UNIVERSITY OF COLORADO - BOULDER

ASEN 4028: DESIGN SYNTHESIS

CHAIR - Cuing via Haptics And Inner-ear Responses
Project Final Report
May 3, 2021

Customer:
Dr. Torin Clark

Advisor:
Dr. Allison Anderson

Author:
Cody BAHAN - 108186775

Author:
Rhys BASS - 109014538

Author:
Sarah FOLEY - 107887503

Author:
Lane FRANKLIN - 107695710

Author:
Cole GRAY - 108250504

Author:
Carter JACKSON - 105974103

Author:
Michelle LIN - 107635518

Author:
Jason MAGNO - 107634678

Author:
Baily RICE - 106987302

Author:
Andrew RINGER - 108730399

Author:
Dean WIDHALM - 107776308

Author:
Aiden WILSON - 106998669

College of Engineering & Applied Science
UNIVERSITY OF COLORADO BOULDER

Contents

List of Figures

List of Tables

List of Symbols

TCS	=	Tactile Cueing System
GVS	=	Galvanic Vestibular Stimulator
CPS	=	Central Processing System
m	=	Test Subject Mass
g	=	Gravitational Constant
F_g	=	Gravity Force on Test Subject
V_{max}	=	Max Output Voltage
I_{max}	=	Max Current
SNR	=	Signal-to-Noise Ratio
L	=	Inductance
R	=	Resistance
K	=	Control Gain

I. Project Purpose

Authors: Rhys Bass

In the last decade more military and civil aviation roles have begun being filled by Unmanned Aerial Vehicles (UAVs). UAVs provide many desirable advantages over traditional aircraft. First, the remote operation of such vehicles ensures that the pilots are not in danger and UAV's can be a lot cheaper to operate than manned aircraft. However, piloting a drone does have its drawbacks. Most notably, remote pilots lack situational and spatial awareness. This lack of spatial orientation sensation on the pilot can potentially lead to crashes due to pilot error in operation. If a drone is knocked off its equilibrium point a steady state roll rate may be created. This gradual change in angle can go undetected by remote pilots and eventually the aircraft may become unstable, resulting in damage or loss of aircraft. Our customer is looking to research the feasibility of providing aircraft attitude information to remote pilots, via additional sensory cues. The idea is to provide cues that could act as analogs of the sensations that a pilot would feel. Specifically, the feeling of the seat pressure and stimulation of the galvanic vestibular system, which is the inner-ear system responsible for our sense of orientation and tilt. The purpose of the CHAIR project is to design a system capable of providing these cues, which our customer can then use to research the effectiveness of providing information via these modes of cueing.

II. Project Objectives & Functional Requirements

Authors: Aiden Wilson

A. Project Objectives

The primary objective of the CHAIR system is to provide tactile and vestibular cues to create a sensation of tilt for a test subject. The success of the project is determined by which of the three levels of success were achieved. These levels of success were determined through conversations with the customer regarding what they would consider a successful design.

1) Discrete/Static Tilt Cueing: The system is capable of cueing a discrete and sustained tilt in either direction. The joystick feedback capabilities desired from the customer can be implemented at the lowest level with the subject determining the direction of tilt through joystick deflection.
2) Sinusoidal Tilt Cueing: The system is capable of cueing a continuous, sinusoidal tilt profile. This level corresponds to what could be described as a swaying motion.
3) Continuous Variable Tilt Cueing: The system is capable of developing a continuous tilt profile in real time through the use of joystick input. Additionally, a predetermined tilt profile can also be used and even combined in real time with the joystick input.

B. Concept of Operations & Functional Block Diagram

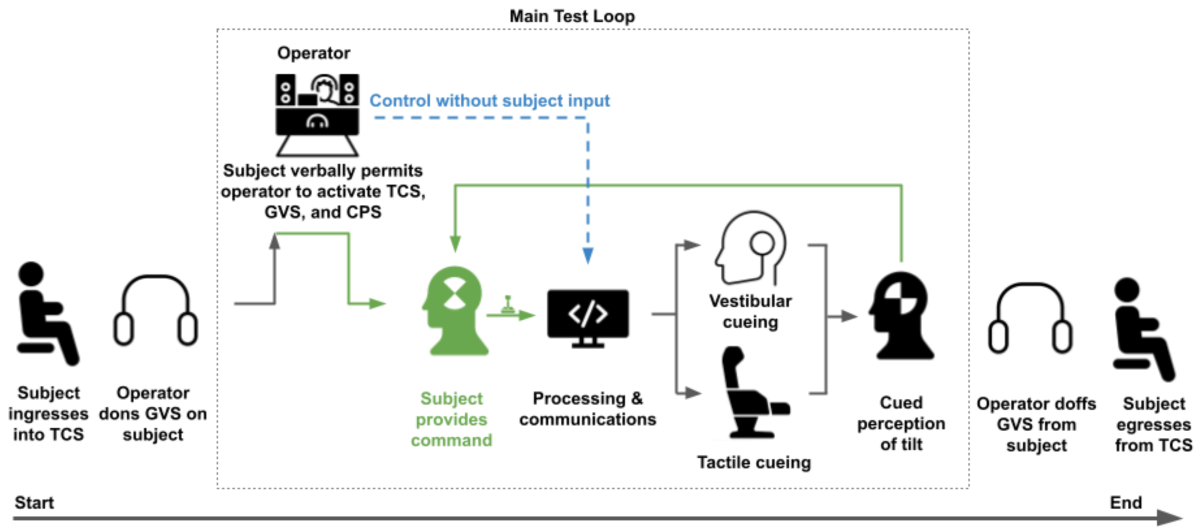


Fig. 1 CHAIR concept of operations.

Figure 1 shows the nominal operation of the CHAIR system. The process begins with the test subject, the person recieving tilt cues, ingressing into the Tactile Cueing System (TCS) by sitting in the chair. Then, the operator dons the Galvanic Vestibular Stimulator (GVS) onto the subject by applying electrodes behind the ears of the test subject. The test subject then permits the test operator, the person overseeing the test, to begin the main test loop. There are two modes of operation within the main test loop. The first of these modes of operation has the test subject providing tilt commands through the use of a joystick. These commands are then processed and communicated to the two cueing systems, vestibular and tactile. The two cueing systems work together to create a perception of tilt for the test subject and the test subject can form a feedback loop by responding to their perceived tilt with a new joystick command. The portions of the concept of operations that are unique to this mode of operation are highlighted in green in Fig. 1. The second mode of operation bypasses the subject input by having the test operator input predetermined tilt profiles directly into the processing communications stage and once again the two cueing systems convey this tilt to the test subject. The portion of the concept of operations that is unique to this mode of operation is highlighted in blue. Of note, the CHAIR system is not limited to a specific mode of operation at any point, meaning that both modes can be in use at the same time. For example, the test subject can use the joystick to alter a predetermined profile that is input by the test operator. Regardless of the mode of operation, the main test loop continues until the test subject and operator are satisfied. Once testing is complete, the electrodes are removed from the subject's head and the subject can egress from the TCS.

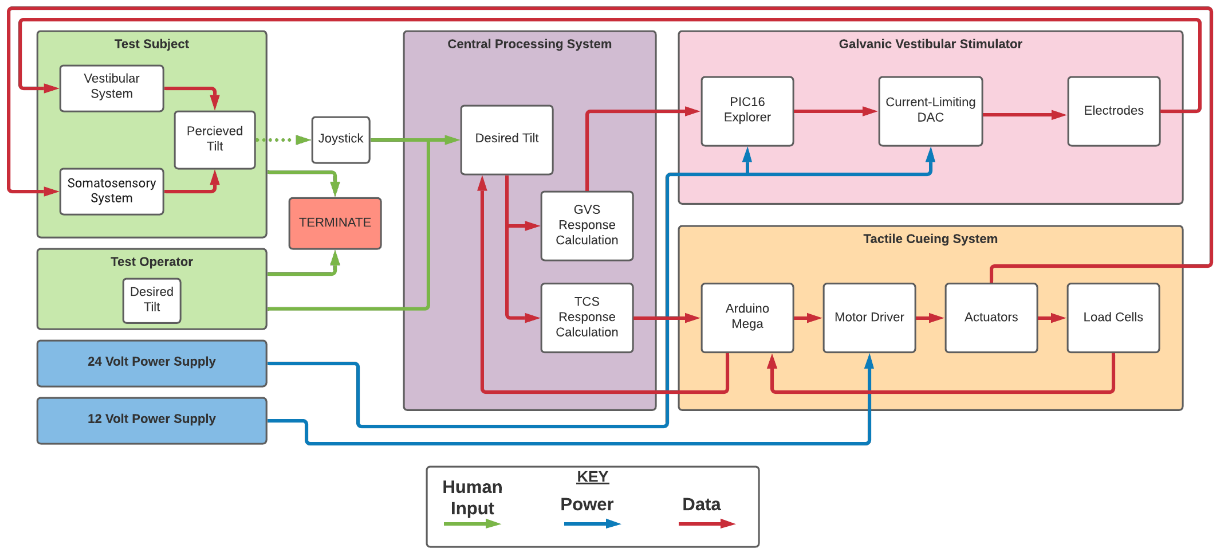


Fig. 2 Functional Block Diagram

The functional block diagram in Fig. 6 shows the interactions between the different subsystems that make up the CHAIR system. As shown in the green boxes on the left, either the test subject or test operator can input desired tilt into the Central Processing System (CPS). The test subject does this through the use of a joystick whereas the test operator inputs directly into the CPS. The CPS then takes this desired tilt and calculates the appropriate force response for the TCS and the appropriate current response for the GVS. For the TCS, this response is sent to an Arduino Mega microcontroller that commands motor drivers and actuators to apply a force to the sides of the test subject, stimulating the subject's somatosensory system. This force is measured by load cells which feed data back to the microcontroller as a control mechanism. The Arduino also sends this data back into the CPS to be used in the response calculations. This is necessary because the time needed for the GVS to command a current is significantly lower than the time needed for the actuators of the TCS to adjust to a force command, meaning that the two systems would be out of sync without some form of feedback. The GVS response, which accounts for the information being received by the CPS from the load cells, is sent to a PIC16 Explorer microcontroller which in turn commands a current-limiting DAC. The DAC controls the current that flows through the electrodes which controls the stimulation of the test subject's vestibular system. The somatosensory and vestibular stimulation of the test subject work in concert to create a perceived tilt for the test subject. Finally, the test subject and test operator are each capable of terminating the test at any point in time and all cueing will immediately stop.

C. Functional Requirements

The high-level functional requirements for the CHAIR system are as follows:

Requirement 1 - The test subject will receive vestibular and tactile cueing to convey representative flight orientation.

Rationale: This requirement encompasses the primary objective of the system as provided by the customer. The system should be capable of conveying tilt to a test subject through tactile and vestibular cues.

Requirement 2 - The test subject will not experience any pain caused by the operation of the system.

Rationale: This requirement is the overarching safety requirement for the system. Since the system is designed for use with human test subjects, it is essential that no pain or discomfort is experienced during use.

Requirement 3 - The GVS and TCS will be integrated to respond to test controller input as well as joystick input.

Rationale: This requirement specifies that the two cueing systems must be integrated to operate in conjunction with one another. The systems must be able to respond to input from the test controller but also respond to input from a joystick. The source of this requirement is that the system currently being used by the customer has few options for control over the system which limits the possible research options.

Requirement 4 - The TCS will be able to operate within the common conditions of a lab.

Rationale: This requirement dictates that the CHAIR system can be used in a common lab environment without the need for special equipment. This is due to the fact that the customer plans to use the CHAIR system within an existing lab space and does not want to make significant changes to the lab space in order to accommodate the CHAIR system.

Requirement 5 - The total development of the combined software and hardware systems must not exceed a total cost of 5,000 USD.

Rationale: The final requirement for CHAIR is the financial requirement. The team is provided with $5,000 and must be certain that the total cost of development for CHAIR does not exceed this limit as no additional funding can be expected.

III. Design Process & Outcome

Authors: Sarah Foley, Laney Franklin, & Aiden Wilson

A. Requirement Flow-down

Functional Requirement 1

The test subject will receive vestibular and tactile cueing to convey representative flight orientation.

Motivation: Customer specified requirement.

DR 1.1 - The Tactile Cueing System (TCS) shall provide tactile cueing in the form of skin pressure to the test subject.

Motivation: Customer specified requirement. Convey flight information through tactile stimulation.

DR 1.1.1 - The TCS shall provide a minimum of 0.384 psi on average over the stimulated areas.

Motivation: The TCS shall provide enough force to be perceived by the test subject.

DR 1.1.2 - The TCS system shall remain static to generate tactile cueing for the test subject.

Motivation: Customer specified requirement.

DR 1.1.3 - The TCS shall be able to cue a roll angle about the body x axis up to 15° from the nominal upright position.

Motivation: Customer specified requirement.

DR 1.2 - The Galvanic Vestibular Stimulator (GVS) shall provide vestibular cueing in the form of electrical currents through nodes placed on the mastoids of the test subject.

Motivation: Customer specified requirement.

DR 1.2.1 - The GVS shall be able to apply a range −4 m to 4 m of current to the test subject, inducing a sensation of roll.

Motivation: Customer specified requirement. Prevent disorientation from too high of angular velocity.

Functional Requirement 2

Test subject will not experience any pain caused by the operation of the system.

Motivation: Users must be able to comfortably use the system for the duration of the mission.

DR 2.1 - The TCS shall provide no more than 3.63 psi on the test subject. [1]

Motivation: Excessive pressure will lead to discomfort of the user.

DR 2.2 - The TCS and test subject shall be electrically grounded.

Motivation: Avoid electrical damages to the system and injury to the subject.

DR 2.3 - The GVS will have a maximum output that will not exceed 4mA of amperage. [2]

Motivation: Excessive current will lead to discomfort of the user as well as potential health hazards.

DR 2.4 - The CPS will not instruct the TCS or the GVS to exceed maximum allowed outputs described in requirements 2.1 and 2.3.

Motivation: Out of an abundance of caution, the software should not allow commands that might injure the user.

DR 2.5 - The CPS will be equipped with an emergency stop switch that will enable all functions to be terminated immediately.

Motivation: Out of an abundance of caution, all cueing can be stopped for whatever reason at any time.

Functional Requirement 3

The GVS and TCS will be integrated to respond to test controller input as well as joystick input.

Motivation: Customer specified requirement.

DR 3.1 - The TCS shall be provided with a method of communication to the Central Processing System (CPS).

Motivation: Customer specified requirement. The test subject will have control over cueing in a closed feedback loop.

DR 3.2 - The CPS shall coordinate the TCS and GVS responses, such that the time delay between the TCS and GVS cues as experienced by the test subject is less than 100 ms. [3]

Motivation: In order to avoid test subject disorientation, the GVS and TCS must cue the same profile with little to no time delay.

DR 3.3 - The CPS shall coordinate the hardware response such that the time delay from the joystick signal commanded by the test subject to the TCS and GVS cueing is within 200ms.

Motivation: In order to create an effective and useful cueing system, the cues must be actuated with little to no time lag from the user-inputted command.

DR 3.4 - The CPS shall incorporate a user interface allowing the controller to choose magnitude and direction of tilt rate cued by the GVS and TCS systems.

Motivation: Controller will be able to induce cueing on test subject prior to addition of joystick and for testing purposes outlined in Requirements 1 and 2.

Functional Requirement 4

The TCS will be operable within common conditions of a lab.

Motivation: Customer specified requirement. CHAIR will be suitable for use in customer research laboratory.

DR 4.1 - Will be able to operate within a space no larger than 6’ x 6’.

Motivation: Customer has limited laboratory space for large equipment.

DR 4.2 - The TCS shall accommodate a male body of 50th percentile height and 50th percentile weight.

Motivation: Ease of manufacturing and operation, fit to customer anthropometry.

DR 4.3 - The TCS shall have a minimum operable time of 30 min.

Motivation: Customer specified requirement. CHAIR will be able to last for an entire flight simulation without intervention.

DR 4.4 - The GVS and TCS systems will be able to operate repeatedly without losing functionality.

Motivation: CHAIR will be used over many flight simulations.

DR 4.4.1 - The joystick shall be designed to withstand 44.8 lbf in all directions.

Motivation: Ensure the joystick does not break upon usage given 80% of the maximum arm exertion force of the 50th percentile male, at a -30 degree angle from horizontal.

DR 4.4.2 - The GVS and TCS shall not lose functionality over multiple uses.

Motivation: The GVS and TCS should be able to withstand many uses without degradation in output.

Functional Requirement 5

The total development of the combined software and hardware systems must not exceed a total cost of 5,000 USD.

Motivation: The team is provided with five thousand US dollars by the AES department.

B. CHAIR Design

Authors: Sarah Foley, Dean Widhalm, Carter Jackson, Cody Bahan, Laney Franklin

1. TCS

To meet the above requirements, the team created the following final structural design for the TCS, as shown in Figure 3 below.

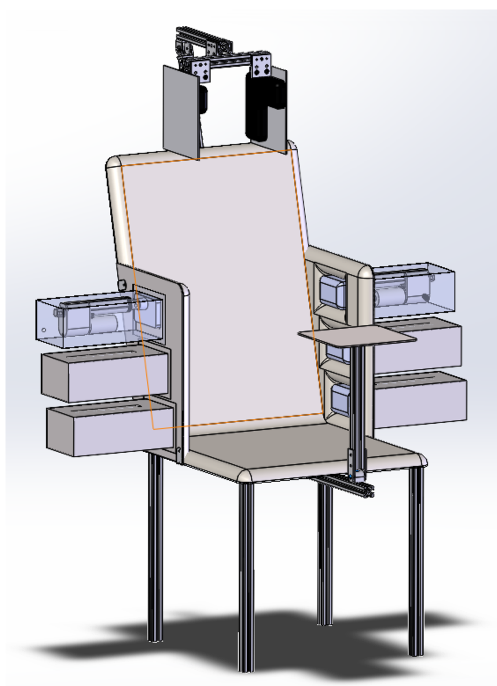


Fig. 3 TCS Final Design

As seen above, there are three main elements to the TCS design. Firstly, the six mechanisms housings on the sides of the chair contain the actuator-load cell assemblies which will provide the cueing to the test subject. Next, the joystick table holds the joystick at an appropriate distance and height for the test subject to use during operations. Finally, there is also an adjustable headrest to stabilize the test subject and hold their head in place during operations. Being designed to firstly accommodate the customer, the following critical dimensions of the chair were chosen, as shown in Table 4 below.

Component	Dimension	Value
Seat Pan	depth	18 in
	width	19 in
Seat Back	height	26 in
	width	19 in
	inclination	15 deg
Side Panel	height	17 in
	width	10 in
Legs	height	17 in

Fig. 4 TCS Critical Dimensions

The rational behind these dimensions were the measured anthropometric dimensions of the customer. It should also be noted that the seat pan, back and sides design also included a one inch foam upholstery for comfort of the test subject. The selected materials for the design were 1/8" stainless steel for all components with the exception of the mechanisms housings and headrest plates, which were chosen to be made of 1/32" stainless steel. Other various components such as the legs, joystick table and headrest are constructed from 80/20 members with accompanying accessories.

The next main assembly of the TCS is inside the mechanisms housings. The final design for this assembly is shown in Figure 5 below.

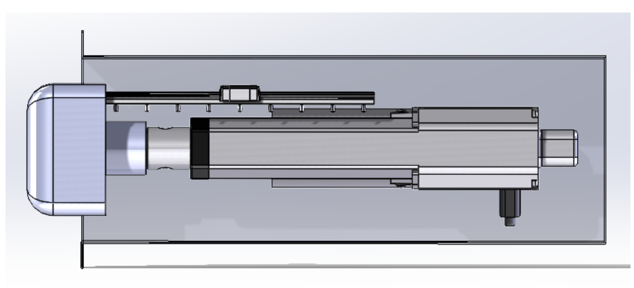


Fig. 5 Mechanisms Housing Design

This assembly, from left to right, is firstly comprised of a pressure module or pad, that will make contact with the test subject. This pad is 2.5 x 4" and constructed of a wood block upholstered with a 1/2" piece of foam. Attached to the backside of this pressure pad is a button load cell fastened by screws. Next is the actuator itself which is sitting on top of a wooden spacer which is not shown in the above model. Finally, the design includes two linear guides secured to the backside of the pressure module as well and also fastened to the housing through wooden spacers which are not shown.

The TCS design functions by applying force to the test subject distributed over the area of the pressure modules onto the test subject. The actuators can extend up to 3 inches to apply the force to the test subject through the load cell and pressure module, all while in a closed feedback loop with the load cell to get a real time feedback of the force being applied to the test subject. The linear guides keep the pressure module in line with the actuator and negate any off axis loading on the load cell and test subject.

The electrical system for the TCS consisted of six individual circuits all of which were connected to a power supply and microcontroller. The power supply outputs 12 V to the main circuit board, where six individual motor drivers are controlled by an Arduino MEGA 2560 microcontroller. These motor drivers distribute the necessary power to each linear actuator to provide the user with the appropriate cueing response. Finally, load cells are attached to the end of each actuator where the force data is sent back to the microcontroller for closed-loop feedback response.

The electrical design needed to fulfill all relevant design requirements that were previously discussed. DR 1.1.3 states that the TCS shall be able to cue a roll angle about the body x axis up to 15°from the nominal upright position. Additionally, DR 2.1 states that the TCS system shall provide no more than 3.63 psi on the test subject. From our design research, the team found that each individual actuator should receive no more than 2.7 A for a 15°cue, and through this, hardware and software systems were in place to ensure each actuator received no more than 3 A.

To achieve this, six motor drivers (one for each actuator) were incorporated to the circuit to help drive the TCS actuators. These motor drivers would receive a PWM signal from an Arduino Mega 2560. This PWM signal could be manipulated, and the duty cycle of this signal directly relates to the force output on the user. Finally, six load cells connected to the actuators sense the force output from the actuators and send the data back to the microcontroller for closed-loop feedback control that meets the team's time delay requirements.

2. GVS

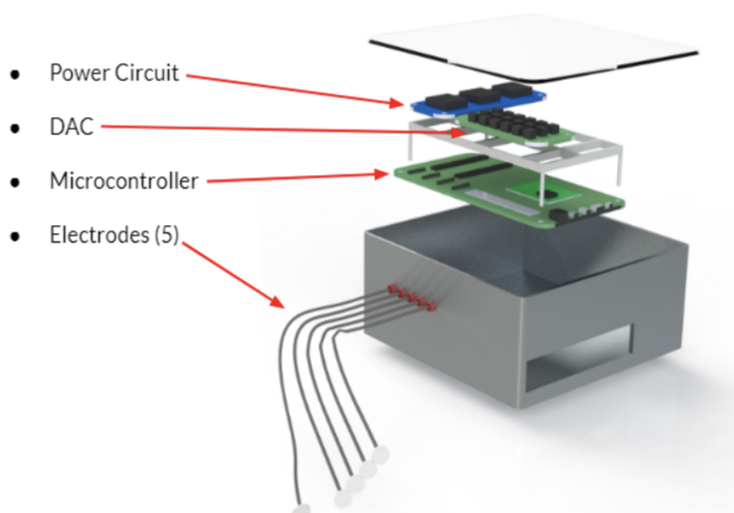


Fig. 6 GVS Final Design

Only two electrodes are required to provide the sensation of roll with GVS. After discussing feasibility with our customer, it was decided that the team would attempt to provide a system capable of being used for multi-axial cueing, which would require 5 electrodes. Multi-axial cueing is an area of active research, and there are competing models available, so our team aimed to provide a system with flexible electrode arrangements and a system that would provide explicit current control for each electrode. This required a system which could control not only how much current was being sourced by a cathode, but also how much was being sunk by each anode. To provide for cathode control, the team selected an LTC2662 current-control DAC. This 5-channel, 16-bit DAC varies voltage independently to control current across an arbitrary load, up to a maximum externally supplied voltage. To control the anode current sinking, we identified a type of IC called a current mirror. These are two matched transistors which, at the system level, have an "input" line and an "output" line. The current of the input line is the maximum current on the output line. With a "full Wilson" configuration, the error in the transfer function between input and output currents becomes very small. Since our system already includes a DAC capable of specifying 5 currents directly, we decided to tie each DAC channel to a specific electrode. Using relays, the current could either be sent out to the electrode, turning that electrode into a cathode, or send into the input line of that electrode's current mirror configuration, allowing that electrode to only sink up to the specified amount of current. This design therefore allows the user to explicitly specify both positive and negative current values for up to 5 electrodes. Since different experimental models require different electrodes, our team selected an off-the-shelf self-adhesive electrode with a special current-distribution system to improve consistency. Using electrodes that can be placed anywhere in a system where the source and sink current values can be explicitly specified, our system should provide improved control over current paths for subjects receiving cueing. The system circuit was verified using both an LTspice simulation as well as breadboarding prior to assembly.

3. CPS

The requirements that had the most pronounced effect on our CPS design choices were the lag time requirement (DR 3.3), the synchronisation lag time (DR 3.2) and the safety requirements (DR 2.4) and (DR 2.2).

The lag time requirements both between the TCS and GVS (DR 3.2), and between the joystick and eventual cues (DR 3.4) necessitated that we minimize the CPS computational time. This was one of the determining factors in the selection of C++ as CPS language. C++ is well known in the software development world for its superior speed, and the teams research confirmed that it would have the fastest computation time of all languages that were familiar to team members **source**. In addition to inherently quick computation, C++ offers several architectural advantages that would allow the team to develop efficient code. In particular, the ability to use multi-threading was an important factor in the decision to use C++. Multi-threaded code can run several processes at once, and the use of C++ pointers allows for parallel processes to access and alter the same set of variables in memory. The team knew this would be invaluable when developing code that needed to communicate with many subsystems simultaneously. While C++ does have a few minor disadvantages, mainly that it is difficult to create a native C++ GUI and that bad memory management can cause lag and other problems, it was decided that its advantages were significant enough to justify choosing this language for the CPS.

The TCS take much longer to execute a commanded cue than the GVS system, which means that in order to meet the 100 ms synchronization requirement the GVS execution must be delayed. This is accomplished by having the TCS communicate its state to the CPS, so that the CPS can determine when to command the GVS. It was decided to use the data from the TCS load cells, which were already needed in order to create the appropriate feedback control for the actuators, to create commands to the GVS. Load cell data from each side of the TCS is read into the CPS, which then filters and processes the data to get a single angular value to send to the GVS. Thanks to multi-threading, the system is able to send data to the GVS at the same time as it commands the TCS. This keeps the synchronization delay to well under the 100ms requirement, as it is limited only by the delays inherent to the load cell data collection, which was found to be negligible.

The team also needed to incorporate TCS load cell feedback into the CPS for safety reasons. In order to make sure that the for applied to the test subject would not exceed the NUMBER limit (DR 2.4), the team needed the CPS to be able to monitor the force being applied. The CPS constantly polls the actuators at a rate of once per millisecond (1000Hz) and, upon detecting a load cell reading above the maximum force value, immediately returns the actuators and GVS to a zero state and terminates the test. While this should never happen due to a large number of safety checks proceeding the load cell feedback gathering (Fig. 7, it does add an extra layer of caution to the CHAIR system.

Due to the inherent requirement (based on our levels of success) that the test subject be able to give some command input to the system, the team needed to be able to take in input from a joystick or other controller. It was decided that a joystick would be the most intuitive choice, as it is most similar to real aircraft controls. Additionally, most aircraft piloting games such as Microsoft Flight Simulator use a joystick, and therefore its use within an aircraft simulator would likely be more intuitive to test subjects than other controls. The team chose a Logitech Extreme 3D Pro joystick for a combination of reasons, including its low price, 3-dimensional cuing abilities, and the presence of several buttons that could be utilized by the CPS code. To integrate the joystick with the CPS, Microsoft's DirectInput library was used. While this design choice did stop the code from being compatible with multiple operating systems, it was the clear choice for the CPS team as it allows for easy access to joystick drivers without having to beg the joystick manufacturing company to provide them.

Since safety factors are named in two of the CPS driving requirements (DR2.4 and DR2.2) and, of course, the CHAIR team has the utmost concern for the safety of the system's users, safety played a major roll in CPS design. It was decided that safety checks should be performed at every possible stage in order to minimize risk to test subjects. The CPS safety process begins with cueing inputs and ends with termination options provided to both the test operator and test subject. A visualization of this can be seen by following the orange arrows in Fig. 7. In modes 1 and 2, the CPS individually checks inputs from the test operator cuing profile. If any of the inputs are found to be outside of the safe cueing range of -15 to 15 degrees, the inputs are rejected and the operator is prompted to input a different cueing profile. Similarly, the joystick commands are limited to ± 15 degrees, where -15 degrees is commanded by pushing the joystick all the way to the left and 15 degrees by pushing it all the way to the right. If the test is in mode 2, the CPS will process the joystick data and operator's cuing profile and combine the two angle commands. It then performs another check to make sure this new angle is also safe. Next, the inputs are processed for sending to the GVS and TCS. This involves changing the angle to either a force value or a current for the TCS and GVS respectively. The CPS will then once again check these values to make sure they are within the safe ranges, before finally sending the commands to the microcontrollers for the two subsystems. The microcontrollers once again perform a check to make sure that the values are within expected ranges prior to commanding the subsystems. As this is occurring, the CPS is also monitoring outputs from the TCS load cells. If an unsafe force value is detected, it will automatically trigger the CPS terminate sequence. Lastly, both the test operator and test subject are provided with a method to terminate the test at any time, as per DR2.4. For the test operator, this is the space bar on the computer keyboard. For the test subject, this is the trigger on the joystick. Either of these buttons can be used to return the TCS and GVS to their zero states (zero current for the GVS and zero force for the CPS) before terminating the CPS code so that no further angles will be sent.

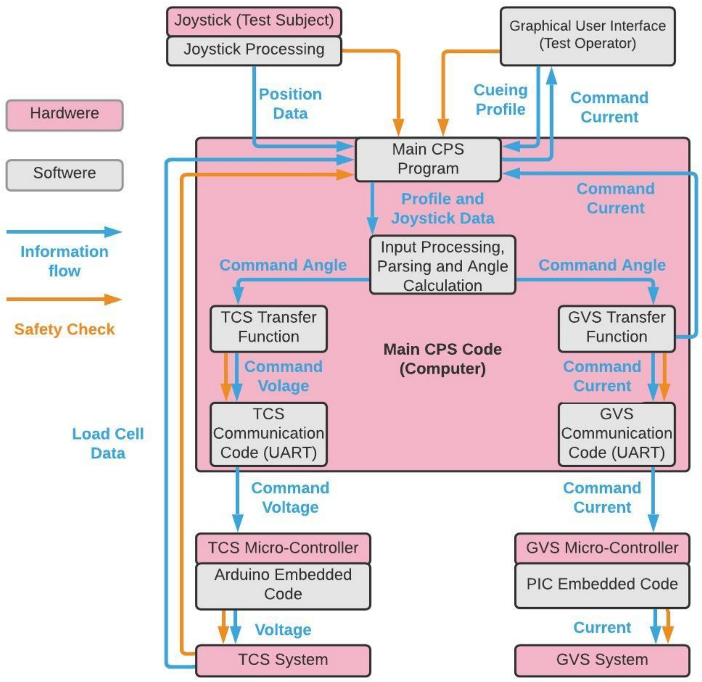


Fig. 7 CPS Final Design Flowchart

IV. Manufacturing

Authors: Cody Bahan, Sarah Foley, Laney Franklin, Carter Jackson, & Dean Widhalm

A. TCS

1. Structures

When designing the TCS structure, the team chose to custom manufacture every part to the anthropometric dimensions of the customer. Although this seemed like a daunting task, the team designed the TCS structure so that most pieces could be cut from flat sheet steel, and assembled with various brackets and fasteners. The first step in the process was to create detailed CAD drawings of each of these flat profiles that were to be cut, including the seat pan, back, sides, headrest plates and joystick table, to be submitted to the machine shop as work orders. After creating the detailed drawings and delivering the correct raw materials to the machine shop, the work orders were submitted. The team chose not to include any fastener holes in these initial work orders so that they could place them as they assembled the components to account for any inaccuracies in the dimensions.

One initial issue that the team ran into was that out of the procured raw materials, the seat pan and back could not be cut in one piece. To mitigate this issue without spending a lot more money on new materials, the team worked with the machine shop to weld two pieces together for the two components; however, this introduced an additional complexity to the design, as during welding both pieces acquired a non-negligible warp close to the center of each piece. The team again worked with the machine shop to straighten these out as much as possible, but the finished product still included a significant warp, making it so that the weld between the seat back and seat pan could not cover the full length of the seam. This stage in manufacturing is shown in Figure 8 below.

Fig. 8 Structure Pre-Upholstery

The team determined that once the side panels were bracketed on, the connection between the seat back and seat pan would be of sufficient strength. Next the legs, spine and joystick table members were cut from 80/20 and attached to the chair, completing the main structure of the chair portion. next, the mechanisms housings were to be constructed. To accomplish this, the team also chose to cut out a flat piece of 1/32" steel and bend it such that it created a three walled box with tabs at one end to attach to the side panels. The team then riveted on end pieces to close the outside end of the boxes and planned to attach hinges with doors to the backsides to fully enclose the electronics. Here the team ran into another issue with the doors as it was difficult to get the required precision when tapping the hinge fastener holes to get the doors to sit level and in place with the back of the mechanism housing. At this point, the team decided on to add doors and to leave the back of the mechanisms housings open to access electronics, which ended up being particularly helpful during mechanism housing assembly.

Next, the team decided to sand and paint all of the components manufactured so far to cover up the grind and heat marks from the welds and to make the structure look more sleek and uniform. After the coat of paint, it was time for upholstery. The seat cushions were made of 1" foam upholstered with faux leather material, making sure to include rectangular cutouts on the side panels for the pressure modules. The harness was also attached at the five points, fitted for comfort of the test subject. Also, the team added a base plate to the legs to increase stability. The pressure modules themselves were then constructed by cutting six 2.5" x 4" x 1" blocks of wood and upholstering them with the same faux leather material and 1/2" foam. The load cells were attached to the back of the pressure modules using two wood screws, and a dremel was used to bore out two holes in the back of the pressure modules in which the linear guides would be laid. At this point, this is what the chair structure looked like as shown in Figure 9 below.

Fig. 9 Structure Pre-Mechanisms Housings

The next step in the manufacturing process was to assemble the mechanisms housing including all electronics. A particularly difficult point of the process was gluing the load cell button to the actuator head, as this was a very small area of contact and the team was initially using Gorilla Glue, which needs to be clamped and set for 24 hours, as shown in Figure 10 below.

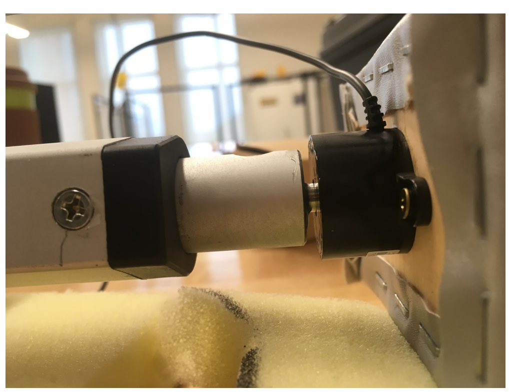


Fig. 10 Actuator-Load Cell Connection Point

The team spent a lot of time making sure to allow each actuator-load cell to have the full curing time, carefully clamped. Next, the linear guides were glued into the recesses in the back of the pressure modules, and the wooden spacers for the actuator to sit on top of and the linear guides to attach to were sized and cutout, custom sized to each actuator-load cell- mechanism housing group. Next, custom brackets were created to secure the actuators to the mechanisms housing but cutting out strips of aluminum, bending them over the actuator to get the exact shape, and cutting holes for the fasteners to attach them. After all of the mechanisms housing were assembled, they were fastened to the side panels. The initial mechanisms housing product is shown in Figure 11 below.

Fig. 11 Mechanisms Housing Assembly

After the preliminary chair was manufactured and assembled, the following dimensions were measured and accommodation capabilities noted, as shown in Table 12 below.

Dimension	Measurement
Pad to pad side width (hip width)	18.2”
Seat pan back to front (thigh length)	19”
Floor to top of seat pan cushion (heel to knee)	18.5”
Seat pan to top of seat back (butt to shoulder)	24.5”
Headrest height accommodation (butt to head)	22.5” - 44.2”
Headrest width accommodation (head width)	2.8” - 7.7”

Fig. 12 TCS Final Dimensions and Test Subject Accommodations

2. Electronics

In order to power the TCS system, the team had to design and manufacture a printed circuit board to drive all six actuators. The board needed to to satisfy all our design requirements, including providing feedback control to all six actuators while being able to run continuously for 30 minutes at full power. Thus, careful board design was needed to prevent overheating while being able to sustain a maximum current load of 3 A per actuator.

As discussed, the entire system is controlled through an Arduino Mega 2560 microcontroller that is attached to the PCB. The Mega 2560 had all the necessary pins required to drive six linear actuators and six load cells while communicating with the CPS within the proper time delay requirement.

The backbone of the electronic system is a set of six DRV8840 DC motor drivers that allow for the force control of the actuators. These motor drivers are fed a pulse-wave modulation (PWM) signal in which the duty cycle of the signal is manipulated in order to provide varying levels of force output. From our design, we utilize a direct relationship between the duty cycle of the input 12V PWM signal and the output force, where a 15°cue corresponds to 2.7 A from the actuator

The DRV8840 motor drivers also include individual fault pins that notify the user when individual actuators are experiencing an overheating or overcurrent problem. In addition to our software protections, each motor driver can limit the amount of output current driving each actuator through the use of five current scaling pins.

The feedback control system works from six individual load cells powered from the Arduino which read force data from each actuator through the analog pins on the Arduino. This system allowed the team to develop a controls model which constantly reads the force output from these load cells and initiate a controlled response to either increase or decrease the commanded force depending on the input from the user. This will be further discussed later.

For the full scale PCB design, the main considerations were making sure that the components were as efficiently laid out as possible while ensuring that component datasheet requirements were satisfied. Through this, we settled on a 4-layer shield PCB design. Thus, all the components were mounted on top of the board, and we used header pins to allow the Arduino to be plugged in from the underside of the board, essentially creating an Arduino Shield. For adequate signal integrity, all necessary capacitors were placed as close as possible to the motor drivers, and each surface mount motor driver was efficiently placed to ensure the shortest possible track lengths to the microcontroller. Additionally, one internal board layer included a power plane to supply power to the board efficiently. The remainder of the layers acted as ground planes for additional efficiency and signal integrity. The final board schematic can be seen below:

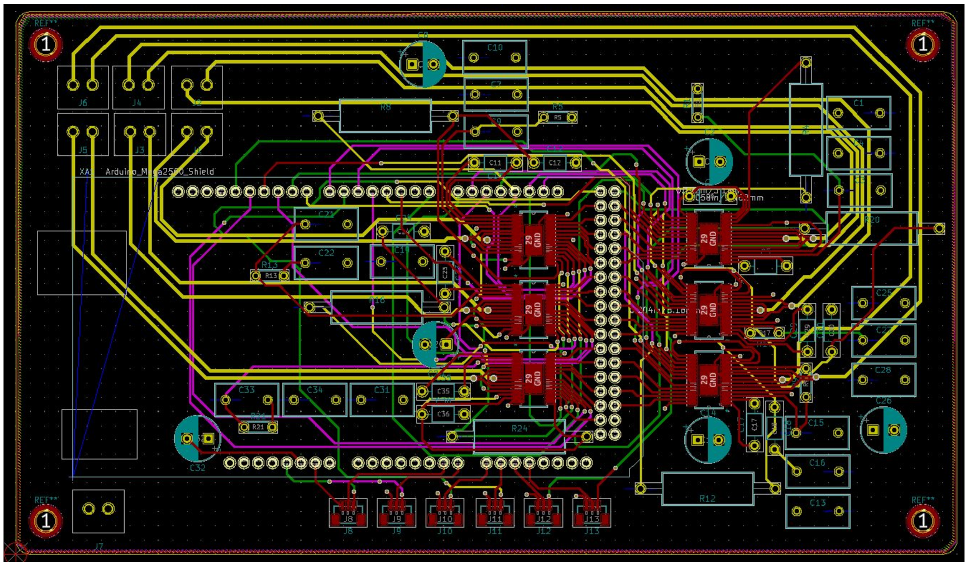


Fig. 13 Full PCB Schematic

Molex Mighty Spox connectors had the necessary current rating to power the board as well as connect all six linear actuators. The team manufactured the connectors with 22 AWG wire and was able to properly trim the wire lengths and house the loose cables in wire sheathing behind CHAIR. The six load cells were initially connected with Molex Pico Clasp connectors that allowed us to use the 30 AWG wire on the load cells. However, the surface mount connectors proved incredibly fragile, and the team had to switch to six green screw connectors which sit on the top of the board for easy removal.

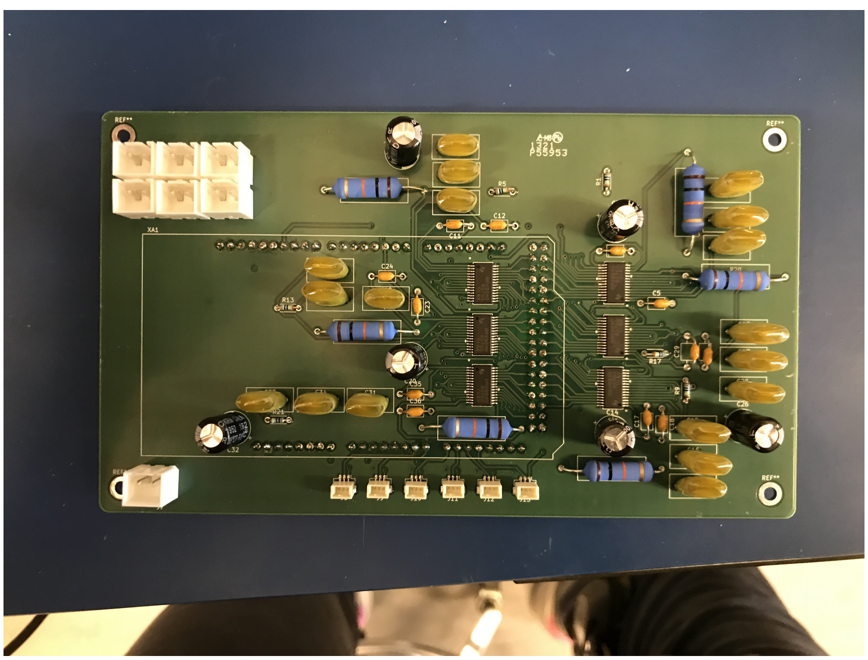


Fig. 14 Full PCB Board

Since the circuit board design was essentially an Arduino shield, the circuit board was attached to the Arduino, and the Arduino was screwed into a 3D printed housing which was attached with Velcro on the back of the CHAIR. The six actuator cables were managed with cable wrap and fed through a hole in the side of the housing to attach to the board, as with the load cells.

B. GVS

1. Structures

Holding the GVS electronics in a housing unit was an essential part of the GVS subsystem. This was needed to not only consolidate all of the different components but also as a means to easily transport and protect the valuables. We first began with creating CAD models for the housing, truss and face-plate. This was needed to ensure each component would have the clearance required to pass the necessary wires. After finalizing the size and layout of each electronic component, as well as needed power supply and banana plug holes, the final rendering of the GVS housing unit was complete. These detailed drawings were then 3D-printed and assembled. The lid, made of clear acrylic, was laser cut. There was only one issue constructing the GVS housing: the hole for the 24V power supply did not account for the nut. This was fixed by filing the truss down a few millimeters so that the nut could be screwed onto the power port. The acrylic top plate, the truss, and the PIC were directly mounted to the housing by drilling and inserting press-fit threaded inserts.

2. Electronics

The DAC was an off-the-shelf product, but still required some manipulation to be integrated. Each of the 5 channels (DAC0 through DAC4) needed to have a soldered connection to the power circuit. The cable connector also had to be removed so that 5V power could be brought in from the PIC to power the IOVCC elements of the DAC. The DAC was fixed to the truss using thread-cutting screws.

The power circuit was designed in KiCAD and sent to Advanced Circuits for fabrication. After the major connections on the board were verified using an ohmmeter, the ICs were mounted to the board with the reflow oven. The circuit was then verified using an ohmmeter and using 3.3V from a power supply to toggle the relays. After this, the power circuit was mounted using more thread-cutting screws.

To connect all of the elements together, pre-crimped JST wires were purchased. The wires from each housing were custom cut to specific lengths to make the cable management more appealing. For all turret connections, the wires were pre-tinned, wrapped around the turret with about 220-270 degrees of rotation, and soldered by following the recommended technique by PACE, inc.

C. CPS

While the CPS was not manufactured in the same way as the hardware systems on CHAIR, the organization of the code and tools used to build it were vital to the success of this critical project element. Various tools were used to help the team collaborate, break up coding tasks, assist with debugging, and perform complex tasks within the CPS.

One of the most critical tools used by the team was GitHub. As in many professional settings, the team chose to use GitHub as a major tool for building and sharing code. Since the CPS system would consist of many code files that would be built by different team members, it was important to ensure that code could be shared and integrated easily. This was even more vital due to Covid restrictions that made in-person meetings undesirable. To make use of GitHub's version control and integration features, the team chose to make a branch for each team member to contribute their code. Once individual code elements were written and debugged, they could be merged with the main code without significant effort from the team. Additionally, GitHub proved to be incredibly valuable during full system integration. As subsystems were integrated one by one, the team was able to make new branches to contain each step of integration, for example one branch for the integrated GVS and CPS codes, and another when the joystick, GVS, and CPS were all integrated together. The code on these branches could then be shared among team members during the debugging process, and eventually pushed to the main branch when debugging and testing efforts were complete.

In addition to GitHub, the team utilized an IDE for building and debugging code, as well as maintaining organization and for creating executable files. The team decided to all use Visual Studios for all C++ coding, as it offers a large suite of built-in libraries and debugging options, as well as Linkers, text coloring, and other support expected of a good IDE. Visual Studios was particularly helpful for the CPS because it allows for multi-threaded debugging configurations, which allow the user to look at the processes happening in individual threads. Combined with the ability to step through code line-by-line, this was incredibly important during CPS testing as it allowed the team to quickly identify and resolve errors without obscene numbers of std::cout statements. The team also hopes that the Visual Studios project structuring will help future users to easily navigate and edit the code.

In addition to organization tools, several C++ libraries and tools were utilized to help perform some of the more complex CPS tasks. For example, Microsoft's DirectInput library was heavily utilized in order to communicate with the joystick. This software allowed the team to use built-in functions to detect and poll the joystick instead of having to obtain joystick drivers, identify the appropriate USB port, and painstakingly write C and C++ scripts for communications. While the team did still write their own functions for joystick setup, X-axis polling, trigger press detection, and joystick detachment, DirectInput provided several supporting functions. Additionally, it allowed the code to be written such that it should be compatible with multiple joystick brands and types, which is certainly not something the team would have the experience to do alone. Along with DirectInput, the team utilized a variety of online open source code libraries as resources when developing much of the UART and joystick code. Among these were a valuable tutorial on how to use DirectInput and several online code repositories containing information on serial communications in C++. By reading through and understanding code that others had written, the team was able to develop their own solutions for the CPS's unique needs, while eliminating some of the legwork that would have been involved in developing the code from scratch. The team was also able to utilize online forums such as Stack Overflow during the debugging process. When encountering unknown errors, it was possible to search these websites and find other programmers who had encountered similar errors as well as various potential solutions to those errors. This proved to be an invaluable tool during the debugging process, particularly when encountering uncommon errors such as Link errors.

While the various tools used for CPS development were no doubt vital for its manufacturing process, it would be neglectful not to also mention the highly effective modular approach utilized by the team. Rather than have multiple team members working on the same tasks and files, each software engineer took responsibility for a single element of the final product. One team member wrote the GVS communication code, two took charge of the joystick, one wrote the TCS communication code, and one wrote the main CPS code. Creating the code for each component separately allowed for rapid parallel development, which meant that the CPS team could easily maintain a strict timeline. Once each component was individually developed, GitHub could be used to easily integrate elements of code. Additionally, it was found that this method was highly effective when working with construct that were unfamiliar to some team members. Rather than all of the team members trying to learn new material at once, one team member could learn it then explain to the rest of the team, saving time.

V. Verification & Validation

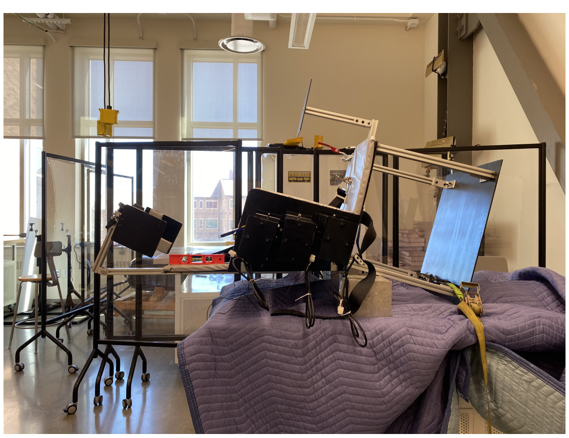
Authors: Cody Bahan, Sarah Foley, Laney Franklin, Jason Magno, Baily Rice, & Aiden Wilson

A. Subsystem Vefification and Validation

1. TCS

1.1 Structural Tests

The team conducted a seat back structural test to validate the strength of the weld between the seat back and the seat pan. To perform this test, the chair is secured to a table so that the seat back is suspended and parallel to the floor. A measurement of the initial distance between the table and the seat back is taken while the chair is in its unloaded position. Then the seat back is loaded up to 90 pounds (three times the expected force on the seat back) in 5 pound increments. During the loading process, the weight is placed 17 inches (two-thirds of the seat height) from the weld. For each loading case, the change in height of the seat back above the table is measured at 17 inches from the weld. Figure 15 shows the seat back structural test setup with the chair unloaded and maximally loaded.



(a) Unloaded

(b) Max Load

Fig. 15 Seat Back Structural Test

The results of the seat back structural test are verified against a model that the team constructed to predict the deflection of the of the seat back given an applied load. To predict the deflection, the seat back is modeled as a simple cantilever beam with the material properties of stainless steel. A comparison of the measured and predicted deflections is given in figure 16. As seen in the graph, the measured deflection is slightly larger than the predicted model. The major source of error seen in the graph is attributed to weakness of the straps that secured the chair to the table. This weakness allowed the legs of the chair to come off of the table, which caused the entire chair to pivot around the block that was supporting the chair. Consequently, some of the measured deflection includes movement of the whole chair instead of pure seat back deflection. At max load, the bottom of the chair was measured to lift off the table by about one-half of an inch. By incorporating this known error, the results show that the actual deflection closely follows the predicted deflection, thereby verifying the strength of the weld and passing the seat back structural test.

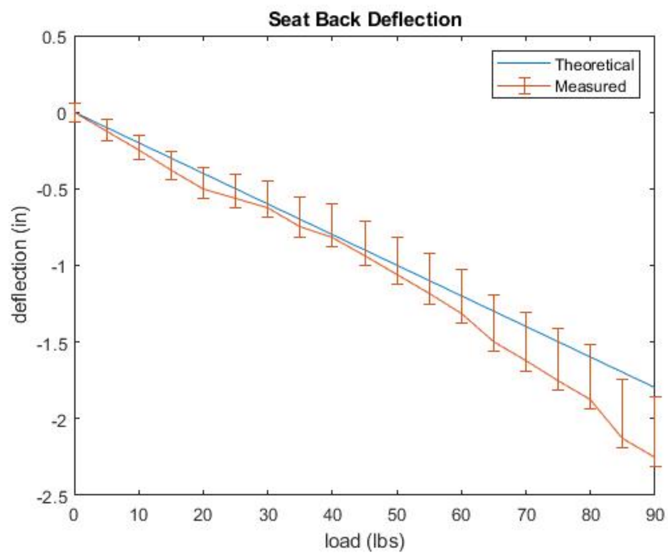


Fig. 16 Comparison of predicted versus measured deflection

1.2 Center of Gravity Test

To validate the stability of the TCS structure, the team found a furniture industry standard test to ensure a chair has a safe center of gravity. This test is conducted by loading the chair with 160 pounds and pulling backwards on the seat back until the chair begins to tip. The chair must be able to remain level while being pulling backwards with at least 35 pounds to be considered safe. The team set up the test by loading the chair with sandbags, and hooked a strain gauge to the spine as shown in Figure 17 below.

Fig. 17 CG Test Setup

Initially, the chair could only be pulled backwards with 15 pounds of force before it tipped. The team mitigated this issue by placing a 63 pound steel block obtained from the machine shop on the front edge of the base plate. After retesting, the chair was able to be pulled backwards with 45 pounds of force before tipping, thereby passing the CG test, as shown in Figure 18 below.

Fig. 18 CG Test 2 Results

1.3 Control Tests

The TCS control scheme was run with an embedded Arduino code. It utilized a PD control. The proportional gain tuned until the response time of the actuators was deemed acceptable. The derivative gain was then set to update itself on every iteration of the loop. It would take the recorded load cell force and subtract the commanded force. This gives a value for the difference between what we command and what we get out. The derivative gain is then tuned based on the magnitude of that value. This allows for a full feedback loop to be cratered; using the load cells as our primary driver. First, the TCS electronics team created a test PCB. It was a scaled down version of the full PCB that ran a single actuator. The first test done was a step response. Single actuator tests were conducted with the pressure pad making perfect contact with a flat and solid surface. This ensure the output is the most ideal circumstances for the actuator; allowing for a look into the control logic itself. First, fig **??**, shows us a step response.

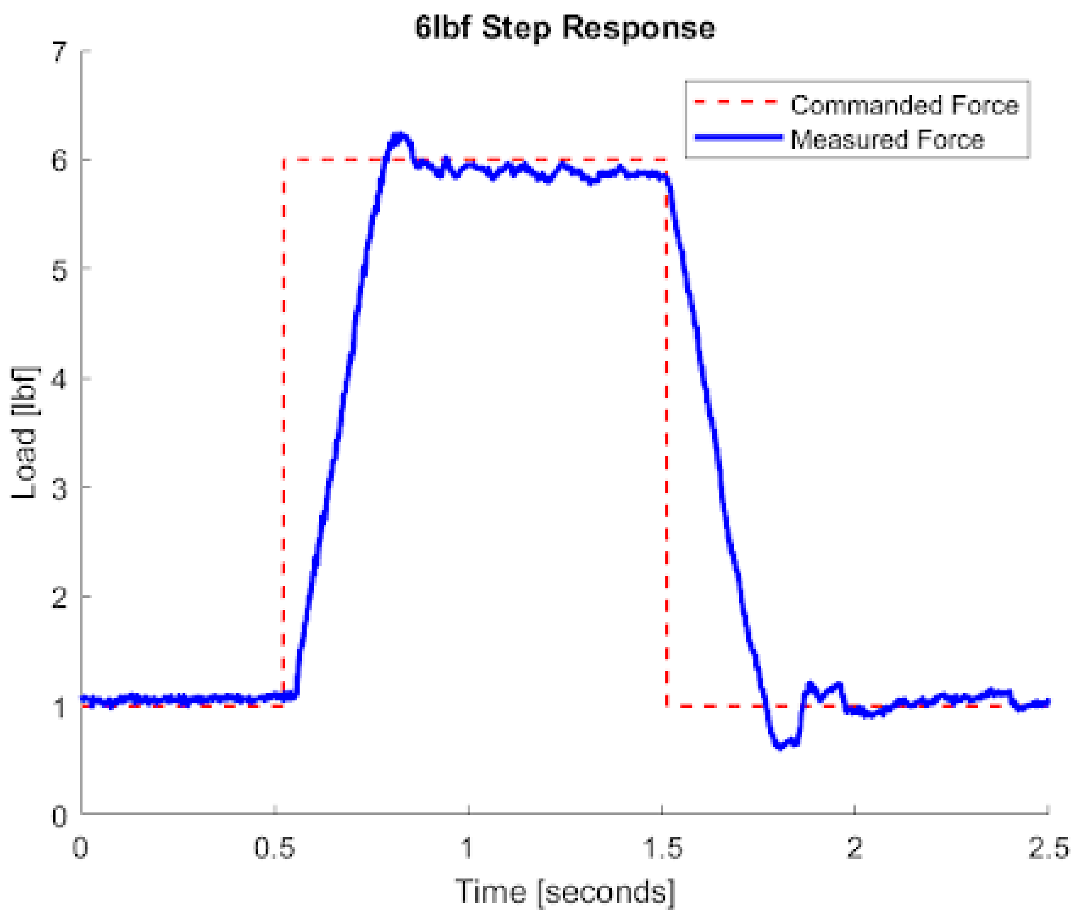


Fig. 19 Test PCB Step Response

As can be seen, the response time of the actuator in this setup is about 0.5s. There is minimal overshoot and then the response settles very quickly. The 0.5s response time gives us plenty of margin to maintain under a 200ms total lag time. Real operation of these actuators will not be step responses however. Instead the force will follow some profile. Rolling is a sinusoidal response and therefore the best way to test the system is to track a sin wave. Figure 20 shows this.

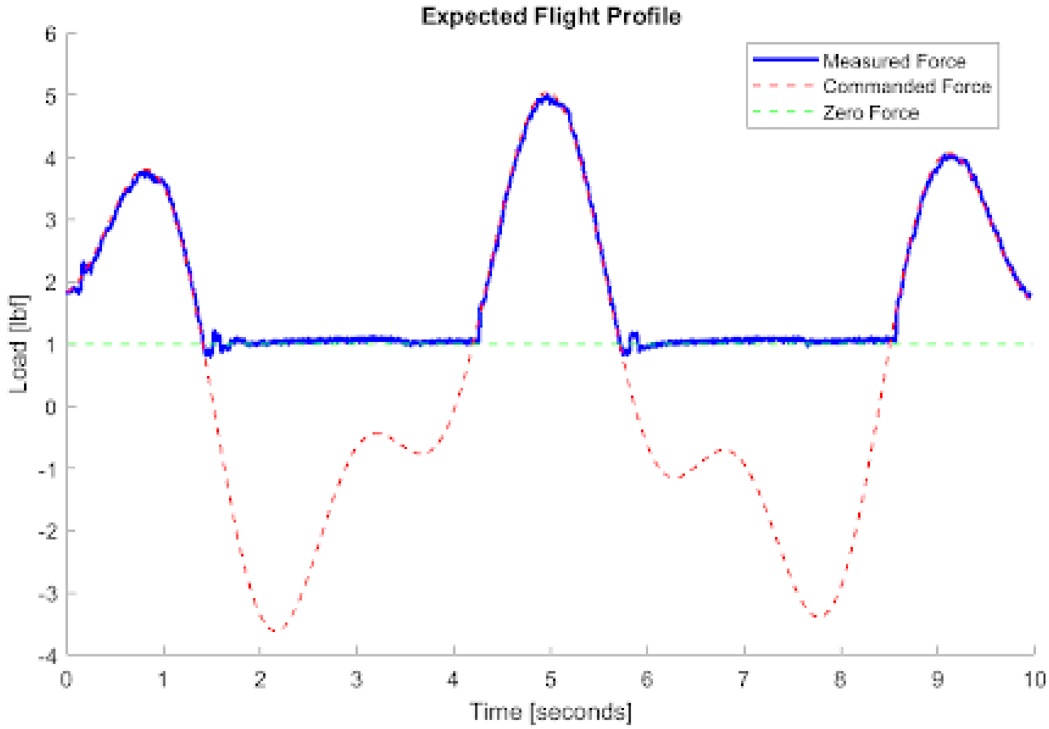


Fig. 20 Test PCB Sin Response

When tracking a sin wave, the response will only track the upper half of the curve. This indicates a positive roll. When the profile is under the minimum force required to zero the TCS pressure pads, the output force is the zero force set. The best way to quantify our tracking is by calculating a root mean square error (RMSE) of the output versus the commanded. RMSE is defined in equation 1.

$$RMSE = \sqrt{\Sigma \frac{(Commanded - Measured)^2}{n}} \tag{1}$$

Using this equation, the RMSE of fig 20 is $0.078lbf$. Next, the TCS was tested with all six actuators integrated into the chair; pressing on a test dummy. However, the hardware of the TCS failed during this testing and no data was able to be captured. The hardware was partially repaired and a test using four of six actuators was accomplished. Data was extracted for the purposes of comparing it to our single actuator data, but the TCS still requires more work before a full analyse of all six actuators can be recorded. Figure 21, shows the response of the middle left actuator tracking a sin wave.

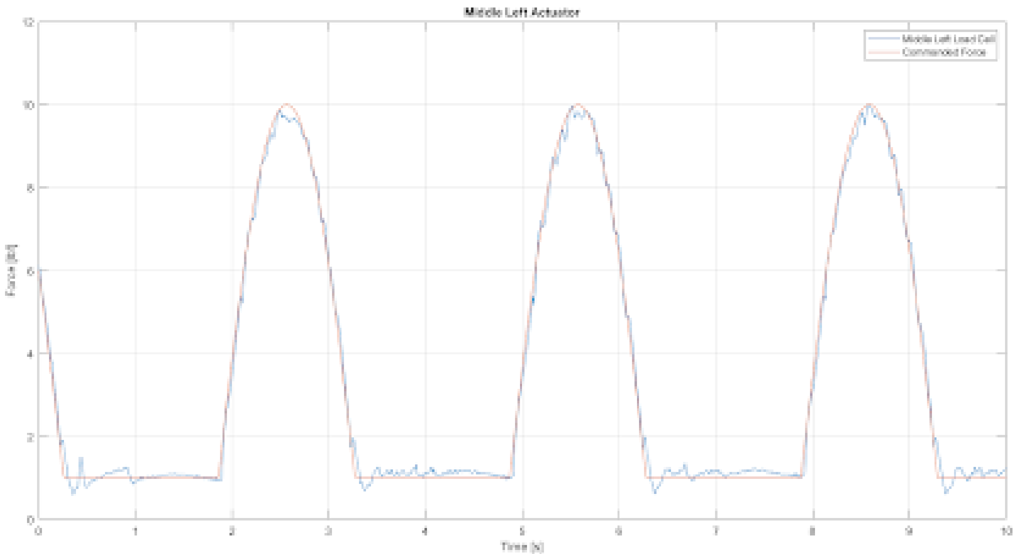


Fig. 21   Four Actuators Middle Left Response

Comparing this data to that of the test PCB sin wave shows a good response for this actuator. The RMSE calculated for this actuator was found determined over a very long time period that has inconsistent data. Thus, the RMSE values from this round of tests was not accurate. Yet, we can compare the tracking of these actuators working on the dummy to the response on a wall. Figure 21, has very similar tracking to the single actuator test. This shows that our controls can be usable on a human shaped object. Unfortunately, this behavior was not consistent on all the control responses. The top left actuator had unstable response when the force transitioned from the sin curve to the zeroing force. The fact that this only happens with certain actuators points to the problem being native to the actuator housing unit rather than the control scheme.

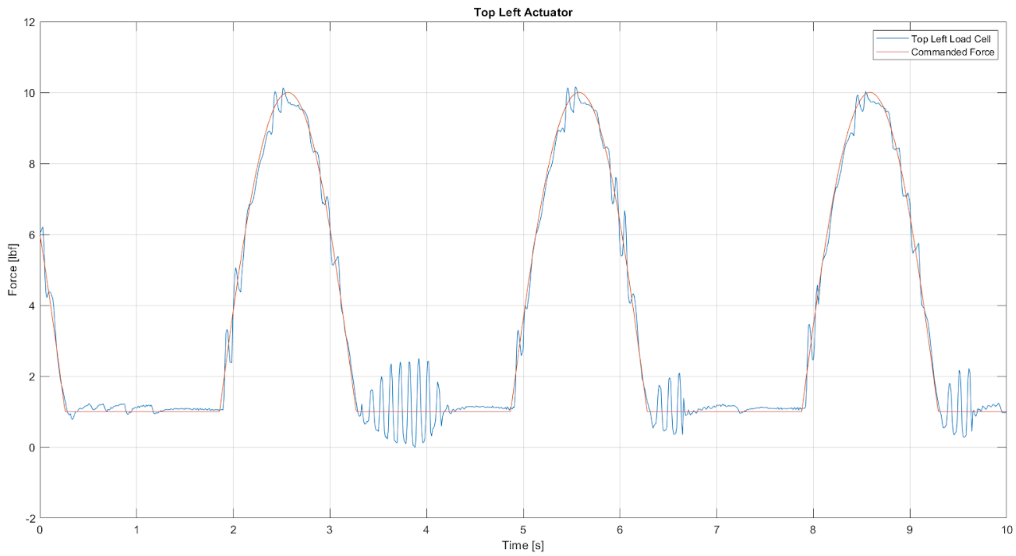


Fig. 22 Four Actuators Top Left Response

A leading theory for why this was occurring is that the load cells may have been damaged during the first hardware failure. We know that one load cell did break, and it is likely others were damaged. The damage could cause for shaky force reading. Which in turn would effect the controls scheme. A few weeks after this test occurred, the top left load cell did eventually stop working entirely. This could indicate that the load cell was damaged during the four actuator testing.

The TCS was able to verify its structural and accommodation requirements but was unable to verify the full operation of applying a tactile force.

2. GVS

The team conducted various tests to verify the functionality of the GVS. The first test was a full scale range test. We verified the GVS could command currents accurately to ± 4mA. As shown below in Figure 23, this graph represent the part of the test going from zero to positive 4mA. It is important to note that the variation in time scales for the step function is not a resolution issue but rather we commanded them in those step sizes over inconsistent time scales. This was the case for many reasons we decided to step up in smaller increments due to the fact that we wanted to limit the extent of dizziness and tingle that the subject may feel from the GVS. It should also be pointed out that we did redo this test after twenty minutes to verify precise performance after continuous operation.

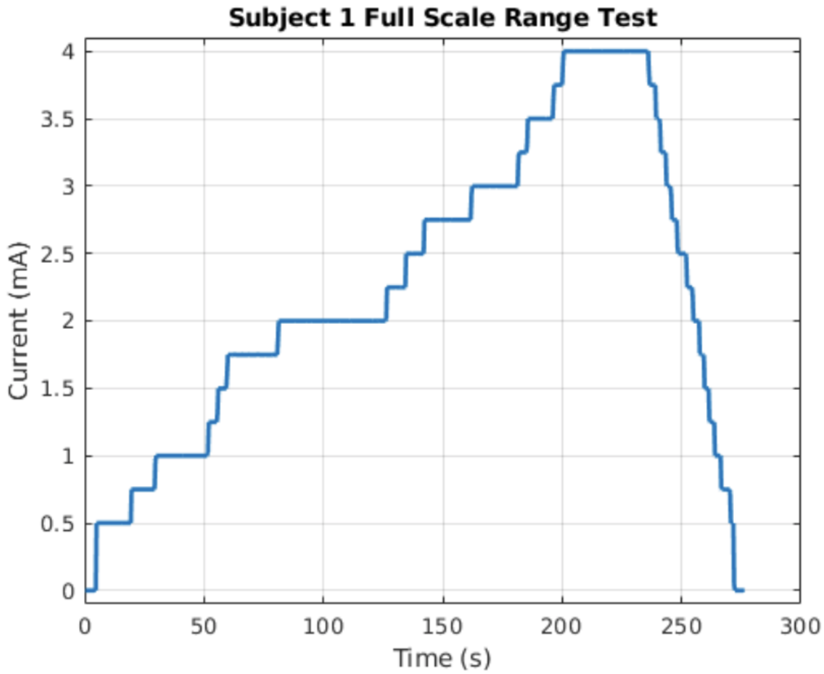


Fig. 23

The next test conducted was a drop out test. We wanted to verify with this test, the efficiency of the overall system. Our model predicted the drop out to be .625mA but we actually measured, from our results, less then .01mA. The value is definitely smaller than .01mA, but we limited the GVS to have a smallest step size of .01mA. The visual results of this test can been seen in Figure 24.

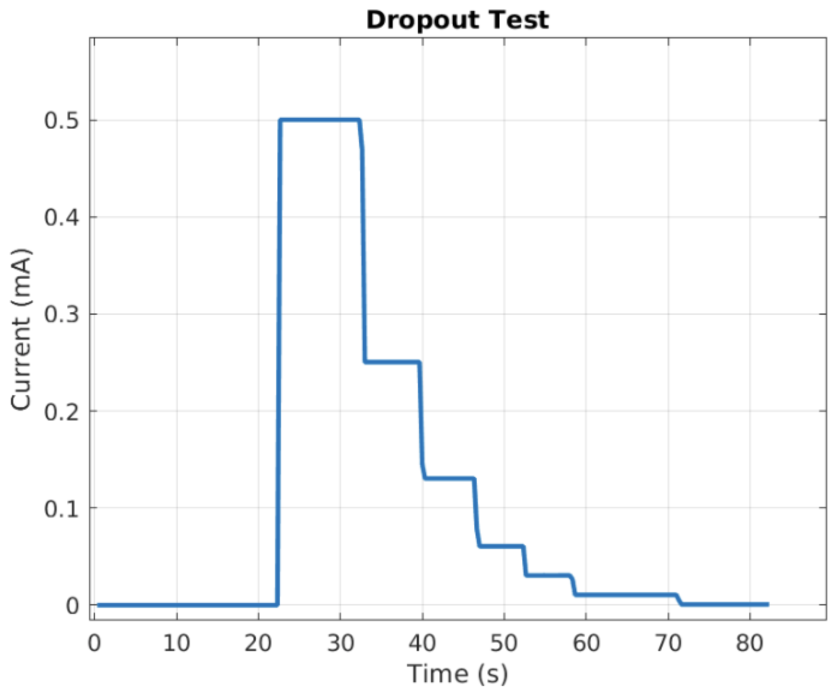


Fig. 24

In our final test, the team wanted to evaluate the error predicted by the GVS and the measured results from our tests. Below, in Figure 25 it can be seen that the two trends line up almost identically. All of our values for gain , offset and standard deviation were much lower than the values found in our ideal situation. Visually, the gain represents the difference in slopes between the actual and expected values while the offset error is the y-intercept of the graph. We predicted 93.8 and 6.25 μA for Gain and offset error, but actually recorded a much smaller 1.3 and 1.48 μA for those error values respectively. Our Standard Deviation was predicted to be roughly 50 μA but we measured to be 29.5nA. Its imporant to note that the values we were comparing our measured results from were taken from a "worst case scenario" model which is why the differences are so vast.

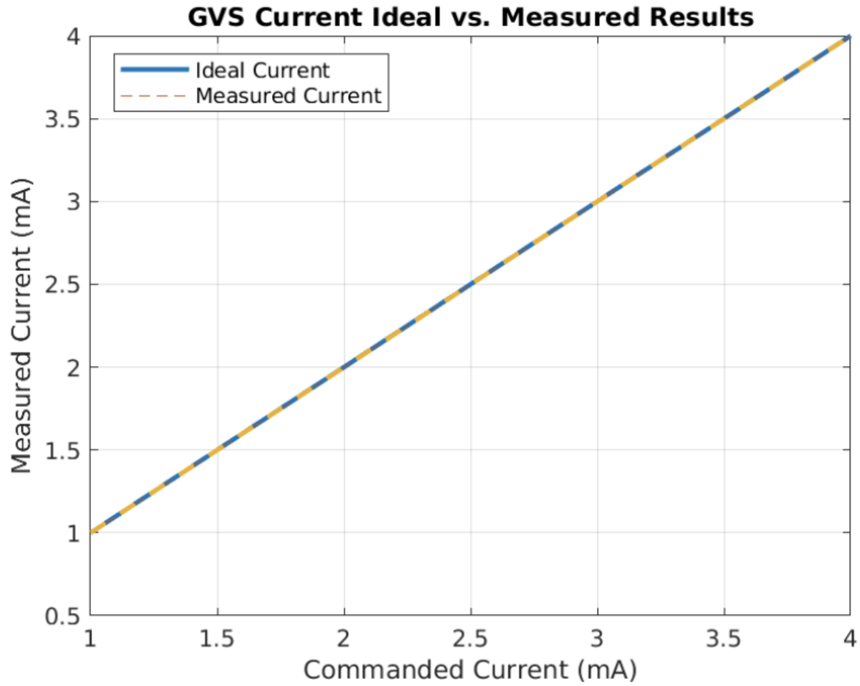


Fig. 25

3. CPS

To validate the functionality of the CPS, the team took a modular approach, choosing to verify individual functions and code components, then slowly integrate these into a functional whole. The combined code was then tested again at each integration step in order to confirm that the files were merged successfully. Each test of the CPS consisted of running the system in each of the three testing modes. For mode one, a set of predetermined cuing profiles were run. These profiles were designed to encompass each level of success as well as check safety systems and memory use. These profiles were created using a Matlab script which was shared with the team on Github in order to ensure consistent testing practices. Additionally, a special profile with only small angle commands was generated for mode 2 testing in order to avoid safety checks initiating termination. For mode 3 tests, the system was run with only joystick commands for a total of two minutes.

Validation began with the main CPS code, which consisted of the CLI, input processing, safety checks, and dummy functions to imitate receiving joystick data and sending commands to the TCS and GVS. Testing of this system started with testing the cuing profile input. A large number of input files were checked, including very large files, and files with several types of errors, such as incorrect formatting or non-linear time data. In each case, it was verified that the CPS either processed the data into the desired format, or rejected the input and prompted the user to try again. After reading in data, the data processing routine was run, and the team manually checked that the processed vectors contained the expected values. While some small bugs were discovered during this phase of testing, there were no major issues and all problems were quickly resolved. The next step was to verify that the output files were being written correctly. These tests were very simple, and only required the running of a cuing profile and the checking of this profile against data in the output file. The same test was run in mode 2 to capture both simulated joystick data and cuing data and confirm that all three sets of values were captured. Once inputs and outputs were verified, dummy functions were added to simulate sending data to the TCS and GVS. These functions were able to process data into the format required by the two hardware systems as well as call skeleton functions for sending data to the microcontrollers. This step of verification was particularly important for testing out the multi-threading structure within the code. Once again, several minor issues were identified during this phase of testing, in particular some memory access issues associated with multi-threading capabilities. However, these issues were quickly resolved. The final step in validating the main CPS code was to verify stop button capabilities. This involved integrating a stop function with the main CPS, which would listen for a space bar press and terminate all functions if one was detected. This proved to be the most complicated element of the CPS to debug, as it was found that several threads needed to be reorganized so that button presses could be detected. The stop button function had to be moved to the main thread, while other functions were relocated to side threads. Once this was achieved, however, the main CPS was round to be working as expected in all three modes of operation.

Concurrent with main CPS verification, the team worked to verify that the joystick processing code was properly receiving data. This was simple to do: it required only that the user move the joystick or pull the trigger and then verify that the outputs to the joystick communications function matched the input. The team then integrated the joystick processing with the main code and verified that main CPS registered the data and would not exceed safety limits for either mode 2 or mode 3. The system was also run without the joystick in mode 1 to ensure that the code would still work without joystick input.

The next task was to verify the GVS communication code was working. This was done by sending predetermined commands to the PIC micro-controller and verifying the correct output from the DAC. Once this was done, the GVS code could be integrated with the main CPS. This was first done with no joystick inputs, simply sending cuing profiles to the GVS from the CPS and using a digital multimeter to check that the current from the GVS matched the angles being sent. Once this was confirmed, a stop button test was performed. This turned out to be vitally important, as it was found that while the stop button terminated communications with the GVS, it did not send a zero angle first. The issue was rapidly resolved, and it was found that the stop button was able to return the GVS to a zero state within 10ms of its being pressed.

With both the GVS and joystick codes working, integration of the two hardware systems could be performed. After combining the code files, the GVS and joystick were both plugged into the laptop running the CPS. The CPS was then run first in mode 3 and then in mode 2. The output file generated by the CPS (which contains commanded angles) was then compared to current data from the GVS. It was found that the joystick commands matched almost perfectly to the values read out from the GVS, confirming successful integration. The joystick stop trigger was also tested, and it was found that the trigger press quickly returned the GVS to a zero current state. While no official time was recorded, we estimate that this occurred within 20ms based on the GVS-CPS communications speeds and data rate. An image of the integrated Joystick-GVS test can be seen in Fig. 26. It should be noted that the GVS current data is plotted against data points, and therefore was not overlaid with the joystick command data in order to avoid confusion. As this figure

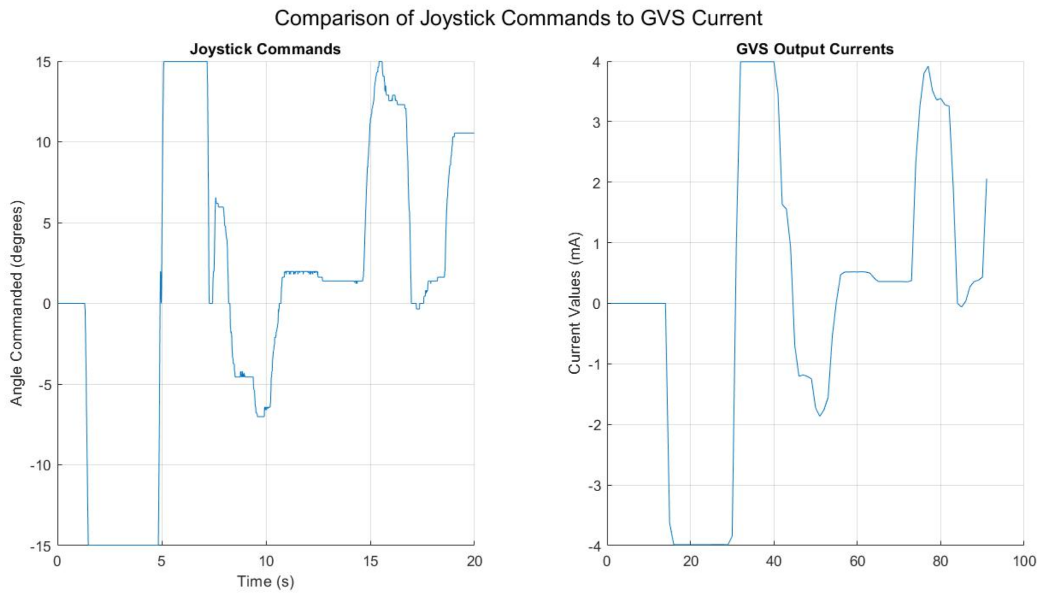


Fig. 26 Commanded angles from the joystick compared to recorded GVS current values.

demonstrates, the GVS was able to successfully command a current profile matching closely to the joystick inputs. The GVS data sampling rate was significantly lower than the joystick data sampling rate, which explains the small differences between the two profiles.

While hardware issues made it impossible to fully test the TCS, the TCS communication code was still integrated with the CPS. The integrated code contains TCS setup, as well as commands to read and write data from the TCS Arduino. Once the communication code was integrated and building, it was run connected to the actual TCS Arduino. The team was able to confirm a serial handshake, as well as verify that commands were being sent appropriately to the Arduino. Load cell feedback was also recorded, and while it was not possible to confirm exact values, it is believed that the load cell feedback was being read correctly. A test was also performed where a team member pushed on the load cells with more force than allowed by safety requirements, and the CPS successfully determined that the force was unsafe and terminated the test in this instance. However, more tests are needed to truly confirm TCS integration and functionality.

While it was not possible to do full integration testing, a fully integrated version of the code was developed with all three hardware systems combined. This code is building successfully, but will still need to be run and tested once the TCS is working.

In addition to functional testing, some basic performance checks were done on the CPS. Using the Visual Studios debugger, it was found that the integrated GVS-joystick code was able to run repeatedly without encountering errors. Additionally, the code used minimal memory– the maximum recorded value was around 3MB, which was less than one percent of the test controller's CPU capacity. The code was also run on three different laptops, and each time the tests were successful. While there were no specific requirements stating the need for portability or low memory use, the team is pleased with this result as it indicates the code's versatility as well as low probability of unexpected application crashes.

B. Full System Verification and Validation

Unfortunately, due to the issues with TCS integration caused by hardware problems, the full system verification and validation was not completed. However, there were still plans put in place for the full system testing. Most of the full system functionality does not change from CPS-GVS and CPS-TCS integration. Therefore, the full system tests are mostly focused on repeating the subsystem integration tests already discussed in the previous verification and validation subsections except with all three subsystems integrated together. Repeating these tests would allow verification and validation of all of the requirements already verified and validated but ensuring that running two cueing systems simultaneously does not have any ill effects on the system capabilities.

One planned test that was different from the subsystems was the syncing/delay test. This test was meant to verify design requirements 3.2 and 3.3. These requirements specify that the time between input and cueing output must be less than 200 s and the time between the two cueing systems must be less than 100 ms. These requirements are meant to ensure that the test subject will not receive any disorienting or confusing cueing due to the subsystems being out of sync. The syncing between the two subsystems can be determined by sending a known command to the two cueing systems, such as a sinusoid, and simultaneously recording the force and current outputs for the TCS and GVS respectively. Comparing the time stamps of key measurements, such as sinusoid peaks, will provide the timing difference between the two cueing methods. The time from joystick input to CPS output can be determined through the recorded data within the CPS. The time that the TCS needs to responds once the actuators have made contact is about while the time needed for the GVS to respond is typically less than 1 ms. These low response times provide a promising outlook for the total delay time even though the full system test could not be completed.

Validation of the full system performance was designed for the three different levels of success. For level one success, static tilt cueing, a test subject would be provided with several successive static cues to either their left or right. The subject would be asked to use the joystick to indicate which direction they were perceiving tilt. The joystick input would not change the tilt profile as it would only be used to record the subject response. The joystick data would provide insight into the perceived direction cueing but would not provide any data regarding magnitude of cues. Level two success, sinusoidal tilt cueing, would use a similar process of recording joystick data without using it as an input. For this level, the subject would be given a sinusoidal cue and asked to track the perceived tilt using the joystick. This test would provide data on perceived tilt as well as perceived magnitude. However, there is no universal or intuitive way to convert one's perceived tilt to joystick deflection. For example, one subject may perfectly match the sinusoid by deflecting the joystick as far as it will go at the peaks of the sinusoid while another subject may only deflect the joystick halfway at their perceived maximum. The data recorded would only show a relative magnitude for each test subject. The third level of success, variable tilt profile, can provide more consistent magnitude information across test subjects. The test subjects would be subjected to a tilt profile that is not known to them. This profile could be a combination of sinusoids, constant tilts, linear changes or any other tilt combinations. For this test, the joystick control would be enabled and contribute to the tilt profile. The subject would be asked to null out any perceived tilt so that they would perceive zero tilt and remain "upright" while the unknown profile was running. This would provide more accurate magnitude information that could be compared across test subjects since there would be feedback that would inform the test subject of their performance.

For the first two of these validation tests, success would be measured by the accuracy of the joystick response. The static tilt test would be a percentage of tilts that were correctly identified. The sinusoidal tilt joystick data would be normalized to match the magnitude of the joystick sinusoids with the cued sinusoids. Then the mean squared error between the two would quantify the performance of the test subject. The variable tilt profile success would be quantified by recording the actual angle cued over the test. A perfectly nulled tilt profile would result in a zero degree tilt being cued for the duration of the test. The mean squared error between the actual cueing angles and zero would therefore quantify the nulling success. In addition to this quantitative data, it was also planned to record qualitative data throughout the tests. This would be gathered by asking the test subjects questions about their experience and perception.

VI. Risk Assessment & Mitigation

Authors: Rhyss Bass, Sarah Foley, Dean Widhalm, Carter Jackson, & Aiden Wilson

A. TCS

1. Primary Structural Failure

Failure of the primary structure of the chair (the legs, seat pan, and seat back) was considered to be the most severe risk that was present for the TCS. Failure here could likely result in the test subject falling to the ground along with many expensive components. Although this risk was considered to have severe consequences, it was given a likelihood rating of "very unlikely" during the initial risk assessment stage of the project. As the design of the TCS neared completion, the team became aware of a potential issue with the center of gravity of the TCS which could cause an increased likelihood for primary structural failure in the form of tipping over with a subject in the chair. To further assess this risk, the team performed a tipping test once assembly was complete but prior to human testing. The test initially failed, confirming the concerns of the team and the need for mitigation. The team mitigated the tipping concerns by placing a 65 pound block on the front of the TCS baseplate. This shifted the center of gravity forward significantly and the TCS passed the repeated tipping test.

A second concern came about after there was an issue during manufacturing which resulted in an incomplete weld between the seat back and seat pan. The imperfect weld meant that any structural calculations done during the design stage would no longer be accurate and tests would need to be done in order to ensure that the likelihood of primary structural failure had not increased. During the test, the back of the chair was loaded with three times the expected force and the deflection was measured as force was applied. The measured deflection was sufficiently close to the expected deflection of a cantilevered beam and it was determined that no further mitigation was necessary.

2. Tertiary Structural Failure

Another risk that the team was weary about since before manufacture was a tertiary structural failure, or failure of the mechanisms housing assembly. This was originally assessed to be possible and a major failure if it occurred. At the time, this was mostly due to uncertainty in the strength of the connection between the load cell button and actuator head as the contact area was so small. The team felt that with the use of strong glue and given proper time to cure, that the connection could be strong enough for operations.

Upon assembling the mechanisms housings, it again became a worry for the team as, once fully glued and set, the connection seemed to be rather flimsy, with the pressure module being able to practically rotate freely about the load cell button. At this point, the team wasn't sure if the linear guides would provide enough stability to the pressure modules, so it was decided to assemble the entire mechanism housing assembly and evaluate from there.

When the entire assembly was finished, the team found that not only the connection between the load cell and actuator was flimsy, but also the connection between the linear guides and the mechanisms housing. When enough force was applied to the pressure pad by the actuator, either the linear guides would tear off of the wooden spacers between it and the mechanisms housing, or the head of the actuator would tear off the load cell. At this point, it was determined that the glue the team as using to secure these components was not strong enough for this application, and also its requirement to set for 24 hours while being clamped was making assembly and reassembly extremely difficult. Thus, the team researched the strongest quick setting super glue and found that Loctite Super Glue works for all materials in use in the project. Also, the team decided to recut the spacers for between the linear guides and mechanisms housing to be one longer piece rather than two separate smaller pieces, to increase contact area with the mechanism housing and make the glued connection stronger. The team also learned at this point that the load cells should not be used with any shearing force on them, and that through initial testing, one actually broke. The team decided from then on not to glue or attach the actuator head to the load cell at all, as during operations, this connection will only experience compression, and following operations, the pressure pads may be pushed back in. Thus, since this connection was particularly weak, bad for the load cells and not integral to the operations, the team removed the connection.

At this point, the team attempted more retests with the dummy in the chair, and the top two actuator assemblies on each side proved to be much stronger, but not perfect. The only issues that they sometimes had was that if the load was increased high enough to the maximum expected load of 19 pounds and run for long enough, since the load cell and actuator were no longer connected, the actuator head slips off the load cell button and hits the wood of the pressure pad. To mitigate this, the team attempted to more securely fasten the actuators to the mechanisms housing, and reinforce the linear guide connections. Although the team believes this method to be successful for the top two actuators for smaller applied loads, ultimately the forces required to cue up to 15 degrees are too large for the current mechanisms housing design, and the team experienced repeated failures with the bottom two pressure pads. In the future, the team believes that a different design omitting the linear guides and also potentially using a different type of load cell will work much better and be far more rigid.

For the bottom two actuator assemblies, the team was firstly experiencing electronics issues with one, which will be discussed separately, and significant strength issues with the other. These strength issues come from the shape of the test dummy at the hip area, in that body is highly curved. This causes a significant twist and off axis load on the pressure module as the actuator keeps pushing for it to make better contact, and the pressure module rotates to be flush with the dummy's hip. The team is not sure how to fully mitigate this, as they believe that in real human testing the pressure pad will make better contact than with the dummy, but it will not make perfect contact. As a temporary fix, the team attempted to reinforce the linear guides to oppose the twisting of the pressure module. With the issue of time, the team was not able to reinforce the top two sets of actuators and linear guides, and was only able to add more glue to the bottom two. As stated above, ultimately the team feels that the current mechanism housing design is insufficient for the magnitude of loads we are applying and the amount of off axis shearing forces that the linear guides are receiving.

3. Electronics

When all six actuators are running, the electronics system needs to be able to handle a maximum of 16.2 A of current, at 2.7 A per actuator. Overcurrent is the biggest risk with the TCS electronics system, as it can cause damage to the board and components and possibly harm to the user. To manage this issue, the code incorporates current limiters that set the current limit to approximately 3 A. The results of these current limiters can be viewed in the TCS Validation section. Additionally, the team included safety measures in the hardware as well, shunting the maximum current driven through each motor driver. These measures ensure that the current never extends beyond 3 A per actuator.

The second risk to the electronics system is overheating. To account for this, the team designed a test PCB initially to test the circuit for a single actuator to make sure that overheating did not take place. The design of the test PCB and the final PCB included wide tracks to be able to account for up to 6 A of current for the outputs from the motor driver. Additionally, the final PCB design included a power plane that helped prevent temperature spikes in the board. Between these two measures, the team noticed no noticeable overheating from the board during endurance tests lasting over 30 minutes.

B. GVS

The major risk that was identified for the GVS project was the inability to procure an LTC2662 Evaluation Board. These are made to order and, due to global conditions, it was not known if these could be manufactured in time. To mitigate this, we placed a purchase order for this device as soon as the design was approved following CDR. We also purchased two of them for redundancy: if we had damaged one of the chips, getting a replacement would have taken a prohibitive amount of time. This was a relatively small extra expense and ensured the success of our project.

Additionally, while the unsafe commands risk is primarily considered a CPS risk, our design was modified to include safety checks within the GVS system as well. When the GVS receives a command, it performs it's own error-checking. It checks to make sure that the first character is a '+' or a '-', then checks to make sure that the subsequent 3 characters are within 0x30 and 0x39, i.e. decimal numbers 0 and 9. Therefore, the safety checks for the GVS commands take place twice.

C. CPS

The CPS had only a small number of risks that could completely prevent the success of the chair system, but the CPS did have a ton of small issues/risks that could reduce the capability of the system. The catastrophic risks for the CPS were the risk that the system wouldn't be able to obtain joystick data and the risk of excessive lag time. The joystick risk was mitigated by getting the joystick processing done as early as possible. The lag time risk was mitigated by using C++ as the CPS language, the use of threading to reduce computational time and optimization of our main CPS code. We have tested our code and the computational time of the CPS system is much smaller than what we predicted. We have also been successful in obtaining joystick data.

Because the CPS is responsible for a lot of sub-functions of the chair system there are a large number of non-catastrophic risks. The first of these risks we identified was the possibility that the system would use excessive memory and processing power. To reduce this risk we used best practices related to dynamic memory allocation to prevent memory leaks and we chose not to use bulky add-ons, even if they would make programming the system easier. One risk that was identified was the risk that the system wouldn't be portable across Mac, Linux and windows. It would be helpful if the system could be run from any type of computer, but it is not necessary for the system to be usable. Our plan to mitigate this was to find cross-platform drivers for the joystick and use universal communication protocols. Unfortunately, we were unable to get the code to run on non-windows computers because we couldn't find any good cross-platform drivers for polling the joystick data and ultimately went with direct input, which is for Windows only.

There were also a number of risks and issues which we failed to predict. None of these were catastrophic, however the usability of the system was somewhat diminished. Despite selecting C++ as the CPS language partially because it was believed to be the language with which most team members were familiar, many team members were not as familiar with C++ as expected. Because of the lack of familiarity, a substantial amount of extra time had to be spent getting people familiar with C++. This also meant that most people learned how to work on one or two aspects of the CPS and it was difficult to get people familiar enough with other areas to check each other's work. Another issue that was overlooked was the problem of getting the computer to auto-detect which USB ports were being used for the joystick, GVS communication and TCS communication. We were unable to get port auto-detect to work and the user has to manually select which port is used for each device. The final issue that we fail to fully account for was that another systems issue would delay the full CPS testing for so long. We were aware that issues with the TCS and GVS could delay CPS testing, but we did not have a good plan for what to do if these delays lasted till the end of the semester. Eventually, what we had to do was test the CPS code with the reduced capability of the TCS. We did this a few days before the system was turned over to the customer and then we spent the next couple of days working out any issues/double-checking that the CPS was safe. We neglected to fully account for the amount of time required to get the CPS to the point we wanted and due to time issues we were unable to get our GUI working, although we do have A robust CLI that provides all necessary functionality. Ultimately there were no issues that reduced the fundamental capabilities of the CPS system, but there were a number of issues that reduced the ease of use of the chair system. These include having a CLI instead of a full GUI, no auto-detection of the COM ports used, and the limitation that the chair system must be run from a Windows computer.

VII. Project Planning

Authors: Cole Gray, & Baily Rice

A. Organizational Chart

To maximize the productivity of the team, each team member was also a lead of a specialized role. Figure 27 shows the organizational breakdown of the team; outlining each members lead position and how it fits into the whole team structure.

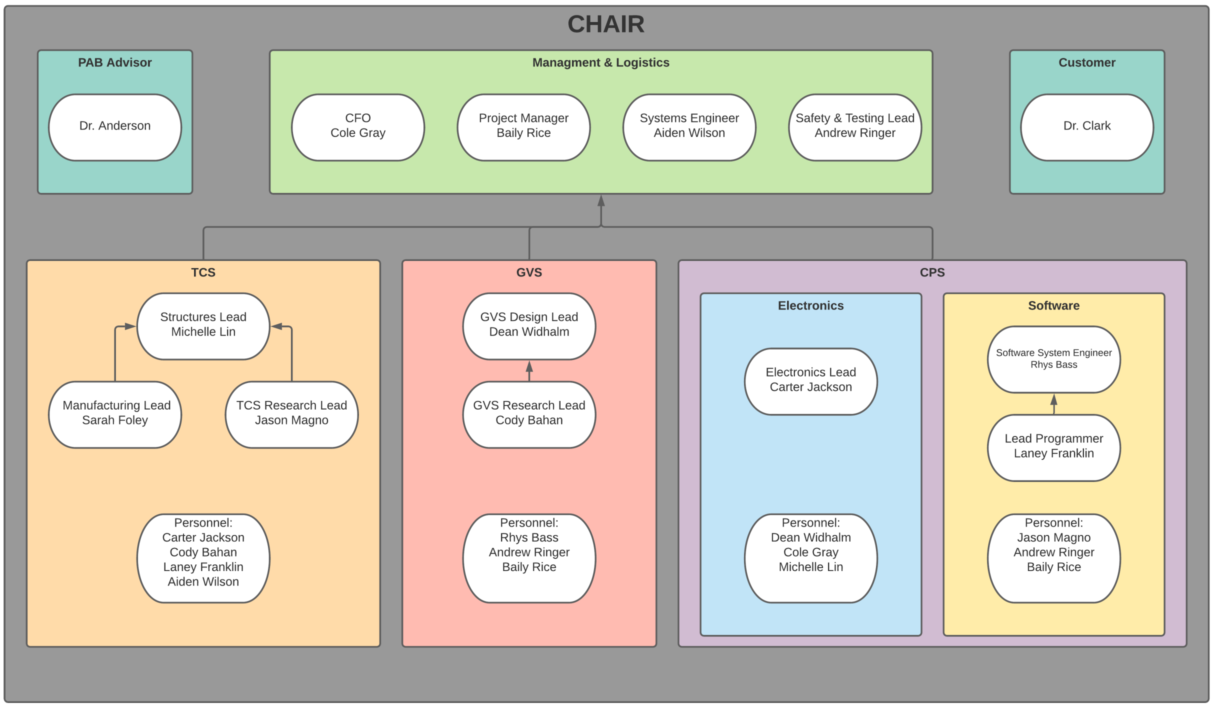


Fig. 27 Team Organizational Chart

In the top left corner of Fig 27, is the Professional Advisory Board advisor for team CHAIR, Dr. Allison Anderson. Just right of this block is the management and logistics block. These members are responsible the planning and coordination between the other subsystems. Tasks this block is responsible for includes budgeting, scheduling, risk analysis, outlining of testing procedure, and other tasks that would have impact on multiple subsystems. Lastly, in the top right is Dr. Torin Clark, the customer of this project. Moving to the bottom of Fig 27, the organization of the three subsystems can be found. At the furthest left is the TCS block. These team leads were responsible for the research, design, and manufacture of the TCS. Similarly, the GVS block contains the personal responsible for the research, design, and manufacture of the GVS circuit. The last subsystem block belongs to the CPS subsystem which is further split into electronics and software. The electronics lead was responsible for circuit simulations and manufacture. The software leads write the programming for all subsystems as well as planning for integration of all these separate functions. Lastly, in each subsystem block there is a personnel list that consist of the manpower that each lead may call upon.

B. Work Breakdown Structure

The work for this project was split into separate tracks for each subsystem of the team. Also, safety was handled separately, such that there was an unbiased analysis of each subsystem to ensure safe operations. Figure 28 can show the progress of each of these subsections and their workflow throughout the past year.

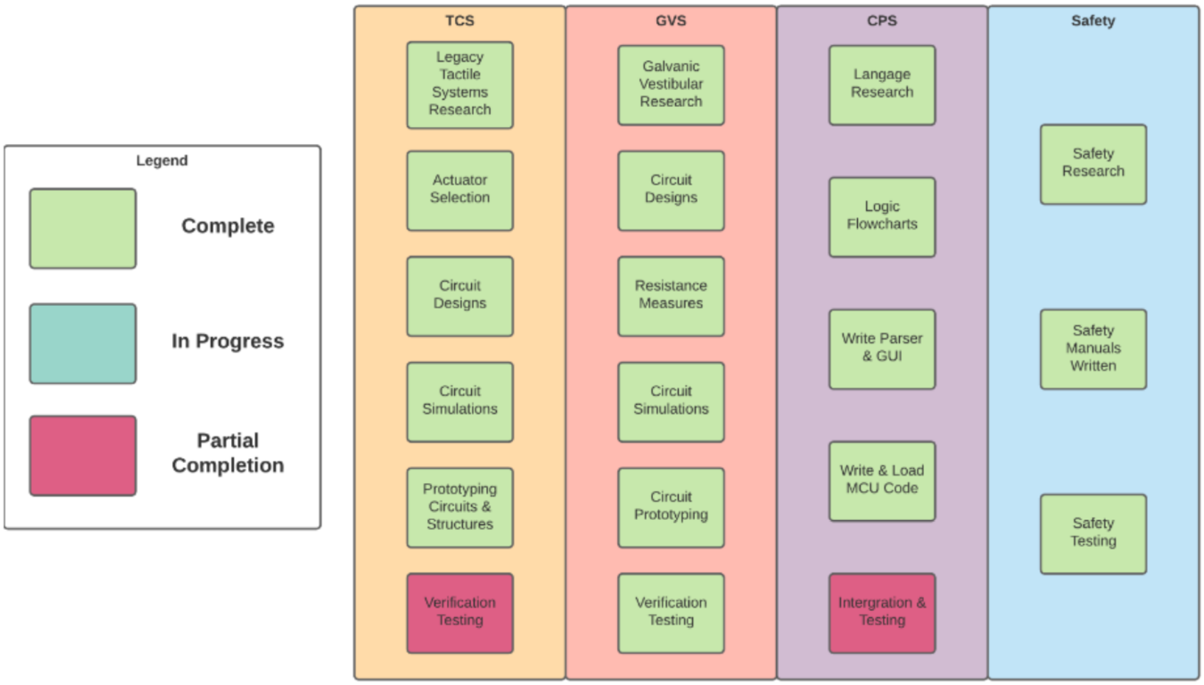


Fig. 28 Work Breakdown Structure (WBS)

1. TCS

The first column shows the work following the TCS subsystem. To begin, similar tactile chairs of the past where researched. This helped the team decide on using actuators to produce the pressure for cues. Actuators were then researched and compared to find the best fit for our design. Once the linear actuators were chosen the circuit design was computationally designed and simulated. After the simulations had been a success a prototype PCB was ordered to begin preliminary testing of the TCS. The prototype board was the same as the design drafted for the entire TCS except reduced in size to run just a single actuator. As the test PCB was being built and tested, the manufacturing team was also building and testing the primary and secondary structures for the TCS. After the basic circuit could be verified we moved on to the full size board. It was at this time the structures team also conducted load and center of gravity testing. However, during the first test of our TCS verification testing there was problems with both the PCB as well as the secondary structures, particularly at the pressure pad, load cell, and actuator head connection. These hardware issues meant the TCS was never fully verfied.

2. GVS

The GVS subsystem was able to finish all their main work tasks listed in the breakdown as well as the reach goal of 5-electrode capability! The year started with research of already exciting GVS systems. Next, the circuit design that the GVS uses was researched and human resistance testing was done to set a baseline. With some resistance values known, the team was able to run circuit simulations. This helped with choosing an appropriate power supply. Some prototyping of the circuit was then conducted on a breadboard to fully prove the GVS circuit design. Lastly, the GVS was able to finish all verification testing. Successfully running with the software; outputting the desired current.

3. CPS

The CPS encompasses any software needed to operate CHAIR. This includes the embedded codes on the PIC and Arduino micro-controllers. The year started with researching the language the team wanted to build the suite in. After C++ was decided, the first logic flowcharts were made; followed by pseudocode. This is followed by the formation of some of the primary functions; such as the main parser and user interface (including a prototype GUI). As the primary functions gain form, the secondary code can be written. This includes an auxiliary lines needed to run the micro-controllers or communication. That then leaves the verification testing. Like the TCS, the CPS was only ever able to reach partial completion. The software suite is complete and operable, but CHAIR was never able to get all of the CPS' functions fully integrated together. The GVS was the only subsystem that was able to be fully integrated and verified with the CPS. Since full integration was not achieved, some of the lag time and timing requirements set for the CPS were not able to be tested. For the WBS to be fully complete the TCS hardware issues must be solved and then the CPS full verification tests would have to be fully complete.

4. Safety

Three primary breakdowns of the Safety work that was conducted: Safety Research, Safety Manuals/Checklist and Safety Testing. The safety research focused around operational safety procedures with regards to software and hardware. Both the TCS and GVS presented possible safety risks, with the TCS being a force risk and the GVS being electrical. Where the CPS required redundant software safety check to safety control the two subsystems.For the TCS quantify the max force and governing that was the primary safety research for the TCS. OSHA standards of safety were researched for the GVS systems. Safety operations/manuals were integrated to the procedures of each of the subsystems. Tests were preformed through out the building of the CHAIR subsystems. These test included component, subsystem and integration testing for the level of integration that CHAIR achieved.

C. Work Plan

Throughout the entire year, the project was organized with a Gantt Chart. The Fall and Spring Semester had separate Gantt Charts. In both semesters, the Gantt Chart was broken into separate branches. Each branch would tackle specific categories of tasks.

1. Fall Semester

Figure 29, shows the color coding scheme used for the Fall Gannt Chart.

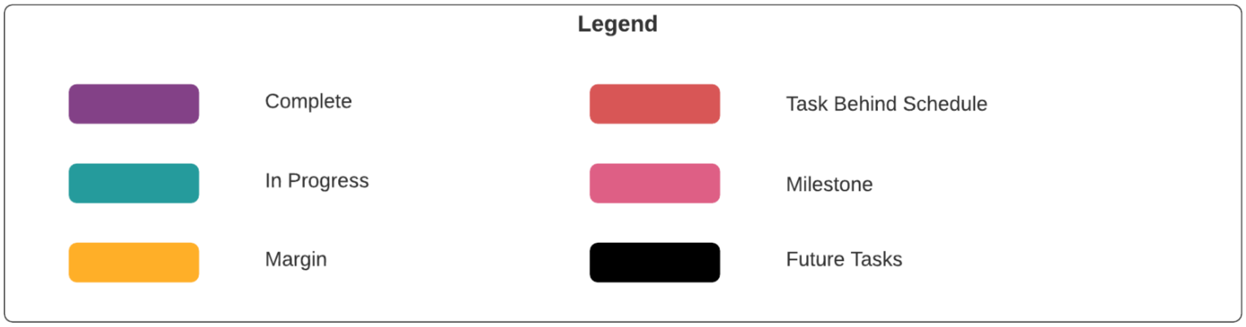


Fig. 29 Fall Gantt Chart Legend

Figure 30, shows all the work from the the Fall. There are separate branches for TCS structures, TCS electronics, GVS research, safety research, software, and finances.

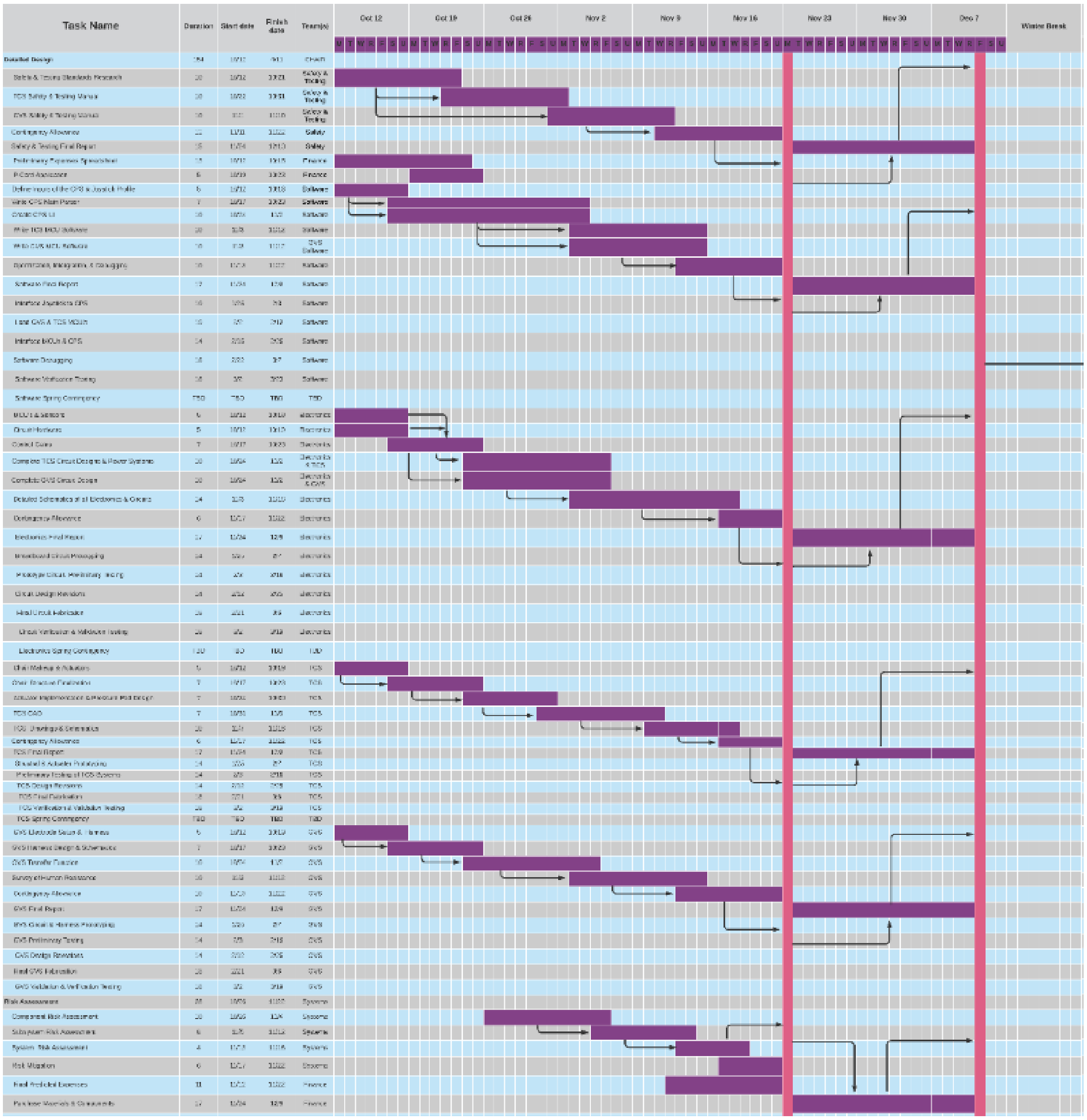


Fig. 30 Fall Gantt Chart

This Fall Gantt is actually just half of a Gantt Chart that shows the work for the entire year, but formatting a Gantt Chart like this is not ideal. When the chart has too many tasks, it becomes very hard to read and follow. The more branches you have the harder it is to follow the critical path. Using these lessons, a more streamlined Gannt Chart was made for the spring semester.

2. Spring Semester

Figure 31, shows the color coding scheme used for the Spring Gannt Chart.

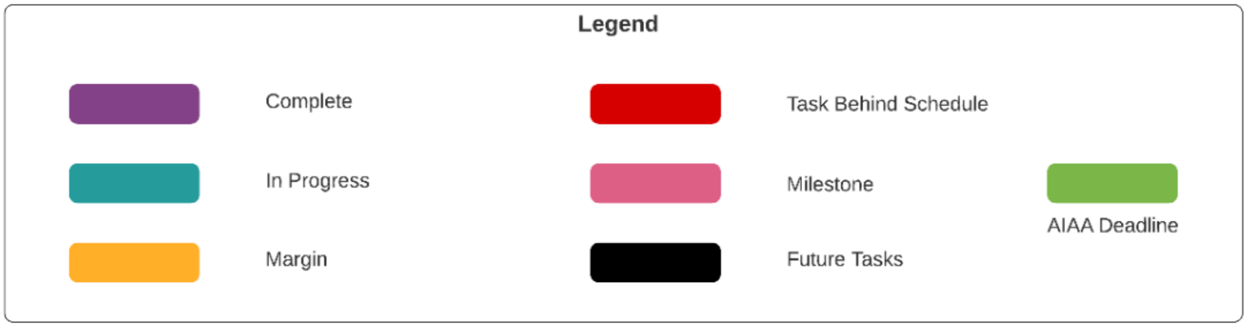


Fig. 31 Spring Gantt Chart Legend

The Spring Gannt was streamlined into a more manageable three branches. Furthermore, they all rejoined together at some point. Figure 32, shows this updated strcuture.

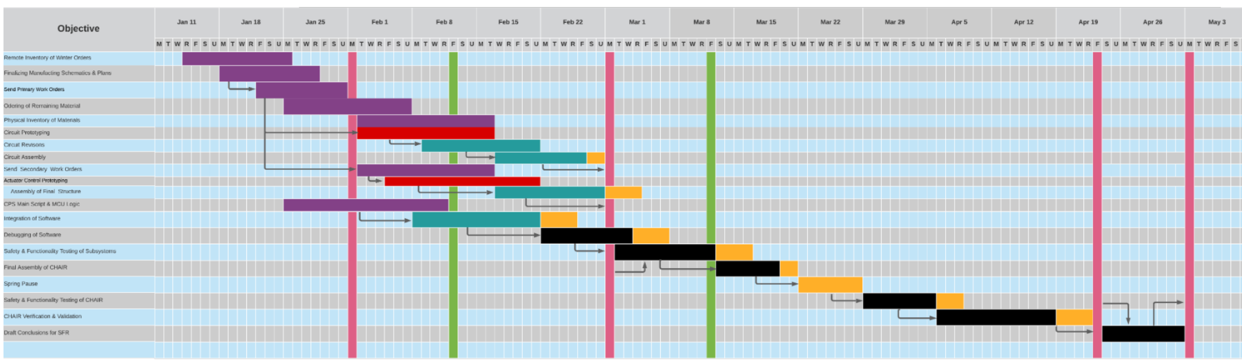


Fig. 32 Spring Gantt Chart

The three branches of the Spring follow each subsystem. About halfway through the spring, the three branches rejoin as a single path. This indicates when the team has reached the point required to progress to integration in each individual subsystem. Margins we also used more strategically in the Spring. The entire week of 'Spring Pause' was designated as a margin. That margin was to serve as extra time in case any issues arose during manufacturing and testing. That ended up being a great idea! Figure 33, shows that updated schedule.

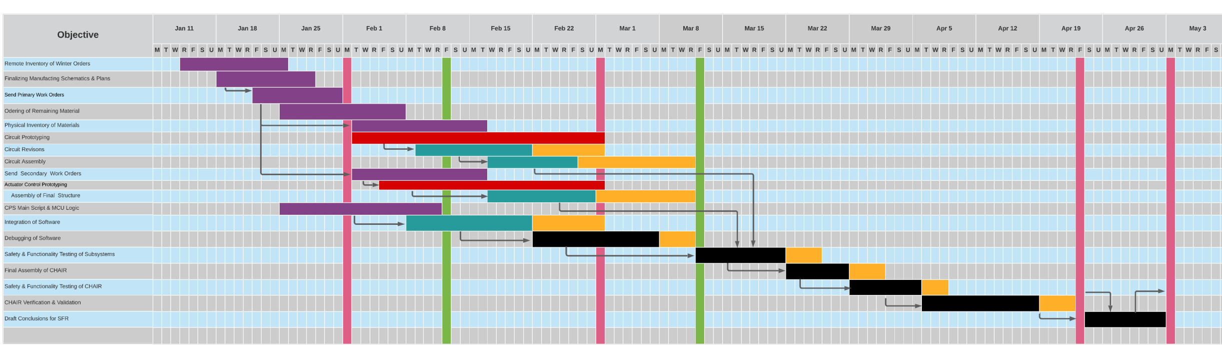


Fig. 33 Updated Spring Gantt Chart

During the single actuator testing of the TCS, the prototype PCB did not function as expected and the prototype PCB had to be reprinted. This pushed back both the TCS software and hardware by about a week. Thankfully, the margin that was built into the 'Spring Pause' was able to be reallocated. This gave us the extra time needed to get back on track!

D. Cost Plan

This section outlines the cost plan for the CHAIR project. Below in figure 34 shows the break down of budget for the CHAIR for each subsystem and the corresponding margin. Each subsystem was allotted a 9% margin, corresponding to a total budget of $4500 and $500 margin for the entire project. This is a slim margin, however it was felt necessary due to the scale of the project. To counteract this, care was taken to ensure that the cost estimations of each system were well researched and justified to avoid underestimating and going over budget. Looking at the right most column of figure 34 are the remaining budgets for each system. Both the GVS, CPS and Misc budgets came in under budget. Only the TCS went over. However, TCS went over budget as a result of broken load cells and the need to rush order new ones at the last minute, which cost $180 and put the subsystem over budget. Overall, the chair expenses came in at $4564.42, well under the $5000 cap.

Subsytem Breakdown				
Subsystem	Budget	Margin	Total Expenses	Remaining Budge
Galvanic Vestibular Stimulator (GVS)	$ 775.00	$ 86.11	$ 772.97	$ 2.03
Tactile Cueing System (TCS)	$ 2,850.00	$ 316.67	$ 3,000.63	$ (150.63
Central Processing System (CPS)	$ 650.00	$ 72.22	$ 570.32	$ 79.68
Misc	$ 225.00	$ 25.00	$ 220.50	$ 4.50
Total	**$ 4,500.00**	**$ 500.00**	**$ 4,564.42**	**$ (64.42**

Fig. 34 Budget Overview of CHAIR

Breaking down how each sub-system contributed to the CHAIR expenses results in the pie-chart shown by figure 35. The largest contributor to cost was the TCS system. The large cost of this subsystem comes from the expensive hardware (load cells, actuators and linear guides) as well as the cost of materials for the chair itself.

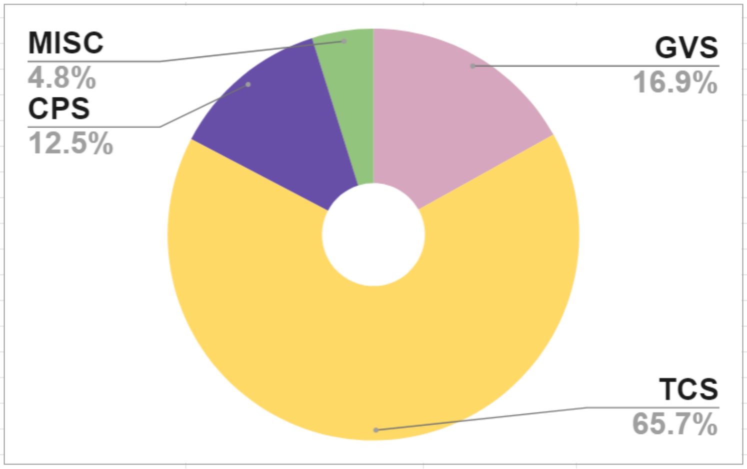


Fig. 35 Distribution of Expenses by Subsystem

Below in figure 36is a procurement breakdown of the CHAIR showing the number of items purchased for each subsystem. As you would expect, the TCS required the procurement of the most parts, nearly as much as the GVS and CPS combined.

Subsystem Procurement Breakdown					
Subsystem	Received	Autoclave	Pending	Backorder	Planned
Galvanic Vestibular Stimulator (GVS)	29	0	0	0	0
Tactile Cueing System (TCS)	51	0	0	0	0
Central Processing System (CPS)	29	0	0	0	0
Total	109	0	0	0	0

Fig. 36 Subsystem Procurement Breakdown

E. Test Plan

Testing arrangements must also be planned to ensure testing in the spring semester goes without issue. Figure 37, shows a general breakdown of the testing procedure.

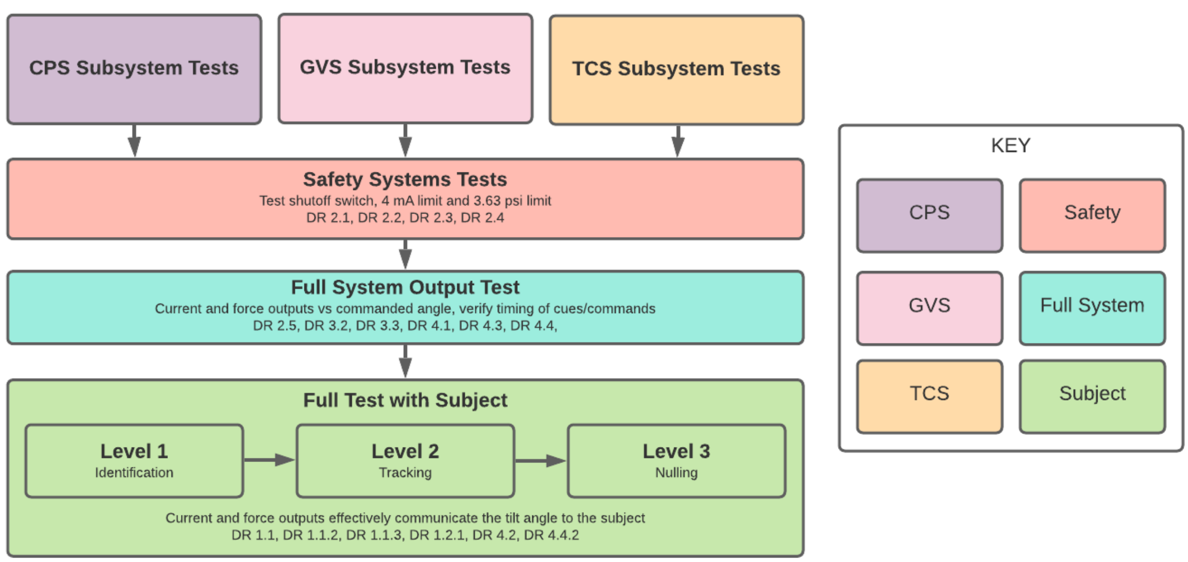


Fig. 37 Test Overview

Subsystem tests will be first. Followed by safety tests and then full system tests without and then with a test subject. Figure **??**, shows more specific tests to be done and when they will occur.

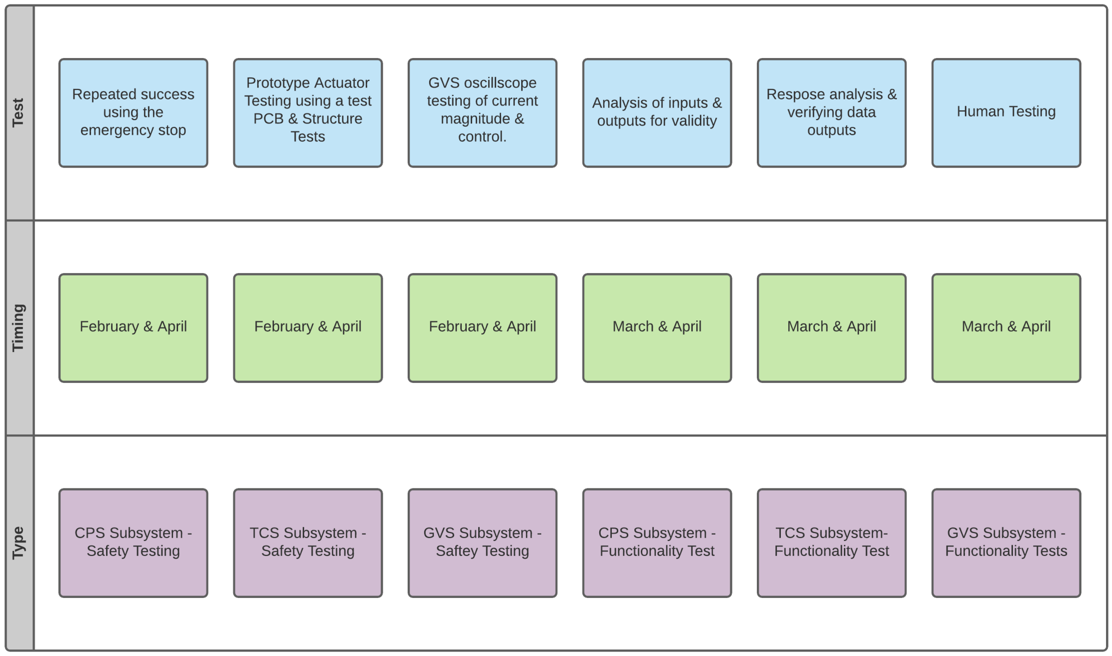


Fig. 38 Test Plan

The top row shows a list of primary tests that need to be accomplished. The second row gives the timeline for the testing. The last row gives the types of tests they classify as. The tests are broken into two stages during the Spring. The first stage begins with safety tests that occur in early to mid February. The second stage is verification and validation testing and will occur in early to mid March. All of these tests will be conducted in the Aerospace building. As April approaches, all of these tests will be redone while subsystems are integrated. As a subsystem was integrated with another, the safety and functionality tests would be done again. Full integration was never reached, however; this procedure holds true for partial integration of subsystems as well as if all three had been integrated.

VIII. Lessons Learned

Authors: Sarah Foley, & Baily Rice

A. Requirement and Objective Clarification

As the project progressed towards completion it became obvious that some of our requirements and objectives were not defined very well. Specifically, the levels of success and requirements pertaining to the full integration were too vague. First, the levels of success were far too general. The manner in which they are defined now, requires that the entire project be integrated to reach even the first level of success. The exact terminology of the levels of success requires that the CHAIR system cues tilt profiles. The exact makeup of that cue was never defined; rather the levels of success refer to the overall system. Thus, the cueing defined in the levels of success refers to a combined tactile and vestibular cue. Since the team was not ale to achieve full integration, technically the project has not even reached the first level of success. For example, even though we reached all these levels of success for the GVS, the GVS's success can not be rightfully recorded as achieving any of the success levels. This then leads to a situation in which individual subsystems reach different levels of success. So, the levels of success are not being limited by their requirements but by the integration status. The moment the project achieves full integration the project will immediately reach level three of success. This issue can be resolved with more well defined levels of success. One solution is to implement a tier system in addition to the levels of success. Each tier would contain the same requirements as the original levels of success. The first tier would require tactile and vestibular cueing to reach each level independently. Then tier 2 would be the same levels of success for the whole system. If the project could be redone, the levels of success would be better formatted like the following.

- Tier 1: Independent Cueing
  - Tactile Cueing

    1) Discrete/Static Tilt Cueing: The system is capable of cueing a discrete and sustained tilt in either direction. The joystick feedback capabilities desired from the customer can be implemented at the lowest level with the subject determining the direction of tilt through joystick deflection.
    2) Sinusoidal Tilt Cueing: The system is capable of cueing a continuous, sinusoidal tilt profile. This level corresponds to what could be described as a swaying motion.
    3) Continuous Variable Tilt Cueing: The system is capable of developing a continuous tilt profile in real time through the use of joystick input. Additionally, a predetermined tilt profile can also be used and even combined in real time with the joystick input.
  - Vestibular Cueing

    1) Discrete/Static Tilt Cueing: The system is capable of cueing a discrete and sustained tilt in either direction. The joystick feedback capabilities desired from the customer can be implemented at the lowest level with the subject determining the direction of tilt through joystick deflection.
    2) Sinusoidal Tilt Cueing: The system is capable of cueing a continuous, sinusoidal tilt profile. This level corresponds to what could be described as a swaying motion.
    3) Continuous Variable Tilt Cueing: The system is capable of developing a continuous tilt profile in real time through the use of joystick input. Additionally, a predetermined tilt profile can also be used and even combined in real time with the joystick input.
- Tier 2: Integrated Cueing

  1) Discrete/Static Tilt Cueing: The system is capable of cueing a discrete and sustained tilt in either direction. The joystick feedback capabilities desired from the customer can be implemented at the lowest level with the subject determining the direction of tilt through joystick deflection.
  2) Sinusoidal Tilt Cueing: The system is capable of cueing a continuous, sinusoidal tilt profile. This level corresponds to what could be described as a swaying motion.
  3) Continuous Variable Tilt Cueing: The system is capable of developing a continuous tilt profile in real time through the use of joystick input. Additionally, a predetermined tilt profile can also be used and even

combined in real time with the joystick input.

Following this structure allows the separate subsystems to reach their own levels of success. Allowing for one subsystem to move forward even if the other falls behind. When level three success has been achieved for both tactile and vestibular cueing, the team moves into the next tier.

In addition to the levels of success, some of the design requirements written in the Spring were not very well defined. The largest confusion stems from the fact that most of our requirements were built around a tilt angle value. CHAIR provides pressure and vestibular stimulation to create a perceived tilt. Human perception is very hard to quantify and therefore our requirements were difficult to verify. The team has updated the requirements to focus less on the perceived tilt and more on empirical values. The pressure and current models were derived using a 15 degree tilt, but the actual requirements should be written in terms of hard values.

B. Tactile Cueing Design

After multiple rounds of testing, patch fixes, and reassembly, the team feels that the mechanism chosen to perform the tactile cueing is insufficient for not only the forces expected but also the shape of the human body. Firstly, the load cells chosen have an incredibly small button head which makes good contact with it and the actuator head extremely difficult to achieve. At also encourages off axis loading and twisting of the pressure module, which also causes detriment to their functionality and too much shearing force is applied. Overall, the team would not have chosen these load cells if given the chance again, and would either more deeply research other types of load cells or found a different way to monitor the force on the pressure module. Another fix which the team felt would have helped if the same load cells had to be used, would be to fully embed them in the pressure module wood itself, as this would potentially decrease the likelihood of the actuator head slipping off the button, and also decrease the potential shearing forces they may be experiencing which causes them to break.

Another issue with the design of the tactile cueing system is the linear guides. For the magnitude of forces being applied in this project, they are too small to fully negate any twisting in the pressure module to keep it perpendicular to the actuator head. While their low friction operation is good, the sliding block is also very small, not providing enough contact area for the glued connection between it and the mechanism housing to be strong. If given the chance to choose this component again, the team would either choose one larger, wider linear guide to attach in a similar manner, or perhaps would design and construct our own rail system with wheels rather than ball bearings to do the job, such as a similar mechanism to rolling drawers.

IX. Individual Report Contributions

A. Cody Bahan

CHAIR Design (GVS), Manufacturing GVS Structures, Verification and Validation (GVS)

B. Rhys Bass

This semester I worked on the General CPS design, joystick code development, CPS to GVS communication, some testing. On the report I work on the CPS risk mitigation, some other CPS stuff and a portion of the appendix.

C. Sarah Foley

TCS Structural Chair Design, TCS Structural Manufacturing, TCS Structural Verification and Validation, TCS Structural Risk Assessment and Mitigation, TCS Lessons Learned, Overall TCS Manufacturing and Testing

D. Lane Franklin

Wrote the CPS Design, CPS Manufacturing, CPS Verification and Validation, and the outline for the CPS Risk Assessment section.

E. Cole Gray

Controls Design, Arduino code, electronics testing, TCS Design process, Budget plans and finances, Full scale integration

F. Carter Jackson

Electronics Design Process (TCS), TCS Design, Requirements, TCS trade studies, CHAIR design (TCS), PCB and circuit design, controls and testing

G. Michelle Lin

Updating CAD models and TCS Documentation, Team red teamer, Design Document and Presentation, Symposium Poster, AIAA work

H. Jason Magno

TCS Structural Verification and Validation, TCS Alternative Designs

I. Baily Rice

TCS V&V, Project Planning, & Lessons Learned. Main contribution to design was writing the communication code between the CPS & Arduino. Management focused on scheduling, organizing team members, updating the customer, & allocating resources to areas that needed them.

J. Andrew Ringer

Safety Work Breakdown, Safety/Test Plans, Risk Assessment & Mitigation (reviewed not authored),GVS Testing, CPS Joystick Integration/Test, Safety

K. Dean Widhalm

CHAIR Design (GVS), Manufacturing GVS Structures & Electronics, Risk Assessment & Mitigation (GVS)

L. Aiden Wilson

Developed and carried out tests, helped manufacture TCS, monitored risks, ensured requirement fulfillment, wrote Project Objectives & Functional Requirements, Design Process & Outline (Design Requirement Flowdown), Verification and Validation (Full System Verification and Validation) and Risk Assessment & Mitigation (TCS) sections.

References

[1] Anders Jespersen, MD, PhD, Kirstine Amris, MD, Thomas Graven-Nielsen, Professor, DMSc., Lars Arendt-Nielsen, Professor, DMSc., Else Marie Bartels, PhD, Søren Torp-Pedersen, MD, Henning Bliddal, Professor, DMSc., Bente Danneskiold-Samsoe, Professor, DMSc., Assessment of Pressure-Pain Thresholds and Central Sensitization of Pain in Lateral Epicondylalgia, Pain Medicine, Volume 14, Issue 2, February 2013, Pages 297–304

[2] "Electric Shock." Electric Shock Hazards, hyperphysics.phy-astr.gsu.edu/hbase/electric/shock.html.

[3] Inglis, J. T., Shupert C. L., Hlavacka, F., & Horak F. B., (February 1995). Effect of Galvanic Vestibular Stimulation on Human Postural Responses During Support Surface Translations. Journal of Neurophysiology, Vol. 73, No. 2, 899

X. Appendices

A. Alternative Designs

1. TCS

For the TCS, the most important design choice was the form of tactile cuing. The team conducted a trade study on two options, actuators and pressure bladders, which were judged on how they met five different criteria. Functionality was the most heavily weighted criteria because the design must fulfill all design requirements and levels of success to be viable. Manufacturablity was the next most important criteria because the team has limited time and resources available to construct a viable system, and there was a possibility of COVID-19 restrictions further limiting manufacturing capabilities. Complexity and cost were given equal weighting as design criteria. Significantly more complicated designs would have required more time to complete, and the team was also required to stay within a budget of 5,000 USD. Safety and the number of failure points were given the small weight in the trade study. In both designs, the failure points are typically known and can be properly addressed to mitigate the risk of failure. Also, neither of the TCS design options were expected to pose any serious safety hazards, and therefore safety was weighted low here while remaining a priority during the design process. A summary of these weighed criteria is given in Table 1.

Criteria	**Weight**	**Driving Requirements**	**Description/Rationale**
Number of Failure Points	10%	Requirement 1	Assessment of potential points in a mechanism that can fail during nominal use, N
Design Complexity	15%	Team Expertise Limitations	Assessment of difficulty to produce a valid system design and model
Functionality	30%	Requirement 1	Assessment of ability to meet functional requirements as defined in specific objectives
Safety	10%	Functional Requirement 2	The safety of the test subject is the highest priority during testing. However, all designs are comparably safe so the weighting for trade study purposes is low.
Manufacturability	20%	Project Timeline	The system's manufacturing needs must be weighed with the team's skills and timeline.
Cost	15%	Functional Requirement 5	With a limited budget cost is a very important aspect to this project and will be tracked extensively. The more complex the CHAIR is, the larger the budget will be.

Table 1 TCS Design Option Weighting Criteria

Both of the design options were given a score ranging from 1 to 5 for each criteria. A detailed chart which describes the scoring requirements for each criteria is shown in Table 2 below.

Criteria	**Scoring Requirements**				
	1	**2**	**3**	**4**	**5**
Number of Failure Points	More than 7	5-7 potential failure points	2-5 potential failure points	1-2 potential failure points	No potential failure points
Design Complexity	Impossible	Difficult	Moderate	Easy	Trivial
Functionality	Can not meet any objectives	Can meet few objectives	Can meet some objectives	Can meet most objectives	Can meet all specific objectives
Safety	Dangerous	High risk	Moderate risk	Low risk	Zero risk
Manufacturability	Entirely developed by team	Manufactured mostly by team, minimal parts COTS	Some parts COTS, some parts manufactured	COTS, needs assembly	COTS, pre-assembled
Cost	$1,500-$2,500	$1,000-$1,499	$500-$999	$0-$499	Free

Table 2 TCS Scoring Rubric

After tallying the scores, the actuator option scored higher and proved to be the more feasible option. This was mainly due to the teams confidence in the functionality of the system along with manufacturability and experience. The individual scores are summarized in Table 3 below.

Criteria	**Weight**	**Scoring**	
		Actuators	Pressure Bladder
Number of Failure Points	10%	2	4
Design Complexity	15%	3	3
Functionality	30%	5	3
Safety	10%	4	3
Manufacturability	20%	4	3
Cost	10%	2	4
Total Score = Σ (Scores*Weights)		**3.55**	**3.05**

Table 3 TCS Design Option Trade Study

The final trade study available for the TCS is for the microcontroller. From this trade study, the team finally decided that an Arduino MEGA 2560 would best suit are needs and be capable to run this circuit.

		Scoring				
Metric	Weight	Arduino Due	Arduino Mega 2560	STM32F0 DISCOVERY	Teensy 4.0	Raspberry Pi 4
Number of Pins	50%	4	5	3	3	4
Functionality with CPS	30%	4	4	3	3	4
Cost	10%	3	2	5	3	2
Processing Power	10%	4	5	4	3	4
Total Score = SUM(Scores*Weights)		4.25	4.4	3.5	3	3.8

Fig. 39 Microcontroller Trade Study

2. GVS

The Galvanic vestibular stimulator is a noninvasive way to stimulate movement by inducing electrical currents into the vestibular system. Our initial reach goal for this project was to cue multi-axial cueing. This was then re-scoped to single axis due to the fact that multi-axial cueing via multiple electrodes is still cutting edge GVS research. To begin our GVS method research, we had to section the trade study into different criteria in which we deemed were the most important. These different criteria and their descriptions are shown in Table 4. The most important characteristic for the team was technology readiness, as the team would not reasonably be able to develop a brand new or little understood GVS system under their time constraints.

Criteria	**Weight**	**Driving Requirements**	**Description/Rationale**
Technology Readiness	35%	Requirement 1.2	Designing a novel architecture to cue with GVS is an active research topic; it requires funding and time beyond the scope of our project. We must pick an approach that is feasible.
Cost	10%	Functional Requirement 5	Cost is unlikely to be a major factor in GVS construction, as most of the components are affordable ICs and microcontrollers.
Safety	15%	Functional Requirement 2	2-failure safety can be integrated into any GVS system easily via external fuses and software watchdogs. Too much resistivity in our system will be a much larger problem than potential shorts.
Ease of Integration	30%	Requirement 3.3	The GVS must be able to receive data from an external controller for our project to be successful.

Table 4 GVS Design Option Weighting Criteria

Now to actually decide on the best method that meets all of these criteria, we researched 5 different methods of this Galvantic Vestibular Stimulator. The first being a SOTERIX device, which is a COTS GVS machine capable of variant current delivery and multiple waveform options. The major downfall to this option was its proprietary firmware, more than expensive cost and inability to be digitally controlled. The next device was a custom two electrode setup. This setup had the ability to easily be integrated into the CHAIR and has a good deal of information among the research community. The only downside was that it had a vast increase in complexity compared to the SOTERIX device. The next two were both multi-electrode setups produced by the Mayo clinic and a Japanese study. Both of these configurations had unknown replicability and these multi-axis cueing systems are still in development. The Mayo-Clinic pulled ahead due to the fact that they derived a transformation matrix for researchers to map desired rotational velocity perception to electrode current. This could be used if we decided to go with multiple axial cueing. The final device was more of a novelty configuration that involved magnetic fields. This was mainly as a scientific curiosity and there actually hasn't been any developments in using these magnetic fields to cue specific directions. Further breakdown with the weighting criteria can be seen in Table 5 below.

Criteria	Weight	Scoring				
		SOTERIX	Custom 2-electrode	Multi-electrode (Mayo Clinic)	Multi-electrode (Japanese)	Magnetic Method
Technology Readiness	35%	4	4	3	1	1
Cost	10%	1	3	3	3	1
Safety	15%	4	3	3	3	1
Ease of Integration	30%	2	4	4	4	2
Total Score = Σ (Scores*Weights)		**2.8**	**3.35**	**3.0**	**2.3**	**1.2**

Table 5 GVS Design Option Trade Study

Once the GVS setup was finalized, the actual electrode type was chosen. The team chose to continue analysing TENS electrodes which are designed for electrical simulation across the skin-electrode interface. For the purpose of this project we chose self adhesive electrodes instead of using a gel or saline solution due to their ability to keep the skin at a constant resistance across many users.

3. CPS

There were a number of alternative CPS designs that were considered, both in terms of the hardware architecture and the software. The first alternative was the choice how what hardware to run the CPS on. We considered several different options including doing all of the processing on a micro-controller, using a main computer with one micro-controller or using a main computer with separate micro-controllers for each of the TCS and GVS system. We considered Cost, time complexity, ease of use and TCS/GVS system independence. The most important criteria for a computer architecture was decided to be the design complexity and subsystem independence. Based on the criteria and weights in tables 6 and 7 we selected the setup with a main computer and two separate micro-controllers. The final scores are shown in tables 8.

Criteria	Weight	Driving Requirements	Description/Rationale
Design Complexity	30 %	Requirement 3	In order to meet deadlines, a low design complexity is ideal to minimize time spent designing, building, and debugging systems.
Cost	20 %	Functional Requirement 5	Computers and microcontrollers capable of performing the required computing tasks are not expected to be egregiously expensive. CU Properties auctions can be leveraged to source affordable computer systems.
User Convenience	20 %	Functional Requirement 3	Consider how easily the architecture allows the test controller to command CHAIR and how easily the CPS can be reprogrammed for continued development.
Subsystem Independence	30 %	Project Timeline	Assess the architecture's ability to isolate the GVS, TCS and computing from each other during development. Systems independence will allow for asynchronous work between sub teams and allow for error isolation during development.

Table 6 Computer Architecture Weighting Criteria

Criteria	Scoring Requirements			
	1	**2**	**3**	**4**
Design Complexity	Requires custom fabrication of PCB	Requires external support hardware to run (e.g., external power supply)	Requires interfacing multiple COTS computational elements	All-in-one COTS solution
Cost	$1,000+	$500-$999	$100-$499	$0-$99
User Convenience	CPS must be flashed to reprogram & commands are input via mechanical switches	CPS Software must be recompiled to change CHAIR configuration parameters each time they are changed.	Controller could command CHAIR with a GUI and software runs on multiple devices.	Controller could command CHAIR with a GUI and all software is run on a single device
Subsystem Independence	CHAIR subsystems are inseparable for development and execution	CHAIR subsystems can be developed separately but require integration for testing	CHAIR subsystems can be developed separately and tested separately with support hardware (e.g. external power supply)	CHAIR subsystems can be developed simultaneously and each subsystem can run independently.

Table 7 Computer Architecture Scoring Rubric

Criteria	**Weight**	**Scoring**			
		Single Computer	Custom Stand-alone Unit	Central CPS, One Microcontroller	Central CPS, Two Microcontrollers
Design Complexity	30%	4	1	4	3
Cost	20%	4	2	3	3
User Convenience	20%	4	1	4	3
Subsystem Independence	30%	1	3	1	4
Total Score = Σ (Scores*Weights)		**3.1**	**1.8**	**2.9**	**3.3**

Table 8 Computer Architecture Trade Study

The second CPS design choice we made was the CPS language selection. We considered a number of languages include C/C++, Python, Matlab, labview and Java. We evaluated each language based on Speed, ability interface with hardware and factors related to amount of extra work required for language including team knowledge, ease of debugging and language support. We most heavily weighted by speed and team experience. The system speed and therefor the CPS speed is part of our design requirements and we felt that team experience would be the predominant factor determining the among of work to be done. We decided to go with C++ based on the trade study shown in tables 9-8.

Criteria	Weight	Driving Requirements	Description/Rationale
Team experience	20%	Requirement 3	In order to complete complete the software in time, there need to be enough team-members who can work on the code in an efficient time-frame.
Computational and Runtime Speeds	35%	Requirement 3.2	Since the code will have to run through some control logic, the speed of the code will contribute to the time delay between control input and the systems response. Compiled languages will have better performance than interpreted languages.
Ability to interface with hardware	20%	Requirement 3	Integrating the CPS with the controllers for the cueing devices is integral to the functioning of the project. Using a language that makes it easy to integrate the CPS and controllers will allow more efficient development and testing.
Language Support	5%	Requirement 3	While all languages have some form of documentation and at least a few built-in functions, it is notable that the speed of development for this project could be significantly impacted by these factors. In particular, built-in language support for manipulation of non-numerical variables and functions that assist with development of control algorithms and GUI creation could be invaluable to the rapid development of a stable and effective CPS.
Ease of Debugging	20%	Requirement 3	Debugging is always an important and often time-consuming component of software development. It is desirable to choose a computing language which allows the team to easily identify and fix errors in the code. Additionally, many languages have online resources and built-in functions and libraries that could assist in software development efforts. Both of these factors will need consideration, as they contribute both to final product quality and assurance of CPS completion within the product development timeline.

Table 9 CPS Language Option Weighting Criteria

Criteria	Scoring Requirements				
	1	2	3	4	5
Team Experience	0 software team-members know the language	1 software team-members know the language	2 software team-members know the language	3 Software team-members know the language	All software team members know the language
Computational and Runtime Speeds	Relative execution time of more than 4 seconds	Relative execution time of 3 to 4 seconds	Relative execution time between 2 and 3 seconds	Relative execution time between 1 and 2 seconds	Relative execution time of one second or less
Ability to interface with hardware	Language has no history of commercial use in hardware control, does not have any built-in support for interfacing with hardware, and does not allow user control over pointers.	Language has no history of commercial use in hardware control, but either allows user control over pointers or has some built-in support for interfacing with hardware.	Language has minimal history of commercial use in interfacing with hardware OR has the ability to interface with a second language that is capable of doing this. Either allows user control over pointers or has some built-in support for interfacing with hardware.	Language has a significant history of commercial use for interfacing with hardware. Allows for user control over pointers and has some built-in support for interfacing with hardware.	Language has a robust history of commercial use for interfacing with hardware or is designed with this purpose in mind.Allows for user control over pointers and has built-in support for interfacing with hardware.
Language Support	Language has no public documentation and only minimal built-in mathematical operations OR Language has built-in functions for handling non-numerical operations but there is little or no documentation describing these functions.	Language has a built-in functions for handling non-numerical operations, such as String comparison. There is documentation describing these functions.	Language has built in functions which include either libraries for control systems and transfer functions OR libraries for GUI creation. These libraries are well-documented. Underlying function code is not available to the programmer.	Extensive libraries of built-in functions, including libraries for control systems and transfer functions AND GUI creation, are available. Underlying code for functions is not available to the programmer, but all functions are documented.	Extensive libraries of built-in functions, including libraries for control systems and transfer functions AND GUI creation, are available. Underlying code for functions is available to the programmer.
Ease of Debugging	There is no IDE that provides assistance with debugging. No error messages are available.	Debugging assistance exists but consists only of error messages or may not be accessible in all coding environments.	Debugging assistance and robust error messaging are available in all coding environments, but debugging support or error messaging are not built into language.	Language either has built-in error messaging or many coding environments include debugging support. Language is interpreted rather than compiled.	Language either has robust built-in error messaging or built-in debugging functionality. Language is compiled rather than interpreted.

Table 10 CPS Language Scoring Rubric

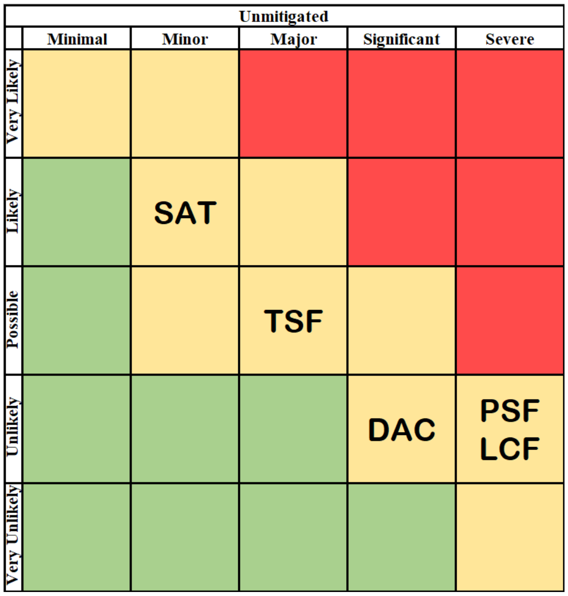
Criteria	**Weight**	**Scoring**					
		C	C++	Python	Java	MATLAB	LabVIEW
Team experience	20%	4	5	5	1	5	1
Computational and runtime speeds	35%	5	5	4	2	3	4
Ability to interface with hardware	20%	5	4	3	1	2	5
Language Support	5%	2	3	4	4	5	5
Ease of debugging	20%	4	4	5	4	4	3
Total Score = Σ (Scores*Weights)		**4.45**	**4.5**	**4.2**	**2.85**	**3.5**	**3.45**

Table 11 CPS Language Option Trade Study

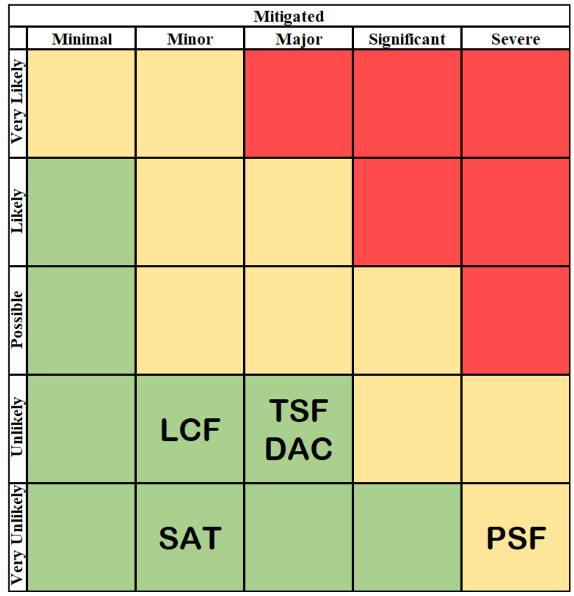
B. Risk of design failure

1. Hardware Risk Assessment

The matrices shown below indicate the likelihood and severity ratings of the five key hardware risks.



(a) Unmitigated hardware risk matrix.



(b) Mitigated hardware risk matrix.

- **SAT:** TCS actuator motor saturation. Sustained force cues can cause the actuator motors to become saturated which will result in the loss of tactile cueing.
  - **Mitigation:** The CPS will not command constant currents from the actuators for more than 10 seconds.
  - **Result:** The likelihood of motor saturation is reduced to being very unlikely but the severity remains the same. This solution will not impact TCS functionality since the selected actuators are capable of sustaining a constant force without power.
- **PSF:** Primary or secondary structural failure. Failure of the chair legs or seat which could result in injury to the test subject.
  - **Mitigation:** Structural analysis has already been undertaken and testing will occur prior to having a subject in the chair.
  - **Result:** The likelihood of primary or secondary structural failure has been reduced to very unlikely but the severity level remains at severe due to the potential for harm to the test subject.
- **TSF:** Tertiary structural failure. Failure of the mechanisms housing or pressure modules which could result in damage to the system and would set back the project timeline while repairs are made.
  - **Mitigation:** Similar to the primary/secondary structural failure risk, structural analysis has been completed and testing will take place prior to full system integration so that any necessary repairs are easier.
  - **Result:** The likelihood of tertiary structural failure has been reduced and the risk is classified as acceptable.
- **DAC:** GVS DAC evaluation board. The evaluation board for the GVS DAC must be made to order which introduces

a risk to the project timeline if suppliers are limited or other procurement issues arise.

- **Mitigation:** The team can make a DAC evaluation board in the event that there are issues with procurement. This will only be used in the event that there are problems with a supplier since it will take the team longer to manufacture the DAC evaluation board than it would a typical supplier.
- **Result:** The severity of the risk has been reduced since its impact on the project timeline will be lessened if the team can manufacture the evaluation board as a contingency.

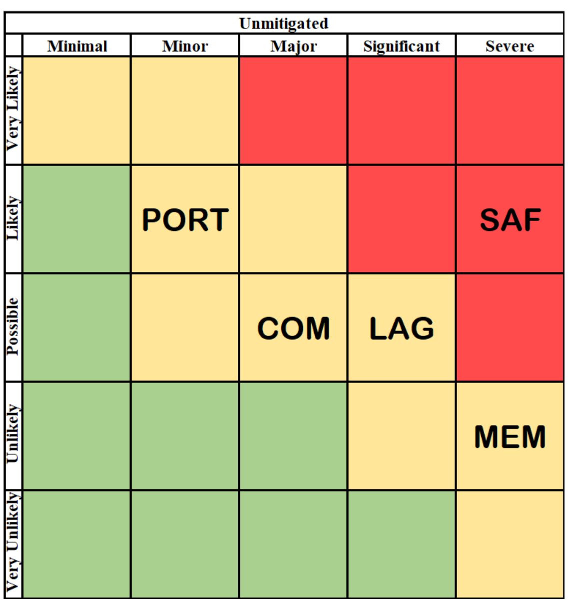
• **LCF:** Load cell feedback failure. Losing feedback from the load cells could result in the actuators exceeding the force safety limits for the test subject.

- **Mitigation:** A current limiter in each actuator will prevent the actuators from exceeding about 20 pounds of force regardless of load cell feedback.
- **Result:** The severity of the risk has been greatly reduced as it no longer poses a safety risk to the test subject.

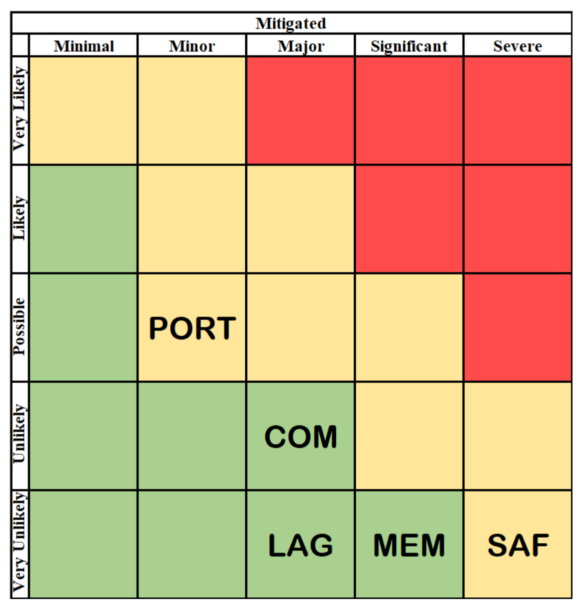
After mitigation, the only risk remaining in the potentially unacceptable region of the risk matrix is primary or secondary structural failure. This will continue to be mitigated with additional structural analysis, center of mass calculations and testing prior to test subject use.

2. Software Risk Assessment

The key software risks for CHAIR were assessed using the same scales as the hardware risks and are once again shown in the matrices below.



(a) Unmitigated software risk matrix.



(b) Mitigated software risk matrix.

• **LAG:** Excessive lag time. High lag times can result in unrealistic and possibly disorienting cues for the test subject.

- **Mitigation:** The team will use C++ as the CPS software language for its computation time capabilities as well as a peer review system to check for inefficient coding.

  - **Result:** The likelihood of the risk has been reduced to very unlikely and the severity has decreased.

- **COM:** Communication between the CPS and hardware. If the CPS is unable to communicate with the TCS and GVS hardware the system will not be able to operate properly.
  - **Mitigation:** The CHAIR system has been designed with two microcontrollers, one for each cueing system, that will facilitate communication between the CPS and the cueing devices.
  - **Result:** The likelihood of communication issues has been reduced to an acceptable level.
- **MEM:** Overuse of memory. Overuse of memory and memory leaks for the CPS will cause the CHAIR system to be unable to function properly.
  - **Mitigation:** A peer review system similar to the lag time risk will be used to ensure proper and efficient coding. Extensive testing will also be used to verify memory usage expectations.
  - **Result:** The severity and likelihood of the risk have decreased to an acceptable level.
- **PORT:** Portability problems. The team will not be providing a computer as part of the CHAIR project. The CPS software must be able to run on the test operator's computer for the system to operate.
  - **Mitigation:** The CPS will be designed with both Mac and Windows users in mind.
  - **Result:** The likelihood of the risk is decreased but it is still possible as some software is simply incompatible with Mac OS.
- **SAF:** Unsafe exports from the CPS to either of the cueing systems can result in injury to the test subject through excessive force or current. This risk has been assigned to the highest severity level as this is the where the greatest potential for unsafe testing conditions lies.
  - **Mitigation:** Multiple hardware and software safety checks for each subsystem, an automatic abort system, as well as a shutoff switch that is available to both the test operator and test subject.
  - **Result:** The risk has been lowered to be very unlikely but it remains in the potentially unacceptable region due to the potential for test subject injury in the case that it does occur.

After mitigation, two risks remain in the potentially unacceptable region. The first of which is portability, which will continue to be mitigated by ensuring that the CPS is capable of running on Windows while still designing for Mac users when possible. The second is unsafe commands for either of the cueing systems. This risk will continue to be mitigated through independent testing of each safety check and safety system prior to integration in the full system and again before test subject use.

C. Tests

1. TCS

The TCS test plan is shown in figure 51. The tests were designed to test the structural integrity of the TCS system, the mechanical performance of the actuators and the safety of the TCS system. The structural integrity of the chair was tested by loading the chair with more than the max weight to verify the strength of the seat pan, seat back and the general structure. The test plan also includes actuator performance tests including tests to verify the correctness of the actuator housing/unit construction, test for load cell feedback, saturation tests and actuator command tests.

Designed
Working
Completed
Issue
Safety ⚠

	Test Name	Description	Driving Requirements	Status
	Electrical component verification test	Verify that individually all components behave as described by the manufacturer	DR 1.1,DR 1.1.1 ,DR 1.1.3,DR 2.1,DR 2.2	
⚠	Single Command Single Actuator PCB test	Verify that the single actuator PCB is capable of commanding an uncontrolled force through the actuator	DR 1.1,DR 1.1.1,DR 1.1.3,DR 2.1	
⚠	Single Actuator Controls Test	Implement designed controls logic to command a range of forces dictated by tester on a single actuator	DR 1.1,DR 1.1.1 ,DR 1.1.3,DR 2.1	
	Actuator Assembly Test	Implement commands to an actuator once assembled in the housing and attached to the chair	DR 1.1, DR 1.1.1 ,DR 2.1	
⚠	Single actuator over commanding test	Verify system actuator does not output more than 25kPa when commanded above the required maximum	DR 1.1,DR 1.1.1 ,DR 1.1.3,DR 2.1	
⚠	Actuator Performance Test	Verify actuator performance is within pressure bounds when assembled and attached to the chair	DR 1.1,DR 1.1.3,DR 2.1	
	Lab space test	Verify that the structure can fit within 6'x6' lab space	DR 4.1,DR 4.2	
	Seat Pan structural test	Verify that the seat pan can support the customer's weight, with a FOS of 1.2	DR 1.1.2,DR 4.2	
	Seat back structural test	Verify that the seat back can support the back forces imparted by the customer with a FOS of 1.2	DR 1.1.2,DR 4.2	
	Center of Gravity Test	Testing the center of gravity is over the seat pan when in operation	DR 1.1.2,DR 4.2	
	Verify correct dimensions test	At least a 0.5" clearance between all surfaces except for the seat back, seat pan, and headrest.	DR 4.1,DR 4.2	
	Actuator Extension Test	Validate approriate extension.	DR 1.1,DR 1.1.3	
	Structural endurance test	Verify structual integrity after prolonged operation.	DR 4.2,DR 4.4,DR 4.4.2	
⚠	Multiple Operations Test	Operation system over mutiple full cycle operations	DR 4.3,DR 4.4,DR 4.4.2	
		DR 1.1 - The Tactile Cueing System (TCS) shall provide tactile cueing in the form of skin pressure to the test subject.		
		DR 1.1.1 - The TCS shall provide a minimum of 2.65 kPa on average over the stimulated areas.		
		DR 1.1.2 - The TCS shall remain static to generate tactile cueing for the test subject.		
		DR 1.1.3 - The TCS shall be able to cue a roll angle about the body x axis up to 15 degrees from the nominal upright position.		
		DR 2.1 - The TCS shall provide no more than 25 kPa on the test subject.		
		DR 2.2 - The TCS and test subject shall be electrically grounded.		
		DR 3.1 - The TCS shall be provided with a method of communication to the Central Processing System.		
		DR 4.1 - Will be able to operate within a space no larger than 6' x 6'.		
		DR 4.2 - The TCS shall accommodate a male body of 50th percentile height and 50th percentile weight.		
		DR 4.3 - The TCS shall have a minimum operable time of 30 minutes.		
		DR 4.4 - The GVS and TCS systems will be able to operate repeatedly without losing functionality.		
		DR 4.4.2 - The GVS and TCS shall not lose functionality over multiple uses		

Fig. 42 TCS Test Safety and Requirements Mapping

TCS	
Seat Structural Test	**Complete**
Center of Gravity Test	Complete
Actuator Assembly Test	Complete
Actuator Performance Test	Complete

Fig. 43 TCS: Seat Structural Test

TCS	
Seat Structural Test	Complete
Center of Gravity Test	**Complete**
Actuator Assembly Test	Complete
Actuator Performance Test	Complete

Fig. 44 TCS: CG Test

TCS	
Seat Structural Test	Complete
Center of Gravity Test	Complete
Actuator Assembly Test	**Complete**
Actuator Performance Test	Complete

Fig. 45 TCS: Actuator Assembly Test

TCS	
Seat Structural Test	Complete
Center of Gravity Test	Complete
Actuator Assembly Test	Complete
Actuator Performance Test	Complete

Fig. 46 TCS: Actuator Performance Test

2. GVS

The GVS Test plan includes tests design to test that the GVS outputs the correct commanded current and that no excessive current are accepted and output. There are test meant to test varying currents and different electrode setups.

Designed
Working
Completed
Issue
Safety ⚠

	Test Name	Description	Driving Requirements	Status
	Phase 1			
	Send a +000 and -000	Verify no operation	DR 1.2	
	Send +400, +000, -400, +000 in order	DR 1.2.1, also verifies current can be stopped with cmd Assess signal stability over time. (what is stable?)	DR 1.2,DR 2.3	
	Send various values between [+400 and -400]	(valid codes, should be accepted and written)	DR 1.2,DR 2.3	
	Send values between [+625 +400] and [-625 and -400]	(valid DAC codes exist, but should be rejected by the uC)	DR 1.2,DR 2.3	
⚠	Send values in excess of +625 and -625	(valid DAC codes do not exist, should be rejected by uC)	DR 1.2,DR 2.3	
	Send junk values	(e.g., 9999, asdf, ~$@#$, all should be rejected) Might need to add num-specific acceptance criteria	DR 2.3	
⚠	Send +3.12, then send +3.13, then repeat for the negatives	Test and measure middle glitch power, i.e. the maximum error on value transition		
	Start a 1mA current, leave running for 20 minutes.	verify current accuracy is acceptable (what is acceptable?). Repeat for 4mA	DR 4.4.2,DR 4.4	
	POR test	Press the Power button. Record current. Press again. Record current.	DR 4.4.2,DR 4.4	
⚠	Unplug UART/Power test	Send 4mA current cmd, unplug device. Record current. Plug back in. Record current.		
⚠	Unplug 24V power test	Send 4mA current cmd, unplug 24V power. Record current. Plug bag in. Record current.	DR 4.4.2,DR 4.4	
	Phase 2	Current sent through 2 electrodes placed on the R forearm + oscilloscope Modified code running 1s self-aborting signal?	DR 1.2	
	Current sent through 2	Send +/- 1mA and +/- 4mA, verify that signals do not spike	DR 1.2	
	Phase 3	Current sent through 2 electrodes placed on the mastoids + oscilloscope	DR 1.2	
	Apply electrodes across the mastoids. Send a +/- 1mA command	Assess current response	DR 1.2	
	Send a +/- 4mA command.	Assess current response.	DR 1.2	
		DR 1.2 - The Galvanic Vestibular Stimulator (GVS) shall provide vestibular cueing in the form of electrical currents through nodes placed on the mastoids of the test subject.		
		DR 2.3 - The GVS will have a maximum output that will not exceed 4mA of amperage.		
		DR 4.4 - The GVS and TCS systems will be able to operate repeatedly without losing functionality.		
		DR 4.4.2 - The GVS and TCS shall not lose functionality over multiple uses.		

Fig. 47 GVS Test Safety and Requirements Mapping

GVS	
Software & Component Testing	**Complete**
Resistor "Dummy Load" Testing	Complete
Human Testing	Complete

Fig. 48 GVS: GVS Softwere And Component Testsing

GVS	
Software & Component Testing	Complete
Resistor "Dummy Load" Testing	**Complete**
Human Testing	Complete

Fig. 49 GVS: Dummy Load Testing

GVS	
Software & Component Testing	Complete
Resistor "Dummy Load" Testing	Complete
Human Testing	**Complete**

Fig. 50 GVS: Human Testing

3. CPS

The CPS test plan was designed to key interfaces and functionality of the CPS system. There are a number of tests to verify input from the joystick, UI inputs and the communication between the main computer and micro-controllers. The test plan also includes tests meant to test the timing and performance tests including sync delay, joystick delay, total delay and terminate time tests. There are also safety test include verifying that unsafe commands don't go through and that termination buttons work. Finally there are test to verify the subsystem functionality and that we've meet general requirements.

Designed
Working
Completed
Issue
Safety ⚠

	Test Name	Description	Driving Requirements	Status
	CPS Safety Test	/		
⚠	Test Subject Button Test	Verify functionality of the test subject's terminate button	DR 2.5	
⚠	Operator Button Test	Verify functionality of the operator's terminate button	DR 2.5	
⚠	Joystick Interfreence Test	Command a safe value but make it unsafe with joystick augmentation and ensure that it still terminates.	DR 2.4	
⚠	TCS Safe Level Test	Record feedback data from the Arduino (load cells) to verify force of actuation. Ensure it is within safe bounds.	DR 2.4	
⚠	TCS Malicous Feedback Test	Create unsafe feedback from the TCS load cells and ensure that the test terminates	DR 2.4	
⚠	CPS Command Test	Record the CPS output while commanding values higher than the allowed maximums. Ensure CPS does not command the in-apprortiate values.	DR 2.4	
	Supporting Tests			
	CPS to GVS Comms Test	Establish UART Communication for the PIC and verify data sent & received.	DR 3.2 DR 3.4 DR 3.3	
	CPS to TCS Comms Test	Establish UART Communication for the Arduino and verify data sent & received.	DR 3.1 DR 3.2 DR 3.3 DR 3.4	
	CPS to TCS Command Test	Integrate Arduino software with the test hardware to fully test and tune a single actuator.	DR 3.1 DR 3.2 DR 3.3 DR 3.4	
	CPS to GVS Command Test	Integrate PIC software with the test hardware to fully test and tune GVS.	DR 3.2 DR 3.3 DR 3.4	
	System Comms Test	Verify successful communication & integration with CPS main program.		
	Joystick Comms Test	Verify that the CPS can read the joystick data?	DR 2.4 DR 2.5 DR 3.3	
⚠	Terminate Response Time Test	Record CPS output and measure time between stop switch and outputs reaching 0	DR 2.5	
	Joystick Delay Test	Record joystick input and CPS output and measure delay between command and cue (will need to be less than 200 ms to allow for actuation)	DR 3.3	
	Sync Test	Use the user interface to command GVS and TCS magnitude and direction. Measure the microcontroller outputs to make sure they match up.	DR 3.4	
	Sync Delay Test	Record signal sent to TCS and signal sent to GVS and measure delay between equal states (will need to be less than 100 ms to allow for actuation)	DR 3.2 - The CPS shall coordinate the TCS and GVS responses, such that the time delay between the TCS and GVS cues as experienced by the test subject is less than 100 ms.	
	Main Safety Verification	**Fullfilled by:** Joystick Comms Test,Joystick Interfreence Test,TCS Safe Level Test,TCS Malicous Feedback Test,CPS Command Test	**DR 2.4** - The CPS will not instruct the TCS or the GVS to exceed the maximum allowed outputs described in requirements 2.1 and 2.3. (25 kPa, 4mA)	
	Stop Switch Verification	**Fullfilled by:** Joystick Comms Test,Test Subject Button Test,Operator Button Test	**DR 2.5** - The CPS will be equipped with an emergency stop switch that will enable all functions to be terminated immediately.	
	TCS/CPS Integration Verification	**Fullfilled by:**TCS Comms Test, TCS Command Test	**DR 3.1** - The TCS shall be provided with a method of communication to the Central Processing System.	
	Cue Sync Verification	**Fullfilled by:** TCS Comms Test,TCS Command Test,GVS Command Test, GVS Comms Test,Sync Delay Test	**DR 3.2** - The CPS shall coordinate the TCS and GVS responses, such that the time delay between the TCS and GVS cues as experienced by the test subject is less than 100 ms.	
	Joystick Time Verificaiton	**Fullfilled by:** Joystick Comms Test,TCS Comms Test,TCS Command Test, GVS Command Test,GVS Comms Test,Joystick Delay Test	**DR 3.3** - The CPS shall coordinate the hardware response such that the time delay from the joystick signal commanded by the test subject to the TCS and GVS cueing is within 200 ms.	
	UI Verification	**Fullfilled by:** TCS Comms Test,TCS Command Test,GVS Command Test, GVS Comms Test,Sync Test	**DR 3.4** - The CPS shall incorporate a user interface allowing the controller to choose the magnitude and direction of tilt cued by the GVS and TCS systems.	

Fig. 51 CPS Test Safety and Requirements Mapping

CPS	
CPS Command	**Complete**
CPS to GVS Comms	Complete
CPS to GVS Command	Complete
Sync Delay	Complete
Test Subject Button	Complete

Fig. 52 CPS: Command Test

CPS	
CPS Command	Complete
CPS to GVS Comms	**Complete**
CPS to GVS Command	Complete
Sync Delay	Complete
Test Subject Button	Complete

Fig. 53 GVS: GVS Softwere And Component Testsing

CPS	
CPS Command	Complete
CPS to GVS Comms	Complete
CPS to GVS Command	**Complete**
Sync Delay	Complete
Test Subject Button	Complete

Fig. 54 CPS: CPS to GVS Command Test

CPS	
CPS Command	Complete
CPS to GVS Comms	Complete
CPS to GVS Command	Complete
Sync Delay	**Complete**
Test Subject Button	Complete

Fig. 55 CPS: Sync Delay Test

CPS	
CPS Command	Complete
CPS to GVS Comms	Complete
CPS to GVS Command	Complete
Sync Delay	Complete
Test Subject Button	**Complete**

Fig. 56 CPS: Test Subject Button