Senior Project Report

Therapeutic Mechanical Horse

California Polytechnical San Luis Obispo Senior Project 2021 - 2022

Aleya Dolorfino
adorfii@calpoly.edu

Zuzanna Dominik
zdominik@calpoly.edu

Cade Liberty
cliberty@calpoly.edu

Peter Phillips
pphill03@calpoly.edu

Luke Watts
luwatts@calpoly.edu
Statement of Disclaimer

Since this project is a result of a class assignment, it has been graded and accepted as fulfillment of the course requirements. Acceptance does not imply technical accuracy or reliability. Any use of information in this report is done at the risk of the user. These risks may include catastrophic failure of the device or infringement of patent or copyright laws. California Polytechnic State University at San Luis Obispo and its staff cannot be held liable for any use or misuse of the project.
Abstract
Jack's Helping Hand and its hippotherapy participants required a device to serve as an alternative to a live horseback riding experience that could also increase the range of riders. This would provide more clients with equine-assisted therapy that has proven to better the lives of people with both physical and mental disabilities. Horses can be unpredictable, tall, and sometimes anxiety-inducing, especially for new riders. Our group’s aim was to develop a mechanical horse that will be able to reduce these issues for equine therapy centers and the riders they help. When a rider gets to practice sitting on the horse without the unpredictability or the height, the rider can develop confidence in riding before sitting on a real horse. After going out into the field to experience horse riding for ourselves and translating our experience into the design for our therapeutic mechanical horse, we have created a mechanical horse that can sit children who weigh up to 160 pounds and can withstand the conditions present in the barn where the machine will be stored.
Introduction

Jack’s Helping Hand is a non-profit organization which provides children with cancer or special needs under the age of 21 with community programs focused on enriching their lives. One of their enrichment programs, Little Riders, involves Equine therapy to help riders improve balance, strength, and coordination, which benefit the participants in a multitude of ways. While Equine therapy is a valuable resource for children with special needs, there are some limitations, such as the wear on lesson horses, low rider weight limits, and the varying levels of confidence and experience of riders, that prevent the Little Riders program to be accessible to more children. Due to this our sponsors, Ms. Orradre, chairperson and executive director, and Mrs. Burt, volunteer and event Coordinator, from Jack’s Helping Hand, reached out to Cal Poly with the hope to develop a device to serve as an alternative to a live horseback riding experience and increase the range of riders. This would provide more clients with equine-assisted therapy that has been proven to better the lives of people with both physical and mental disabilities.

To achieve making such a device, Cal Poly’s Mechanical Engineering Department assigned our team to plan, design, and fabricate the mechatronic system for a therapeutic mechanical horse frame being built by another senior project team. Our team consists of five members:

Aleya Dolorfino: 4th year mechanical engineering major with a concentration in mechatronics
Zuzanna Dominik: 4th year computer engineering major
Cade Liberty: 4th year mechanical engineering major with a concentration in mechatronics
Peter Phillips 4th year computer science major
Luke Watts 4th year computer science major

We saw this project as a wonderful opportunity to utilize all the knowledge we have acquired over the past three years to make a significant impact on the community. We share a passion for helping others and are extremely excited to work on the Therapeutic Mechanical Horse Team.

The purpose of this document is to clearly describe our final design plan all the steps taken to manufacture and assemble the final product for the Therapeutic Mechanical Horse Project. This report consists of four distinct parts as follows:

1. Scope of Work – The Scope of Work describes the specifications for the Therapeutic Mechanical Horse Project. Included in this document is background information of the project along with research done to aid in the process of designing the mechanical horse. We state our goals, and the project management plans to accomplish the set goal.
2. Preliminary Design Review – This is the first check-in document after the Scope of Work, focusing mostly on the design direction for the mechanical horse. It describes the team’s ideation process and our justification for the chosen direction of our design. In detail we explain our concept design and the plan we constructed to be able to complete said design.
3. Critical Design Review – Overall, the Critical Design Review is the full description of our final design along with the justification that the final design meets all the specification requirements for our project. We provide a detailed explanation for each component of the final design and the reasoning for choosing them. It includes the planned manufacturing steps to complete the verification prototype and all of the test we planned to ensure the final product would meet all of the specifications.
4. Final Design Review – This document consists of explains all the updates and changes to the final design since the Critical Design Review. It explains all the steps taken during the manufacturing of the verification prototype. Additionally, the Final Design Review describes all the of the test performed on our verification model and the result from said test. Lastly, we conclude the report with a discussion about the final design and any recommendations we have for improving or continuing the project.
Scope of Work

California Polytechnical San Luis Obispo
Senior Project 2021 - 2022

Aleya Dolorfino
adorfi@calpoly.edu

Zuzanna Dominik
zdominik@calpoly.edu

Cade Liberty
cliberty@calpoly.edu
Scope of Work

**Therapeutic Mechanical Horse**

California Polytechnic San Luis Obispo
Senior Project 2021 - 2022

Aleya Dolorfino
adorfi@calpoly.edu

Zuzanna Dominik
zdominik@calpoly.edu

Cade Liberty
cliberty@calpoly.edu
Abstract

Jack's Helping Hand, Equine therapy centers, and participants require a device to serve as an alternative to a live horseback riding experience and increase the range of riders. This would provide more clients with equine-assisted therapy that has proven to better the lives of people with both physical and mental disabilities. Horses can be unpredictable, tall, and sometimes anxiety-inducing, especially for new riders. Our group aims to develop a mechanical horse that will be able to mitigate these issues for equine therapy centers and the riders they help. When a rider gets to practice sitting on the horse without the unpredictability or the height, the rider can develop confidence in riding before sitting on a real horse. To accurately create our therapeutic horse, we extensively researched customer needs, competitive products, and technical documentation relevant to the project. This document captures a summary of our investigations and the engineering specifications they apply to. We have also included our planned milestones and the dates by which we will need to achieve them.
Table of Contents

Abstract......................................................................................................................................................... 1

1. Introduction............................................................................................................................................... 1

2. Background............................................................................................................................................... 1

   2.1 Stakeholders Needs/Wants.................................................................................................................. 1

   2.2 Existing Solutions............................................................................................................................... 2

      2.2.1 MiraColt....................................................................................................................................... 2

      2.2.2 Equicizer ...................................................................................................................................... 3

      2.2.3 Racewood Equestrian Simulator .................................................................................................. 3

      2.2.4 iJoy Ride ...................................................................................................................................... 4

      2.2.5 Persival......................................................................................................................................... 4

   2.3 Technical Research............................................................................................................................. 5

      2.3.1 Emulating a Horse’s Walk ........................................................................................................... 5

      2.3.2 Emulating a Horse’s Trot............................................................................................................. 6

      2.3.3 Real and Mechanical Horse Comparison ..................................................................................... 6

      2.3.4 Sensor Programming.................................................................................................................... 7

      2.3.5 Kinect Sensor Analysis of Mechanical Horses ............................................................................ 7

      2.3.6 Inertial Sensor Analysis of Mechanical Horse ........................................................................... 7

      2.3.7 IEEE Robotics Ontology Standard............................................................................................... 8

      2.3.8 Physical Safety in Robotics.......................................................................................................... 8

      2.3.9 Electrical Hazards & Safety Standard.......................................................................................... 9

3. Project Scope ...................................................................................................................................... 10

4. Objectives ........................................................................................................................................... 11

   4.1 Quality Function Deployment (QFD) ............................................................................................... 11

   4.2 Engineering Specifications.................................................................................................................. 12

      4.2.1 Specification 1: Withstand Rider Weight .................................................................................. 12

      4.2.2 Specification 2: Mech Horse Weight ......................................................................................... 12

      4.2.3 Specification 3: Movement Matches Walk Data ....................................................................... 12

      4.2.4 Specification 4: Movement Matches Trot Data ......................................................................... 12

      4.2.5 Specification 5: Accessible Parts ............................................................................................... 12

      4.2.6 Specification 6: Overall Cost ..................................................................................................... 12

      4.2.7 Specification 7: Withstands Dust ............................................................................................... 12

      4.2.9 Specification 9: Minimal Buttons/ Switches.............................................................................. 13

      4.2.8 Specification 8: Withstands Water............................................................................................. 13
1. Introduction

Jack’s Helping Hand is a non-profit organization which provides children with cancer or special needs under the age of 21 with community programs focused on enriching their lives. One of their enrichment programs, Little Riders, involves Equine therapy to help riders improve balance, strength, and coordination, which benefit the participants in a multitude of ways [1]. While Equine therapy is a valuable resource for children with special needs, there are some limitations, such as the wear on lesson horses, low rider weight limits, and the varying levels of confidence and experience of riders, that prevent the Little Riders program to be accessible to more children. Due to this, our sponsors, Mrs. Orradre and Mrs. Burt, from Jack’s Helping Hand, reached out to Cal Poly with hope to develop a device to serve as an alternative to a live horseback riding experience and increase the range of riders. This would provide more clients with equine-assisted therapy that has been proven to better the lives of people with both physical and mental disabilities.

To achieve making such a device, Cal Poly’s Mechanical Engineering Department assigned our team to plan, design, and fabricate the mechatronic system for a therapeutic mechanical horse frame being built by another senior. Our team consists of three members:

Aleya Dolorfino: 4th year mechanical engineering major with a concentration in mechatronics
Zuzanna Dominik: 4th year computer engineering major
Cade Liberty: 4th year mechanical engineering major with a concentration in mechatronics

We saw this project as a great opportunity to utilize all the knowledge we’ve acquired over the past three years to make a significant impact in the community. We share a passion for helping others and are very excited to work on the Therapeutic Mechanical Horse Team. In addition, we are working very closely with another senior project team, who’s about a quarter ahead of us, in the design process and working on the more mechanical side of the Therapeutic Mechanical Horse, to ensure both teams’ designs are compatible and targeted towards the same goals.

This purpose of this document is to clearly define what we expect to accomplish and how we plan to achieve said expectations for the Therapeutic Mechanical Horse Project. Included is background information focusing on the wants and needs of the project’s stakeholders, current designs that achieve a similar goal to what we want to accomplish, and some technical information used to enhance our understanding of the challenge at hand. In addition, we lay out our project scope and the objectives we plan to complete throughout the process of this project. Finally, we share a detailed plan on the next steps we will take to solving this design challenge and take a larger look at the work that we had ahead for this project.

2. Background

2.1 Stakeholders Needs/Wants

To produce the best design, we researched the customers that would be using this device.

The first customer that we found was our sponsor, Jack’s Helping Hand. Jack’s Helping Hand is a non-profit organization who provides children with cancer or special needs under the age of 21 with community programs focused on enriching their lives. Through their project definition and our subsequent interviews with them, we found that there is a need to provide an inexpensive replica of a walking horse for their riders. They want to use this device to help their new riders become used to the motion that horses produce while walking and enable them to train their muscles and gain confidence...
while riding in a safer and less intimidating experience than on a real horse. They have also identified that this device should be simple and easy to use, as many of the volunteers that help them do not have a STEM background. Finally, the sponsor explained that they would like to have this device last for a significant period of time despite being exposed to a dusty, and possibly wet environment.

After interviewing our sponsor, we also looked for other stakeholders that are involved in the process this device. The next stakeholder we identified was the riders with special needs which would be the primary users of this device. Jack’s Helping Hand serves hundreds of people, all of them with unique challenges that can cause getting onto a horse to be difficult and sometimes dangerous. Thus, we need to ensure our solution is safe so the riders can feel comfortable and confident on top of it. Beyond that, we determined that this device should be fun to ride and mimic horse motion closely. The complex interaction between the rider and the horse is a big part of the effectiveness of hippotherapy. Thus, this device must mimic a horse in looks as well as motion. Finally, we determined that the rider should have something to hold on to, similar to reins on an actual horse, this would give the rider experience in balancing themselves which will apply benefits to when they move on to using a live horse for their sessions.

The last major stakeholder that we determined are the people dedicated to continuous improvement on our device. We think that with this device there is a significant amount of room to grow and for it to adapt and fit multiple situations. To accomplish this, we would like the device to be simple in terms of design and manufacturing. Often, designs can get rather complicated as time goes on, until only those who designed them can make any sense of what is going on. We would like for others to be able to use this as a stepping stone, enabling them to build on this device and use it in other applications. Because of this, we want the electrical connections, the controls, and the overall manufacturing to be simple and modular to allow for easy understanding. On top of this, we want the device to have more capability than is defined in the scope that Jack’s Helping Hand may use, such as being able to carry more weight than the limit our sponsor gave us. This would allow for the system to be robust enough to be used and adapted easily into other situations. Finally, for this stakeholder, the device should be inexpensive to make so that others will not be gated by the entry cost.

2.2 Existing Solutions
Equine therapy is a practice that has been around for a very long time but the need for a mechanical alternative is recent. The following are a few existing solutions and designs for an alternative to live horse riding. It’s important to note that are no patents included in the existing solution because the technology for this scope is in such a niche field that the patents even remotely related to this project don’t provide must insight for our scope of work.

2.2.1 MiraColt
Designed by Chariot Innovations, the Miracolt is one of the leading technologies in mechanical hippotherapy, and something our sponsors identified as too expensive a. The MiraColt realistically reproduces the three-dimensional, pitch, roll, and yaw, motion patterns of a walking horse. It is one of the few riding simulators with both side-to-side and forward-backward motion. Some important characteristics of this design is the adjustable handlebar for adaptability to riders and the locking wheels that allow for easy transportation of the project. Additionally, the control scheme for this system is very simple with a go and stop button, and minimal speed settings. However, what this design lacks are aesthetics which we have deemed more important to our personal design after experiencing what an equine therapy session really entails and the relationship between the rider and the horse. [2]
2.2.2 Equicizer
Developed in 1982 by Frank Lovato Jr., who designed the system for rehabilitation after fracturing his leg in a racing accident, the Equicizer is another well-used design for equine therapy. The Equicizer, while one of the most durable designs with its maximum weight of 500lbs, is the only non-motorized design. The Equicizer’s moment is fully dependent on the rider, since all its motion comes from a spring-balanced mechanism that, when moved, provides the sensation of riding a horse. This lack of motorization wouldn't be able to accomplish what our sponsor wants from our design, because the children who are riding the horses don’t have enough control of their movements to safely maneuver a non-motorized mechanism. Some children are also novice riders and do not know how to replicate a walking motion. However, the use of a springboard is quite novel and has the potential to be built upon in our design as a cost-effective way to add more force into the motorized movements of our machine. [3]

2.2.3 Racewood Equestrian Simulator
The Racewood Equestrian Simulator’s design is more focused on the user interface than the other existing designs listed in this Scope of Work. The mechanical horse itself is connected to at least one monitor with a screen projecting a rider’s predicted environment to simulate not only the motion of a horseback riding experience, but also the other sensory triggers a rider would experience when riding a live horse. The Racewood Equestrian Simulator is generally used for experienced riders, and while it does find its way into the sector of Equine Therapy, the multiple feedback sensors placed within the saddle are mostly used to evaluate the movement of the rider rather than help develop the rider’s skill. The Racewood Equestrian Simulator does a great job of making the rider feel as if they were on a real horse with its visual, audio, and motion simulation but it all comes at a price. The Racewood Equestrian Simulator is by far the most expensive design on the market and while we do want to accomplish the feat of having a life-like horse riding experience, cost is a large limitation for our project. [4]
2.2.4 iJoy Ride

The iJoy Ride isn’t technically a horse simulator but it does execute the same results that our project hopes to achieve. The iJoy Ride is an exercise machine that is advertised to help improve the balance and core strength of the user, both goals of a good equine therapy session. The major downside of this design is the lack of movement in the yaw direction. While this is a characteristic missing among many horse simulators, the iJoy Ride lacks this movement the most, as it moves in a more circular pattern than the swinging hip action of a real horse. Analyzing this design may prove beneficial from the physical perspective because, while it doesn’t move like a horse, it does improve the rider’s balance and strengthens their core in ways riding a horse does. [5]

2.2.5 Persival

In 1986 a competition was organized by the National Robotics Committee in hopes of finding technical solutions to develop the first mechanical horse. A large array of engineers put their minds together to develop the Persival. The movement of the horse simulator, Persival, located at the French National Equestrian School is derived from a medley of components working together. This includes the action of six SKF linear actuators with roller screws, linked triangular plates, one on the floor and the other at a 60-degree offset attack to the body of the horse, and electric motors to adjust each leg. With a linear potentiometer to regulate the movements precision and a computer to control all the mechatronics withing the system, the Persival is able achieve the all the movements of a horse, including the jumps. Like the Racewood horse this design is on the more expensive side of the cost spectrum, but the extensive use of motors and linear actuators is a great inspiration for our mechatronic design. [6]
2.3 Technical Research
2.3.1 Emulating a Horse’s Walk
A significant part of the project definition is emulating a horse’s walk. Quadrupeds have a four-beat walk during which the majority of the time is spent with three feet on the ground, as can be seen in Figure 3. The solid lines in the figure show when a cat had its paws on the ground while walking. As can also be seen in the same graph, the vertical motion can be closely approximated by a sine wave, which our team plans on implementing to simplify the programming. It is also important to note that, although our product is stationary, the forward motion data shows how the motion should not feel linear, but rather follow some sort of periodic pattern. [7]

Figure 3: Body’s center of gravity velocity in vertical and forward components. Periods during which feet contact the ground also shown.

Figure 4: Center of gravity acceleration in 3 dimensions.
It is also vital to examine a horse’s acceleration in the 3 degrees of freedom that we have to work with. Beyond examining how the center of gravity shifts with movement, we also need to examine how fast those changes occur in order to properly emulate the motion of a quadruped. As can be seen in Figure 4, the quadruped moves much faster in the X and Z directions. However, it must be noted, that this experiment was done by analyzing frames on a video reel, thus it is possible the y axis movement was so smooth because of the camera angle. Regardless, these accelerations also follow a smooth, periodic motion that should be easily emulated by sine waves. [7]

2.3.2 Emulating a Horse’s Trot

Although the trot function is secondary to the walk, it is still important to design with it in mind. Trotting is a two-beat motion during which diagonal pairs of legs are lifted and placed ahead. In their research, Hugh Herr and Thomas McMahon used proportional-derivative servos to position the joints on a quadruped model. To move a horse model, the servos supplied a \( \tau = -G_p(\theta_m - \theta_t) - G_v(V_m - V_t) \) where \( G \) are position and velocity gains, \( \theta \) are measured and target joint positions, and \( V \) are measured and target velocities. [8]

It is also important to know how long each foot will be in the air, which is modelled by the equation:

\[ T_{delay} = \frac{G_1v}{g} + G_2(a_{prev} - a) \]

Here, \( G \) are gains, \( v \) is the vertical takeoff velocity of the center of mass, \( g \) is the gravitational constant, and \( a_{prev} - a \) is the difference between beginning and end body pitch. [8]

![Figure 5: The center of mass, plotted on Z-axis over time](image)

It is also important to note, that for the rider, the most significant difference between the trot and the walk is the amount of vertical acceleration. Luckily, this movement is a fairly constant oscillation, and should simplify the modelling that needs to be done, which can be seen in Figure 5. [8]

2.3.3 Real and Mechanical Horse Comparison

Now that we have explained the two motions our mechanical horse must mimic, we must explore possible ways to achieve those goals. We looked at how well the overall mechanical horse market matches real horse data in order to determine what makes a good mechanical horse. It turns out, that although many horse simulators have excellent reliability (\( R^2 = 0.976 \)) and resultant acceleration (\( R^2 = 0.997 \)), they do not match the x-acceleration data well (\( R^2 = 0.177 \)). [9] To be competitive, therefore, we must primarily
focus on matching the overall acceleration. If our x-acceleration is not matching the data as well as it should, that’s ok because many other people are struggling with the same problem.

2.3.4 Sensor Programming
In partnership with the Irish Turf Club, a regulatory body for horse racing, researchers gathered data on how jockeys rode mechanical horses for practice. It is important to note the data we find more interesting than the conclusion is the control and calibration of the sensors. The article outlines how all the data was split into events, with clear start and end times, that took into account jockey data and accelerometer data. Our team could use a similar layout to standardize our product testing. [10]

The pre-recorded events are also displayed in the research, showing the acceleration in the x, y, and z direction that accompanies certain beats of a canter. [10] Although this project will not include a canter, it is useful to see how any gait could be integrated into a sensor system to get an idea of how we should do it with existing software and hardware.

2.3.5 Kinect Sensor Analysis of Mechanical Horses
A Kinect sensor is built to analyze human motion, as it was originally intended to allow people to play games without a remote. The researchers used these capabilities of a Kinect sensor to determine the frequency of different joints on the body as a volunteer rode a mechanical horse. It was determined that the head, neck, left shoulder, and right shoulder were the primary discriminators to determine how fast the horse was going. The higher the frequency, the “faster” the subjects were going. It was found that although the walk had a very similar frequency (around 50 Hz) on all the major joints, the trot varied much more (between 60 Hz for the Left Shoulder and 120 Hz for the head). [11] It is useful to know what points we could look at to determine how well our mechanical horse is matching desired data with a person on it.

2.3.6 Inertial Sensor Analysis of Mechanical Horse
A group of researchers set out to find how a body on a mechanical horse is impacted by inertia. They tested different inert charges (between 50 and 90 kg, at 10 kg intervals) to see how the load was accelerated by the mechanical horse. The frequency at which the horse operated was also incremented between 1 and 2.2 Hz, with a 0.4 Hz increment. The conclusion was the rider “adversely influences the simulator” because the stable frequency changes depending on the load. The researchers found that 70kg was the breakpoint between one stable frequency and another. It appears that under 70kg the mechanical horse is stable at 1.75 Hz and above 70kg, it’s stable at 1.85 Hz. A depiction of the acceleration of a 70kg mass at 1.8 Hz is shown in Figure 6. Keeping this research in mind, our team must implement some sort of detection in the software to ensure that no matter the weight of the rider (as they can be between 4 and 21 years old) the simulator should still act relatively the same. [12]
2.3.7 IEEE Robotics Ontology Standard
Robotics is still a relatively new branch of the IEEE, thus their standards are often revisited and revised as more robotics technologies emerge. We decided to research how robotics technologies are made so we can better tailor our ideation and creation phases with what professional engineers are doing. The main development activities had 6 steps: Environment Study, Conceptualization, Formalization and Implementation, Evaluation, Maintenance, and Documentation. [13]

Our senior project class will most likely guide us through all these steps. Nevertheless, we must remember to first acquire information on the platform we will use (Environment Study). Second, we must provide a conceptual model highlighting the information we have acquired (Conceptualization), which we are doing currently. Next, we transform our model into a computable one (Formalization). This is where we will start implementing our functional code. Then, we must check the consistency of our output (Evaluation). And we must also make sure to keep our system updated (Maintenance). Finally, all our work must be documented so the process is replicable (Documentation). [13]

2.3.8 Physical Safety in Robotics
Our mechanical horse is a robot; thus it must meet certain standards for being safe around humans. Especially since it will be interacting with children. It is important to keep in mind both phase 1 and phase 2 of impact, as shown in Figure 7. Phase 1 is the short impact, the first moment a person and a robot would collide. Phase 2, on the other hand depends on whether or not the person is trapped or free [14]. An example would be the different injuries that may be caused by a robot hitting someone who is trapped against a wall versus a person who has no wall and just falls. There are many equations that go into determining exactly how much damage a robot could cause, but in interest of time, we simply wish to state our goal is to decrease any impact our robot could have with a customer so that no serious injuries occur.
2.3.9 Electrical Hazards & Safety Standard

We would hate if any of the customers were to get electrically shocked by our system. There are two ways of grounding the system, so that electricity will flow to into the wall instead of through a person.

As can be seen in Figure 8 we can choose to either double-insulate all of our electrical components, or to ground the horse frame such that any electrical charges would go to ground. It is important to note that our solution will depend on the types of outlets available where the mechanical horse will be stored. Additionally, Figure 9 provides a look into how these wall outlets work and the AC wave they generate. Please note the plumbing in this diagram shows how water can also ground the circuit and cause some issues with how it operates. [15]
3. Project Scope
The Therapeutic Mechanical Horse project is split up among two distinct senior project teams. The first team was established in Spring Quarter of 2021 and, after recognizing that designing both the mechanical frame and the mechatronics/controls of the mechanical horse was too large a scope, they decided to delegate the mechatronics to us. Our project scope is centered around the user interface and the development of the mechatronics within the mechanical horse. We will work with the other senior project team to develop the “skeleton” of the Therapeutic Mechanical Horse and once they are done with their design process, we can really start our work. Taking the data the other team collected for their design and translating it into a language that the motor can understand is the first goal of our project scope. Then, after fine tuning the movements of our mechanical horse, we’ll transition to focusing on the controls of the system and making it usable for our sponsors and the people involved in the Little Riders Program. A description of the functions that our design needs to be able to accomplish is shown in Figure 10 below. Each item is a listed function that we will need to accomplish in the final Therapeutic horse but only those in the red box are under the scope of this project.

Figure 9: A schematic for a 120V wall outlet

Figure 10: Functional Decomposition diagram with associated scope of work outlined in red box
4. Objectives
To create our final product in a timely manner, we must first decide on the specifications that our final product must meet. Thus, we completed a Quality Function Deployment (QFD) and outlined the main specifications as well as why they are important to the customers.

4.1 Quality Function Deployment (QFD)
The Quality Function Deployment, found in Appendix A, quantifies customer needs, engineering specifications, and competitor data. It is used to determine the most important specifications for the project as well as what engineering specifications must be focused on in order to best satisfy the customer. We analyzed the wants and needs of equine therapy centers, riders, manufacturers, and continuous improvement teams and assigned specifications that would satisfy the wants of as many customers as possible which is listed in table 1 below.
4.2 Engineering Specifications

<table>
<thead>
<tr>
<th>Spec #</th>
<th>Specification Description</th>
<th>Requirement or Target</th>
<th>Tolerance</th>
<th>Risk</th>
<th>Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Withstand Rider Weight</td>
<td>200 lbs</td>
<td>Min</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>2</td>
<td>Mech Horse Weight</td>
<td>125 lbs (with saddle)</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
<tr>
<td>3</td>
<td>Movement Matches Walk Data</td>
<td>Data within 20%</td>
<td>Target</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>4</td>
<td>Movement Matches Trot Data</td>
<td>Data within 20%</td>
<td>Target</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>5</td>
<td>Accessible Parts</td>
<td>No permanent electronic connections</td>
<td>Target</td>
<td>Low</td>
<td>Inspection</td>
</tr>
<tr>
<td>6</td>
<td>Overall Cost</td>
<td>$500</td>
<td>Max</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>7</td>
<td>Withstands Dust</td>
<td></td>
<td>Target</td>
<td>Medium</td>
<td>Test</td>
</tr>
<tr>
<td>8</td>
<td>Withstands Water</td>
<td></td>
<td>Target</td>
<td>Medium</td>
<td>Test</td>
</tr>
<tr>
<td>9</td>
<td>Minimal Buttons/Switches</td>
<td>3 buttons/switches</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
<tr>
<td>10</td>
<td>Cost Aesthetic Supplies</td>
<td>$20</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
<tr>
<td>11</td>
<td>Functions on Different Surfaces</td>
<td>Data within 30%</td>
<td>Target</td>
<td>Medium</td>
<td>Test</td>
</tr>
<tr>
<td>12</td>
<td>Mech Horse Size</td>
<td>2ft x 4ft x 3ft</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
</tbody>
</table>

Table 1: Engineering Specifications

4.2.1 Specification 1: Withstand Rider Weight
As the therapeutic horse must service many riders, it is critical to maximize the rider weight acceptable on the system. We must ensure that the walking and trotting functions still execute properly no matter the rider weight.

4.2.2 Specification 2: Mech Horse Weight
In the interest of helping the therapy centers, the mechanical horse must not weigh too much. Jack’s Helping Hand expressed an interest in moving the horse from a tack room and outside for each use, thus, the weight of the horse must be reduced to increase portability.

4.2.3 Specification 3: Movement Matches Walk Data
The main objective of this project is to mimic the movement of a horse. We want to get as close to the walking data gathered from a real horse as possible. Thus, we must perform under the maximum offset for pitch, yaw, and roll functions.

4.2.4 Specification 4: Movement Matches Trot Data
The sponsor is most interested in mimicking a walk, but we aim to include the trot as well. We want to approximate the trotting data gathered from a real horse by performing under the maximum offset for pitch, yaw, and roll functions.

4.2.5 Specification 5: Accessible Parts
Ideally, the mechanical horse will be in use for a long period of time. It is important to allow for broken parts to be replaced quickly and easily. So, we strive to ensure no electronic connections are permanent.

4.2.6 Specification 6: Overall Cost
Our senior project team is assigned $500 from the ME Department. We must not exceed this amount. We do not receive funding from our sponsor, as they are a non-profit organization and any money that goes over the $500 would have to be supplied by a grant.

4.2.7 Specification 7: Withstands Dust
The mechanical horse will be stored at a barn and in a tack room. Horses and their equipment generate a lot of dust, thus the mechanical horse must be impervious to dust buildup.
4.2.9 Specification 9: Minimal Buttons/ Switches
The volunteers at Jack’s Helping Hand switch out often and there is only a handful that stay on full-time. Thus, the system must be easy to learn and operate by a layperson.

4.2.8 Specification 8: Withstands Water
The mechanical horse will be used in lessons. It is possible someone would spill a water bottle on it. This must not completely ruin the mechanical horse. It does not have to withstand rain, however, because the horse will be stored under a tarp or indoors.

4.2.10 Specification 10: Cost Aesthetic Supplies
During our observation of a therapy session, we saw how important connection with the horses was to the children riding them. Thus, we decided to include a budget to make the mechanical horse look horse-like to foster a kinship between the children and the machine.

4.2.11 Specification 11: Functions on Different Surfaces
The barn has an outside arena and the mechanical horse will mostly be used outside. Additionally, Jack’s Helping Hand is working on moving their equine therapy program to a different barn. Thus, we want to ensure the mechanical horse will still function similarly no matter what surface it is placed on, as long as it is relatively flat.

4.2.12 Specification 12: Mech Horse Size
For each use, the mechanical horse will be moved, thus it cannot be too large. Also, it has to be stored in a tack room, so it must fit through a doorframe.

5. Project Management
The design process consists of several parts, our first task was gathering data on current products and those who will use this product. From there we have been working on defining the problem, ensuring that we understand and can convey the complexity of the work that we will accomplish. This document serves as our understanding of this problem definition. In this stage, we used several design tools, such as a quality function deployment and a functional decomposition, to quantify the needs and wants of our sponsors to concrete specifications that we can hold ourselves to. Next, we will be working on ideation, creating several designs that will be able to solve our design challenge. We will be using a myriad of design tools to create these solutions, such as brainstorming and brainwalking, to generate many different ideas to solve our design challenge. After this step, we will present these solutions to the sponsor in the Preliminary Design Review, where we will choose which solution to pursue.

Moving on from Fall quarter we will be focusing on developing one design solution to make as our final product. A detailed plan of the required steps to finalize this design is summarized below in table 2 with an associated Gantt chart documenting all steps in Appendix B. The first step in this process is performing detailed analysis and finding all the components that we will use for the solution. After getting confirmation with the sponsor about the design, we will move into ordering and manufacturing a structural prototype of our design. During the manufacturing of this device, we will be performing tests that validate that our design functions to our specifications in a safe manner. Finally, we will finish our design with a Final Design Review that will cover our full design with inclusion of our prototype and an operational manual for use.
### Table 2. Summary of key milestones and descriptions with associated competition dates

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Description</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Concept Prototype</td>
<td>Physical model of various initial design solutions</td>
<td>11/9/21</td>
</tr>
<tr>
<td>Preliminary Design Review</td>
<td>Detailed review of initial design solutions</td>
<td>11/18/21</td>
</tr>
<tr>
<td>Interim Design Review</td>
<td>Detailed review of design solution and analysis</td>
<td>1/13/21</td>
</tr>
<tr>
<td>Critical Design Review</td>
<td>Detailed review of components, costs, analysis, and proposed solution</td>
<td>2/11/21</td>
</tr>
<tr>
<td>Manufacturing &amp; Testing Review</td>
<td>Status update of Manufacturing, updated test plan, and updated schedule of project completion</td>
<td>3/10/21</td>
</tr>
<tr>
<td>Prototype Sign Off</td>
<td>Verification of functionality of Final Prototype</td>
<td>4/26/21</td>
</tr>
<tr>
<td>Senior Expo</td>
<td>Showcase of structural prototype and Expo poster</td>
<td>5/27/21</td>
</tr>
<tr>
<td>Final Design Review</td>
<td>Final review and handoff of prototype and design report</td>
<td>6/3/21</td>
</tr>
</tbody>
</table>

6. Conclusion

Overall, the scope of this document was to document our understanding of the design challenge that our sponsor has given us. Overall, our sponsors want an alternative solution to riding a live horse that will enable riders of the Little Riders program train their confidence and skills in a safe manner off a live horse. However, because there is another Senior Project team working on this project, our scope is limited to only focusing on the mechatronic system to enable motion and the user interface that goes with it. Through our research we have identified several needs of our stakeholders and attributed measurable specifications that we will be able to hold ourselves to as we design this system. Moving beyond this document we will be working on creating several ideas that will solve our design challenge and propose these to our sponsor in 4 weeks at the Preliminary Design Review, after which we will choose one design to progress for our final design. By writing this document we hope our sponsors attest that the content here is accurate, confirming our understanding of the challenge given to us and the efficacy of our execution plan.
References


## Appendix A – Quality Function Deployment House of Quality

<table>
<thead>
<tr>
<th>Row #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>5</td>
<td>6</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>9</td>
<td>5</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Rating</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Kano</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Importance</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Relationship</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

### WHAT: Customer Requirements

<table>
<thead>
<tr>
<th>WHAT: Customer Requirements</th>
<th>Suppliers</th>
<th>Design</th>
<th>Value Chain</th>
<th>Competition</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Comfort</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Safety</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Efficiency</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Durability</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Style</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Cost</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### WHERE: Suppliers

<table>
<thead>
<tr>
<th>WHERE: Suppliers</th>
<th>Design</th>
<th>Value Chain</th>
<th>Competition</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Comfort</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Safety</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Efficiency</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Durability</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Style</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Cost</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### HOW: Design

<table>
<thead>
<tr>
<th>HOW: Design</th>
<th>Value Chain</th>
<th>Competition</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Comfort</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Safety</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Efficiency</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Durability</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Style</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>- Cost</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

### Analysis

<table>
<thead>
<tr>
<th>Column</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
<th>16</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our Current</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Competitor A</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Competitor B</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
<tr>
<td>Competitor C</td>
<td>1</td>
<td>2</td>
<td>3</td>
<td>4</td>
<td>5</td>
<td>6</td>
<td>7</td>
<td>8</td>
<td>9</td>
<td>10</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>14</td>
<td>15</td>
<td>16</td>
</tr>
</tbody>
</table>

### Chart

- Chart showing correlation and relationship between WHAT, WHERE, and HOW.
- Diagram illustrating the QFD House of Quality with customer requirements, supplier analysis, and design considerations.

### Conclusion

- Detailed analysis of customer, supplier, and design factors with respective weights and ratings.
- Comparison of current product with competitors in terms of ratings and analysis.

---

**Appendix A – Quality Function Deployment House of Quality**

**Revision Date:** 10/12/2022

**Correlations**
- Positive
- Negative
- No Correlation

**Relationships**
- Strong
- Moderate
- Weak

**Objectives of Improvement**
- Machine
- Target
- Minimum

**Product Requirements**
- Comfort
- Safety
- Efficiency
- Durability
- Style
- Cost

**Customer Requirements**
- Comfort
- Safety
- Efficiency
- Durability
- Style
- Cost

**Supplier Requirements**
- Comfort
- Safety
- Efficiency
- Durability
- Style
- Cost

**Value Chain**
- Design
- Value Chain
- Competition

**Competitor Analysis**
- Competitor A
- Competitor B
- Competitor C
- Current Product
Preliminary Design Review
California Polytechnical San Luis Obispo
Senior Project 2021 - 2022

Aleya Dolorfino
adorffi@calpoly.edu

Zuzanna Dominik
zdominik@calpoly.edu

Cade Liberty
cliberty@calpoly.edu

Peter Phillips
pphill03@calpoly.edu

Luke Watts
luwatts@calpoly.edu
Preliminary Design Review

**Therapeutic Mechanical Horse**

California Polytechnical San Luis Obispo
Senior Project 2021 - 2022

Aleya Dolorfino
adolorfi@calpoly.edu

Zuzanna Dominik
zdominik@calpoly.edu

Cade Liberty
cliberty@calpoly.edu
Abstract
Jack's Helping Hand, Equine therapy centers, and participants require a device to serve as an alternative to a live horseback riding experience and increase the range of riders. This would provide more clients with equine-assisted therapy that has proven to better the lives of people with both physical and mental disabilities. Horses can be unpredictable, tall, and sometimes anxiety-inducing, especially for new riders. Our group aims to develop a mechanical horse that will be able to reduce these issues for equine therapy centers and the riders they help. When a rider gets to practice sitting on the horse without the unpredictability or the height, the rider can develop confidence in riding before sitting on a real horse. To accurately create our therapeutic horse, we brainstormed possible designs, evaluated their feasibility, and chose the best ones. In this document are outlined our original ideas and how they informed the preliminary design we have now.
# Table of Contents

Abstract ......................................................................................................................................................... 1

1. Introduction........................................................................................................................................... 1

2. Concept Development........................................................................................................................... 1
   2.1 User Interface Controls Ideation ......................................................................................................... 2
      2.1.1 Three Input Knob ......................................................................................................................... 2
      2.1.2 Two Buttons and a Speed Control Knob ...................................................................................... 2
   2.2 User Interface Connection Ideation .................................................................................................... 3
      2.2.1 Bluetooth ...................................................................................................................................... 3
      2.2.2 Post ............................................................................................................................................... 3
   2.3 Movement Ideation ............................................................................................................................. 4
      2.3.1 Two Motor ................................................................................................................................... 4
      2.3.2 Two Motors with a Linear Actuator ............................................................................................. 4
   2.4 Electronic Ideation .............................................................................................................................. 5
      2.4.1 Voltage Multiplier ........................................................................................................................ 5
      2.4.2 Motor Controller .......................................................................................................................... 6
   2.5 Programming Ideation ........................................................................................................................ 6
      2.5.1 No Structure ................................................................................................................................. 7
      2.5.2 Structure Multiple Motion ........................................................................................................... 8
      2.5.3 Structure Singular Motion ............................................................................................................ 9
   2.6 Controlled Convergence ..................................................................................................................... 9

3. Concept Design ................................................................................................................................... 10

4. Concept Justification ........................................................................................................................... 12
   4.1 Design Specifications ........................................................................................................................ 12
      4.1.1 UI Control Scheme ..................................................................................................................... 12
      4.1.2 UI Connection ............................................................................................................................ 13
      4.1.3 Movers ....................................................................................................................................... 13
      4.1.4 Electrical Diagram ..................................................................................................................... 13
   4.2 Design Hazards ................................................................................................................................... 13
      4.2.1 T Joint Pinch Point ..................................................................................................................... 13
      4.2.2 Two-Link Arm Pinch Point ......................................................................................................... 13
      4.2.3 Seat is a Moving Mass ............................................................................................................... 14
   4.3 Current Challenges ............................................................................................................................ 14

5. Project Management ........................................................................................................................... 14
6. Conclusion ................................................................................................................................. 15

Appendix A – Extra Ideation .............................................................................................................. A-1
   A1 User Interface Layout .................................................................................................................. A-1
      A1.1 Three Buttons and a Switch ................................................................................................. A-1
      A1.2 Three Buttons ...................................................................................................................... A-1
      A1.3 Two Buttons and a Switch .................................................................................................. A-2
   A2 User Interface Connection: ...................................................................................................... A-2
      A2.2 Free Floating Cable ............................................................................................................ A-2
   A3 Movers ..................................................................................................................................... A-3

Appendix B – Pugh Matrices ............................................................................................................. B-1
   B1 User Interface Pugh .................................................................................................................... B-1
   B2 User Interface Connection Pugh ................................................................................................ B-1
   B3 Mover Pugh ............................................................................................................................. B-2
   B4 Electrical Pugh ........................................................................................................................ B-2
   B5 Coding Pugh ............................................................................................................................ B-3

Appendix C – Weighted Decision Matrix .......................................................................................... C-1
   C1 UI Control Decision Matrix ..................................................................................................... C-1
   C2 UI Connection Decision Matrix .............................................................................................. C-2
   C3 Mover Decision Matrix .......................................................................................................... C-4
   C4 Coding Decision Matrix ........................................................................................................... C-5

Appendix D – Design Hazard Checklist ........................................................................................... D-1

Appendix E – Gantt Chart .................................................................................................................. E-1
1. Introduction

Jack’s Helping Hand is a non-profit organization which provides children with cancer or special needs under the age of 21 with community programs focused on enriching their lives. One of their enrichment programs, Little Riders, involves Equine therapy to help riders improve balance, strength, and coordination, which benefit the participants in a multitude of ways [1]. While Equine therapy is a valuable resource for children with special needs, there are some limitations, such as the wear on lesson horses, low rider weight limits, and the varying levels of confidence and experience of riders, that prevent the Little Riders program to be accessible to more children. Due to this our sponsors, Mrs. Orradre, chairperson and executive director, and Mrs. Burt, volunteer and event Coordinator, from Jack’s Helping Hand, reached out to Cal Poly with hope to develop a device to serve as an alternative to a live horseback riding experience and increase the range of riders. This would provide more clients with equine-assisted therapy that has been proven to better the lives of people with both physical and mental disabilities.

To achieve making such a device, Cal Poly’s Mechanical Engineering Department assigned our team to plan, design, and fabricate the mechatronic system for a therapeutic mechanical horse frame being built by another senior project team. Our team consists of three members:

- Aleya Dolorfino: 4th year mechanical engineering major with a concentration in mechatronics
- Zuzanna Dominik: 4th year computer engineering major
- Cade Liberty: 4th year mechanical engineering major with a concentration in mechatronics

We saw this project as a wonderful opportunity to utilize all the knowledge we have acquired over the past three years to make a significant impact on the community. We share a passion for helping others and are extremely excited to work on the Therapeutic Mechanical Horse Team.

The purpose of this document is to clearly define what we expect to accomplish and how we plan to achieve said expectations for the Therapeutic Mechanical Horse Project. Included in this document are the concept development we completed to develop our final design, a detailed explanation of our design, and justification for our chosen system. In addition, we share a detailed plan of the next steps we will take to solve this design challenge and take a larger look at the work that we have for this project.

2. Concept Development

Overall, the first part in our design development was to perform ideation. Ideation is the process of producing many different ideas that could solve the system. To maximize our thinking and to make our ideas simpler we first produced 5 different sections of our design. The first section was the user interface control scheme. In this section we developed different ideas on how this system could be controlled and how it is able to operate. The next section was the user interface connection where we covered how the overall user interface would be connected to and communicate with the controls on the mechanical horse. After the UI we move into the movement ideation. In this section, we developed different ideas that we thought could provide the motion that we wanted from the mechanical horse. Beyond the movement ideation we move into the electrical diagram section. In this section we brainstormed diverse ways to connect the electrical systems and even put what electrical systems we want to have in the device. Finally, our last section was how we organize the code. In this section we developed different ideas on what the code would look like and different methods that we could put into our code. A discussion of the top ideas for each section and how they would function can be found in the following sections.
In this section we go over the top 2 ideas for the user interface control scheme. This control scheme describes the set of inputs that the user can use to control its speed. Primary concerns for this function were its ability to be adapted, its ability to control speed, and its simplicity.

2.1.1 Three Input Knob

![Image of a UI controlled by a knob]

This design in Figure 1 here features a knob that can be set to three distinct outputs. These outputs are listed from the slowest speed on the left to the fastest speed on the right and include an off speed that turns the machine off. We determined that this was a beneficial design since it was quite easy to understand but also allowed for instant feedback as there was no guessing at what speed the device would be moving at. However, this device lacked precise control of the speed and could not vary the speed if we wanted to have a faster walk or slower trot.

2.1.2 Two Buttons and a Speed Control Knob

![Image of a UI controlled by a knob]

The design shown above in Figure 2 consists of two buttons and a knob able to send outputs based on the position of the knob. One of the two buttons would act as a power switch to begin the flow of voltage to the mechatronic system of the mechanical horse. The other button would be a power off button that would signal to the mechatronic system to stop the flow of voltage stopping anymore movement of the mechanical horse. The knob would adjust the speed of the mechanical horse based on outputs corresponding to the position of the knob. For example, outputs for a slower walk would be obtained from
the knob being more towards the left and outputs for a fast walk, or even a trot, would be obtained from
the knob being more towards the right. An advantage of this design is that the knob opens the opportunity
for this system to be modified and allows room for other movement patterns if they are desired in the
future. The knob allows for an almost unlimited number of outputs compared to a button that could only
allow for a single signal to be sent to the microcontroller.

2.2 User Interface Connection Ideation
In this section we will go over the top two ideas for the User Interface connection. The User Interface
connection is the system that connects the controls on the User Interface to the moving electrical
components on the actual device. The primary concerns for this were the lack of safety concerns, the
ability not to be disconnected during operation, and the systems’ ability to withstand the environment.

2.2.1 Bluetooth

![Figure 3: Conceptual Building of a Bluetooth Connection between the UI and the System](image)

This design found in Figure 3 features a Bluetooth connection that sends the input controls from the User
interface to the electrical system on the horse. This system is beneficial since it will be hard to disconnect
during use, poses little safety concerns and allows for easy moving around the full system. However,
since this will be tied to a remote, it can be easy to lose. One adjustment we will add is to make some sort
of harness to allow for the remote to be stored on the actual horse when not in use.

2.2.2 Post

![Figure 4: Conceptual Building of a Post Connection between the UI and the System](image)

This design in Figure 4 shows a post connection that will hold the UI in one spot relative to the horse
system. Overall, this design is beneficial as it is hard to lose where the UI is connected and would be hard
to disconnect the UI during operation. However, this system poses a safety concern as there is a moving
piece next to this stationary post which may allow people to hit it during operation. Also, it can cause a tripping hazard if the volunteer helping the rider needs to move around the device.

2.3 Movement Ideation

The purpose of the therapeutic mechanical horse is to mimic the movements of a horse through producing pitch, roll, and yaw motions. This can be achieved with a movement system that is responsible for jostling the seat of the mechanical horse. Since there are many ways to achieve movement for the mechanical horse system, ideation was needed to narrow the options of the movement systems. During ideation and controlled convergence, we focused on two main goals for our design. One, to make sure it could be easily implemented into the current design of the other senior project team. And two, that our design produced a large motion in the yaw direction.

2.3.1 Two Motor

![Figure 5: Two Motor Movement System](image)

In the figure above, Figure 5, is one of the first ideations for a system to create movement in the seat of the mechanical horse. This design consists of two two-link arms that are each connected to a motor attached to the base of the frame of the mechanical horse. As the two motors rotate, the two-link arms move in a way that produces both pitch and roll motion for the seat of the mechanical horse, however it produces little to no motion in the yaw direction. Although this design best works with the other team’s current design, it does not produce enough yaw to fully mimic the movements of horse, so another design was necessary.

2.3.2 Two Motors with a Linear Actuator

![Figure 6: Movement System with Two Motors and a Linear Actuator](image)

The design in Figure 6 is similar to the previous ideation found in Figure 5 but with an added linear actuator. The linear actuator adds side to side of motion to the system, and when connected to the seat with a single pivot point in the center of the frame the, the linear actuator produces the yaw movements...
the other designs were lacking. Since this design was an adaptation from the other senior project team’s current design for the mechanical horse, we know that the implementation of this design should be easy.

2.4 Electronic Ideation
Overall because our system must plug into the wall, which outputs 120V AC, we run into a power issue for our motors. The issue is that not all motors and microcontrollers run on AC voltage. Thus, we produced ways to connect our electronic components to see what electrical components we may have to buy and how they would be connected. An electronic component that was not included in these drawings but will be necessary is a surge protector. Should power go out while the system is operating, we do not want it to harm anyone, so there will be a surge protector wired from the wall to the AC/DC converter.

2.4.1 Voltage Multiplier

![Concept drawing of a viable way to wire the system](image)

Figure 7: Concept drawing of a viable way to wire the system

First, we must convert the voltage into something the microcontroller can use. Since it runs on 5V DC, we will need to first convert the AC wall voltage into a DC one. Then, we will need to decrease the conversion to 5V and use that to power the microcontroller. The microcontroller will only transmit either “high” which is around 3V or low which is approximately 0 V. So, whatever generates movement could be powered by amplifying the signal coming directly from the microcontroller, this would be done in the voltage multiplier. The issue with this idea is that it only generates two speeds for the motors: on and off. However, we have at least 3 speeds we are emulating: standby, walk, and trot.
2.4.2 Motor Controller

There are a variety of motor controllers available in the market for both AC and DC-powered motors. All they require is a connection to the microcontroller providing digital values corresponding to voltage levels and a connection to the highest voltage possible for the motors. It is a simple solution that would not require us to design a whole new circuit board and will make the parts for the mechanical horse commercially available.

2.5 Programming Ideation
Before we start implementing the code for the mechanical horse as a team, we must first agree on how the code should be structured and how it should work. We chose to ideate in the C programming language because it has little overhead and is usually used for microprocessor coding and as such will likely be the language we end up using in our final design.
In order to organize our code, we first developed a finite state machine (FSM). As can be seen in figure 8, the FSM is a representation of all the states the machine will need to be in and how it can transition from one state to another. We decided figure 8 is how the code should interact with the buttons on the controller. Each state (in a black circle) would hold the code responsible for moving the mechanical horse in a manner that replicates the movement it is named after.

```python
Interrupt_Stop_Button(){
    Set move_slower flag
    Unset interrupt_stop_button flag
}

Interrupt_Walk_Button(){
    If current_motion is stop
        Set move_faster flag
    Else if current_motion is trot
        Set move_slower flag
    Unset interrupt_walk_button flag
}

Interrupt_Trot_Button(){
    Set move_faster flag
    Unset interrupt_trot_button flag
}
```

Figure 9: Possible code implementation for a three-button UI

For our code we use buttons to interact with our control loops by interrupting it. When a button is pressed, it triggers an interrupt, and the code automatically jumps to the code written to resolve that interrupt. Additionally, it is important for the code in the interrupt to execute as swiftly as possible because the entire system is waiting on the interrupt to be resolved. Ideations 2.5.1-2.5.3 rely on the interrupt code found in Figure 9.

2.5.1 No Structure
1. Define standby (0), walk (1), and trot (2)
2. Set current_motion to standby
3. Set a move_slower and move_faster flag to off (0)
4. While horse is on
   a. If current_motion is standby
      i. If move_faster flag set, run the transition to a walk, unset all flags, and set current_motion to walk
      b. If current_motion is walk
         i. If move_slower set, run transition to stop, unset all flags, set current_motion to standby
         ii. If move_faster set, run transition to trot, unset all flags, set current_motion to trot
         iii. If neither of the flags are set, run walk function
   c. If current_motion is trot
      i. If move_slower set, run transition to walk, unset all flags, set current_motion to standby
      ii. Otherwise, run trot function

Figure 10: Possible main function structure
Overall, as can be seen in figure 10, we gave an example of code with no structure. As can be seen there is a lot more writing and steps involved if a finite state machine is written this way. This could make the code harder to read and organize, as each state must hold at least two if statements that are both managing the flags and running the movement code. Additionally, the programmers would have to keep track of the transitions and functions and make sure they are putting them in the right place. Overall, this would make the code messy and harder to update in the future.

2.5.2 Structure Multiple Motion

```c
typedef struct movement {
    void (*pitch_func)(void);
    void (*yaw_func) (void);
    void (*roll_func) (void);
    void (*transition_up) (void);
    void (*transition_down) (void);
} movement;
```

Figure 11: Possible movement code structure

To clean up the code, we decided to create a structure all movements must follow. This way, any continuous improvement must match what we did in some manner. Note, each of these variables (one per line in figure 11) points to a function that will run in our code. That means this structure just outlines what the functions for each motion should look like, in terms of what information they take (void means nothing) and what information they return (once again, void means nothing).

1. Make standby with move_func = NULL, transition_up = stop_to_walk, transition_down = NULL
2. Make walk with move_func = move_walk, transition_up = walk_to_trot, transition_down = walk_to_stop
3. Make trot with move_func = move_trot, transition_up = NULL, transition_down = trot_to_walk
4. Set current_motion to standby
5. Set move_slower and move_faster flags to off
6. While device is on
   a. If move_slower is set
      i. Run current_motion.transition_down()
      ii. Unset all flags
   b. If move_faster is set
      i. Run current_motion.transition_up()
      ii. Unset all flags
   c. If neither is set
      i. Run current_motion pitch, yaw, and roll simultaneously;

Figure 12: Possible main function structure

The structure outlined in Figure 11 would most likely have a main function that looks like the one in Figure 12. As you can see, this is much easier to read and update. When the device is on, only 3 things can be happening and no matter how many speeds there are, and this function wouldn’t have to be changed. However, the implementation of the structure in figure 11 had an issue which is that it is possible the pitch, yaw, and roll functions cannot be isolated, and if they can, may not be able to run on the microcontroller at once.
2.5.3 Structure Singular Motion

```c
typedef struct movement {
    void (*move_func)(void);
    void (*transition_up) (void);
    void (*transition_down) (void);
} movement;
```

Figure 13: Possible Movement code structure

The structure in Figure 13 allows for a much larger range of structuring the movement function as it allows pitch, yaw, and roll to be implemented in any way the developer desires. Additionally, we don’t have to worry about how to run the pitch, yaw, and roll functions simultaneously.

1. Make standby with move_func = NULL, transition_up = stop_to_walk, transition_down = NULL
2. Make walk with move_func = move_walk, transition_up = walk_to_trot, transition_down = walk_to_stop
3. Make trot with move_func = move_trot, transition_up = NULL, transition_down = trot_to_walk
4. Set current_motion to standby
5. Set move_slower and move_faster flags to off
6. While device is on
   a. If move_slower is set
      i. Run current_motion.transition_down()
      ii. Unset all flags
   b. If move_faster is set
      i. Run current_motion.transition_up()
      ii. Unset all flags
   c. If neither is set
      i. Run current_motion.move_func();

Figure 13: Possible main function structure

The main function in Figure 13 behaves the same as the one shown before. The difference is in how the structure is set up. Now, when no flags are set, all we do is run move_func() and let it do its job. The function does require some overhead in the fact that the motions must be defined as global constants, but this should not be much of an issue.

2.6 Controlled Convergence

After producing our ideation, we had to move into choosing our final design direction. To choose our final design direction, we performed a process called controlled convergence. The first step for controlled convergence was to take each of our ideations and place them into distinct functions based on their similarities. We found that for our design we had 5 main functions, the user interface control scheme, the user interface connection, the mover's selection, the code selection, and finally the electrical layout. From there we eliminated many of the ideas that were farfetched or impractical for our design until we were left with about 3-6 ideas. For each of these functions we performed a Pugh matrix analysis, which can be found in Appendix B. In the Pugh matrix analysis, we first decomposed the specifications that we listed in the Scope of Work and then converted them into applicable standards in which to base our decisions on the Pugh Matrices. From there we selected a base design, typically the design that we thought would work the best, and we tested each of the other ideas off of it. In this analysis we specified if we thought that the
design would underperform, outperform, or do the same as the base design. Finally, we tallied up the overall score to see how we our designs lined up against our base design.

Finally, after determining what idea we thought was best we moved into selecting our final design through using several weighted decisions matrices. Overall, since our functions were quite different from each other, we determined that it would not be helpful to place these into concept level designs, but rather we used a weighted decision matrix to determine what we thought was the best solution for each function which can be found in appendix C. To perform this analysis, we first had to do a pairwise comparison on the specifications that we used in our weighted decision matrices. This pairwise comparison allows us to critically think about which specifications we thought were the most important for each function. From there we transmitted these into weights, making sure that all the weights when added together equaled 100, and used these to analyze our current top designs. For this we measured each design based on the specifications and gave it a score from 1 – 10 with 10 being the best. From there we multiplied the score by the weights and then finally summed the total weighted scored together to produce each design's final score. From this we were able to determine which idea was the best from each function and then combine those into our final concept direction. For our design we found that the best solution would be to have a user interface that had two buttons and a speed control knob that was connected through Bluetooth to the device. We also found that the best mover selection was a series of two motors and a linear slider to create the yaw, pitch, and roll. Finally, we determined that the best idea for the code was to make it non modular and combine it with a simple motor controller-based design.

3. Concept Design

Upon finishing our controlled convergence, we collected the top ideas for each function and combined them into a single system design for the therapeutic mechanical horse. As displayed in the figure below, we adapted the ideation with the two motors and linear actuators to better fit our own engineering judgements. Instead of a linear actuator, we decided to replace it with another motor because when attached to a rotating tee holding the other two motors, like in our design, the rotation of the seat is more controlled compared to if the linear actuator was connected to the original design.

Above, in Figure 14, is the isometric drawing for design direction with its main components labelled to aid with the description of the functionality of our design. The overall design is a rotating T-bar with motors attached to each end of the bar located under the seat, as depicted in figure above. The center
motor, Motor 3, is the heavy lifter of this mechanism as it is the main point of rotation for the entire system. As the T-bar rotates it produces a yaw motion to reproduce the motion of a horse swinging its front shoulders back and forth as it walks. The other two motors, Motor 1 and 2, located at the two ends of the T-bar control the two-link arms. As Motor 1 and 2 the two-link arms rotate adjusting the position of the seat to mimic the pitch and roll movements of a horse. When all three motors are working simultaneously, the three movements, yaw, pitch, and roll, horse-like motions can be produced, and the seat can mimic a horse’s walk and trot.

![Isometric Drawing of the Design Direction done in CAD](image1)

**Figure 15: Isometric Drawing of the Design Direction done in CAD**

We plan to have our mechatronics system activated by a user interface with an on and off button and a dial to control the speed of the mechanical horse. When the power button is pressed, a Bluetooth signal will call for electricity from an outlet in a wall to start flowing through our system. The user then can move the dial to their desired speed which sends an input to our system that our code will translate to be compatible with our control systems. The structured code will decide what functions need to be triggered to have the motors execute the correct walking pattern.

![The Concept Prototype for our Design Direction](image2)

**Figure 16: The Concept Prototype for our Design Direction**
To ensure that our intuition and controlled convergence was correct we made a concept prototype with PVC pipe and wooden blocks to act as the motors which can be found in Figure 16. Overall, we found that our design can work as intended provided we select the correct motors to use. Looking forward to our final design, a lot of decisions will be based off what the other senior project team accomplishes in the next few weeks as they start putting together their final design. A lot of our dimensions and material selection are dependent on their final design and as soon as we get the information from them, we are set to start finalizing our design. We have been working closely with the other senior project team and have developed a plan to ensure that our timelines are compatible with one another, but until we know their expected results we can only do so much. However, we have started our search for motors and other electrical components for our mechatronics system so while we wait for the confirmations on hardware, we can get a head start on firmware.

4. Concept Justification

Overall, we think that the design that we have developed will provide the best solution to the design challenge that we face due to its simple yet robust nature. By considering all the distinct functions above we were able to find the solution that best fits each of the specifications that we have laid out for this project. A description of how each function design meets the specifications found in Table 1 can be found below.

<table>
<thead>
<tr>
<th>Spec #</th>
<th>Specification Description</th>
<th>Requirement or Target</th>
<th>Tolerance</th>
<th>Risk</th>
<th>Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Withstand Rider Weight</td>
<td>200 lbs</td>
<td>Min</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>2</td>
<td>Mech Horse Weight</td>
<td>125 lbs (with saddle)</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
<tr>
<td>3</td>
<td>Movement Matches Walk Data</td>
<td>Data within 20%</td>
<td>Target</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>4</td>
<td>Movement Matches Trot Data</td>
<td>Data within 20%</td>
<td>Target</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>5</td>
<td>Accessible Parts</td>
<td>No permanent electronic connections</td>
<td>Target</td>
<td>Low</td>
<td>Inspection</td>
</tr>
<tr>
<td>6</td>
<td>Overall Cost</td>
<td>$500</td>
<td>Max</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>7</td>
<td>Withstands Dust</td>
<td></td>
<td>Target</td>
<td>Medium</td>
<td>Test</td>
</tr>
<tr>
<td>8</td>
<td>Withstands Water</td>
<td></td>
<td>Target</td>
<td>Medium</td>
<td>Test</td>
</tr>
<tr>
<td>9</td>
<td>Minimal Buttons/Switches</td>
<td>3 buttons/switches</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
<tr>
<td>10</td>
<td>Cost Aesthetic Supplies</td>
<td>$20</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
<tr>
<td>11</td>
<td>Functions on Different Surfaces</td>
<td>Data within 30%</td>
<td>Target</td>
<td>Medium</td>
<td>Test</td>
</tr>
<tr>
<td>12</td>
<td>Mech Horse Size</td>
<td>2ft x 4ft x 3ft</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
</tbody>
</table>

Table 1: Engineering Specifications

4.1 Design Specifications

4.1.1 UI Control Scheme

Overall, the UI Control Scheme offers a simple interface that will allow for easy understanding of how to control the system. Because the system itself offers two buttons and a control knob we can easily indicate to the user when the system is on and how to set the speed of the system. This relates to specification 9 where we were able to reduce the number of buttons and switches that are used to control the system. Beyond that, using buttons and a control knob we are also in line with specification 5 using accessible parts as these parts are commercially available.
4.1.2 UI Connection
In addition to being simple, the UI connection allows for a robust connection of the controls of the system to the actual moving parts of the device. Using Bluetooth, the system can communicate to the device over a greater distance that allows for freedom to move around the system during use. Overall, this connection relates to specification 2 as it dramatically reduces the weight of the system by only adding a simple remote. Also, we can satisfy part of specification 11 as having a moveable remote not tied to the ground allows for an easy use above any terrain. Beyond that this Bluetooth connection is in line with specification 5 as Bluetooth devices are simple, intuitive to use, and available all over the place. Next by having a Bluetooth device we can bring down the cost of the overall system as this is a simple connection that we can find at commercial prices which relates to specification 6. Finally, by having a small remote as the connection, we can achieve specification 7 and 8 by making the remote waterproof. This allows the remote device to be able to be placed anywhere and there is no worry that it will be damaged.

4.1.3 Movers
With the expected increase in yaw, pitch, and roll our design for the movement system, we will be able to hit the success criteria for specifications 3 and 4. Although we have not had the opportunity to run any analysis for the moments impacting our current system when we have a rider sitting in the saddle, we already have many ideas to enhance our design to ensure it can withstand the rider’s weight and fulfill specification 1. In addition, we plan to cover our mechatronics system with a removable enclosure to allow the parts to still be accessible when needed but also protect our system from both water and dust. The enclosure would allow our movement system to also cover specifications 5, 6, and 7. The only concern is specification 6 because needing three motors for our system could be expensive but we have countered that by applying for multiple grants to try to lighten the load of motor costs.

4.1.4 Electrical Diagram
The only impact the electrical system has on our specifications is the cost. With a microcontroller that will cost approximately $20 and motor controllers that cost as low as $10. Without the motor, the electrical components will cost approximately $60. This design’s greatest cost will be motors that can move the seat with a rider on it.

4.2 Design Hazards
Part of ensuring safety for both the rider and the instructor is analyzing the possible hazards our design could pose. We filled out the Design Hazard Checklist in Appendix D as a way of exploring the possible dangers inherent in our design. There were only two hazard types that our project could have: pinch points and large moving masses. We will elaborate on where these occur and how we plan to solve them below.

4.2.1 T Joint Pinch Point
Should someone put their hand into the machine while it is operating, they could get hurt by the T joint that is swinging the arm-propelling motors back and forth. To prevent this from happening by accident, the team plans to install a sheet that runs along the top of the frame with only slits cut out for the arms. This addition will also prevent dust and water from reaching the motors and will extend the life of and protect the motors.

4.2.2 Two-Link Arm Pinch Point
During operation, the two-link arms will be pinching together and pulling apart periodically. The elbow joint could harm someone if they put their hand near it while it was moving. To prevent the mechanical horse from hurting anyone, the arm will be padded.
4.2.3 Seat is a Moving Mass
Large moving masses are design hazards as they cannot be easily stopped. Our biggest moving mass is the seat, so to prevent it from causing any harm if it goes haywire, the system will be outfitted with a kill switch that is within reach of the instructor. As a secondary measure, the system can always be unplugged from the wall.

4.3 Current Challenges
While we believe that our design will confidently solve our design challenge there are still a few issues that we need to account for as we design our system. The first issue that we noticed was the significant moment that acts upon the yaw motor. Overall because we have the two pitch and roll motors moving the system with a rider on it, they have to produce a significant amount of force. This force is then translated away from the motors onto the T bar which is then supported by the yaw motor causing a significant moment to act on the motor. Due to this we need to come up with a way to reduce the moment that the motor sees, ideally making it zero, so that we don’t damage the motor. Secondly related to the moment issue we also have to consider the size of the yaw motor and make sure that it will fit into our system. Because the yaw motor must move a significant amount of weight and force, it will need to be fairly large to compensate. However, there is a finite size that we can make our motor, so we need to be careful that we don’t choose a motor that is too large to fit into our system. Finally, the last concern that we have is how to protect the internals and install safety devices. Overall, we need to consider how to protect our electrical components as they may be exposed to dust and water. On top of that we need to make sure that our system includes safety stops in places to allow for a safer riding experience. So, while we think our system will work to solve our design challenge, we must also consider certain key areas of our design and make sure that they work with our system.

5. Project Management
Overall, after choosing and obtaining approval for our design direction we are ready to move into the detailed design where we will start to analyze our system. To start this analysis, we are planning to use our concept prototype that we developed for the preliminary design review to perform early functionality testing. We plan to see how this design will move and function inside the limited space to see if there are any spacing concerns. Also, we want to use this device to test our protection system. In conjunction to this testing, we will be doing engineering calculations to produce our key design criteria that we will then use to inform our parts selection. After completing this design, we will be moving into our Critical Design Review (CDR) where we will obtain approval for our overall fully designed system.

After CDR we will start to purchase our items and build our structural prototype. The first and most crucial step after CDR is purchasing all our materials, especially those that have exceedingly long manufacture and delivery times. After ordering these parts we will move into developing a plan that we will then use to construct our structural prototype. Finally, after receiving all our materials, we will put our structural prototype together and start to perform tests on it to verify that it functions in the way that we expected and solve issues as they occur. A detailed look of this process can be found in the Gannt chart found in Appendix E and a summary of the dates in which these steps need to complete by can be found in Table 2.
<table>
<thead>
<tr>
<th>Milestone</th>
<th>Description</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interim Design Review</td>
<td>Detailed review of design solution and analysis</td>
<td>1/13/21</td>
</tr>
<tr>
<td>Critical Design Review</td>
<td>Detailed review of components, costs, analysis, and proposed solution</td>
<td>2/11/21</td>
</tr>
<tr>
<td>Manufacturing &amp; Testing</td>
<td>Status update of Manufacturing, updated test plan, and updated schedule of project completion</td>
<td>3/10/21</td>
</tr>
<tr>
<td>Prototype Sign Off</td>
<td>Verification of functionality of Final Prototype</td>
<td>4/26/21</td>
</tr>
<tr>
<td>Senior Expo</td>
<td>Display of structural prototype and Expo poster</td>
<td>5/27/21</td>
</tr>
<tr>
<td>Final Design Review</td>
<td>Final review and handoff of prototype and design report</td>
<td>6/3/21</td>
</tr>
</tbody>
</table>

Table 2: Summary of key milestones and descriptions with associated competition dates

6. Conclusion
Overall, the scope of this document was to present our ideation and controlled convergence processes to determine our preliminary design. Through the use of Pugh matrices and weighted decision matrices, we were able to decide the best design for each system and then combine those into a system design. Moving beyond this document we will be working on developing our key design criteria and deciding on our final parts for the preliminary design we created. We will propose these to our sponsor at the Critical Design Review. By writing this document we hope our sponsors attest that the content here is accurate, confirming our understanding of the challenge given to us and the efficacy of our execution plan.
Appendix A – Extra Ideation

A1 User Interface Layout

A1.1 Three Buttons and a Switch

Figure A1: Conceptual build of a UI with an ON/OFF switch and 3 speed buttons

This design lists the three gaits side by side. Additionally, they are listed from slowest to fastest, so they are in a logical order. However, someone could hit a button while reaching for the switch. So, we must ensure our system can exit from any point by being shut down and not corrupt our programming. Which should not happen anyway, but you never know. Also, with the planned coding implementation, if the horse is in standby mode, the worst that could happen is it would transition into a walk before it was shut down. Another issue would be that the UI would have to be re-designed should more functionalities be implemented into the system.

A1.2 Three Buttons

Figure A2: Conceptual drawing of a simple UI with only 3 buttons

The three-button user interface only has three inputs, an off button and a walk and trot button that both acts as an on button for the system. The greatest limitation to this design is the lack of adaptation in the future. With only three buttons to produce inputs, there is no room for other patterns to be added to this
design. Although remarkably simple, there is a lot this design lacks for the criteria we expect from our mechanical horse, so it was decided to go with another option.

A1.3 Two Buttons and a Switch

![Conceptual build of a UI with an ON/OFF switch and two speed buttons](image)

Figure A3: Conceptual build of a UI with an ON/OFF switch and two speed buttons

It is always easy to see whether the horse is on or off, even if it is in standby mode because the ability to turn the mechanical horse on and off will rest in the switch. This design would allow for further improvement of the project without the need to re-design the UI, as both the speed increment and the number of speeds would be dependent on the code written.

The one thing I did notice, is with the buttons being mostly the same shape and color it could be easy to confuse the two. Plus, it requires an extra step while thinking to decide whether the next gait you want is faster or slower than what the mechanical horse is doing right now. And extra logic steps can lead to mistakes.

A2 User Interface Connection:

A2.2 Free Floating Cable

![Chord Connected Interface](image)

Figure A4: Chord Connected Interface
A3 Movers

Figure A5: Four Motor Ideation

Figure A6: Spring and Pulley Ideation

Figure A7: Revolving Shell Interface
### B1 User Interface Pugh

<table>
<thead>
<tr>
<th>User Interface Control Scheme</th>
<th>3 Button + Switch</th>
<th>3 Button</th>
<th>Speed Control Knob</th>
<th>2 Button + Switch</th>
<th>3 Input Knob</th>
<th>2 Button + Speed Control Knob</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to understand/use</td>
<td>S</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Ability to control Speed</td>
<td>S</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Distinct controls</td>
<td>S</td>
<td>S</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Adaptable</td>
<td>S</td>
<td>S</td>
<td>+</td>
<td>S</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>Avoid Unintentional Inputs</td>
<td>S</td>
<td>S</td>
<td>-</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Intuitive</td>
<td>S</td>
<td>+</td>
<td>-</td>
<td>-</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Size</td>
<td>S</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>2</td>
<td>4</td>
<td>-3</td>
</tr>
</tbody>
</table>

Table B1: User interface pugh

### B2 User Interface Connection Pugh

<table>
<thead>
<tr>
<th>User Interface Connection</th>
<th>Bluetooth</th>
<th>Free Floating Cable</th>
<th>Post</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ability to Disconnect during operation</td>
<td>S</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Ability to be misplaced</td>
<td>S</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Size</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Impact on Volunteers</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Impact on System</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Long Term Power Supply</td>
<td>S</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Cost</td>
<td>S</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Ability to Withstand Damage</td>
<td>S</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>0</td>
<td>-2</td>
<td>2</td>
</tr>
</tbody>
</table>

Table B2: User interface connection pugh
### B3 Mover Pugh

<table>
<thead>
<tr>
<th>Max. Weight</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Less Motres</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Achieve Pitch</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Achieve Rolls</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Achieve yaw</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>S</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>No input from Rider</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>S</td>
<td>-</td>
<td>S</td>
<td>S</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Works with other teams design</th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>-</td>
<td>-</td>
<td>S</td>
<td>-</td>
</tr>
</tbody>
</table>

Table B3: Mover pugh

### B4 Electrical Pugh

#### Aleya

<table>
<thead>
<tr>
<th></th>
<th>One Voltage Multiplier</th>
<th>Multiple Voltage Multipliers</th>
<th>Op Amp &amp; Microphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Adaptability</td>
<td>S</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Different Speeds</td>
<td>S</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>+1</td>
<td>-1</td>
</tr>
</tbody>
</table>

Table B4: Aleya’s Electrical Pugh

#### Cade

<table>
<thead>
<tr>
<th></th>
<th>One Voltage Multiplier</th>
<th>Multiple Voltage Multipliers</th>
<th>Op Amp &amp; Microphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>S</td>
<td>-</td>
<td>+</td>
</tr>
<tr>
<td>Adaptability</td>
<td>S</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Different Speeds</td>
<td>S</td>
<td>+</td>
<td>S</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>+1</td>
<td>0</td>
</tr>
</tbody>
</table>

Table B5: Cade’s Electrical Pugh
### Appendix B – Pugh Matrices

**Zuzanna**

<table>
<thead>
<tr>
<th>Feature</th>
<th>One Voltage Multiplier</th>
<th>Multiple Voltage Multipliers</th>
<th>Op Amp &amp; Microphone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Adaptability</td>
<td>S</td>
<td>S</td>
<td>S</td>
</tr>
<tr>
<td>Different Speeds</td>
<td>S</td>
<td>++</td>
<td>S</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

*Table B6: Zuzanna’s Electrical Pugh*

**B5 Coding Pugh**

**Aleya**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Modular Structure</th>
<th>Non-Modular Structure</th>
<th>No Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptable</td>
<td>S</td>
<td>S</td>
<td>-</td>
</tr>
<tr>
<td>Easy to Test</td>
<td>S</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>Easy to Run</td>
<td>S</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Less Global Variables</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Simplicity/ Readability</td>
<td>S</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>+1</td>
<td>-1</td>
</tr>
</tbody>
</table>

*Table B7: Aleya’s Coding Pugh*

**Cade**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Modular Structure</th>
<th>Non-Modular Structure</th>
<th>No Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptable</td>
<td>S</td>
<td>S</td>
<td>-</td>
</tr>
<tr>
<td>Easy to Test</td>
<td>S</td>
<td>S</td>
<td>+</td>
</tr>
<tr>
<td>Easy to Run</td>
<td>S</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Less Global Variables</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Simplicity/ Readability</td>
<td>S</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>1</td>
<td>-1</td>
</tr>
</tbody>
</table>

*Table B8: Cade’s Coding Pugh*

**Zuzanna**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Modular Structure</th>
<th>Non-Modular Structure</th>
<th>No Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptable</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Easy to Test</td>
<td>S</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Easy to Run</td>
<td>S</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Less Global Variables</td>
<td>S</td>
<td>S</td>
<td>-</td>
</tr>
<tr>
<td>Simplicity/ Readability</td>
<td>S</td>
<td>+</td>
<td>-</td>
</tr>
<tr>
<td>Total</td>
<td>0</td>
<td>1</td>
<td>-3</td>
</tr>
</tbody>
</table>

*Table B9: Zuzanna’s Coding Pugh*
## C1 UI Control Decision Matrix

<table>
<thead>
<tr>
<th>Pairwise Comparison</th>
<th>Easy to Understand</th>
<th>Control Speed</th>
<th>Distinct controls</th>
<th>Adaptable</th>
<th>Avoid inputs</th>
<th>Intuitive</th>
<th>Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to understand/use</td>
<td>Understand</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ability to control Speed</td>
<td>Speed</td>
<td>Speed</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distinct controls</td>
<td>-</td>
<td>Speed</td>
<td>Controls</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adaptable</td>
<td>Adaptable</td>
<td>Speed</td>
<td>Adaptable</td>
<td>Adaptable</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Avoid Unintentional Inputs</td>
<td>Understand</td>
<td>Speed</td>
<td>Controls</td>
<td>Adaptable</td>
<td>Inputs</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Intuitive</td>
<td>-</td>
<td>Speed</td>
<td>-</td>
<td>Adaptable</td>
<td>Intuitive</td>
<td>Intuitive</td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td>Understand</td>
<td>Speed</td>
<td>Adaptable</td>
<td>Adaptable</td>
<td>Inputs</td>
<td>Intuitive</td>
<td>Size</td>
</tr>
</tbody>
</table>

| Count | 4 | 7 | 3 | 6 | 2 | 4 | 1 |
| Weight | 14.81 | 25.93 | 11.11 | 22.22 | 7.41 | 14.81 | 3.70 |

### Table C1: UI Control Pairwise Comparison

<table>
<thead>
<tr>
<th>User Interface Control Scheme</th>
<th>Weight</th>
<th>3 Button + Switch Score</th>
<th>3 Button Score</th>
<th>Speed Control Knob Score</th>
<th>3 Input Knob Score</th>
<th>2 Button + Speed Control Knob Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to understand or use</td>
<td>14.81</td>
<td>10</td>
<td>14.8</td>
<td>9</td>
<td>13.3</td>
<td>4.4</td>
</tr>
<tr>
<td>Ability to control Speed</td>
<td>25.93</td>
<td>8</td>
<td>20.7</td>
<td>8</td>
<td>20.7</td>
<td>25.9</td>
</tr>
<tr>
<td>Distinct controls</td>
<td>11.11</td>
<td>10</td>
<td>11.1</td>
<td>10</td>
<td>11.1</td>
<td>10</td>
</tr>
<tr>
<td>Adaptable</td>
<td>22.22</td>
<td>2</td>
<td>4.4</td>
<td>2</td>
<td>4.4</td>
<td>8</td>
</tr>
<tr>
<td>Avoid Wrong Inputs</td>
<td>7.41</td>
<td>8</td>
<td>5.9</td>
<td>8</td>
<td>5.9</td>
<td>5</td>
</tr>
<tr>
<td>Intuitive</td>
<td>14.81</td>
<td>9</td>
<td>13.3</td>
<td>10</td>
<td>14.8</td>
<td>9</td>
</tr>
<tr>
<td>Size</td>
<td>3.70</td>
<td>3</td>
<td>1.1</td>
<td>9</td>
<td>3.3</td>
<td>7</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
<td>71.5</td>
<td>73.7</td>
<td>71.5</td>
<td>79.6</td>
<td>80.4</td>
</tr>
</tbody>
</table>

### Table C2: UI Control Weighted Decision Matrix
### C2 UI Connection Decision Matrix

<table>
<thead>
<tr>
<th>Pairwise Comparison</th>
<th>Cost</th>
<th>Size</th>
<th>Weight</th>
<th>Withstand Environment</th>
<th>Function on Multiple Surfaces</th>
<th>Hard to Misplace</th>
<th>Hard to Disconnect</th>
<th>Lifetime</th>
<th>Easy to Integrate</th>
<th>Safety</th>
<th>Withstand Damage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>C</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td>C</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Weight</td>
<td>W</td>
<td></td>
<td>W</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Withstand Environment</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td>E</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Function on Multiple Surfaces</td>
<td>C</td>
<td>S</td>
<td>W</td>
<td>E</td>
<td>S</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hard to Misplace</td>
<td>C</td>
<td>-</td>
<td>M</td>
<td>E</td>
<td>M</td>
<td>M</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hard to Disconnect during Operation</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Lifetime</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>-</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td>L</td>
<td></td>
<td>L</td>
</tr>
<tr>
<td>Easy to integrate with System</td>
<td>I</td>
<td>I</td>
<td>-</td>
<td>E</td>
<td>I</td>
<td>M</td>
<td>D</td>
<td>L</td>
<td>L</td>
<td></td>
<td>I</td>
</tr>
<tr>
<td>Safety</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td>S</td>
<td></td>
<td>S</td>
</tr>
<tr>
<td>Withstand Damage</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>-</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td>D</td>
<td></td>
<td>S</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Score</th>
<th>4</th>
<th>2.5</th>
<th>4.5</th>
<th>8</th>
<th>1</th>
<th>4</th>
<th>9</th>
<th>9</th>
<th>4</th>
<th>10</th>
<th>8</th>
<th>count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weight</td>
<td>6.25</td>
<td>3.90625</td>
<td>7.03125</td>
<td>12.5</td>
<td>1.5625</td>
<td>6.25</td>
<td>14.0625</td>
<td>14.0625</td>
<td>6.25</td>
<td>15.625</td>
<td>12.5</td>
<td>64</td>
</tr>
</tbody>
</table>

Table C3: UI Connection Pairwise Comparison
## Appendix C – Weighted Decision Matrices

<table>
<thead>
<tr>
<th>User Interface Connection</th>
<th>Weights</th>
<th>Bluetooth with pocket</th>
<th>Score</th>
<th>Free Floating Cable</th>
<th>Score</th>
<th>Post</th>
<th>Score</th>
<th>Post Holder and Cable</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cost</td>
<td>6.25</td>
<td>5</td>
<td>3.13</td>
<td>10</td>
<td>6.25</td>
<td>7</td>
<td>4.375</td>
<td>5</td>
<td>3.125</td>
</tr>
<tr>
<td>Size</td>
<td>3.91</td>
<td>10</td>
<td>6.25</td>
<td>7</td>
<td>4.375</td>
<td>4</td>
<td>2.5</td>
<td>2</td>
<td>1.25</td>
</tr>
<tr>
<td>Weight</td>
<td>7.03</td>
<td>10</td>
<td>6.25</td>
<td>9</td>
<td>5.625</td>
<td>2</td>
<td>1.25</td>
<td>2</td>
<td>1.25</td>
</tr>
<tr>
<td>Withstand Environment</td>
<td>12.50</td>
<td>8</td>
<td>5.00</td>
<td>4</td>
<td>2.5</td>
<td>10</td>
<td>6.25</td>
<td>10</td>
<td>6.25</td>
</tr>
<tr>
<td>Function on Multiple Surfaces</td>
<td>1.56</td>
<td>10</td>
<td>6.25</td>
<td>8</td>
<td>5</td>
<td>7</td>
<td>4.375</td>
<td>7</td>
<td>4.375</td>
</tr>
<tr>
<td>Hard to Misplace</td>
<td>6.25</td>
<td>2</td>
<td>1.25</td>
<td>7</td>
<td>4.375</td>
<td>10</td>
<td>6.25</td>
<td>10</td>
<td>6.25</td>
</tr>
<tr>
<td>Hard to Disconnect during Operation</td>
<td>14.06</td>
<td>6</td>
<td>3.75</td>
<td>1</td>
<td>0.625</td>
<td>10</td>
<td>6.25</td>
<td>10</td>
<td>6.25</td>
</tr>
<tr>
<td>Lifetime</td>
<td>14.06</td>
<td>4</td>
<td>2.50</td>
<td>5</td>
<td>3.125</td>
<td>10</td>
<td>6.25</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>Easy to integrate with System</td>
<td>6.25</td>
<td>10</td>
<td>6.25</td>
<td>8</td>
<td>5</td>
<td>2</td>
<td>1.25</td>
<td>2</td>
<td>1.25</td>
</tr>
<tr>
<td>Safety</td>
<td>15.63</td>
<td>10</td>
<td>6.25</td>
<td>4</td>
<td>2.5</td>
<td>6</td>
<td>3.75</td>
<td>5</td>
<td>3.125</td>
</tr>
<tr>
<td>Withstand Damage</td>
<td>12.50</td>
<td>5</td>
<td>3.13</td>
<td>2</td>
<td>1.25</td>
<td>10</td>
<td>6.25</td>
<td>9</td>
<td>5.625</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td>100</td>
<td><strong>50.00</strong></td>
<td><strong>40.625</strong></td>
<td><strong>48.75</strong></td>
<td><strong>43.75</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table C4: User Interface Connection weighted decision matrix
### C3 Mover Decision Matrix

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Max Weight</th>
<th>Less Motors</th>
<th>Generate Pitch</th>
<th>Generate Roll</th>
<th>Generate Yaw</th>
<th>No Rider Input Needed</th>
<th>Works with Other Team's Design</th>
<th>Safety</th>
<th>Cost</th>
<th>Manufacturability</th>
<th>Lifespan</th>
<th>TOTAL</th>
<th>weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Weight</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>12.30</td>
</tr>
<tr>
<td>Less Motors</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>3.28</td>
</tr>
<tr>
<td>Generate Pitch</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>9.84</td>
</tr>
<tr>
<td>Generate Roll</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>9.84</td>
</tr>
<tr>
<td>Generate Yaw</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>9.84</td>
</tr>
<tr>
<td>No Input from Rider Needed</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.74</td>
</tr>
<tr>
<td>Works with Other Team's Design</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5.74</td>
</tr>
<tr>
<td>Safety</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>3.5</td>
</tr>
<tr>
<td>Cost</td>
<td>0.5</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>3.5</td>
</tr>
<tr>
<td>Manufacturability</td>
<td>0.5</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>1</td>
<td>1</td>
<td>5.74</td>
</tr>
<tr>
<td>Lifespan</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>13.11</td>
</tr>
<tr>
<td>TOTAL score</td>
<td>7.5</td>
<td>2</td>
<td>6</td>
<td>6</td>
<td>3.5</td>
<td>3.5</td>
<td>8</td>
<td>7</td>
<td>4</td>
<td>7.5</td>
<td>7.5</td>
<td>61</td>
<td>61</td>
</tr>
<tr>
<td>weight</td>
<td>12.30</td>
<td>3.28</td>
<td>9.84</td>
<td>9.84</td>
<td>5.74</td>
<td>5.74</td>
<td>13.11</td>
<td>11.48</td>
<td>6.56</td>
<td>12.30</td>
<td>100</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

Table C5: UI Control Pairwise Comparison
### Appendix C – Weighted Decision Matrices

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Weight</th>
<th>2 Motors</th>
<th>Score</th>
<th>2 Motors with twisting shaft</th>
<th>Score</th>
<th>2 Motor with Springs</th>
<th>Score</th>
<th>2 Motors with Linear Slider</th>
<th>Score</th>
<th>4 Motors</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>Max Weight</td>
<td>12.30</td>
<td>10</td>
<td>12.30</td>
<td>8</td>
<td>9.84</td>
<td>7.5</td>
<td>9.22</td>
<td>10</td>
<td>12.30</td>
<td>7.5</td>
<td>9.22</td>
</tr>
<tr>
<td>Less Motors</td>
<td>3.28</td>
<td>10</td>
<td>3.28</td>
<td>7.5</td>
<td>2.46</td>
<td>7.5</td>
<td>2.46</td>
<td>7.5</td>
<td>2.46</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>Generate Pitch</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>9.84</td>
<td></td>
</tr>
<tr>
<td>Generate Roll</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>0</td>
<td>9.84</td>
</tr>
<tr>
<td>Generate Yaw</td>
<td>9.84</td>
<td>5</td>
<td>4.92</td>
<td>10</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>10</td>
<td>9.84</td>
<td>0</td>
<td>0.00</td>
</tr>
<tr>
<td>No Input from Rider Needed</td>
<td>5.74</td>
<td>10</td>
<td>5.74</td>
<td>10</td>
<td>5.74</td>
<td>0</td>
<td>0.00</td>
<td>10</td>
<td>5.74</td>
<td>10</td>
<td>5.74</td>
</tr>
<tr>
<td>Works with Other Team's Design</td>
<td>5.74</td>
<td>10</td>
<td>5.74</td>
<td>5</td>
<td>2.87</td>
<td>5</td>
<td>2.87</td>
<td>10</td>
<td>5.74</td>
<td>2.5</td>
<td>1.43</td>
</tr>
<tr>
<td>Safety</td>
<td>13.11</td>
<td>10</td>
<td>13.11</td>
<td>7.5</td>
<td>9.84</td>
<td>5</td>
<td>6.56</td>
<td>10</td>
<td>13.11</td>
<td>8</td>
<td>10.49</td>
</tr>
<tr>
<td>Cost</td>
<td>11.48</td>
<td>10</td>
<td>11.48</td>
<td>7.5</td>
<td>8.61</td>
<td>5</td>
<td>5.74</td>
<td>8</td>
<td>9.18</td>
<td>4</td>
<td>4.59</td>
</tr>
<tr>
<td>Manufacturability</td>
<td>6.56</td>
<td>10</td>
<td>6.56</td>
<td>5</td>
<td>3.28</td>
<td>6</td>
<td>3.93</td>
<td>10</td>
<td>6.56</td>
<td>6</td>
<td>3.93</td>
</tr>
<tr>
<td>Lifespan</td>
<td>12.30</td>
<td>9</td>
<td>11.07</td>
<td>8</td>
<td>9.84</td>
<td>8</td>
<td>9.84</td>
<td>9</td>
<td>11.07</td>
<td>4</td>
<td>4.92</td>
</tr>
<tr>
<td>TOTAL</td>
<td>100.00</td>
<td></td>
<td></td>
<td>93.85</td>
<td>81.97</td>
<td></td>
<td>70.12</td>
<td>95.66</td>
<td></td>
<td>60.00</td>
<td></td>
</tr>
</tbody>
</table>

Table C6: UI Control Weighted Decision Matrix

### C4 Coding Decision Matrix

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Adaptable</th>
<th>Easy to Test</th>
<th>Easy to Run</th>
<th>Less Global Variables</th>
<th>Simplicity/Readability</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptable</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Easy to Test</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>0</td>
<td>0.5</td>
</tr>
<tr>
<td>Easy to Run</td>
<td>0</td>
<td>0</td>
<td>0.5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Less Global Variables</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>Simplicity/Readability score</td>
<td>0.5</td>
<td>0.5</td>
<td>1</td>
<td>0.5</td>
<td>0.5</td>
</tr>
<tr>
<td>weight</td>
<td>20</td>
<td>20</td>
<td>36</td>
<td>8</td>
<td>16</td>
</tr>
</tbody>
</table>

Table C7: Pairwise Comparison
<table>
<thead>
<tr>
<th>Criteria</th>
<th>weight</th>
<th>Modular Structure</th>
<th>Non-Modular Structure</th>
<th>No Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptable</td>
<td>20</td>
<td>0.9</td>
<td>0.7</td>
<td>0.5</td>
</tr>
<tr>
<td>Easy to Test</td>
<td>20</td>
<td>0.8</td>
<td>0.7</td>
<td>0.3</td>
</tr>
<tr>
<td>Easy to Run</td>
<td>36</td>
<td>0.2</td>
<td>0.9</td>
<td>0.9</td>
</tr>
<tr>
<td>Less Global Variables</td>
<td>8</td>
<td>0.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td>Simplicity/Readability</td>
<td>16</td>
<td>0.7</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td>Score</td>
<td>100</td>
<td>59.6</td>
<td>82</td>
<td>53.2</td>
</tr>
</tbody>
</table>

Table C8: Weighted Decision Matrix
### Appendix D – Design Hazard Checklist

For any “Y” responses, the following is included on the next page:

1. a complete description of the hazard,
2. the corrective action(s) you plan to take to protect the user, and
3. a date by which the planned actions will be completed.

<table>
<thead>
<tr>
<th>Y/N</th>
<th>Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>1. Will any part of the design create hazardous revolving, reciprocating, running, shearing, punching, pressing, squeezing, drawing, cutting, rolling, mixing or similar action, including pinch points and shear points?</td>
</tr>
<tr>
<td>X</td>
<td>2. Can any part of the design undergo high accelerations/decelerations?</td>
</tr>
<tr>
<td>X</td>
<td>3. Will the system have any large moving masses or large forces?</td>
</tr>
<tr>
<td>X</td>
<td>4. Will the system produce a projectile?</td>
</tr>
<tr>
<td>X</td>
<td>5. Would it be possible for the system to fall under gravity creating injury?</td>
</tr>
<tr>
<td>X</td>
<td>6. Will a user be exposed to overhanging weights as part of the design?</td>
</tr>
<tr>
<td>X</td>
<td>7. Will the system have any sharp edges?</td>
</tr>
<tr>
<td>X</td>
<td>8. Will any part of the electrical systems not be grounded?</td>
</tr>
<tr>
<td>X</td>
<td>9. Will there be any large batteries or electrical voltage in the system above 40 V?</td>
</tr>
<tr>
<td>X</td>
<td>10. Will there be any stored energy in the system such as batteries, flywheels, hanging weights or pressurized fluids?</td>
</tr>
<tr>
<td>X</td>
<td>11. Will there be any explosive or flammable liquids, gases, or dust fuel as part of the system?</td>
</tr>
<tr>
<td>X</td>
<td>12. Will the user of the design be required to exert any abnormal effort or physical posture during the use of the design?</td>
</tr>
<tr>
<td>X</td>
<td>13. Will there be any materials known to be hazardous to humans involved in either the design or the manufacturing of the design?</td>
</tr>
<tr>
<td>X</td>
<td>14. Can the system generate elevated levels of noise?</td>
</tr>
<tr>
<td>X</td>
<td>15. Will the device/system be exposed to extreme environmental conditions such as fog, humidity, cold, hot temperatures, etc?</td>
</tr>
<tr>
<td>X</td>
<td>16. Is it possible for the system to be used in an unsafe manner?</td>
</tr>
<tr>
<td>X</td>
<td>17. Will there be any other potential hazards not listed above? If yes, please explain on the reverse.</td>
</tr>
<tr>
<td>Description of Hazard</td>
<td>Planned Corrective Action</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Rotating T joint can pinch or hit people if they touch it</td>
<td>There will be a sheet that goes along the top of the frame covering the moving components of the horse. This will reduce the amount of moving components that a user could interact with directly.</td>
</tr>
<tr>
<td>2-link arms have a pinch point at the elbow</td>
<td>The 2-link arms will be padded to prevent them from pinching anyone should they be touched during operation.</td>
</tr>
<tr>
<td>Seat is a large moving mass</td>
<td>The system will be outfitted with a kill switch that is within reach of the instructor should the system behave strangely or otherwise be required to shut off in an emergency.</td>
</tr>
</tbody>
</table>
### F76 Therapeutic Mechatronic...

**Scope of Work**

<table>
<thead>
<tr>
<th>Task</th>
<th>Hours</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDR</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Ideation</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Brainstorm/Ideation Creation</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Incorporation of Applicable Techno...</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Build Concept Models</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Evaluation</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Evaluate Ideas</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Pugh Matrices</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Design Weighted Decision Matrix</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Preliminary Analysis</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Perform Preliminary Analysis</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Concept Prototype</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Design Concept Prototype CAD</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Build Concept Prototype</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Preliminary Design Review Prese...</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Prepare PDR Presentation</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>Write PDR Report</td>
<td>0</td>
<td>100%</td>
</tr>
<tr>
<td>IDR</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Design Analysis</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Perform Design Analysis</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Perform FMEA</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Perform D/M/A</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Design CAD</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Perform IDR Presentation</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>CDR</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Detailed CAD</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Finalize Detailed CAD</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Structural Prototype</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Build Structural Prototype</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Finalize Parts</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Finalize Parts</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Draft out Detailed Manufacturing</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Draft out Detailed Manufacturing P...</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Write CDR report</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Write CDR report</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Rework CDR</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Send CDR to Sponsor</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Perform CDR presentation</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Perform Risk Assessment</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Manufacturing</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Order Parts</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Perform Manufacturing</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Draft Operations Manual</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Perform Experimental Testing Design</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Manufacturing and Testing Review</td>
<td>0</td>
<td>0%</td>
</tr>
<tr>
<td>Verification Prototype Sign-Off</td>
<td>0</td>
<td>0%</td>
</tr>
</tbody>
</table>
Critical Design Review

California Polytechnical San Luis Obispo
Senior Project 2021 - 2022

Aleya Dolorfino
adorfii@calpoly.edu

Zuzanna Dominik
zdominik@calpoly.edu

Cade Liberty
cliberty@calpoly.edu

Peter Phillips
pphill03@calpoly.edu

Luke Watts
luwatts@calpoly.edu
Critical Design Review

Therapeutic Mechanical Horse

California Polytechnical San Luis Obispo
Senior Project 2021 - 2022

Aleya Dolorfino
adorfi@calpoly.edu

Zuzanna Dominik
zdominik@calpoly.edu

Cade Liberty
cliberty@calpoly.edu

Peter Phillips
pphill03@calpoly.edu

Luke Watts
luwatts@calpoly.edu
Abstract
Jack’s Helping Hand, Equine therapy centers, and participants require a device to serve as an alternative to a live horseback riding experience and increase the range of riders. This would provide more clients with equine-assisted therapy that has proven to better the lives of people with both physical and mental disabilities. Horses can be unpredictable, tall, and sometimes anxiety-inducing, especially for new riders. Our group aims to develop a mechanical horse that will be able to reduce these issues for equine therapy centers and the riders they help. When a rider gets to practice sitting on the horse without the unpredictability or the height, the rider can develop confidence in riding before sitting on a real horse. After going out into the field to experience horse riding for ourselves and translating our experience into the design for our therapeutic mechanical horse, we now have a more solid model for our final design of the mechanical horse. In this document, we describe our final design, justify our design decisions, and present our plan to manufacture and assemble the final product.
Table of Contents

Abstract ......................................................................................................................................................... 1
1. Introduction .................................................................................................................................................... 1
2. System Design ........................................................................................................................................... 2
   2.1 Major Subsystems and Components ................................................................................................. 2
      2.1.1 Arms Carts ................................................................................................................................... 2
      2.1.2 Yaw Motor and T-Bar .............................................................................................................. Error! Bookmark not defined.
      2.1.3 Baseplate ...................................................................................................................................... 3
      2.1.4 Electronics – On The Horse ......................................................................................................... 4
      2.1.5 Electronics – In The Controller ................................................................................................... 5
      2.1.5 Controller Housing ....................................................................................................................... 6
      2.1.4 Programming ................................................................................................................................ 6
3. Design Justification ....................................................................................................................................... 6
   3.1 Analyses .................................................................................................................................................. 7
   3.2 Similarity to Existing Designs ............................................................................................................... 9
      3.2.1 Motors .......................................................................................................................................... 9
      3.2.2 Motor Controller .......................................................................................................................... 9
      3.2.3 Power Supply ............................................................................................................................... 9
      3.2.4 Controller Electronics ................................................................................................................ 9
   3.4 Prototype Tests ..................................................................................................................................... 9
4. Manufacturing Plan ...................................................................................................................................... 10
   4.1 Procurement ....................................................................................................................................... 10
   4.2 Manufacturing ................................................................................................................................... 10
      4.2.1 Arm Carts ................................................................................................................................... 10
      4.2.2 T Bar ......................................................................................................................................... Error! Bookmark not defined.
      4.2.3 Baseplate .................................................................................................................................... 11
      4.2.4 Controller .................................................................................................................................. 11
      4.2.5 Horse Electronics ...................................................................................................................... 11
   4.3 Assembly ........................................................................................................................................... 11
5. Design Verification Plan ............................................................................................................................ 11
   5.1 Evaluating Specs ................................................................................................................................. 11
   5.2 Planned Tests ..................................................................................................................................... 12
6. Conclusion .................................................................................................................................................. 13
Appendix A – Bill of Materials .................................................................................................................. A- Error! Bookmark not defined.
Appendix B – Drawing and Spec Packages ................................................................. B-1
Appendix C – Design Verification Plan ................................................................. C-1
Appendix D – Gantt Chart .................................................................................. D-1
Appendix E – Purchased Materials ................................................................. E-1
Appendix F – Code Organization UML .......................................................... F-1
1. Introduction

Jack’s Helping Hand is a non-profit organization which provides children with cancer or special needs under the age of 21 with community programs focused on enriching their lives. One of their enrichment programs, Little Riders, involves Equine therapy to help riders improve balance, strength, and coordination, which benefit the participants in a multitude of ways. While Equine therapy is a valuable resource for children with special needs, there are some limitations, such as the wear on lesson horses, low rider weight limits, and the varying levels of confidence and experience of riders, that prevent the Little Riders program to be accessible to more children. Due to this our sponsors, Mrs. Orradre, chairperson and executive director, and Mrs. Burt, volunteer and event Coordinator, from Jack’s Helping Hand, reached out to Cal Poly with the hope to develop a device to serve as an alternative to a live horseback riding experience and increase the range of riders. This would provide more clients with equine-assisted therapy that has been proven to better the lives of people with both physical and mental disabilities.

To achieve making such a device, Cal Poly’s Mechanical Engineering Department assigned our team to plan, design, and fabricate the mechatronic system for a therapeutic mechanical horse frame being built by another senior project team. Our team consists of five members:

- Aleya Dolorfino: 4th year mechanical engineering major with a concentration in mechatronics
- Zuzanna Dominik: 4th year computer engineering major
- Cade Liberty: 4th year mechanical engineering major with a concentration in mechatronics
- Peter Phillips: 4th year computer science major
- Luke Watts: 4th year computer science major

We saw this project as a wonderful opportunity to utilize all the knowledge we have acquired over the past three years to make a significant impact on the community. We share a passion for helping others and are extremely excited to work on the Therapeutic Mechanical Horse Team.

The purpose of this document is to clearly describe our final design plan and how we plan to fully manufacture and assemble the final product for the Therapeutic Mechanical Horse Project. Included in this document is the analysis we completed to develop our final design and detailed explanation and justification of our chosen design. In addition, we share a complete plan of the next steps we will take to complete the fabrication of the final product for the Therapeutic Mechanical Horse Project.
2. System Design

Overall, our system contains 4 major subsystems that all work together to create the motion of the horse. The connection for all 4 of the systems to the frame can be found in Figure 1. The subsystems are the arm motor assembly, the internal electronics, the controller, and the baseplate. All these subsystems, except for the controller, which can be moved independently of the horse itself, are contained in the horse device and function in tandem to create the motion of the horse. Each following subsection will go over in more detail the design of each subsystem and the functions that it provides for the horse.

Figure 1: Full Mechatronic Assembly with Frame, Seat, Arm Motor Assembly, and Controller Shown

2.1 Major Subsystems and Components

The purpose of the therapeutic mechanical horse is to mimic the movements of a horse through producing pitch, roll, and yaw motions. This can be achieved with a movement system that is responsible for jostling the seat of the mechanical horse. Since there are many ways to achieve movement for the mechanical horse system, ideation was needed to narrow the options of the movement systems. During ideation and controlled convergence, we focused on two main goals for our design. Firstly, to make sure it could be easily implemented into the current design of the other senior project team. Secondly, that our design produced a large motion in the pitch and roll directions.

2.1.1 Arm Motors
Figure 2: Arm Motor Assembly with Motor, Gear Box, Bearing, and Holding Clips

Overall, the arm motor assembly is made of 4 components and come together to produce the roll and pitch and to a slight extent yaw motion for the seat. There will also be 2 of these systems on the horse, one on either side of the seat. The 4 components are the motor, the gearbox, the bearing, and the holding clips, which can be found in Figure 2. The motor is a Teknic ClearPath motor that can produce 9.5 lbf ft of Torque. However, since we need much more torque than that we used a 15:1 gearbox to increase the Torque. We then applied a pillow block bearing between the gearbox and the link shaft which composes the arm to take any radial load that may be applied to the gearbox and motor in order to prevent wear or damage to those devices. For location these components will sit on top of the bottom base plate and as close to possible to the side walls of the baseplate. Overall, this is one of our most expensive subsystems reaching about $500. The reason for this is because both the motor costs $338 and the gearbox costs $88.

2.1.3 Baseplate

The therapeutic mechanical horse is an outdoor mechanism that will be exposed to the elements while in use. To avoid the elements from interrupting or damaging our mechanical horse, it was necessary for us to waterproof and dust proof the horse’s mechatronic system. We did this by developing a baseplate to function as a shield for both the motors and the electrical components for our mechatronics system. The baseplate consists of a bottom plate cut in half to fit around the gusset and have a small notch where the bump stop attaches to the base frame, and a top plate cut in half with one have containing 3 cutouts, one for the bump stops and two for where the motor arms are located as well as four side panels. A
Solidworks model of the baseplate can be seen below in Figure 4.

Figure 4: Baseplate Assembly

The bottom plate will be two pieces cut from the same solid piece that covers the entire 3” x 5” frame. It is going to be bolted down into the frame the other senior project team already manufactured. Its purpose is to be the initial barrier between the ground and the mechatronics system. It provides a fully waterproof and dust-proof layer so no damage can come from underneath the mechanical horse. The top plate would consist of two parts each covering half of the frame. The half over the electrical components would be a solid piece to fully waterproof and dust-proof that side. Then, the half over the motors and other mechanical components will have a cutout for each arm to allow the arms free movement as the mechatronics system moves. Having the cutout gives water and dust an access point to our mechatronics system but to counter this we plan to attach loose waterproof fabric on the moving arms. The waterproof fabric is not rigid which means that it can move as the arms move. This fluidity with the fabric proves helpful because no matter where the arms are located the cutout will be fully covered by the fabric.

2.1.4 Internal Electronics

Electronics – On The Horse

Figure 5: Electrical Wiring Diagram for Horse System

Overall the electronics section consists of two different parts, one on the horse and the other in the controller. As shown in Figure 5 the electronics on the horse contain two electrical systems, a 24V DC system and a 75V DC system. The Teknic IPC-5 power supply will supply 75V DC power through daisy chained connectors in each Teknic ClearPath motor, while the 24V power supply will supply logical power to the motor controller. These will both be plugged into a surge protector to protect the system.
against the power supplies drawing too much current and protect the system from power grid surges. All power will be able to be cut from the system by hitting the emergency stop button which will sever power to all devices in case of emergency. A Teknic ClearCore motor controller will serve as both as a controller for the movement of the motors and as a Web Server for accepting Hypertext Transfer Protocol (HTTP) requests over a representational state transfer (REST) application programming interface (API) for the controller client over peer-to-peer Wi-Fi, essentially allowing for us to communicate to the controller through Wi-Fi. 16 Gauge wire will be used for all 75V DC cables, and 22 Gauge wire will be used in 24V DC connections and motor controller signaling wiring.

2.1.5 Controller Electronics – In The Controller

As for the section on the controller we are going to be placing all of this inside a small 3D printed enclosure that allows for handheld control of the horse motion. As shown in Figure 6 the controller will be based around a WEMOS D32 microcontroller which houses an ESP-32 microcontroller with an embedded Bluetooth and Wi-Fi Modem. To allow the controller to wirelessly send commands to the mechanical horse system, the ESP-32 will act as a web client connected to the horse’s motor controller peer-to-peer Wi-Fi web server and will broadcast HTTP requests over the REST API. The controller will be powered by a Lithium-Ion battery which will be able to be charged via Micro-USB. Also included on the device is a power switch to turn the controller on and off. Connected to the WEMOS D32 there will be a potentiometer with 10 detents (or positions) to select the current movement mode of the horse. There will be a stop button that will immediately stop the motion of the horse in any mode and the potentiometer will need to be reset to the off position before the horse starts to move again. A status LED will also be connected to the WEMOS and will display a solid light if the connection is stable and will flash in the event of a disconnect. We will use 22 Gauge wire to wire all components for the controller.
2.1.5 Controller Housing

![Figure 7: Controller Casing](image)

The housing of the controller will be a simple 3D printed box as shown in figure 7. The potentiometer will be situated on the front face with labels for each mode the horse can be in with stop mode being at the left. Above the potentiometer will be a stop button that will stop the motion of the horse no matter the position of the of the potentiometer and in order to get the horse back in motion the potentiometer will have to be move back to the stop position and then into the desired mode. Above the button will be a simple status LED which will flash if the controller in not connected to the horse and be solid if the controller is connected properly. On the side on the controller will be a power switch to turn on and off the controller. Also on the side will be a micro-USB charging port to recharge the controller. The electronics will be closed into the box with a sliding back panel.

2.1.4 Programming

To create the oscillations the seat undergoes, we will need to organize how the controller sends signals to the horse. The full code organization is in Appendix F, which outlines how a potentiometer input will be transmitted through the pipeline and eventually provide a state for the motors. Each movement pattern will most likely have an array of states that it must iterate through in order to accurately conform to the movement.

3. Design Justification

Part of the design process is ensuring our system will successfully meet the requirements we and our sponsors set at the beginning of the design process. As a reminder, the standards we wish to meet have been included below in Table 1. Overall due to framing concerns which were outside the scope of our project the rider's weight was reduced to 160 lbs rather than 200 lbs. On top of that we also got more gain more funding so our overall budget increased from $500 to $2000.

<table>
<thead>
<tr>
<th>Spec #</th>
<th>Specification Description</th>
<th>Requirement or Target</th>
<th>Tolerance</th>
<th>Risk</th>
<th>Compliance</th>
</tr>
</thead>
</table>
Table 1: Engineering Specifications

<table>
<thead>
<tr>
<th></th>
<th>Withstand Rider Weight</th>
<th>200 lbs</th>
<th>Min</th>
<th>High</th>
<th>Analysis</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Mech Horse Weight</td>
<td>125 lbs (with saddle)</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
<tr>
<td>3</td>
<td>Movement Matches Walk Data</td>
<td>Data within 20%</td>
<td>Target</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>4</td>
<td>Movement Matches Trot Data</td>
<td>Data within 20%</td>
<td>Target</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>5</td>
<td>Accessible Parts</td>
<td>No permanent electronic connections</td>
<td>Target</td>
<td>Low</td>
<td>Inspection</td>
</tr>
<tr>
<td>6</td>
<td>Overall Cost</td>
<td>$500</td>
<td>Max</td>
<td>High</td>
<td>Analysis</td>
</tr>
<tr>
<td>7</td>
<td>Withstands Dust</td>
<td>Target</td>
<td>Medium</td>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Withstands Water</td>
<td>Target</td>
<td>Medium</td>
<td>Test</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Minimal Buttons/Switches</td>
<td>3 buttons/switches</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
<tr>
<td>10</td>
<td>Cost Aesthetic Supplies</td>
<td>$20</td>
<td>Max</td>
<td>Low</td>
<td>Analysis</td>
</tr>
<tr>
<td>11</td>
<td>Functions on Different Surfaces</td>
<td>Data within 30%</td>
<td>Target</td>
<td>Medium</td>
<td>Test</td>
</tr>
<tr>
<td>12</td>
<td>Mech Horse Size</td>
<td>2ft x 4ft x 3ft</td>
<td>Max</td>
<td>Medium</td>
<td>Inspection</td>
</tr>
</tbody>
</table>

### 3.1 Analyses

Overall, we have performed many different analyses to ensure that our system would be able to hit all our goals. The first and most critical analysis that we performed was regarding specifications 1, 3, and 4. Overall we used a Force Body Diagram (FBD) along with a Kinetic Diagram (KD) to create the necessary force that the arm bars that connect the seat to the motor carts needed to supply to bring our rider back from our worst-case positions. We determined that our worst-case positions were found in 2 spots, first with the seat pitched all the way forward to its max pitch and the rider sitting in the front leaning in line with the seat and second for the roll case the seat rolled all the way to its max roll angle with the rider similarly leaning in line with the angle of the seat. We chose these as our worst-case loadings because they represent the maximum amount of force that the system will need to supply. We chose to have the rider leaning in line with the seat because our sponsor informed us that many of the people using this device will be new to hippotherapy sessions and may not be able to hold themselves upright like an experienced horse rider could. Therefore, we figure that these would be our realistic worst-case loadings and allow for us to design a robust system.

From these loading cases we performed a series of hand calculations to figure out the torque needed to bring the rider back from each case by our motors. In order to ensure that our motor would be able to supply enough torque we assumed that our rider was weighed our maximum allowable weight which was 160 lbs. From there we used a Safety Factor of 3 to determine the loads on our system which comes from the standards we researched at the beginning of this project. This standard comes specifically from toys for children but is the most applicable standard for our use case thus being a good representation of our system. Through these considerations we found that our worst-case loading came from our pitch case requiring our motor to produce about 90 lbf-ft of torque for the arm motors. We also performed a calculation to find the torque required for the Yaw motor. For this analysis we used another FBD and KD this time on the motor and T bar to determine what the torque needed to be for this scenario. From our calculations we found that this yaw motor would need to supply about 15 lbf-ft of torque.

Using these two worse case scenarios we were able to find and select motors and gearbox combinations that were able to manage these torques. For the Arm motor we selected a CPM- MCVC-3441S-RLN ClearPath Motor with a 15:1 Gear box. The motor provides 9.5 lbf-ft of torque and the gearbox has a 15:1 ratio thus increasing the maximum torque to 142.5 lbf-ft which is well above our needed torque. For the yaw motor we chose to go with a smaller ClearPath Motor which only supplies 5 lbf-ft of torque and used
a 4:1 right angle reducer to increase the torque. The reason for using a right-angle gear box is for ease of assembly as with a concentric gearbox we would have to place both components on the steel post of our system which would be rather hard to do. Thus, having the right-angle gearbox allows us to place both components on our baseplate which is simpler and easier to do. Finally, the gearbox and motor together can produce 25 lbf-ft of torque which is again well above our necessary torque. We would also like to clarify that the torque values we produce need to be peak torque values, however we do not plan our motor to have to run close to their peak torque values as that can damage them. Thus, we need a comfortable overshoot to properly run our motors to ensure that they do not break.

In addition to the torque calculation analysis, we also performed sensor data analysis. To ensure the movements of our mechanical horse would mimic the movement of an actual horse we needed to have data from a real horse. We used our phones with the MATLAB application downloaded as sensors to collect data for the orientation, position, and acceleration of both the rider and the horse. Specifically, orientation measured the change of the yaw, pitch, and roll in degrees over time, acceleration measured the linear accelerations in the X, Y, and Z directions, and position recorded the longitudes, latitudes, and elevations during the ride. Data from both the orientation and acceleration were implemented into the torque calculations but the position data did not seem to be immensely helpful in our design. Aside, from using the data in the torque calculations, we also translated orientation and acceleration data for both the rider and the horse into graphs so we would have a visual for the how a horse move. We first had to smooth the data using libraries from MATLAB. Then, we trimmed the data to a section that showed a consistent movement pattern. The trimmed data points were then plotted and seen in Figure 8. We plan to use the graphs we generated in the upcoming quarter to compare our design movement to. If we can have our final system’s movement data match with the data collected from the actual horse, then we know our design is a success. The graphs below in Figure 8 is of the orientation of the horse over time. The graphs represent the change in yaw, pitch, and roll over time and are what we plan to compare our future final system’s movement to. Finally, after taking into consideration our budget and the relative amount of yaw that is produced by our system due to roll and talking with our sponsor we decided to move the arm motors out to the side of the system. The reason for this shifting is to allow the motors to produce yaw which comes from the roll action. Overall, the motors are placed 1.5 inches away from the sides of the baseplate and in line with the connection of the seat. We place the motors here for two reasons. First, we found that a certain amount of yaw is produced when rolling and this motion is enhanced the farther out the motors are. Next, we placed them 1.5 inches in from the sides as to offer the motors protections. Since these components are going to be moving there is concern that they can shift due to vibration. Also, there is a hazard if someone were to kick the side baseplate. Thus, we kept it sufficiently in to avoid both of these issues while also enhancing out yaw motion as much as possible.

Figure 8: Three Graphs showing the yaw, pitch, and roll change over a five second time interval.
3.2 Similarity to Existing Designs
While there are few examples of existing horse simulators, and all of their designs exceed our project’s budget. Because of this we were unable to model our mechatronic design from the existing horse simulators. Most existing horse simulators have at least four motors. For example, the Miracolt has a motor for each of the “legs” of their mechanical horse. Sadly, it is just not within our budget to recreate something like that. So instead, the only inspiration we took from the existing horse simulator were on the structural side of the design and not the mechatronics side.

3.3 Electrical Justification
There are many reasons as to why we chose the electronics we will be using in the final system. It is important to have electronics that are compatible and do not require conversions for each step. As a result, our controllers, power supplies, and motors all come from the same company, so we can be sure there will be no issues with electronic communications. This is very common in the industry as it eliminates complexity in the system. Below, we have included specific reasons why we chose each of the electrical components.

3.3.1 Motors
We choose to use Teknic ClearPath Servo Motors in the design for their high-torque and high RPM while also providing extremely accurate positioning data which will allow us to accurately move the mechanical horse to give the rider the closest experience to riding a horse as possible. These motors are also very quiet compared to similar motors due to their construction which provides a better user experience.

3.3.2 Motor Controller
As sourcing all motor parts from one manufacturer simplifies the programming and controlling process, we decided to procure the ClearCore Motor Controller from Teknic as well. This motor controller also has Wi-Fi and Bluetooth connection capabilities, so it eliminated the need for the raspberry pi in our original design.

3.3.3 Power Supply
The Teknic IPC-5 was chosen as it was designed for specifically ClearPath motors and delivers up to 900W of power which will power all 3 of our motors with enough overhead. The IPC-5 also offers over current protection and can safely store regenerated power from the motors.

3.3.4 Controller Electronics
We chose to go with the WEMOS D32 Pro as it contains an ESP-32 Microcontroller which houses a Wi-Fi alliance certified 802.11n up to 150 Mbps modem that has been proven reliable for IOT (Internet of Things) devices. The WEMOS D32 Pro also contains a dual-core 32-bit processor which will allow for asynchronous computing for handing REST API requests and handling controller state changes at the same time. The WEMOS D32 also has 16MB of embedded flash storage eliminating the need for external storage. The WEMOS D32 also only draws 90 mA with the Wi-Fi modem active, making it ideal for the controller which will be battery powered. We choose to go with a Lithium Ion 18650 battery for its size and easy charging and high capacity. All other controller electronics were chosen for their size and feature set, as they are generic.

3.4 Prototype Tests
The prototype we built was used to test if we could reach the maximum angles of rotation with the current configuration of the horse system. Upon observing how the prototype moved, the team determined the seat movement was satisfactory. Through the testing of the system we found that we were able to hit our
maximum pitch roll and yaw angles from the data that we collected shown in figure 8. This gives us confidence that our design will be able to reach our max orientation values and be able to truly mimic the motion as we intend.

4. Manufacturing Plan

Overall, to put together our final prototype we will need to manufacture and assemble many different components together. In this section we will go over the methods that we used to procure our parts that we will then assemble. From there we discuss the manufacturing plans that we have for each of our subassemblies. Finally, we go over the methods to assemble our system together and create our final verification prototype.

4.1 Procurement

There are many constraints that shape an engineer’s design, including space limitations, weight limits, and sponsor preferences. One of the most important constraints is a team’s budget. Originally our team and the other senior project team working on the mechanical horse had a budget of $1000 to share. Knowing what we wanted to accomplish required a larger budget together we applied to the Baker Koob Grant and receive $2865 in additional funding. However, even with the additional funding, price is still one of the biggest concerns in our design process. Many of our design sessions, such as going over how many motors and gearboxes to get and what materials to make our base plate out, decided how we could accomplish our goal. With much thought and consideration, we adjusted our design to best fit our sponsors once and still stay within our project’s budget.

Since we were a recipient of the Baker Koob Grant, our process of procuring our materials must follow a process of getting approval by both team members’ senior project coaches. Additionally, both the winter and fall senior project teams now have a shared budget sheet and must have a clear line of communication when it comes to buying materials for our project. We know that the electrical components are going to be the most expensive so most of the additional funding we got from the grant went to us so we could purchase necessary electronics and motors. We also are aware electronic components take a longer time to be shipped and require a longer lead time, so they became our priority when completing our first purchase orders.

4.2 Manufacturing

Part of ensuring safety for both the rider and the instructor is analyzing the possible hazards our design could pose. We filled out the Design Hazard Checklist in Appendix G as a way of exploring the possible dangers inherent in our design. There were only two hazard types that our project could have: pinch points and large moving masses. We will elaborate on where these occur and how we plan to solve them below. We will also elaborate on how we plan on manufacturing each subsystem and further talk about how these subsystems will be assembled.

4.2.1 Arm Motor Assembly

Overall to create motion we use the arm motor assembly. This assembly takes the relevant components listed about and combines them to create motion. It is important to note that this system only requires one piece to be made, the holding clips. These clips are going to be made out of the extra aluminum sheet metal that we ordered. In order to make this we are going to cut the profile using a band saw or water jet and then drill a hole into both ends of the piece. From there we are going to bend it to sit snugly on the motors to hold them in place.
4.2.3 Baseplate
We plan to manufacture the baseplate from three 36” x 60” x 1/8” sheets of steel or aluminum, depending on the whether or not the material is donated or bought. The bottom plate will be a 3”x5” rectangle cut in half and bolted to the top of the existing frame. The top plate will be manufactured in a similar fashion, however cutouts will be made into one half of the sheet metal to leave room for free movement of the arm and for the post located in the center of the system. We plan to have a piece of 9” long steel square tube at each corner of the plate to support the top plate in its elevated position. Then, the side plates will be bolted into said square posts. The fabric will most likely be attached with a strong adhesive such as gorilla glue.

4.2.4 Controller
The controller casing will be 3D printed, so that it can be the custom size we require. Any cutouts will simply be printed into the casing. Additionally, the plastic will add as another level of insulation between the electrical components and the users, further preventing any event where the user could electrocute themselves. The components will be connected by a 22 AWG wire, which will be installed and soldered on by the teammates. Because this system runs under 40V, the campus electrician will not inspect the controller, however, the team members will ensure it works with a multimeter continuity sensor.

4.2.5 Horse Electronics
These electronics will be stored in a plastic box, under the baseplate. It would be almost impossible for someone to accidentally access the system electronics. The motor controller runs on a 24V power supply, which allows the team members to create connections between the motor controller and the motors without campus electrician approval. However, the motors that meet specifications for our design run on 75V. All these connections will need to be inspected before we can add them into the horse. Additionally, because we can manufacture many of the required for the system, we will be crimping and assembling terminals for all wires that are either powering or controlling the motors.

4.3 Assembly
Finally, after manufacturing all our parts, we will finally be able to assemble all our pieces together to create our final verification prototype. An advantage that our system has is that it is able to assemble in small chunks that can be done independently and therefore at the same time as other assemblies. For each assembly we can create all the subsystems at the same time, however we cannot implement them until the bottom baseplate is placed on the horse. The reason for this is that many of the components rest on the top of this bottom plate and then get covered by the top plate of the baseplate assembly. Finally, after placing in the carts, T bar and yaw motor and the internal electronics we can place the top plate of the baseplate on the system. Finally, we can place the side plates of the base plate together and cover the whole internal system thus protecting them.

5. Design Verification Plan
Finally after manufacturing and assembling our full system we are going to be spending a significant amount of time testing and verifying our prototype. This section discusses how we plan on verifying each spec that we came up with and the relevant tests we plan to do.

5.1 Evaluating Specs
Finally, after manufacturing and assembling our device, we will finally be able to verify that our system works. For this section we go over every specification and determine how we will verify that we meet these criteria or if the specification changed how and why the specification has changed.
5.1.1 Withstand In order to test whether this system will withstand the rider's weight we will perform a function test. In this sense withstanding the riders weight means being able to fully move and function as a normal horse would while loaded at the maximum 160 lb weight limit. After verifying that the system can maintain the constant pattern under a load, we will start to add more and more weight until finally we read the 160-pound final design weight. As a final test we will have our team member Aleya who is near but under the 160-pound weight limit test the system to verify that it functions as expected.

5.1.2 Mech Horse Weight
In order to test this specification, we will take each of the individual components and measure their weight on a scale and add them to get the overall weight. For this we will already have the weight of the frame and measure the weight of our system in order to make sure that it stays beneath the overall weight that we have specified.

5.1.3 Movement Matches Walk Data
In order to test to see if we match this criterion, we will take our system and place a phone with the MATLAB mobile app on it and track the output data from the horse. Through this we will compare it to the live data we captured and find a percent different between the two.

5.1.4 Movement Matches Trot Data
After considerations with our sponsor during our extra testing session we determined that this goal was not needed anymore and thus not a specification that we need to hit.

5.1.5 Accessible Parts
One of our priorities was for the electrical components to be easily replaceable. Because of the system we have chosen, all electronic connections are done by removable cables. This means any hardware component can be unplugged and easily replaced should it break without the need to reconstruct the whole system.

5.1.6 Overall Cost
The specification for this has changed rather significantly. Overall, we started this project with a budget of $500 but received the Baker-Koob award which increased our funding to $2000. Therefore, our specification limit has increased to $2000. Overall to evaluate if we meet this goal, we will keep a budget sheet with costs and track them through the project making sure that we don’t go over.

5.1.7 Withstand Dust
To test this specification, we shall perform a series of tests. The first of these tests is after building the water and dust protection we shall subject these to a forced dust case. For this test we will sprinkle dust onto the system and see if any gathers underneath the protection. The second test in this series is to leave the device in a dusty area for an extended period and check to see if there is any dust that has gathered in the protected region of the test.

5.1.8 Withstand Water
For testing this system, we shall perform a very similar series of tests to the withstand dust specification. For the first test we will subject the system to a small spray of water similar to that of a spray bottle. We will see if there is water that is collected at the bottom of the device. For the second test we will perform another forced water test but leave the water on the system for a significant period and then come back and see if any of the water has leaked into the protected zone.
5.1.9 Minimal Buttons/Switches
The users will have one button, one switch, and one potentiometer on the controller. The switch will be used to turn the controller on and off. The button will establish a connection with the horse and the potentiometer will allow the user to set a particular speed and movement pattern on the horse. This meets the three-button requirement.

5.1.10 Cost Aesthetic Supplies
For this specification we shall confirm that we will have the budget to add these design components into the system. This specification is the least important to accomplish because there are other components more important to invest our budget than our system’s aesthetics.

5.1.11 Functions on Different Surfaces
The system will be primarily used on one surface. So, we will first optimize our system for a flat surface. However, the servo motors we will use have velocity control. This means that they will automatically increase the amount of torque they provide should they be moving slower than they are set to. As a result, we believe this system should work on all surfaces with many different riders, no matter what.

5.1.12 Mech Horse Size
Overall, in order to test this specification, we shall test and ensure that every piece of our equipment can fit within the confines of the frame through placing it all on the device and ensuring that none of it interferes with each other.

5.2 Next Steps
Overall our next steps, after receiving approval is to start to order parts and have them shipped so we can start to manufacture and assemble our system. We plan on using the beginning of next quarter to develop the system and finish putting all of it together by the end of week 5. From there we will be performing tests on the system ensuring that it functions to our specifications. During this time, we will also have code being developed that will use and run the motors. This is important as we plan on hitting the ground running after week 5 in testing hopefully finishing up testing before week 9 of next quarter. Finally, we will be preparing for our Senior Design Expo and Final Design Review at the end of next quarter. A full list of the steps to come can be found in the Gantt chart in Appendix D and a summary of the dates when our next steps need to be completed can be found in table 2.

<table>
<thead>
<tr>
<th>Milestone</th>
<th>Description</th>
<th>Due Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manufacturing &amp; Testing</td>
<td>Status update of Manufacturing, updated test plan, and updated schedule of</td>
<td>3/10/21</td>
</tr>
<tr>
<td>Review</td>
<td>project completion</td>
<td></td>
</tr>
<tr>
<td>Prototype Sign Off</td>
<td>Verification of functionality of Final Prototype</td>
<td>4/26/21</td>
</tr>
<tr>
<td>Senior Expo</td>
<td>Display of structural prototype and Expo poster</td>
<td>5/27/21</td>
</tr>
<tr>
<td>Final Design Review</td>
<td>Final review and handoff of prototype and design report</td>
<td>6/3/21</td>
</tr>
</tbody>
</table>

Table 2: Summary of key milestones and descriptions with associated competition dates

6. Conclusion
Overall, we described our final design, justify our design decisions, and present our plan to manufacture and assemble the final product. Moving beyond this document we will be working on implementing these
designs into our final product. By writing this document we hope our sponsors attest that the content here is accurate, confirming our understanding of the challenge given to us and the efficacy of our final design
## Therapeutic Mechanical Horse

### Indented Bill of Material (iBOM)

<table>
<thead>
<tr>
<th>Assy Level</th>
<th>Part Number</th>
<th>Descriptive Part Name</th>
<th>Qty</th>
<th>Part Cost</th>
<th>Source</th>
<th>URL</th>
<th>More Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lvl0</td>
<td>Lvl1</td>
<td>Lvl2</td>
<td>Lvl3</td>
<td>Lvl4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>100000</td>
<td>Final Assy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>110000</td>
<td>Mechatronics Assembly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>120000</td>
<td>Controller</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>121000</td>
<td>Wemos (Microcontroller)</td>
<td>1</td>
<td>$18.99</td>
<td>Amazon</td>
<td><a href="https://ww">https://ww</a></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>122000</td>
<td>Power Switch</td>
<td>2</td>
<td>$0.99</td>
<td>Jameco</td>
<td><a href="https://ww">https://ww</a></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>123000</td>
<td>Button</td>
<td>2</td>
<td>$1.25</td>
<td>Jameco</td>
<td><a href="https://ww">https://ww</a></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>124000</td>
<td>Battery Casing</td>
<td>1</td>
<td>$7.99</td>
<td>Amazon</td>
<td><a href="https://ww">https://ww</a></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>125000</td>
<td>18650 Battery</td>
<td>1</td>
<td>$17.99</td>
<td>Amazon</td>
<td><a href="https://ww">https://ww</a></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>126000</td>
<td>Potentiometer</td>
<td>2</td>
<td>$1.69</td>
<td>Jameco</td>
<td><a href="https://ww">https://ww</a></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>127000</td>
<td>22 awg Black Wire</td>
<td>1</td>
<td>$10.95</td>
<td>Jameco</td>
<td><a href="https://ww">https://ww</a> 100ft</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>128000</td>
<td>22 awg Red Wire</td>
<td>1</td>
<td>$10.95</td>
<td>Jameco</td>
<td><a href="https://ww">https://ww</a> 100ft</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>130000</td>
<td>Horse Control</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>131000</td>
<td>IPC - 5 (Power Supply)</td>
<td>1</td>
<td>$248.00</td>
<td>Teknic</td>
<td><a href="https://tek">https://tek</a> 75 VDC Power Supply</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>132000</td>
<td>AC Power Cable</td>
<td>1</td>
<td>$14.00</td>
<td>Teknic</td>
<td><a href="https://tek">https://tek</a> Wall to 8-pin Mutex</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>133000</td>
<td>Motor Controller</td>
<td>1</td>
<td>$99.00</td>
<td>Teknic</td>
<td><a href="https://tek">https://tek</a> 24 VDC req to power</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>134000</td>
<td>Wiring</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>134100</td>
<td>16awg black wire (25ft)</td>
<td>1</td>
<td>$6.95</td>
<td>Jameco</td>
<td><a href="https://ww">https://ww</a> For 75V power cables</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>134200</td>
<td>16awg red wire (25ft)</td>
<td>1</td>
<td>$6.95</td>
<td>Jameco</td>
<td><a href="https://ww">https://ww</a> For 75V power cables</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>135000</td>
<td>Connectors</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Assy Level</td>
<td>Part Number</td>
<td>Descriptive Part Name</td>
<td>Qty</td>
<td>Part Cost</td>
<td>Source</td>
<td>More Info</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
<td>-----------------------------------------</td>
<td>-----</td>
<td>-----------</td>
<td>---------</td>
<td>--------------------------------</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>135100</td>
<td>Molex Min-Jr 8-pin terminal</td>
<td>70</td>
<td>$0.07</td>
<td>Digikey</td>
<td>Controller to motor terminals</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>135200</td>
<td>Molex Mini-Jr connector 8-pin</td>
<td>10</td>
<td>$0.61</td>
<td>Digikey</td>
<td>Controller to motor housings</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>135300</td>
<td>Molex Mini-Jr terminal (male)</td>
<td>20</td>
<td>$0.32</td>
<td>Digikey</td>
<td>Male power terminals</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>135400</td>
<td>Molex Mini-Jr terminal (female)</td>
<td>20</td>
<td>$0.16</td>
<td>Digikey</td>
<td>Female power terminals</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>135500</td>
<td>Molex Mini-Jr connector</td>
<td>5</td>
<td>$0.62</td>
<td>Digikey</td>
<td>Power housings</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>135600</td>
<td>Molex Sabre connector</td>
<td>5</td>
<td>$0.82</td>
<td>Digikey</td>
<td>Power supply housing</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>135700</td>
<td>Molex Sabre terminal</td>
<td>10</td>
<td>$0.11</td>
<td>Digikey</td>
<td>Power supply terminal</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>135800</td>
<td>terminal block plugs</td>
<td>3</td>
<td>$1.40</td>
<td>Digikey</td>
<td>Housing for controller power</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>136000</td>
<td>24V Switching power supply</td>
<td>1</td>
<td>$17.99</td>
<td>Amazon</td>
<td>Power for controller</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>137000</td>
<td>24V Power Supply Power Cable</td>
<td>1</td>
<td>$7.89</td>
<td>Amazon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>140000</td>
<td>Arm Motor Assembly</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>141000</td>
<td>Arm Motor</td>
<td>2</td>
<td>$402.00</td>
<td>Teknic</td>
<td>Servo Motor</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>142000</td>
<td>Gearbox</td>
<td>2</td>
<td>$88.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>143000</td>
<td>Pillow Block Bearing</td>
<td>2</td>
<td>$25.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>144000</td>
<td>Holding Clips</td>
<td>4</td>
<td>$-</td>
<td></td>
<td>Leftover sheet metal</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>150000</td>
<td>Baseplate Assy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>151000</td>
<td>Bottom Plate</td>
<td>2</td>
<td>$99.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>152000</td>
<td>Top Plate</td>
<td>1</td>
<td>$99.00</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>153000</td>
<td>Side Panels</td>
<td>1</td>
<td>$-</td>
<td></td>
<td>Scrap Metal from</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>154000</td>
<td>Waterproof Fabric</td>
<td>1</td>
<td>$-</td>
<td></td>
<td>From other team</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>155000</td>
<td>Steel Post 72&quot;</td>
<td>1</td>
<td>$25.71</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>160000</td>
<td>Safety Assy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>161000</td>
<td>Surge Protector</td>
<td>1</td>
<td>$16.70</td>
<td>Amazon</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>162000</td>
<td>Zip Ties</td>
<td>1</td>
<td>$6.99</td>
<td>Amazon</td>
<td>For Securing Wires</td>
<td></td>
</tr>
</tbody>
</table>
## Appendix A – Bill of Materials

<table>
<thead>
<tr>
<th>Assy Level</th>
<th>Part Number</th>
<th>Descriptive Part Name</th>
<th>Qty</th>
<th>Part Cost</th>
<th>Source</th>
<th>URL</th>
<th>More Info</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>163000</td>
<td>Wire Loom</td>
<td>1</td>
<td>$14.99</td>
<td>Amazon</td>
<td><a href="https://www.amazon.com">https://www.amazon.com</a></td>
<td>For bundling wires</td>
</tr>
<tr>
<td>2</td>
<td>164000</td>
<td>E-Stop</td>
<td>1</td>
<td>$12.99</td>
<td>Amazon</td>
<td><a href="https://www.amazon.com">https://www.amazon.com</a></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>165000</td>
<td>On/Off Switch</td>
<td>1</td>
<td></td>
<td>Home Depot</td>
<td>#3-1/2-in</td>
<td></td>
</tr>
<tr>
<td><strong>Total Parts</strong></td>
<td></td>
<td></td>
<td><strong>182</strong></td>
<td><strong>$1,913.04</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ITEM NO.</td>
<td>PART NUMBER</td>
<td>DESCRIPTION</td>
<td>QTY.</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>---------------------</td>
<td>------</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Baseplate Assembly</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>CPM-<strong>-</strong>- - 3441-LS</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>Pillow Block Bearing</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>Concentric Gearbox</td>
<td></td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Holding Clip</td>
<td></td>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>Blackbox</td>
<td></td>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B – Drawing and Spec Packages

Part Number 120000
## Technical specs

<table>
<thead>
<tr>
<th>Specification</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operating Voltage</td>
<td>3.3V</td>
</tr>
<tr>
<td>Supported Battery</td>
<td>Lipo 3.7V</td>
</tr>
<tr>
<td>Battery Connector</td>
<td>PH-2 2.0mm</td>
</tr>
<tr>
<td>Digital I/O Pins</td>
<td>22</td>
</tr>
<tr>
<td>Analog Input Pins</td>
<td>6 (VP, VN, 32, 33, 34, 35)</td>
</tr>
<tr>
<td>Analog Output Pins</td>
<td>2 (25, 26)</td>
</tr>
<tr>
<td>LED_BUILTIN</td>
<td>GPIO5</td>
</tr>
<tr>
<td>Clock Speed (Max)</td>
<td>240MHz</td>
</tr>
<tr>
<td>Flash</td>
<td>16M/4M Bytes</td>
</tr>
<tr>
<td>PSRAM</td>
<td>4M Bytes</td>
</tr>
<tr>
<td>Size</td>
<td>65*25.4mm</td>
</tr>
<tr>
<td>Weight</td>
<td>7.5g</td>
</tr>
<tr>
<td>Specification</td>
<td>Value</td>
</tr>
<tr>
<td>----------------------------</td>
<td>--------------------------------------------</td>
</tr>
<tr>
<td>Pins</td>
<td>2</td>
</tr>
<tr>
<td>Contact Form</td>
<td>SPST</td>
</tr>
<tr>
<td>Switch Function</td>
<td>ON-OFF</td>
</tr>
<tr>
<td>Current Rating</td>
<td>10A @ 125VAC, 6A @ 250VAC</td>
</tr>
<tr>
<td>Actuator Type</td>
<td>Rocker</td>
</tr>
<tr>
<td>Body Color</td>
<td>Black</td>
</tr>
<tr>
<td>Color - Actuator/Cap</td>
<td>Black</td>
</tr>
<tr>
<td>UL Flammability Code</td>
<td>94V-2</td>
</tr>
<tr>
<td>Actuator Length</td>
<td>0.141 in.</td>
</tr>
<tr>
<td>Lead Spacing A (inch)</td>
<td>0.279</td>
</tr>
<tr>
<td>Length</td>
<td>0.826 in.</td>
</tr>
<tr>
<td>Width</td>
<td>0.59 in.</td>
</tr>
<tr>
<td>Height</td>
<td>0.964 in.</td>
</tr>
</tbody>
</table>

![Image of a black switch](image)
<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Contact Form</td>
<td>SPST-NO</td>
</tr>
<tr>
<td>Switch Function</td>
<td>OFF-(ON)</td>
</tr>
<tr>
<td>Contact Rating</td>
<td>1 A @ 125 VAC</td>
</tr>
<tr>
<td>Body Color</td>
<td>Black</td>
</tr>
<tr>
<td>Color - Actuator/Cap</td>
<td>Red</td>
</tr>
<tr>
<td>Termination Style</td>
<td>Solder Lugs</td>
</tr>
<tr>
<td>Pins</td>
<td>2</td>
</tr>
<tr>
<td>Length</td>
<td>1.1 in.</td>
</tr>
<tr>
<td>Actuator Type</td>
<td>Round Button</td>
</tr>
<tr>
<td>Illumination Type</td>
<td>N/A</td>
</tr>
<tr>
<td>Mounting Style</td>
<td>Panel, Rear</td>
</tr>
</tbody>
</table>
Part Number: 124000

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Package Dimensions</strong></td>
<td>4.88 x 3.39 x 0.87 inches</td>
</tr>
<tr>
<td><strong>Item Weight</strong></td>
<td>1.44 ounces</td>
</tr>
<tr>
<td><strong>Manufacturer</strong></td>
<td>MELIFE</td>
</tr>
<tr>
<td><strong>ASIN</strong></td>
<td>B08D3MBVY8</td>
</tr>
</tbody>
</table>

18650 Battery Shield V3 ESP32 wildly used for Raspberry Pi and Arduino

**Features:**
Battery protection (Over charge or Over discharge)
Micro USB port Input
Type-A USB Output
0.5A current charging
1 switch control USB output
5~8V Input Voltage
3V 1A Output *3
5V 2A Output *3

**Special Attention:**
The installation of the battery must be determined positive and negative. You should follow the direction of "+" "-" on the PCB. If put wrong direction, charging chip will be destroyed. 18650-battery does not include.

**Package Including:**
1 * Battery Shield Module (18650 battery not include)
1 * Micro USB Cable
**Part Number: 126000**

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resistance (Ohm)</td>
<td>10000</td>
</tr>
<tr>
<td>Power Rating</td>
<td>0.125 W</td>
</tr>
<tr>
<td>Tolerance</td>
<td>20%</td>
</tr>
<tr>
<td>Shaft Style</td>
<td>Round</td>
</tr>
<tr>
<td>Shaft Length</td>
<td>0.590 in.</td>
</tr>
<tr>
<td>Termination Style</td>
<td>Solder, Panel Mount</td>
</tr>
<tr>
<td>Taper</td>
<td>Linear</td>
</tr>
<tr>
<td>Body Diameter</td>
<td>16 mm</td>
</tr>
<tr>
<td>Manufacturer Series</td>
<td>RV16AF</td>
</tr>
<tr>
<td>Voltage Rating</td>
<td>200 V</td>
</tr>
<tr>
<td>Ganging Number</td>
<td>1</td>
</tr>
<tr>
<td>Resistive Element Material</td>
<td>Carbon</td>
</tr>
<tr>
<td>Number of Turns</td>
<td>1</td>
</tr>
<tr>
<td>Resistance</td>
<td>10 KOhm</td>
</tr>
<tr>
<td>Mounting</td>
<td>Panel Mount</td>
</tr>
<tr>
<td>Shaft Diameter</td>
<td>0.25 in, 6 mm</td>
</tr>
</tbody>
</table>
#### Part Number: 127000

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Type</td>
<td>Hook-up Wire</td>
</tr>
<tr>
<td>Wire Gauge (AWG)</td>
<td>22</td>
</tr>
<tr>
<td>Number of Conductors</td>
<td>1</td>
</tr>
<tr>
<td>Conductor Strand</td>
<td>7 x 30</td>
</tr>
<tr>
<td>Color</td>
<td>Black</td>
</tr>
<tr>
<td>Length</td>
<td>25 ft.</td>
</tr>
<tr>
<td>Insulation Type</td>
<td>PVC</td>
</tr>
<tr>
<td>Voltage Rating</td>
<td>300 V</td>
</tr>
<tr>
<td>Package</td>
<td>Reel</td>
</tr>
<tr>
<td>Additional Feature</td>
<td>Made in USA</td>
</tr>
</tbody>
</table>

#### Part Number: 128000

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Type</td>
<td>Hook-up Wire</td>
</tr>
<tr>
<td>Wire Gauge (AWG)</td>
<td>22</td>
</tr>
<tr>
<td>Number of Conductors</td>
<td>1</td>
</tr>
<tr>
<td>Conductor Strand</td>
<td>7 x 30</td>
</tr>
<tr>
<td>Color</td>
<td>Black</td>
</tr>
<tr>
<td>Length</td>
<td>25 ft.</td>
</tr>
<tr>
<td>Insulation Type</td>
<td>PVC</td>
</tr>
<tr>
<td>Voltage Rating</td>
<td>300 V</td>
</tr>
<tr>
<td>Package</td>
<td>Reel</td>
</tr>
<tr>
<td>Additional Feature</td>
<td>Made in USA</td>
</tr>
</tbody>
</table>
Appendix B – Drawing and Spec Packages

Part Number 130000

Part Number 131000 and 132000

Technical Manual begins on next page
### Part Number 133000

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Dimensions</strong></td>
<td>5.0&quot; x 3.5&quot; x 1.0&quot; (127mm x 88.9mm x 25.4mm)</td>
</tr>
<tr>
<td><strong>Weight (with cover)</strong></td>
<td>0.40 lbs (181 g)</td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>3mm thick Polycarbonate cover, aluminum mount frame</td>
</tr>
<tr>
<td><strong>Voltage Input</strong></td>
<td>20-28 VDC</td>
</tr>
<tr>
<td><strong>Output Current Capability</strong></td>
<td>I/O 0,1,2,3: 375 mA RMS, (750mA peak) I/O 4,5: 750 mA RMS, (1,000mA peak)</td>
</tr>
<tr>
<td><strong>Logic Compatibility</strong></td>
<td>All I/O is compatible with 3.3 to 24 VDC logic. (ClearCore outputs are internally pulled up to 24V (Vsupply), so 3.3 to 5V logic may require an external clamping diode - see device mfr's datasheet.)</td>
</tr>
<tr>
<td><strong>Indicator LEDs for each input</strong></td>
<td>Yes</td>
</tr>
<tr>
<td><strong>IP rating</strong></td>
<td>IP20</td>
</tr>
<tr>
<td><strong>Operating Temperature/Humidity</strong></td>
<td>-20C to 50C, 0-90% non-condensing</td>
</tr>
<tr>
<td><strong>Storage Temperature</strong></td>
<td>-40C to 85C</td>
</tr>
<tr>
<td><strong>Power Consumption</strong></td>
<td>300mA@24V or 500mA@12V Adding an XBee will add as much as an additional 100mA@24V or 200mA@12V</td>
</tr>
<tr>
<td><strong>Protection features</strong></td>
<td>Overcurrent protection on all outputs Inductive clamping on all outputs Board master overvoltage and overcurrent protection ESD protection features on all I/O circuits 13 built-in I/O points, software configurable as any combination of up to 13 digital inputs, 4 analog inputs, 6 digital outputs, 2 HBridge/speaker outputs and 1 analog output (4-20mA or 0-20mA).</td>
</tr>
</tbody>
</table>
### Appendix B – Drawing and Spec Packages

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total I/O</strong></td>
<td>Another 64 digital I/O can be added by using optional 8-point I/O expansion modules (p/n CCIO-8). All configuration of I/O hardware is controlled by software, i.e., no jumpers, DIP switches, trim-pots, etc. need to be manually set.</td>
</tr>
<tr>
<td><strong>Serial communication</strong></td>
<td>2 Multi-functional, individually configurable serial ports that can be used as a UART, SPI, or RS-232 at up to 115.2kBaud. Rates up to 2MBaud are achievable depending on cable length, slave transceiver circuit and grounding. 5V power pins are available on each port.</td>
</tr>
<tr>
<td><strong>Ethernet</strong></td>
<td>10Base-T/100Base-TX Ethernet</td>
</tr>
<tr>
<td><strong>USB</strong></td>
<td>USB 2.0</td>
</tr>
<tr>
<td><strong>Wireless connectivity</strong></td>
<td>Accepts Xbee modules for wireless connectivity (Wi-Fi, Bluetooth, Mesh, etc.)</td>
</tr>
<tr>
<td><strong>Programming Language</strong></td>
<td>C++</td>
</tr>
<tr>
<td><strong>Memory Capability</strong></td>
<td>512 KB Flash 192 KB RAM Storage expansion via onboard SD card drive</td>
</tr>
<tr>
<td><strong>Supported development environment</strong></td>
<td>Windows 10, Windows 7* (see note)</td>
</tr>
<tr>
<td><strong>Processor type and speed</strong></td>
<td>32 bit floating point ARM M4F processor 120 MHz (p/n SAME53N19A)</td>
</tr>
<tr>
<td><strong>Max Step Rate</strong></td>
<td>500kHz</td>
</tr>
</tbody>
</table>

**Technical Manual begins on next page**
### Part Number 134100

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Type</td>
<td>Hook-up Wire</td>
</tr>
<tr>
<td>Wire Gauge (AWG)</td>
<td>16</td>
</tr>
<tr>
<td>Number of Conductors</td>
<td>1</td>
</tr>
<tr>
<td>Conductor Strand</td>
<td>7 x 30</td>
</tr>
<tr>
<td>Color</td>
<td>Black</td>
</tr>
<tr>
<td>Length</td>
<td>100 ft.</td>
</tr>
<tr>
<td>Insulation Type</td>
<td>PVC</td>
</tr>
<tr>
<td>Voltage Rating</td>
<td>300 V</td>
</tr>
<tr>
<td>Package</td>
<td>Reel</td>
</tr>
<tr>
<td>Additional Feature</td>
<td>Made in USA</td>
</tr>
</tbody>
</table>

### Part Number 134200

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Product Type</td>
<td>Hook-up Wire</td>
</tr>
<tr>
<td>Wire Gauge (AWG)</td>
<td>16</td>
</tr>
<tr>
<td>Number of Conductors</td>
<td>1</td>
</tr>
<tr>
<td>Conductor Strand</td>
<td>7 x 30</td>
</tr>
<tr>
<td>Color</td>
<td>Red</td>
</tr>
<tr>
<td>Length</td>
<td>100 ft.</td>
</tr>
<tr>
<td>Insulation Type</td>
<td>PVC</td>
</tr>
<tr>
<td>Voltage Rating</td>
<td>300 V</td>
</tr>
<tr>
<td>Package</td>
<td>Reel</td>
</tr>
<tr>
<td>Additional Feature</td>
<td>Made in USA</td>
</tr>
<tr>
<td>MATERIAL</td>
<td>F</td>
</tr>
<tr>
<td>----------</td>
<td>---</td>
</tr>
<tr>
<td>REFLOWED MATTE TIN 0.00090/0.00035 MIN. (PREPLATE) (FINISH IS BRIGHT IN APPEARANCE, THICKNESS AS APPLIED PRIOR TO REFLOW)</td>
<td></td>
</tr>
<tr>
<td>PHOSPHOR BRONZE</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.177</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>REFLOWED MATTE TIN 0.00090/0.00035 MIN OVER COPPER 0.00050/0.00020 MIN. (PREPLATE) (FINISH IS BRIGHT IN APPEARANCE, THICKNESS AS APPLIED PRIOR TO REFLOW)</td>
<td></td>
</tr>
<tr>
<td>BRASS</td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.177</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>0.24</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.35</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>PLATING</td>
<td></td>
</tr>
<tr>
<td>MATERIAL</td>
<td></td>
</tr>
<tr>
<td>F</td>
<td>E</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>N/A</td>
<td>39-00-0084</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>39-00-0067</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
### Appendix B – Drawing and Spec Packages

<table>
<thead>
<tr>
<th>MATERIAL</th>
<th>F</th>
<th>E</th>
<th>D</th>
<th>C</th>
<th>B</th>
<th>A</th>
<th>INS. RANGE</th>
<th>EDP NO.</th>
<th>ENG. NO.</th>
<th>FORM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>PHOS BRONZE</strong></td>
<td>(0.9)</td>
<td>(4.5)</td>
<td>(3.6)</td>
<td>(0.6)</td>
<td>(2.7)</td>
<td>(2.3)</td>
<td>Ø 13.1 - .122 MAX.</td>
<td>N/A</td>
<td>39-00-0080</td>
<td>5556 PBT3L</td>
</tr>
<tr>
<td></td>
<td>0.035</td>
<td>0.177</td>
<td>0.142</td>
<td>0.024</td>
<td>0.106</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.6)</td>
<td>(2.3)</td>
<td>(2.3)</td>
<td>(0.4)</td>
<td>(1.65)</td>
<td>(1.8)</td>
<td>Ø 0.9 - 1.8</td>
<td>20+22</td>
<td>-0.079</td>
<td>PBT3</td>
</tr>
<tr>
<td></td>
<td>0.024</td>
<td>0.091</td>
<td>0.091</td>
<td>0.016</td>
<td>0.065</td>
<td>0.071</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.9)</td>
<td>(4.5)</td>
<td>(3.6)</td>
<td>(0.5)</td>
<td>(2.3)</td>
<td>(1.9)</td>
<td>Ø (1.3 - 3.1) (0.5 - 1.22</td>
<td>18-24</td>
<td>-0.059</td>
<td>PBT</td>
</tr>
<tr>
<td></td>
<td>0.035</td>
<td>0.177</td>
<td>0.142</td>
<td>0.020</td>
<td>0.091</td>
<td>0.075</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>BRASS</strong></td>
<td>(0.9)</td>
<td>(4.5)</td>
<td>(3.6)</td>
<td>(0.6)</td>
<td>(2.7)</td>
<td>(2.3)</td>
<td>Ø 13.1 - .122 MAX.</td>
<td>N/A</td>
<td>39-00-0038</td>
<td>5556 T</td>
</tr>
<tr>
<td></td>
<td>0.035</td>
<td>0.177</td>
<td>0.142</td>
<td>0.024</td>
<td>0.106</td>
<td>0.091</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.6)</td>
<td>(2.3)</td>
<td>(2.3)</td>
<td>(0.4)</td>
<td>(1.65)</td>
<td>(1.8)</td>
<td>Ø 0.9 - 1.8</td>
<td>20+22</td>
<td>-0.047</td>
<td>T2L</td>
</tr>
<tr>
<td></td>
<td>0.024</td>
<td>0.091</td>
<td>0.091</td>
<td>0.016</td>
<td>0.065</td>
<td>0.071</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(0.9)</td>
<td>(4.5)</td>
<td>(3.6)</td>
<td>(0.5)</td>
<td>(2.3)</td>
<td>(1.9)</td>
<td>Ø (1.3 - 3.1) (0.5 - 1.22</td>
<td>18-24</td>
<td>-0.039</td>
<td>TL</td>
</tr>
<tr>
<td></td>
<td>0.035</td>
<td>0.177</td>
<td>0.142</td>
<td>0.020</td>
<td>0.091</td>
<td>0.075</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Notes:**
- **REFLOWED MATTE TIN**: 0.000090(0.00035) MIN. (PREPLATE), FINISH IS BRIGHT IN APPEARANCE, THICKNESS AS APPLIED PRIOR TO REFLOW.
- **REFLOWED MATTE TIN OVER COPPER**: 0.000050(0.00020) MIN. (PREPLATE), FINISH IS BRIGHT IN APPEARANCE, THICKNESS AS APPLIED PRIOR TO REFLOW.
Part Number: 135200 (8-pin) and 135500 (4-pin)
NOTES:
1. MATERIAL: SEE CHART.
2. THIS CONNECTOR HOUSING FOR USE WITH 43375-**** TERMINALS.
3. THIS CONNECTOR HOUSING WILL MATE TO A-43160-****, 43180-****, AND 43680-**** HOUSINGS.
4. PRODUCT SPECIFICATION: PSX-44441-9999.
5. PACKAGING: PK-44441-001.
6. FOR V-0 AND GLOW-WIRE MATERIAL: PARTS CONFORM TO CLASS 'B'
   REQUIREMENTS OF COSMETIC SPECIFICATION PS-45499-002 BUT
   MEETS CLASS 'C' REQUIREMENTS FOR BUBBLES, SINK AND DISCOLORATION.
7. TEXT ON PART IS FOR REFERENCE ONLY. TEXT AND TEXT LOCATION MAY VARY
   DEPENDING ON PART NUMBER AND/OR TOOL.
8. LAST CIRCUIT I.D. NOT PRESENT ON 2 CIRCUIT PARTS.
9. DIMENSION C IS .600/(15.24) FOR 2 AND 8 CIRCUIT PARTS
   DIMENSION C IS .632/(16.05) FOR ALL OTHER CIRCUIT SIZES.
10. THIS DRAWING REPLACES DRAWINGS SD-44441-*.
## Appendix B – Drawing and Spec Packages

### Polarization Layout

<table>
<thead>
<tr>
<th>CIRCUITS</th>
<th>ITEM NUMBER</th>
<th>MATERIAL</th>
<th>COLOR</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>44441-1002</td>
<td>PA66 NYLON, UL94V-2</td>
<td>NATURAL (TRANSLUCENT)</td>
</tr>
<tr>
<td>3</td>
<td>44441-1003</td>
<td>PA66 NYLON, UL94V-2</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>44441-1004</td>
<td>PA66 NYLON, UL94V-0</td>
<td>BLACK</td>
</tr>
<tr>
<td>5</td>
<td>44441-1005</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>44441-1006</td>
<td>PA66 NYLON, UL94V-2</td>
<td>BLACK</td>
</tr>
<tr>
<td>7</td>
<td>44441-1007</td>
<td>PA66 NYLON, UL94V-2</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>44441-1008</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>44441-2002</td>
<td>PA66 NYLON, UL94V-2</td>
<td>BLACK</td>
</tr>
<tr>
<td>3</td>
<td>44441-2003</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>44441-2004</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>44441-2005</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>44441-2006</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>44441-2007</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>44441-2008</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>44441-3002</td>
<td>PA66 NYLON, UL94V-2 IEC 60335-1 GLOW WIRE CAPABLE</td>
<td>BLACK</td>
</tr>
<tr>
<td>3</td>
<td>44441-3003</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>44441-3004</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>44441-3005</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>44441-3006</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>44441-3007</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>44441-3008</td>
<td>PA66 NYLON, UL94V-0</td>
<td></td>
</tr>
</tbody>
</table>
Appendix B – Drawing and Spec Packages

Part Number: 135700

NOTES:
1. MATERIAL: 0.121(0.30) THICK C26000 BRASS.
2. FINISH: HOT TIN DIP.
3. PRODUCT SPECIFICATION: PS-44441-9909.
4. PACKAGING SPECIFICATION: 43375000-PK
5. FOR USE IN 44441 SERIES HOUSINGS.
6. SINGLE CRIMP TO #16 & #14 AWG WIRE WITH 0.31"(0.78) MAXIMUM INSULATION THICKNESS.
7. NO PORTION OF ONE LOCK TANG BEAM SHALL TOUCH ANY PORTION OF THE OTHER LOCK TANG BEAM.
8. PARTS CONFORM TO CLASS 'B' REQUIREMENTS OF PS-45489-002.

Molex

Terminal Receptacle
125(3.18) X 220(5.51) Flat Blade System - TPA

Product Customer Drawing

SD-43375-0001

Document Status: P1
Release Date: 2018/06/18 16:00:37

C SIZE 43375

Document SD-43375-0001

Page 1 of 1
Appendix B – Drawing and Spec Packages

Part Number 135100 - 135800

Technical Manual begins on next page
Part Number: 136000

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Product Dimensions</strong></td>
<td>7.87 x 3.94 x 1.57 inches</td>
</tr>
<tr>
<td><strong>Item Weight</strong></td>
<td>1.1 pounds</td>
</tr>
<tr>
<td><strong>ASIN</strong></td>
<td>B078RTV7HV</td>
</tr>
<tr>
<td><strong>Item model number</strong></td>
<td>AL24V5AT</td>
</tr>
<tr>
<td><strong>Is Discontinued By Manufacturer</strong></td>
<td>No</td>
</tr>
<tr>
<td><strong>Manufacturer</strong></td>
<td>ALITOVE</td>
</tr>
<tr>
<td><strong>Product Dimensions</strong></td>
<td>7.87 x 3.94 x 1.57 inches</td>
</tr>
<tr>
<td><strong>Item Weight</strong></td>
<td>1.1 pounds</td>
</tr>
<tr>
<td><strong>Input</strong></td>
<td>AC110V/220V 50/60Hz</td>
</tr>
<tr>
<td><strong>Output</strong></td>
<td>DC 24V : 5A max 120 W max</td>
</tr>
<tr>
<td><strong>Fix Screw Hole Diameter</strong></td>
<td>2.5mm (0.1inch)</td>
</tr>
<tr>
<td><strong>Working temperature</strong></td>
<td>10 to 50 degree Celsius</td>
</tr>
<tr>
<td><strong>Storage temperature</strong></td>
<td>20 to 60 degree Celsius</td>
</tr>
<tr>
<td><strong>Environmental humidity</strong></td>
<td>10-95%</td>
</tr>
<tr>
<td><strong>Material</strong></td>
<td>Metal, Electronic Parts</td>
</tr>
<tr>
<td><strong>Safety Compliance</strong></td>
<td>CCC / FCC / CE</td>
</tr>
</tbody>
</table>
Teknic ClearPath®
Fractional HP, DC Input
NEMA 34

- All dimensions are in inch [mm] unless otherwise noted.
- Sheet Scale 1:1.5
- Undimensioned features are per 3D file for model of interest for reference use.
- Unless otherwise specified untoleranced dimensions are nonbasic and default to:
  X.XXX: ± .010" [0.25mm]
  X.XXX: ± .005" [0.125mm]
  Angles: ± 1°

Part Number 141000

<table>
<thead>
<tr>
<th>Model Number</th>
<th>L</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPM-341</td>
<td>3.138 [79.70]</td>
</tr>
<tr>
<td>CPM-342</td>
<td>3.879 [98.52]</td>
</tr>
<tr>
<td>CPM-343</td>
<td>4.627 [117.52]</td>
</tr>
<tr>
<td>CPM-344</td>
<td>5.382 [136.70]</td>
</tr>
</tbody>
</table>

Teknic, Incorporated
Victor, NY 14564 USA

This document and related electronic files are the sole property of Teknic, Incorporated and contain proprietary information.

Third Angle Projection
Interpret Geometric Tolerances per ANSI/ASME Y14.5 (2009)

Additional documentation including product specifications, up-to-date compliance information, and manuals can be downloaded from Teknic's website.
Visit https://www.teknic.com to access that documentation.
Part Number: 142000

SPLF90-L1

SPLF90-L2
NOTES
UNLESS OTHERWISE SPECIFIED
1.  ALL DIMS IN INCHES
2.  TOLERANCES
   X.XX= ±0.01
   X.XXX= ±0.005
   ANGLES= ±2°
3.  BREAK SHARP EDGES .02 MAX
4.  TOLL RAIDUS .02 MAX
5.  ☑️ FAO

2 x Ø0.38 ± 0.10
R1.90
R2.00

Part Number 144000
Part Number 151000

<table>
<thead>
<tr>
<th>ITEM NO.</th>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>QTY.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bottom Plate</td>
<td>Bottom Plate</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Top Plate (Motor Side)</td>
<td>Top Plate (Motor Side)</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Top Plate (Electronic Side)</td>
<td>Top Plate (Electronic Side)</td>
<td>1</td>
</tr>
</tbody>
</table>

Lab Section: 07  Title: Baseplate Assembly  
Dwg. #: 150000  Date: 3/11/22  Scale: 1:10
<table>
<thead>
<tr>
<th>ITEM NO.</th>
<th>PART NUMBER</th>
<th>DESCRIPTION</th>
<th>QTY.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Bottom Plate</td>
<td>Bottom Plate</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Top Plate (Motor Side)</td>
<td>Top Plate (Motor Side)</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>Top Plate (Electronic Side)</td>
<td>Top Plate (Electronic Side)</td>
<td>1</td>
</tr>
</tbody>
</table>
## DVP&R - Design Verification Plan (& Report)

<table>
<thead>
<tr>
<th>Test #</th>
<th>Specification</th>
<th>Test Description</th>
<th>Measurements</th>
<th>Acceptance Criteria</th>
<th>Required Facilities/Equipment</th>
<th>Parts Needed</th>
<th>Responsibility</th>
<th>TIMING</th>
<th>Notes on Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 10lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/7/2022</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 20lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/9/2022</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 40lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/11/2022</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 80lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/13/2022</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 160lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/15/2022</td>
<td></td>
</tr>
</tbody>
</table>
## DVP&R - Design Verification Plan (& Report)

**Project:** F76 - Mechanical Horse  
**Sponsor:** Jack's Helping Hand  
**Edit Date:** 2/18/2022

<table>
<thead>
<tr>
<th>Test #</th>
<th>Specification</th>
<th>Test Description</th>
<th>Measurements</th>
<th>Acceptance Criteria</th>
<th>Required Facilities/Equipment</th>
<th>Parts Needed</th>
<th>Responsibility</th>
<th>TIMING</th>
<th>TEST RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Movement Matches Walk Data</td>
<td>We will ensure the mechanical horse has a similar walk pattern to that of a real horse.</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data must be within 20% of the data we collected from a real horse</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly</td>
<td>Luke</td>
<td>5/17/2022</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Withstands Dust</td>
<td>We will lightly sprinkle sand over the system and check to see if there is any that has filtered through our protection</td>
<td>Observation, is there too much dust</td>
<td>There must be little to no dust on the motors and electronics</td>
<td>a place to leave the system that will be secure, yet dusty. Possibly, our sponsor's barn.</td>
<td>Baseplate assembly and sand</td>
<td>Cade</td>
<td>4/6/2022</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>Withstands Dust</td>
<td>We will leave the system in an area that gathers dust and inspect how much dust got past our defenses.</td>
<td>Observation, is there too much dust</td>
<td>There must be little to no dust on the motors and electronics</td>
<td>a place to leave the system that will be secure, yet dusty. Possibly, our sponsor's barn.</td>
<td>Baseplate assembly and sand</td>
<td>Cade</td>
<td>4/6/2022</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>Withstands Water</td>
<td>We will lightly spray the system with water and see if any water reaches past the protected surface</td>
<td>Observation, did any water reach the protected surface</td>
<td>Protected surface is dry</td>
<td>somewhere outside where we can spill a bottle of water on the system.</td>
<td>Baseplate assembly and water</td>
<td>Cade</td>
<td>4/7/2022</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>Withstands Water</td>
<td>We will spray the system in water enough to let it pool a small amount and leave it over time to ensure that the system is able to withstand water over time</td>
<td>Observation, did any water reach the protected surface</td>
<td>Protected surface is dry</td>
<td>somewhere outside where we can spill a bottle of water on the system.</td>
<td>Baseplate assembly and water</td>
<td>Cade</td>
<td>4/7/2022</td>
<td></td>
</tr>
</tbody>
</table>
# DVP&R - Design Verification Plan (& Report)

## TEST PLAN

<table>
<thead>
<tr>
<th>Test #</th>
<th>Specification</th>
<th>Test Description</th>
<th>Measurements</th>
<th>Acceptance Criteria</th>
<th>Required Facilities/Equipment</th>
<th>Parts Needed</th>
<th>Responsibility</th>
<th>TIMING</th>
<th>TEST RESULTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Functions on Different Surfaces</td>
<td>Run the system on a slight incline</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data must be within 30% of a seat on flat ground</td>
<td>somewhere with an outlet and slightly inclined ground.</td>
<td>completed physical assembly</td>
<td>Zuzanna</td>
<td>5/24/2022</td>
<td>Numerical Results</td>
</tr>
</tbody>
</table>
## Appendix D – Team Gantt Chart

### F76 Therapeutic Mechatronics

#### Scope of Work
- Start: 08/26/23
- End: 10/26/23
- 100%

<table>
<thead>
<tr>
<th>Activity</th>
<th>Start</th>
<th>End</th>
<th>Duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>PDR</td>
<td>08/26/23</td>
<td>10/26/23</td>
<td>20%</td>
</tr>
<tr>
<td>IDR</td>
<td>11/21/23</td>
<td>01/08/24</td>
<td>100%</td>
</tr>
<tr>
<td>COR</td>
<td>01/14/23</td>
<td>02/24/23</td>
<td>100%</td>
</tr>
</tbody>
</table>

#### Manufacturing
- Start: 02/11/23
- End: 03/06/23

- Perform Risk Assessment: 02/11 - 02/17
- Manufacturing: 02/24 - 03/02
- Receive Parts: 03/03 - 03/09
- Perform Manufacturing: 03/05 - 04/24
- Manufacture Components: 03/05 - 04/14
- Manufacture Electronics: 03/10 - 04/24
- 3D Print Electronic Casing: 03/10 - 03/18
- Cut wires to size: 03/11 - 03/18
- Crimp terminals onto wire: 03/11 - 03/18
- Place completed terminals in casing: 03/17 - 03/17
- Get wires for 75V inspected: 04/06 - 04/06
- Test Ptf: 04/25 - 04/29
- Remanufacture if needed: 04/26 - 04/29
- Perform Experimental Testing Design: 05/03 - 05/08
- Manufacturing and Testing Review: 05/13 - 05/13
- Verification Prototype Sign-Off: 05/23 - 05/25

#### Assembly
- Start: 05/13/23
- End: 05/23/23

- Assembly Controller: 05/11 - 05/13
- Assembly Bottom Front Placard: 05/21 - 05/22
- Assembly Arm Motors: 05/23 - 05/26
- Assembly Horse Electronics: 05/26 - 06/07
- Assembly Top Plate: 06/07 - 06/09
- Assembly Protection: 06/09 - 06/09
- Check Full Assembly: 06/09 - 06/10

#### Testing
- Start: 06/06/23
- End: 06/22/23

- Perform Testing: 06/06 - 06/22
- Perform Motor Verification: 06/08 - 06/10
- Perform Communication Verification: 06/11 - 06/12
- Perform Controller Verification: 06/11 - 06/12
- Perform Waia Testing: 06/06 - 06/08
- Perform Independent system wa...: 06/06 - 06/08
- Perform integrated system wa...: 06/06 - 06/08
- Perform Safety Testing: 06/12 - 06/15
- Controller Disconnect: 06/13 - 06/13
- Controller connection during me...: 06/13 - 06/13
- E stop: 06/13 - 06/13
- Controller OFF Test: 06/13 - 06/13

#### Weight Testing
- Start: 06/22/23
- End: 06/22/23

- Weight limit testing: 06/22 - 06/22
- Weight limit and movement test: 06/22 - 06/22

#### Movement Verification
- Start: 06/22/23
- End: 06/22/23

- Perform Full system movement: 06/22 - 06/22
- Perform weighted movement test: 06/22 - 06/22
- Perform weight test case start...: 06/22 - 06/22
- Perform max weight movement t...: 06/22 - 06/22
- DUPO Sign-Off: 06/22 - 06/22

#### Project Wrap-up
- Start: 06/22/23
- End: 06/22/23

- Complete Documentation: 06/22 - 06/22
- Finalize CAD: 06/22 - 06/22
- Finalize Operations Manual: 06/22 - 06/22
- Finalization: 06/22 - 06/22
- Write PDR Report: 06/22 - 06/22
- Final Design Review (FDR): 06/23 - 06/23
- Clean out workspaces: 06/23 - 06/23
## Appendix E – Purchased Materials

### Purchase Request 1

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product Name</th>
<th>Part Number</th>
<th>Product Hyper Link</th>
<th>Qty</th>
<th>Price/Ea</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MELIFE 18650 Battery Shield, Micro USB V3 for ESP32 Battery Charging Module for Raspberry Pi and Arduino ESP32 WiFi with Micro USB Cable</td>
<td>B08D3MBVY8</td>
<td><a href="https://www.amazon.com/">https://www.amazon.com/</a></td>
<td>1</td>
<td>$7.99</td>
<td>$7.99</td>
</tr>
</tbody>
</table>

| Amazon Taxes    |                                                                 | 1           | $5.19                          |
| Amazon Shipping and Handling |                                                                | 1           | $10.44                         |
## Appendix E – Purchased Materials

<table>
<thead>
<tr>
<th>Vendor (name, website, phone, or fax)</th>
<th><strong>Product Name</strong> (paste the exact product title, include all text)</th>
<th><strong>Part Number</strong></th>
<th><strong>Product Hyper Link</strong></th>
<th><strong>Qty</strong></th>
<th><strong>Price/Ea</strong></th>
<th><strong>Total</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPST Rocker Switch ON-OFF Snap-In 10A@125VAC 6A@250VAC Black</td>
<td>316014</td>
<td><a href="https://www.jameco.com/">https://www.jameco.com/</a></td>
<td>2</td>
<td>$0.99</td>
<td>$1.98</td>
</tr>
<tr>
<td></td>
<td>10k 1/8W 20% 0.590&quot; Long Round Shaft 16mm Linear Taper Potentiometer (10 detents)</td>
<td>286273</td>
<td><a href="https://www.jameco.com/">https://www.jameco.com/</a></td>
<td>2</td>
<td>$1.69</td>
<td>$3.38</td>
</tr>
<tr>
<td></td>
<td>Orange 22 AWG Stranded Hook-up Wire 100 ft</td>
<td>2138089</td>
<td><a href="https://www.jameco.com/">https://www.jameco.com/</a></td>
<td>1</td>
<td>$10.95</td>
<td>$10.95</td>
</tr>
<tr>
<td></td>
<td>22 AWG Black Stranded Tinned-Copper Hook-up Wire 100 Feet</td>
<td>734311</td>
<td><a href="https://www.jameco.com/">https://www.jameco.com/</a></td>
<td>1</td>
<td>$10.95</td>
<td>$10.95</td>
</tr>
<tr>
<td></td>
<td>Hookup Wire 16 AWG Stranded (26x30) Tinned Copper Black 0,092 Inch (OD) PCV 25 Foot 300V 105C</td>
<td></td>
<td><a href="https://www.jameco.com/">https://www.jameco.com/</a></td>
<td>1</td>
<td>$6.95</td>
<td>$6.95</td>
</tr>
<tr>
<td></td>
<td>Hookup Wire 16 AWG Stranded (26x30) Tinned Copper Red 0,092 Inch (OD) PCV 25 Foot 300V 105C</td>
<td></td>
<td><a href="https://www.jameco.com/">https://www.jameco.com/</a></td>
<td>1</td>
<td>$6.95</td>
<td>$6.95</td>
</tr>
<tr>
<td><strong>Jameco Taxes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td><strong>$4.50</strong></td>
</tr>
<tr>
<td><strong>Jameco Shipping and Handling</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td><strong>$7.96</strong></td>
</tr>
<tr>
<td></td>
<td>AC Power Cable for IPC-3 and IPC35-CABLE110</td>
<td><a href="https://teknic.com/IPC35-CABLE110">https://teknic.com/IPC35-CABLE110</a></td>
<td>1</td>
<td>$14.00</td>
<td><strong>$14.00</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ClearCore™ - I/O and Motion Controller CLCR-4-13</td>
<td><a href="https://teknic.com/">https://teknic.com/produ</a></td>
<td>1</td>
<td>$99.00</td>
<td><strong>$99.00</strong></td>
<td></td>
</tr>
<tr>
<td><strong>Teknic Taxes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td><strong>$31.60</strong></td>
</tr>
<tr>
<td><strong>Teknic Shipping and Handling</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>Total</strong></td>
<td><strong>$26.65</strong></td>
</tr>
<tr>
<td>Vendor (name, website, phone, or fax)</td>
<td>Product Name (paste the exact product title, include all text)</td>
<td>Part Number</td>
<td>Product Hyper Link</td>
<td>Qty</td>
<td>Price/Ea</td>
<td>Total</td>
</tr>
<tr>
<td>-------------------------------------</td>
<td>-------------------------------------------------</td>
<td>-------------</td>
<td>--------------------</td>
<td>-----</td>
<td>---------</td>
<td>-------</td>
</tr>
<tr>
<td>Digikey, (800) 858-3616, <a href="https://digikey.com/">https://digikey.com/</a></td>
<td>Female 22AWG Terminal Molex 0039000046</td>
<td>39000046</td>
<td><a href="https://www.digikey.com/">https://www.digikey.com/</a></td>
<td>70</td>
<td>$0.07</td>
<td>$4.97</td>
</tr>
<tr>
<td>8-pin Connector Molex MiniFit-Jr 0039039082</td>
<td>39039082</td>
<td><a href="https://www.digikey.com/">https://www.digikey.com/</a></td>
<td>10</td>
<td>$0.61</td>
<td>$6.10</td>
<td></td>
</tr>
<tr>
<td>Male 16AWG Terminal Molex 0039000082</td>
<td>39000082</td>
<td><a href="https://www.digikey.com/">https://www.digikey.com/</a></td>
<td>20</td>
<td>$0.32</td>
<td>$6.40</td>
<td></td>
</tr>
<tr>
<td>Female 16AWG Terminal Molex 0039000078</td>
<td>39000078</td>
<td><a href="https://www.digikey.com/">https://www.digikey.com/</a></td>
<td>20</td>
<td>$0.16</td>
<td>$3.20</td>
<td></td>
</tr>
<tr>
<td>4-pin Connector Molex MiniFit-Jr 0039039042</td>
<td>39039042</td>
<td><a href="https://www.digikey.com/">https://www.digikey.com/</a></td>
<td>5</td>
<td>$0.62</td>
<td>$3.10</td>
<td></td>
</tr>
<tr>
<td>2-pin Connector Molex Sabre 044412002</td>
<td>444412002</td>
<td><a href="https://www.digikey.com/">https://www.digikey.com/</a></td>
<td>5</td>
<td>$0.82</td>
<td>$4.10</td>
<td></td>
</tr>
<tr>
<td>Female 16AWG Terminal Sabre 0433750001</td>
<td>433750001</td>
<td><a href="https://www.digikey.com/">https://www.digikey.com/</a></td>
<td>10</td>
<td>$0.11</td>
<td>$2.26</td>
<td></td>
</tr>
<tr>
<td><strong>DigiKey Taxes</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$3.06</td>
<td></td>
</tr>
<tr>
<td><strong>DigiKey Shipping and Handling</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$6.99</td>
<td></td>
</tr>
<tr>
<td><strong>TOTAL</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td><strong>$581.39</strong></td>
<td></td>
</tr>
</tbody>
</table>
## Purchase Request 2

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product Name</th>
<th>Part Number</th>
<th>Product Hyper Link</th>
<th>Qty</th>
<th>Price/Ea</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>CBJJ 3.7V 9900mAh Rechargeable Lithium ion Battery-18X67mm, (U.S. Shipping) (2 Pack)</td>
<td></td>
<td><a href="https://www.ams">https://www.ams</a></td>
<td>1</td>
<td>$17.99</td>
<td>$17.99</td>
</tr>
<tr>
<td>Amazon</td>
<td>Belkin Power Strip Surge Protector with 8 AC Multiple Outlets, 8 ft Long Flat Plug Heavy Duty Extension Cord for Home, Office, Travel, Computer Desktop, Laptop &amp; Phone Charging Bricks (2,500 Joules)</td>
<td></td>
<td><a href="https://www.ams">https://www.ams</a></td>
<td>1</td>
<td>$16.70</td>
<td>$16.70</td>
</tr>
<tr>
<td>Amazon</td>
<td>eDragon NEMA 5-15P to Standard Roj Power Cord, Black, 18/3 (18AWG 3 Conductor) SVT, 10 Amp / 125 Volt, 6 Foot</td>
<td></td>
<td><a href="https://www.ams">https://www.ams</a></td>
<td>1</td>
<td>$7.98</td>
<td>$7.98</td>
</tr>
<tr>
<td>Amazon</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Amazon Taxes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$6.21</td>
</tr>
<tr>
<td>Amazon Shipping and Handling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$</td>
</tr>
<tr>
<td>Teknic</td>
<td>ClearPath® - Integrated Servo System</td>
<td>CPM-MCVC-34415-RLS</td>
<td><a href="https://teknic.com">https://teknic.com</a></td>
<td>2</td>
<td>$402.00</td>
<td>$804.00</td>
</tr>
<tr>
<td>Teknic Taxes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$58.29</td>
</tr>
<tr>
<td>Teknic Shipping and Handling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$35.36</td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$964.52</td>
</tr>
</tbody>
</table>
## Purchase Request 3

<table>
<thead>
<tr>
<th>Vendor</th>
<th>Product Name</th>
<th>Part Number</th>
<th>Product Hyper Link</th>
<th>Qty</th>
<th>Price/Ea</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon</td>
<td>TWTADE / 22mm 2 NC Red Mushroom Latching Emergency Stop Push Button Switch 10A 600V (Warranty 3 years) YW1B-V4E02R</td>
<td>B07NP44T74</td>
<td><a href="https://www.amazon">https://www.amazon</a></td>
<td>1</td>
<td>$12.99</td>
<td>$17.99</td>
</tr>
<tr>
<td>Amazon Taxes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$1.14</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Amazon Shipping and Handling</td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Home Depot</td>
<td>Falken Design 36 in. x 72 in. x 1/8 in. Thick Aluminum Composite ACM White Sheet</td>
<td></td>
<td><a href="https://www.hon">https://www.hon</a></td>
<td>2</td>
<td>$98.72</td>
<td>$197.44</td>
</tr>
<tr>
<td>Home Depot</td>
<td>Everbilt 1 in. x 72 in. x 0.0625 in. Square Tube</td>
<td></td>
<td><a href="https://www.hon">https://www.hon</a></td>
<td>1</td>
<td>$25.71</td>
<td>$25.71</td>
</tr>
<tr>
<td>Home Depot Taxes</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$19.53</td>
<td></td>
</tr>
<tr>
<td>Home Depot Shipping and Handling</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$261.81</td>
<td></td>
</tr>
</tbody>
</table>
For any “Y” responses, the following is included on the next page:

<table>
<thead>
<tr>
<th>Y</th>
<th>N</th>
</tr>
</thead>
<tbody>
<tr>
<td>X 1. Will any part of the design create hazardous revolving, reciprocating, running, shearing, punching, pressing, squeezing, drawing, cutting, rolling, mixing or similar action, including pinch points and shear points?</td>
<td></td>
</tr>
<tr>
<td>X 2. Can any part of the design undergo high accelerations/decelerations?</td>
<td></td>
</tr>
<tr>
<td>X 3. Will the system have any large moving masses or large forces?</td>
<td></td>
</tr>
<tr>
<td>X 4. Will the system produce a projectile?</td>
<td></td>
</tr>
<tr>
<td>X 5. Would it be possible for the system to fall under gravity creating injury?</td>
<td></td>
</tr>
<tr>
<td>X 6. Will a user be exposed to overhanging weights as part of the design?</td>
<td></td>
</tr>
<tr>
<td>X 7. Will the system have any sharp edges?</td>
<td></td>
</tr>
<tr>
<td>X 8. Will any part of the electrical systems not be grounded?</td>
<td></td>
</tr>
<tr>
<td>X 9. Will there be any large batteries or electrical voltage in the system above 40 V?</td>
<td></td>
</tr>
<tr>
<td>X 10. Will there be any stored energy in the system such as batteries, flywheels, hanging weights or pressurized fluids?</td>
<td></td>
</tr>
<tr>
<td>X 11. Will there be any explosive or flammable liquids, gases, or dust fuel as part of the system?</td>
<td></td>
</tr>
<tr>
<td>X 12. Will the user of the design be required to exert any abnormal effort or physical posture during the use of the design?</td>
<td></td>
</tr>
<tr>
<td>X 13. Will there be any materials known to be hazardous to humans involved in either the design or the manufacturing of the design?</td>
<td></td>
</tr>
<tr>
<td>X 14. Can the system generate elevated levels of noise?</td>
<td></td>
</tr>
<tr>
<td>X 15. Will the device/system be exposed to extreme environmental conditions such as fog, humidity, cold, hot temperatures, etc?</td>
<td></td>
</tr>
<tr>
<td>X 16. Is it possible for the system to be used in an unsafe manner?</td>
<td></td>
</tr>
<tr>
<td>X 17. Will there be any other potential hazards not listed above? If yes, please explain on the reverse.</td>
<td></td>
</tr>
</tbody>
</table>

(1) a complete description of the hazard,
(2) the corrective action(s) you plan to take to protect the user, and
(3) a date by which the planned actions will be completed.
<table>
<thead>
<tr>
<th>Description of Hazard</th>
<th>Planned Corrective Action</th>
<th>Planned Date</th>
<th>Actual Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotating T joint can pinch or hit people if they touch it</td>
<td>There will be a sheet that goes along the top of the frame covering the moving components of the horse. This will reduce the number of moving components that a user could interact with directly.</td>
<td>6/3/2022</td>
<td></td>
</tr>
<tr>
<td>2-link arms have a pinch point at the elbow</td>
<td>The 2-link arms will be padded to prevent them from pinching anyone should they be touched during operation.</td>
<td>6/3/2022</td>
<td></td>
</tr>
<tr>
<td>Seat is a large moving mass</td>
<td>The system will be outfitted with a kill switch that is within reach of the instructor should the system behave strangely or otherwise be required to shut off in an emergency.</td>
<td>6/3/2022</td>
<td></td>
</tr>
<tr>
<td>Machined and stock metal could have sharp edges</td>
<td>We will debur all metal edges to prevent any sharp edges making it to the final design.</td>
<td>6/3/2022</td>
<td></td>
</tr>
<tr>
<td>The motors will be powered with 75V DC</td>
<td>The 75V DC system will be built to the manufactures specification and will be inspected by a Cal Poly electrician before first power up. All 75V cables will the insulated and covered to keep out of reach from users during operation.</td>
<td>6/3/2022</td>
<td></td>
</tr>
<tr>
<td>The Controller will have a Li-Ion battery and can be explosive</td>
<td>The Li-Ion battery will be electronically controlled by a pre-built system to prevent over charging and over discharging. We will mention to check the battery for bulging and overheating in the maintenance manual.</td>
<td>6/3/2022</td>
<td></td>
</tr>
<tr>
<td>Someone can stand on the device and turn it on causing injury</td>
<td>We will put a warning for this case in the operations manual as it is improper use of the mechanical horse.</td>
<td>6/3/2022</td>
<td></td>
</tr>
<tr>
<td>The is a possibility that someone falls of the device while in use</td>
<td>In this case there is going to be both an emergency stop on the horse and a stop on the controller to stop the horse as soon as possible to prevent further injury. Furthermore, there will always be a therapist on the side of the</td>
<td>6/3/2022</td>
<td></td>
</tr>
<tr>
<td>Hazard Description</td>
<td>Prevention Measures</td>
<td>Date</td>
<td></td>
</tr>
<tr>
<td>-----------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------</td>
<td>---------</td>
<td></td>
</tr>
<tr>
<td>Someone could accidentally turn on the device when someone is not properly situated and ready to start</td>
<td>We will put a warning for this case in the operations manual as it is improper use of the mechanical horse. In the case that this happens there is going to be both an emergency stop on the horse and a stop on the controller to stop the horse as soon as possible to prevent further injury.</td>
<td>6/3/2022</td>
<td></td>
</tr>
<tr>
<td>A person could turn off the horse with the controller in a running position and turn back on the horse.</td>
<td>In this case when the horse is turned back on the potentiometer will have to be returned to the stop position before moving it back to a running position for the horse to stop moving.</td>
<td>6/3/2022</td>
<td></td>
</tr>
</tbody>
</table>
Final Design Review
California Polytechnical San Luis Obispo
Senior Project 2021 - 2022

Aleya Dolorfino
adolorfi@calpoly.edu

Zuzanna Dominik
zdominik@calpoly.edu

Cade Liberty
cliberty@calpoly.edu

Peter Phillips
pphill03@calpoly.edu

Luke Watts
luwatts@calpoly.edu
# Table of Contents

1. Design Updates ................................................................................................................................. 1
   1.1 Mechanical Updates.......................................................................................................................... 1
   1.2 Software and Electrical Updates: .................................................................................................... 1
2. Manufacturing Plan............................................................................................................................... 1
   2.1 Procurement .................................................................................................................................... 2
   2.2 Manufacturing................................................................................................................................. 3
      2.2.1 Electrical System......................................................................................................................... 3
      2.2.2 Software .................................................................................................................................. 3
      2.2.3 Baseplate ................................................................................................................................. 4
      2.2.3 Motor Covers ............................................................................................................................ 4
      2.2.4 Arms and Arm Links.................................................................................................................. 5
      2.2.5 Holding Clips and Bars ............................................................................................................. 6
      2.2.6 Seat......................................................................................................................................... 6
   2.3 Challenges....................................................................................................................................... 6
      2.3.1 Arm Links ................................................................................................................................. 6
      2.3.2 Motor Covers ............................................................................................................................ 7
      2.3.3 Electrical System....................................................................................................................... 7
      2.3.4 Software System ...................................................................................................................... 7
3. Design Verification Chapter .................................................................................................................. 8
   3.1 Test descriptions ............................................................................................................................. 9
   3.2 Numerical Test Description ............................................................................................................ 12
   3.3 Failed Tests ................................................................................................................................... 17
   3.4 Lessons Learned............................................................................................................................ 17
4. Discussion and Recommendations .......................................................................................................... 18
   4.1 What we Learned ........................................................................................................................... 18
   4.2 What to do next.............................................................................................................................. 18
   4.3 Design Changes ........................................................................................................................... 19
   4.4 Manufacturing Changes ............................................................................................................... 19
   4.5 Future Production......................................................................................................................... 19
   4.6 Recommendations for use ............................................................................................................ 20
5. Conclusion ........................................................................................................................................ 20

Appendix A – Code Base ....................................................................................................................... A-Error! Bookmark not defined.
Appendix B – Final Budget ................................................................................................................... B-Error! Bookmark not defined.
1. Design Updates

1.1 Mechanical Updates
As we manufactured our system, we made 5 changes that we determined would be beneficial towards the use of our design. The first of these changes was swapping the top of the baseplate from 1 large cover to two small covers that only went over the motor and gearboxes. We determined that the large cover would be difficult to work with and that the material we received for the covers would make it difficult to assemble to the system in such a way that would enable easy access to the motors and gearboxes. We changed this design to a much smaller motor cover that would go over both motor assemblies and protect them while also leaving room on the inside for the electronics to be placed in between. On top of this, the motor covers would reduce the weight of the overall system allowing us to move the system more easily. The second update that we made was to have the bottom baseplate only on one side. We determined that the electronics were small enough to be fit inside a plastic box that would go in between the motor covers. Since we no longer needed the second bottom plate, we decided to remove it as it was only offering significant weight to the system. Our third update, as has been mentioned before, was to move the electrical box in between the motor covers. We deemed this beneficial as we would not need to run as much wire around the baseplate since the distance between the electricals and motor assembly could be dramatically reduced. The 4th edit that we made was to add a supporting member underneath the baseplate in order to support the motor assembly. After doing some calculations and working with the baseplate material, we realized that it would not be able to support the weight of the motor assembly during use and as such we added another bar to help offer support to the system. The final edit that we made was to add set screws to the arm links. In order to ensure that our arms would not come into contact with the motor shaft we had to place them all the way on the end of the motor shaft. This meant that the arms could fall off if they were not secured on the shaft, thus we added set screws to the arm links that would allow for the system to be placed onto the motor shaft securely and not worry about falling off.

1.2 Software and Electrical Updates:
There has been a major change in our electrical system as once delivered it was discovered that our Teknic ClearCore motor controller does not have a Wi-Fi module built in and is an optional addition, something the documentation did not make clear. To solve this problem, we have added a Raspberry Pi computer to the system to handle Wi-Fi and interface with the motor controller. This change was decided on as it was discussed in preliminary design talks and a team member already had a Raspberry Pi and was willing to donate it to the project. Since the Raspberry Pi has a much more sophisticated Wi-Fi module than the one that was expected to be on the ClearCore, we were able program the Raspberry to act as a Router and host a web page to act as an additional controller. This means that any internet capable device can control the horse, in addition to the horse controller, allowing the system to be more versatile.

2. Manufacturing Plan
To fabricate our final verification prototype, we needed to manufacture and assemble many different components together. In this section we will go over the methods that we used to procure all necessary material for our final prototype. From there we discuss the manufacturing steps taken to complete each of our subassemblies. Finally, we go over the procedures to assemble the final system together and create our final verification prototype.
2.1 Procurement

There are many constraints that shape an engineer’s design, including space limitations, weight limits, and sponsor preferences. One of the most important constraints is a team’s budget. Originally our team and the other senior project team working on the mechanical horse had a budget of $1000 to share. Knowing what we wanted to accomplish required a larger budget together we applied to the Baker Koob Grant and receive $2865 in additional funding. However, even with the additional funding, price is still one of the biggest concerns in our design process. Many of our design decisions, such as going over how many motors and gearboxes to get and what materials to make our base plate out, were dependent on how those decisions would affect the budget. With much thought and consideration, we adjusted our design to best fit our sponsors wants without overexerting the project’s budget.

Since we were a recipient of the Baker Koob Grant, our process of procuring our materials must follow a process of getting approval by both team members’ senior project coaches. Additionally, both the winter and fall senior project teams now have a shared budget sheet and must have a clear line of communication when it comes to buying materials for our project. We knew that the electrical components were going to be the most expensive so most of the additional funding we got from the grant went to our senior project team so we could purchase necessary electronics and motors. We are also aware electronic components would take a longer time to ship and require a longer lead time, so ordering the electronics became our priority when completing our first purchase orders.

The first purchase request consisted of mostly electrical components from Amazon and DigiKey that we knew were necessary for the electrical system of our prototype. Since our team was not able to finish the motor calculations by the time, we needed to send out the first purchase request, we needed to wait for the second purchase request to start ordering the more specific electrical components of the mechatronics system. The second purchase order contained some of the most important parts in our system, the motors and motor controller from Teknic. After the second purchase order, we had the majority of the electrical components to complete the verification prototype, all that was missing was the gearboxes to help increase the torque of our motor. The gearboxes were the hardest component to find but luckily, we were able to contact an international company called High Precision who were able to make a custom gearbox for us within our budget. We informed the company that we were part of a senior project team and that we had a strict budget we had to abide by, and they were happy to work with us to develop a gearbox. There was a lot of emailing back and forth across time zones to figure out the exact specification needed for the gearbox but in the end, we were able to receive a well discounted product that fit the exact specification we needed.

As for the procurement for the hardware for our Mechatronics system, most of it came from various amazon orders and a few trips to our local Home Depot. While most of the procurement for the hardware was from larger companies with the money from our budget, we did manage to get the sheet metal for our verification prototype donated to us from Southland Industries. Aleya interned with them for the past couple of summers and plans to continue her career there after graduation so we decided to see if they would be interested in helping her in the completion of her senior project. After a few phone calls to confirm the size and thickness of sheet metal Southland Industries had our order ready for pick up in a matter of a couple of days. Cade took the trip up to Union city to secure the sheet metal and bring it back to San Luis Obispo.
2.2 Manufacturing
We will elaborate on how we manufactured each subsystem and further talk about how these subsystems were assembled in the following sections.

2.2.1 Electrical System
To build our final verification prototype we built the power and data cables by cutting wire to length and crimping all the Molex a2nd Sabre terminals. These cables were then added to the system with the power cables running from the motors to the 75V power supply and the data cables running from motors to ClearCore motor controller. The Raspberry Pi connected to the ClearCore motor controller via a USB A to B cable. We made a cable to connect the 24V power supply to the ClearCore for power and wired in 110V AC power into our 24V power supply from our bought pigtailed US power cord. The Raspberry Pi and both the 75V and 24V power supplies plugged into the surge protector which plugged into the wall. The power cord from the surge protector was spliced and an emergency stop button was wired into the hot line while the earth ground was screwed into the emergency stop housing. Since the Emergency stop button housing is bolted to the baseplate the whole base was connected to earth ground and we connected two bolts with a 10-gauge wire to ground the seat steel to the base steel, we have ensured that all metal is grounded and protects user from static shocks and any loose wire shorting with the metal.

![Image 1: Electrical box of the mechatronic system](image)

2.2.2 Software
Since our Raspberry Pi now acts as an interface between the motor controller, which controllers motor movement and our horse controller, the Raspberry Pi runs multiple pieces of software. The Wi-Fi module on the Pi has been changed to act as an internet access point and router. It broadcasts its own SSID which any device with internet capability can connect to. Once connected, a device can access a Python tornado webserver hosting a webpage and WebSocket server. The webpage acts as an alternative to the horse controller and sends the same WebSocket commands to the Raspberry Pi as the horse controller. The Raspberry Pi then forwards all commands via USB serial communication to the motor controller. The motor controller uses Teknic’s Arduino ClearPath wrapper library to control the motors based on serial input from the Raspberry Pi.
2.2.3 Baseplate
To construct the baseplate, we took the 1/8\textsuperscript{th} inch thick sheet metal and used the water jet to cut the metal into the correct shape. Once we picked up the baseplate from the water jet, we made markings on the top of the plate to show where the holding bars and motor covers would be attached to the plate. We then took a hand drill and drilled holes at their marked destinations. After we drilled the holes into the baseplate, it was ready to be permanently fixed to the existing frame of the mechanical horse. We did this by first welding two 1/4\textsuperscript{th} square tube pieces to the existing frame that would lay across the center of the base plate to offer extra support. Then we took self-tapping metal screws to secure the baseplate into the bottom frame of the mechanical horse. First, we had to drill tapping holes into the baseplate and bottom frame and then follow it through with the self-tapping metal screws. Then after drilling four screws on each side of the baseplate and two on the back, the baseplate was fully secured.

![Figure 2: The baseplate on the frame of the horse](image)

2.2.3 Motor Covers
To manufacture the Motor Covers we performed several manufacturing steps. The first one was to take our 1/16\textsuperscript{th} inch steel plate and water jetted it to an expanded box shape with the two holes cut out, one for the arms to go through and the other for the wires to enter from. From there we then used the finger press to bend the box into shape. We then drilled holes into the bent-out feet. From there we added angle brackets to secure the box together and finally attached it to the baseplate using nuts and bolts to hold it into place. On the next page are the motor covers as they came out of the waterjet and then folded.

![Figure 3: The folded and unfolded motor covers](image)
2.2.4 Arms and Arm Links

To create the arms for the mechanical horse, we first measured the distance between the seat and the shaft of the gearbox. We then subtracted the length of the arm links to determine the length at which to cut the steel rods. After cutting the rods to length we used a ¾-16 die to make external threads onto the rod so the rod end bearing eyelets could be screwed onto each of the ends of the rod. Once the eyelets were secured, we worked on manufacturing the arm links. To manufacture the arm links, we used the waterjet to cut out the initial arm link shape out of a ½ inch steel plate. The shape resembled a teardrop shape with two holes, one with a shaft key cutout to allow the arm link to attack to the gearbox and one for the steel rod to be inserted to connect the arm to the arm link. To attach the steel rod to the water jetted piece, we cut a two-inch piece of steel rod, inserted it into the designated hole, and then welded it into place. Next, we allowed for the weld to cool and then added external threads to the steel rod with the same process as the die as earlier. We then drilled into the side of the arm link to make a hole for a set screw for the hole that would be slipped onto the gearbox shaft. To finalize the manufacturing process, we slipped the arm links onto the gearbox shaft, secured them onto the shaft by tightening the set screw, slipped the steel rod into the eyelet, and then secured the arms with a nut on the rod’s external thread.

Figure 4: The eyelets secured onto the arms

Figure 5: Holding bars and holding clips secured around the motor and gearbox
2.2.5 Holding Clips and Bars
To create the holding bars, we first took a 1.5-inch bar stock and cut it down to 12 inches. From there we marked where the holding bars would sit on the baseplate. Next, we drilled the necessary holes in both the bar stock and the baseplate. After lining up the holes we secured the holding bars to the baseplate with a few nuts and bolts. The next component was to make the holding clips by cutting inch thick strips of metal with the scraps of our 1/16\textsuperscript{th}-inch thick sheet metal about 12 inches long. We screwed one side of the metal strip onto the side of the holding bar, and then used gloved hands to bend the metal over the motor and gearbox combination to secure them into place. We finished the holding clip by screwing in the other end of the holding clip into the other holding bar.

![Figure 6: Modified wooden frame for seat of the horse](image)

2.2.6 Seat
To make the seat, we modified the seat that was created by the previous senior project working on this project. For the modifications we cut down the wood section of their seat to make it shorter and then attached all the pieces together. From there we drilled holes into the side to create offsets for the arms to attach to. And finally, we drilled, using wood to metal screws, into the wood and metal of the seat to attach them together.

2.3 Challenges
While manufacturing the horse, the team faced many challenges. From the software for the motors to the construction on the arm links, we needed to find separate ways to adjust components of our final design. This section will break down in more details the challenges we faced and the solutions we found to overcome said challenges.

2.3.1 Arm Links
For the arm links, the most difficult challenge was figuring out a way to attach it to the motor shaft. Due to spacing concerns on the horse, we had a clash concern if the motor shafts stuck too far past the links, as the arms would not be able to rotate in a full circle. However, if the links were placed on the ends, then there is a concern that they could slip off and cause damage to the links and baseplate as well as whatever is on the seat. Due to this concern we had to devise a way to ensure that the system would stay on the end of the motor shaft while also not have the concern with falling off the end. For this we decided first to create a press fit where we would force the link onto the shaft. However, as we made it, we oversized the holes and as such found that this would not work as a solution. Due to this we pivoted to the current
design which was to make setscrew that would push into the motor shaft creating a tight fit. This worked great as it was easier to remove the links and ensure that the system could be serviced.

2.3.2 Motor Covers
For the motor covers, the hardest part about manufacturing them was ensuring that the system was able to fully protect the motors while also allow the arms to spin freely. For this system we had to make two different iterations due to sizing concerns with the arms. Overall, we used a female eyelet that had a large diameter rather than the male eyelet that we were expecting to use. Due to this we had to ensure that the slot was wide enough to hold this extended size. For the first iteration, we did a fit test and found that with the slot the system hit the top of the motor cover as they were too large to the center. After finding this issue we created another set of motor covers that had widened slots to ensure that the arms were able to rotate freely inside the motor cover while also ensuring that the motor and gearbox were covered completely.

2.3.3 Electrical System
The primary concern with the electrical system was creating the box the electronics would be housed in. It required modifying a plastic box by cutting it in half and gluing it back together. Challenges arose when we realized that hot glue did not hold the box together sufficiently, and we had to resort to pressing the pieces together while the glue was still very hot so that the plastic would attach to the glue. Another issue we had was our electronics not fitting properly in the box. This was fixed by suspending some electronics from the top of the box using zip ties and sticky squares.

2.3.4 Software System
On the software development side there were several challenges that we faced. The first of these was the homing of the motors. For this homing means placing the arm links at a neutral position so that the top of the seat sat flat. For our system we wanted this to be where both links are pointed fully outwards. The reason this was an issue was because the motors don’t have a separate torque rating for whether torque is applied clockwise or counterclockwise, since there was no easy way to determine whether the arms were pointed outward or inward. Additionally, each motor would apply a different torque when pressing against the hard stop, thus making configuring the stoppage torque for each motor a challenge. On top of that, we had to ensure that the motors supplied enough torque to push the horse over the top of the motors but ensure that they didn’t produce too much that could damage the sensitive motor attachments.

The configuration of the DNS server for the Raspberry Pi was also challenging as the domain routing was complicated and dynamically assigning IP addresses for modern web devices is challenging. Particularly for overriding the DNS of a device to make the domain of “horse.io” point to a local IP rather than make a call to a global DNS for which the Raspberry Pi has no access to when in normal offline operation. The controller software also proved to be challenging as MicroPython is still in preliminary stages of development, especially with boards like the ESP-32, and common errors and fixes are not readily available online, so it took some trial and error to get correct. Programming the ESP-32 was also a challenge as most programming software only runs on Linux based machines for which our team did not have on hand and had to use hacky solutions to program the board in Windows and MacOS.
3. Design Verification Chapter

Overall, to verify that our design had met our specifications we performed a series of tests to check its motion and that it could hold the specified rider weight that we needed. As can be seen below in table 2, we had to meet the 10 specifications that we laid out in CDR. It is through these specifications we were able to verify if our system would function to the satisfaction of our sponsor. For these specifications we developed a series of 10 tests which are laid out in table 1 below. Many of these tests revolve around the different functions of the end movement of the horse. The reason for selecting many of these tests on the end movement of the horse was because we determined that this would show us the greatest number of problems and would allow us to have a goal to shoot for while debugging our system. Overall, it is through these tests that we were able to learn some of the limitations of our system and what needs to be fixed in the system.

<table>
<thead>
<tr>
<th>Test Name</th>
<th>Description</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weighted Motion Test (10 lbs.)</td>
<td>Is the system able to maintain constant motion while under a 10-pound weight?</td>
<td>Pass. The system was able to maintain constant motion with 10 lbs. on the saddle.</td>
</tr>
<tr>
<td>Weighted Motion Test (20 lbs.)</td>
<td>Is the system able to maintain constant motion while under a 20-pound weight?</td>
<td>Pass. The system was able to maintain constant motion with 20 lbs. on the saddle.</td>
</tr>
<tr>
<td>Weighted Motion Test (40 lbs.)</td>
<td>Is the system able to maintain constant motion while under a 40-pound weight?</td>
<td>Pass. The system was able to maintain constant motion with 40 lbs. on the saddle.</td>
</tr>
<tr>
<td>Weighted Motion Test (80 lbs.)</td>
<td>Is the system able to maintain constant motion while under an 80-pound weight?</td>
<td>Pass. The system was able to maintain constant motion with 80 lbs. on the saddle.</td>
</tr>
<tr>
<td>Weighted Motion Test (160 lbs.)</td>
<td>Is the system able to maintain constant motion while under a 160-pound weight?</td>
<td>Pass. The system was able to maintain constant motion with 160 lbs. on the saddle.</td>
</tr>
<tr>
<td>Motion Matches Walk Data</td>
<td>Does the motion produced match the recorded data with an error of 20%?</td>
<td>Fail. The motion that we created was based on ease of coding as we were unable to use the data, we recorded to create the movement.</td>
</tr>
<tr>
<td>Wear Test (Short Term Dust)</td>
<td>Does dust enter the system in a short period?</td>
<td>Pass. The system protected the motors and stopped any dust from collecting on the test platform.</td>
</tr>
<tr>
<td>Wear Test (Long Term Dust)</td>
<td>Does dust enter the system in a long period of time?</td>
<td>Pass. The system protected the motors and stopped any dust over a long period of time from collecting on the test platform.</td>
</tr>
<tr>
<td>Wear Test (Short Term Water)</td>
<td>Does water enter the system in a short period?</td>
<td>Pass. The system protected the motors and stopped any water from collecting on the test platform.</td>
</tr>
<tr>
<td>Wear Test (Long Term Water)</td>
<td>Does water enter the system in a long period of time?</td>
<td>Pass. The system protected the motors and stopped any water from collecting on the test platform.</td>
</tr>
</tbody>
</table>

Table 1: DVPR Tests
<table>
<thead>
<tr>
<th>Spec #</th>
<th>Specification Description</th>
<th>Requirement or Target</th>
<th>Tolerance</th>
<th>Compliance</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Withstand Rider Weight</td>
<td>160 lbs.</td>
<td>Min</td>
<td>Met</td>
</tr>
<tr>
<td>2</td>
<td>Mechatronics Weight</td>
<td>50 lbs.</td>
<td>Max</td>
<td>Not Met</td>
</tr>
<tr>
<td>3</td>
<td>Movement Matches Walk Data</td>
<td>Data within 20%</td>
<td>Target</td>
<td>Not Met</td>
</tr>
<tr>
<td>4</td>
<td>Accessible Parts</td>
<td>No permanent electronic connections</td>
<td>Target</td>
<td>Met</td>
</tr>
<tr>
<td>5</td>
<td>Overall Cost</td>
<td>$2365</td>
<td>Max</td>
<td>Met</td>
</tr>
<tr>
<td>6</td>
<td>Withstands Dust</td>
<td>Less than 1 gram inside test area</td>
<td>Target</td>
<td>Met</td>
</tr>
<tr>
<td>7</td>
<td>Withstands Water</td>
<td>Less than 10 ml inside test area</td>
<td>Target</td>
<td>Met</td>
</tr>
<tr>
<td>8</td>
<td>Minimal Buttons/Switches</td>
<td>3 buttons/switches</td>
<td>Max</td>
<td>Met</td>
</tr>
<tr>
<td>9</td>
<td>Cost Aesthetic Supplies</td>
<td>$20</td>
<td>Max</td>
<td>Met</td>
</tr>
<tr>
<td>10</td>
<td>Don’t Increase Size of Mech Horse</td>
<td>Our Parts Fit</td>
<td>Max</td>
<td>Met</td>
</tr>
</tbody>
</table>

Table 2: Engineering Specifications

3.1 Test descriptions

Place descriptions of each test and what happened during the tests

1. The Weighted Motion 10-pound test was performed to make sure the system ran when loaded with 10 pounds of ballast. We loaded the ballast onto the horse and made sure it was secured with zip-ties and duct tape. We ran the horse for 5 minutes and verified that the motion was consistent compared to the horse running unloaded. We made sure that there was no variation in motor movement and monitored the internal motor reading through Teknic MSP software to make sure each motor did not exceed torque or voltage limits while making sure the internal encoder reached its peak limits of oscillation. Having verified all these parameters were within our predefined limits, we concluded that this test passed.

2. The Weighted Motion 20-pound test was performed to make sure the system ran when loaded with 20 pounds of ballast. We loaded the ballast onto the horse and made sure it was secured with zip-ties and duct tape. We ran the horse for 5 minutes and verified that the motion was consistent compared to the horse running unloaded. We made sure that there was no variation in motor movement and monitored the internal motor reading through Teknic MSP software to make sure each motor did not exceed torque or voltage limits while making sure the internal encoder reached its peak limits of oscillation. Having verified all these parameters were within our predefined limits, we concluded that this test passed.

3. The Weighted Motion 40-pound test was performed to make sure the system ran when loaded with 40 pounds of ballast. We loaded the ballast onto the horse and made sure it was secured with zip-ties and duct tape. We ran the horse for 5 minutes and verified that the motion was consistent compared to the horse running unloaded. We made sure that there was no variation in motor movement and monitored the internal motor reading through Teknic MSP software to make sure each motor did not exceed torque or voltage limits while making sure the internal encoder reached its peak limits of oscillation. Having verified all these parameters were within our predefined limits, we concluded that this test passed.
4. The Weighted Motion 80-pound test was performed to make sure the system ran when loaded with 80 pounds of ballast. We loaded the ballast onto the horse and made sure it was secured with zip-ties and duct tape. We ran the horse for 5 minutes and verified that the motion was consistent compared to the horse running unloaded. We made sure that there was no variation in motor movement and monitored the internal motor reading through Teknic MSP software to make sure each motor did not exceed torque or voltage limits while making sure the internal encoder reached its peak limits of oscillation. Having verified all these parameters were within our predefined limits, we concluded that this test passed.

5. The Weighted Motion 160-pound test was performed to make sure the system ran when loaded with 160 pounds of ballast. We loaded the ballast onto the horse and made sure it was secured with zip-ties and duct tape. We ran the horse for 5 minutes and verified that the motion was consistent compared to the horse running unloaded. We made sure that there was no variation in motor movement and monitored the internal motor reading through Teknic MSP software to make sure each motor did not exceed torque or voltage limits while making sure the internal encoder reached its peak limits of oscillation. Having verified all these parameters were within our predefined limits, we concluded that this test passed.

6. The Motion Matches Walk Data test was performed to test the accuracy of changes in orientation on the mechanical horse in its walk pattern state relative to gyroscopic data that our team previously collected from a living horse. By recording angular data on the mechanical horse in a fashion similar to how data was collected on the live horse, we were able to compare the similarities between the two walking motions in terms of yaw, roll, pitch, and overall difference (using cosine similarity distance).
7. The short-term dust protection test was performed to ensure that over a short but intense period of dust exposure the protected zone inside the motor covers was free of dust. The protected zone is defined as the section underneath the motor cover where the motor would be located. For this test we grabbed a handful of dirt and threw it onto the motor covers and then recorded the amount of dust that was able to enter the protected zone. For this test we found that the system passed with no visible dirt found inside the protected zone.

![Figure 8: Short-term dust protection test](image)

8. The long-term dust protection test was performed to ensure that over a long period of dust exposure the protected zone remained dirt free. For this test we grabbed a handful of dirt and placed it onto the motor covers. We then left the motor covers inside the senior project room for 24 hours. We found for this test we found that the system passed with no visible dirt found inside the protected zone during the tested time.

9. The short-term water protection test was performed to ensure that over a short but intense period of water exposure the protected zone inside the motor covers was free of water. For this test we grabbed a small bottle of water and dumped it onto the motor covers and then recorded the amount of water that was able to enter the protected zone. For this test we found that the system passed with no visible water found inside the protected zone.

![Figure 9: Short-term water protection test](image)
The long-term water protection test was performed to ensure that over a long period of water exposure the protected zone remained water free. For this test we grabbed a small bottle of water and placed it onto the motor covers. We then left the motor covers inside the senior project room for 24 hours. We found for this test we found that the system passed with no visible water found inside the protected zone during the tested time.

Figure 10: Dry ground under motor cover after long-term water protection test

3.2 Numerical Test Description
To test the accuracy of the mechanical horse’s motion, our team used gyroscopic sensors to record the changes in orientation data as the mechanical horse performed cycles in its walking state. This newly collected data was smoothed with a low pass filter, mean adjusted, and trimmed so that it could be compared to the original collected data from the living horse’s movements. Once the two data sets were ready for comparison, we found their minimum, maximum, and average differences in yaw, roll, pitch, and cosine similarity scores. Below are the results of our automated data testing:

Figure 11: Live horse pitch over time
Figure 12: Live horse roll over time

Figure 13: Live horse yaw over time

Figure 14: Mechanical horse pitch over time
Figure 15: Mechanical horse roll over time

Figure 16: Difference between mechanical horse and live horse pitch over time
Figure 17: Difference between mechanical horse and live horse roll over time

Figure 18: Difference between mechanical horse and live horse yaw over time
Figure 19: Difference between mechanical horse and live horse cosine similarity over time

Yaw Diff min: -30.092367799999977 at index 113
Yaw Diff max: 26.274632200000028 at index 4347
Yaw Diff average: 1.7644197214394808e-14
Roll Diff min: -5.9834531779999995 at index 4778
Roll Diff max: 5.7873068220000015 at index 1777
Roll Diff average: 6.593836587853729e-16
Pitch Diff min: -3.1056815694600006 at index 2172
Pitch Diff max: 2.664198430540001 at index 4317
Pitch Diff average: -8.810729923425242e-17
Cos Sim min: -0.9992941888389115 at index 3931
Cos Sim max: 0.9998542569797372 at index 3474
Cos Sim average: 0.10561632006300446

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Index</th>
<th>Description</th>
<th>Value</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yaw Diff Min</td>
<td>-30.0923</td>
<td>113</td>
<td>Pitch Diff Min</td>
<td>-3.1057</td>
<td>2172</td>
</tr>
<tr>
<td>Yaw Diff Max</td>
<td>26.2746</td>
<td>4347</td>
<td>Pitch Diff Max</td>
<td>2.5541</td>
<td>4317</td>
</tr>
<tr>
<td>Yaw Diff Average</td>
<td>1.7644 x 10^-14</td>
<td></td>
<td>Pitch Diff Average</td>
<td>-8.8107 x 10^-17</td>
<td></td>
</tr>
<tr>
<td>Roll Diff Min</td>
<td>-5.9345</td>
<td>4778</td>
<td>Cos Diff Min</td>
<td>-0.9993</td>
<td>3931</td>
</tr>
<tr>
<td>Roll Diff Max</td>
<td>5.7873</td>
<td>1777</td>
<td>Cos Diff Max</td>
<td>0.9999</td>
<td>3474</td>
</tr>
<tr>
<td>Roll Diff Average</td>
<td>6.5938 x 10^{-16}</td>
<td>Cos Diff Average</td>
<td>0.1056</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 3: Minimum and Maximum difference of yaw, pitch, roll, and cosine similarity

3.3 Failed Tests
Overall, we had one failed test and 1 test that we were unable to perform. The test that we failed was the horse walking motion matching the data collected for a live horse walk. We found through our development of the system and the testing process that there were several issues that we could not solve in order to allow this test to pass. The first reason for this test failing was our difficulty with mathematically modeling our system to create motor angles. As with our data collection we were able to find the roll, pitch, and yaw angles from the top of the saddle. However, to control our system we created a model that transformed these angles into motor angles. After meeting with several Mechatronic professors, we developed a model and tried to use it to develop these angles, however the model itself would not solve for the angles correctly. We tried several methods to solve this issue but to no avail. Thus, due to time concerns we had to abandon the model in favor of creating a repeatable movement. Beyond that we also had issues programming the motors as when the motors were loaded, they would not hold constant torque when commanded to move at zero velocity, this resulted in the motors slowing moving down causing misalignment. In addition, we found out that there is no way to access the internal motor encoder so there was no way to accurately monitor to movement of the motor causing both safety concerns and concerns around the motors over rotating and causing the system to rip apart. Finally, we also had concerns mechanically that stopped us from being able to solve this issue. Overall, our seat was unconstrained for yaw motion. This meant that the seat was able to shift around thus changing the orientation of the seat and stopping us from being able to accurately model the system. This was an issue that was noticed late in the testing process and due to time concerns we were unable to develop a mechanical solution to the problem as we were informed by the other senior project working on this system that this issue would be dealt with.

3.4 Lessons Learned
Unfortunately, our testing showed us a lot about our design and brought up some troubling concerns that we believe need to be addressed before this system can be operational. Overall, we found that we were unable to match the data that we recorded from riding a live horse. There were various reasons for this which were discussed above. We also found several safety issues with our system. Overall, our motors and gearboxes would slide on the holding bars that we created. We placed the holding clips to stop this motion however these clips were not attached tightly enough to the motor or gearbox which allowed for slipping. On top of that the holding bars were not tall enough to allow for full rotation of the system through 360 degrees. The links when made were longer and although they were sanded down to try to fit there was a significant clash between them and the baseplate. Additionally, we found that our system was unconstrained in yaw rotation which would cause the seat to shift around significantly while the horse was moving. Finally, the last concern we found was a clash concern with the motor covers. The motor covers when bolted down were placed with the motor and gearboxes aligned at the back of the holding bars. However due to the shifting of the motor and gearboxes, they would shift forward causing the links to be able to run into the motor cover and damage it in one case. Thus, through testing we have made the decision that this system is unsafe to be used without engineering oversight until these issues are covered.
4. Discussion and Recommendations

At the completion of our project, we would like to clarify how it has impacted our professional growth and how we see the further development of this project could improve the prototype we have built.

4.1 What we Learned

Through this project we were able to gain a lot of experience and understanding how to take a project from ideation to completion. As we designed, refined, and manufactured the mechanical horse, we were able to expand not only our knowledge of electromechanical systems, but also use many of the skills we learned from our classes in a real-world application. Collaborating with a team over multiple quarters also served as a great insight into how to maintain professional relationships and how to translate our Cal Poly experiences into the workforce.

4.2 What to do next

We found there is still significant work that must be completed before our specifications are met. First, we would address the concern of the motor and gearbox assembly sliding. We found, through our testing, that the motor and gearboxes can slide along the holding bars and their position never comes to a complete stop. Due to this, we have issues with clearance for the holding bars as well as reliability concerns for moving the horse as the position of the motor and gearboxes changes how the system can move. We found that this issue comes from our holding clips, which should have held the motors in place were not placed tightly enough on the system to ensure the assembly would not slide. To solve this, we would place two end stops on the holding bars that would constrain the motion of the motor and gearbox assembly, thus allowing us to hold them in place.

The next step that we would take to continue work on this project would be the development of the walking pattern. To achieve this, our team would have to overcome the challenge of computing motor angles given live orientation data in one of two ways. The first option would be to correct the kinematic model used for prediction. The alternative option would be to reverse engineer the model using our mechanical horse. This second option was not feasible during this quarter due to the time required to manufacture, assemble, and program the mechanical horse, but now that is fully constructed to take in a range of input values to produce motion, there is a possibility that our team could run numerous tests on a wide range of input motor velocity timeseries values and record gyroscopic data from these walking motions. With this data set, we could reverse engineer a model that takes in changes in yaw, pitch, and roll as input and produces motor velocities at time indices using a second order linear regression model. Either of these approaches, if done correctly, would allow our team to find the motor velocities required to move the mechanical horse in a more accurate walking pattern relative to our live collected data.

Finally, as a last concern, we would like to add more horse-like components that would make the system feel more like a horse. Overall, the entire point of this project was to create a system that feels like a horse so when children ride it, they feel like they are riding a horse. We would like to continue this by adding more horse-like parts to the horse, such as adding a head or a tail and maybe even creating a neighing sound simulator that could create the sound of a horse. We figured that these components would ensure that the system felt like a horse and would give the best possible approximation of riding a horse as possible.
4.3 Design Changes

Overall, we found that our design could have been better in a couple of areas and ensure that we met our customer expectations. The first of these areas is to add a third yaw motor to the system. Overall due to budget concerns we were unable to buy a third motor that would be used to control the yaw motion of the horse. Due to this we were unable to fully constrain this motion which is why the system moves significantly in the yaw during use and outside of use. To stop this motion, we had to create a bump stop that the system would hit. Overall, we would like to control this motion and ensure that the bump stop is not needed to stop the system from over-rotating.

It would be beneficial to replace the motors on the horse with more capable Teknic MCPV motors instead of the Teknic MCVC motors currently on the horse. This would allow for better control of the motors with up to 16 positions instead of two and built-in hard stop homing functionality. We believed that the cost saving of the MCVC motors was worth it but found that these do not hold torque under velocity control mode when commanded at 0 velocity. To resolve this, we resulted to use the 2-position mode on the motors which held torque and were much more consistent in homing then the velocity control. The MCPV motors would allow more positions and would allow a more robust model to be simulated by the motors.

Additionally, we would like to redesign the electrical box for more airflow. There should be a filtered fan moving air through the electrical box to dissipate more heat as during testing we found that after running the horse for a 30-to-45-minute session there was heat buildup in the electrical box especially from the 75V power supply.

4.4 Manufacturing Changes

Overall, we believe we only have two changes that we would make to our manufacturing. The first change would be to assist more in the manufacturing of the frame. We found that there was a big disconnect between our team and the other senior project team and because of that we had different expectations for the frame than the one we got. Because of that we had to work on the frame, changing the seat and manufacturing the arms which we believed would have been completed by the other senior project team. Thus, we would have liked to have been more a part of the manufacturing for their team so that we could help put everything together as well as get a better understanding of what we were going to receive.

The next change that we would make for manufacturing would be to create simpler parts that had larger tolerances. We found that we had a lot of difficulty manufacturing the parts that we created and while we did not have to scrap many parts, we spent more time than we anticipated on these parts. To combat this, we would have liked to create simpler parts that did not require so many manufacturing steps and to create these parts to have higher tolerances so that we could more easily and speedily create the parts.

4.5 Future Production

Regarding further development of this device for production means we believe that another prototype should be completed before getting to full production of this device. Overall, we noticed many different concerns and problems with our device and would like to have another system that is better able to solve these challenges before feeling confident that this system can be produced and turned into a production element.
4.6 Recommendations for use
As can be seen from the recommendations above, we believe there is a significant amount of work that needs to be completed before we believe that the system is ready to be used with children. Thus because of these issues, we believe that the system should only be used in a testing and development context. Overall, the system itself works and can produce stable oscillation, however this motion does not resemble the motion of a horse. We have also done weighted testing and showed that the system itself can handle higher weights. However, the motion still needs to be refined to that of a horse. On top of that we would like to address some of the mechanical concerns with the horse and ensure that these are solved before placing a rider on the system. We found that the sliding of the motors, the collision of the arms with the baseplate, and the collision with the motor covers are all issues that need to be addressed before we are able to fully put a person on the system.

5. Conclusion
Through our senior design project process our team was to develop a mechanism that was able to move a seat in a repeated oscillating motion. While our final design of the therapeutic mechanical horse met many of our design criteria, it failed one of the most important tests of the design -- having our machine match the walking pattern of a live horse. With a little bit more time to perfect the model that translated the live horse data into motor angles and with better motors that would have given our programmers access to the encoder data within the motor, our team believes we could have delivered a more successful product.

However, despite the setback of replicating the exact movement of a live horse our final design for the mechanical horse did have its achievements. The horse was able to withstand water and dust very well, and the final oscillating movement that we settled on moved the rider in such a way that it the swinging movement produced was similar to walking. Additionally, the rider needed to use their core to keep themselves upright while riding, another aspect of equine therapy that benefited a rider physically. And while the final height of the mechanical horse was intimidating, it was to scale with a live horse so if getting used to height was needed to prepare a rider to a live horse then our verification prototype would have been helpful.

If we could have a second shot at this project, we think a better line of communication between us and the team before us would have made the biggest difference.
All code can also be found on our GitHub: https://github.com/peter-phillips/F76-Therapeutic-Mechanical-Horse

ClearCore Code:

Move2PosTest

/* Requirements:
 * 1. A ClearPath motor must be connected to Connector M-0 (left motor) and M-1 (right motor).
 * 2. The connected ClearPath motor must be configured through the MSP software
 *    for Move To Absolute Position, 2 Positions (Home to Switch) mode (In MSP
 *    select Mode>>Position>>Move to Absolute Position, then with "2 Positions
 *    (Home to Switch)" selected hit the OK button).
 * 3. Homing must be configured in the MSP software for your mechanical system
 *    (e.g. homing direction, switch polarity, etc.). To configure, click the
 *    "Setup..." button found under the "Homing" label on the MSP’s main window.
 * 4. The ClearPath motor must be set to use the HLFB mode "ASG-Position
 *    w/Measured Torque" with a PWM carrier frequency of 482 Hz through the MSP
 *    software (select Advanced>>High Level Feedback [Mode]... then choose
 *    "ASG-Position w/Measured Torque" from the dropdown, make sure that 482 Hz
 *    is selected in the "PWM Carrier Frequency" dropdown, and hit the OK
 *    button).
 * 5. The ClearPath must have defined Absolute Position Selections set up in the
 *    MSP software (On the main MSP window check the "Position Selection Setup
 *    (cnts)" box and fill in the two text boxes labeled "A off" and "A on").
 * 6. Ensure the Input A & B filters in MSP are both set to 5ms (In MSP
 *    select Advanced>>Input A, B Filtering... then in the Settings box fill in
 *    the textboxes labeled "Input A Filter Time Constant (msec)" and "Input B
 *    Filter Time Constant (msec)" then hit the OK button).
 * /

//NOTE:
//POSITIVE RPM = CCW Rotation
//NEGATIVE RPM = CW Rotation
#include "ClearCore.h"

// Defines the motor's connector as ConnectorM0
#define motorL ConnectorM0
#define motorR ConnectorM1

// Select the baud rate to match the target device.
#define baudRate 9600

#define INPUT_A_B_FILTER 5
int state;
uint32_t onTimeL = 0;
uint32_t onTimeR = 0;

int Lpos = 1;
int Rpos = 1;

int switchTime = 2000;

// Called once on ClearCore boot
void setup() {

    // Sets all motor connectors to the correct mode for Follow Digital
    // Velocity, Bipolar PWM mode.
    MotorMgr.MotorModeSet(MotorManager::MOTOR_ALL,
                            Connector::CPM_MODE_A_DIRECT_B_DIRECT);
    // Put the motor connector into the HLFB mode to read bipolar PWM (the
    // correct mode for ASG w/ Measured Torque)
    motorL.HlfbMode(MotorDriver::HLFB_MODE_HAS_BIPOLAR_PWM);
    motorR.HlfbMode(MotorDriver::HLFB_MODE_HAS_BIPOLAR_PWM);
    // Set the HLFB carrier frequency to 482 Hz
    motorL.HlfbCarrier(MotorDriver::HLFB_CARRIER_482_HZ);
    motorR.HlfbCarrier(MotorDriver::HLFB_CARRIER_482_HZ);
    // Sets up serial communication and waits up to 5 seconds for a port to open.
    // Serial communication is not required for this example to run.
    Serial.begin(baudRate);
    uint32_t timeout = 5000;
    uint32_t startTime = millis();
    while (!Serial || millis() - startTime < timeout) {
        continue;
    }
    Serial.println("active");
    state = -1;
}

void loop() {
    // Read the voltage on the analog sensor (0-10V).
    if(state == -1){
        Serial.println("Starting homing");
        Homing();
        Serial.println("Homing complete");
        state = 0;
// Checks Serial for new message
String serialIn = CheckSerial();
// Serial available and not in EM STOP MODE
if(serialIn != "" && state != 2){
  // Set to start ramping up horse to full speed
  if(serialIn == "on_h"){
    state = 1;
    onTimeL = millis();
    onTimeR = millis() + 1000;
  }
  // Set to start ramping down horse
  else if(serialIn == "off_h"){
    state = 0;
    motorL.MotorInA State(false);
    motorR.MotorInA State(false);
  }
  // Set to
  else if(serialIn == "em_stop"){
    state = 2;
    motorL.EnableRequest(false);
    motorR.EnableRequest(false);
  }
  // Print status of horse to Serial
  else if(serialIn == "stat_h"){
    PrintStatus();
  }
  // Manual test of homing sequence
  else if(serialIn == "home_test"){
    motorL.EnableRequest(false);
    motorR.EnableRequest(false);
    delay(100);
    Serial.println("Starting homing test");
    Homing();
    Serial.println("Homing test complete");
  }
}

// If horse on or off in ramp down
if(state == 1){
  if(millis() - onTimeL > switchTime){
    MoveToPositionL();
    onTimeL = millis();
  }
}
if (millis() - onTimeR > switchTime) {
    MoveToPositionR();
    onTimeR = millis();
}
}

bool MoveToPositionR() {
    // Check if an alert is currently preventing motion
    if (motorR.StatusReg().bit.AlertsPresent) {
        Serial.println("Motor status: 'In Alert'. Move Canceled.");
        return false;
    }

    int positionNum;
    if (Rpos == 1) {
        positionNum = 2;
        Rpos = 2;
    } else {
        positionNum = 1;
        Rpos = 1;
    }

    switch (positionNum) {
        case 1:
            // Sets Input A "off" for position 1
            motorR.MotorInAState(false);
            break;
        case 2:
            // Sets Input A "on" for position 2
            motorR.MotorInAState(true);
            break;
        default:
            // If this case is reached then an incorrect positionNum was entered
            return false;
    }

    // Ensures this delay is at least 2ms longer than the Input A, B filter
    // setting in MSP
    delay(2 + INPUT_A_B_FILTER);
    return true;
}
bool MoveToPositionL() {
    // Check if an alert is currently preventing motion
    if (motorR.StatusReg().bit.AlertsPresent) {
        Serial.println("Motor status: 'In Alert'. Move Canceled.");
        return false;
    }
    int positionNum;

    if (Lpos == 1)
        positionNum = 2;
    Lpos = 2;
    else {
        positionNum = 1;
        Lpos = 1;
    }

    switch (positionNum) {
    case 1:
        // Sets Input A "off" for position 1
        motorL.MotorInAState(false);
        break;
    case 2:
        // Sets Input A "on" for position 2
        motorL.MotorInAState(true);
        break;
    default:
        // If this case is reached then an incorrect positionNum was entered
        return false;
    }

    // Ensures this delay is at least 2ms longer than the Input A, B filter
    // setting in MSP
    delay(2 + INPUT_A_B_FILTER);
    return true;
}

// Checks Serial for new messages from Raspberry pi
String CheckSerial() {
    if (Serial.available() > 0) {
        String data = Serial.readStringUntil(\n);
        return data;
    }
}
// Read HLFB of right motor, returns percent of torque or 0 if no torque read
double ReadHlfbRight()
{
    // Check the current state of the ClearPath's HLFB.
    MotorDriver::HlfbState hlfbState = motorR.HlfbState();

    // Write the HLFB state to the serial port
    if (hlfbState == MotorDriver::HLFB_HAS_MEASUREMENT) {
        // Writes the torque measured, as a percent of motor peak torque rating
        return motorR.HlfbPercent();
    }
    else {
        return 0;
    }
}

// Read HLFB of left motor, returns percent of torque or 0 if no torque read
double ReadHlfbLeft()
{
    // Check the current state of the ClearPath's HLFB.
    MotorDriver::HlfbState hlfbState = motorL.HlfbState();

    if (hlfbState == MotorDriver::HLFB_HAS_MEASUREMENT) {
        // Writes the torque measured, as a percent of motor peak torque rating
        return motorL.HlfbPercent();
    }
    else {
        return 0;
    }
}

// Homing both motors to center
void Homing()
{
    // bool toCenterL = true;
    // bool toCenterR = true;

    bool nDoneL = true;
    bool nDoneR = true;

    motorL.EnableRequest(true);
    motorR.EnableRequest(true);
/limit torque to 5% of max
// LimitTorque(5);

// slowly rotate motors inward
// CommandVelocityL(-16);
// CommandVelocityR(16);

while (nDoneL || nDoneR){
  if(nDoneL && ReadHlfbLeft() < -4){
    Serial.println("left stopped");
    Serial.println(ReadHlfbRight());
    motorL.MotorInBState(true);
    motorL.MotorInAState(false);
    delay(100);
    motorL.MotorInBState(false);
    nDoneL = false;
  }
  Serial.println(ReadHlfbRight());
  if(nDoneR && ReadHlfbRight() > 7.2){
    Serial.println("right stopped");
    Serial.println(ReadHlfbRight());
    motorR.MotorInBState(true);
    motorR.MotorInAState(false);
    delay(100);
    motorR.MotorInBState(false);
    nDoneR = false;
  }
}

// Prints current status of horse to Serial
void PrintStatus(){
  if(state == 1){
    Serial.println("Horse is on and at full speed");
    return;
  }
  if(state == 0){
    Serial.println("Horse is off and ready");
    return;
  }
  if(state == 2){
    Serial.println("Emergency Stop was triggered on Horse, please cycle power on horse when safe to do so");
    return;
  }
}
HorseMovement-VelocityAndVariableTorque

/* Requirements:
 *
 * 1. A ClearPath motor must be connected to Connector M-0 and M-1
 *  
 * 2. The connected ClearPath motor must be configured through the MSP software
 * for Follow Digital Velocity Command, Bipolar PWM Command with Variable
 * Torque mode (In MSP select Mode>>Velocity>>Follow Digital Velocity
 * Command, then with "Bipolar PWM Command w/ Variable Torque")
 * 
 * 3. The ClearPath must have a defined Max Speed configured through the MSP
 * software (On the main MSP window fill in the "Max Speed (RPM)" box with
 * your desired maximum speed). Ensure the value of maxSpeed below matches
 * this Max Speed. Max Speed of motors is 840
 * 
 * 4. Set the PWM Deadband in MSP to 1.
 * 
 * 5. In MSP, ensure the two checkboxes for "Invert Torque PWM Input" and
 * "Invert Speed PWM Input" are unchecked.
 * 
 * 6. A primary Torque Limit and Alternate Torque Limit must be defined using
 * the Torque Limit setup window through the MSP software (To configure,
 * click the "Setup..." button found under the "Torque Limit" label. Then
 * fill in the textbox labeled "Alt Torque Limit (% of max)" and hit the
 * Apply button). Use only symmetric limits. These limits must match the
 * "torqueLimit" and "torqueLimitAlternate" variables defined below. Torque
 * limit should be 100% and alternate should be 5% for homing.
 * 
 * 7. The connected ClearPath motor must have its HLFB mode set to ASG with
 * measured torque through the MSP software (select Advanced>>High Level
 * Feedback [Mode]... then choose "ASG-Position, w/Measured Torque" or
 * "ASG-Velocity, w/ Measured Torque" and hit the OK button).
 * Select a 482 Hz PWM Carrier Frequency in this menu.
 * 
 */

//NOTE:
//POSITIVE RPM = CCW Rotation
/NEGATIVE RPM = CW Rotation
#include "ClearCore.h"

// Defines the motor's connector as ConnectorM0
#define motorL ConnectorM0
#define motorR ConnectorM1

// Select the baud rate to match the target device.
define baudRate 9600

// This is the commanded speed limit in RPM (must match the MSP value). This speed
// cannot actually be commanded, so use something slightly higher than your real
// max speed here and in MSP.
#define maxSpeed 845;

// Defines the default torque limit and the alternate torque limit
// (must match MSP values)
#define torqueLimit 100.0;
#define torqueLimitAlternate 5.0;

// A PWM deadband of 2% prevents signal jitter from effecting a 0 RPM command
// (must match MSP value)
#define pwmDeadBand 1.0;

int idxL;
int idxR;
int counter;
//0 = off
int state;
int ramp;
float threshold;
uint32_t onTimeL;
uint32_t onTimeR;

//max index of time model
define TIME_SERIES_MAX_IDX 4

// RPM of L and right motor from model
//long modelL[] = {50, 100, 50, -50, -100, -50, 25, -25, 25, -25};
//long modelR[] = {25, -25, 25, -25, -50, -100, -50, 50, 100, 50};

//duration in MS of corresponding RPM idx
//long modelTimeL[] = {300, 400, 300, 400, 400, 300, 200, 200, 200, 200};
//long modelTimeR[] = {200, 200, 200, 200, 300, 400, 200, 300, 400, 300};
Appendix A – Codebase

long modelL[] = {-55, 50, -55, 50};
long modelR[] = {-50, 50, -50, 50};
long modelTimeL[] = {500, 500, 500, 500};
long modelTimeR[] = {500, 500, 500, 500};

// Called once on ClearCore boot
void setup() {
    // Sets all motor connectors to the correct mode for Follow Digital
    // Velocity, Bipolar PWM mode.
    MotorMgr.MotorModeSet(MotorManager::MOTOR_ALL,
        Connector::CPM_MODE_A_PWM_B_PWM);
    // Put the motor connector into the HLFB mode to read bipolar PWM (the
    // correct mode for ASG w/ Measured Torque)
    motorL.HlfbMode(MotorDriver::HLFB_MODE_HAS_BIPOLAR_PWM);
    motorR.HlfbMode(MotorDriver::HLFB_MODE_HAS_BIPOLAR_PWM);
    // Set the HLFB carrier frequency to 482 Hz
    motorL.HlfbCarrier(MotorDriver::HLFB_CARRIER_482_HZ);
    motorR.HlfbCarrier(MotorDriver::HLFB_CARRIER_482_HZ);
    // Sets up serial communication and waits up to 5 seconds for a port to open.
    // Serial communication is not required for this example to run.
    Serial.begin(baudRate);
    uint32_t timeout = 5000;
    uint32_t startTime = millis();
    while (!Serial && millis() - startTime < timeout) {
        continue;
    }
    // Enables the motor
    motorL.EnableRequest(true);
    motorR.EnableRequest(true);
    Serial.println("Motor Enabled");
    // Set torque to limit 100% and command velocity to 0
    motorR.MotorInBDuty(0);
    motorL.MotorInBDuty(0);
    LimitTorque(100);
    // Set up initial variables
    idxL = 0;
    idxR = 0;
    counter = 0;
    state = -1;
    ramp = 0;
    threshold = 0;
void loop() {
    // Read the voltage on the analog sensor (0-10V).
    if (state == -1) {
        Serial.println("Starting homing");
        Homing();
        Serial.println("Homing complete");
        state = 0;
    }
    // Checks Serial for new message
    String serialIn = CheckSerial();
    // Serial available and not in E M STOP MODE
    if (serialIn != "" && state != 2) {
        // Set to start ramping up horse to full speed
        if (serialIn == "on_h") {
            state = 1;
            ramp = 1;
            threshold = 1;
            onTimeR = millis();
            onTimeL = millis();
        }
        // Set to start ramping down horse
        else if (serialIn == "off_h") {
            state = 0;
        }
        // Set to
        else if (serialIn == "em_stop") {
            state = 2;
            threshold = 0;
            motorL.EnableRequest(false);
            motorR.EnableRequest(false);
        }
        // Print status of horse to Serial
        else if (serialIn == "stat_h") {
            PrintStatus();
        }
        // manual test of homing sequence
        else if (serialIn == "home_test") {
            Serial.println("Starting homing test");
            Homing();
            Serial.println("Homing test complete");
        }
        else if (serialIn == "enable") {
            motorL.EnableRequest(true);
        }
    }
}
motorR.EnableRequest(true);
}
else if (serialIn == "disable"){
    motorL.EnableRequest(false);
    motorR.EnableRequest(false);
}

// Ramp up threshold to provide smooth start to movement
if (state == 1 && threshold < 1){
    threshold += .01;
}

// Ramp down threshold to provide smooth stop to movement
if (state == 0 && threshold > 0){
    threshold -= .01;
}

// If horse on or off in ramp down
if (state == 1 || (state == 0 && threshold > 0)){
    if (millis() - onTimeR > modelTimeR[idxR]){
        idxR += 1;
        onTimeR = millis();
    }
    // IDX rap around
    if (idxR > TIME_SERIES_MAX_IDX){
        idxR = 0;
    }
}

if (millis() - onTimeL > modelTimeL[idxL]){
    idxL += 1;
    onTimeL = millis();
}

// IDX rap around
if (idxL > TIME_SERIES_MAX_IDX){
    idxL = 0;
}

// new RPM with threshold factored
long commandedVelocityR =
    static_cast<int32_t>(round(modelR[idxR] * threshold));
long commandedVelocityL =
    static_cast<int32_t>(round(modelL[idxL] * threshold));
// Move at the commanded velocity.
CommandVelocityL(commandedVelocityL);
CommandVelocityR(commandedVelocityR);
}
// If off hold motors in position
else{
    motorR.MotorInBDuty(0);
motorL.MotorInBDuty(0);
}

/*------------------------------------------------------------------------------
* CommandVelocity
*
* Command the motor to move using a velocity of commandedVelocity
*
* Parameters:
* int commandedVelocity - The velocity to command in rpm
*
* Returns: True/False depending on whether the velocity was successfully
* commanded.
*/
bool CommandVelocityR(long commandedVelocity) {
    if (abs(commandedVelocity) >= abs(maxSpeed)) {
        Serial.println("Move rejected, requested velocity at or over the limit. Motor right");
        return false;
    }

    // Check if an alert is currently preventing motion
    if (motorR.StatusReg().bit.AlertsPresent) {
        Serial.println("Motor right status: 'In Alert'. Move Canceled.");
        motorR.MotorInBDuty(0);
        return false;
    }

    // If there is a deadband defined, the range of the PWM scale is reduced.
    double rangeUnsigned = 127.5 - (pwmdDeadBand / 100 * 255);

    // Find the scaling factor of our velocity range mapped to the PWM duty cycle
    // range (the PWM to the ClearPath is bipolar, so the range starts at a 50%
    // duty cycle).
    double scaleFactor = rangeUnsigned / maxSpeed;
// Scale the velocity command to our duty cycle range.
double dutyRequest;
if (commandedVelocity < 0) {
    dutyRequest = 127.5 - (pwmDeadBand / 100 * 255) + (commandedVelocity * scaleFactor);
} else if (commandedVelocity > 0) {
    dutyRequest = 127.5 + (pwmDeadBand / 100 * 255) + (commandedVelocity * scaleFactor);
} else {
    dutyRequest = 128.0;
}
// Command the move.
motorR.MotorInBDuty(dutyRequest);
return true;

bool CommandVelocityL(long commandedVelocity) {
    if (abs(commandedVelocity) >= abs(maxSpeed)) {
        Serial.println("Move rejected, requested velocity at or over the limit. Motor left");
        return false;
    }
    // Check if an alert is currently preventing motion
    if (motorL.StatusReg().bit.AlertsPresent) {
        Serial.println("Motor Left status: 'In Alert'. Move Canceled.");
        motorL.MotorInBDuty(0);
        return false;
    }
    // If there is a deadband defined, the range of the PWM scale is reduced.
    double rangeUnsigned = 127.5 - (pwmDeadBand / 100 * 255);
    // Find the scaling factor of our velocity range mapped to the PWM duty cycle
    // range (the PWM to the ClearPath is bipolar, so the range starts at a 50%
    // duty cycle).
    double scaleFactor = rangeUnsigned / maxSpeed;
    // Scale the velocity command to our duty cycle range.
    double dutyRequest;
    if (commandedVelocity < 0) {
        dutyRequest = 127.5 - (pwmDeadBand / 100 * 255) + (commandedVelocity * scaleFactor);
    } else if (commandedVelocity > 0) {
        dutyRequest = 127.5 + (pwmDeadBand / 100 * 255) + (commandedVelocity * scaleFactor);
    } else {
        dutyRequest = 128.0;
    }
    // Command the move.
motorR.MotorInBDuty(dutyRequest);
    return true;
}
+ (commandedVelocity * scaleFactor);
}
else if (commandedVelocity > 0) {
    dutyRequest = 127.5 + (pwmDeadBand / 100 * 255) + (commandedVelocity * scaleFactor);
}
else {
    dutyRequest = 128.0;
}

// Command the move.
motorL.MotorInDuty(dutyRequest);
return true;

// Limits torque of both motors
bool LimitTorque(double limit) {
    if (limit > torqueLimit || limit < torqueLimitAlternate) {
        Serial.println("Torque limiting rejected, invalid torque requested.");
        return false;
    }
    Serial.print("Limit torque to: ");
    Serial.println(limit);
    Serial.println("%.");

    // Find the scaling factor of our torque range mapped to the PWM duty cycle range (255 is the max duty cycle).
    double scaleFactor = 255 / (torqueLimit - torqueLimitAlternate);

    // Scale the torque limit command to our duty cycle range.
    int dutyRequest = (torqueLimit - limit) * scaleFactor;

    // Command the new torque limit.
    motorL.MotorInADuty(dutyRequest);
    motorR.MotorInADuty(dutyRequest);

    return true;
}

// Checks Serial for new messages from Raspberry pi
String CheckSerial(){
    if (Serial.available() > 0) {
        String data = Serial.readStringUntil(\n);
        Serial.println("You sent me: ");
Serial.println(data);
    return data;
}
return ""
}

// Read HLFB of right motor, returns percent of torque or 0 if no torque read
double ReadHlfbRight(){
    // Check the current state of the ClearPath's HLFB.
    MotorDriver::HlfbState hlfbState = motorR.HlfbState();

    // Write the HLFB state to the serial port
    if (hlfbState == MotorDriver::HLFB_HAS_MEASUREMENT) {
        // Writes the torque measured, as a percent of motor peak torque rating
        return motorR.HlfbsPercent();
    } else {
        return 0;
    }
}

// Read HLFB of left motor, returns percent of torque or 0 if no torque read
double ReadHlfbLeft(){
    // Check the current state of the ClearPath's HLFB.
    MotorDriver::HlfbState hlfbState = motorL.HlfbState();

    if (hlfbState == MotorDriver::HLFB_HAS_MEASUREMENT) {
        // Writes the torque measured, as a percent of motor peak torque rating
        return motorL.HlfbsPercent();
    } else {
        return 0;
    }
}

// Homing both motors to center
void Homing(){
    bool toCenterL = false;
    bool toCenterR = false;

    bool nDoneL = true;
    bool nDoneR = true;

    // limit torque to 10% of max
LimitTorque(10);

// slowly rotate motors outward
CommandVelocityL(-32);
CommandVelocityR(32);

while (nDoneL || nDoneR) {

    // if(toCenterL && ReadHlfbLeft() < 2)
    //    toCenterL = false;
    //    Serial.print("L CENTER: ");
    //    Serial.println(ReadHlfbLeft());
    // }

    // if(toCenterR && ReadHlfbRight() < 2)
    //    toCenterR = false;
    //    Serial.print("R CENTER: ");
    //    Serial.println(ReadHlfbRight());
    // }

    if (toCenterR && ReadHlfbRight() < 2.5) {
        Serial.println(ReadHlfbRight());
    }

    Serial.println(ReadHlfbLeft());
    if (nDoneL && !toCenterL && ReadHlfbLeft() < -2) {
        CommandVelocityL(0);
        Serial.print("left stopped: ");
        Serial.println(ReadHlfbLeft());
        nDoneL = false;
    }

    if (nDoneR && !toCenterR && ReadHlfbRight() > 2) {
        CommandVelocityR(0);
        Serial.print("right stopped: ");
        Serial.println(ReadHlfbRight());
        nDoneR = false;
    }

    // if(nDoneL && !toCenterL){
    //    Serial.println(ReadHlfbLeft());
    // }

    //start spinning motors back and reset torque delay to upright and stop
    CommandVelocityL(128);
    CommandVelocityR(-128);
    LimitTorque(100);
delay(1500);
CommandVelocityL(0);
CommandVelocityR(0);
}

// Prints current status of horse to Serial
void PrintStatus(){
    if(state == 1 && threshold < 1){
        Serial.println("Horse is on and ramping to full speed");
        return;
    }
    if(state == 1 && threshold == 1){
        Serial.println("Horse is on and at full speed");
        return;
    }
    if(state == 0 && threshold > 0){
        Serial.println("Horse is ramping down to a stop");
        return;
    }
    if(state == 0 && threshold == 0){
        Serial.println("Horse is off and ready");
        return;
    }
    if(state == 2){
        Serial.println("Emergency Stop was triggered on Horse, please cycle power on horse when safe to do so");
        return;
    }
}

// Horse Movement Test

/*
 * Title: FollowDigitalVelocity
 *
 * Objective:
 * This example demonstrates control of the ClearPath-MC operational mode
 * Follow Digital Velocity Command, Unipolar PWM Command.
 *
 * Description:
 * This example enables a ClearPath motor and executes velocity moves based
 * on the state of an analog input sensor. During operation, various move
* Statuses are written to the USB serial port.
* Consider using Manual Velocity Control mode instead if you do not wish to
* use an analog sensor to command velocity, if you require greater velocity
* command resolution (i.e. more commandable positions), or if HLF B is needed
* for "move done/at speed" status feedback.

* Requirements:
  1. A ClearPath motor must be connected to Connector M-0.
  2. The connected ClearPath motor must be configured through the MSP software
     for Follow Digital Velocity Command, Unipolar PWM Command mode (In MSP
     select Mode >> Velocity >> Follow Digital Velocity Command, then with
     "Unipolar PWM Command" selected hit the OK button).
  3. The ClearPath must have a defined Max Speed configured through the MSP
     software (On the main MSP window fill in the "Max Speed (RPM)" box with
     your desired maximum speed). Ensure the value of maxSpeed below matches
     this Max Speed.
  4. Ensure the "Invert PWM Input" checkbox found on the MSP's main window is
     unchecked.
  5. Ensure the Input A filter in MSP is set to 20ms, (In MSP
     select Advanced >> Input A, B Filtering... then in the Settings box fill in
     the textbox labeled "Input A Filter Time Constant (msec)" then hit the OK
     button).
  6. An analog input source (0-10V) connected to ConnectorA9 to control
     motor velocity.

* Links:
  ** ClearCore Documentation: https://teknic-inc.github.io/ClearCore-library/
  ** ClearPath Mode Informational Video: https://www.teknic.com/watch-video/#OpMode8

* Copyright (c) 2020 Teknic Inc. This work is free to use, copy and distribute under the terms of
  the standard MIT permissive software license which can be found at https://opensource.org/licenses/MIT
*/

#include "ClearCore.h"

// Defines the motor's connector as ConnectorM0
#define motorL ConnectorM0
#define motorR ConnectorM1

// The INPUT_A_FILTER must match the Input A filter setting in MSP
// (Advanced >> Input A, B Filtering...)
#define INPUT_A_FILTER 20

// Select the baud rate to match the target device.
#define baudRate 9600

#define timeSequenceTotalMS 1000
#define TIME_SERIES_MAX_IDX 1

// This is the commanded speed limit in RPM (must match the MSP value). This speed
// cannot actually be commanded, so use something slightly higher than your real
// max speed here and in MSP.
double maxSpeed = 845;

// Declares our user-defined helper function, which is used to command velocity
// The definition/implementation of this function is at the bottom of
// this example.
bool CommandVelocity(long commandedVelocity);

int idx;
int counter;
//0 = off
int state;
int ramp;
float threshold;
uint32_t onTime;

//long fakeVolt[] = {1, -1, 2, -2, 3, -3, 5, -5, 6, -6, 10, -10};
long fakeVolt[] = {9.9, -9.9, 9.9, -9.9, 9.9, -9.9, 9.9, -9.9, 9.9, -9.9, 9.9, -9.9};
long modelL[] = {200, -200};
long modelR[] = {-200, 200};
long modelTime[] = {500, 500};

void setup() {
    // Put your setup code here, it will only run once:
    // Sets all motor connectors to the correct mode for Follow Digital
    // Velocity, Unipolar PWM mode.
    MotorMgr.MotorModeSet(MotorManager::MOTOR_ALL,
                           Connector::CPM_MODE_A_DIRECT_B_PWM);

    // Sets up serial communication and waits up to 5 seconds for a port to open.
    // Serial communication is not required for this example to run.
```cpp
Serial.begin(baudRate);
uint32_t timeout = 5000;
uint32_t startTime = millis();
while (!Serial && millis() - startTime < timeout) {
    continue;
}

// Enable the motor
motorL.EnableRequest(true);
motorR.EnableRequest(true);
Serial.println("Motor Enabled");
idx = 0;
counter = 0;
state = 0;
ramp = 0;
threshold = 0;
}

void loop() {
    // Read the voltage on the analog sensor (0-10V).
    String serialIn = CheckSerial();
    if (serialIn != "" && state != 2) {
        if (serialIn == "on_h") {
            state = 1;
            ramp = 1;
            threshold = 0.001;
            onTime = millis();
        } else if (serialIn == "off_h") {
            state = 0;
        } else if (serialIn == "em_stop") {
            state = 2;
            threshold = 0;
            motorL.EnableRequest(false);
            motorR.EnableRequest(false);
        }
    } else if (state == 1 && threshold < 1) {
        threshold += .01;
        Serial.print("threshold: ");
        Serial.println(threshold);
    }
```

if (state == 0 && threshold > 0){
    threshold -= .01;
}

if (state == 1 || (state == 0 && threshold > 0)){
    if (millis() - onTime > modelTime[idx]){
        idx += 1;
        onTime = millis();
    }
    if (idx > TIME_SERIES_MAX_IDX){
        idx = 0;
    }
}

// Convert the voltage measured to a velocity within the valid range.
long commandedVelocityR =
    static_cast<int32_t>(round(modelR[idx] * threshold));
long commandedVelocityL =
    static_cast<int32_t>(round(modelL[idx] * threshold));

// Move at the commanded velocity.
Serial.println("Sending velocity");
Serial.println(commandedVelocityR);
CommandVelocityL(commandedVelocityR);
CommandVelocityR(commandedVelocityL);
} // See below for the detailed function definition.
else{
    motorR.MotorInBDuty(0);
    motorL.MotorInBDuty(0);
}

/* CommandVelocity
   * Command the motor to move using a velocity of commandedVelocity
   * Prints the move status to the USB serial port
   * Parameters:
   *   int commandedVelocity - The velocity to command
   * Returns: True/False depending on whether the velocity was successfully commanded.
*/
```cpp
bool CommandVelocityR(long commandedVelocity) {
    if (abs(commandedVelocity) >= abs(maxSpeed)) {
        Serial.println("Move rejected, requested velocity at or over the limit. Motor Right");
        return false;
    }

    // Check if an alert is currently preventing motion
    if (motorR.StatusReg().bit.AlertsPresent) {
        Serial.println("Motor Right status: 'In Alert'. Move Canceled.");
        return false;
    }

    if (commandedVelocity >= 0) {
        motorR.MotorInAState(false);
    } else {
        motorR.MotorInAState(true);
    }

    // Delays to send the correct filtered direction.
    delay(2 + INPUT_A_FILTER);

    // Find the scaling factor of our velocity range mapped to the PWM duty cycle range (255 is the max duty cycle).
    double scaleFactor = 255 / maxSpeed;

    // Scale the velocity command to our duty cycle range.
    int dutyRequest = abs(commandedVelocity) * scaleFactor;

    // Command the move.
    motorR.MotorInBDuty(dutyRequest);

    return true;
}

bool CommandVelocityL(long commandedVelocity) {
    if (abs(commandedVelocity) >= abs(maxSpeed)) {
        Serial.println("Move rejected, requested velocity at or over the limit. Motor Left");
        return false;
    }

    // Check if an alert is currently preventing motion
    if (motorL.StatusReg().bit.AlertsPresent) {
        Serial.println("Motor Left status: 'In Alert'. Move Canceled.");
        return false;
    }
```
```python
if (commandedVelocity >= 0) {
    motorL.MotorInAState(false);
} else {
    motorL.MotorInAState(true);
}
// Delays to send the correct filtered direction.
delay(2 + INPUT_A_FILTER);

// Find the scaling factor of our velocity range mapped to the PWM duty cycle range (255 is the max duty cycle).
double scaleFactor = 255 / maxSpeed;

// Scale the velocity command to our duty cycle range.
int dutyRequest = abs(commandedVelocity) * scaleFactor;

// Command the move.
motorL.MotorInBDuty(dutyRequest);
return true;
}

String CheckSerial(){
    if (Serial.available() > 0) {
        String data = Serial.readStringUntil(\n);  
        Serial.println("You sent me: ");
        Serial.println(data);
        return data;
    }
    return "";
}

// Web Server Code:

Server.py

#!/usr/bin/python

from asyncio.log import logger
```
import os.path
import tornado.httpserver
import tornado.websocket
import tornado.ioloop
import tornado.web
import logging
import CCInterface as CC
import time

# Logging config
logging.basicConfig(filename="serverLog.txt",
    filemode='a',
    format='%(asctime)s,%(msecs)d %(name)s %(levelname)s %(message)s',
    datefmt='%H:%M:%S',
    level=logging.DEBUG)

#Tornado Folder Paths
settings = dict(
    template_path = os.path.join(os.path.dirname(__file__), "templates"),
    static_path = os.path.join(os.path.dirname(__file__), "static"))

#Tornado server port and listen address
PORT = 80

# sudo nano /etc/udev/rules.d/99_usbdevices.rules
# should contain
# SUBSYSTEM="tty", ATTRS{idVendor}="2890", ATTRS{idProduct}="8022", SYMLINK="clearCore"
# This makes sure that port is always the same for the motor controller

inter = CCInterface("/dev/clearCore")

# Handler for web application, hosts index.html at local host
# When connected to raspberry pi this is 192.168.50.1 or Horse.io
class MainHandler(tornado.web.RequestHandler):
    # Renders index.html on get on /
    def get(self):
        print("[HTTP] (MainHandler) User Connected.")
        self.render("index.html")
        print("rendered")

# Web Socket handler on /ws for 192.168.50.1 or Horse.io on raspberry pi
class WSHandler(tornado.websocket.WebSocketHandler):
    # On open only print to log
def open(self):
    print('[WS] Connection was opened.')
    logger.info('[WS] Connection was opened.')

# On message switch on message and send to ClearCore through CCinterface
def on_message(self, message):
    print('[WS] Incoming message: ' + message)
    logger.info('[WS] Incoming message: ' + message)

    # Default return message
    horse_stat = "No communication from horse"

    # Message on_h turns on horse
    if message == "on_h":
        print("Sending on message to horse")
        logger.info("Sending on message to horse")
        inter.send(message)
        time.sleep(.1)
        horse_stat = inter.receive()

    # Message off_h turns off horse
    if message == "off_h":
        print("Sending off message to horse")
        logger.info("Sending off message to horse")
        inter.send(message)
        time.sleep(.1)
        horse_stat = inter.receive()

    # Message em_stop turns off horse immediately, disables motors and wait for power cycle to turn back on
    if message == "em_stop":
        print("Sending emergency stop message to horse")
        logger.info("Sending emergency stop message to horse")
        inter.send(message)
        time.sleep(.1)
        horse_stat = inter.receive()

    # Message stat_h retrieves current status of the horse
    if message == "stat_h":
        inter.send(message)
        time.sleep(.1)
        horse_stat = inter.receive(True)
    # Send status back to horse
    self.write_message(horse_stat)
# on close of web socket send turn off signal to horse

def on_close(self):
    print('[WS] Connection was closed.')
    logger.info('[WS] Connection was closed.')
    inter.send("off_h")

application = tornado.web.Application([  
    (r'/', MainHandler),
    (r'/ws', WSHandler),
], **settings)

if __name__ == '__main__':
    try:
        http_server = tornado.httpserver.HTTPServer(application)
        http_server.listen(PORT)
        main_loop = tornado.ioloop.IOLoop.instance()

        print('Tornado Server started')
        logger.info('Tornado Server started')
        main_loop.start()

    except Exception as e:
        print('Exception triggered - Tornado Server stopped.')
        print(e)
        logger.info('Exception triggered - Tornado Server stopped.')
        logger.info(e)

ClearCore Interface:

import serial

#Class for communicating with ClearCore over USB
class CCInterface():
    #Initialize serial communication with clearcore motor controller
def __init__(self, port, baudrate=9600):
        try:
            self.clearCore = serial.Serial(port=port, baudrate=baudrate, timeout=.1)
        except:
            self.clearCore = False

    #Send message to ClearCore Motor Controller
def send(self, message):
if not(self.clearCore):
    return

self.clearCore.write(bytes(message + '\n', 'utf-8'))

# Receive message to ClearCore Motor Controller, all allows you to pull multiple in case of stat_h
def receive(self, all=False):
    if not(self.clearCore):
        return "No connection to ClearCore and motors, try powering horse off and on again"

    line =
    while self.clearCore.in_waiting > 0:
        line = self.clearCore.readline().decode('utf-8').strip()
        print("Incoming from ClearCore: " + line)
        return line

    if len(line) == 0:
        return "No new status"

    if all:
        return '\n'.join(line)
    else:
        return line[-1]

Webpage

<html>
<head>
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1, user-scalable=no">
</head>
<title>Therapeutic Mechanical Horse</title>
<style type="text/css">
    .button {
        background-color: #4CAF50; /* Green */
        border: none;
        color: white;
        padding: 12px 24px;
        text-align: center;
        text-decoration: none;
        display: inline-block;
        font-size: 12px;
        margin: 4px 2px;
        -webkit-transition-duration: 0.4s; /* Safari */
        transition-duration: 0.4s;
        cursor: pointer;
    }
</style>
.button1 {
    background-color: white;
    color: black;
    border: 4px solid #4CAF50;
}

.button1:hover {
    background-color: #4CAF50;
    color: white;
}

.button2 {
    background-color: white;
    color: black;
    border: 4px solid #008CBA;
}

.button2:hover {
    background-color: #008CBA;
    color: white;
}

.button3 {
    background-color: white;
    color: black;
    border: 4px solid #f44336;
}

.button3:hover {
    background-color: #f44336;
    color: white;
}

.button4 {
    background-color: white;
    color: black;
    border: 4px solid #e7e7e7;
}

.button4:hover {
    background-color: #e7e7e7;
}

.button5 {
    background-color: white;
Appendix A – Codebase

```html
.color: black;
    border: 4px solid #555555;
}

.button5:hover {
    background-color: #555555;
    color: white;
}

</style>
</head>
<body>
<br>
<h1>Therapeutic Mechanical Horse</h1>
<h3>Turn On and Off The Horse Here</h3>
<button id="horse_off" class="button button5">OFF</button>
<button id="horse_on" class="button button1">ON</button>
<br>
<button id="horse_em_stop" class="button button3">EMERGENCY STOP</button>
<p>NOTE: EMERGENCY STOP will disable motors in Software, this is not a replacement for the physical Emergency Stop on the Horse</p>
<br>
<h2>Connection Status:</h2>
<br>
<div id="ws-status">Waiting...</div>
<br>
<h2>Horse Status:</h2>
<br>
<div id="horse-status">Waiting...</div>
<br>
<button id="horse_update_status" class="button button5">UPDATE STATUS</button>
<br>
<!-- Scripts -->
<script src="{{ static_url("jquery-1.8.3.min.js") }}"></script>
<script src="{{ static_url("ws-client.js") }}"></script>
</body>
```
WebSocket Client

```javascript
$(document).ready(function(){

    var WEBSOCKET_ROUTE = "/ws";

    if(window.location.protocol == "http:"{ //localhost
        var ws = new WebSocket("ws://" + window.location.host + WEBSOCKET_ROUTE);
    }
    else if(window.location.protocol == "https:"{ //Dataplicity
        var ws = new WebSocket("wss://" + window.location.host + WEBSOCKET_ROUTE);
    }

    ws.onopen = function(evt) {
        $("#ws-status").html("Connected");
    };

    ws.onmessage = function(evt) {
        $("#horse-status").html(evt.data);
    };

    ws.onclose = function(evt) {
        $("#ws-status").html("Disconnected");
    };

    $("#horse_on").click(function(){
        ws.send("on_h");
    });

    $("#horse_off").click(function(){
        ws.send("off_h");
    });

    $("#horse_em_stop").click(function(){
        ws.send("em_stop");
    });

    $("#horse_update_stat").click(function(){
        ws.send("stat_h");
    });
});
```
Controller Code:

Main

```python
from pin_manager import *
from wifi_manager import *

pin_manager = PinManager()
wifi_manager = WiFiManager()

horse_on = False
just_changed = change_cyc = 1000
pot_pos = 0
while True:
    pot_pos = pin_manager.read_pot(pot_pos)
    pressed = pin_manager.button_press()

    if just_changed == 0 and pressed:
        if horse_on:
            pin_manager.led.value(1)
            wifi_manager.send_code("off_h")
        else:
            pin_manager.led.value(0)
            wifi_manager.send_code("on_h")

        horse_on = not horse_on
        just_changed = change_cyc

    elif not pressed:
        if just_changed > 0:
            just_changed -= 1

Pin Manager

from machine import Pin, ADC
```
class PinManager:
    def __init__(self):
        self.__set_up_pins()

    def __set_up_pins(self):
        self.led = Pin(5, Pin.OUT)

        self.button = Pin(34, Pin.IN)
        self.button_power = Pin(32, Pin.OUT)
        # button high will be 3.3V
        self.button_power.value(1)

        self.pot = ADC(Pin(33))
        self.pot.atten(ADC.ATTN_11DB)

        # range of values for pot positions 0 is all left
        self.pos_range = [
            [4095, 4095], [4094, 3850], [3849, 3040],
            [3039, 2510], [2509, 2090], [2089, 1680],
            [1679, 1250], [1249, 780], [779, 380], [379, 0]]

    def read_pot(self, pot_pos):
        val = self.pot.read()

        # if value is outside of range, find new position
        if val > self.pos_range[pot_pos][0] or self.pos_range[pot_pos][1] > val:
            for i in range(0, len(self.pos_range)):
                # if value is in the position, return the new position
                if self.pos_range[i][1] <= val <= self.pos_range[i][0]:
                    return i

            print(pot_pos, val)

        return pot_pos

    def button_press(self):
        if self.button.value() > 0:
            # button not pressed
            self.led.value(0)  # led on
            return True
        else:
            # button pressed
            self.led.value(1)  # led off
            return False
WiFi Manager

```python
import network
import time
import machine
from network import WLAN
import uwebsocket.client

class WiFiManager:
    ssid = 'MechHorseP2P'
    password = '12345678910'

    def __init__(self):
        print("Connecting to WiFi")
        # self.__connect_static("192.169.50.13")
        self.__connect_dynamic()
        self.__connect_socket()

    def __connect_dynamic(self):
        # print("attempting connection")
        # ip 192.168.50.13
        sta_if = WLAN(network.STA_IF)

        # ap_if = network.WLAN(network.AP_IF)
        # # IP addr, netmask, gateway, DNS
        # print("ap config:", ap_if.ifconfig())
        # ap_if.active(False)

        sta_if.active(True)

        while not sta_if.isconnected():
            sta_if.connect(self.ssid, self.password)
            time.sleep(.5)

        # IP addr, netmask, gateway, DNS
        self.gateway = sta_if.ifconfig()[2]
        print("WiFi Connected:", sta_if.ifconfig())

    def __connect_static(self, ip_addr):
        wlan = WLAN(network.STA_IF)
        wlan.active(True)
        wlan.ifconfig((ip_addr, '255.255.255.0', '192.168.50.1', '8.8.8.8'))

        if not wlan.isconnected():
```
# change the line below to match your network ssid, security and password
wlan.connect(self.ssid, self.password)
while not wlan.isconnected():
    machine.idle() # save power while waiting

self.gateway = wlan.ifconfig()[2]
# print("WiFi Connected:", wlan.ifconfig())

def __connect_socket(self):
    self.websocket = uwebsockets.client.connect("ws://" + self.gateway + ":80/ws")

    connected = False
    while not connected:
        try:
            self.websocket.send("ping" + "\n")
            connected = True
        except:
            time.sleep(0.5)
    self.websocket = uwebsockets.client.connect("ws://" + self.gateway + ":80/ws")

def send_code(self, msg):
    self.websocket.send(msg + "\n")

def disconnect(self):
    self.websocket.close()
Dhcpd.conf

# A sample configuration for dhcpcd.
# See dhcpcd.conf(5) for details.

# Allow users of this group to interact with dhcpcd via the control socket.
#controlgroup wheel

# Inform the DHCP server of our hostname for DDNS.
hostname

# Use the hardware address of the interface for the Client ID.
clientid
# or
# Use the same DUID + IAID as set in DHCPv6 for DHCPv4 ClientID as per RFC4361.
# Some non-RFC compliant DHCP servers do not reply with this set.
# In this case, comment out duid and enable clientid above.
#duid

# Persist interface configuration when dhcpcd exits.
persistent

# Rapid commit support.
# Safe to enable by default because it requires the equivalent option set
# on the server to actually work.
option rapid_commit

# A list of options to request from the DHCP server.
option domain_name_servers, domain_name, domain_search, host_name
option classless_static_routes
# Respect the network MTU. This is applied to DHCP routes.
option interface_mtu

# Most distributions have NTP support.
#option ntp_servers

# A ServerID is required by RFC2131.
require dhcp_server_identifier

# Generate SLAAC address using the Hardware Address of the interface
#slaac hwaddr
# OR generate Stable Private IPv6 Addresses based from the DUID
slaac private
# Example static IP configuration:

```plaintext
#interface eth0
#static ip_address=192.168.0.10/24
#static ip6_address=fd51:42f8:caae:d92e::ff/64
#static routers=192.168.0.1
#static domain_name_servers=192.168.0.1 8.8.8.8 fd51:42f8:caae:d92e::1
```

# It is possible to fall back to a static IP if DHCP fails:

```plaintext
# define static profile
#profile static_eth0
#static ip_address=192.168.1.23/24
#static routers=192.168.1.1
#static domain_name_servers=192.168.1.1
```

# fallback to static profile on eth0

```plaintext
#interface eth0
#fallback static_eth0
interface wlan0
  static ip_address=192.168.50.1/24
  nohook wpa_supplicant
```

### dnsmasq.conf

```plaintext
interface=wlan0
dhcp-range=192.168.50.2,192.168.50.20,255.255.255.0,24h
domain=wlan
domain-needed
bogus-priv
no-resolv
server=8.8.8.8
server=8.8.4.4
address=/horse.io/192.168.50.1
```

### hostapd.conf

```plaintext
interface=wlan0
hw_mode=g
channel=7
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
```
wpa=2
wpa_key_mgmt=WPA-PSK
wpa_pairwise=TKIP
rsn_pairwise=CCMP
ssid=MechHorseP2P
wpa_passphrase=12345678910
<table>
<thead>
<tr>
<th>Test #</th>
<th>Specification</th>
<th>Test Description</th>
<th>Measurement</th>
<th>Acceptance Criteria</th>
<th>Required Facilities/Equipment</th>
<th>Parts Needed</th>
<th>Responsibility</th>
<th>TIMING</th>
<th>Notes on Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 10lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/26/2022</td>
<td>5/26/2022</td>
</tr>
<tr>
<td>2</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 20lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/26/2022</td>
<td>5/26/2022</td>
</tr>
<tr>
<td>3</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 40lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/26/2022</td>
<td>5/26/2022</td>
</tr>
<tr>
<td>4</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 80lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/26/2022</td>
<td>5/26/2022</td>
</tr>
<tr>
<td>5</td>
<td>Withstand Rider Weight</td>
<td>We will verify the system can maintain a constant walking pattern with a load of 160lbs</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data for the load must be within 10% of unladen movement</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly and weights</td>
<td>Peter, Zuzanna, Aleya, Cade, Luke</td>
<td>5/26/2022</td>
<td>5/26/2022</td>
</tr>
<tr>
<td>Test #</td>
<td>Specification</td>
<td>Test Description</td>
<td>Measurement/Equipment</td>
<td>Acceptance Criteria</td>
<td>Required Facilities/Equipment</td>
<td>Parts Needed</td>
<td>Responsibility</td>
<td>TIMING</td>
<td>Notes on Testing</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>------------------</td>
<td>-----------------------</td>
<td>---------------------</td>
<td>-------------------------------</td>
<td>--------------</td>
<td>----------------</td>
<td>--------</td>
<td>------------------</td>
</tr>
<tr>
<td>6</td>
<td>Movement Matches Walk Data</td>
<td>We will ensure the mechanical horse has a similar walk pattern to that of a real horse.</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data must be within 20% of the data we collected from a real horse</td>
<td>a phone to record matlab data and somewhere to plug in the system</td>
<td>Completed physical assembly</td>
<td>Luke</td>
<td>5/26/2022</td>
<td>5/26/2022</td>
</tr>
<tr>
<td>7</td>
<td>Withstands Dust</td>
<td>We will lightly sprinkle sand over the system and check to see if there is any that has filtered through our protection</td>
<td>Observation, is there too much dust</td>
<td>There must be little to no dust on the motors and electronics</td>
<td>a place to leave the system that will be secure, yet dusty. Possibly, our sponsor’s barn.</td>
<td>Baseplate assembly and sand</td>
<td>Cade</td>
<td>5/22/2022</td>
<td>5/22/2022</td>
</tr>
<tr>
<td>8</td>
<td>Withstands Dust</td>
<td>We will leave the system in an area that gathers dust and inspect how much dust got past our defenses.</td>
<td>Observation, is there too much dust</td>
<td>There must be little to no dust on the motors and electronics</td>
<td>a place to leave the system that will be secure, yet dusty. Possibly, our sponsor’s barn.</td>
<td>Baseplate assembly and sand</td>
<td>Cade</td>
<td>5/23/2022</td>
<td>5/23/2022</td>
</tr>
<tr>
<td>9</td>
<td>Withstands Water</td>
<td>We will lightly spray the system with water and see if any water reches past the protected surface</td>
<td>Observation, did any water reach the protected surface</td>
<td>Protected surface is dry somewhere outside where we can spill a bottle of water on the system.</td>
<td></td>
<td>Baseplate assembly and water</td>
<td>Cade</td>
<td>5/23/2022</td>
<td>5/23/2022</td>
</tr>
<tr>
<td>10</td>
<td>Withstands Water</td>
<td>We will spray the system in water enough to let it pool a small amount and leave it over time to ensure that the system is able to withstand water over time</td>
<td>Observation, did any water reach the protected surface</td>
<td>Protected surface is dry somewhere outside where we can spill a bottle of water on the system.</td>
<td></td>
<td>Baseplate assembly and water</td>
<td>Cade</td>
<td>5/23/2022</td>
<td>5/24/2022</td>
</tr>
</tbody>
</table>
## TEST PLAN

<table>
<thead>
<tr>
<th>Test #</th>
<th>Specification</th>
<th>Test Description</th>
<th>Measurement</th>
<th>Acceptance Criteria</th>
<th>Required Facilities/Equipment</th>
<th>Parts Needed</th>
<th>Responsibility</th>
<th>TIMING</th>
<th>Notes on Testing</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Functions on Different Surfaces</td>
<td>Run the system on a slight incline</td>
<td>Matlab data on seat oscillation</td>
<td>The matlab data must be within 30% of a seat on flat ground</td>
<td>somewhere with an outlet and slightly inclined ground.</td>
<td>completed physical assembly</td>
<td>Zuzanna and Peter</td>
<td>5/26/2022 5/26/2022</td>
<td>This test was a pass/fail test</td>
</tr>
</tbody>
</table>
Short Term Dust Exposure Test Procedure

Test Name:
Short Term Dust Exposure

Purpose:
The purpose of this test is to verify that the dust protection system that we put into place functions and reduces or eliminates the amount of contaminants that can come into the system.

Scope:
This test is for the protection of the electronics and motors as dust can wear down the components or cause them to stop working.

Equipment:
1. Fully completed top base plate with dust protection added
2. Sand or dust
3. A bucket
4. A scale

Hazards:
There are no hazards to this test.

PPE Requirements:
No PPE is required.

Facility:
This should occur outdoors where sand can be placed. The Bonderson Highbay should be sufficient for this test.

Procedure: (List number steps of how to run the test, can include sketches and/or pictures):
1) Weigh the bucket
2) Place the completed baseplate with dust protection over the bucket.
3) Lightly sprinkle a handful of sand or dust on top of the baseplate and protection elements
4) Check to see if there is any sand or dust in the bucket and weigh the final bucket weight
5) Repeat step 3 and 4 again but more forcefully place sand and dust on the system
6) Repeat steps 3 - 5 two more times to verify the system functions over time

Results: Pass Criteria, Fail Criteria, Number of samples to test
The resulting sand and dust in the bucket should be minimal. In order to verify that there is little sand
weight the bucket the weight should be within +/- .01 grams

Test Date(s):

5/2/22

Test Results:

Performed By:

Cade Liberty
Short Term Water Test Procedure:

Test Name:
Short Term Water Exposure Test

Purpose:
The purpose of this test is to verify that the water protection system that we put into place functions and reduces or eliminates the amount of water that can come into the system over a period of time.

Scope:
This test is for the protection of the electronics and motors as water can cause significant damage to the system if not properly secured against it.

Equipment:
1. Fully completed top base plate with dust protection added
2. Water
3. A bucket
4. A scale

Hazards:
There are no hazards to this test.

PPE Requirements:
No PPE is required.

Facility:
This should occur outdoors where sand can be placed. The Bonderson Highbay should be sufficient for this test.

Procedure: (List number steps of how to run the test, can include sketches and/or pictures):
1) Weigh the bucket
2) Place the completed base plate with water protection over the bucket.
3) Lightly spray water on top of the base plate and protection elements.
4) Let the water sit for a few seconds to a minute and
5) Weigh the bucket.
6) Repeat steps 3-5 but this time drop water on to the system from a bottle.
7) Repeat steps 3-5 but spray it with a hose.
8) Repeat steps 3-7 two more times to verify the security of the system.

Results: Pass Criteria, Fail Criteria, Number of samples to test.
The resulting sand and dust in the bucket should be minimal. In order to verify that there is little sand...
weight the bucket the weight should be within +/- .1 grams

**Test Date(s):**

5/6/22

**Test Results:**

**Performed By:**

Cade Liberty
Therapeutic Horse Test Procedure

Test Name:
Weight Limit Testing

Purpose: (This is the purpose of the test)
The purpose of this test is to see if our system is able to with maximum loading and maximum angle return to a neutral zone.

Scope: (Defines what feature or function the test is for)
This test tests the power of the motor to return the seat back to a flat position from a maximum load and maximum angle case. From this we will be able to verify that we are able to fully move the person on the system and verify that our bump stops work.

Equipment: (List of equipment necessary)
- Fully built and function system
- Many bags of rice
- Some sort of holder like tape or zip ties

Hazards: (list hazards associated with the test)
- Heavy moving weights
- Electrical power running through the system
- High Torque

PPE Requirements: (e.g. safety goggles, respirators)
Overall there is no PPE required but rather everyone should make sure to stay away from the system while it is moving.

Facility: (Where the test should occur)
This should occur in an open ad flat space such as the Bonderson High Bay or outside

Procedure: (List number steps of how to run the test, can include sketches and/or pictures):
1) Place rice bags on the system
2) Secure the rice bags to the system and ensure that they can remain still
3) Place the seat in the maximum angle (choose one of roll, pitch or yaw)
4) Power on the system
5) Run the motors and have the system return to the neutral point
6) Repeat steps 3-5 for the other two of Roll, Pitch and Yaw
7) Power off the system

Results: Pass Criteria, Fail Criteria, Number of samples to test
Overall for this test we need to see that the system is visibly able to return to the neutral point of a flat seat relative to ground from its worst case loading. If this cannot occur in Roll, Pitch or Yaw then the system fails and a solution needs to be determined. We plan on performing this test for each orientation case at least 3 times to verify that it works as expected.

Test Date(s):
June 2, 2022

Test Results:
A successful test will have the seat return to the neutral zone from any loading case.

Performed By: Everyone
Test Procedure Template

Test Name:
Weighted Walk Verification Test

Purpose: *(This is the purpose of the test)*
The purpose of this test is to confirm the measurements for motion in the Walking Motion Verification Test is not affected by additional weight.

Scope: *(Defines what feature or function the test is for)*
After the Walking Verification Test we will have a baseline for how our mechanical horse is expected to move, however results from the Walking Verification Test only account for the movement of the mechanical horse if no weight is applied to the seat. To ensure that the motion is constant in varying weights up to the 160lbs weight limit we will use strap varying amounts of sandbags to the seat of the mechanical horse and run the same Walking Verification Test.

Equipment: *(List of equipment necessary)*
- Fully Completed Horse System with all subassemblies attached
- Fully connected controller
- A phone with the MATLAB Mobile app downloaded
- A soft connection to attach the phone to the horse system
- Electrical power
- Sandbags or bags of rice
- Human rider

Hazards: *(list hazards associated with the test)*
- Large Moving Masses
- Electrical shortages

PPE Requirements: *(e.g. safety goggles, respirators)*
- Rider:
  - Helmet
  - Safety Glasses
  - Elbow and Knee pads
- Team:
  - No other PPE will be required for this test but while undergoing the test the test organizers should stand away from the device

Facility: *(Where the test should occur)*
This test should be performed in a large open space such as the Bonderson High Bay

Procedure: *(List number steps of how to run the test, can include sketches and/or pictures):*
1) Strap 40lbs worth of sandbags to the seat of the mechanical horse

2) Connect the horse to power

3) Power on the Controller and ensure that there is connection with the Horse

4) Ensure that the Safety systems are ready to be implemented but not engaged

5) Start the Phone collecting data

6) Turn the Controller to Walk mode

7) Set the Controller to stop mode

8) Power down the Horse

9) Stop and store data collection onto the phone

10) Repeat steps 1-9 three more times increasing the sandbag weight by 40lbs each time

11) Repeat steps 2-9 but instead of sandbag have the human rider sit on the horse

12) Ensure that device is fully shut off

13) Plot data received and find the maximum values for acceleration and orientation

**Results:** Pass Criteria, Fail Criteria, Number of samples to test

To verify that our system is unaffected by the varying weight we are going to need several different points of data. The first of this is the maximum values for the orientation data (Yaw, Pitch, and Roll) and the Acceleration data in the X, Y, and Z directions. We can then compare those data values to the average of the values found in the Walking Verification Test is the weighted data is within +/- 20% of the baseline data we can plot the data to further compared the two sets of data. From these graphs find a phase offset and verify that this angle is no larger than 20 degrees. If the collected data falls within these limits then it proves that our system accurately models horse movement and verifies our control scheme

**Test Date(s):** June 2, 2022

**Test Results:**

**Performed By:** Everyone
Long Term Dust Exposure Test Procedure:

**Test Name:**
Long Term Dust Exposure

**Purpose:**
The purpose of this test is to verify that the dust protection system that we put into place functions and reduces or eliminates the number of contaminants that can come into the system over a long period of time.

**Scope:**
This test is for the protection of the electronics and motors as dust can wear down the components or cause them to stop working.

**Equipment:**
1. Fully completed top base plate with dust protection added
2. Sand or dust
3. A bucket
4. A scale

**Hazards:**
There are no hazards to this test

**PPE Requirements:**
No PPE is required

**Facility:**
This should occur outdoors where sand can be placed. Inside of the Bonderson High Bay will be sufficient for this

**Procedure:** (List number steps of how to run the test, can include sketches and/or pictures):

1) Weigh the bucket
2) Place the completed baseplate with dust protection over the bucket.
3) Lightly sprinkle a handful of sand or dust on top of the baseplate and protection elements
4) Let the sand and dust sit on the system for several hours to days
5) Come back and weigh the bucket
6) Repeat step 3 and 4 again but more forcefully place sand and dust on the system

**Results:** Pass Criteria, Fail Criteria, Number of samples to test

The resulting sand and dust in the bucket should be minimal. In order to verify that there is little sand
weight the bucket the weight should be within +/- .01 grams

Test Date(s):

5/2/22-

5/6/22 Test

Results:

Performed By:

Cade Liberty
Long Term Dust Exposure Test Procedure:

Test Name:
Long Term Dust Exposure

Purpose:
The purpose of this test is to verify that the dust protection system that we put into place functions and reduces or eliminates the number of contaminants that can come into the system over a long period of time.

Scope:
This test is for the protection of the electronics and motors as dust can wear down the components or cause them to stop working.

Equipment:
1. Fully completed top base plate with dust protection added
2. Sand or dust
3. A bucket
4. A scale

Hazards:
There are no hazards to this test.

PPE Requirements:
No PPE is required.

Facility:
This should occur outdoors where sand can be placed. Inside of the Bonderson High Bay will be sufficient for this.

Procedure: (List number steps of how to run the test, can include sketches and/or pictures):
1) Weigh the bucket
2) Place the completed baseplate with dust protection over the bucket.
3) Lightly sprinkle a handful of sand or dust on top of the baseplate and protection elements
4) Let the sand and dust sit on the system for several hours to days
5) Come back and weigh the bucket
6) Repeat step 3 and 4 again but more forcefully place sand and dust on the system

Results: Pass Criteria, Fail Criteria, Number of samples to test
The resulting sand and dust in the bucket should be minimal. In order to verify that there is little sand
weight the bucket the weight should be within +/- .01 grams

Test Date(s):

5/2/22-
5/6/22 Test

Results:

Performed By:

Cade Liberty