Robotic Finger Hardware and Controls Design for Dynamic Piano Playing

by

Ruben Castro Ornelas

Submitted to the
Department of Mechanical Engineering
in Partial Fulfillment of the Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

at the

Massachusetts Institute of Technology

May 2022

Signature of Author: ____________________
Department of Mechanical Engineering
May 6, 2022

Certified by: ____________________
Daniel Gilbert
Technical Instructor in Mechanical Engineering
Thesis Supervisor

Accepted by: ____________________
Kenneth Kamrin
Associate Professor of Mechanical Engineering
Undergraduate Officer

Robotic Finger Hardware and Controls Design for Dynamic Piano Playing

by

Ruben Castro Ornelas

Submitted to the Department of Mechanical Engineering
on May 6, 2022 in Partial Fulfillment of the
Requirements for the Degree of

Bachelor of Science in Mechanical Engineering

ABSTRACT

Robotic manipulators have traditionally been designed with the industrial world in mind. High degrees of freedom (DOF) robotic hands are starting to enable a new section of tasks like grasping unfamiliar objects with complex shapes, in-hand re-orientation, and even solving Rubik's cubes. This work explores a new kind of task: playing piano. Unlike most common robotic hand objectives, playing piano requires much higher accelerations and control over velocity rather than position. Proper velocity control allows us to play the piano like professionals, with different music dynamics, allowing us to play soft and hard depending on the emotion of the notes being played. To understand the critical factors, constraints, and next steps to creating a full robotic hand for dynamic piano playing, I designed and built several one degree of freedom fingers capable of the high accelerations and velocity control needed for playing the piano using actuators that are appropriately sized for a 10+ DOF robotic hand. With this setup, I was able to consistently play individual notes within 1dB of their respective desired output volume, ranging from -25dB to -4dB, at song speeds of up to 129 beats per minute (BPM). This range is equivalent to playing pianissimo (very quiet) to fortissimo (very loud).

Thesis Supervisor: Daniel Gilbert
Title: Technical Instructor in Mechanical Engineering

Introduction

Robots from science fiction have started coming to life. We have robots that can make us pizza, robots that can bartend for us, robots that build rocket ships, and even robots that can do backflips. The robotics market is expected to quadruple over the course of the next seven years [2], and robots like Boston Dynamics' Spot have started becoming household names.

Most of this growth has been fueled by a need to increase manufacturing productivity and throughput, but these innovations have led to adaptations in other fields. One such area is that of entertainment robots, with the classic case being Disney. Disney theme parks have brought their characters to life using animatronics. Their latest invention, the A1000 animatronic, is capable of making every face gesture you would like, talk, use their fingers, turn around, and much more. This animatronic is very customizable, allowing it to be used for many different characters by changing its costume and face.

Figure 1: Internals of Disney's A1000 animatronic (left) and it being used with a costume on top (right) [5]

Disney's robots aren't just limited to stationary robots. For the Avengers Campus at Disneyland California Adventure park, their engineers have created a robotic skeleton which can adjust its limbs to perform many stunts mid-air.

Figure 2: Disney's custom animatronic with a spiderman suit in the middle of a maneuver during one of its performances (left). Internals of the spiderman robotic skeleton while in a test (right). [3] [4]

Compressorhead is an animatronic rock band created by Berlin artists that has performed live in front of thousands of people. There is no trickery. The robots use pneumatics to move their fingers and arms in order to play real instruments. They even blink and head bop to the beat of the song.

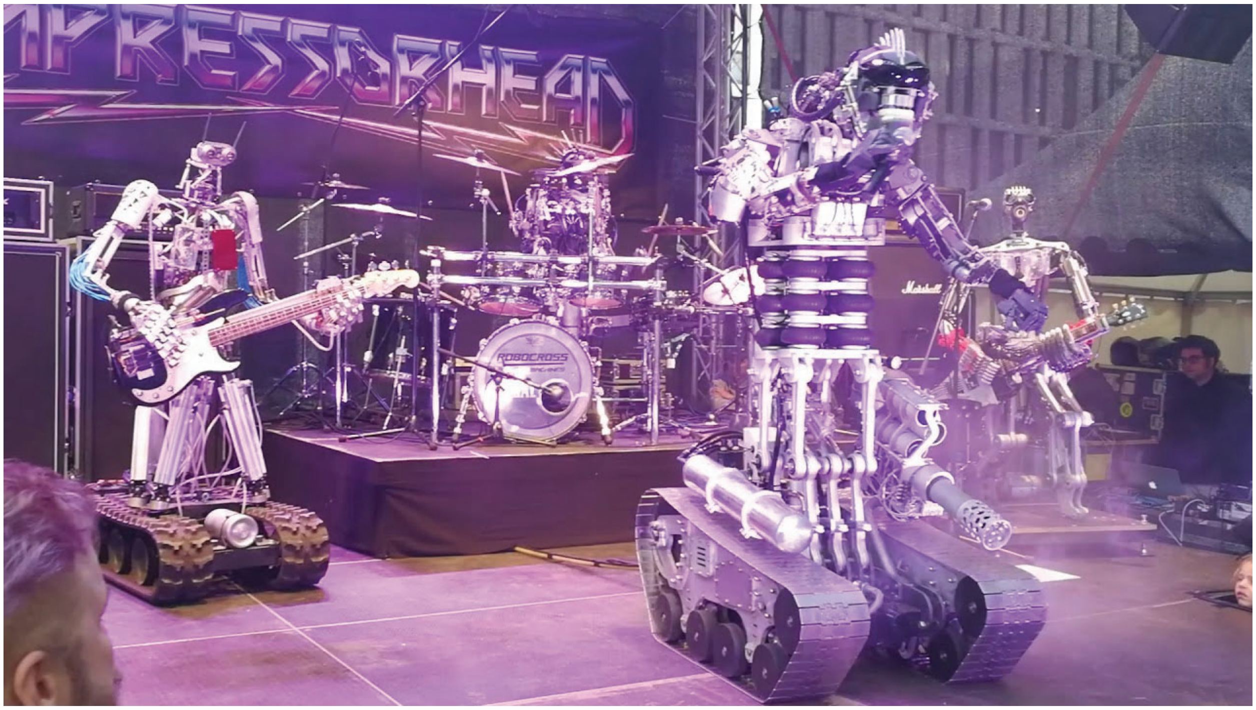


Figure 3: Compressorhead performing at Futurium 2017 in Berlin. [6]

No list of cool robots to watch is complete without Boston Dynamics' latest robot demo videos. Boston Dynamics has become a household name in the US and their videos garner hundreds of millions of views. While their robots were not created with the intention to excite an audience, they have succeeded in capturing the country's attention. There was even a Samuel Adams Superbowl commercial which was recorded in the Boston Dynamics headquarters and features some of their most prominent robots. Other feats include: Spot(s) dancing to Rolling Stones, BTS, and Bruno Mars, Atlas performing backflips and parkour, Atlas dancing to Do You Love Me by The Contours, and more.

Robotics' growth has fueled a new field of robots made to entertain. This work seeks to uncover more information on what it would take to make a kind of these robots: robot pianists.

The Case Study: Piano fingers

Piano playing makes sense as one of the first test cases to explore for music robots as it's one of the most popular instruments and can work well as a standalone musician. Most critical to piano playing are fingers.

Human fingers are incredible; Think of the best pianists -- they sometimes use their whole chest to get enough force to hit the key and are still able to play 6 different keys at once and move their fingers tens of times a second without missing a single note. The fingers are the most critical part of a robot meant to play piano since that is the point of contact with the keys and also is where the most research work needs to happen. There exist robot arms with few degrees of freedom that can place end effectors within 0.001", but there are no robotic hands with 15+ DOFs that can dynamically play Beethoven. Traditional robot hands and controllers for them are meant to be slow and careful since this fast timescale is not of the essence when it comes to most research objectives such as pick and place, household activities, and solving rubik's cubes. The struggles with creating a robotic hand for piano playing application is the high number of degrees of freedom, the high bandwidth needed (being able to play fast), and the forces that the fingers have to be able to exert . The high number of degrees of freedom makes it hard to develop high level controls algorithms, as computation gets exponentially harder with every new DOF. The high bandwidth and high forces require a lot of power (around 1W per actuator) and are a big hardware limitation as ideally you have motors that are compact enough to fit 15+ into an average sized human arm. On the positive side, playing piano requires less positional precision than traditional hand applications, you can be anywhere on the key as long as you are pressing the key. Laterally, that gives about 1mm of freedom from each side.

If we can create a finger design and controller that can play fast and strong while using compact actuators, then we solve the largest unknown problem to be able to physically play the piano dynamically and can be integrated into a larger system which includes robot arms or a linear rail. Here we create a simplified, proof of concept version of a finger to demonstrate that piano playing with robot parts is possible, and also serves to understand the critical factors needed to build the next version.

Background

Music Dynamics

Dynamics in music refers to the volume at which each note is played. If there is a sad moment in the piece, then perhaps each note will be played soft and gently, but when it picks up, the notes could get played intensely and aggressively. That variation and range is what gives emotion to music. It's not just hitting the keys at the right time, it's hitting them with purpose and feeling. Being able to play with proper music dynamics is one of the differences between a novice and professional pianist.

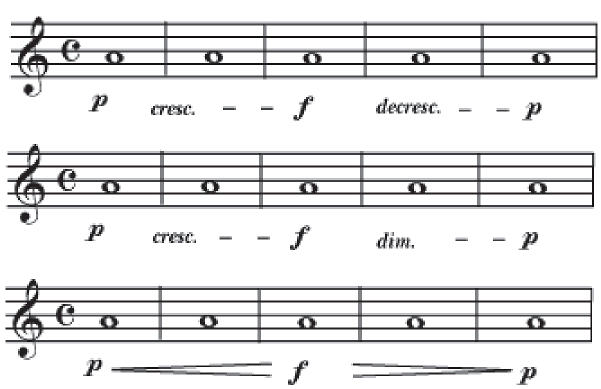


Figure 4: Sample basic sheet music demonstrating music dynamic notation

Typically, these ranges are written in sheet music with markings that represent the levels at which to play the notes. "p" or piano represents soft playing, while "f", or forte, is playing loud. There are many prefixes or suffixes to adjust the softness or loudness, but for the sake of this work, we will mostly deal with opposite extremes: fortississimo (very, very loud) and pianissimo (very soft). This will give us a good breadth upon which to test our robotic fingers.

Robot Piano Players

Two robot pianists to see as case studies are here:

Teotronico

Figure 5: Pictures from performance at Shanghai Symphony Hall.

Teotronico has a plastic finger for each key of the piano, letting it play incredibly fast, but without music dynamics. Each note is played exactly the same, leading to a robotic or monotonous melody. At the same time, Teotronico can move his head, smile, wink, and even talk. Despite its limitations, its novelty and its personification have gathered it a following. Teotronico has been on stage with established pianists, and you can even book an act with it via the internet.

Steinway self-playing piano

Self-playing pianos, or “player pianos”, are an interesting case study because they can reproduce any piano sound made by real piano players (and even pieces that are impossible for humans to play) using a real piano. While the technology details are closely guarded by private companies, some information from public channels show that they primarily use solenoids with advanced motion control to create the music dynamics of each note. Because the solenoids are internal, the machine system can be fully actuated and much easier to execute, as there is one for every key. Having this same level of delicate control while being external is a hard problem.

So let's take a look at what causes this problem of creating hands and fingers that can dynamically play the piano delicately as well as loudly.

Piano Physics

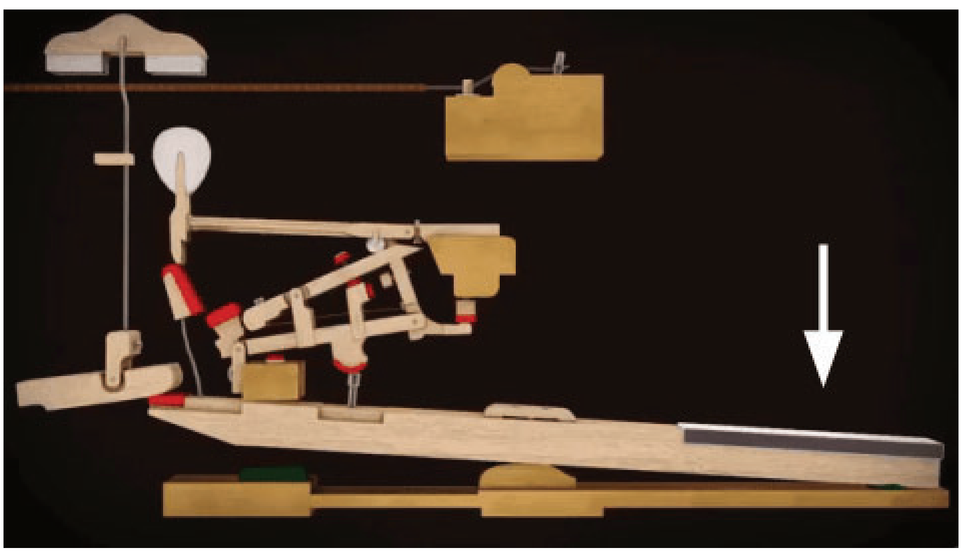


Figure 6: Internals of a piano, looking at a cross section of an individual key. Important to note is that the key doesn't directly drive the hammer, it goes through a series of levers to move the hammer at about 10x the speed of the key. The key itself only travels 1 cm at the furthermost end. [1]

To numerically understand the forces and dynamics of all of these moving pieces, we must find out what the inertias of every component are and how they add up to the effective inertia of the system, as well as find the overall gravitational forces acting on the components and the accelerations.

To find the inertias, we model the elements of the piano and solve as if they were rectangular prisms rotating at their respective pin points. Because some components rotate faster than others due to varying transmission ratios (i.e. the hammer travels about 10x faster than the key), those components have a greater effect on the effective inertia of the system.

$$I_a = I_b * (a/b)^2$$

This increased effect is equal to the square of the transmission ratio (a/b) . This means that the hammer which is geared to rotate 10x faster than the finger will have its inertia reflected 100x. After analyzing every component, we see that the majority of the effective inertia of playing a piano key comes from accelerating the hammer. In total, we get an estimated effective inertia of the system of 0.0056 $kg\ m^2$.

By measuring a piano with a force gauge, we also ascertain that the key requires 1.8 N of force to be held at a constant position throughout the stroke, this means 1.8 N of force is constantly pushing against the finger.

With inertial and gravitational forces in mind, we can now take a look at the acceleration of the key to find the final torque required. To do this, slow motion footage of an expert piano player was gathered, and we found a constant acceleration of 1.88 m/s^2 while playing pianissimo, and a constant acceleration of 5.01 m/s^2 while playing in fortississimo.

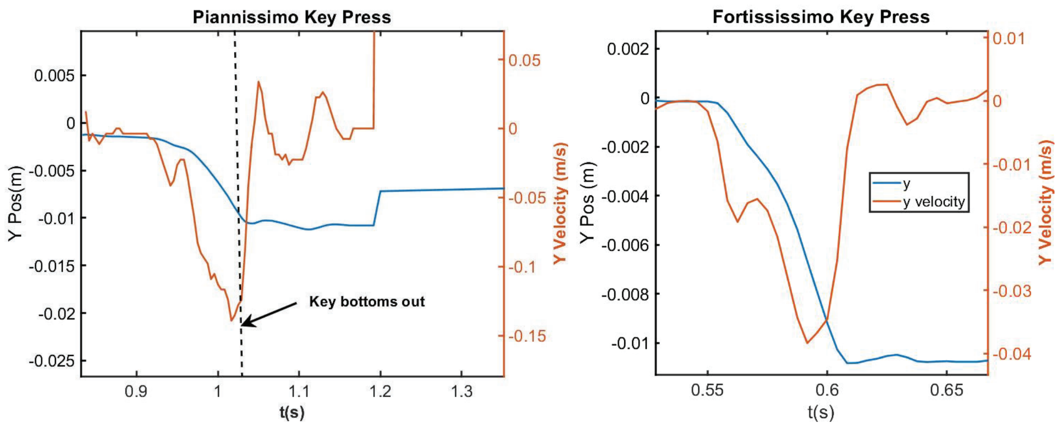


Figure 7: Slow motion footage analysis of individual pianissimo and fortississimo key pressings analyzed by hand using Logger Pro. The maximum velocity of the measurements we made were in fortississimo. This velocity approached 0.04 m/s under near constant acceleration of 5.01 m/s^2.

Finally, using the accelerations, gravity forces, and inertia, we can find the torques and power required from a motor with a finger of average human length, approximated at 8.9 cm.

$$\tau_{motor} = I_{eff} * \alpha_{motor} + \tau_g$$

Based on our estimated effective inertia (I_{eff}) found above, measured motor angular acceleration (α_{motor}) and measured gravitational torque (τ_g), we can calculate a final torque on the motor of 0.191 Nm while playing fortississimo, and 0.144 Nm while playing pianissimo. The difference between the two is entirely due to inertial differences since the gravitational torque influence is constant in any case. By multiplying each torque by its respective angular acceleration, we can obtain the power required by the motor. For fortississimo, this ends up being 0.64W, and for pianissimo, 0.24 Nm.

All of these key variables can be summarized below:

Variable name	Value
Key Stroke	1.0 cm
Key Effective Mass	0.18 kg
Effective Inertia	0.0056 kg m^2
Effective Gravitational Torque	0.11 Nm
Pianissimo Acceleration at Finger Tip	1.88 m/s^2
Fortississimo Acceleration at Finger Tip	5.01 m/s^2
Pianissimo Torque Required	0.144 Nm
Fortississimo Torque Required	0.191 Nm
Pianissimo Power Output	0.24 W
Fortissimo Power Output	0.64 W

Table 1: Summary of critical variables for design

With these findings, we can choose an appropriate actuator for our application.

1 DOF Finger

With all of the physics in mind, this thesis presents a hardware prototype that can be used to demonstrate the feasibility of a controller for piano playing and to learn what key factors are necessary for future iterations**.** While the ideal design involves 3+ DOFs for each finger, we only need 1 degree of freedom in order to test the response of an individual key. We focus here on a 1 DOF finger to prototype controllers and understand other important variables.

The requirements of this 1 DOF mechanism are threefold. (1) To adequately accelerate the key to fortississimo within typical keystroke movement. (2) Understand reasonable micro-controller and actuators for the final version. (3) To be easily moved and rearranged for testing different keys and for demos.

Motor choice

For actuating our system, we could get massive motors and the sheer power output would allow us to very quickly and with high force play piano, but in order to satisfy requirement (2), we must look for motors that are small enough that 15+ of them can fit into a human arm-sized container. From Table 1, we can see that we need a motor that can regularly supply 0.64 W. Additionally, because we do not need to move faster than 32 RPM to play fortissimo, we can optimize for the amount of torque we can output in that speed range.

The Polulu 25D series brushed dc motors are a great choice for this. They come in a wide range of gear reductions, can supply up to 7.6W of power, are of appropriate size, and can be easily used with hobby level electronics. While many bio-inspired applications ask for lower gear reductions so as to have a lower motor inertia and be proprioceptive [12], our application does not require precise torque control and backdrivability, rather, it requires precise speed control. This means we can get away with as high a gear ratio as needed. Let’s take a look at the 75:1 gear reduction motor curves.

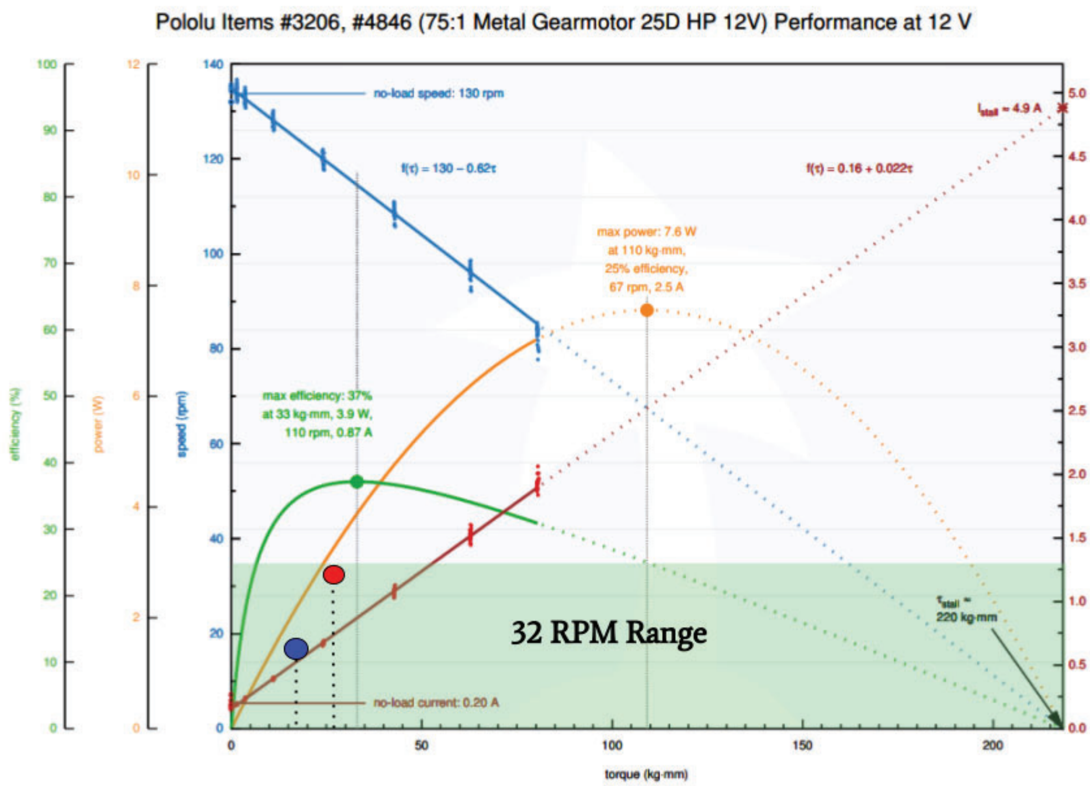


Figure 8: Motor curves for our chosen motor. In red we can see the point at which we operate while playing Fortississimo, and in blue we can see the point at which we operate while playing Pianissimo. The green area represents our total potential desired operating range. We do not need to go faster than 32 RPM, but it may be desirable to use a higher torque in order to accommodate a slower controller or any other deficiencies.

Using this motor allows us to run at ~30% maximum recommended voltage and still be able to meet our harshest piano playing demands. Even though this motor fits within our sizing constraints, we would be able to be even more compact if desired.

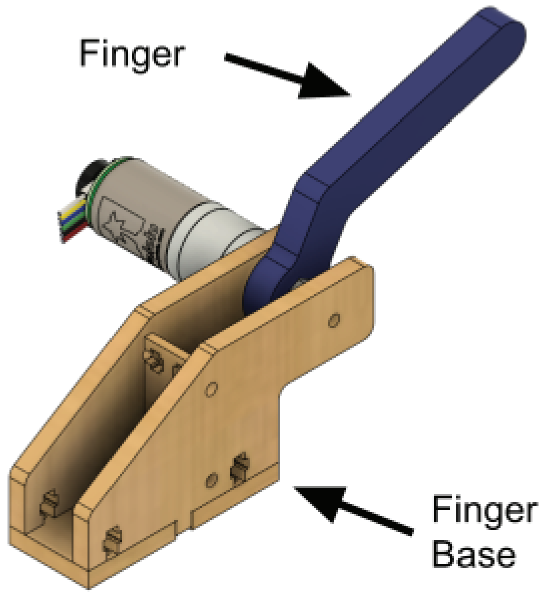


Figure 9: 1 DOF Finger CAD. In beige are MDF Laser Cut pieces joined together using captured nut-style joinery. In blue is a 3D printed finger that is fastened to a Polulu 5mm Shaft Hex Motor Adapter.

The base of the finger is made out of joined laser cut MDF pieces. You can see more pictures of the design in Appendix A. The base has two separate magnets press fit onto them so that the finger can be quickly snapped onto a metal base, and can be easily detached and reattached somewhere else to test a different key. This flexibility allows us to satisfy our mobility requirement (3).

The finger itself is made out of PLA using a SLA 3D printer. The dimensions approximate that of a human finger, 8.9 cm, and are designed to be able to hit the black keys, even at an angle. This finger is directly driven by the motor. While this would be unacceptable in a final prototype since it makes the fingers very bulky, we can get away with directly driving the finger for the sake of the proof of concept.

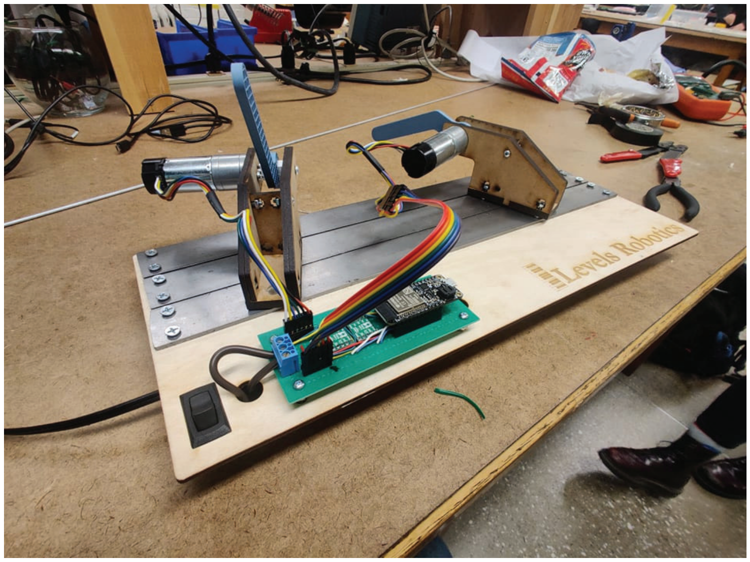


Figure 10: Picture of the piano bed with two fingers magnetically attached to the steel bars.

The bed is laser cut out of ⅛" thick balsa wood, and there is 2x4 wood underneath it for additional stiffness support. The bed has three strips of steel bar for the fingers to magnetically attach to. The bed is wide enough to be able to move 3 fingers on it comfortably to hit any set of three keys. There are also mounting holes for electronics so that everything is consolidated into this one board that can be easily moved around.

Electronics

To control our system, we used a Teensy 4.0, which at the time of this writing is the fastest micro-controller on the market [10]. This speed will be necessary to be able to control a few fingers at once at the update frequency that we require. According to our slow-mo data shown previously, we only have ~60-100 ms to adjust to the desired velocity during the piano key stroke.

We used the embedded encoders that come with the Polulu 25D series motors. They have 48 counts per revolution. Because the gear ratio we chose for our motor is 75:1, this means we get a total of 3592 counts per revolution of the finger. At its fastest while playing fortississimo, our piano robot finger will need to spin at 32 RPM, or 0.5 RPS. During that time, we will receive 1915 encoder updates per second. This sets the limit to the speed we can run our controller since we cannot modify our output without receiving enough inputs to respond properly.

Controller

The goal of our system is to reproduce a series of notes with varying duration (quarter notes, half notes, etc.) and varying music dynamics.

Transcription from sheet music to robot commands happens offline. Each finger's series of notes are transcribed into an array with the music dynamics with which to hit the key and its beat duration (quarter note, etc.). These values are then passed through a MATLAB program which converts the note durations and music dynamics to actual speeds in encoder counts per second and durations in milliseconds. The MATLAB program also has the option to adjust the durations and speeds for legato which produces a smoother transition between notes.

The array is then fed into the Teensy using Serial communication. Once running, code in the Teensy cycles through each desired speed in its command array and goes through a cycle of three primary control modes: 1. Velocity control mode to reach the desired velocity in the finger and key, 2. Holding torque mode to sustain the finger at the key's bottom without exerting unnecessary force, and 3. Position Control to quickly back off the key and move to a position ready to hit the next note.

These modes are cycled through a heuristics based, empirically tuned trajectory. Varying speeds and durations were played on the 1 DOF finger hardware on a real piano and the following trajectory was followed at each cycle:

1. Velocity Control: Constant velocity set point, shifts to next mode once it detects its position is within 5 encoder counts of the key's bottom (0.07cm) in order to prevent from over stalling the motor.
2. Hold constant 3.0 V until current time = next beat - time to position - time to press. Where time to position is a constant 75 ms which was determined by empirically testing the rise time of the PID position controller, and time to press is determined by the next travel distance over the next key pressing desired speed.
3. Position control to optimal position ready to strike the next key (the harder the next key pressing, the higher off the key it will go in anticipation). This is based off a linear interpolation between an empirically tuned range (from 1.4cm off the key, to 0.2cm above the key).

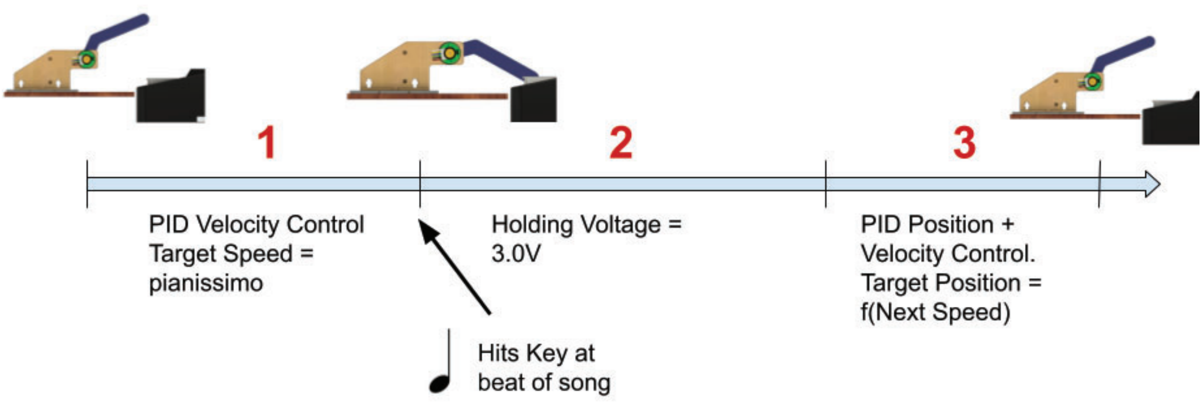


Figure 11: Block diagram of position + velocity controller used for our prototype. When in mode 1 , we are not using the position control portions, and when in mode 2, we are only feeding a constant voltage. B. Graphic diagram to show the trajectory and what the

position of the finger is at the beginning and end of each control mode. At the end of mode 1 is where the beat hits. It's important to know that these modes cycle and the beat always hits after mode 1. The trajectory is created to ensure that the finger strikes the bottom of the stroke exactly at the beat.

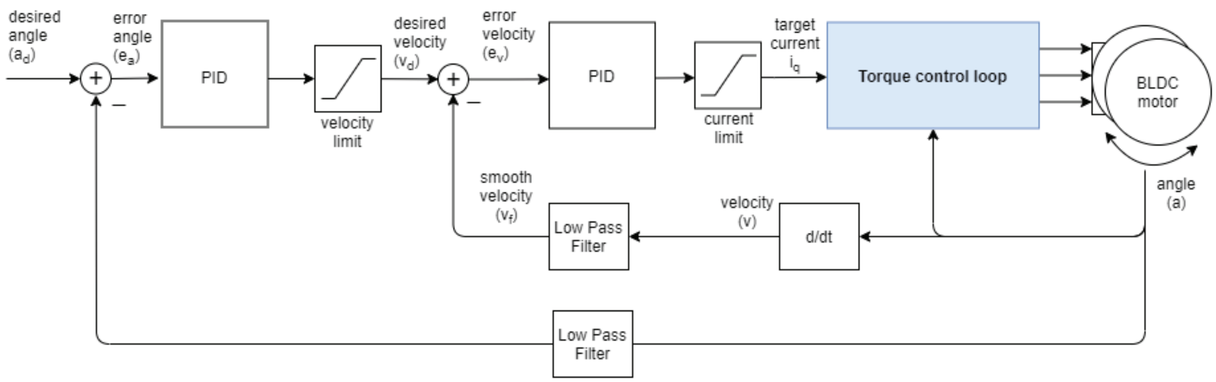


Figure 12: Block diagram of our position + velocity control loop.

For Velocity control, we use a PID controller with feedforward torque control. The feedforward torque takes in the PID controller's decided torque output to the finger and automatically accounts for back-emf, gravity loads and the change in inertia before and after the finger collides with the key. To calculate velocity, there is a separate loop that runs at 100 Hz which adds up the number of encoder counts which have passed and divides by the time step. Unfortunately, because the update rate of the encoder is the same order of magnitude as that of the controller and velocity loop, sometimes there is only one count which has passed, and at times even 0. So if the finger is moving slowly, the velocity feedback will vary between 1 count/10ms and 0count/10ms. In order to account for this, we run this input through a low pass filter which smooths out the signal and leads to a more accurate reading. This lowpass filter, though, also adds a delay in the sensing of velocity.

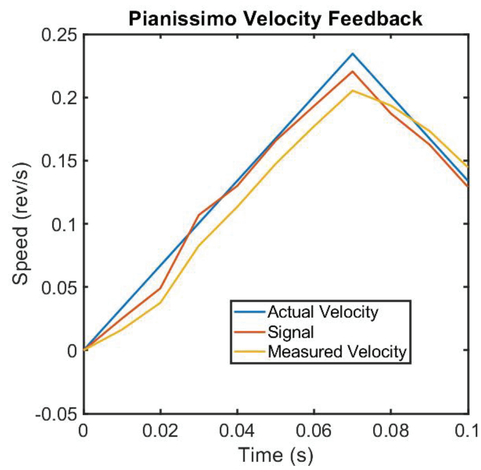


Figure 13: This graph shows a sample potential velocity profile along with a simulation of what the signal could look like, and how the low pass filter reads it. Most important to note is that the low pass filter is able to filter out discrepancies, but it's at the cost of ~8ms of delay, almost an entire controller cycle.

While this delay is significant (8 ms vs 100ms of the stroke), it is still usable because we are not constantly changing the desired speed during the stroke, there is only one desired speed, so once we reach the desired speed we are no longer "behind".

In order to hold the key, we give a constant voltage of 3V to the motor. This was enough to actively hold the key and also to ensure the finger stays at the bottom of its stroke.

When coming back up, we wrap the velocity PID loop with a position PID loop. Our heuristics based trajectory takes into account what the speed of the next key is, and will move higher if we need to accumulate more speed in preparation for a strong note, or will only move slightly above the key if we are playing softly. This is a simple linear interpolation between immediately above key and 3 cm above.

Because there is no proper optimization done here, this is where we expected to see sub-optimal results, but we should still be able to perform well enough to play a song.

Test Procedure and Results

In order to judge the performance of our robot according to our requirements, we must see if we have been able to accelerate the finger fast enough to play the desired speeds, and also if it is consistent enough. We ran a series of tests on the finger by having it play at three different target speeds: 4.2, 10, and 31.7 RPM. This covers the full range of tones that we are hoping to receive.

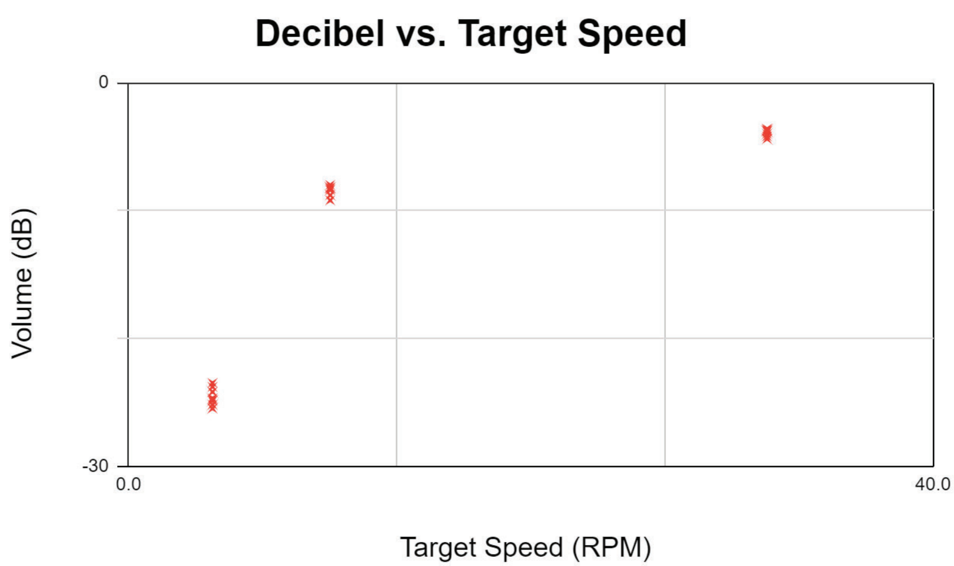


Figure 14: Scatter plot of volume of key pressings and the target speed. Each speed has 10 data points. At each speed, the speeds are within 1 dB of the average.

With this test, we can visually see each target speed lead to a distinctly unique volume range. At each speed tested, the volume outcomes are within 1dB of the average, with the volume range getting smaller as the speed increases. For the 31.7 RPM target speed, we saw an average volume of -3.80 dB with a standard deviation of 0.28 dB. For the 10 RPM target speed, we saw an average of -8.44 dB with a standard deviation of 0.42 dB. For the 4.2 RPM target speed, we saw an average of -24.58 dB with a standard deviation of 0.69 dB. These points along with their spread are far away from each other. Each speed can clearly be seen as distinct, and we can create a method of interpolation to pair up volumes/music dynamics to speeds. We have successfully attained repetition.

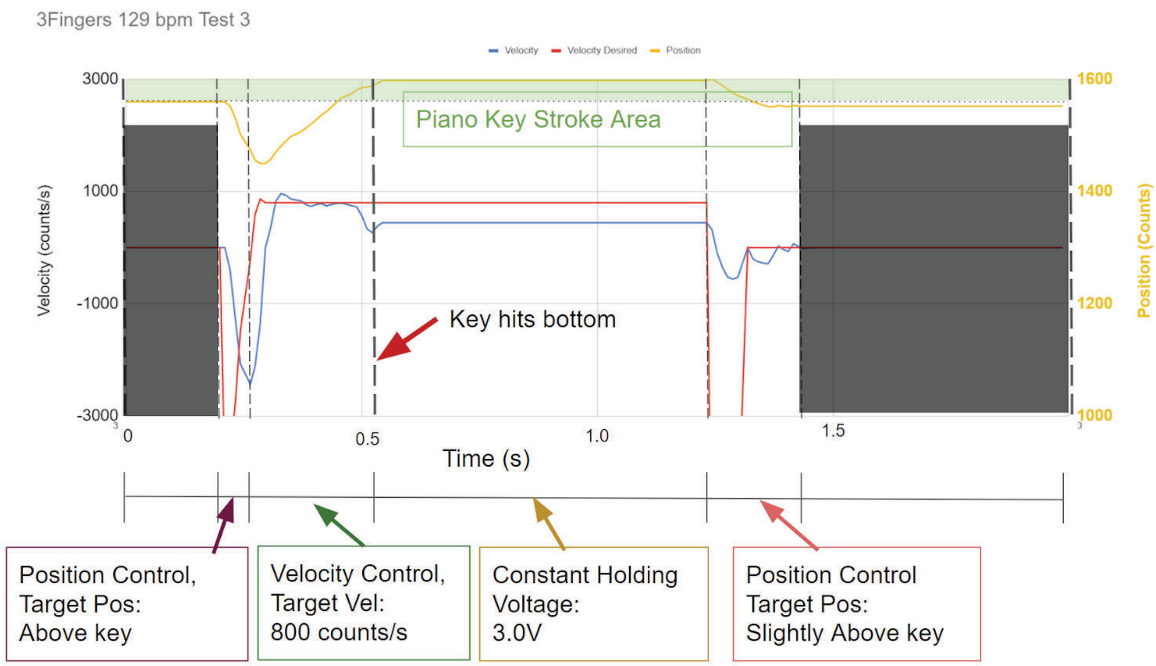


Figure 15: Controller response from a sample key pressing cycle. When the yellow line hits the green area, that's when the finger collides with the key, and when the yellow line reaches the top, the finger hits the hard stop. To understand this graph, it's important to follow the actual velocity signal and track it throughout its stroke. The finger can reach that speed before collision, and is able to maintain after the collision.

Another useful test to categorize our performance is to run a cycle and look at the controller response. From Figure X we can see the controller is capable of accelerating to the desired speed of 800 counts/s (13.3 RPM) in 60 ms. In this case, that happens before the finger even hits the key, and the controller is capable of maintaining the speed after collision and throughout the stroke. This successfully verifies requirement (1). We are reaching the top speeds we are looking for. In other demonstrations, we have seen similar responses from target speeds of up to 32 RPM. While this means we can reach our target acceleration within a tolerable time, it also means that our trajectory is not ideal. In order for our trajectory to be ideal, we would spend as little time coasting at the target speed in the air. The ideal response is having the finger be constantly accelerating. To do this, we can shorten the height at which the finger starts prior to the stroke. This gives us less air time, but it may sacrifice consistency. In order to truly minimize air time, we need to implement higher level controls algorithms.

Figure 16: Side view shot of one hand pattern for the demo. Critical to the demo is the quality of the audio and video, so this was recorded on a Canon 70D and using a Shure

MV7 microphone. You can see all three fingers on the keyboard, with some of them pressing a key, and the middle finger in motion, about to press a key.

All of this work culminates in the demo. This is perhaps the most important part. The song of choice is Will.I.Am's "This is Love" ft Eva Simons. Three fingers were built and laid onto the bed in different configurations to create different hand patterns. The sheet music was transcribed into pitches and durations for our MATLAB program with the help of a professional pianist to understand what the appropriate music dynamics are for each note. With each hand pattern, we recorded a 30 second clip of the notes from the song that correspond to the keys in that pattern. In total, 14 setups were used to address the melody and the bass. By overlaying these clips in video editing software, we can see the voices align with each other and add up to the first 30 seconds of the song.

Figure 17: Frame from demo video with several shots from a top view. From this frame you can understand how the hand patterns were different and rearrangeable using the

magnetic bottoms. Each hand pattern can hit up to three keys. By overlaying several of them, we can hit as many keys as desired.[13]

When you hear this overlaid final demo, you can clearly tell the difference in the music dynamics throughout the piece. Certain elements are stronger and more emphasized, pass-through notes are light. The controller also does a good job of keeping tempo. Despite there being 14 separate clips, the key pressings between them align properly. You can find this demo in reference 15.

Discussion

This work takes a step in using modern hobby electronics to control a robotic finger and use it for playing the piano. We created a demo which can be used as an example for attracting interest in the project, and we shall talk about what is critical for future development, as well as key variables to keep in mind.

When evaluating piano robots, the following metrics have been useful to evaluate the controller + hardware performance:

- Rise time of key desired velocity
- Air time
- Accuracy over spectrum range
- Performance at low speeds due to motor deadband

While the 1 DOF system allows us to play keys and can be mounted in a linear rail in order to play any key, it is impossible to rigidly attach 5 of these together and play any hand configuration on the piano. We need more degrees of freedom, and an effective way to control all of these degrees of freedom.

Hardware

There are a few key lessons from the 1 DOF prototype to keep in mind while developing the next version.

1. Stiffness/slip
    a. Because our fingers are not attached to a human, they needed to be clamped down onto something that was stationary and could handle the loads. In our case, the bed did not line up with the top of the piano, so we had to use books to lift the height. This caused the piano bed to lose its stiffness relative to the piano and it would bend backwards when all three fingers were pressing at the same time with forte music dynamics or higher. Future iterations must look at the stiffness loop between the piano and the fingers.
2. 3 DOF + Thumb
    a. To be able to play the full range of hand configurations of a real person, we must add two DOFs per finger for a total of 3. 3 DOFs approaches human fingers and that's how we can play the most complex configurations. At the same time, the thumb needs to be different from the rest of the fingers. In piano, when one hand plays many keys in increasing pitch order, the thumb usually does a crossover move, which is when it goes under the hand and in front of the index finger.
3. Encoder:
    a. Because our controllers should be running in the high kHz range, we need to have encoders which update at least 10x more than the current ones we have.
4. Controller capabilities
    a. While the Teensy 4.0 was able to run the controllers for the 3 motors concurrently, any more may overwhelm the controller and make it not be able to run fast enough. Additionally, it only has so many ports for extra motors. Because of these, it is necessary

to upgrade to using many of these micro controllers chained together under a master controller.

5. Motor choice
    a. The current motor we have chosen has enough power output, but we may want to switch to brushless motors for higher efficiencies and also motors with less backlash.
6. Transmission
    a. With the 1 DOF system, we did not care about the geometry of the finger, as long as it could play the white and black keys. That is why we could get away with directly driving the finger. But with future iterations, we will want the motors to be further removed from the finger in order to fit all the required actuators in the arm. This will require many developments in transmission. Some current finger systems use cable drives, but those have been shown to have significant wear and tear deficiencies, and cable drives may be too large to be used. Significant progress needs to be made in order to carry the motor loads to the corresponding degree of freedom.
7. Considering series elasticity
    a. The last joint in a human finger acts like a spring. Adding a spring into our system may allow us to conserve energy lost during collisions, meaning that we may be able to bounce back faster with the same sized actuator.

Software

While we were able to transform a simple song into its desired speeds and durations and transfer that to a 1 DOF system, things get much more complex when using more degrees of freedom. In order to be able to properly play a full song, we also need to implement trajectory optimization over long stretches of time. The combination of these means that further development of a piano playing robot requires serious software and controls work. While the hardware developments outlined above are requirements in order to play hard coded sophisticated music, some of the software developments are optional and serve just to expand the breadth of songs without extensive offline transcription work.

1. Recording/playing real human player
    a. The easiest way to move forward with more complex songs is to record the velocities and fingering patterns of professional pianists through motion capture or other means and have an offline transcriber transform that into arrays. The selection of songs would be severely limited and would require extensive effort to add new songs, but it would allow the hardware testing to continue
2. MPC + Sheet music automatic transcription
    a. MPC trajectory optimization can allow our fingers to choose where on the piano they land and which fingers to use. Pairing this with sheet music, automatic transcription would allow us to remove the manual work of transcribing sheet music to arrays for the controller to understand.
3. Machine learning/dynamic programming for music dynamic intuition
    a. If we add machine learning to understand the emotion that pieces should be played with based on sheet music, we can truly have a robotic pianist that surpasses human ability.
4. Audio/video auto-calibration
    a. The last level up for software is adding cameras and audio feedback to calibrate the robot's movements in real time. Currently, the speeds are hard coded. If we can add a way for the robot to understand loudness and pitch, as well as be able to see the position of the fingers, it can adjust that by itself over time.

Character

These technical feats are necessary to develop the capabilities of the robot. But in order to create a robotic pianist, special considerations have to be made for the aesthetics. In the same way that the piano robot must be able to replicate human emotion while playing, it must also move its body and gest. Because this application of robotics is so public-facing, it needs to have detailed care of its aesthetics. When showing the demo to others, it has not been sufficient to show a single finger actuating. People only get excited once they see the conjunction of them playing together a song that they can recognize.

The aesthetics and technical feats are both requirements to push this project further.

Conclusion

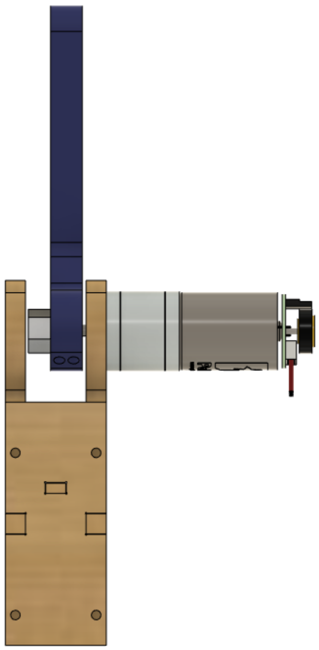
There is enormous potential for music entertainment robots. Disney and Boston Dynamics have introduced a new category of robots, entertainment robots, and there is a ripe opportunity and excitement to expand upon that.

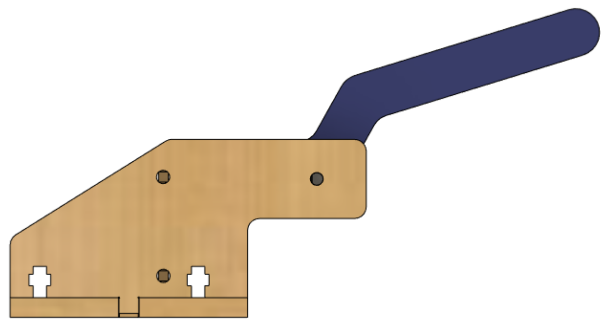
Technology for the music and show industry has been limited to stages for musicals and software for audience retention, but recent advances in robotics allows affordable and doable robots for them. We have outlined a way of using DC motors to consistently hit keys with the desired music dynamics ranging from pianissimo to fortississimo with less than 1dB difference per desired target volume, and we have outlined what it would take to expand upon this project.

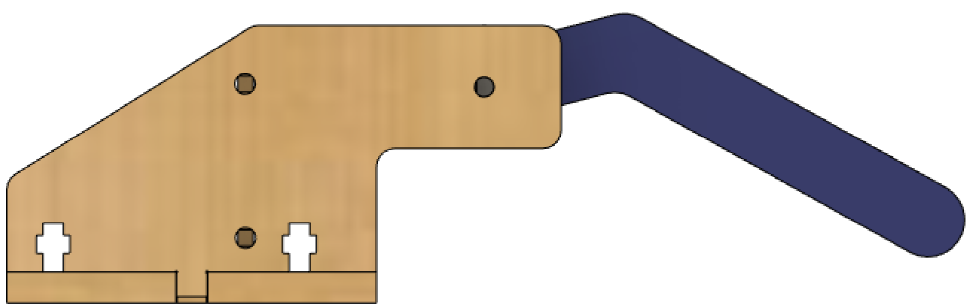
Acknowledgements

Thank you so much to everyone who has supported me in my undergraduate years while at MIT: Chris Mayer, Amanda Mayer, Ed Moriarty, my closest friends (you know who you are) and my fraternity PKT. Thank you to my undergraduate advisor Professor Alex Slocum for helping me figure out what I want to do post-graduation and being a mentor over the years. Thank you to Daniel Gilbert advising me and for helping me push through this thesis. Thank you to Jake Read, my freshman and sophomore year UROP supervisor who taught me how to make stuff and gave me a great kickstart to my MIT career. Lastly, thank you to my family for giving me the values I take with me through life: my mother Raquel Ornelas, my father Jaime Castro, and my brother Ivan Castro.

Appendix A







References

[1] Owen, Jared. “How Does a Grand Piano Work? - Part 1 - YouTube,” March 30, 2018. https://www.youtube.com/watch?v=NDvS2V7HbnY.

[2] Lassig, Ralph, Markus Lorenz, Emmanuel Sissimatos, Tilman Buchner, and Ina Wicker. “Robotics Outlook 2030: How Intelligence and Mobility Will Shape the Future.” BCG Global, June 17, 2021. https://www.bcg.com/publications/2021/how-intelligence-and-mobility-will-shape-the-future-of-the-robotics-industry.

[3] Schaben, Allen, and Taylor Arthur. “Marvel Superheroes Rock Disneyland Avengers Campus: Photos.” Los Angeles Times, June 3, 2021. https://www.latimes.com/entertainment-arts/story/2021-06-02/marvel-superheroes-rock-disneyland-avengers-campus-photos.

[4] “‘Stuntronics’ Bring The Amazing Spider-Man to Life Above Avengers Campus - LaughingPlace.Com,” June 29, 2021. https://www.laughingplace.com/w/news/2021/06/29/stuntronics-bring-the-amazing-spider-man-to-life-above-avengers-campus/.

[5] Walt Disney Imagineering. *A1000 Advanced Robotics | Walt Disney Imagineering*, 2019. https://www.youtube.com/watch?v=5qteMlMAaNM.

[6] Mario Sixtus. *Compressorhead*, 2017. https://www.youtube.com/watch?v=tdfwOf1OjpU.

[7] Shauw, Jeacker. “Dream Wedding---Played by Piano Robot - YouTube.” https://www.youtube.com/watch?v=IwuiOwZrzCM.

[8] “Teotronico.” http://www.teotronico.it/.

[9] “PID Theory Explained.” https://www.ni.com/en-us/innovations/white-papers/06/pid-theory-explained.html.

[10] “Teensy 4.0 - DEV-15583 - SparkFun Electronics.” https://www.sparkfun.com/products/15583.

[11] "TeoTronico and Orchestra - YouTube."https://www.youtube.com/watch?v=5TQUGGcbVaQ.

[12] Wensing, Patrick M., Albert Wang, Sangok Seok, David Otten, Jeffrey Lang, and Sangbae Kim. "Proprioceptive Actuator Design in the MIT Cheetah: Impact Mitigation and High-Bandwidth Physical Interaction for Dynamic Legged Robots." *IEEE Transactions on Robotics* 33, no. 3 (June 2017): 509–22. https://doi.org/10.1109/TRO.2016.2640183.

[13] Ruben Castro. *Albert the Piano Playing Robot -- Test 3*, 2022. https://www.youtube.com/watch?v=HZM4nkR-uTg.