Guitar Hero Bot: Motion Planning to Solve Rhythm Games with Robotic Manipulation

1st Ryan Hourican
6.4210 Robotic Manipulation
Cambridge, United States
ryanh24@mit.edu

2nd John Readlinger
6.4210 Robotic Manipulation
Cambridge, United States
jred@mit.edu

3rd Noah Fisher
6.4210 Robotic Manipulation
Cambridge, United States
nfisher7@mit.edu

***Abstract*—We investigate whether a robotic arm has the speed and dexterity to perform at Guitar Hero better than a human. Guitar Hero, like many rhythm games, requires quick hand-eye coordination. Thus, it provides an interesting context to test the limits of robotic perception and manipulation. This problem was solved with hard-coded hand positions on a wsg robotic hand. The key challenges addressed are the transitions between buttons and the speed at which it happens. The final product includes a simulated guitar with 5 buttons on it and a kuka iiwa that will play the guitar with near perfect accuracy.**

***Index Terms*—**

I. Introduction

Growing up, many kids play Guitar Hero or some equivalent music-based game. Although they are games, these activities can be seen as exercises in hand-eye coordination and motor control. Conscious effort to play Guitar Hero soon becomes second nature, and it feels as though your own human fingers become more dexterous and can reach farther for notes.

From the beginning of the term, our group of three was drawn toward the completion of games that require dexterous hand movement. Programming a robotic arm to play Guitar Hero fulfills this goal, and it is also a project that has lots of potential for expansion. Oftentimes, in the field of robotic manipulation, robots are confronted with problems that are easy for humans but difficult to program (ex. tying shoe laces). We want to program a problem that is difficult for a human to accomplish, pushing the boundaries of dexterous hand movement, particularly speed.

Rhythm game simulation has been used in a variety of contexts, like physical therapy. The bottom line is that our project doesn't interface with humans, it is mainly to test the software, and eventually hardware. Our project is designed to be an iterative process, with possible improvements in perception and incorporation of multiple robot units in the future. If time and resources permit, we aim to replicate at least one actual game of Guitar Hero on hardware by the end of the term.

II. Related Work

A. Playing Piano with a Robotic Hand

We drew the most inspiration from a past project in 6.4210, Playing Piano with a Robotic Hand by Seong Ho Yeon [5]. His work was especially helpful in showing the ability to construct hinges and buttons in a Drake environments which was essential when constructing our simulated guitar. Additionally, his demonstration videos showed us how it was possible to stretch the joints of the allegro hand beyond what it appears to be able to handle, which was something we aimed to integrate but we are excited to be working towards in the future.

B. Optimal Control with Learned Local Models: Application to Dexterous Manipulation

To help us successfully complete this project, we draw inspiration from previous research. One important paper we reference is "Optimal Control with Learned Local Models: Application to Dexterous Manipulation" [1], which gives us a guiding example for how a guitar hero playing project has been executed but within a different framework. Namely, the focus of this paper is an FPGA-based approach for playing guitar hero, but lacks the dexterous manipulation which we will employ in our project. Having said that, this paper gave us an idea for how to potentially create our perception pipeline.

C. Dexterous Manipulation for Multi-Fingered Robotic Hands With Reinforcement Learning: A Review

Another paper we use is "Dexterous Manipulation for Multi-Fingered Robotic Hands With Reinforcement Learning: A Review" [2], which gives us a deeper insight to manipulation with multi-fingered robotic hands.

D. A Survey of Multifingered Robotic Manipulation: Biological Results, Structural Evolvements, and Learning Methods

To gain an even better understanding of working with robot hands, we also reference "A Survey of Multifingered Robotic Manipulation: Biological Results, Structural Evolvements, and Learning Methods" [3]. The main contribution of this paper is an increased focus on tactile sensing and hand-centered visual sensing, which helped us determine the button locations with respect to the hand in a more human-like manner.

E. Decentralized Adaptive Control for Collaborative Manipulation

Lastly, we reference "Decentralized Adaptive Control for Collaborative Manipulation" [4]. This paper gives us insight on how our two robot hands can work together. One of our stretch goals was to use two robotic hands to hold strum and play the guitar just like a human would. Having read this paper, we will be able to implement this in the future.

III. Method Overview

A. Designing the model guitar and buttons

Our simulation environment consists of a kuka iiwa 7 arm with an attached wsg schunk end-effector. Additionally, we programmed a table sdf in a temporary directory, and imported a sdf guitar model from Turbosquid. This guitar model has been heavily tampered with. We got into the raw sdf file to make frames for each button to more easily allow for trajectory mapping of the wsg. Additionally, we created prismatic joints to model buttons that actually press down into the guitar. "Prismatic" means that the joints can only move in one direction, and that direction is a straight line, as opposed to a hinge. Our buttons additionally needed added spring physics in order to maintain their position against the force of gravity.

We deliberated for a bit on the best way to populate the environment with the guitar. Ideally, we didn't want to weld the guitar to a frame in the world, since that was unrealistic to the actual nature of the guitar hero controller. In the end, we made the guitar weigh a great deal, in order to avoid the effects of the end-effector violently pushing into it. The guitar hero controller only moves slightly.

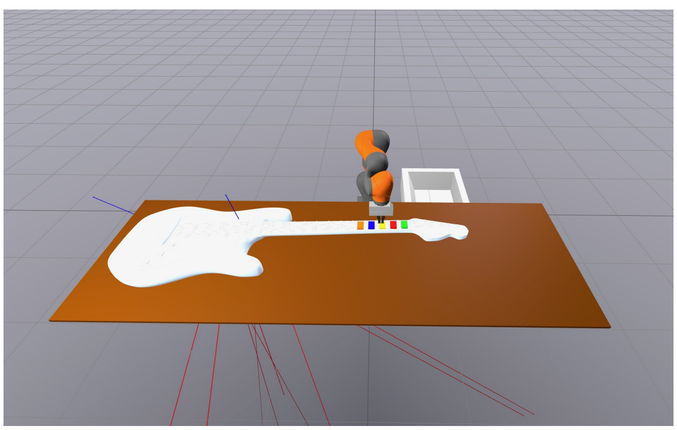

Fig. 1. This shows the simulation environment

B. Determining trajectories and poses that match all possible game hand positions

All possible button-to-button trajectories for each finger are pre-computed. To do so, we utilized the fact that most of guitar hero doesn't require more than two buttons to be pressed simultaneously. Consequently, we will be able to play most of guitar hero with only the 2 grippers on the wsg hand while still satisfying the constraint that each gripper will press at most one button at any given time. We only have 5 buttons and 12 different ways for them to be pressed, either one by itself or combined with either the button next to it or 2 buttons away. We already know where each button is and used the fact that we can change the distance between both grippers in the wsg hand to create every possible pose the hand may be in. For pressing single buttons, we would create a pose where the hand is fully closed and directly above the button. For two buttons being pressed at once, our hand would find the middle point between the two buttons and would either be half way open if they were next to each other or fully open if they were 2 apart. To map trajectories, we can input a list of number, each number being 1-5, and our robot would be able to press those buttons in the correct order, originally having the same amount of time between each press. The issue with this is that guitar hero songs aren't evenly spaced between notes. To fix this we then created an algorithm that would take in the amount of time between each button press and would create a list of the times that we would combine with the poses using a linear Piecewise pose, making the final robot be able to play the correct notes at the correct time.

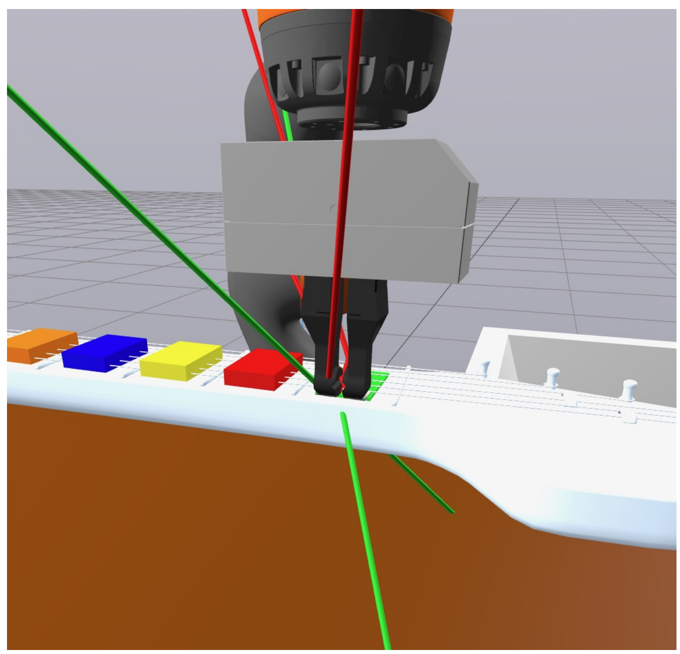

Fig. 2. Here shows the wsg hand pushing the button along with the forces from both the hand and button

Trajectories executed by the fretting robot are based on the current button state and the previous button states. These states are encoded as 2 digit numbers, the tens place being the lower button on the guitar being pressed or 0 if not pressed and the second digit being the second button being pressed. Then, to determine the trajectory, the fretting robot simply looks at the current state for the hand and the next state, and maps this to the appropriate trajectory.

C. Perception Pipeline

For perception, we use color segmentation with the help of the OpenCV library in python. First, we convert our RGB images to HSV format. Then, we use 5 separate predetermined hue ranges (one for each chord) and look for pixels that lie in those ranges. Then, we find contours around the points lying within a valid range and draw an enclosing circle around each contour. If the size of the circle is consistent with our expected chord size, we identify it as a chord. Finally, we look at the position of each chord, and if the position of a chord is located within the strumming plane, the robot is instructed to play the corresponding note.

IV. Evaluation

A. Measuring Performance

We measured the success of our robot by simulating songs that exist in guitar hero games we own and comparing the success vs human subjects. Success was measured based on the percentage of the notes hit as well as how our robot compared to human subjects. Our human subjects range from those who have never played the game before to those that are very skilled at the game. Shown below is our setup for examining guitar hero performance amongst people.

Fig. 3. Setup for examining guitar hero performance amongst users. Here, guitar hero game is being played on the left monitor. The person shown in the figure is holding the "remote," which is in the shape of a typical guitar.

In our evaluation, we found that Guitar Hero Bot was able to outperform the average person on each difficulty. Our robot was able to score 100% every time on both easy and medium difficulties. Unfortunately, it didn't reach this number on hard mode, but it was still able to average an impressive 95%. On the other hand, the average scores for the people we tested were 87% on easy, 72% on medium, and 66% on hard.

B. Challenges

There were many challenged that we faced when working on this robot. To start, it was very difficult creating the buttons and keeping them to remain on the guitar. Throughout our tests, we knew we wanted to make visual buttons in the simulation

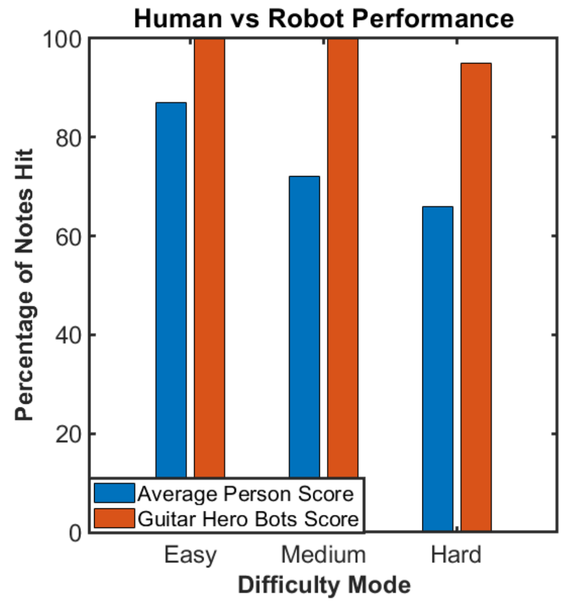

Fig. 4. Bar Chart of our results. Both human and robot performance were analyzed on easy, medium, and hard difficulties. Human performance is shown in blue, and robot performance is shown in red. For all three difficulty levels, the Guitar Hero robot out-performed people.

but kept failing to either put them on the guitar or to get them to remain where the guitar is. We overcame this by creating multiple frames directly on the guitar and attaching the buttons to their respective frames. Once this was done we had no issues with the buttons.

There were also many troubles with the Perception Pipeline. To start, it took many tries to create the hue ranges for the different colors. This was a quick fix but still took up a lot more time than expected. Also, it was very difficult for the perception pipeline to pick up when multiple notes were played at once. To fix this, we changed the location range of where we looked and solely looked at the strumming range so that when multiple notes come at the same time it would know to play both together.

V. Discussion and Future Work

Guitar hero bot demonstrates an impressive ability to master virtually every song with a level of precision that surpasses the average human player. Its proficiency, however, falls short when compared to highly skilled human players who possess intricate techniques and more dexterous movements. The robot's limitations are evident in its use of only two fingers and its slightly slower transitions between notes.

One could expand on the work that we have done by using reinforcement learning to adapt to new button setups. Additionally, if this system where integrated into a robot body, the ability to hold the guitar would become critical. As it stands, we assume that the guitar is bolted to the table/surface. We are also exploring the idea of using an allegro hand to carry out more dexterous movements and to be able to play the 3 note chords that appear within the game.

CONTRIBUTIONS

Ryan worked on implementing the button pressing on the guitar and created algorithms that allowed the group to easily input songs from the guitar hero games into the simulation to watch how the guitar would perform. He also added a way to make the times line up to the actual song.

John worked on creating the model of our environment. He added the Iwwa and the wsg hand, the guitar, and created the hydroelastic springs that acted as the buttons on the guitar.

Noah worked on creating the perception pipeline to allow us to view guitar hero and the guitar so there are no need for predetermined button locations or predetermined notes and timings.

We all worked together to create the video and the write-up.

ACKNOWLEDGMENT

Thank you to Prof. Tedrake, Nora Jackson, and all who helped in the development of this project.

REFERENCES

[1] V. Kumar, E. Todorov, and S. Levine, "Optimal control with learned local models: Application to dexterous manipulation," in 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden: IEEE, May 2016, pp. 378–383. doi: 10.1109/ICRA.2016.7487156.

[2] C. Yu and P. Wang, "Dexterous Manipulation for Multi-Fingered Robotic Hands With Reinforcement Learning: A Review," Front. Neurorobotics, vol. 16, p. 861825, Apr. 2022, doi: 10.3389/fnbot.2022.861825.

[3] Y. Li, P. Wang, R. Li, M. Tao, Z. Liu, and H. Qiao, "A Survey of Multifingered Robotic Manipulation: Biological Results, Structural Evolvements, and Learning Methods," Front. Neurorobotics, vol. 16, p. 843267, Apr. 2022, doi: 10.3389/fnbot.2022.843267.

[4] P. Culbertson and M. Schwager, "Decentralized Adaptive Control for Collaborative Manipulation," in 2018 IEEE International Conference on Robotics and Automation (ICRA), May 2018, pp. 278–285. doi: 10.1109/ICRA.2018.8461263.

[5] seongho-yeon. "Playing-Piano-with-a-Robotic-Hand." Jupyter Notebook, October 18, 2023. https://github.com/seongho-yeon/playing-piano-with-a-robotic-hand.