University of Colorado
Department of Aerospace Engineering Sciences
ASEN 4018

Conceptual Design Document (CDD)

CHAIR - Cueing via Haptics And Inner-ear Responses

Project Customers

Name:	Torin Clark
Email:	torin.clark@colorado.edu
Phone:	(303) 492-4015

Team Members

Name: Cody Bahan Email: coba4813@colorado.edu Phone: 310-780-1964 Position: GVS Research Lead	Name: Daniel Cole Gray Email: dagr5648@colorado.edu Phone: 678-699-8551 Position: Financial Lead	Name: Baily Rice Email: bari4395@colorado.edu Phone: 815-762-7959 Position: Project Manager
Name: Rhys Bass Email: hba2118@colorado.edu Phone: 720-398-1222 Position: Software Systems Lead	Name: Carter Jackson Email: caja6733@colorado.edu Phone: 720-412-2383 Position: Electronics Lead	Name: Andrew Ringer Email: anri7493@colorado.edu Phone: 480-427-1694 Position: Testing, Safety Lead
Name: Sarah Foley Email: safo9930@colorado.edu Phone: 617-833-9658 Position: Manufacturing Lead	Name: Michelle Lin Email: shuyu.lin@colorado.edu Phone: 720-427-5038 Position: Structures Lead	Name: Dean Widhalm Email: kawi7969@colorado.edu Phone: 972-658-3724 Position: GVS Software Lead
Name: Laney Franklin Email: hefr7048@colorado.edu Phone: 978-491-0501 Position: TCS Software Lead	Name: Jason Magno Email: jama7200@colorado.edu Phone: 719-285-3301 Position: TCS Research Lead	Name: Aiden Wilson Email: aiwi8410@colorado.edu Phone: 208-871-8132 Position: Systems Lead

Table 1: Team Member Information

Contents

1 Project Description 1
1.1 Project Purpose 1
1.2 Specific Objectives 1
1.3 Concept of Operations 3
1.4 Functional Block Diagrams 4
1.5 Functional Requirements 5

2 Design Requirements **5**

3 Alternative Designs **9**
3.1 TCS Alternatives 9
3.1.1 Actuator Tactile Cueing 11
3.1.2 Pressure Bladder Tactile Cueing 13
3.2 GVS Alternatives 15
3.2.1 SOTERIX Galvanic Vestibulation Stimulator 15
3.2.2 Custom GVS, Two electrodes 15
3.2.3 Custom GVS, Multiple electrodes (Mayo Clinic Configuration) 16
3.2.4 Custom GVS, Multiple electrodes (Japanese Study Configuration) 17
3.2.5 Magnetic Vestibular Stimulation 18
3.3 CPS Language Alternative Designs 19
3.3.1 C 19
3.3.2 C++ 20
3.3.3 Java 20
3.3.4 Python 21
3.3.5 Matlab 21
3.3.6 LabVIEW 22
3.4 CPS Architecture Alternative Designs 22
3.4.1 Single Computer Setup 22
3.4.2 Standalone Processing Unit 23
3.4.3 Central CPS, One Shared Microcontroller for TCS and GVS 23
3.4.4 Central CPS, Two Separate Microcontrollers for TCS and GVS 24

4 Trade Studies **25**
4.1 TCS Design Options 25
4.2 GVS Trade Study 27
4.3 Computer Language Trade Study 29
4.4 Computer Architecture Trade Study 34

5 Baseline Selection **37**
5.1 TCS Design Selection 37
5.2 GVS Design Selection 37
5.3 CPS Language Selection 37
5.4 Computer Architecture Selection 37
5.5 Baseline Design Selection 38

List of Figures

1 Body Centered Axes of Remote Pilot 1
2 Concept of Operations 4
3 CHAIR Functional Block Diagram 4
4 (a) Side view of subject positioning for CG measurements modified from FAA documentation. (b) Front view of subject tilted 15° with gravity force shown. Modified from NASA documentation. 10
5 Point Mass Model of TCS Subject 10
6 Preliminary Model of Actuator Cued TCS 12
7 Firgelli Automations Linear Actuator 13
8 Harley Davidson Motorcycle Seat Pad 14
9 Preliminary Model of Bladder Tactile Cueing 14
10 SOTERIX GVS Setup 15
11 Diagram of custom 2 electrode design. The head is drawn as a variable resistor, connecting to two nodes at the mastoid processes 16
12 Mayo Clinic Study GVS Configuration 17
13 Japanese Study GVS Configuration 18
14 Example of an MRI Machine capable of stimulating vestibular perceptions 19
15 Single Computer Setup Diagram 23
16 Standalone Processing Unit Setup Diagram 23
17 Central CPS, One Shared Microcontroller Setup Diagram 24
18 Central CPS, Two Separate Microcontrollers Setup Diagram 25

List of Tables

1 Team Member Information i
2 SOTERIX Pros & Cons 15
3 Custom GVS, Two electrodes Pros & Cons 16
4 Mayo Clinic configuration Pros & Cons 17
5 Japanese Researchers configuration Pros & Cons 18
6 Magnetic Vestibular Stimulation Pros & Cons 19
7 C Pros & Cons 20
8 C++ Pros & Cons 20
9 Java Pros & Cons 21
10 Python Pros & Cons 21
11 Matlab Pros & Cons 22
12 LabVIEW Pros & Cons 22
13 TCS Design Option Weighting Criteria 26
14 TCS Scoring Rubric 26
15 TCS Design Option Trade Study 27
16 GVS Design Option Weighting Criteria 28
17 GVS Scoring Rubric 29
18 GVS Design Option Trade Study 29
19 CPS Language Option Weighting Criteria 31
20 CPS Language Scoring Rubric 33

21 CPS Language Option Trade Study 34
22 Computer Architecture Weighting Criteria 35
23 Computer Architecture Scoring Rubric 36
24 Computer Architecture Trade Study 36

Nomenclature

CHAIR	Cueing via Haptics And Inner-ear Responses
CPS	Central Processing System
GVS	Galvanic Vestibular System
TCS	Tactile Cueing System
COTS	Consumer Off The Shelf
Controller	Person using CPS to command cues
Test Subject	Person using joystick to command cues and experiencing cues

1 Project Description

1.1 Project Purpose

In the modern age more military and civil aviation roles are 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, even in the circumstance of disaster. 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. The CHAIR project is designed to combat these very issues. The CHAIR will provide tactile and vestibular stimulation to remote pilots related directly to the drone's attitude. These sensations will allow remote pilots to hone in on the motion of their aircraft for greater control that may lead to less drone crashes due to pilot error.

1.2 Specific Objectives

For the remainder of the project, the team will refer to the bodily axes defined in Figure 1 below when referencing motion of the pilot. Also, the team will use the term test subject to refer to the person receiving the tactile and vestibular cues, and controller for the person operating CHAIR from the Central Processing System.

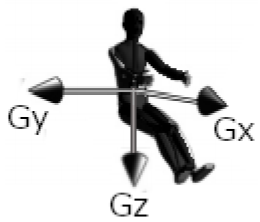

Figure 1: Body Centered Axes of Remote Pilot

The following levels describe the specific objectives that the CHAIR system will accomplish, in order of importance.

Level 1

- 1.1 - Computation
  - 1.1.1 - The Central Processing System (CPS) will be able to simultaneously communicate to both the Galvanic Vestibular Stimulator (GVS) and Tactile Cueing System (TCS) elements to cue the sensation of a motion about the Gx axis.

    - 1.1.2 - The CPS will allow the controller to induce either left or right rolls and the test subject shall be subsequently stimulated by the GVS and TCS.
    - 1.1.3 - The test subject will be allowed to choose direction of static tilt with a joystick and be subsequently stimulated by the GVS and TCS.
- 1.2 - GVS
    - 1.2.1 - The GVS will cue a static roll angle of up to fifteen degrees about the Gx axis.
    - 1.2.2 - Test subjects will not experience any pain caused by the GVS during operation
- 1.3 - TCS
    - 1.3.1 - The TCS will cue a static roll angle of up to fifteen degrees about the Gx axis .
    - 1.3.2 - The TCS will cue a change in orientation on the test subject by providing contact force on the test subject proportional to gravity and the degree sine of roll angle about the Gx axis.
    - 1.3.3 - The chair will be designed to fit a male body of 50th percentile height and 50th percentile weight.
    - 1.3.4 - Test subject will not experience any pain caused by the TCS during operation.
- 1.4 - Systems Integration and Verification
    - 1.4.1 - The GVS and TCS will work in concert to simulate the sensation of a roll angle about the Gx axis; the lag time between the two system perceived cues on the test subject will be less than 100ms.
    - 1.4.2 - The test subject will correctly identify direction of roll induced by controller.

Level 2

- 2.1 - Computation
    - 2.1.1 - The CPS will create a variable sinusoidal roll profile that the GVS and TCS shall cue on the test subject.
- 2.2 - GVS
    - 2.2.1 - The GVS will cue a sinusoidal roll angle of amplitude fifteen degrees about the Gx axis.
- 2.3 - TCS
    - 2.3.1 - The TCS will cue a sinusoidal roll angle of amplitude fifteen degrees about the Gx axis.
- 2.4 - Systems Integration and Verification
    - 2.4.1 - The lag time between the joystick commands and the GVS and TCS perceived cues will be reduced to less than or equal to 200ms
    - 2.4.2 - The test subject will correctly identify direction of roll induced by controller.

Level 3

- 3.1 - Computation
  - 3.1.1 - The CPS will interface with a joystick input to allow the test subject to input roll direction and magnitude about the Gx axis
- 3.2 - GVS
  - 3.2.1 - The GVS will induce a static roll angle of up to twenty degrees about the Gx axis.
  - 3.2.2 - The GVS will cue the roll angle profile input by the test subject using the joystick.
- 3.3 - TCS
  - 3.3.1 - The TCS will induce a static roll angle of twenty degrees about the Gx axis.
  - 3.3.2 - The TCS will cue the roll angle profile input by the test subject using the joystick.
- 3.4 - Systems Integration and Verification
  - 3.4.1 - The lag time between the GVS and TCS perceived cues will be reduced to less than or equal to 50ms.
  - 3.4.2 - The lag time between joystick input and GVS and TCS perceived cues will be reduced to less than or equal to 100 ms.

1.3 Concept of Operations

The concept of operations is seen in Figure 3 below. To begin the cueing of the test subject some preparation must be done. First, the test subject is to be secured into the TCS and the GVS will then be mounted. With the permission of the test subject, the controller will initiate the CPS. The test subject will then be cued according to the CPS commands. Alternatively, the test subject may also control the cueing with joystick input. The test subject perceives said cueing and can then feedback into the control with new joystick input.

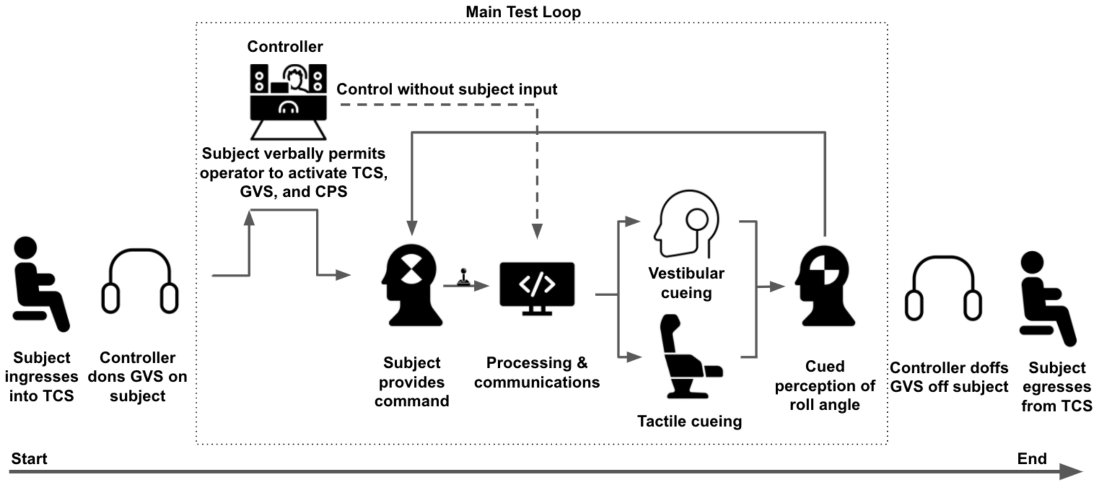

Figure 2: Concept of Operations

1.4 Functional Block Diagrams

The CHAIR project consists of three major sub-systems, the TCS, GVS, and CPS. Below is a functional block diagram showing how all these subsystems interact.

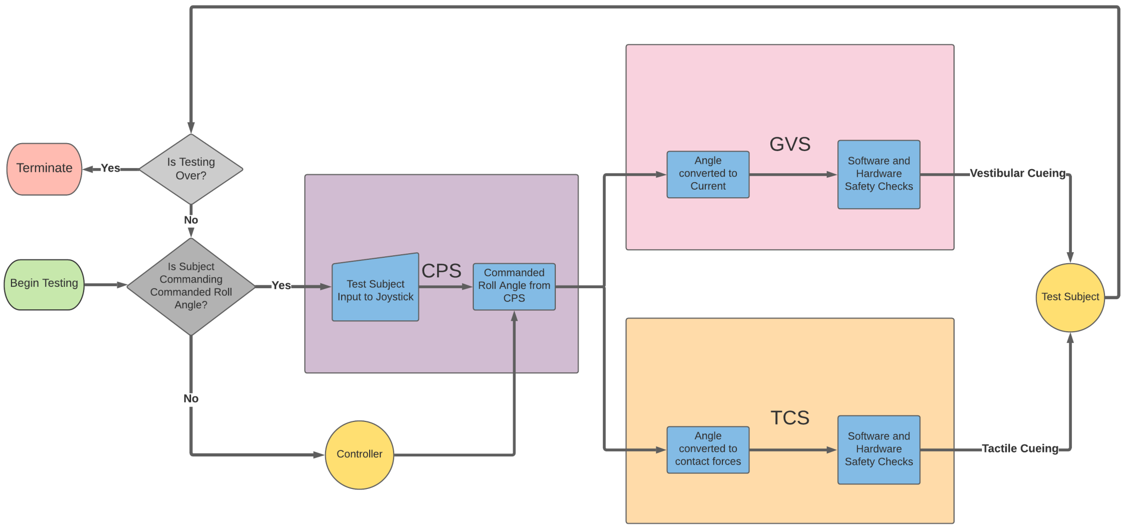

Figure 3: CHAIR Functional Block Diagram

1.5 Functional Requirements

1. The test subject will receive vestibular and tactile cueing to convey representative flight orientation.
2. Test subject will not experience any pain caused by the operation of the system.
3. The GVS and TCS will be integrated to respond to test controller input as well as joystick input.
4. The TCS will be operable within common conditions of a lab.
5. The total development of the combined software and hardware systems must not exceed a total cost of 5,000 USD.

2 Design Requirements

Functional Requirement 1

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

Motivation: Customer specified requirement.
Verification: Demonstration. Show that TCS and GVS performs according to expectations outlined in 1.1 and 1.2 sub-requirements.

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.
Verification: Demonstration. Show that TCS mechanisms perform according to expectations outlined in 1.1 sub-requirements.

DR 1.1.1 - The TCS shall provide a minimum of 2.65 kPa on average over the stimulated areas.
Motivation: The TCS shall provide enough force to be perceived by the test subject.
Verification: Test. Force sensors placed on the subject's stimulated areas.

DR 1.1.2 - The TCS system shall remain static to generate tactile cueing for the test subject.
Motivation: Customer specified requirement.
Verification: Demonstration. Show that the TCS frame does not move during operation.

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.
Validation: Test subject correctly identifies tilt direction during at least 10° deviation from the nominal upright position; test subject correctly identifies magnitude differences between a 5° and 15° deviation from the nominal position.
Verification: Inspection. TCS is capable of receiving input of a continuous tilt profile with maxima 15° deviation from nominal in either direction and mechanisms will perform commensurate with the direction and magnitude of the command.

DR 1.2 - The Galvanic Vestibular Stimulator (GVS) shall provide vestibular cuing in the form of electrical currents through nodes placed on the mastoids of the test subject.
Motivation: Customer specified requirement.
Verification: Demonstration. Show that GVS performs according to expectations outlined in 1.2 sub-requirements.

DR 1.2.1 - The GVS shall be able to cue roll angles about the body x axis of up to 15° from the nominal upright position. [4]
Motivation: Customer specified requirement. Prevent disorientation from too high of angular velocity.
Validation: Test subject correctly identifies tilt direction with at least 10° roll angle; test subject correctly identifies magnitude differences between a 5° or 15° roll angle.
Verification: Inspection. GVS is capable of receiving input of a continuous tilt profile with maxima 15° tilt in either direction and electrodes will perform commensurate with the direction and magnitude of command.

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.
Verification: Assess test subject pain level before, during, and after using CHAIR.

DR 2.1 - The TCS shall provide no more than 25 kPa on the test subject. [2]
Motivation: Excessive pressure will lead to discomfort of the user.
Validation: Test subject will report if any pain is felt.
Verification: Test. Pressure sensors placed on the subject's stimulated areas.

DR 2.2 - The TCS and test subject shall be electrically grounded.
Motivation: Avoid electrical damages to the system and injury to the subject.

Verification: Test. Voltage measurements indicate electrical grounding.

DR 2.3 - The GVS will have a maximum output that will not exceed 4mA of amperage. [1]
Motivation: Excessive current will lead to discomfort of the user as well as potential health hazards.
Validation: Test subject will communicate report if any pain is felt.
Verification: Test. Extrema of possible outputs will be tested with a digital multimeter to ensure the electrode outputs do not exceed maximum current.

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.
Validation: Software will terminate if currents commanded to TCS and GVS exceed allowable limits.
Verification: Test. Software will be instructed to command above maximum currents to observe termination.

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.
Verification: Demonstrate. Stop switch will be enabled during testing.

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.
Verification: Test. TCS and GVS will respond in concert to specified software commands regardless of the source.

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.
Verification: Demonstrate. The CPS will receive joystick input from TCS.

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. [5]
Motivation: In order to avoid test subject disorientation, the GVS and TCS must cue

the same motion with little to no time delay.
Verification: Test. Observe the time difference between sensor feedback from the TCS and GVS systems.

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.
Verification: Test. Observe the time difference for the signals from the joystick to sensor feedback from the TCS and GVS systems.

DR 3.4 - The CPS shall incorporate a user interface allowing the controller to choose magnitude and direction of roll 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.
Verification: Demonstrate. The user interface will be present and will input correct roll characteristics to CPS.

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.
Verification: Inspection and Test. Customer will participate in the final check of CHAIR operations and verify that the requirements have been met.

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.
Verification: Demonstrate. The TCS shall be fully tested in a 6' x 6' area.

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.
Verification: TCS will fit a 6 ft, 175 lb male body with up to a 1 in clearance in all dimensions.

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.
Verification: Demonstrate. TCS will be operated at maximum time.

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.
Verification: Test. CHAIR will be in the same condition before and after all tests are completed.

- **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 5th percentile male, at a -30 degree angle from horizontal.
  Validation: The customer can operate for the maximum operable time without damages to the joystick.
  Verification: Test and Inspection. Joystick does not break when subjected to 44.8 lbf in all directions.
- **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.
  Verification: Test and Inspection. The output of the GVS and TCS should remain the same with respect to any given input after multiple uses.

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.
Verification: Record. The financial lead will maintain a detailed log of all funds.

3 Alternative Designs

The team next researched potential alternative designs and ideas for the TCS, GVS, CPS software language and CPS architecture.

3.1 TCS Alternatives

In accordance with the design requirements, the TCS will be applying between 3 and 25 kPa of pressure on the test subject. While the exact effects of the induced pressure on the test subject will depend on the final design, an understanding of how the forces will be applied is beneficial for determining design viability. The center of gravity of a person sitting with one hand on a control stick was determined from a Federal Aviation Agency (FAA, now Federal Aviation Administration) report [6]. It was claimed that among the sample of males studied, the average center of gravity was 9 1/16 inches from the back of the chair and 9 7/8 from the bottom of the chair. The study did not mention the hip breadth of their subjects, so in order to determine the lateral location of the center of gravity the measurement for a 50th percentile male from NASA standards [7] was used. The NASA documentation reported this measurement to be 15.1 inches. Assuming a symmetrical weight distribution across the bodies midline, this puts the center of gravity 7.55 inches from forces being applied from the side. A sketch of the test subject with gravitational force at a 15° roll angle is shown in Figure 4 below.

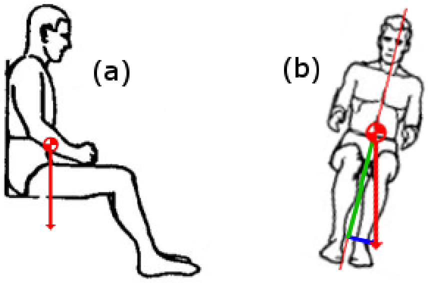

Figure 4: (a) Side view of subject positioning for CG measurements modified from FAA documentation. (b) Front view of subject tilted 15° with gravity force shown. Modified from NASA documentation.

The maximum side force applied during the maximum TCS pressure of 25 kPa can be simply modelled by approximating the pilot as a point mass. This model is shown in Figure 5 and shown by equations 1 and 2.

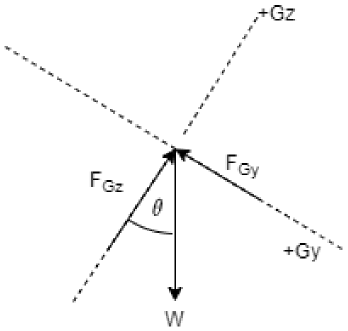

Figure 5: Point Mass Model of TCS Subject

$$\Sigma F_{G_y} = mgsin(\theta) - F_{G_y} \tag{1}$$

$$\Sigma F_{G_z} = -mgcos(\theta) + F_{G_z} \tag{2}$$

The team will consider both actuator based tactile cueing as well as pressure bladder based tactile cueing. Both of these cueing mechanisms are based on the contact forces exerted by a seat on a pilot during flight. These forces are directly related to the tilt angle, as seen in the free body diagrams above, and can be intuitively represented by applying proportional forces to the test subject. The team also briefly considered less intuitive forms of tactile cueing, such as variable vibration levels that could indicate roll angle, but ultimately decided against in depth research since they do not fulfill the customer specified Design Requirement 1.1.

3.1.1 Actuator Tactile Cueing

The first design considered for the TCS cueing consists of actuators attached to pressure plates at various points on the body. This TCS system includes the base chair with various servos attached at desired cueing spots with respect to the test subject. These actuators could be either linear servos or rotational servos. These actuators are attached to smaller flat plates that apply pressure to the test subject through actuation. The positive traits of this design option are that the team has considerable experience in working with actuators, including calibration and integration with software. Also, this design option is not a new concept as massage chairs are similar to this concept, which gives the team a technological foundation to build upon. However, there are also negative traits of this design option that dissuade the team from moving forward with it. These include a potentially more difficult manufacturing process in the attempt to attach the servos to the chair and the pressure plates due to the large number of connections. Similarly, this design may also be less durable due to the amount of attachments that would have to be made, thus increasing the risk of damage through testing. The cost associated with this design also poses some problems. Because this system requires more parts than some alternative designs, it could push the team closer to the budget cap in order to achieve the desired sensitivity and functionality. Finally, this design option may make it more difficult to achieve larger pressures on the test subject due to the limited range and "strength" of certain actuators that are susceptible to jamming at higher pressures.

A preliminary design for an actuator cued TCS is shown in Figure 6 below. This design features six seat plates, six back plates, and two side plates. This chair would be either outfitted using a preexisting chair or a new chair that the team would build to specific dimensions.

Figure 6: Preliminary Model of Actuator Cued TCS

During initial analysis of this design option, the most important element to consider is the actuators, which would be applying the force to the pressure plates on the test subject. As stated in requirement 2.1, the maximum pressure which shall be applied to the test subject is 25 kPa or approximately 3.6 psi. Assuming an average pressure plate size of approximately 6" by 6", the maximum force the actuator must be capable of applying is 130 lb. A common linear actuator from Firgelli Automations, shown in Figure 7 below, is capable of applying a dynamic force 150 lb at a rate of 1 in/s. This actuator can also hold a static load of 300 lb, well above the 50th percentile weight design requirement on the test subject. These servos cost approximately $130 each, so if the team were to have 14 pressure plates, the cost would be upwards of $1800 for actuators alone, not including extra parts.

Figure 7: Firgelli Automations Linear Actuator

3.1.2 Pressure Bladder Tactile Cueing

The next design option considered for TCS cueing is a pressure bladder array controlled by a compressor. This design controls the pressure felt by the test subject by having separate hoses to independently inflate and deflate each bladder. Individual pressure bladders will be made to specific sizes and placed at every cueing spot on the test subject. A positive trait associated with this design option is mechanical simplicity because there are no mechanisms included in the system, aiding in the manufacturability. Pressure bladders also allow the team to apply a “nominal” pressure upon the test subject at the upright position so that some bladders may also be deflated to aid cueing, which would be more difficult to achieve with other design options. A drawback of the pressure bladder cueing is a more difficult configuration and calibration process between the software and the pressure bladders. The team must create their own transfer function between electrical commands to the compressor and desired cueing pressures on the test subject. Furthermore, pressure gauges are needed in the system to monitor the bladders and for testing purposes, which imposes additional cost. This design option may also make it more difficult for the team to meet lag time requirements between the test subject joystick input and actuation output depending on the strength of the compressor. Finally, team members have the least experience with the aspects of this design option including the compressor and bladder construction.

After researching similar concepts in existing products, one which looks like what the team was envisioning is an inflatable seat cover for motorcycle seats produced by Harley Davidson, shown in Figure 8 below. This product encapsulates the team’s idea of using small separated pressure bladders at specific areas on the test subject. This concept would be extended to cover an entire chair, rather than just the bottom, demonstrated in Figure 9.

Figure 8: Harley Davidson Motorcycle Seat Pad

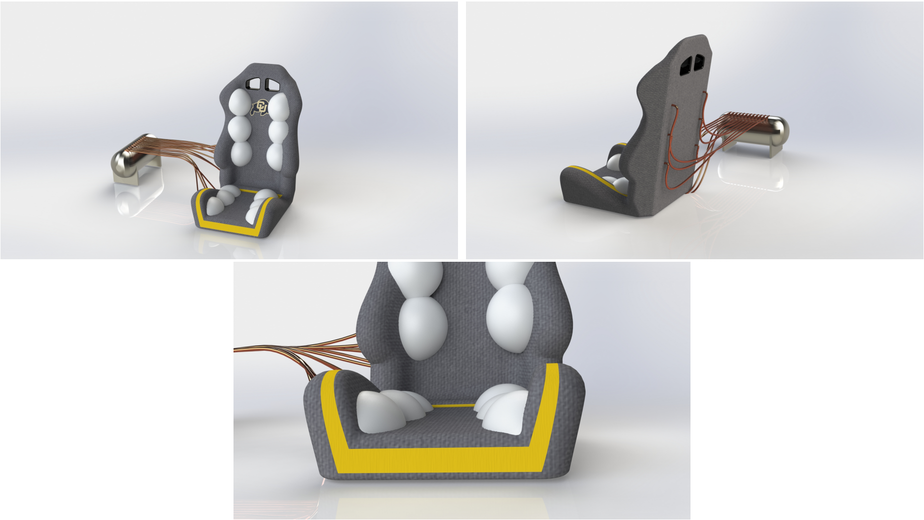

Figure 9: Preliminary Model of Bladder Tactile Cueing

This product also includes an integrated hand pump and metered air release valve to control the amount of air, whereas the team's design would include an air compressor for these functions. This leads the team to the next significant factor of this design option which is the choice of compressor. For preliminary design purposes, the pressure bladders which would put the greatest load on the compressor would be the seat pads. These seat pads would have a nominal load of a 50th percentile male which corresponds to approximately 150 lb, and continuing with the assuming of approximately 6" by 6" sized pressure areas, this corresponds to a nominal pressure of 4 psi. Assuming a nominal pad thickness of about 3 inches, to raise the test subject by 2 inches for cueing would take approximately 6.5 seconds at the max performance of the compressor, and could support 20 such pressurizing maneuvers.

3.2 GVS Alternatives

3.2.1 SOTERIX Galvanic Vestibulation Stimulator

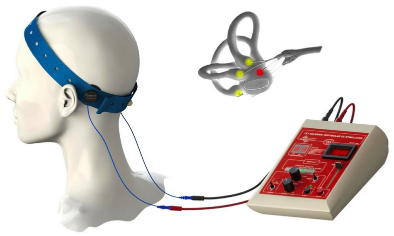

Figure 10: SOTERIX GVS Setup

The SOTERIX is the only commercial off the shelf (COTS) product that is designed expressly for GVS that the team has been able to identify. An image of this product is shown in Figure 10 to the left. Picking this design would greatly simplify the implementation of the GVS technology. It also comes with a multitude of features out of the box, such as DC and Sine waveforms, white noise, and others. The system can adapt to changes in impedance to deliver a controlled, set current, which is an expected challenge that all other solutions will need to overcome. The customer also has significant heritage with this product. However, this system comes with its own controller, electronics, and firmware. This eliminates the chance for digital control via the team's CPS without significant hacking into the system, the viability of which is not known without access to the hardware. This makes achieving the top-level project goals with this product a major risk. A summary of these assessments is shown in Table 2 below.

Pros	**Cons**
COTS Product	Uses a controller not designed for communication with a computer
Multiple waveform options	Proprietary firmware
Adaptive current delivery based on changing impedence	Expected cost of $7000
Customer is familiar with product	

Table 2: SOTERIX Pros & Cons

3.2.2 Custom GVS, Two electrodes

Rather than worry about hacking into a COTS product, another option is to design a GVS ourselves. This gives us full control over the system, but represents a significant increase in project complexity. A review of the literature shows that researchers in GVS have begun experimenting with multi-electrode setups. This is now technically out of scope for our project, so our initial assessment will focus on using the typical two electrode configuration. This is a beneficial design as the vast majority of research in GVS has focused on mastoid-mastoid stimulation, and many researchers have designed their own system to study the effects of GVS using a custom two electrode system. By building a GVS from the ground up, we will have complete control over the internal logic and electrical framework of the GVS, including the ability to interface the system with the CPS. A schematic of this system is shown in Figure 11 below along with a summary of these assessments in Table 3.

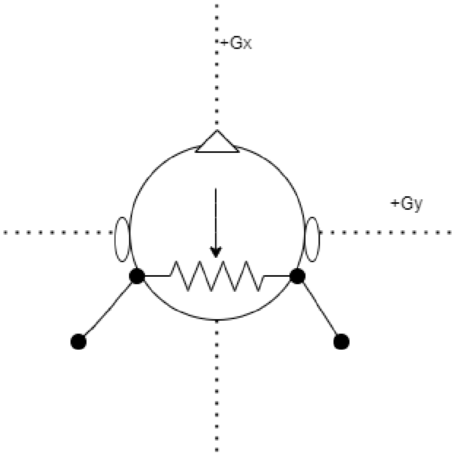

Figure 11: Diagram of custom 2 electrode design. The head is drawn as a variable resistor, connecting to two nodes at the mastoid processes

Pros	Cons
Two-electrode configuration is the most commonly studied and implemented form of GVS	Significant increase in project complexity relative to a COTS solution
Total control over the firmware & software	
Easily integrated with CHAIR software	

Table 3: Custom GVS, Two electrodes Pros & Cons

3.2.3 Custom GVS, Multiple electrodes (Mayo Clinic Configuration)

As a reach goal, our project may want to incorporate an expanded configuration of electrodes. This would reflect the results of two papers, one by the Mayo clinic and one by a team of Japanese researchers. The Mayo clinic authors have reported that their configuration allowed for the simultaneous cueing of pitch, roll, and yaw. Their paper also features a control matrix they derived to translate desired rotations into electrode stimulations which should result in those perceptions. However, a proprietary company was working closely with the Mayo clinic with the desire to turn their research into a commercial product, so it is unknown how much information was withheld due for intellectual property reasons. As such, our ability to replicate their success is not known at this time. A summary of these assessments is shown in Table 4 below along with a basic configuration diagrams from the Mayo Clinic in Figure 12.

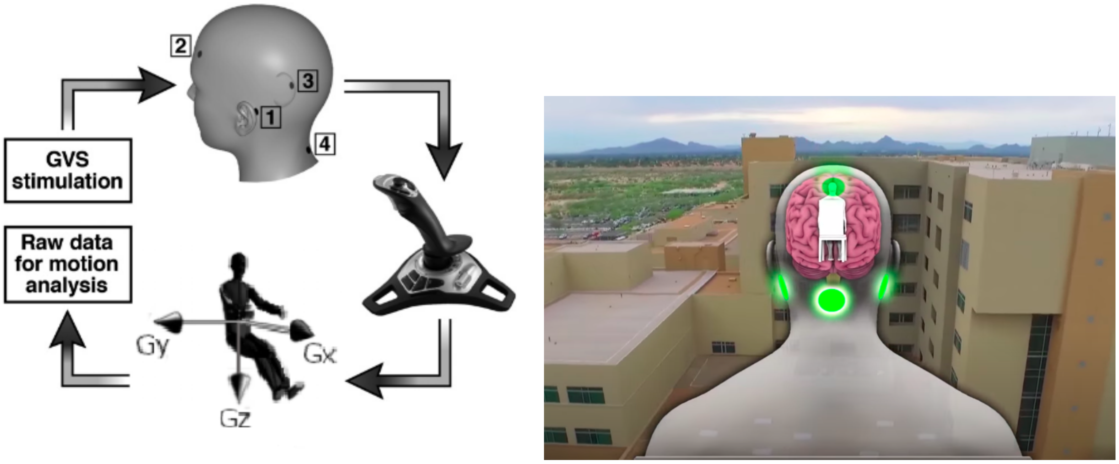

Figure 12: Mayo Clinic Study GVS Configuration

Pros	Cons
Transformation matrix derived by researchers to map desired rotational velocity perception to electrode current	Unknown replicability
Reported ability to simultaneously cue rotational velocity in roll, pitch, and yaw	Unknown if paper is leaving out essential components to success (product development teams are working closely with researchers)
	This implementation of GVS is still in development.

Table 4: Mayo Clinic configuration Pros & Cons

3.2.4 Custom GVS, Multiple electrodes (Japanese Study Configuration)

Similarly to the Mayo Clinic, another group of researchers has identified another arrangement of electrodes which they claim isolates cueing the three traditional roll, pitch, and yaw axes. However, this group did not generate a transformation matrix to cue for a combination of rotations simultaneously. This means that our cueing would be limited to individual maneuvers and eliminate combined maneuvers (e.g. a banked turn). While the researchers do not identify any companies that they are working with, it remains true that this is an unproven, experimental version of GVS. It has also fallen out of the scope of our project, given that we are focused on roll cueing. A summary of these assessments is shown in Table 5 below along with a basic configuration diagrams from the study in Figure 13.

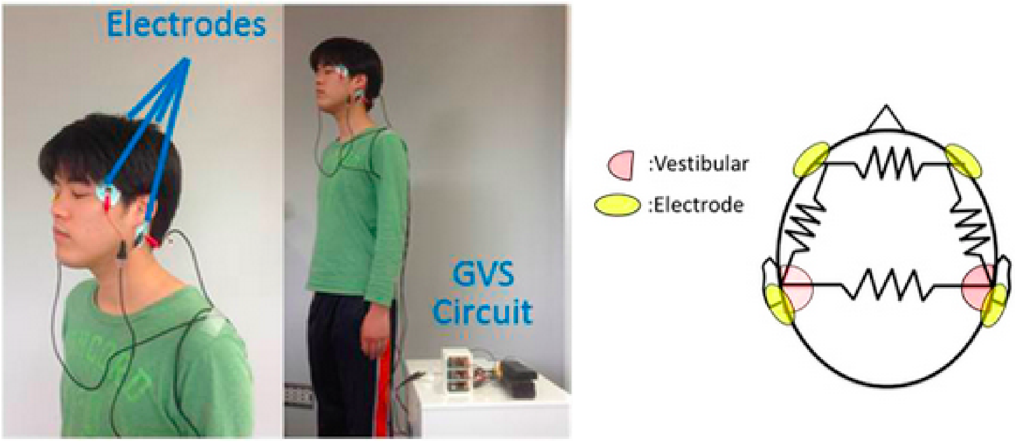

Figure 13: Japanese Study GVS Configuration

Pros	Cons
Easier implementation of multi-axis cueing than the Mayo Clinic configuration	Unknown replicability
	Cueing limited to a single axis at a time
	No transformation matrix derived
	Highly experimental implementation of GVS

Table 5: Japanese Researchers configuration Pros & Cons

3.2.5 Magnetic Vestibular Stimulation

More of a laboratory curiosity, it has been noted that strong magnetic fields can induce vestibular perceptions by inducing forces on the ionic fluid inside the semicircular canals. So far, this has been studied only in the presence of enormously powerful magnets, particularly those within MRI machines, an example of which is pictured in Figure 14. These magnets have field strengths in the Teslas and are prohibitive on both cost, size, and maintenance. One major difference that makes this system stand out, is its ability to stimulate a constant angular acceleration. However, that is beyond the scope of the class. Nonetheless, it is an extremely novel method of cueing for roll, and may be a more viable option in the future. A summary of these assessments is shown in Table 6 below.

Figure 14: Example of an MRI Machine capable of stimulating vestibular perceptions

Pros	**Cons**
Novel approach to stimulation of angular displacement	Development from scientific curiosity into viable technology has not happened
	Requires a magnetic field of several Teslas to induce perception
	Technology has not been used to cue for specific rotations

Table 6: Magnetic Vestibular Stimulation Pros & Cons

3.3 CPS Language Alternative Designs

This project is heavily dependant on the underlying software, and therefore will require very careful consideration of what programming language or languages to use for the CPS. A wide variety of languages are considered below.

3.3.1 C

C has the advantage of being a very versatile language with excellent cross-platform compatibility. This language is the lowest-level language being considered for this project, and as such choosing to opt for programming in C directly results in a higher performance than a high-level language might support. C is the basis of most modern operating systems (including Windows and Mac iOS) and is often the language of choice for programming hardware. Due to its versatility and inherent portability, the use of C would make interfacing between multiple types of hardware very simple. However, C has several significant drawbacks in terms of its use for this project. The group generally lacks knowledge of the language, and C is known to have a steep learning curve; both of these considerations could result in significant time lost if C is chosen as the primary CPS language. Finally, C does not include support for object oriented programming (OOP) which is widely considered to be the nexus of modern programming and software development. A summary of these assessments is shown in Table 7 below.

Pros	Cons
Does not require user memory management	Minimal group knowledge
Excellent computing and runtime performance	Considered difficult to debug
Cross platform portability	Lacks support for object-oriented programming
Close to hardware language	Few built-in functions and libraries
Popular in microcontrollers	Very difficult to create GUI or UI applications

Table 7: C Pros & Cons

3.3.2 C++

C++ is one of the most widely used modern programming languages, and for good reason. The language is extremely versatile and, while slower than C, still boasts very rapid computation and execution speeds. Like many modern languages, it offers a good amount of support for object-oriented programming and includes many built-in libraries containing high-level data types and their associated functions. C++ is a popular language both for creating graphic applications and for interfacing with C (used by many microcontrollers) which makes it ideal for the needs of the CPS system. Additionally, C++ gives its user a great deal of control over memory allocation and data and/or memory management. While this can be a good thing, an inexperienced programmer may find that it causes issues, as C++ does not have inherent garbage collection. Additionally, C++ is often considered to have stricter and more complicated syntax than higher-level languages such as Java and Python. Along these same lines, C++ debugging can be somewhat complicated, as the innate debugging capabilities of the language are somewhat lacking. Despite its excellent computation power, the potential issues with C++ may be compounded for this project since only a few group members know the language. A summary of these assessments is shown in Table 8 below.

Pros	Cons
Excellent run time speeds	Requires user memory management
Supported on many machines	Can be more difficult to debug than higher-level languages
Built-in support for OOP	Strict, sometimes complicated syntax
Offers low-level control over memory	Less group experience

Table 8: C++ Pros & Cons

3.3.3 Java

Java is one of the most popular languages in the software development world, in part due to its extensive portability. This language is also very easy to learn and contains many features that support object-oriented programming. This makes it a good choice for large software development tasks, as there is extensive innate class and package organization. Depending on the IDE used by the developer, debugging in Java is often quite easy as well. This language is also widely considered a good option for application and GUI development, and these tasks are fairly simple when using this language. However, Java is generally slower than both C++ and C. Additionally, it does not have good compatibility with hardware or other languages, which could prove a major issue given the requirements of the CPS. Like other very high-level languages, Java also does not offer significant control over memory management. A summary of these assessments is shown in Table 9 below.

Pros	Cons
Highly portable	Slower than some lower level languages
Extensive open-source libraries and tutorials	Limited control over memory
Extensive structuring and support for OOP	Difficult to interface with other languages
Simple GUI and application creations	

Table 9: Java Pros & Cons

3.3.4 Python

Python is a widely used programming language in software development, having the most pull requests of any coding language on Github. Similar to Matlab, is has a large library of built in libraries which aid the developer in a number of ways during development. Python is capable of Rapid Application Development since there is no compilation, so the software development cycle is very fast. One of the most useful features of Python is the ease at which the developer can move through the debugging process using the many built in adaptations. Python is said to be useful for small and large scale projects using its object-oriented approach. However, Python is slower than other common languages such as C and C++ and also consumes more memory. Python users also report a limited database access and a few common runtime errors. A summary of these assessments is shown in Table 10 below.

Pros	Cons
Considered easy to learn	Typically slow to execute
Simple syntax	Limited ability to interface with hardware
Many built-in libraries available	

Table 10: Python Pros & Cons

3.3.5 Matlab

Matlab is by far the most familiar programming language among team members. It is widely used in the industry, and the various toolboxes and GUI/simulation capabilities would be valuable during the testing process. Its simplicity and familiarity amongst team members, along with built-in debugging tools, could help speed up the design process. Additionally, Matlab includes built-in functions to translate scripts into C or C++, which may allow for easy interfacing with hardware. Despite these advantages, Matlab has the disadvantage of slow computation and execution times. Additionally, it is a very high-level language and does not afford the same design control that lower-level languages such as C++ offer. Support for object-oriented code structures does exist but is generally poor in quality, and interfacing with hardware is generally difficult. Additionally, one of the project software leads has experience with Matlab's C and C++ interpretation capabilities and found them to contain major bugs that may crash the application. Matlab licenses are also very expensive, with additional cost for any toolboxes used, which could prove to be an issue for the customer. Finally, Matlab is a very high-level language, and while this does make it easy to learn and debug, it also means that the user has limited control over certain functionality such as memory management. A summary of these assessments is shown in Table 11 below.

Pros	Cons
Well-known to all group members	Lacks portability
Extensive libraries and documentation	Difficult to interface with hardware
Built-in interfacing with C language	Licenses are expensive
	Lacks backwards compatibility between software versions
	Offers very little control over memory and performance

Table 11: Matlab Pros & Cons

3.3.6 LabVIEW

LabVIEW provides features that could be valuable for hardware testing. Even though the team is largely unfamiliar with the language, basic utilization is relatively straightforward even with no prior programming knowledge. LabVIEW is able to communicate with C++ and C as well, which would make hardware integration and communication much smoother. Because of the simplicity of the language, LabVIEW code building can be more arduous, and it usually takes longer to code in LabVIEW than in Java, C, or C++. A summary of these assessments is shown in Table 12 below.

Pros	Cons
Basic utilization is easy without any programming knowledge	Building code directly in Labview usually takes longer than Java, C, or C++
Has capability to interface with other languages (C, C++)	Unfamiliar to most team members
High compilation speeds	May not be able to interface with all hardware
Very easy for customer to use and edit code	Does not offer extensive control over outputs

Table 12: LabVIEW Pros & Cons

3.4 CPS Architecture Alternative Designs

The CPS Architecture refers to the general electronics setup required to send a process all the commands and data required by the GVS and TCS.

3.4.1 Single Computer Setup

A single computer setup would be a simple solution to the interfacing of the CPS system. In this design, a computer will run a yet to be determined software program that will control both the GVS as well as the TCS. To enable such a design, the TCS will have to be provided with it's own on board power system. This allows the actuators to be on standby, awaiting commands, from the central computer. Also, the GVS only requires micro-amps of current to operate and this power can be provided by the central computer. Lag time would be easier to control with only one central processor to worry about. An external device, the computer, would be required to operate CHAIR. A basic diagram of this setup is shown in Figure 15 below.

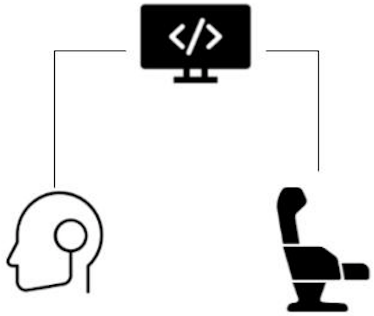

Figure 15: Single Computer Setup Diagram

3.4.2 Standalone Processing Unit

A similar design would require the creation of a standalone controller. This would require it's own power system, processor, circuitry, memory, and on board programming similar to a single-board computer. The CPS would be a single box and interface directly to the GVS and TCS. Switches and dials could be implemented, but the system would not interface like a traditional computer. Also, a memory flash would need to be performed to drastically update the control profiles for the TCS and GVS. A basic diagram of this setup is shown in Figure 16 below.

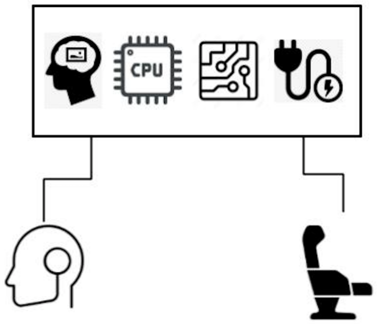

Figure 16: Standalone Processing Unit Setup Diagram

3.4.3 Central CPS, One Shared Microcontroller for TCS and GVS

Under this system, the CPS would run on a computer (desktop, laptop, etc.) and handle all computation of inputs and translating them to the desired outputs. This information will then be sent to a single microcontroller which then informs TCS and GVS how to produce cues that match the desired output from the CPS. A benefit for this system is its ability to integrate different languages to be run on the CPS and microcontroller, so long as they are compatible. This is particularly pertinent due to the customer's desire for the controller to have access to a GUI to use during testing. This system would be need to run a more GUI friendly language on the CPS and a computation friendly software language on the microcontroller. A basic diagram of this setup is shown in Figure 17 below.

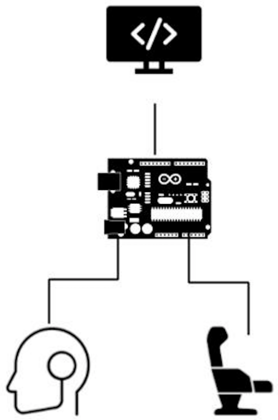

Figure 17: Central CPS, One Shared Microcontroller Setup Diagram

3.4.4 Central CPS, Two Separate Microcontrollers for TCS and GVS

This system features the CPS running on a computer much like in the previously discussed design. However, this design would dedicate two individual microcontrollers for both the TCS and GVS. The CPS will once again handle the majority of the computation but will communicate only the relevant information to the respective microcontrollers. A benefit to this design is that it allows each microcontroller to be ran independently, thus allowing the TCS and GVS themselves to act independently. However, independence of the two microcontrollers introduces the risk of latency issues. The latency between the TCS and GVS systems must be sufficiently small in order to provide a believable experience to the test subject. Using this system would require careful consideration of the latency caused by each microcontroller. A basic diagram of this setup is shown in Figure 18 below.

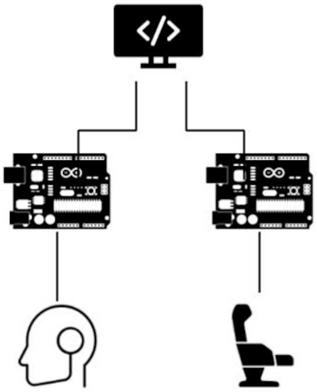

Figure 18: Central CPS, Two Separate Microcontrollers Setup Diagram

4 Trade Studies

4.1 TCS Design Options

To select a design for the TCS, the team weighed several areas of importance to inform the decision process. Firstly, the most heavily weighted characteristic for the TCS design was chosen to be functionality. Fulfilling the design requirements is of highest priority and any design that does not properly address them is not viable. The next most heavily weighted criteria is the manufacturability of the system. The team has limited time and resources available to construct a viable system and must also consider the possibility of COVID-19 restrictions further limiting manufacturing capabilities. Design options which are particularly difficult to manufacture or require tools and facilities that the team does not have access to would make a design infeasible. The next two qualities that the team deemed to be of equal importance are design complexity and cost. Similar to manufacturability, the team has a limited amount of time to design a functioning system. Designs that are significantly beyond the team's expertise level will require more time to form a complete design. The team is also provided with a budget of 5,000 USD, and thus the cost of parts for the system must not exceed this budget, unless alternative funding is obtained. Finally, the least influential qualities that the team are considering for TCS trade studies are number of failure points and safety. By nature, more complex designs have a larger number of possible failure points in the system, thus giving a higher probability that the system could break or fail during use. These failure points are typically known and can be addressed by a proper design. Safety is of utmost importance to the team throughout design, manufacturing, and testing. The team will ensure that no team member or test subject is ever put in an unsafe environment. However, neither of the TCS design options are expected to pose any serious safety hazards and therefore safety is weighted low for decision making purposes while remaining a priority during the design process. A summary of these weighting criteria are shown in Table 13 below.

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 13: TCS Design Option Weighting Criteria

To properly score the design options using the weighted criteria, the team developed a rubric for scores ranging from 1 to 5 for all criteria. A 1 on this scale represents the design being impossible, improbable, or entirely unacceptable while a 5 represents a trivial or perfect solution. For criteria such as the number of failure points and cost, the team imposed a range of numbers for each score. For the other criteria, the team created a qualitative scale which gradually increases in success for each score. A detailed chart which describes the scoring requirements for each criteria is shown in Table 14 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 14: TCS Scoring Rubric

Finally, the actuator driven and pressure bladder driven design options were scored for each of the criteria in Table 13 using the rubric in Table 14. For the number of failure points, the pressure bladder option scored higher than the actuator option. The team deemed that the connection to the compressor and the pressure bladder popping were the only failure points for the pressure bladder, while each individual actuator in the actuator option has the potential to jam. The pressure bladder also scored higher in design complexity as there would be less connections both physically and electrically for this option. The actuator driven design received the highest score for functionality as it would theoretically be able to meet all of the design requirements, while the pressure bladder design may be difficult to control the direction of the force applied. The actuator option scored higher in the safety criteria because the actuators being used would not feasibly be able to harm the test subject as much as a popped pressure bladder would. The actuator option scored higher in the manufacturability criteria due to prior team experience using actuators versus a virtual nonexistent experience with compressors and inflatables that the pressure bladder design entails. Finally, the pressure bladder option scored higher for the cost criteria as the cost of a single actuator, the most expensive product for the actuator option, costs the same as a compressor, the most expensive product for the pressure bladder option. However, the actuator design requires at least eight actuators while the pressure bladder option would only require one or two compressors, making the actuator design significantly more expensive, although both should feasibly work with the given budget. After summing the product of each designs score and weight for each criteria, the actuator option scored higher and proved to be the more feasible option. The individual scores are summarized in Table 15 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 15: TCS Design Option Trade Study

4.2 GVS Trade Study

The most heavily weighted considerations for the GVS trade study were technology readiness and integration potential. There are multiple GVS approaches which have been studied in the literature, but not all of them are equally prepared to be implemented into an integrate cueing system. If we were to select a model of GVS cueing that is proposed in the literature which did not behave as the researchers claim, the GVS subsystem would be a failure. Similarly, if an approach to GVS is not able to be digitally controlled, it will exceptionally difficult to integrate into the larger CHAIR architecture. Cost and safety are both the smaller weighted considerations. Cost was weighted small because our GVS components are unlikely to be exorbitantly expensive. Microcontrollers, ICs, DAQs, and most other components that will go into our GVS system have typical costs at less than $100 dollars, even less for ICs. However, certain COTS GVS solutions are out of budget and will need to be considered. Any GVS system can have ample safety elements built into the system in the form of software limits, current limiters, and fuses to control how much power is being delivered to the subject. As such, safety is not weighted as heavily technology readiness and ease of integration. A summary of these scoring criteria is shown in Table 16 below.

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 has good achievability.
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 16: GVS Design Option Weighting Criteria

Four levels of scoring were allotted to each category. For technology readiness, our team determined how many publications had been made using a given approach and whether or not any COTS implementations for the approach existed, as this would be a strong indicator that the effect of cueing could be reliably reproduced across a wide range of users. Next, we considered cost. Though the actual cost will depend on the actual nature of the GVS implementation, estimations were made based on the customer's history with working on GVS systems and from the expected number and grade of components that we would need to purchase. For safety, we considered how many safety mechanisms were in place, or estimated where appropriate. For the custom methods, it was assumed that we would be implementing the safety mechanisms ourselves. For ease of integration, we weighted based on how many communication protocols could be used based on the intrinsic nature of the system itself versus any actual microcontroller. For instance, the magnetic field approach necessitates optical communication, as the strong fields would interfere with traditional electrical wires. Similarly, some COTS solutions do not readily provide the user with the ability to interface with a computer. A summary of these scoring requirements is shown in Table 17 below.

Criteria	**Scoring Requirements**			
	1	2	3	4
Technology Readiness	This approach has had one or zero publications	This approach has had multiple publications	COTS solutions are in development for this approach	COTS solutions exist for this approach
Cost	\$1,500+	\$1,000-\$1,499	\$500-\$999	\$0-\$499
Safety	1-failure safety is easy to implement	1-failure safety is already implemented	2-failure safety is easy to implement	2-failure safety is already implemented
Ease of Integration	The system cannot be reasonably expected to receive external communication	The system must be modified to incorporate a communication chip	The system can use a common communication protocol	Any communication protocol can be used without needing additional modification

Table 17: GVS Scoring Rubric

Finally, our scores were calculated and tabulated in Table 18 below. The highest levels of technology readiness were given to the SOTERIX and custom 2-electrode GVS system. This is because COTS solutions exist for the approach of roll-only 2-electrode cueing. Because products are currently in development for the Mayo Clinic approach, this was scored a 3. The remaining methods have little to no research and were left as a 1. All 3 custom solutions were approximated to have a cost in the \$500-\$999 range, which is perhaps an overestimation. The other two solutions are prohibitively expensive for our project without external funding. For safety, it was assumed that the SOTERIX has at least two forms of safety embedded within. The custom solutions, as discussed previously, could have multiple-fault tolerances built in without too much effort. Finally, the SOTERIX received a low score for ease of integration due to the fact that it is a closed system. We would need to hack into the hardware in order to directly control the system digitally. Similarly, the magnetic method would require special communication algorithms to penetrate the enormous magnetic field. The custom solutions, as we will be picking our own hardware, allow us to consider virtually any communication protocol we might wish to use.

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 18: GVS Design Option Trade Study

4.3 Computer Language Trade Study

The criteria that is most heavily weighted for computer language trade studies is the language speed. Reducing lag times between commands and cues is essential for creating a believable experience for the test subject, requiring tight time constants on the lag time between input from the joystick to the output from the GVS and TCS. The other criterion having a major impact on software choice is the team's experience with a given language. The team has a limited timeline for developing the software and selecting a language that the team has more experience with will allow for a shorter development time. Ease of writing and debugging goes hand in hand with team experience in terms of reducing software development time and ensuring that all deadlines can be met. There are also criteria relating to the ease with which language can interact with micro-controllers and the UI. These criteria are summarized in Table 19 below.

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 19: CPS Language Option Weighting Criteria

To aid the scoring of each language against the design criteria, a scoring rubric has been created. This rubric quantitatively defines the scores for the team experience by number of members who have experience with the language. Languages are considered to be more desirable if a large number of team members have experience with the language, and extra consideration is given to the knowledge of the three software leads. A team poll was taken to determine the number of team members with experience in each language. Quantitative determination of which language was fastest was based on a 2018 publication from the University of Pennsylvania[12]. This study assigned relative computation times to a large number of computer languages based on the average time taken to complete a set of computing tasks. It is notable that LabVIEW was not included in this study. Therefore, LabVIEW's score was determined based on information from the National Instruments website, which compared the speed of LabVIEW to C++ [13]. Ability to interface with controllers was scored based on several language attributes. The presence of user control over pointers, historical use for the purpose of hardware control, and language support for hardware control are all considered in this section. Language support is defined by the availability of relevant built-in constructs within the language structure, as well as the quality and extent of documentation of any functions that exist. Quantification of debugging capability is defined based on language and IDE support for debugging, as well as robustness of error messages, availability, and whether the language is compiled or interpreted. This last criteria is relevant because an interpreted language will only return one bug at a time, and only for the functions being run in a given script. A compiled language can be debugged more quickly because a large number of bugs from all program files can be simultaneously discovered and fixed. A summary of these scoring requirements is shown in Table 20 below.

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 20: CPS Language Scoring Rubric

Finally, taking the scoring requirements into consideration, each language was scored and the totals were tallied. Following scoring, it became clear that Java, MATLAB and LabVIEW fell short, particularly in the areas of team experience, ability to interface with hardware, and speed. These shortfalls essentially eliminated those languages from consideration. Python, C and C++ all finished with higher scores, with C and C++ being nearly tied for the highest. Ultimately, however, the superior language support in C++ resulted in this language receiving the highest score. A summary of these scores is shown in Table 21 below.

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 21: CPS Language Option Trade Study

4.4 Computer Architecture Trade Study

The final aspect of the project that the team chose to trade study was the computing and control architecture between the CPS and GVS/TCS. The most important criteria for a computer architecture was decided to be the design complexity and subsystem independence. The subsystem independence is an important criteria of the computer architecture as the TCS and GVS systems will be dealt with essentially independently for the initial development process, so the team must have the ability to work them separately for maximum efficiency. The final two least weighted criteria are the cost and user convenience of the computer architecture. A summary of these weighting criteria is shown in Table 22 below.

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 22: Computer Architecture Weighting Criteria

The separate weighting of each criteria is now assigned scoring brackets from one to four. Design complexity is mostly a function of how much of the design can achieved using COTS solutions. The more custom fabrication required means more complexity in the design. The cost was scaled based on the total budget of five thousand dollars. Spending more than one fifth of the total budget on the CPS is considered undesirable. User convenience ranks the ease of manipulation the controller has over how the CPS functionality. Lastly, subsystem independence is an important score based on the ability for the CPS architecture design to be developed on multiple subsystems. This allows greater tolerance. If troubles arise in a singular subsystem an architecture with high subsystem independence would ensure no other subsystems are interfered with issues from another aspect of the project. Figure 23 shows the score distributions.

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 23: Computer Architecture Scoring Rubric

Each architecture design was scored against each other. The weights of each criteria is then used to create a final score from one to four for each possible configuration. All options but the standalone system rank high, at around a three. The single computer system scores high but has potential to bottleneck the whole project if issues occur in development. The combination of a computer and micro controllers wins out. With the flexibility of having two micro controllers winning the top score. This is documented below in Table 24.

Criteria	**Weight**	**Scoring**			
		Single Computer	Custom Standalone 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 $= \Sigma$ (Scores*Weights)		**3.1**	**1.8**	**2.9**	**3.3**

Table 24: Computer Architecture Trade Study

5 Baseline Selection

Now that all aspects of this project have been considered and ranked an overall design choice must be made.

5.1 TCS Design Selection

As a result of the trade studies of the TCS system, the actuator design was deemed the best to fit the design requirements, mostly due to the guaranteed fulfillment of all design requirements and manufacturability due to team experience with actuator components. Also, although this design does come at increased cost, the team is fairly certain that the budget will still be able to accomodate this. Thus, the team will move ahead with an actuator driven design, with additional trades studies being conducted in the future for each particular aspect of this design such as actuator selection and chair build versus buy.

5.2 GVS Design Selection

Based on our preliminary review of the GVS systems, the Custom 2-electrode setup is the clear winner. The technology readiness was one of the most important considerations of this project, which restricted our solution to COTS 2-electrode and custom 2-electrode elements. As the SOTERIX GVS could not be easily integrated into the central processing system and was outside of our project budget, the custom 2-electrode system is the best configuration for achieving project success.

5.3 CPS Language Selection

After reviewing six different computing language options, the two languages that proved most effective for this project were C and C++. Both languages offer very similar functionality and compatibility between devices. Additionally, these languages ranked highest among the two categories weighted most significantly, speed and team experience. While the two languages scored almost equally across all categories, C++ ultimately performed better in the trade study, and as such this language was selected for the CPS. Since C++ and C are nearly identical in many respects, the team may decide to use C for some hardware considerations. This would be a reasonable choice, as it should be very easy to integrate C with C++ and C often has better access to hardware components such as microcontrollers.

5.4 Computer Architecture Selection

After reviewing and assessing the characteristics of each architecture, the trade studies showed that the setup of one central CPS and two separate microcontrollers for the TCS and GVS each. This is mostly due to the fact that this configuration allows independent development and testing of the TCS and GVS to better allocate the team resources. This configuration also lends well to the customer requirements in that it may easily be operated with a GUI by a controller. Also, the design complexity and cost were reasonable in that this design only requires assembling a few different COTS products which aren't particularly expensive. Ultimately, this design is best to fit the group's skill set and fast-paced schedule.

5.5 Baseline Design Selection

The trade study process has helped the team fine-tune necessary design requirements in order to finalize a baseline design for this project. The CHAIR project will consist of two major elements: the Tactile Cueing System (TCS) and the Galvanic Vestibular System (GVS). The TCS will consist of several actuators to provide physical pressure cueing to the test subject. This design allows for increased functionality and ease of integration with the computing architecture. The baseline design will also consist of a GVS with a custom 2-electrode setup due to its ease of integration with the remaining design elements. Both the TCS and the GVS will be controlled by one central CPS with two separate microcontrollers for the TCS and GVS respectively. Programming the microcontrollers for the TCS and GVS will be done with either C++ or C, largely depending on which microcontroller will integrate best as the team furthers the design process. However, the main CPS will certainly be programmed in C++.

References

[1] “Electric Shock.” Electric Shock Hazards, hyperphysics.phy-astr.gsu.edu/hbase/electric/shock.html.

[2] 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

[3] Adami, M., & Seibel, A. (2018, December 23). On-Board Pneumatic Pressure Generation Methods for Soft Robotics Applications.

[4] St George, R. J., Day, B. L., & Fitzpatrick, R. C. (2011). Adaptation of vestibular signals for self-motion perception. The Journal of physiology, 589(Pt 4), 843–853

[5] 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

[6] Swearingen, J. J., "Determination of Centers of Gravity of Man," Federal Aviation Agency, August 1962.

[7] Christensen, J. M., McBarron, J. W., McConville, J. T., Pogue, W. R., Williges, R. C., and Woodson, W. E., "Man-Systems Integration Standards," NASA-STD-3000, Vol.1, 1995

[8] Ward, Bryan K., et al. “A Decade of Magnetic Vestibular Stimulation: from Serendipity to Physics to the Clinic.” Journal of Neurophysiology, 23 May 2019, journals.physiology.org/doi/full/10.1152/jn.00873.2018.

[9] Aoyama, Kazuma, et al. “Four-Pole Galvanic Vestibular Stimulation Causes Body Sway about Three Axes.” Scientific Reports, Nature Publishing Group, 11 May 2015, www.ncbi.nlm.nih.gov/pmc/articles/PMC4426693/.

[10] Cevette, Michael, et al. Oculo-Vestibular Recoupling Using Galvanic Vestibular Stimulation to Mitigate Simulator Sickness. www.vmocion.com/docs/Oculo-Vestibular-Recoupling-Using-Galvanic-Vestibular-Stimulation-to-MitigateSimulator-Sickness.pdf.

[11] “Galvanic Vestibular Stimulation.” Galvanic Vestibular – Soterix Medical, soterixmedical.com/research/vestibular.

[12] Aruoba, S. Boragan and Fernandez-Villaverdey, Jesus. "A Comparison of Programming Languages in Economics: An Update." University of Pennsilvania. March 25, 2018.

[13] National Instruments, "Can LabVIEW C?". Updated Nov 7, 2019. https://www.ni.com/en-us/innovations/white-papers/06/can-labview-c-.html