We want the LED's to light up within 100ms of the fingers vibrating. In our preliminary testing, there is a delay between the tablet and the controller board as well as a delay between the controller board and the glove boards, but this delay appears to be small and constant. This will allow us to add in constant delays that should allow all signals to be delivered to the user within the 100ms specification.

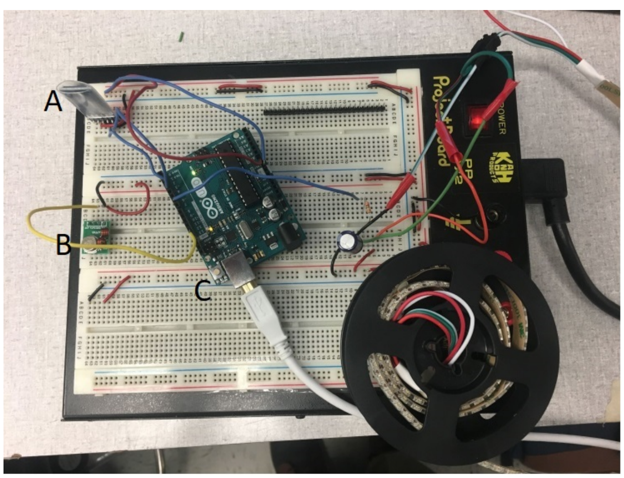

Fig. 2. Current controller board implemented with an ATmega328P micro-controller on an Arduino platform (C) with an AT-09 Bluetooth chip (A) and 433MHz wireless transmitter chip (B).

E. Gloves

The glove board currently consists of an ATmega328P micro-controller on an Arduino platform with a 433MHz wireless receiver chip, and five 1cm vibration motors. The glove boards will be made on PCBs which will allow them to be small and unobtrusive. The 433MHz wireless receiver provides a serial link between the controller board and both glove boards. There are no channels, so both glove boards receive the same signal. The controller board sends a string of numbers such as "049" which is interpreted by the glove boards which maps each number to one finger as "vibrate left pinky, left thumb, and right pinky". Each glove board will be programmed to be the left or the right glove so it only responds to numbers 0-4 or 5-9 respectively. When a finger is vibrated, it's corresponding vibration motor is turned on for a fixed amount of time.

The glove board is responsible for meeting the range, battery life, and non-obstructive system specifications. We were able to successfully transmit data from the controller board to the glove board from more than 30 feet away with no loss. From these results we believe that we will be easily able to meet the 10 foot range specification with the current transmitter and receiver hardware. We also measured power consumption of the entire glove board, finding a 100mW power draw when idle and a 425mW draw with all 5 motors running. From these numbers we believe that a small rechargeable battery will be able to give us the 2 hour battery life while being light and non-obstructive when mounted to the user's upper arm.

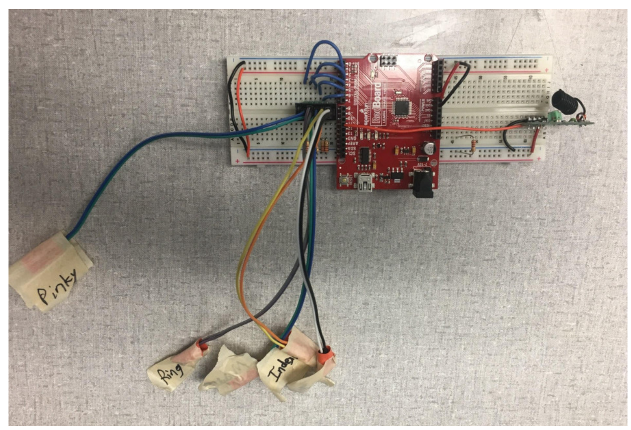

Fig. 3. Current glove controller board with an ATmega328P micro-controller on an Arduino platform with a 433MHz wireless receiver chip, and five 1cm vibration motors.

III. PROJECT MANAGEMENT

A. Deliverable

For MDR we were able to assemble the interfaces that our system uses to connect all sub components. We defined a custom syntax for transmitting signals between the Android interface and main controller board via Bluetooth, and wrote a parsing function to be run on the main controller board which relays relevant information to the glove boards. We were able to demonstrate this by sending signals from an Android phone to the controller board which illuminated LEDs along the LED strip at the desired key number and illuminated LEDs connected to the glove board to simulate the vibration stimulus of the gloves. We had one working example of a vibration motor connected to the glove board, but used LEDs for the demo for convenience. We successfully wrote an audio signal processing program which accepts a stream of audio information and in near-real-time identified the letter name of the piano note being played if there was one. This program uses the unique frequency footprint of the piano note to confidence check the identified fundamental tone at each time-step. For CDR we are planning on moving almost all of our systems closer to their final forms. We want to mount the LED strip on its rigid backing and program it to align with the keys on a piano. We want to have our glove boards on PCBs and mounted on a prototype arm sleeve. We also want to improve our note classification and have it running on an Android device instead of a laptop. We believe that having these tasks done by CDR will put us in a great spot for finishing the entire project on schedule.

B. Team Member's Expertise

Each member has taken ownership of a single subsystem which they are primarily responsible for. This section will outline each member and their expertise. Aleksa takes primary ownership of the Android application and implementing the Bluetooth capabilities on the Android device which will be a part of our final custom app. Matt takes primary ownership of the micro-controller and embedded system programming. He has developed the code which runs on our micro-controller and has assembled the a portion of the current hardware implementation. Cassius takes primary ownership of the hardware implementation and connections between our actuation mediums (LEDs and vibration motors) and the micro-controllers. Joe takes primary ownership of the software performing the audio signal processing for note recognition.

C. Collaboration

Though each member takes ownership of a subsystem there is implicit collaboration between members when interfacing subsystems. Joe and Aleksa work closely on the Android and audio signal processing code as this code will soon be moved onto the Android. Matt and Cassius work closely on the hardware and embedded system software. And we all work together on bringing the full loop together.

D. Communication

Our team communicates in an ad hoc fashion through Slack, the open source chat room application. We meet weekly with our team chair Professor Holcomb, as well as meeting as needed as a full team prior to bench sides and large presentations. When implementing subsystems into the full loop or when developing interfaces between subsystems the group members taking ownership of the relevant subsystems work together in person.

IV. CONCLUSION

Despite falling short of one MDR deliverable, in its current state our project is still on track. Our original goal for MDR was to demonstrate a system capable of listening to a note and displaying it by lighting up the correct LED on the LED strip, however we were unable to fully complete this loop. Still, all the critical components of our system are functional:

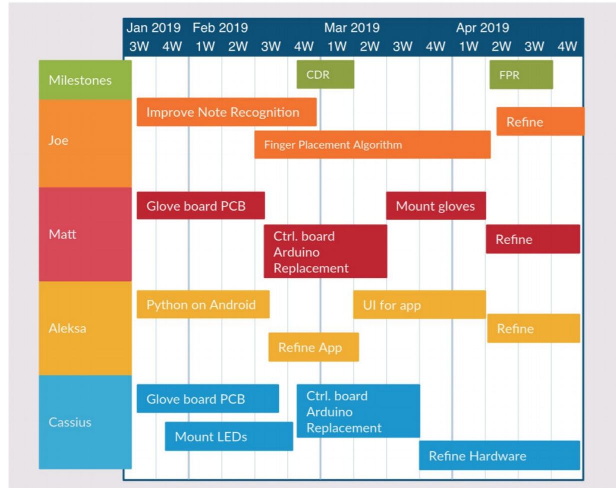

Fig. 4. Gantt chart of next semester

the main communication line from the Android device to the control board, then to the LED strip and glove boards, is working and we have a baseline model for detecting notes. Our focus is now on merging these two systems together and further refining them.

Moving forward the amount of work that we are going to have to put in will most likely increase. On the software side, the note detection algorithm needs to be improved significantly and the code for it, which is currently written in Python, needs to be ported over to Android. We hope that using one of many libraries that make it possible to execute Python on Android will help us avoid having to rewrite the note detection code from scratch. Similarly, the app that we have is currently very priliminary, therefore we also plan to spend a limited amount of time improving its user interface. Our hope is to have note detection completely functional on Android by our CDR presentation, after which focus will entirely shift to developing a finger placement algorithm.

On the hardware side of things all the individual components are currently working and running off three separate ATMega32 controllers mounted on top of Arduinos. However, Arduinos are only allowed for prototyping our design which means that we will have to find another way to run the controllers. Preferably, this will be our own custom printed circuit board. Because we hope to stick with the ATMega32 as it is cheap and meets our design specifications, we assume that almost all of the code we currently have will still be usable, though it is possible that certain sections relating to how the controllers communicate to the wireless modules will have to be expanded or rewritten.

Finally, we would like to take small steps towards finalizing our project by moving away from breadboards and wires and building finalized components. This means mounting the LED strip on some sort of rigid backing or bar that will allow it to be placed over the piano keys and getting a proper pair of gloves and arm sleeves to mount the system on. During this time we would also like to test where and how the vibration motors should be placed on one's hands and design and get manufactured custom PCBs for each of our individual hardware subsystems.

Overall, we are happy with how our project is progressing and though there is still lots of work to be done, we are confident that we will be able to complete the project.

ACKNOWLEDGMENT

We would like to extend our thanks to our advisor, Professor Daniel Holcomb, for his support and guidance throughout the semester. We would also like to thank Professor Frasier, Professor Noh, Professor Hollot, Shira Epstien, and Baird Soules for providing us with useful feedback during evaluations and bench-side meetings being great resources.

REFERENCES

[1] Takelessons. (December 2018). Piano lessons near Franklin, MA. Retrieved from https://takelessons.com/

[2] Synthesia. (December 2018). Synthesia index page. Retrieved from https://www.synthesiagame.com/

[3] PianoTouch: A wearable haptic piano instruction system for passive learning of piano skills. 2008 12th IEEE International Symposium on Wearable Computers, Wearable Computers, 2008 ISWC 2008 12th IEEE International Symposium on. 2008:41. Accessed Dec 18, 2018. doi: 10.1109/ISWC.2008.4911582.

[4] McLeod. P, Wyvill. G, "A Smarter Way to Find Pitch", Proc. International Computer Music Conference, Barcelona, Spain, September 5-9, 2005, pp 138-141.

[5] Android Developers, Bluetooth Overview. [Online]. Available: https://developer.android.com/guide/topics/connectivity/bluetooth. [Accessed Oct. 2018.]