

# Pod I Final Demonstration Documentation

European Hyperloop Competition 2021

University of Toronto Hyperloop Team

March 14, 2021



## **Table of Contents**

<b>Table of Contents</b>	<b>2</b>
<b>Executive Summary</b>	<b>8</b>
<b>Introduction</b>	<b>9</b>
UTHT and Members	9
POD I	9
Safety	10
Awards Per Subsystem	11
<b>Structures</b>	<b>12</b>
Overview of Structures	12
Parts List	12
Full Description of Structures	14
Safety	22
<b>Propulsion</b>	<b>44</b>
Overview of Propulsion	44
Propulsion Components	46
Front and Rear Assemblies Simulations	52
Safety	59
<b>BRAKING</b>	<b>65</b>
General Overview	65
Modes of operation	65
Safety	71
FMEA	72
Safety	86
<b>Energy Systems</b>	<b>89</b>
General Overview	89
High Voltage Battery System (HVB)	92
Motor Inverter System (MIS)	103
LV Battery Module (LVB)	108
LV Power Distribution System	115
Charging System	120
<b>Electronics</b>	<b>126</b>
General Overview	126
Sensor System	127
Low-Level Control System	141

Infrastructure System	147
<b>Software</b>	<b>157</b>
General Overview	157
Communication System	158
High Level Control Systems	164
Infrastructure	168
<b>Macro-Integration of Subsystems</b>	<b>174</b>
Structure System	174
Propulsion System	174
Stability System	174
Braking Subsystem	175
Energy System	175
Electronics System	176
Electrical Systems Integrated Tests	177
<b>Transportation of POD</b>	<b>179</b>
Overview of Carrier	179
Parts List	180
Full description of subsystem separated in sections by major subassemblies	181
Safety	185
<b>Appendix ST - Stability</b>	<b>189</b>
<b>ST1: Assembly drawing of the shock testing device</b>	<b>189</b>
<b>Appendix E - Electronics</b>	<b>190</b>
E1: Primary and Auxiliary Sensors	190
E2: Microcontrollers & Master Computer	198
E3: Complete Wiring Schematics	201
E4: Serial CAN Bus Module Interface with PM100DX	203
E5: PM100DX Inverter Startup Sequence	208
<b>Appendix S - Software</b>	<b>212</b>
<b>References - Software</b>	<b>213</b>

## **Executive Summary**

The University of Toronto Hyperloop Team (UTHT) is a multidisciplinary engineering team whose individuals strive to contribute to the advancement of high-speed transportation. This document presents to a high-technical extent our first-generation pod design. This Final Demonstration Documentation (FDD) shows the pod sub-systems details description and interconnection to the main assembly, safety and operability test at the sub-system and pod level, and pod transportation mechanism. Besides, this pod is contesting for Complete Pod Award, the Electronics in combination with the Software subsystem are contesting for Electrical Subsystem Award, and the Structures subsystem is contesting for the Mechanical Subsystem Award.

Pod I is a fully-autonomous pod propelled by an in-wheel electric drivetrain which allows for a maximum theoretical speed of Pod I maximum dimensions are 2.16 length, 0.66 width, and 0.64 m

## **Introduction**

### **UTHT and Members**

The University of Toronto's Hyperloop Team (UTHT) is an engineering design team consisting of more than 200 members from the various engineering disciplines at the University of Toronto. UTHT's mission is to develop world-class talent at the intersection of business and engineering that contributes to the advancement of high speed transportation - this is what has led it to develop its first pod (Pod I). With funding from the University of Toronto's Faculty of Engineering and industry sponsors, the manufacturing and assembly of Pod I is underway with an expected completion in early 2021. The team responsible for Pod I is listed below.

<b>Team Captain</b>	Juan Egas*
<b>Chief Mechanical Engineer</b>	Hamza Arshad
<b>Propulsion Lead I</b>	Javad Siahkamari
<b>Propulsion Lead II</b>	Hongchen Wang
<b>Braking Lead</b>	Anton Liu
<b>Stability Lead</b>	Fernando Estrada
<b>Structures Lead I</b>	Jovan Phull
<b>Structures Lead II</b>	Phil Cuvin
<b>Chief Electrical Engineer</b>	Anaqi Afendi
<b>Energy Systems Lead I</b>	Jun Seo
<b>Energy Systems Lead II</b>	Anaqi Afendi
<b>Electronics Lead I</b>	Urmilkumar Modi
<b>Electronics Lead II</b>	Orest Cobani
<b>Software Lead</b>	Davendra Seunarine Maharaj

\* Juan Egas will be UTHT's representative in contact with the EHW committee.

**POD I**

Pod's major subsystems and components have all been designed on a CAD platform with some starting the physical machining and sourcing of parts necessary in early 2020. Pod's physical development is occurring mainly on the University of Toronto's campus - this includes manufacturing of small, machinable parts at the university's machine shop. It is expected that subsystems will be able to assemble the pod on campus in early 2021, following COVID-19 guidelines outlined by the University of Toronto's Faculty of Engineering.

Pod I is a precursor to UTHT's Pod II which will utilize a fully functional, non-contacting propulsion system (ie. hovering/contactless). For this reason, Pod I's main objective is to demonstrate a functioning hyperloop pod system with integrated subsystems that can be adopted to a more advanced Pod II - the development of this pod is also underway. Other objectives of Pod I include to:

- understand how to create a better stability system at high speeds by taking sensor measurements of the current system's performance
- create an efficient energy system design and eventually reiterate to create an energy dense system in the next pod
- demonstrate that this technology can be scaled up for human transportation.

UTHT's Pod I can be divided into seven subsystems: structures, propulsion, braking, stability, energy systems, electronics and software. These will be discussed in the sections that follow. Figure I1 illustrates the pod and its associated subsystems.

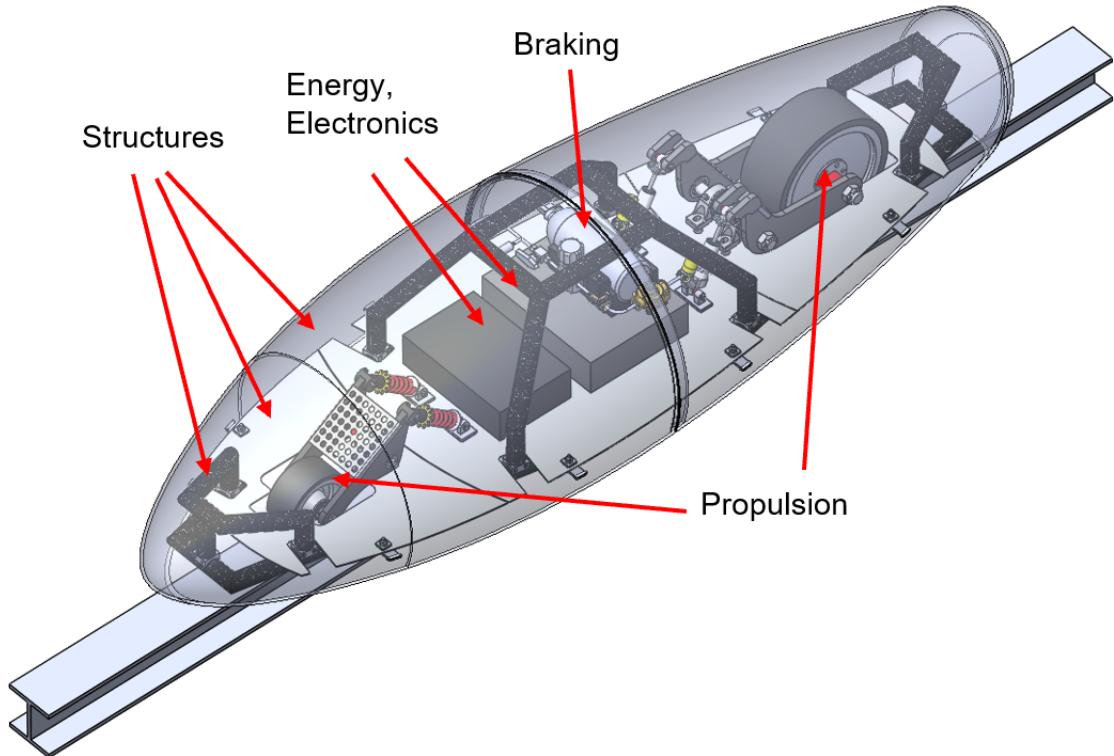


Figure I1: Isometric view of Pod I with subsystems (stability system not shown, located on the bottom of the pod)

## Safety

The main focus of this document is to go over technical details of each subsystem and their eventual role in the overall pod. Additionally, safety is emphasized throughout the document to verify each subsystem's feasibility in a real-world environment. To show this, mechanical subsystems will use a failure modes and effects analysis table (FMEA table) to document major failure modes and the effect they can have on the subsystem and overall pod. Failure mitigation strategies are also described by both mechanical and electrical teams to decrease the chance and/or effect of any recognized failure modes. For the FMEA table, a target risk priority number (RPN) of 30 was used (RPN is calculated by multiplying a scaled 1 - 10 number assigned for severity, frequency of occurrence, and difficulty of detection. The goal for any failure mitigation strategy is to eventually have an RPN for its associated failure below the RPN. 30 is a low number (severity = 5, frequency of occurrence = 3, and difficulty of detection = 2) but is used as the threshold since a large emphasis on safety is stressed for the EHW competition.

## Awards Per Subsystem

Awards the team wants to nominate POD I for are:

- Best mechanical system award

- Best propulsion subsystem award
- Best electrical subsystem award
- Complete pod design award

## **Structures**

### **Overview of Structures**

The structures system is responsible for the design, material selection, simulation and manufacturing, and integration of three components of the pod: (1) the shell, (2) the frame, (3) a carrier for the entire pod. (1) and (2) integrated is shown in Figure ST1. All of the material selection for the components was assessed using CES EduPack Eco 3, the design and CAD work was conducted on SolidWorks, and all simulations will run on ANSYS 2019 R2 and 2020. The goal of the Structures team is to minimize the mass of the components without compromising the pod strength - this was primarily achieved throughout the material selection process and through the reduction of unnecessary weight in the design stage. The mass and maximum dimensions of the structural components are shown in Table ST2.

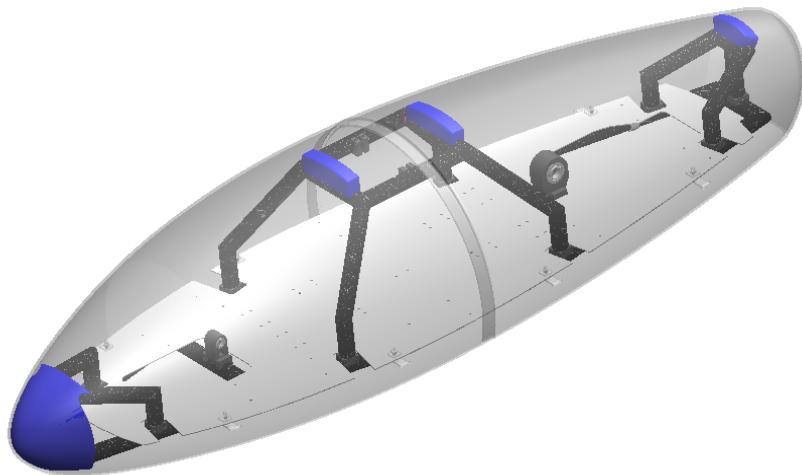


Figure ST1: Shell and Frame Assembly Integrated

### **Parts List**

Table ST1: Structures BOM

Part Name	Material	Dimension/ Unit	Qty	Function
<b>Shell</b>				
Prepreg	Prepreg 3K, 2x2 Twill	50" x 36"	16	Main component in building the shell

	Weave Carbon			
Nomex Honeycomb	Honeycomb	36" x 96"	3	Sub-component to enhance material properties of the shell
Epoxy	Epoxy	5L	2	Bonding agent, used in multiple processes of shell manufacturing
Offset Surface Brackets	Aluminum	1" x 4-1/4" x 1-1/2"	8	To attach the shell to the sheet metal frame
<b>Frame</b>				
Sheet Metal	Cold Rolled Steel C1008	96" x 48" x 0.104"	1	Main component for supporting loads of all subsystems components
Routing Clamps	Steel	Inner Diameter: 1-11/16" Length: 4-3/8"	10	Attaches the Carbon Fiber Tubing Assembly to the sheet metal frame
Carbon Fiber Tubes	Filament Wound Carbon Fiber	Inner Diameter: 1.50" Outer Diameter: 1.625" Length: 12"	27	Main material used in the backbone of the frame assembly
CARBONNect Fixed Connector	6061 T6 Aluminum (screws also included)	Diameter: 1.50"	12	Used to secure Carbon Fiber Tubes together at intersecting angles of 90 degrees
3D Printed Insert	PLA	Outer Diameter: 1.50"	24	Used to aid in the manufacturing of Carbon Fiber Tubes which are joined at angles other than 90 degrees
Foam	PVC	6' x 2'	3	Dampens vibrations on the entire structure

Table ST2: Dimensions of all Structures Assemblies

Assembly	Length (maximum dimension)	Width (maximum dimension)	Height (maximum dimension)	Estimated Mass
Shell	2.16m	0.66m	0.44m	12.20kg
Frame	2.08m	0.62m	0.36m	26.85kg

## Full Description of Structures

### Material Selection

After creating rough geometry sketches while keeping manufacturing considerations in mind, the main design guideline was through material selection. All of the material selection was conducted using CES EduPack Eco 3. An objective and constraint function were selected for each structural component, with the goal of minimizing the mass of each structure and obtaining a stiff and strong foundation. Through parameterization, a free variable (which is present in both the objective and constraint function) is isolated in terms of the constraint function and then substituted in the objective function. The objective function is then categorized into three sections: F (functional variables), G (geometrical variables) and M (material properties). The materials performance index (MPI) is determined by only including the M variables in the objective function. The MPI is set to be maximized, therefore if the objective is to minimize a variable, the reciprocal of the M variables will be chosen. The MPI is then converted into the form of a linear equation which is graphed in the CES program. Through CES, a line with the slope in the MPI equation is graphed onto the list of materials, and the goal is set to maximize the MPI. Once this line is plotted, various filters are applied to the list of materials in order to design towards any appropriate constraints for the purpose of the specific application.

### Shell Material Selection

The equation of the objective and constraint functions are shown in Equations 1 and 2. As referred to in Section ST1, the objective for the shell is to minimize its mass, as a lighter structure will allow the pod to accelerate faster. The constraint for the shell is to minimize the shell's deflection during the pod's operation, such that the total deflection is less than the maximum deflection the shell's material can withstand.

$$\text{Equation 1 (Objective Function)} : m = \rho v = \rho w t L$$

$$\text{Equation 2 (Constraint Function)} : \delta = \frac{FL^3}{CEI} \sim \frac{FL^3}{EI}, \text{ where}$$

- $m$ : mass
- $\rho$ : density

- $v$ : volume
- $w$ : width
- $t$ : thickness
- $L$ : length
- $F$ : Force
- $C$ : Integration constant (assumed to be  $\sim 1$ , as this value will not make a difference in the MPI)
- $E$ : Young's modulus
- $I$ : Polar moment of inertia

As the shell is an organic shape, it is assumed the shell is composed of infinitesimally small plates, for the sake of simplifying the moment of inertia. The moment of inertia of a plate is shown in Equation 3. From this, the free variable is the thickness of the shell, as this is something that can be easily adjusted when manufacturing the shell. From Equation 3, thickness is parametrized and displayed in Equation 4. Equation 4 is then substituted back into Equation 1, to give Equation 5, which is separated into M, F, and G sections respectively in each bracket.

$$\text{Equation 3: } I = \frac{wt^3}{12}$$

$$\text{Equation 4: } t = \left( \frac{12FL^3}{Ew\delta} \right)^{1/3}$$

$$\text{Equation 5: } m = \left( \frac{\varrho}{E^{1/3}} \right) \left( \frac{12F}{\delta} \right)^{1/2} (L^{2/3} w)$$

From Equation 5, the reciprocal of the M variables are taken to yield Equation 6, which is the MPI. The logarithm of this equation is then taken to obtain a linear equation, shown in Equation 7.

$$\text{Equation 6: } \text{MPI} = \frac{E^{1/3}}{\varrho}$$

$$\text{Equation 7: } \log(\text{MPI}) = \frac{1}{3} \log(E) - \log(\varrho)$$

On CES, a chart with density (Y axis) and Young's modulus (X axis) is displayed with the full set of "Bulk Materials" initially selected. The slope of the line in Equation 7 is plotted on the graph with the objective of maximizing the MPI. Before this was completed, various filters on the appropriate materials were defined. Due to the complexities of the shape of the shell, all metals and ceramics were filtered out to minimize manufacturing difficulties. The final filter was inputted to only output isotropic materials, as the loads on the shell experiences will act on various angles and positions; due to this limitation, only woven composites were accepted. Figure ST2 shows the top three materials, and from this, woven prepreg carbon fiber with a biaxial lay-up was selected for the shell.

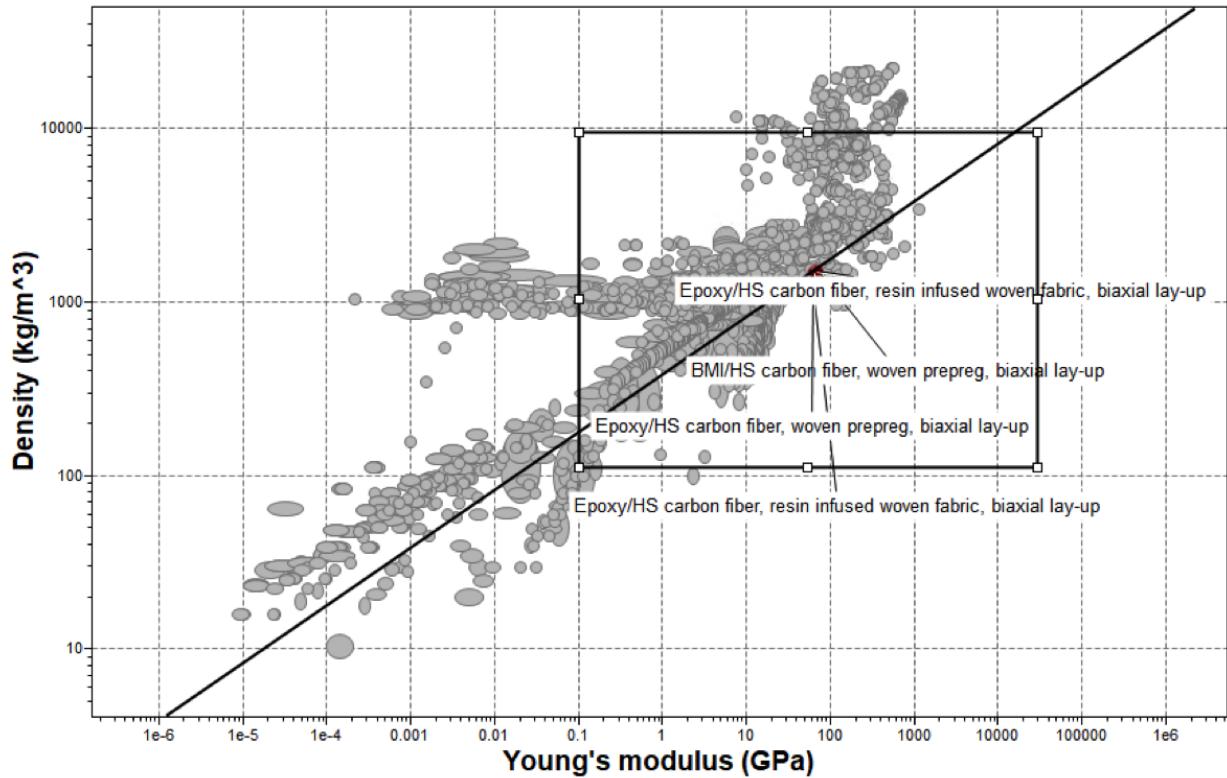


Figure ST2: CES Plot for Selection of the Shell Material

### Frame Material Selection

With the exception of the filters applied to material selection for the shell, the methodology of selecting the material for the frame, including objective and constraint functions, was identical to the shell material selection. Due to the price and machinability of the sheet (water-jet or CNC), alloys of aluminium and steel will be compared. As this material also must be solid, any filler or fibrous metals were also filtered out. Figure ST3 shows that carbon steels provide the best MPI. With the remaining materials on the CES graph, the optimal alloy, C1008 Cold Rolled Steel, was chosen based on a cost analysis.

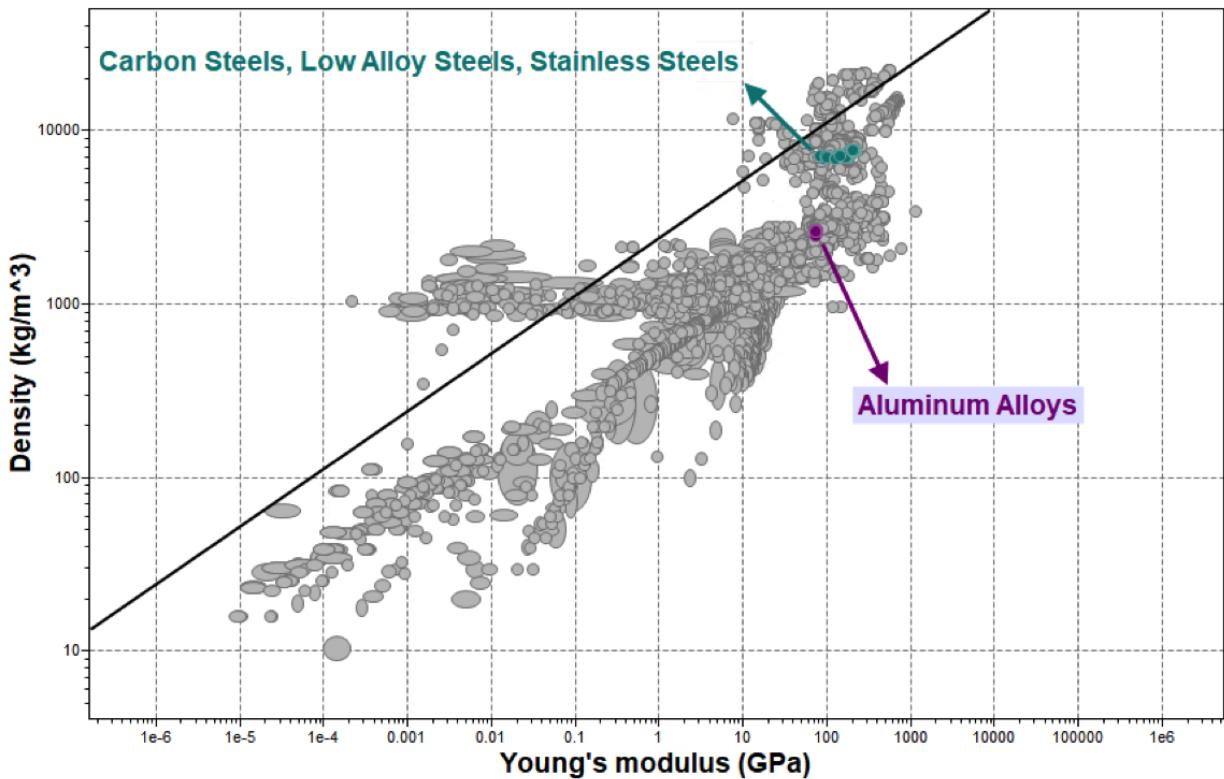


Figure ST3: CES Plot for Selection of the Sheet Material

### Shell Design Consideration

The geometry of the shell was iterated to achieve optimal aerodynamic qualities (lowest pressure, lowest turbulence), with preliminary CFD analysis of several candidate shell geometries in ANSYS Fluent. The three candidates include three different shapes: (1) airplane nose, (2) bullet, (3) airfoil, which were all inspired by existing aircraft geometries. The results of this preliminary analysis is shown in Table ST3. As the airplane nose geometry had the lowest maximum pressure and a fairly low maximum turbulent kinetic energy, it was chosen as the optimal geometry to be constructed from a sandwich structure of prepreg carbon fiber and Nomex honeycomb. To manufacture these composite materials, a male and female mold were created prior to the lay-up of the sandwich structure. Due to the nature of manufacturing, which includes removing the actual composite build up from the female mold, the shell needed to be designed in two pieces. Since the shell wraps around greater than 180 degrees, it cannot be removed from the female mold without being two pieces, which is shown in Figure ST4. A structural simulation on the shell based on aerodynamic forces is presented in Section ST4.4, which verified the rigidity of the materials and the drag forces that would be experienced during the pod's operation.

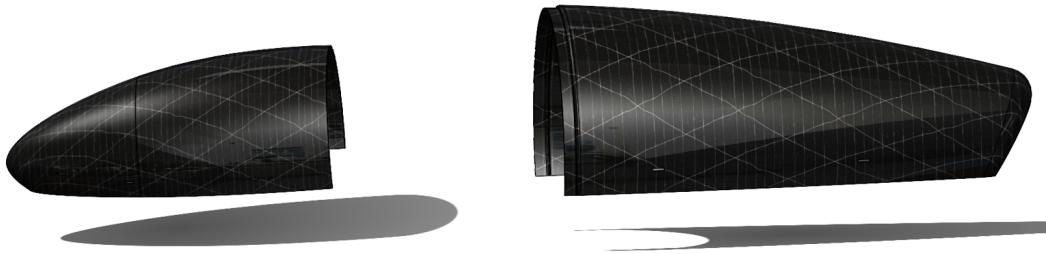


Figure ST4: Two Halves of the Shell

Table ST3: Criteria Selection for Shell Geometry

Variable	Airplane Nose	Bullet	Airfoil
<b>Max Pressure (Pa)</b>	3295.45	3550.12	3456.36
<b>Max Turbulent Kinetic Energy (m<sup>2</sup>s<sup>2</sup>)</b>	6.27	11.9	5.33

### Frame Design Considerations

The frame component is broken down into two sub-components: (1) the carbon fiber assembly, (2) the sheet metal base. The full frame assembly is shown in Figure ST5. C1008 cold rolled steel was chosen as the material for the baseplate due to its high stiffness, relatively low density, and ease of manufacturing. The sheet metal base, shown in Figure ST6, serves as a mounting point for the components of other components (shell, braking tank, etc.), while the carbon fiber assembly serves as the chassis of the pod, providing the majority of the pod's structural rigidity. The carbon fiber frame is assembled with RockWest carbon fiber tubes (Figure ST7), which are joined together via RockWest CARBONNect adaptors (Figure ST8) or epoxy and prepreg. The latter approach for joining tubes together as required for tubes which are connected at angles other than 90+/- degrees due to the constraints of the RockWest adaptors. Such joints will be created by cutting the carbon fiber tubes and using 3D printed inserts (Figure ST9) to aid in the angular accuracy of bonding them together via epoxy. Finally, these joints will be wrapped in prepreg to reinforce the bonding. Preliminary static FEA analysis was conducted through an iterative process to determine the optimal placement of carbon fiber tubes. Two main simulations conducted on the frame are presented in Section ST4.4: (1) a static simulation to determine the strength of the frame and baseplate when fully-loaded with all the other subsystem components,

(2) a dynamic simulation investigating the stresses in the frame structure during an extreme braking case.

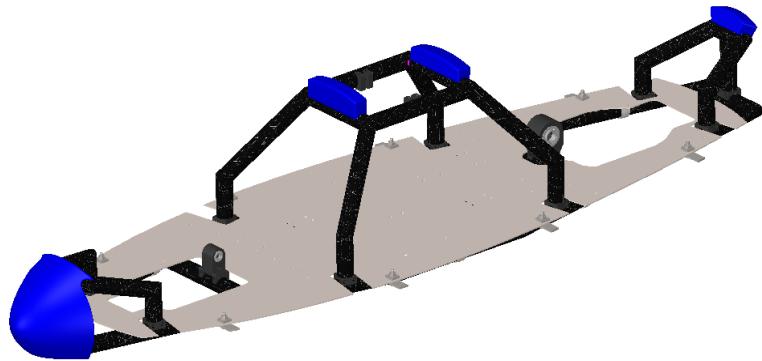


Figure ST5: Full Frame Assembly

*Sheet Metal*

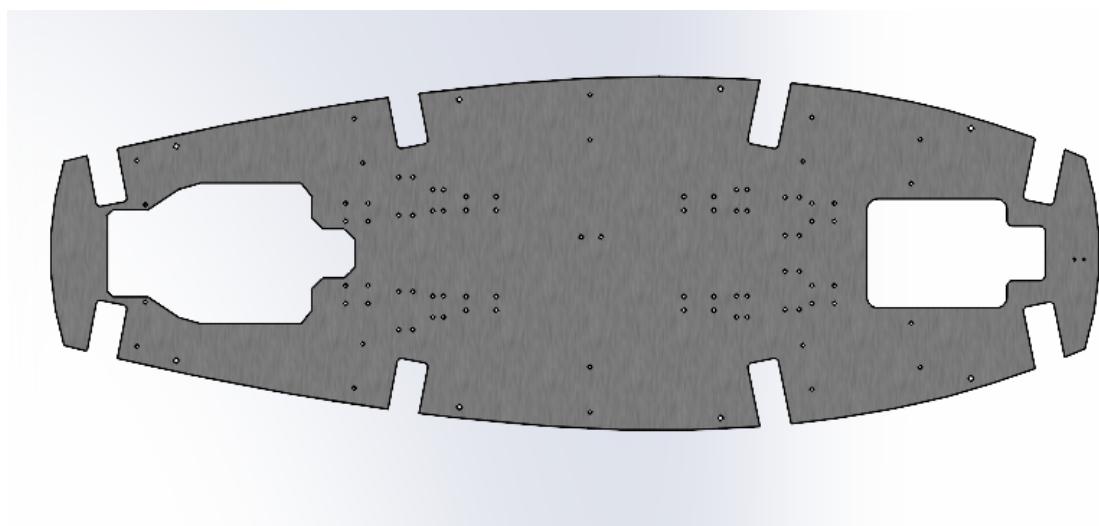


Figure ST6: Sheet Metal Base

*Carbon Fiber Tubes*



Figure ST7: Carbon Fiber Tubes

*CARBONNect*

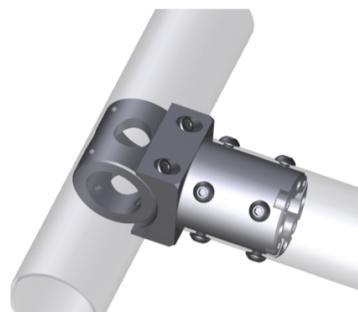


Figure ST8: CARBONNect System

*3D Printed Inserts*

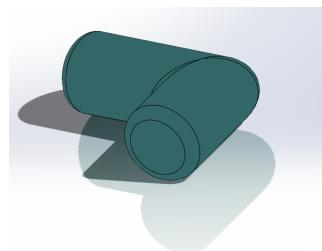


Figure ST9: 3D Printed Inserts

## Micro-Integration of Structures Assemblies

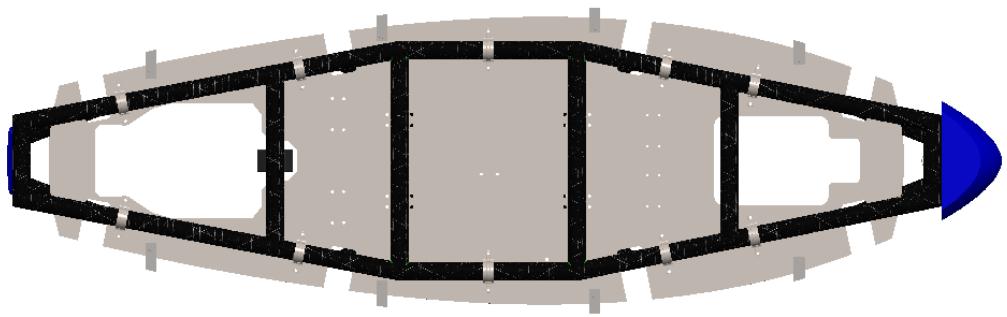


Figure ST11: Bottom View of the Frame Assembly. U-Clamps and L-brackets shown.

Figure ST11 shows the frame is attached to the sheet metal base by 10 U-clamps (Figure ST12) that run along the bottom of the frame, with the sheet metal base in turn being epoxied to the shell by 8 L-brackets (Figure ST13). The shell also rests on four foam blocks located at the front, mid, and back of the frame, which provides additional support and vibration dampening for the shell, which are shown as blue in Figures ST14.

### *Routing-Clamps*



Figure ST12: Routing Clamps

### Offset Surface Brackets



Figure ST13: Offset Surface Brackets

### Foam Supports (Front, Middle(2), Back)

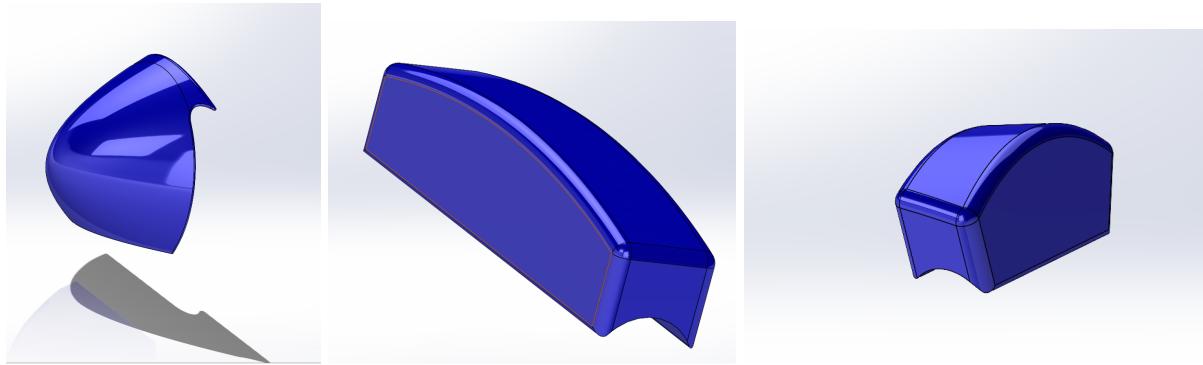


Figure ST14: Foam Supports (from left to right: Front, Middle, Back)

## Safety

This section aims to highlight all the safety aspects involved in the Structures subsystem. This was primarily conducted through FMEA analysis, with the risks assessment presented in Table ST4. As a team, UTHT chose to set a Risk Priority Number( RPN) as 30, with anything over this limit being considered as a higher risk. The risks highlighted in red are justified and explained in the later subsections of Section ST4. These subsections also display which ID number on Table ST4 to which they correspond.

Table ST4: Structures Subsystem FMEA Risk Assessment

ID	Process/ Mechanism	Risk	Effects	Causes	Prevention Control	Detection Control	Severity Rating (S)	Frequency of Occurrence (O)	Ease of Detection (D)	Risk Priority Number (RPN)
1	Shell operation	The two halves of the shell peeling apart during run of the	Increased vibrations from aerodynamic loading. Potential damage to all	- Overlap between two halves of the shell is not sufficient enough in	The design of the halves and how they meet causes the	When the tolerance between the two halves of the shell is	6	3	1	18

		pod	components if directly exposed to aerodynamic loads.	depth/width. - Under fastening/over fastening of offset surface bracket - Back flow of air at the top of the top.	front half of the shell to press into the back half during operation.	greater than 1mm.					
2	Shell manufacturing	Use of hazardous sprays during manufacturing of the shell	Members' health could be compromised.	Improper use of the sprays, not wearing the appropriate PPE (masks)	PPE: P100 grade respirators	Coughing/trouble breathing while manufacturing	5	1	1	5	
3	Frame load support	Frame failure on the sheet metal	All components mounted to the frame are at risk of damage. Track/wheels can also be damaged, causing major failure during operation	Material/mechanical failure, stress concentrations	FEA analysis (static/dynamic), loading tests prior to operation	Deflection in the sheet metal base	10	2	6	120	
4	Shell load support	Shell failure - Delamination of the composite	Rupture of composite. Internal components at risk to the environmental conditions, causing potential damage	- Manufacturing defects - Failure of L-brackets	FEA analysis (static + CFD)	- Plastic deformation in the shell - Non Uniformity of the shell (i.e. bulge)	10	2	2	40	
5	Frame (CF tube assembly)	Failure of Carbon Fiber Tube Joints (two types: prepreg wrap and RockWest adaptors)	Total failure of the frame leading to destruction of other components	- Overloading - Quality of the product (prepreg wrap vs RockWest adaptors)	FEA of the tubing; tensile/compressive tests;	Snapping of tubes during static loading	10	2	6	120	
6	Frame (CF tube assembly)	Cracks or Manufacturing errors in Carbon Fiber Tubes	Tube failure can lead to catastrophic failure	- Manufacturing errors in Carbon Tubes (cutting specifically) - Material Handling - Residual stresses from Machining - Overloading	- Enforce proper manufacturing protocols - Certification from suppliers	Abnormalities on tube surface	10	2	3	60	

7	Frame (sheet metal + U clamps; L-bracket + sheet metal)	Failure (loosening/tightening) of bolts/nuts	Bolt failure	Under fastening/Over fastening	- Ensuring proper tightening torque standards. - Use lock nuts in the areas where vibration can loosen nuts - Checklist to list the areas where nuts to be checked for fastening.	-Components loosening - Visual Inspection using checklist	10	1	3	30	
8	Frame (CF tube assembly)	Compressing on tubes/Tearing from manufacturing (fastening/assembly)	Tube failure can lead to catastrophic failure	Over fastening of fasteners	Enforce proper manufacturing protocols	Wear on components	7	1	1	7	
9	Frame assembly	Subsystems does not fit inside the center structure	Critical components could be missing from the system if there is not enough room for it	Incorrect Width/Height of center structure	- Ensuring proper geometric tolerancing for inspection during the design stage. - Quality checks post manufacturing	- Quality Checks for primary dimensions that ensure complete placement for components in the center structure.	4	1	3	12	
10	Frame assembly	Suspension or other parts not getting mounted	- Assembly stopped - or might cause heavy wear on the fasteners	Shift of mounting points	- Ensuring proper geometric tolerancing for inspection during the design stage. - Quality checks post manufacturing	- Quality Checks for primary dimensions that ensure complete placement - Template can be printed to place and check if all the mounting holes are in position	8	1	3	24	

11	Frame assembly, shell and inner part assembly	Failure of carbon tubes, shell separation and bolts loosening	-Tube failure can lead to catastrophic failure of pod - damage to the testing tube	- Sudden jerk due to loss of momentum - Emergency braking case	FEA of the tubing; tensile/compressive tests; certification of the material	Loss of structural integrity	10	3	2	60	
----	-----------------------------------------------	---------------------------------------------------------------	------------------------------------------------------------------------------------	-------------------------------------------------------------------	-----------------------------------------------------------------------------	------------------------------	----	---	---	----	--

## Simulations

### Static Shell Analysis - ID4

Delamination, a common mode of failure in composite, will most likely be one of the failure mechanisms should the shell be unable to withstand the loads applied to it. A finite element analysis was carried out to ensure that it is capable of withstanding these loads. The geometry used in this analysis is presented in Figure ST15. Two notable changes were made in this geometry compared to the actual shell. The first is that the shell is a single piece, in order to reduce complexity of the simulation. An assumption was also made that the two halves of the shell would essentially be fixed and act as a single piece, as discussed further in Section ST4.4.1. The second deviation is the lack of foam/frame components used, also to reduce complexity of the simulation. Due to this deviation, any deformation/stress on the pod will actually be lower in actuality, due to the dampening properties of the foam/frame.

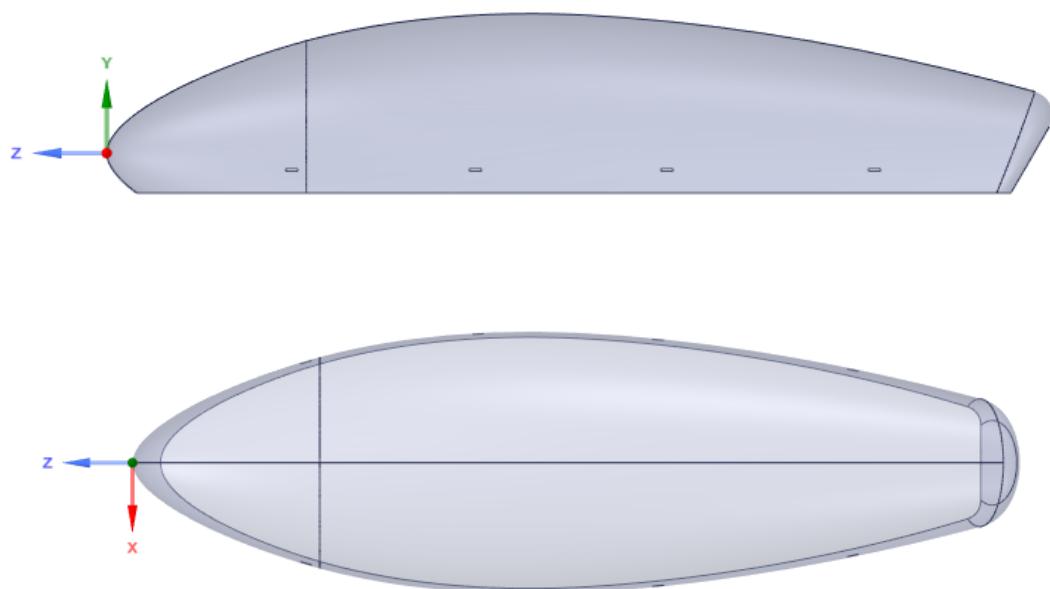


Figure ST15: Geometry of the shell used in the FEA analysis

Table ST5: Loads and Constraints in the Static Shell Analysis

<b>Prepreg (one layer)</b>	0.3048mm
<b>Honeycomb</b>	5.334mm
<b>Load</b>	3295Pa

Loads:

The load applied to the entire shell is the maximum pressure calculated from the CFD analysis, presented in the Section ST4.1.2, which is 3.295MPa.

Constraints:

The model is constrained using fixed supports at the rectangular holes which is to simulate the shell being fixed to the frame of the pod in reality. The holes connect the shell to the frame.

Meshing convergence study:

Generally, triangular or tetrahedral first order elements are too stiff, especially in bending. Therefore, it is ideal to use only quad elements throughout the model, however, due to the geometrical configuration and achieving convergence, triangular elements are also included in the mesh. 8 node quad and 6 node triangular elements are used because they will produce a higher accuracy and since the geometry is not complex, the computational time is not a big factor to consider.

To study the mesh convergence, the global mesh element size was set to a coarse mesh of 100mm and sensitivity parameter is maximum deformation because we are interested in understanding how the shell contorts/conforms under the load. The element size is varied linearly until we arrive at a point where the graph begins to converge. This point indicates that the results generated are not tied to variations of the mesh element size. The value of the element size at this point was calculated to be 40mm. Element refinement is necessary at the holes because this is where fixed support is added, so accuracy is desired in this region. So, refinement values of 1 and 2 are compared and as evident, convergence is more prominent on 2, so this is also selected as a fixed parameter going forward.

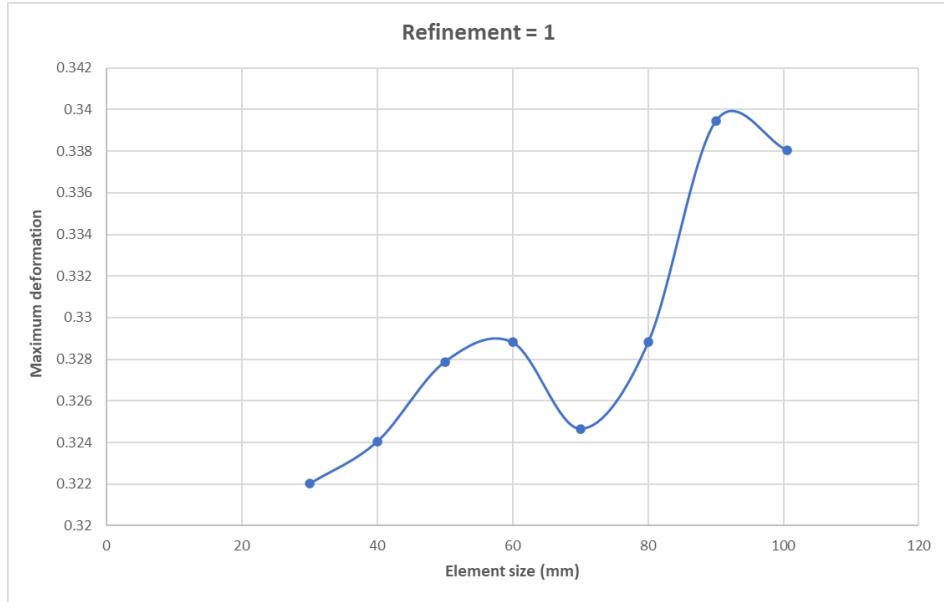


Figure ST16: Mesh Sensitivity Study, Refinement = 1

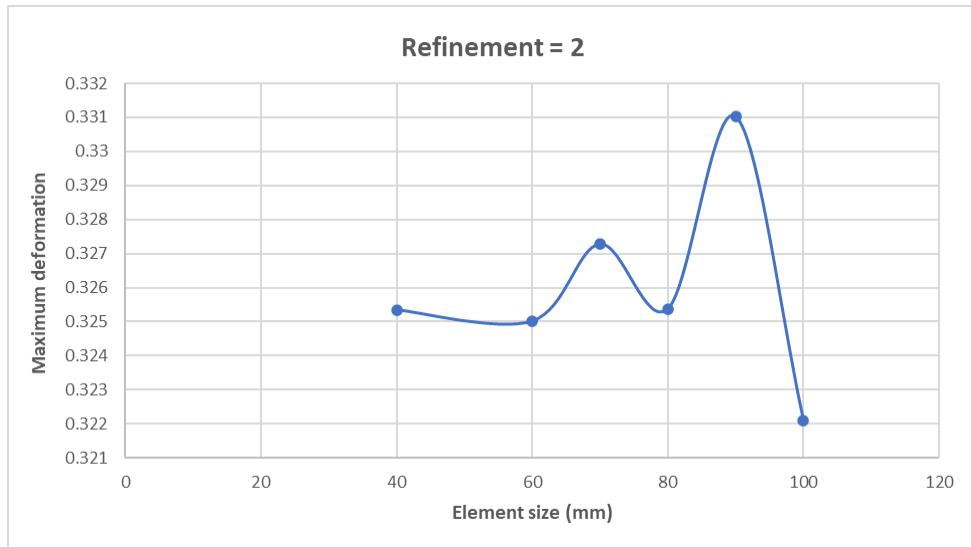


Figure ST17: Mesh Sensitivity Study, Refinement = 2

For the analysis, the maximum deformation, von Mises stress (layer), and checks with all the failure criterion (Tsai-Wu, Tsai-Hill, Hoffman, Hashin, Puck, LaRC ana Cuntze) are desired. The maximum deformation is calculated as 0.3251mm but in actuality it will be lower due to the presence of the frame and foam that acts as a damper in these regions of maximum deformation.

The minimum stiffness based on the stack-up arrangement shown below as 8000MPa in the  $45^0$  direction and the maximum equivalent von Mises stress calculated for the layer is 25.911MPa

therefore the structure is suitable for the withstand this load, and hence will not fail under compression/tension loading modes.

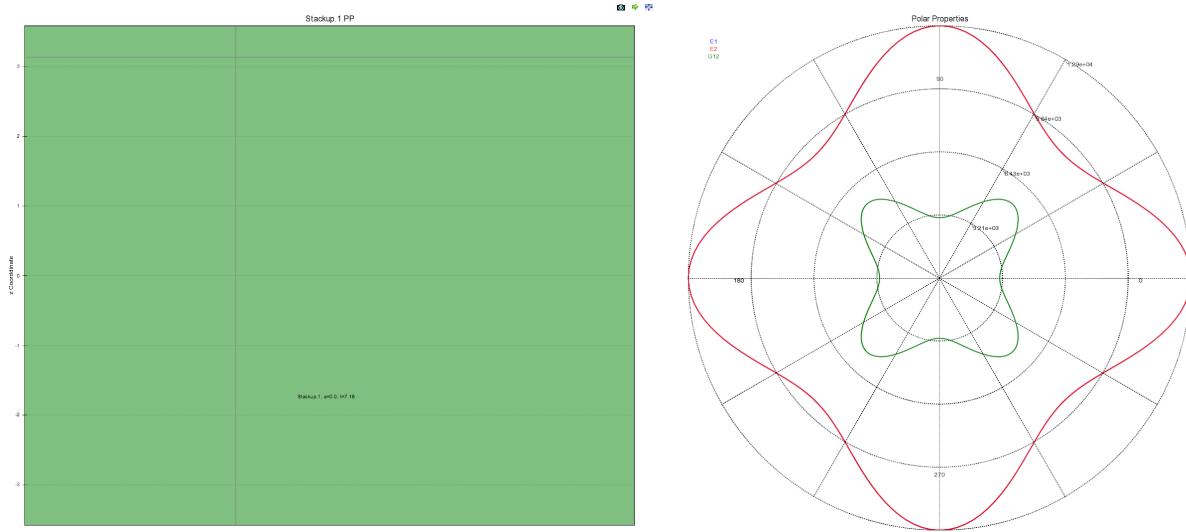


Figure ST18: Composite stackup orientation

Comparison of various stackup orientation:

The composite analysis is done using ANSYS ACP (Pre) and the image below shows how the plies are stacked in a bottom-up configuration. The angles are symmetric after the Honeycomb matrix. Because we are interested in the deformation in particular, the best orientation is [  $45^0$   $0^0$   $45^0$  honeycomb  $45^0$   $0^0$   $45^0$  ] from iteration 6 which has the lowest maximum deformation.

Table ST6: Stack Up Orientation

Prepreg 6
Prepreg 5
Prepreg 4
Honeycomb
Prepreg 3
Prepreg 2
Prepreg 1

Table ST7: Ply Angle Analysis Study

Iteration no.	No of Layers/Stack	Angle 1	Angle 2	Angle 3	Maximum Deformation (mm)	Maximum Stress (MPa)
1	2	45	45	-	0.6593	33.822
2	2	45	90	-	0.46197	40.349
3	2	90	90	-	0.67705	29.144
4	3	90	0	90	0.45894	19.919
5	3	90	0	45	0.3251	25.911
6	3	45	0	45	0.31943	20.612
7	3	45	0	90	0.3231	23.167
8	3	90	45	0	0.32352	24.039
9	3	45	90	0	0.3231	23.167

### CFD Analysis

CFD analysis is used to analyze complex problems involving fluid-fluid, fluid-solid or fluid-gas interaction. Here the pod would be travelling through a 12-meter long vacuum tube. The CFD analysis has been used to compare and optimize different aerodynamic shapes for the shell. We selected three different shapes for the pod based on research of previous literature. They were: bullet, airplane nose and airfoil. Fluid, pressure, turbulent kinetic energy, and drag defined the shape that the team further optimized and took to the manufacturing process.

The fluid pressure and eddy turbulent viscosity results from the different shapes are provided below in the table:

Table ST8: Criteria selection for the shape of the Hyperloop shell

Variable	Airplane Nose	Airfoil	Bullet
<b>Max Pressure (Pa)</b>	3295.45	3456.36	3550.12
<b>Max Turbulent Kinetic Energy (m<sup>2</sup>s<sup>-2</sup>)</b>	6.27	5.33	11.9

Based on the results from above, we chose Airplane Nose since it provided the best performance. The simulation setup started by extracting the geometry of the pod (Figure ST15) from the 12m long EHW Tube using a boolean operation in Design Modeller. The pod was placed in the middle of the track as shown in Figure ST19

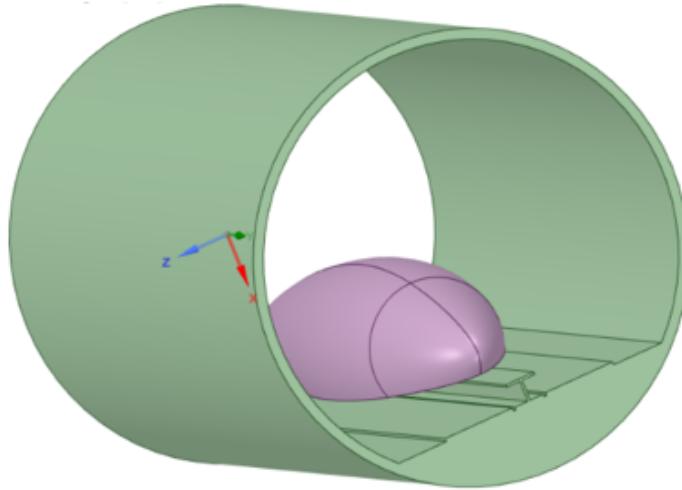


Figure ST19: Volume Extraction analysis for CFD

#### Mesh Generation:

By consulting the ANSYS Learning Hub, we established a set of meshing best practices that secured the most accurate results. For this analysis, we used CFX solver with a mesh of 0.05 m sized elements in most of the volume. The surface that enclosed the Hyperloop shell was inflated with 15 layers and a maximum element size of 0.01 m with the purpose of capturing pressure and velocity gradients at the boundary layer. The tiny sections of the I-beam and the gap between the track aluminum plates and the I-beam had local mesh size reductions to 0.012 from 0.05 m (I-beam flange thickness is 0.0127 m).

#### Properties for solver:

The following table presents the boundary conditions and solver control settings for the CFD analysis:

Table ST9: Boundary conditions and solver control settings for CFD

BC/Settings	Value	Justification
Model	k-epsilon	Small pressure gradients
Inlet velocity (m/s)	70	Top-speed of the Hyperloop pod generated by our electrical EMRAX motor.
Pressure (atm)	0.001	Min pressure limit inside the EHW Track
Outlet Relative Pressure (Pa)	0	Section of the tube long enough that pressure gradients are dissipated.

Temperature (°C)	28	External Reference Temperature provided by EHW
RMS Residual	1e-5	CFD best practices (Calvin Moes)
Number of iterations	2000	CFD best practices (Calvin Moes)
Heat Transfer	Isothermal	Not expecting a significant temperature change

The first assumption in this analysis is that components protruding from the shell, such as the stability system will not affect the aerodynamics of the structure as a whole due to the lack of considerable air resistance (because of near vacuum condition in the tube). We have also assumed that the shell acts as a closed body. Therefore, no fluid would enter the internal space of the pod. These assumptions were made to reduce unnecessary complexity of the simulation since the accuracy results would not be significantly affected by these factors. The Hyperloop pod has a stainless steel sheet covering the bottom of the pod shell and the stability system is a relatively small subsystem in comparison to the size of the pod.

#### Results and Discussion:

The CFD analysis was carried out on the airplane nose shape. Three main sections of interest are the nose, top, and back of the pod. From Figure ST20, we see that the most of the pressure is concentrated at the nose, as expected. As the pod increases in speed, air accumulates at the front of the pod, which generates the largest pressure gradient at the nose. This maximum pressure value is extracted and used for static shell analysis as mentioned in 4.1.1.

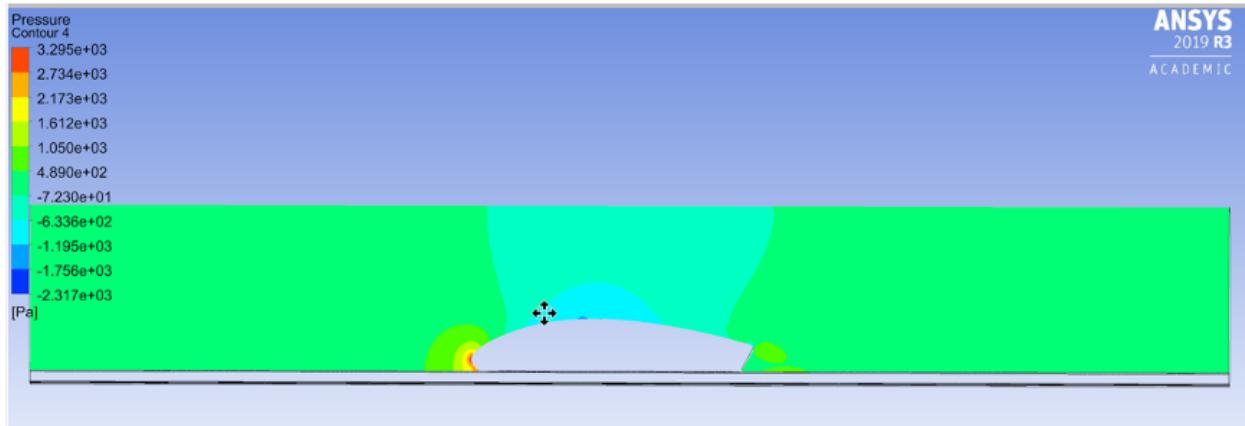


Figure ST20: Pressure distribution in a central plane parallel to the Y-axis

Next, we move on to the top of the shell on Figure ST21, where we see that the velocity of air is maximum. According to the results, the speed of the air is faster than that of the pod, and the air's velocity vector slightly points downwards. This means that the pod experiences an

aerodynamic downforce effect, which would increase in intensity as the pod accelerates. This is a desired aerodynamic effect, since it provides longitudinal stability and prevents the pod from pitching up and losing traction.

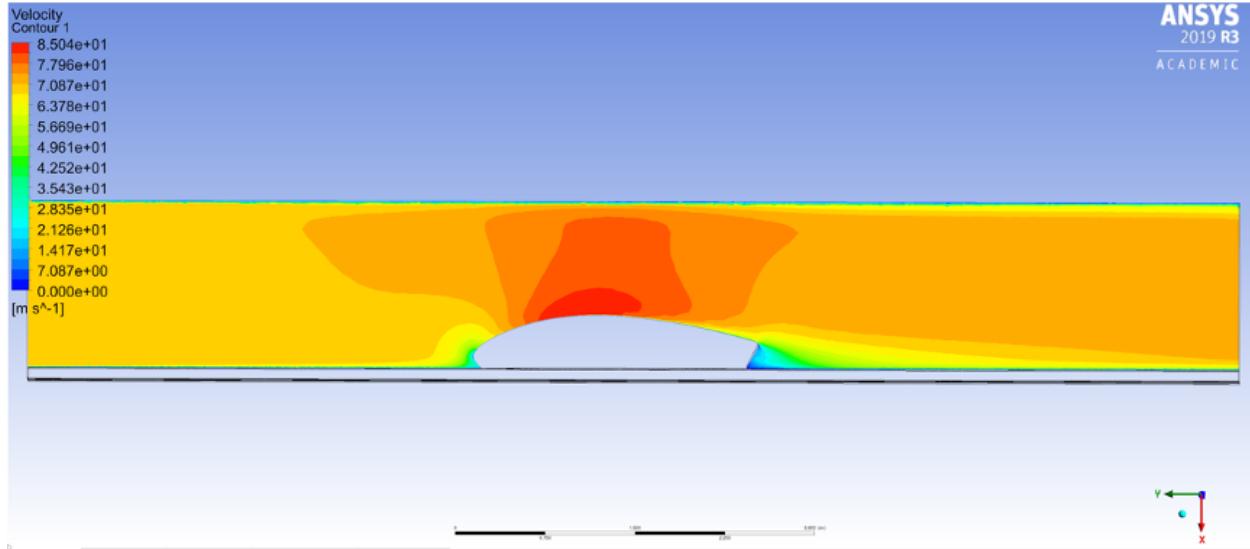


Figure ST21: Velocity distribution in a central plane parallel to the Y-axis

The back of the pod has an interesting effect, due to the inclined shape. The pod receives a small thrust from air being forced in the direction of travel of the pod. This can be seen from the velocity and pressure vector field as shown in Figure ST22 and ST23, respectively. In a larger pod this thrust might have a significant effect on the motion of the vehicle. However, in the case of our pod shell, the additional thrust is insignificant.

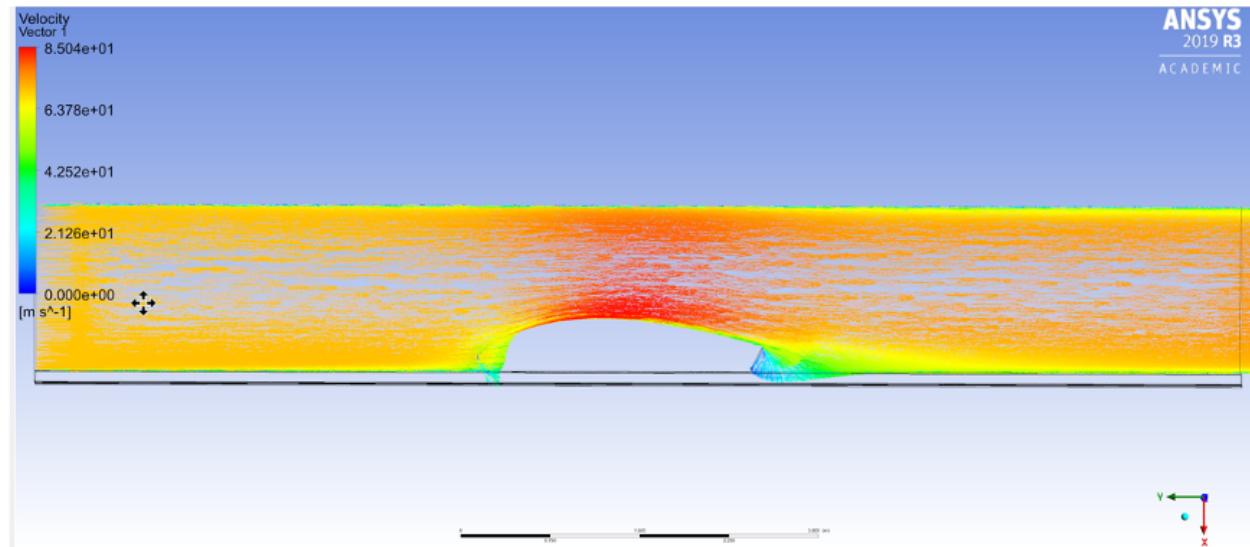


Figure ST22: Velocity vector field in a central plane parallel to the Y-axis

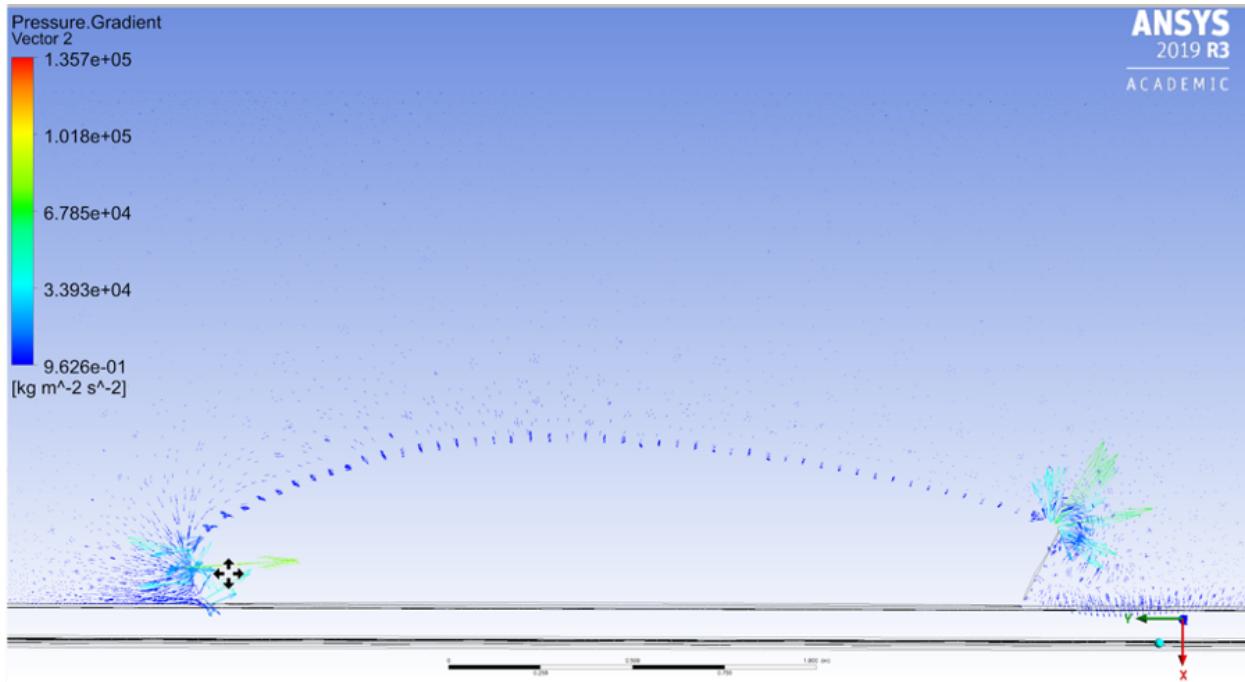


Figure ST23: Pressure gradient vector field in a central plane parallel to the Y-axis

#### Static Frame Analysis - ID3 (sheet metal), ID5 (carbon fiber tubes)

To assess the structural integrity of the frame to withstand the loads of all other subsystems components, a static FEA analysis was performed in ANSYS. The materials selected for the sheet metal and tubes were identical to those purchased (C1008 Cold Rolled Steel and Filament Wound Carbon Fiber respectively). To minimize the risks presented in ID3 and ID5 of Table ST4 verification of these simulations requires a Safety Factor of at least 2.<sup>1</sup>

##### Loads and Constraints:

The load of the pod (~150kg) was distributed based on the location of components throughout the structure, with the pod being fixed at the two holes where the wheels lie.

##### Results:

Figures ST24 And ST25 below show the deformation and stress plots, respectively. The deformation plot shows most deflection in the center of the frame, as this area is heavily loaded with components such as batteries and suspension. However, even the maximum deformation of 6.3564mm is considered to be insignificant. This is further shown in the stress plots, where the maximum stress is ~25MPa. Based on the yield strength of the steel plate being ~250MPa, this provides a Safety Factor of ~10, and thus this reduces the risks presented on the frame.

<sup>1</sup> "Hyperloop", *SpaceX*, 2020. [Online]. Available: <https://www.spacex.com/hyperloop>. [Accessed: 13-Dec-2020].

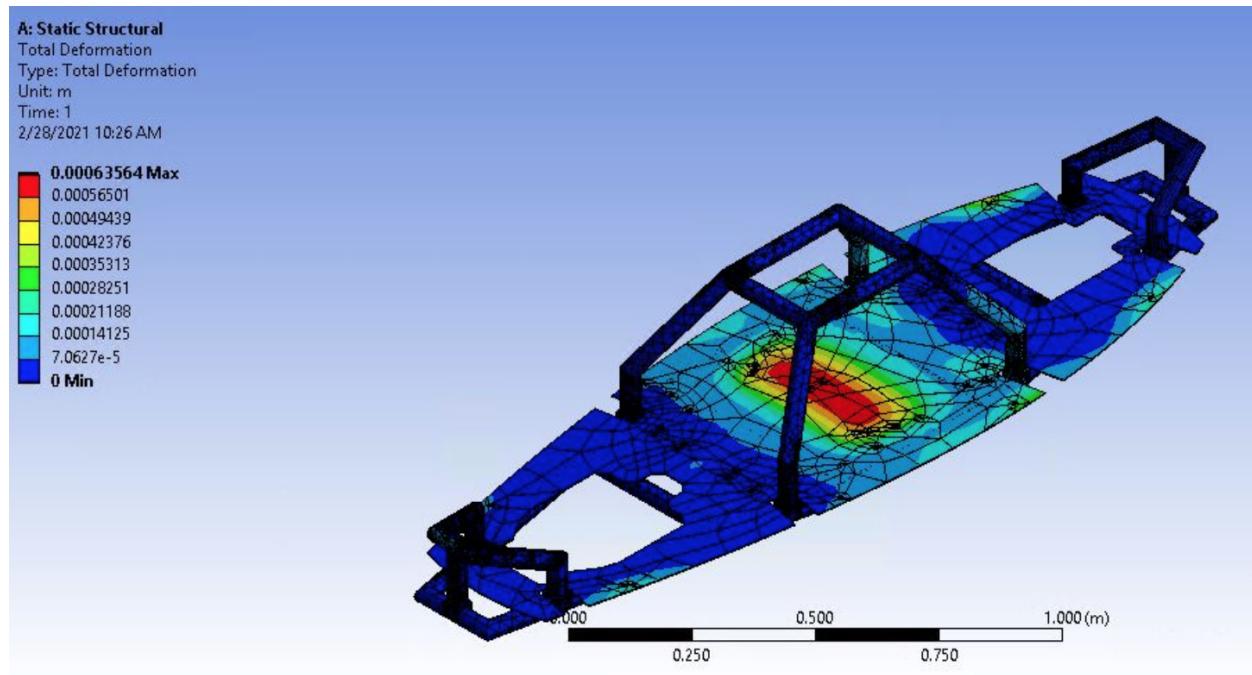


Figure ST24: Deformation of the Static Frame Analysis

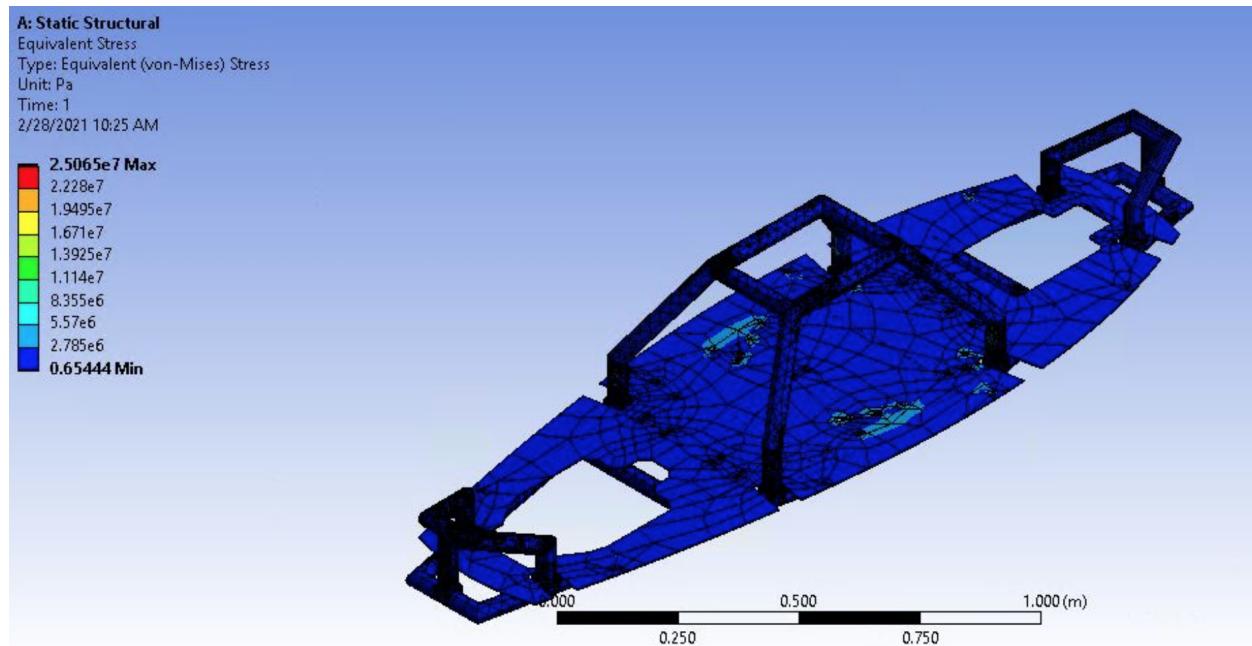


Figure ST25: Von Mises Stress of the Static Frame Analysis

## Dynamic Frame Analysis - ID3 (sheet metal), ID5 (carbon fiber tubes)

To assess the structural integrity of the frame and sheet metal base under an emergency braking scenario, a dynamic FEA analysis was performed in ANSYS. The materials chosen and mesh for the frame model were identical to the static frame analysis, these being identical to those purchased (C1008 Cold Rolled Steel and Filament Wound Carbon Fiber respectively). To minimize the risks presented in ID3 and ID5 of Table ST4 verification of these simulations requires a Safety Factor of at least 2.<sup>2</sup>

### Loads and Constraints:

The load of the pod (~150kg) was distributed in an 80/20 front/back distribution (this is discussed further in the table below), on the front and rear horizontal frame elements where the wheel brackets are attached (5th and 2nd vertical frame elements in ST11, respectively). The pod's fixed boundary conditions (fixed supports) were located at the attachment points of the wheel suspension, identically to the static simulation.

The table below indicates the time-variable force loading conditions that were applied to the front and back wheel-supporting frame elements. The front element was expected to experience greater loading, due to the impact of braking shifting the relative weight distribution of the pod forward. These loads were determined by considering the maximum braking acceleration the pod experiences when applying and releasing the brakes in an emergency braking scenario. The first step provides the force on the and back tube for the time period of 0 to 0.2 seconds. The equivalent stress and total deformation is generated as a part of solution. After that, the second step is carried out changing the force to the value for a time period of 0.2 seconds to 7.8 seconds. Here, the simulation takes the result of step 1 as a base and continues. It is also in this second step that the force on the front and back elements is at its peak. Finally, simulation parameters are once again changed to those in step 3, and a stress and deformation result is obtained. Only the result with the greatest stress and deformation will be discussed, as this is the result from which the true safety factor of the design will be derived.

Table ST10: Time-Variable Force Loading Conditions

<b>Step</b>	<b>Time (s)</b>	<b>Back(N)</b>	<b>Front (N)</b>
1	0 – 0.2	$6000*t*0.2$	$6000*t*0.8$
2	0.2 – 7.8	240	960
3	7.8 – 8.45	$-6000*t*0.2 + 10140$	$-6000*t*0.8 + 40560$

---

<sup>2</sup> "Hyperloop", *SpaceX*, 2020. [Online]. Available: <https://www.spacex.com/hyperloop>. [Accessed: 13-Dec-2020].

## Results:

Figures ST26 And ST27 below show the deformation and stress plots at the maximum stress and deformation created by the braking load, respectively occurring during step 2 of the braking case loading pattern.

The deformation plot shows the most deflection towards the front of the frame, as this is where most of the force is concentrated and where the pod will bend the most during braking. However, the maximum deformation is 4.5255 mm, well within the deformation limit of the front of the pod. The stress plot further demonstrates the pod is able to withstand the forces of emergency braking, with the maximum stress being 1 MPa. Based on the yield strength of the steel plate being  $\sim 250$  MPa, this provides a Safety Factor of  $\sim 250$ , indicating that the pod will readily maintain structural integrity in an emergency braking scenario.

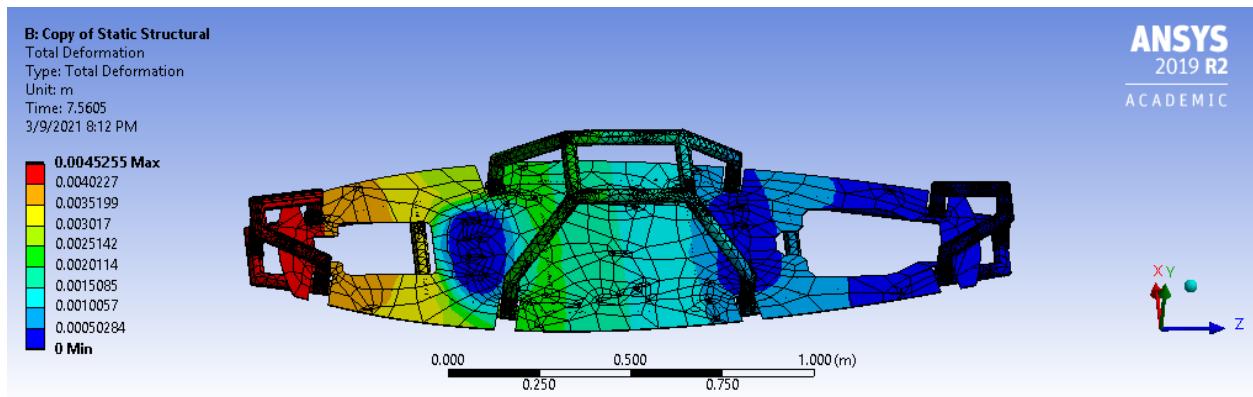


Figure ST26: Peak Deformation of the Static Frame Analysis

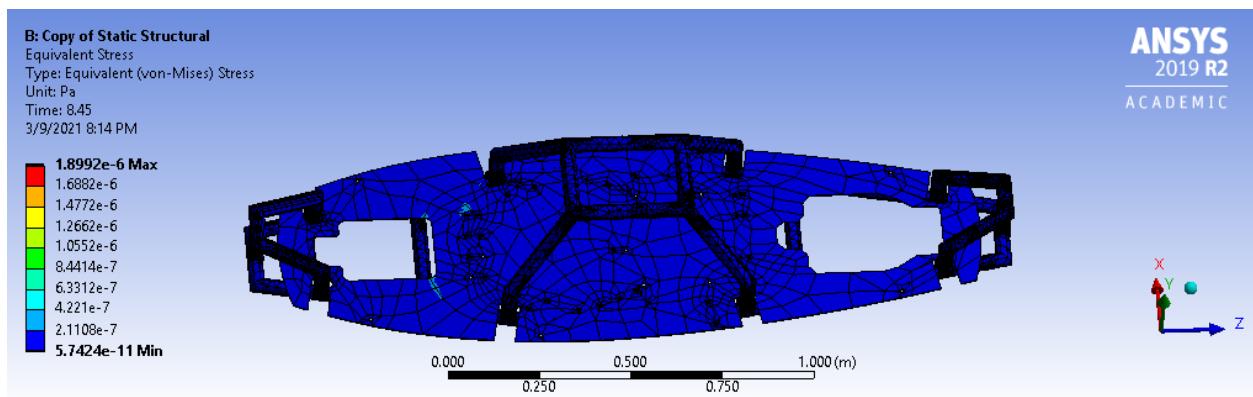


Figure ST27: Peak Von Mises Stress of the Static Frame Analysis

## Standards

### Bolt Standards - ID7, ID8

Choosing of the bolts and nuts was based on the following criteria:

- Shear strength of the bolt
- Material
- Use of specific type of bolt
- Cost
- Diameter with respect to the L brackets
- Threading
- Highest grade possible for the above requirements

The force which will be applied due to loss of momentum will be:

$$Force = mass * \frac{(Change\ in\ velocity)}{(Change\ in\ time)}$$

To get the maximum force that will be experienced by the shell, an unrealistic stoppage time of 0.1 sec was considered. The maximum velocity the pod achieves is 113 km/hr. i.e., 31.3889 m/s. By applying the safety factor of 2, we get the force that will be experienced by the pod. For the bolt, the shear strength of the bolt is given by:

$$Shear\ strength = \frac{(Tensile\ strength)}{\sqrt{3}} * \frac{((n_n * A_n * 0.78 + n_s * A_n))}{(Factor\ of\ Safety)}$$

Where,  $n_n$  will be equal to one if the shear plane lies on the threaded part of the bolt or will be zero if the plane lies on the non-threaded part of the bolt. Similarly,  $n_s$  will be equal to one if shear plane lies on the non-threaded part of the bolt and will be zero if shear plane lies on the threaded part of the bolt. As the bolt considered is fully threaded, the  $n_n$  will be equal to one and  $n_s$  will be zero. The diameter of the bracket holes is 0.25 and 0.375 inches,  $A$  which is the cross sectional area of the shear plane is the area of the radius of the bolt. The factor of safety was considered as two. In this case the bolts for L-bracket and U-clamps were considered. All the shear strengths calculated for the bolts were greater than the force experienced by the pod due to sudden stoppage and hence the pod was considered safe to operate in braking conditions. Further dynamic analysis needs to be done for carbon tubes and the joints for the force experienced due to sudden loss of momentum.

## Torque Standards - ID7, ID8<sup>3</sup>

To ensure that the bolt and nut can fasten together securely without breaking, standard tightening torque is to be followed. Too little torque makes our material untightened and structurally unsafe and too much torque could damage the structural integrity of the bolt or the material. The below image clearly shows the types of loads that are applied while fastening.

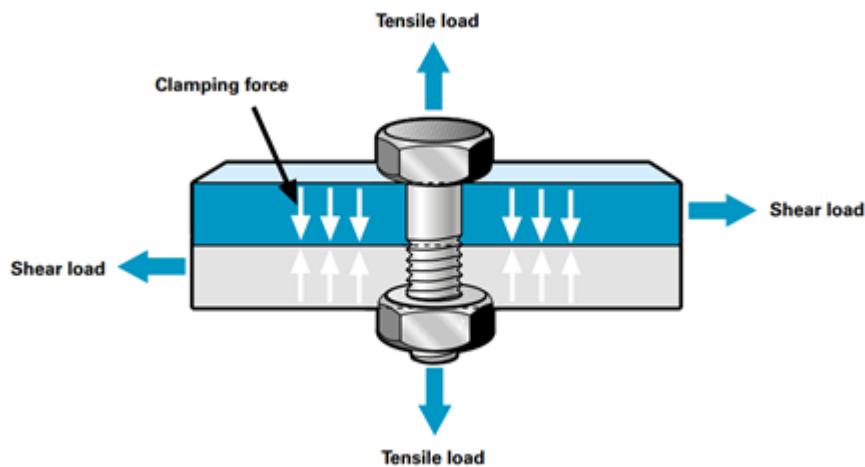


Figure ST28: Loads Applied during Fastening

As the screw is tightened, clamping force is built up within the material and the screw is stressed. This stress might bring in an elastic deformation in the screw. This elongation will continue until the stress in the screw is equal to the tensile strength at which the screw will break. As long as the clamping force generated is within the Elastic Curve (Hooke's Law) the fastening should be structurally safe. Any plastic deformation on the screw that happens after the yield point is going to permanently damage the screw. This idea of identifying the right tightening torque to ensure the screw does not take a clamping force more than the yield point is known as the tightening torque.

As per ASTM A574 – International Specification for Steel SoCket Head Cap Screws, the following are the tightening torque for the selected screws:

- 1) 1/4" – 20 Thread Size, 1" Long à 142 in-lbs (~16.043 Nm)
- 2) 3/8" – 16 Thread Size, 1-1/2" Long à 43 ft-lbs (~58.3 Nm)

---

<sup>3</sup> ASTM A574-17, Standard Specification for Alloy Steel SoCket-Head Cap Screws, ASTM International, West Conshohocken, PA, 2017, [www.astm.org](http://www.astm.org)

The torque values were calculated from the formula  $\text{Torque} = K \cdot D \cdot F$ , where K is the torque coefficient, D is the nominal diameter, and F is the Clamping Force. Here, the torque coefficient is chosen as K=0.17 as it is used for thread locking nuts and we are using nylon thread lock nuts.

Tightening torque can be measured using two methods. The static measurement is where the torque is measured after the tightening is completed. It is one of the common methods to check tightening torque where a torque wrench equipped with a clutch that can be pre-adjusted to a specific torque. If the torque is greater than the pre-set torque value, the clutch will release with a click and if the torque is less we can still tighten the nut until the wrench clicks. The UTHT team will be using a torque wrench to ensure the right tightening torque is given to the nuts.

### Manufacturing (Quality Assurance)

Machining Standard for Carbon Fiber Tubes - ID6:

#### **Cutting<sup>4</sup>**

The materials required for the safe cutting of carbon fiber tubes are a diamond coated abrasive cut-off blade, safety glasses and masks for cutting personnel and for the finishing of the tubes, sandpaper. The blade used for cutting should be a diamond coated abrasive and a toothed tool cannot be used. Teeth from a toothed tool can result in delamination and splintering. The surface of the blade needs to cool before to prevent the activation of epoxy. For reduction of heat, segmented blades can be used. For cutting the tube, the tube needs to be secure via support for a clean cut. Once the cutting is done, the edge of the tube needs to be cleaned. Sandpaper can be used for sanding the burrs and fibers. The problems that can occur during the cutting are delamination, burring and overheating. To avoid delamination, the pace of cutting should be slow or for thicker tubes thinner blades with finer grits can be used. Burring can be avoided by providing adequate support to both ends of the tube. Clamps for better support should be used. Overheating can be avoided using a coolant for reduction of heat from the cutting process.

#### **Drilling<sup>5</sup>**

For drilling, higher RPM and sharper tools are preferable. HSS drill bits are used for high speed but can be used to make 5-10 holes if they become dull soon and if the tool is dull the heat production during the drilling starts increasing. Drill bits which have carbide or diamond coating last longer, 30 holes approximately.

---

<sup>4</sup> Clearwater Composites. 2021. *How to Cut Carbon Fiber* | Clearwater Composites, LLC. [online] Available at: <<https://www.clearwatercomposites.com/resources/how-to-guides/cut-carbon-fiber-tubes/>> [Accessed 6 March 2021].

<sup>5</sup> Latteier, P., 2021. *The Good, Better and Best Ways to Drill Carbon Fiber - Elevated Materials*. [online] Elevated Materials. Available at: <<https://www.elevatedmaterials.com/drilling-carbon-fiber/>> [Accessed 6 March 2021].

## Machining properties of CFRP<sup>6</sup>

Major determinant of a perfect hole in a composite is the spindle speed. 20,000 rpm speed was observed as the best speed for the spindle for a clean cut according to [1]. The feed rate determines the push out delamination and can be reduced by reducing the feed rate. The best combination of feed rate and spindle speed for thin CFRP laminates was observed at 0.137 mm/rev and 2,000 rpm, respectively.

During the process of machining, all electronics need to be covered as carbon dust is created during the machining process and is conductive in nature. Hence, there is a possibility that electronics may fry if not covered properly.

## Tests

### Tolerance Test - ID1

Due to cost constraints, the majority of tests conducted by the structures system on the entire pod will be non-destructive. The main non-destructive test is a tolerance test. The tolerance test will be used to ensure the gap between the two halves of the shell are flesh with each other. The pass criteria of this test is to ensure there is no greater than 0.01mm (based on accuracy of calipers used for measuring) between the two halves. This value is chosen to be as low as possible to avoid any unnecessary drag from airflow seeping through the gap. Shall this test fail, foam will be placed to aid in closing the gap between the two halves of the shell.

### Carbon Fiber Tube Test - ID5

Due to the bonding of tubes via epoxy and prepreg (using the 3D inserts) being too complex to simulate with acceptable accuracy, a bending test will be conducted to determine the strength of these joint tubes. Should this connection be able to withstand a specific loading (TBD through simulations), it will be deemed as safe and the entire frame can then proceed with manufacturing.

### Future Planned Tests

Other planned non-destructive tests include vibration tests and loading tests, which will continue to be researched prior to June 2021.

---

<sup>6</sup> Reference: Krishnaraj, V., Prabukarthi, A., Ramanathan, A., Elanghovan, N., Senthil Kumar, M., Zitoune, R., and Davim, J. P., “Optimization of machining parameters at high-speed drilling of carbon fiber reinforced plastic (CFRP) laminates,” Composites Part B: Engineering, vol. 43, 2012, pp. 1791–1799.

## **Failure Mitigation**

### Two Halves of the Shell - ID1

In addition to the Tolerance Test described in Section ST4.3.1, the design of the two halves of the shell mitigate the risks from ID1 in Table ST4. From Figure ST4, it can be seen that the front half of the shell will lie on a “lip” on the back half of the shell. This was designed such that during operation, the aerodynamic forces on the front half of the shell would cause it to push against the “lip”, causing the two halves to stay in the acceptable tolerance zone. This is a redundant feature added to our shell to aid in the compression of bolts to secure the two halves together.

### Identify Critical Dimensions - ID9, ID10

There are two major areas with respect to the structures subsystem that are significant for the forthcoming assembly that is going to happen on the frame. Any dimensional deviation or position errors in this dimension during manufacturing will stop the assembly until corrected. Hence it is very important to assure certain critical dimensions before proceeding to assemble components on the structure.

#### **Area 1: Center Structure Dimensions**

The manufacturing of the center structure and its symmetry with respect to overall of the pod is significant. Any deviation in the center structure width, height and distance from the end of the pod will affect the subsequent assembly operations. Hence we have identified a few critical dimensions that will be ensured during the manufacturing process and all necessary steps and actions will be taken to adhere to the same. Below is the sketch of the frame with critical dimensions.

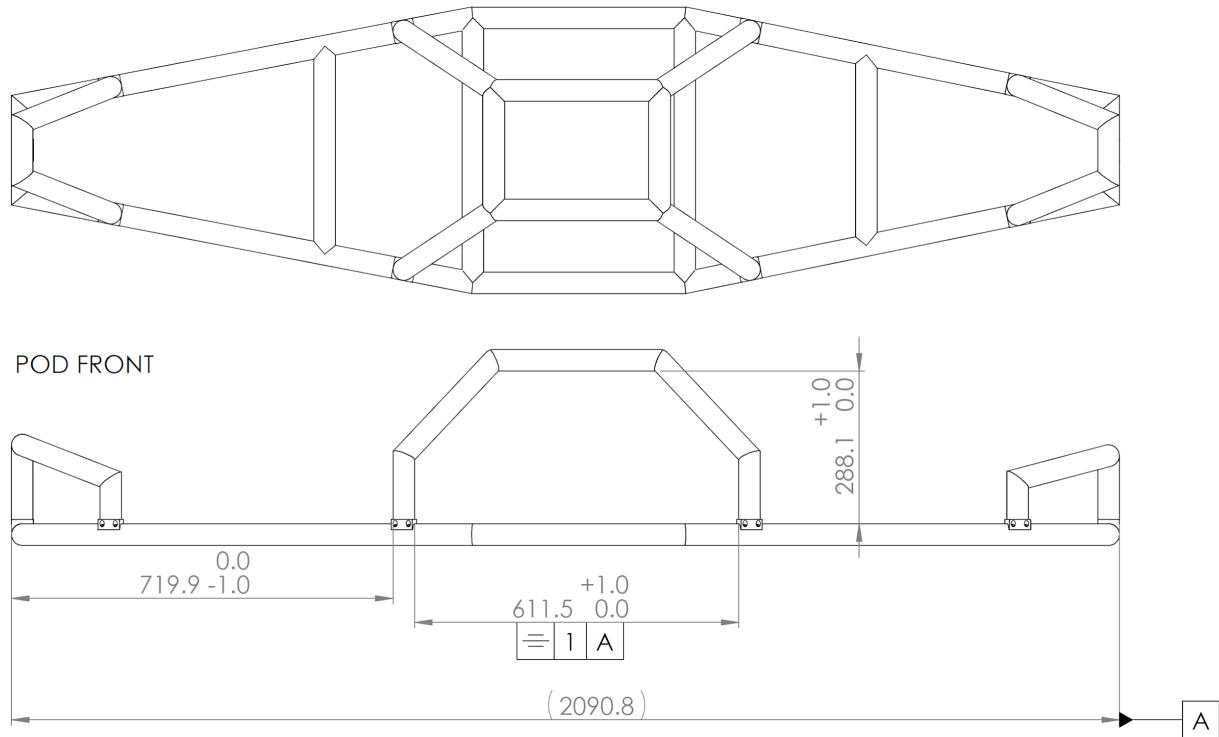


Figure ST29: Critical Dimensions of the Frame

### Area 2: Hole Position Error in the Pod Floor Plate

In the Floor Plate of the pod, there are several mounting holes on which various components such as suspension, propulsion, frame, braking components and other sensors are mounted. Hence, any shift in the position of these mounting holes will cause significant alteration to the subsequent assembly. The plate is cut through a VMC in a single setting which will cut the outer profile and also the cutouts involved. The plate itself will look like the below image after VMC operations are done.

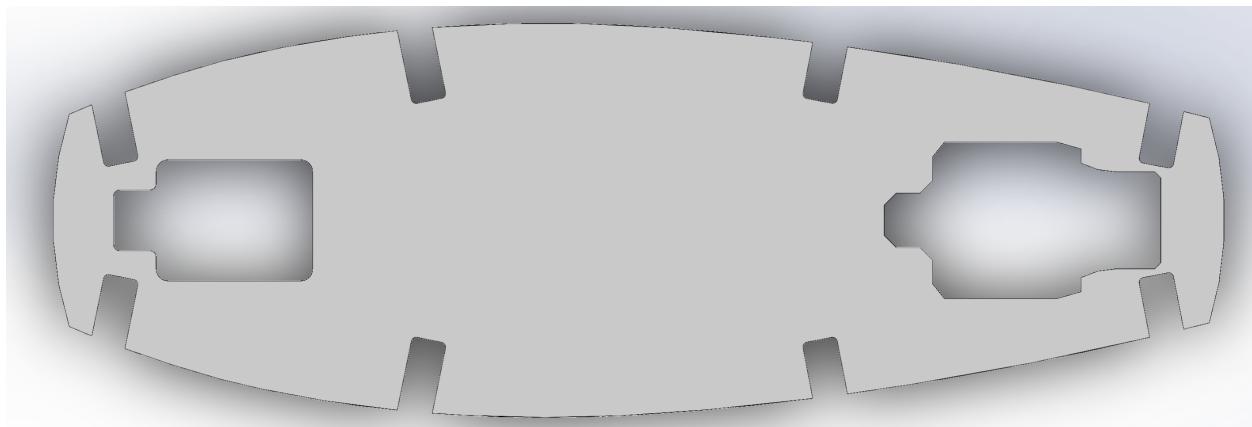


Figure ST30: Sheet Metal after VMC Operations

After the plate has been cut, the fixture and the holding setup of the plate will be changed in order to accommodate all the holes that have to be drilled on the plate. The initial primary holes that connect the frame and the propulsion units are identified and these are checked with reference to certain edges which can be used during manufacturing to take reference from. Below is the sketch with the identified critical dimensions. The holes that are marked with a red circle will be drilled first, checked for the dimension and the position tolerance will be ensured. Once it is assured, we will be proceeding further with drilling all the other holes.

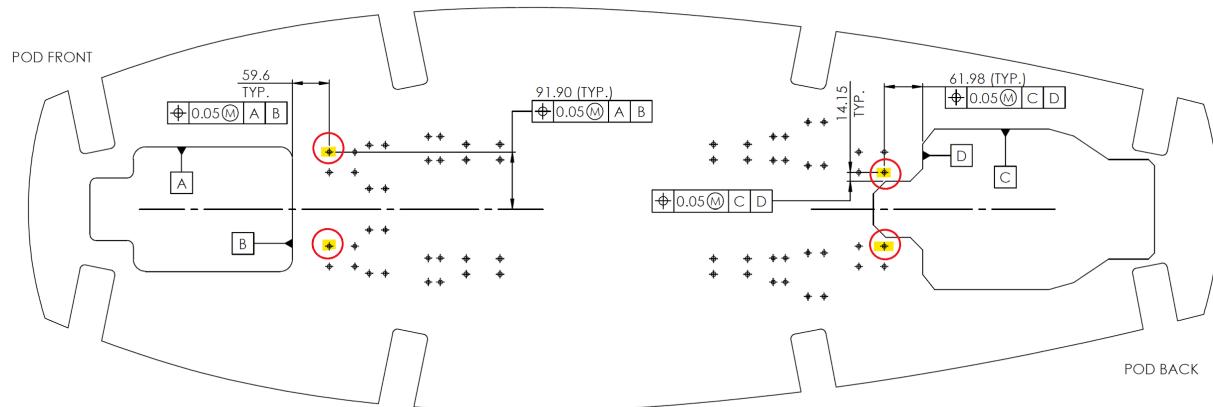


Figure ST31: Sheet Metal with Identified Critical Dimensions

## **Propulsion**

### **Overview of Propulsion**

In the final demonstration of the propulsion subsystem, the team will first provide a part list, in BOM format, including the quantity, dimension, material, and functionality of each component. Next, the team will provide a detailed description of how these components form each sub-assemblies involved in the propulsion of UTHT Pod I. The reliability and general safety of this subsystem will be demonstrated through a Failure Modes and Effects Analysis (FMEA), which the team has conducted to identify the major safety risks that require failure mitigation strategies and mechanisms. These strategies will be outlined below and followed by the methods of testing for the major components within each sub-assembly, to be performed prior to EHW 2021. In this section, the team will give an overview of the propulsion subsystem.

The propulsion of the UTHT Pod I is achieved by an in-wheel drive system which is composed of two main subsystems, the Front Wheel Assembly and Rear Wheel Assembly (see Figure P1). As outlined in the sections below, a pair of shock absorbers are utilized in both assemblies to mitigate the vibration during the run. To ensure proper functionality of the shocks, anti-rotation plates have also been used in both assemblies to prevent unwanted motion of the pivot arms. To further enhance the stability and provide extra traction force of the propulsion system, the front and rear wheel assemblies incorporate 4 pairs of idler wheels. These idler wheels are clamped onto the top flange and web of the I-beam track to achieve both vertical and lateral stability. For further details, please refer to the stability system section. The propulsion of the pod is achieved by an electric motor mounted directly to an inner aluminum 6061-T6 wheel core of the rear drive wheel. The pod will start propelling at around 0.2 g, whereby it will keep accelerating to reach the desired velocity. The pod also incorporates a braking system, which utilizes a control system to trigger the clamping of braking pads on the top flange of the track. This will decelerate the pod at around 0.7 g once the braking distance is reached or in the case where emergency brakes are needed. A more detailed breakdown of this system can be found in the braking subsystem section.

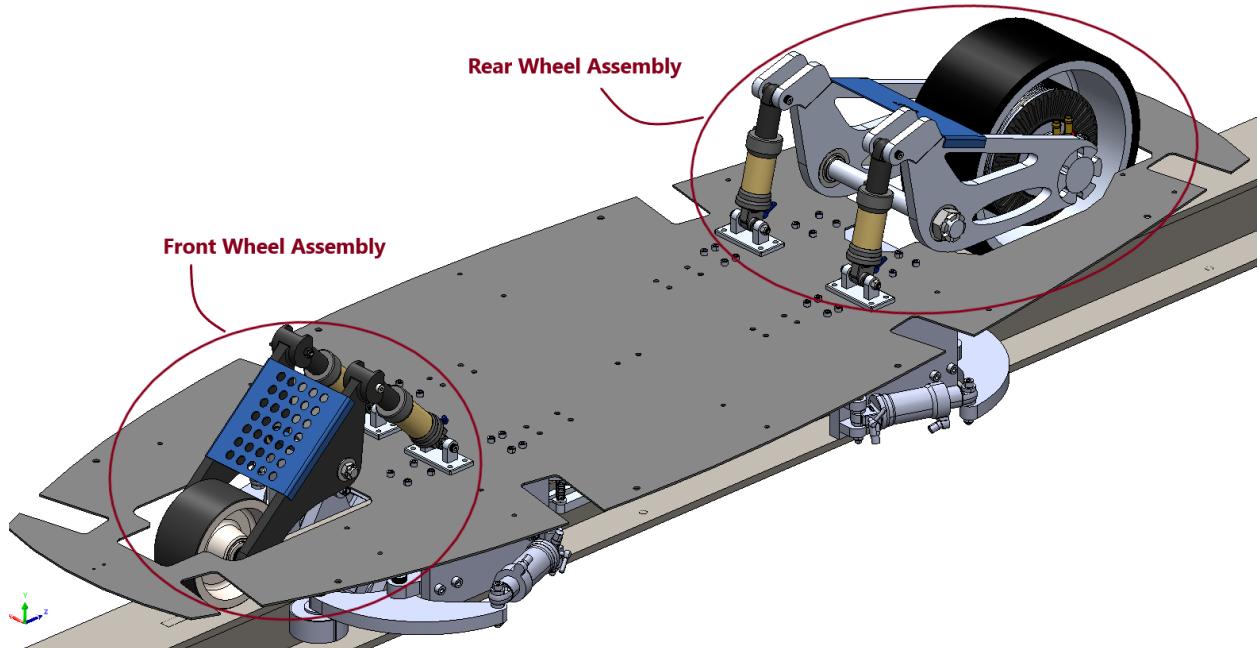


Figure P1: Overview of the Propulsion System's components

The UTHT engineers strived to ensure that the end product is a subsystem that is safe, high performing, and compact. A design for manufacturing was utilized and components like the suspension arms were designed following the ANSI Imperial dimensioning standard because that is what Canadian manufacturers use, while some components like the shock absorbers, which are going to be bought off the shelves, follow the ANSI Metric standard and so our designs had to be optimized for both dimensioning standards wherever two components attach to each other.

During the design phase, the team focused on designing phenomenal components with a factor of safety no lower than 2. This allowed the team to leverage the exceptional mechanical capabilities of a propulsion subsystem while minimizing any risk to uphold safety as a top priority.

## Propulsion Components

Tables P1 and P2 outline the main components of the propulsion system for front and rear drive assemblies respectively. The tables are accompanied with Figures P2 and P3 to indicate the specific locations of each part in the pod.

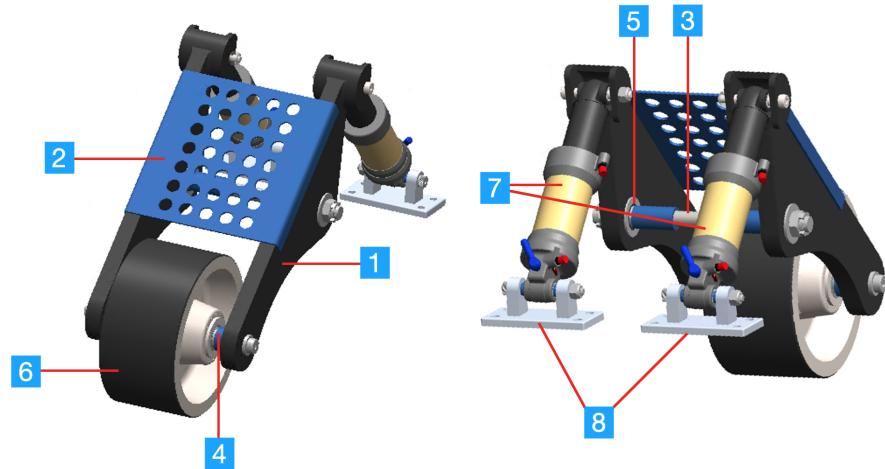


Figure P2: Locations of main components for front-drive assembly

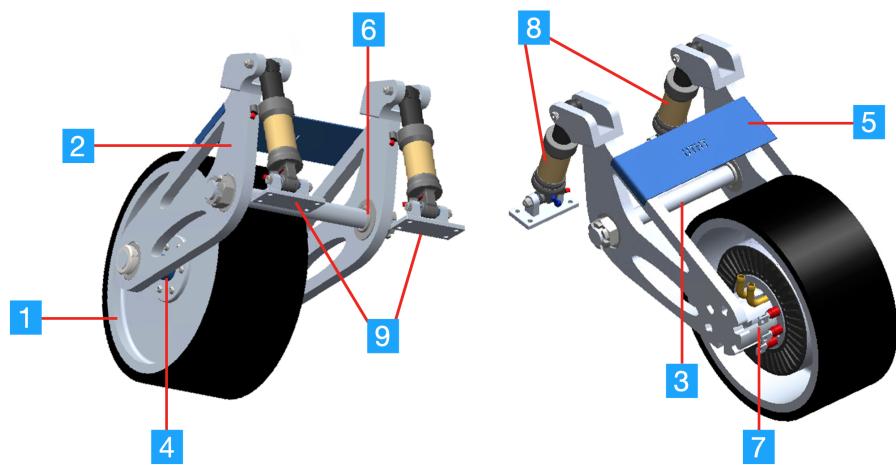


Figure P3: Locations of main components for rear-drive assembly

Table P1: Propulsion system main components for front-drive assembly

No.	Component	Qty.	Dimension	Material	Functionality
1	Suspension Arms	2	Length: 12.71in Width: 3.93in Thickness: 1.70in	Aluminum 6061-T6	Work in conjunction with shock absorbers to manage smooth ride and steering.
2	Front Anti Twist Plate	1	Length: 8in Width: 8in Thickness: 1in	Aluminum 6061-T6	Prevents unwanted rotation.
3	Connector Shaft	1	Diameter: 0.5in Length: 7.2in	Alloy Steel	Connects and holds two suspension arms together.
4	Shaft	1	Diameter: 0.75in Length: 4.75in	Alloy Steel	Used as an axis for the wheel.
5	Ball Bearing	4	Shaft Dia: 0.5in Width: 0.375in	Steel	To transfer load from the lateral wheel to the shock absorber.
6	Polyurethane Wheel with Steel Core	1	Diameter: 6in Width: 3in Axel Dia: 0.75in	Polyurethane Rubber and Steel	Used as the front wheel.
7	DNM Shock Absorber	2	Diameter: 46mm Height: 193.5mm	----	Damps the forces and disturbances that encounter the pod during its motion.
8	Clevis	2	Length: 1.65in Width: 1.39in Height: 3.44in	Aluminum 6061-T6	Holds the shock absorber in place.

Table P2: Propulsion system main components for rear-drive assembly

No.	Component	Qty.	Dimension	Material	Functionality
1	Wheel Core	1	Diameter: 279.4mm Height: 127mm	Aluminium 6061-T6 (SS)	Used as the rear wheel.
2	Suspension Arms	2	Depth: 50.8mm Length: 391.6mm Height: 225.9mm	Aluminium 6061-T6 (SS)	Supports the wheel core and the motor at the wheel by connecting with the wheel adapter shaft.
3	Pivot Shaft	1	Diameter: 1in Length: 10.8in	Alloy Steel	Connects the suspension arms, and supports their weight.
4	Wheel Adapter Shaft	1	Shaft Dia: 1in Plate Dia: 3.46in Length: 3.05in	Aluminium 6061	Supports the wheel along with the left suspension arm.
5	Anti-rotation plate	1	Height: 76.2mm Width: 266.4mm	Aluminium 5052-H34	Offers support against torsion.
6	Ball Bearing	5	Bore Dia: 1in Outside Dia: 2in Overall Width: 0.75in	52100 Bearing Steel	Supports Radial load on the pivot shaft (part from motion Canada)
7	Motor Spacer	1	Internal Dia: 46mm Outer Dia: 74mm Height 44.5mm	Aluminium 6061-T6	Separates the motor from the right suspension arm in order to reduce impact due to contact.
8	DNM Shock Absorber	2	Diameter: 46mm Height: 193.5mm	----	Damps the forces and disturbances that encounter the pod during its motion.
9	Clevis	2	Length: 1.65in Width: 1.39in Height: 3.44in	Aluminum 6061-T6	Holds the shock absorber in place.

## Rear Assembly

As shown in Figure P4, the main components which make up the Rear Wheel Assembly are the wheel core, pivot arms, shock absorbers, and an electric motor (EMRAX 188). Through the use of a motor spacer and a wheel adapter shaft, the pivot arms will connect the shock absorbers to the rear wheel system.

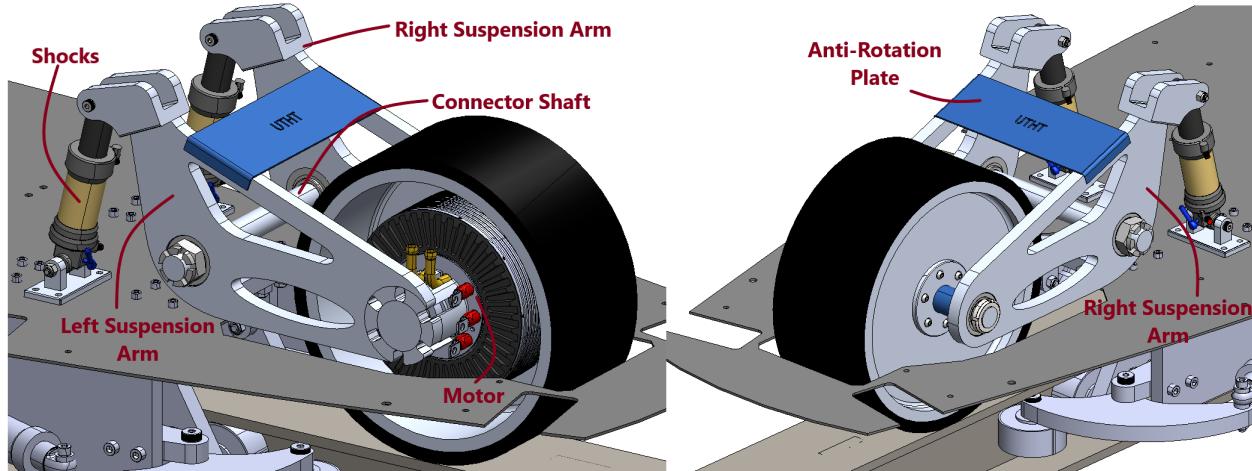


Figure P4: Captures of the rear wheel assembly.

### Electric Motor

The motor used in the pod is an in-wheel electric motor, model 188 produced by EMRAX. The motor is an out-runner, meaning it is the opposite to a conventional electric motor by having the stator surrounded by the rotor. It has a 188 mm casing diameter and weighs 7 kg. It supports peak continuous power and peak continuous torque values of 30 kW and 90 Nm, respectively, as well as supporting overall peak power and peak torque values of 52 kW and 90 Nm, respectively. Summarized mechanical and electrical data sheets can be seen in Table P4 below:

The EMRAX 188 motor features an in-wheel drive mechanism for optimal space efficiency. The motor is directly mounted to the rear wheel and is connected to the left suspension arm via the motor spacer and provides 50 Nm of continuous torque.

Table P4: Mechanical and Electrical Specifications of the motor

Mechanical		Electrical	
Type:	Axial Flux motor / generator	Maximum Battery Voltage:	430 (HV) / 300 (MV) / 110 Vdc (LV)
Casing Diameter:	188 mm	Peak Power (at 6500 RPM):	52 kW
Axial Length:	77 mm	Continuous Power*:	up to 30 kW
Dry Mass:	7,0 kg (AC) / 7,2 kg (CC) / 7,3 kg (LC)	Peak Torque:	90Nm
Stator Cooling:	Air (IP21) / Combined (IP21) / Liquid (IP65)	Continuous Torque*:	up to 50 Nm
Mounting:	Front: 6x M6 threaded holes Back: 8x M6 threaded holes	Efficiency:	92-98%
Stacking:	Two motors can be stacked together to achieve doubled power / torque.	*Subject to drive cycle, thermal conditions, and controller capability.	

### Rear Wheel

The rear wheel consists of a core made of Aluminum 6061-T6 and a coating made up of castable polyurethane shore A80 with an outer diameter of 12 in and a width of 5 in. As shown in Figure P4, the rear wheel also has a hollow section to mount the motor. The core has an asymmetrical geometry to balance the center of mass when assembled to the electric motor. Once the core is made, it will be placed into a pre-made Medium Density Fibreboard mold for the polyurethane coating. The polyurethane mixed with hardener will be casted into the mold and bond to the core with metal-polymer adhesives.

### Rear Suspension

The two rear suspension arms are made of a 0.6in thick Aluminum 6061-T6 sheet and are connected to the connector shafts in between the shocks and the rear wheel. Two identical adjustable air shocks are used in the rear suspension assembly. The shocks are connected to the floor of the body and to the rear arms which in turn are connected to the rear wheel. The rear anti-rotation plate is 9 x 3 inches and is made from a 2mm thick Aluminium 6061-T6 sheet bent to create a plate that sits between the suspension arms and prevents their unwanted rotation. Figure P4 shows an overview of the components connected to the rear wheel.

## Front Assembly

As shown in Figure P5, the main components which make up the Front Wheel Assembly are the wheel core, pivot arms, front anti-rotation plate, and shock absorbers with the pivot arms connected through the use of a steel shaft.

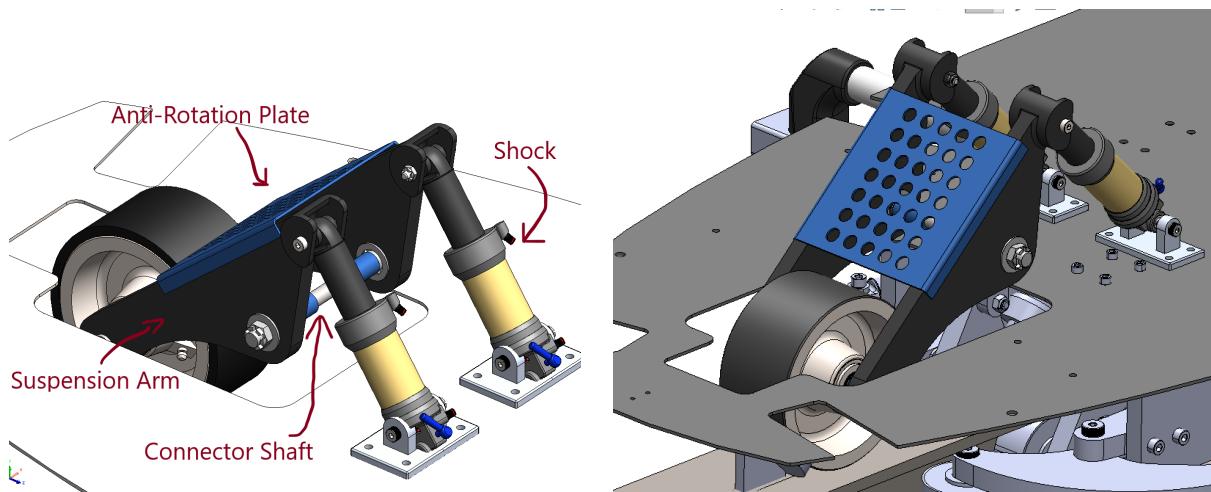


Figure P5: Captures of the front wheel assembly

## Front Wheel

The front wheel has an overall outer diameter of 6 in, a width of 3 in, and capacity of 2000 lbs. The front wheel is made of Alloy Steel and has a coating made of polyurethane with a hardness of Shore A 95.

## Front Suspension

The suspension arms on the front are also made of 0.6in thick Aluminum 6061-T6 sheets. As seen in Figure P5, the suspension arms are connected to the connector shafts in between the shocks and the front wheel. Two adjustable air shocks are used with an eye-to-eye distance of 165 mm, shocks can travel 35 mm and they weigh 395 g. For further details of shocks, refer to the Stability section. The shocks are connected to the floor of the body and also to the two front suspension arms. The front anti-rotation plate is 6x6 in and is made from a 2 mm thick Aluminium 6061-T6 sheet with circular cutouts to reduce weight. The sheet was bent to create a plate that sits between the suspension arms and prevents their unwanted rotation. (see Figure P5 - right).

## Front and Rear Assemblies Simulations

Finite Element Analysis (FEA) static analysis was conducted on major propulsion components such as the drive wheel, suspension arms, shafts, etc. Loading conditions were chosen based on the estimated pod weight (200kg/2000N), and the motor torque (50Nm) of the EMRAX 188 electric motor. The simulation assumed that the weight of the pod is evenly distributed to the 4 suspension arms (i.e. 500N per suspension arm) and the attached components. The resulting safety factors of each analyzed component will be presented to justify their geometry design and material selection. Simulated parts are labeled in Figures P6 and P7 below.

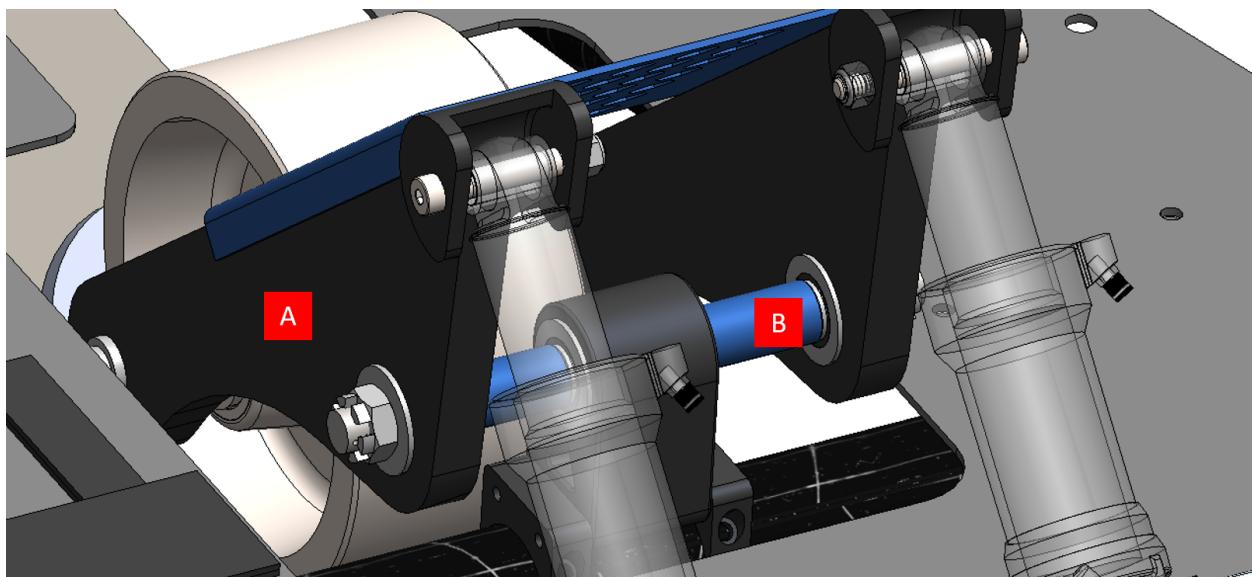


Figure P6: Simulated parts from the front assembly as labeled

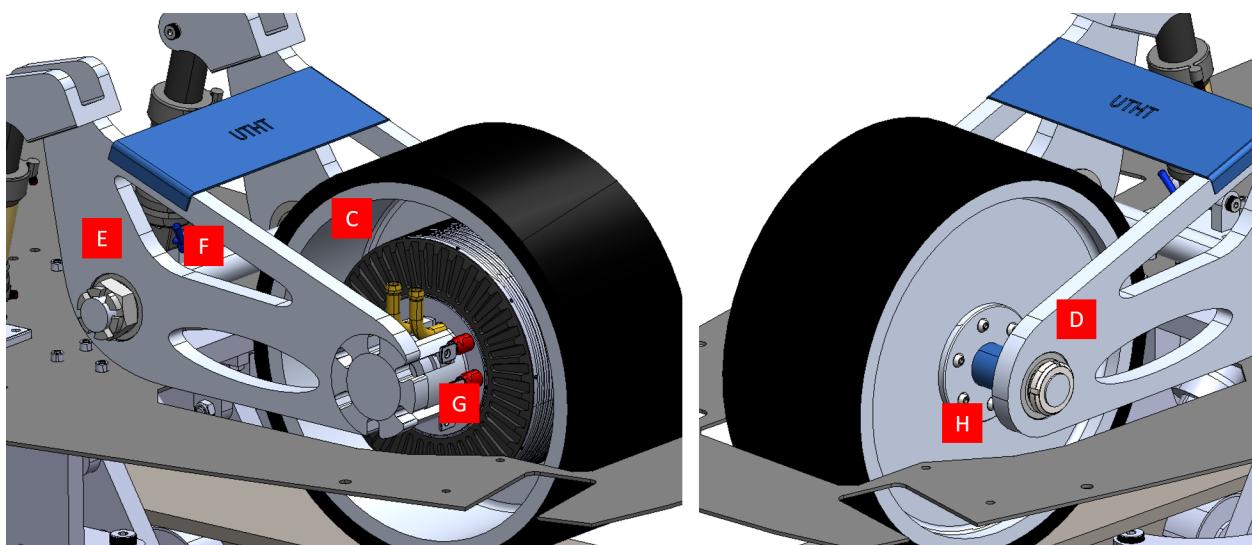


Figure P7: Simulated parts from the rear assembly as labeled

### A. Front Suspension Arm - LHS & RHS (mirrored) (FOS=14)

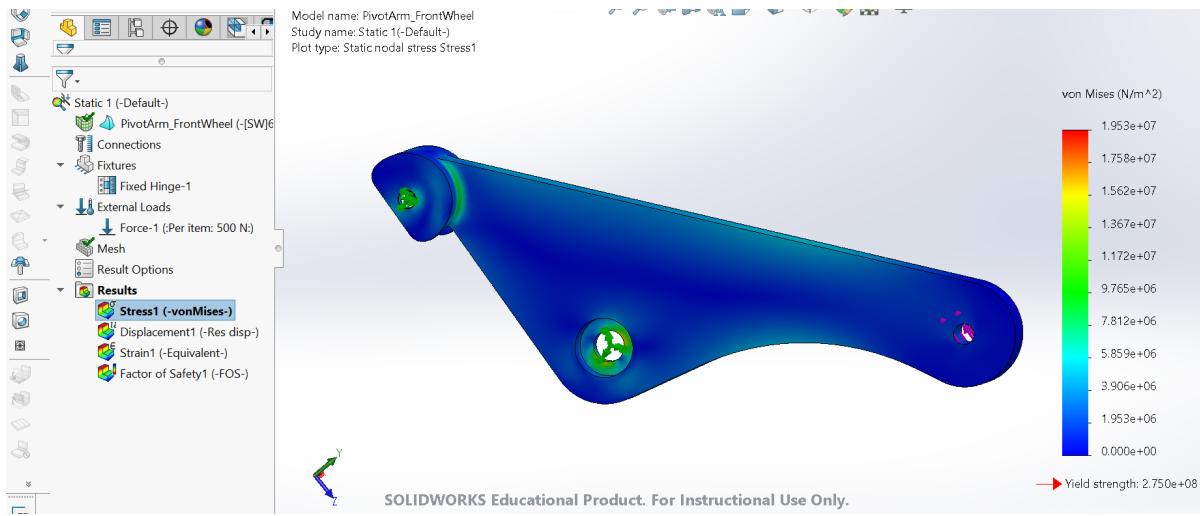


Figure P8: Simulation result of the front suspension arms

The front suspension arms pivot around the connector shaft and connect the front wheel to the shocks for damping. These arms are made of Aluminum 6061-T6. This simulation has considered the weight of the pod by fixing the center pivot and top left attachment (where the shock is connected to) and applying a bearing load in the wheel shaft slot. The maximum stress was found to be 19.5 MPa.

### B. Front Connector Shaft (FOS = 2.7)

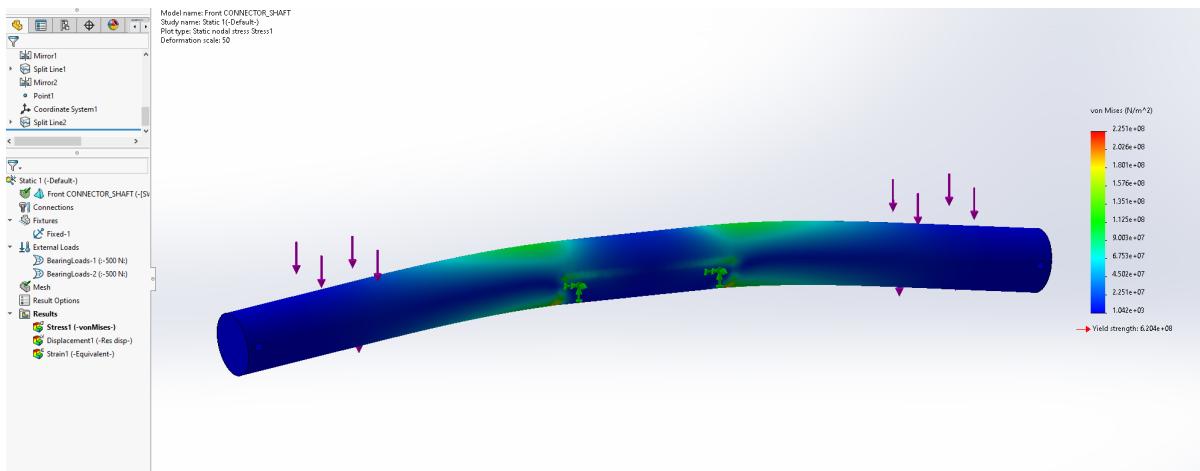
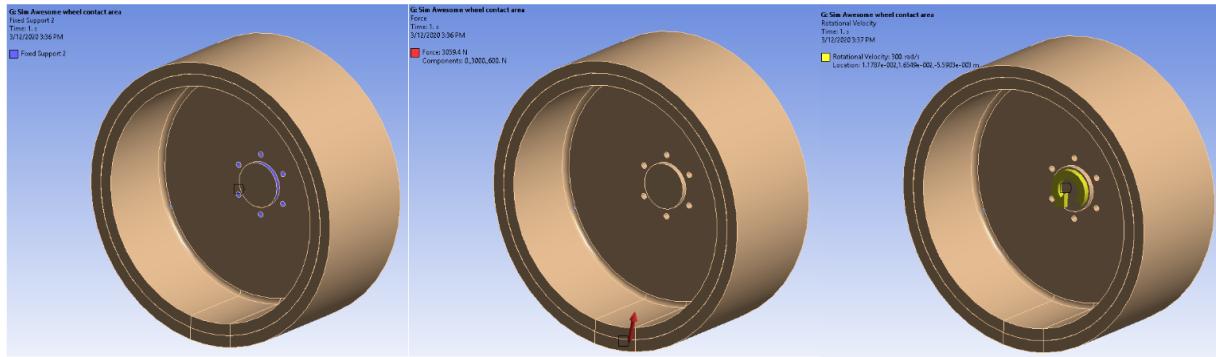


Figure P9: Simulation result of the front connector shaft

The front connector shaft pivots the suspension arms and connects them to the frame of the pod. The material chosen for this shaft is alloy steel. Simulation considered the bearing load at both ends exerted from the suspension arms. Maximum stress is 225 MPa around the pivot.

### C. Drive Wheel (FOS=10)



Fixed Support: Fixed on where the motor is mounted.

Force: Forces applied representing pod weight (3000N) and traction (600N)

Rotational velocity: centrifugal load at high speed (300 Rad/s)

Figure P10: Drive wheel loading conditions

A vertical load of 3000N was applied to represent double the pod weight (for the conservatism of the analysis results), as well as a traction force of 600N and a rotational velocity of 300 rad/s to induce a centrifugal load (both of these are also doubled from the actual expected values to maintain conservatism of the analysis results). A fixed support was only applied to the location where the motor is secured to the rest of the drive system assembly.

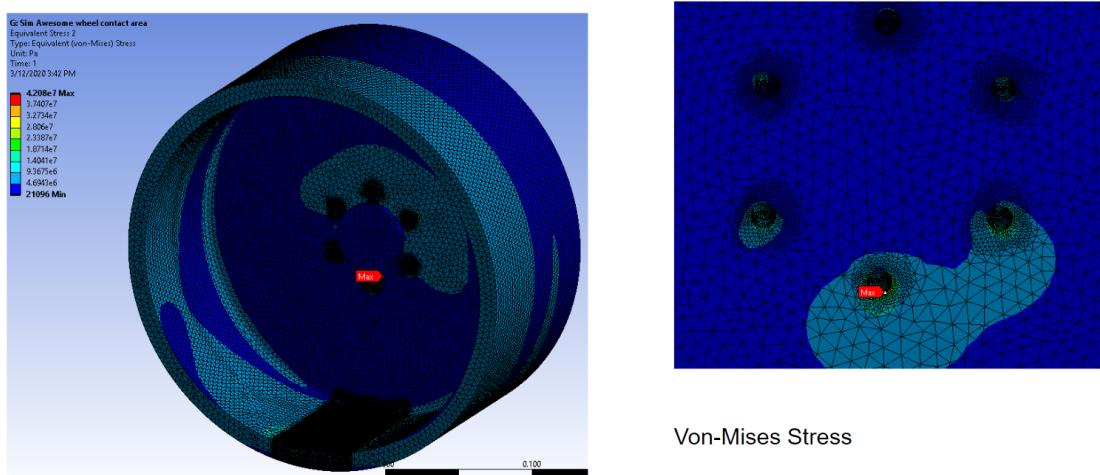


Figure P11: Simulation result of the drive wheel

A peak von Mises stress of approx. 42 MPa was observed on the wheel and was observed near the center of the wheel where the bending (due to the wheel's asymmetric cross-section) is initiated from. Being made from Aluminum 6061-T6, these stresses are not an issue, as its yield strength is approximately 240 MPa. A factor of safety greater than 10 is determined on the drive wheel.

## Drive wheel modal analysis:

The natural frequency of the rear wheel was analyzed as it will be coupled directly to the EMRAX 188 motor. If this motor were to vibrate, the wheel would experience that same vibration and it is important to understand its natural frequency to ensure that it is greater than whatever frequencies the wheel may actually be exposed to in order to avoid mechanical resonance (it would cause a catastrophic failure if it occurred).

The below plot shows that the frequencies of the first 4 harmonics are as follows:

Mode 1: 621.84 Hz Mode 2: 621.88 Hz Mode 3: 968.94 Hz Mode 4: 968.99 Hz

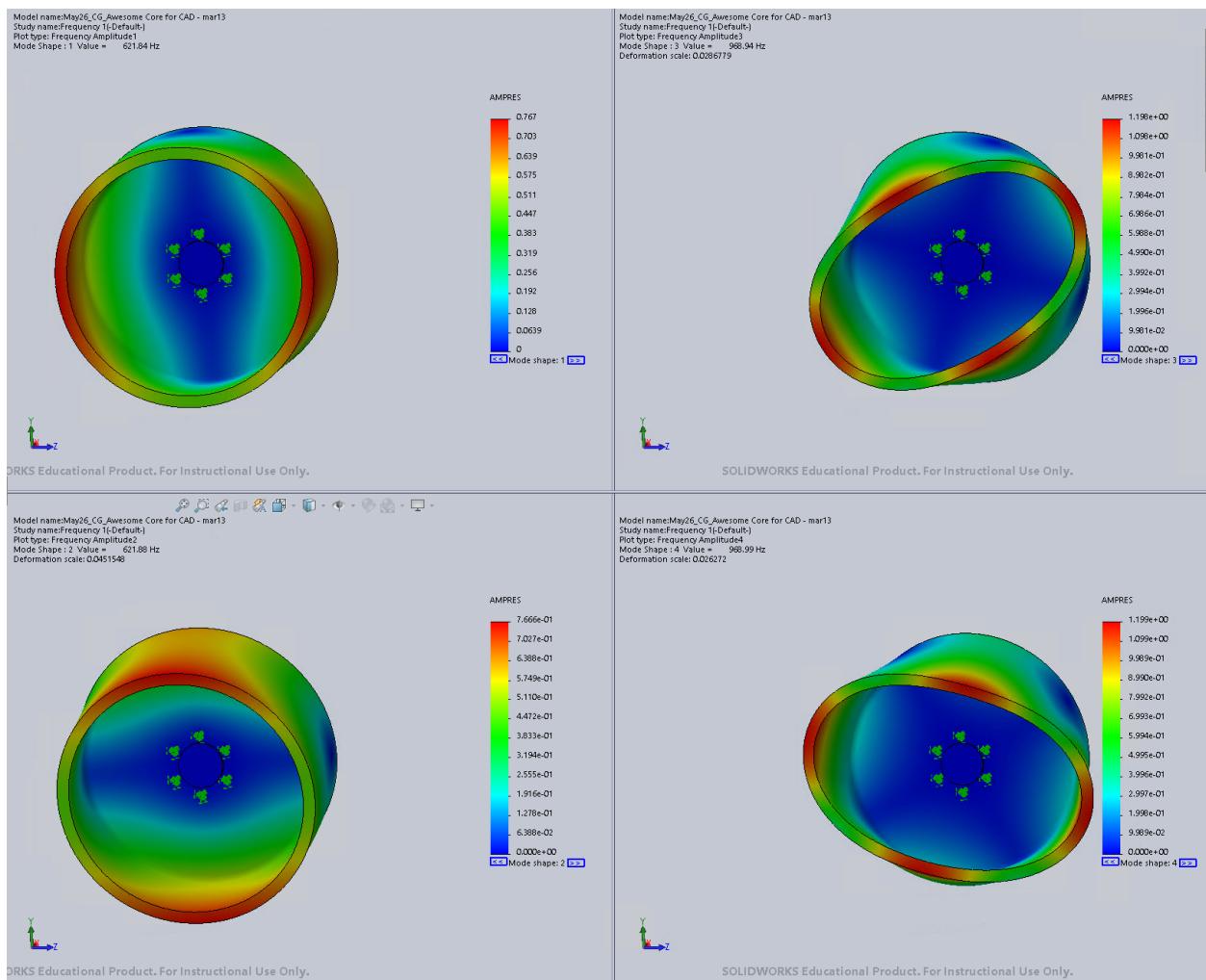


Figure P12: Modal analysis result of the drive wheel

As stated on page 4 of the EMRAX motors user manual a vibration-free experience is guaranteed with the EMRAX 188. That said, if there was a slight asymmetric loading onto the motor shaft, that could result in vibrations that the wheel would experience. With a maximum rotational speed

of 8000 rpm, an asymmetric loading on the motor could cause vibrations of approximately 133.33 Hz [  $(8000 \text{ cycles / min}) / (60 \text{ sec / min}) = 133.33 \text{ cycles / sec} = 133.33 \text{ Hz}$  ].

Since the analysis showed the natural frequency of the wheel to be 621.84 Hz (greater than 133.33 Hz by a factor of 4.6X), it is not anticipated that the wheel will endure a vibratory frequency that can induce mechanical resonance.

#### D. Rear Suspension Arm -LHS (FOS=42):

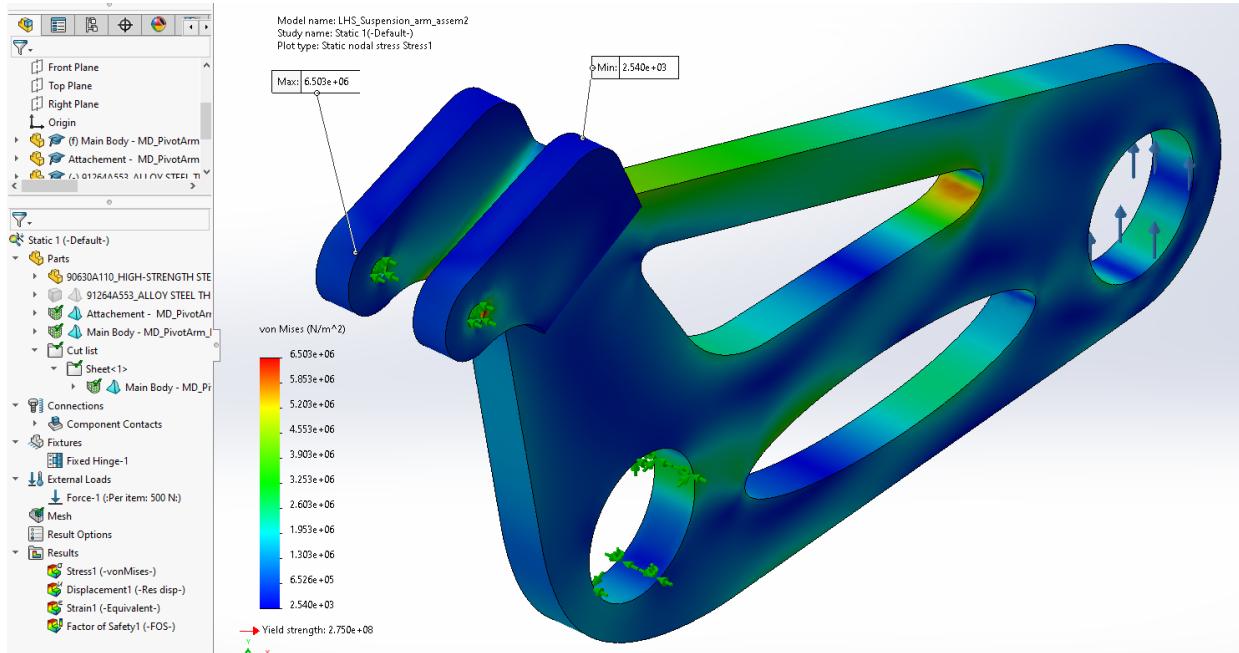


Figure P13: Simulation result of the rear suspension arm - LHS

The rear suspension arms pivot around the rear pivot shaft and connect the drive wheel to the rear shocks. These arms are made of Aluminum 6061-T6. This simulation has considered the bearing load from the weight of the pod. The LHS suspension arm supports the wheel through a bearing around the wheel adapter shaft (H). The maximum stress found on the left hand side was 6.5 MPa.

### E. Rear Suspension Arm - RHS (FOS=28):

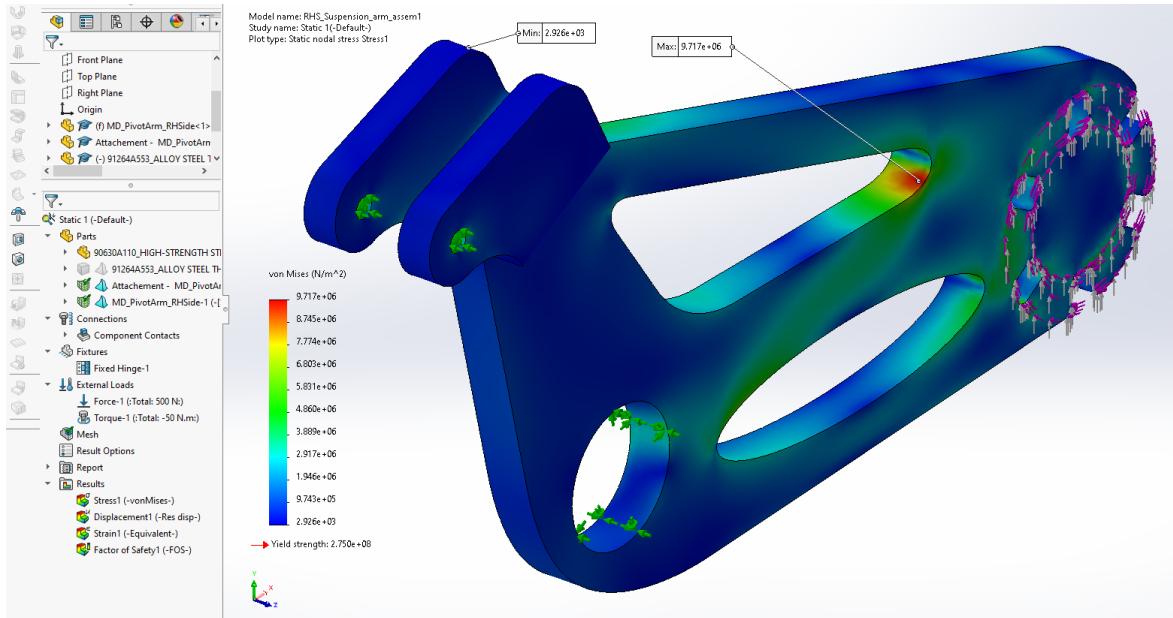


Figure P14: Simulation result of the rear suspension arm - RHS

Unlike the LHS suspension arm, the RHS arm connects the motor stator through welding to the motor spacer (G). Thus, instead of bearing load, a shear force was applied to the welded area. The maximum stress found on the RHS arm was 9.7 MPa.

### F. Pivot Shaft (FOS=5.2):

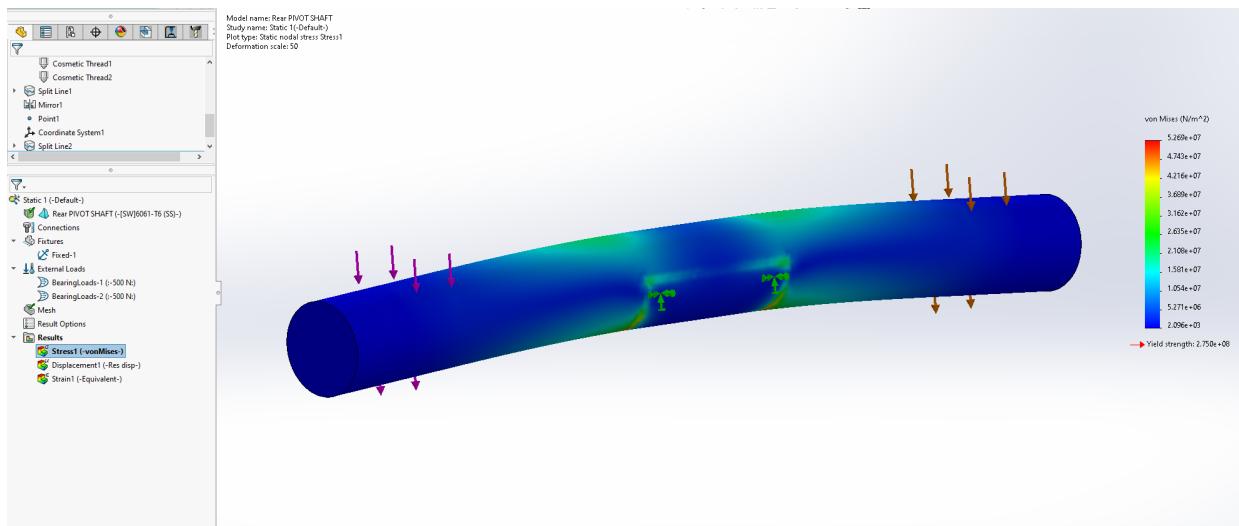


Figure P15: Simulation result of the rear pivot shaft

The rear pivot shaft connects the rear suspension arms to the pivot, which is mounted to the frame of the pod. The material chosen for the pivot shaft is Aluminum 6061-T6. This simulation

has taken into account the weight of the pod. The maximum stress found on the shaft was 52.7 MPa around the pivot.

### G. Motor Spacer (FOS=13):

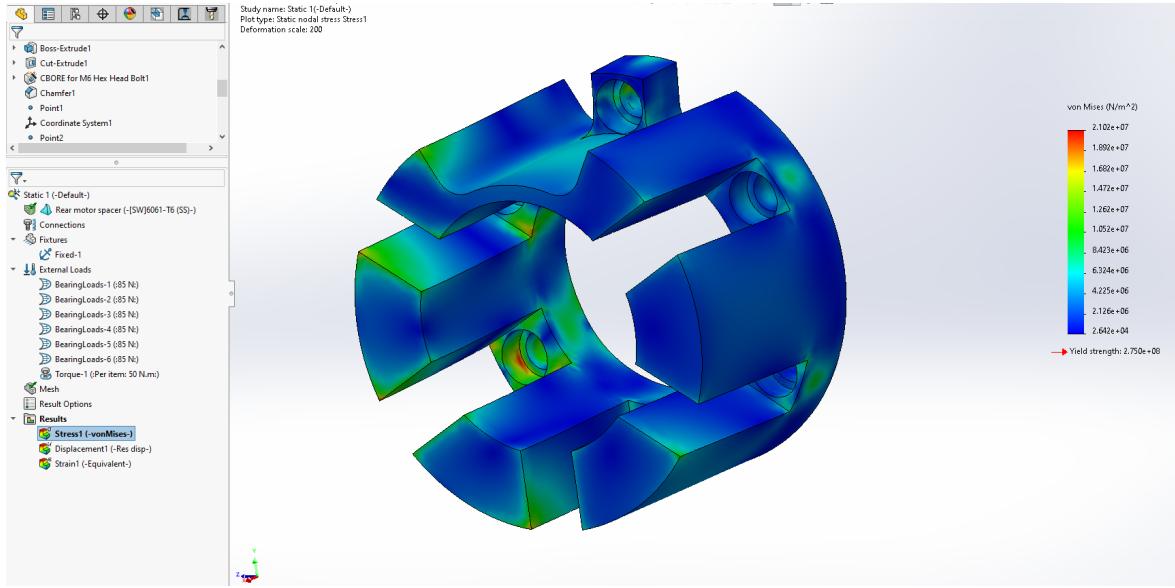


Figure P16: Simulation result of the motor spacer

The motor spacer is welded to the RHS suspension arm to connect it to the motor stator through the bolts. The material chosen for the motor spacer is Aluminum 6061-T6. Both the weight of the pod and torque of the motor have been considered in the simulation. The maximum stress was 21MPa.

## H. Wheel Adaptor Shaft (FOS=4.2):

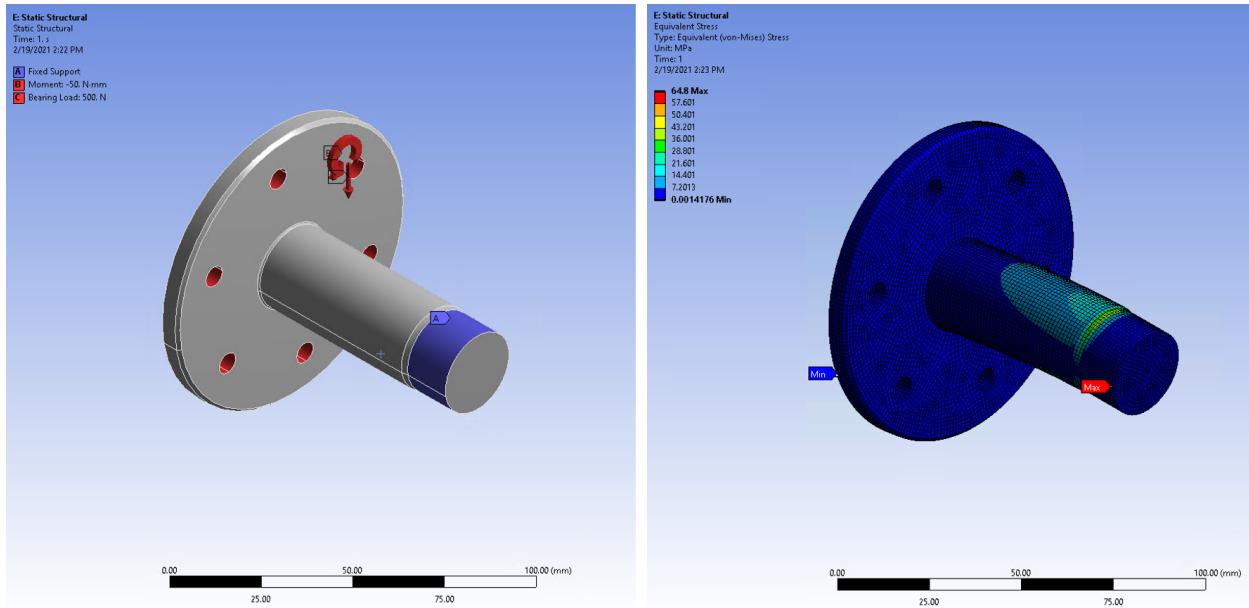


Figure P17: Simulation result of the wheel adaptor shaft

The wheel adapter shaft connects the drive wheel to the LHS suspension arm. The weight of the pod and torque transmitted from the motor were taken into account. The shaft is made of Aluminum 6061-T6 and the maximum stress was found to be 64.8MPa.

## Safety

The safety of the propulsion system will be demonstrated through an FMEA analysis as summarized in Table P5. Major failure modes and mitigations are discussed after the FMEA in the Failure Mitigation section, followed by the testing protocols of the propulsion system.

Table P5: FMEA of Propulsion

Process/Mechanism	Safety Measure	Effects	Causes	Prevention Control	Detection Control	Severity Rating (S)	Frequency of Occurrence (O)	Ease of Detection (D)	Risk Priority Number*
Spring Failure of any mode of suspension	Spring failure	POD will be damaged due to undamped vibrations	Microfractures - Fasteners failing - Improper tune up	Pre-test run - Check the tune-up and equalize across all springs	Any path deviation - Height sensors	9	2	3	54
Front/Rear Wheel Suspension	Shock absorber failure	POD will be damaged to due undamped vibrations	Improper pressurization of shock absorber - valve caps coming off - Materials failure	Pre-test run - check the pressure and equalize in all shocks - Routinely check valve caps	When the pod sags to the I-beam - Pressure sensors on the shock absorber - Height sensors	9	2	3	54
Idler Wheel Suspension	Shock absorber failure	POD will be damaged to due undamped vibrations	Improper pressurization of shock absorber, valve caps blow off, Materials failure	Pre-test run, check pressure and equalize in all shocks - replace valve caps with new ones	When the pod sags to the I-beam - Pressure sensors on the shock absorber - Height sensors	9	2	3	54
Frame Arms at Any Location	Material failure	POD will be damaged due to propulsion components coming loose and/or fully separating	Fractures	Preferably the material has been stress tested by the source company - Visual inspection for surface fractures - Checking that the components are rigidly in place before each run	Collision/friction with track - Height sensors	10	1	3	30
Front/Rear Wheel Suspension	Arm failure	Front/Rear Wheel Suspension system fails	Suspension arm material yielding - Bolts failing	Use FEA static structural - Dynamic simulations	When sparks fly out of the pod - Height sensors	10	1	1	10

Idler Wheel Suspension	Arm failure	Idler Wheel Suspension system fails	Suspension arm material yielding - Bolts failing	Use FEA static structural - Dynamic simulations	When sparks fly out of the pod - Height sensors - Pod deviating laterally	10	1	1	10
Motor Placement	Fastener/Wire connection failure	Rear wheel propulsion severely damaged in the case of fastener failure or power loss in case of wire failure	Micro fractures - material yielding - Improper wire connection	Pre-run tests - Wire connection checks before each run - Pre-run power check	Propulsion power loss detection via controls - Pod collision in case of fastener failure - Rattling noise from the rear wheel in tests/runs	10	1	1	10
Stability wheels interference	Stability wheels touching due to track defects	Pod damage due to path deviation	Track defects - improper clearance	Track check for any unusual defects - Vibration analysis - High clearance	Pod colliding with the track - Height sensors	9	1	1	9
Motor Cooling	Motor overheating	Loss of performance/ power	Clogged cooling tubes - disconnected cooling tubes - damaged cooling tubes	Pre-run tests - Check for leaks	Temperature sensors	5	1	1	5

\* Risk Priority Number = Severity Rating (S)\*Frequency of Occurrence (O)\*Ease of Detection (D)

## Failure Mitigation

The propulsion assemblies mainly consist of the front idler wheel, rear drive wheel, suspension arms, shafts, and electric motor. Potential failure modes include structural failure of the wheels and suspension, spring failures of the shocks, and motor failures from overheating or improper tuning. The following section will demonstrate the safety considerations and failure mitigation of all the major propulsion components.

FEA static structural analysis was conducted on the wheels, suspension arms, and shafts taking into consideration the mass of the pod and motor torque. A minimum safety factor of 2 was achieved with the designed geometries and selected materials (Aluminum 6061-T6 and Alloy Steel). Additional modal analysis was also conducted on the drive wheel showing that it is not anticipated that the wheel will endure a vibratory frequency that can induce mechanical resonance. The rear drive wheel was designed to adopt the in-wheel motor and ensure the center of mass is centered to avoid tilting. The wheel is coated with a layer of polyurethane with a hardness of shore A80 to provide damping, protection to the track, and maximize the traction (coefficient of friction). Depending on the track material, the team will conduct asSoCiated material testing (tribological studies) on the polyurethane for COF and heat profiles.

To avoid possible spring failures of the suspension system, shocks will be tested with a frequency testing device which simulates the pre-load and mass of the pod (refer to the stability section).

The EMRAX 188 electric motor uses a liquid cooling mechanism which is integrated into the motor by the manufacturer. The key risk factors for the motor are the cooling mechanism and voltage control. Since the pod's design does not utilize a gearbox, the motor operates at a high power value throughout the run and for different velocity profiles; therefore, keeping the temperature profile of the motor in check is of extreme importance. For safety purposes, the motor features a peak power of 52 kW yet it only operates at 30% of its maximum output (15 kW) continuously throughout the run. Moreover, due to the high voltage value being fed to the inverter and the variable frequency of the drive, voltage related complication with the inverter during the run are important risk factors. Prior to EHW, the motor will be extensively tested without being connected to the pod in order to verify the coherence and effectiveness of integration with controls and electronics systems.

## EHW Compliance

Propulsion components are all above the top flange of the I-beam track. They do not interfere with the keep out zones of the track specified by the EHW 2021 Rules & Regulations.

## Testing

In order to evaluate various sections of the propulsion system against the relevant safety metrics, the following tests will be designed and conducted:

Electric motor:

The motor testing setup utilizes the X-bracket provided by the manufacturer as the mount, as shown in the Figure P18 below.



**Figure P18: Motor Mounted on the X-Bracket**

Motor temperature benchmark test:

The motor will be mounted on the X-bracket as seen in Figure P18 and an infrared gun will be pointed at the top section of the casing. Motor will be run first at the peak power for 60 seconds and then at the projected operational power for 60 seconds and the temperature will be monitored and recorded. Then, the cooling mechanism will be connected to the motor via the two brass tubes seen on top of the shaft in Figure P18. The test will be repeated and the temperature will be monitored and recorded to assess the feasibility of keeping the temperature under the nominal temperature recommended by Emrax. Both phases of the test will be repeated for the middle section and the bottom section of the casing.

Aggregate propulsion test:

Front wheel, rear wheel, shocks, and suspension arms will be mounted on the bottom plate as seen in Figure P1. Energy and Software subsystems will be fully integrated into propulsion to provide power and the controllers respectively. The bottom plate will then be mounted and secured to a wooden testing rig which elevates the pod 0.3 meters off the ground. Full run scenarios will be tested to verify and validate a successful integration with power source and controllers, wheels at high speed, and the margins of error when actual and predicted run data are compared.

**Post-integration test:**

Every aspect of propulsion in aggregate with all the relevant subsystems will be tested and the results will be compared to the predicted data after the pod is fully integrated and physical runs are conducted on UTHT's 150 m test track.

## the table. **BRAKING**

### General Overview

The purpose of the braking system is to slow down and stop the pod safely. The braking units clamp onto the top flanges of the I beam and the pressure is applied by an internal spring. To release the clamping force, compressed air (145psi) must be fed into the brakes to counteract the spring compression force. To supply the air, a compressor is used to fill up a 100oz tank to 1000psi. Then the pressure is stepped down with a pressure regulator to the 145 psi needed and the flow is controlled by a three-way (normally closed) solenoid valve. When unpowered, the valve allows air to be sent to the braking units. When powered, the valve vents air to the environment and triggers a quick exhaust valve to release air from brakes faster.

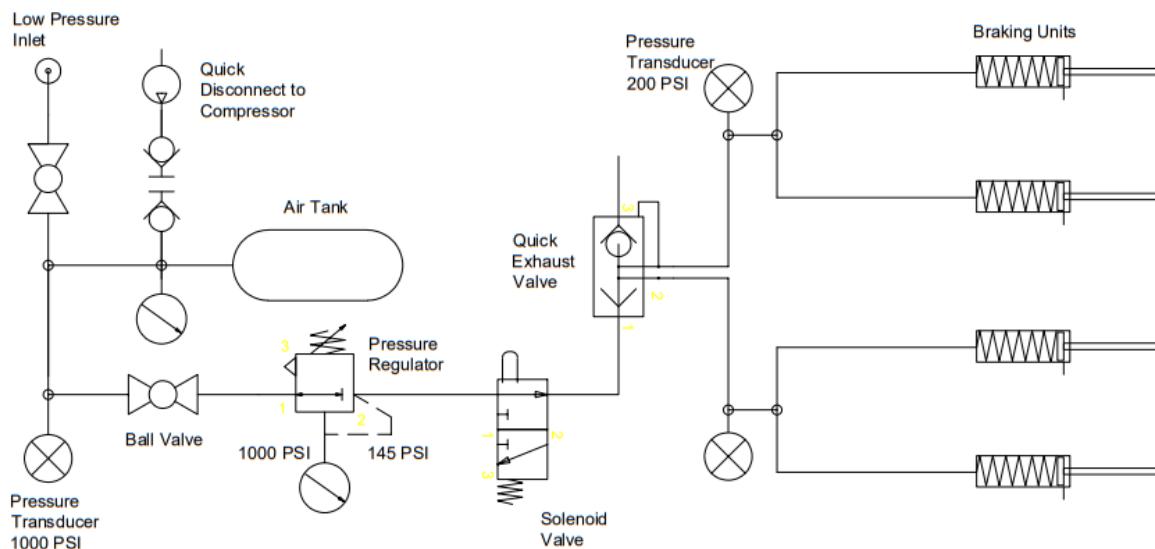


Figure B1: Pneumatic Diagram of Braking System

The general size of the entire system is 500 x 493 x 420 mm (width x height x depth).

### Modes of operation

#### Brakes off

When the 3-way (normally closed) solenoid valve is energized, it allows air to pass through from its cylinder port to the pressure port which allows air to pass through itself. This pressure fills the air cylinder in the braking units to push up through a spring mechanism. The braking pads are now detached from the I-beam track.

#### Brakes on

When the 3-way (normally closed) solenoid valve is de-energized, it allows air to pass through from its cylinder port to the exhaust port which vents air out to the environment. This pressure differential causes the quick exhaust valve to release air at a fast rate. Connected to four pipes, the quick exhaust valve quickly vents air out of the braking units. The springs in the braking units force the braking pads to clamp onto the I-beam and the pod comes to a stop.

### Refilling

There are two methods of refilling the tank. In the first method, both ball valves are closed, and a high pressure air compressor is connected to the quick-disconnect. The tank is pressurized, and the pressure transducer may be used to verify the pressure reading on the air-compressor's pressure gauge.

The other method of filling the tank is for a low pressure test. The ball valve that connects to the pressure regulator is closed, but the ball valve which connects to the low pressure inlet is open. Either a bicycle pump or an automotive grade air compressor can be used to fill the tank via the low pressure inlet. This method is mostly for initial tests for verifying the safety of the system, and testing the pneumatic logic before filling it with high pressure.

Images of the CAD model of the braking system are shown below, and components of the system are listed in Table B1.

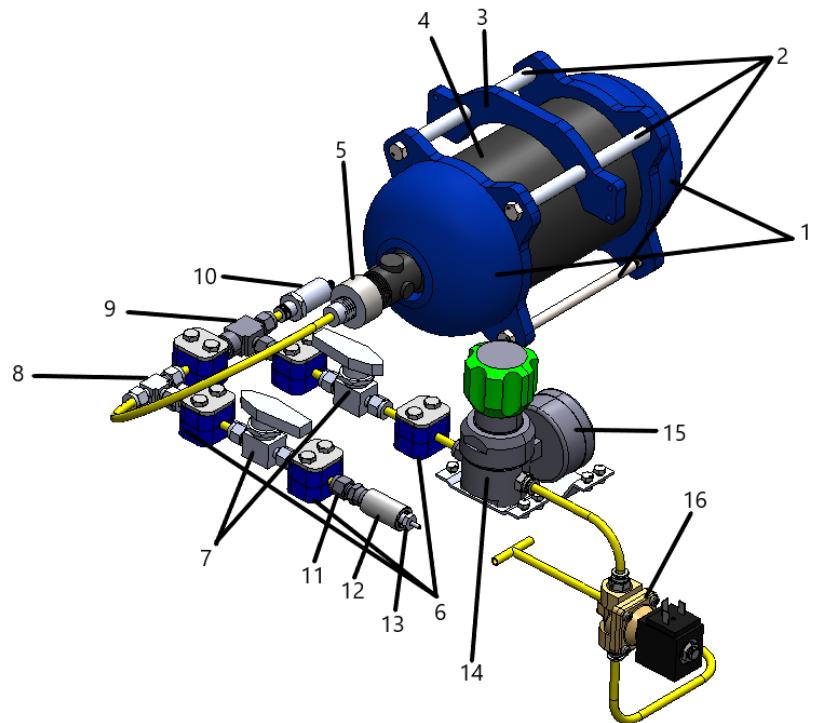


Figure B2a: Locations of main components of braking assembly

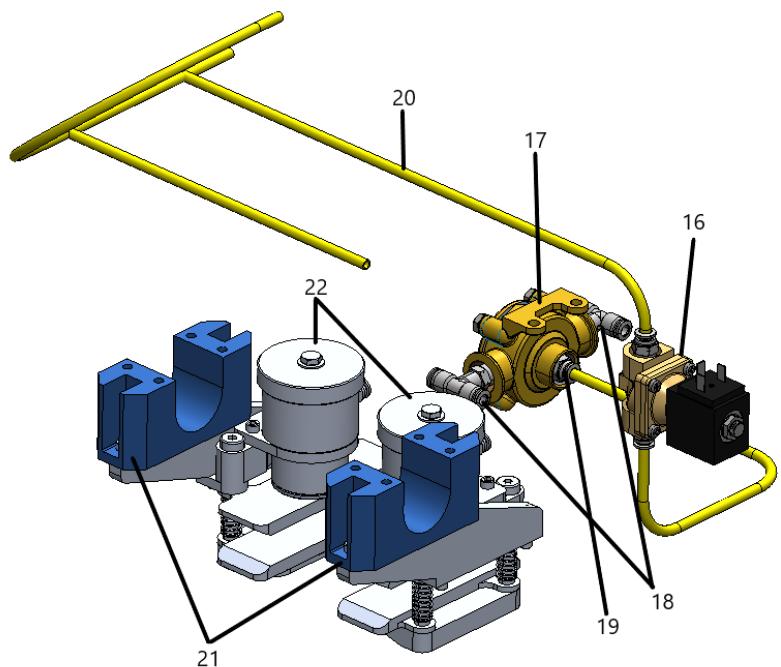


Figure B2b: Locations of main components of braking assembly

Table B1: Component list

Ref. No.	Name	Quantity and Dim.	Description
1	Tank Holders	2 parts, Both approx 2.75" 5.75" 5.75"	Holds the air tank in place along with the tank support arm. Long bolts clamp the tank within the tank holders, and the support arm secures it to the pod structure.
2	Tank Spacer	4x ø 0.47" 2.45"	Slots between the tank holders and the tank support arm in order to support the tank assembly.
3	Tank Support Arm	7.28" 2.5" 0.4"	Supports the air tank in conjunction with the tank holders when it is supported by the pod frame via brackets.
4	First Strike Hero 2.0 100/4500 HPA Carbon Fibre Tank	12.4" 4.5" 4.5"	Stores compressed air safely and allows many uses of the braking units; rated for 4500 psi and has outlet pressure of 300-1200 psi.
5	ASA	ø 1.375" 2"	Allow pressure into the tank when opened and prevents air from escaping when closed.
6	Stacking Bolted Plastic Clamp Tube Support 304-s1-pp-4t	5x 1.45" 1.34" 1.2"	Secures the pipes of the pneumatic system to the pneumatic system mounting sheet. <a href="https://www.swagelok.com/en/catalog/Product/Detail?part=304-S1-PP-4T-ST">https://www.swagelok.com/en/catalog/Product/Detail?part=304-S1-PP-4T-ST</a>
7	Swagelok ball valve SS-43GS4	2x 2.73" 1.85" 1"	Isolates the downstream components during tank refilling, isolates the air fill valve during operation to prevent leaks. <a href="https://www.swagelok.com/en/catalog/Product/Detail?part=SS-43GS4">https://www.swagelok.com/en/catalog/Product/Detail?part=SS-43GS4</a>
8	Tee Connector	2.12" 1.3" 0.56"	Connects the air tank to the pipes that allow air flow through the pneumatic system.

9	Tee Connector	2.05" 1.5" 0.69"	Connects air tank input to pressure transducer and pressure regulator. One end is female threaded to save pipe and reduce potential for leakage.
10	ProSense SPTD25 pressure transmitter	2.25" 0.95" 0.82"	Thin-film sensing element provides very fast response time to provide real time pressure in pneumatic system
11	NPT Tube Fitting	1.5" 0.56"	Connect pipe to air inlet valve.
12	Pipe Fitting 45525K562	$\phi$ 0.75" 1.38"	Connects pipe to schrader valve for refilling tank <a href="https://www.mcmaster.com/45525K562/">https://www.mcmaster.com/45525K562/</a>
13	Air Inlet Valve 8063K37	0.85" 0.4375"	Allows for refilling of air tank with an air compressor or another external source of compressed air <a href="https://www.mcmaster.com/8063K37/">https://www.mcmaster.com/8063K37/</a>
14	Swagelok Stainless Steel Pressure Regulator KPR1GRB412A20000	$\phi$ 2.16" 4.66"	Pressure-reducing, spring-loaded regulators that minimize droop in high-flow conditions while maintaining a constant downstream pressure; rated for 3600 psig inlet and 250 psig outlet <a href="https://www.swagelok.com/en/catalog/Product/Detail?part=KPR1GRB412A2000">https://www.swagelok.com/en/catalog/Product/Detail?part=KPR1GRB412A2000</a>
15	Pressure Gauge	$\phi$ 2.46" 2.39"	Allows operator to read output pressure from pressure regulator and adjust accordingly. <a href="https://www.swagelok.com/en/catalog/Product/Detail?part=PGI-63B-PG160-CAOX-J">https://www.swagelok.com/en/catalog/Product/Detail?part=PGI-63B-PG160-CAOX-J</a>
16	OMEGA-FLOT™ 3-Way Normally Closed Direct Acting Solenoid Valve	3.5" 2.125" 1.0625"	When a potential difference is applied through the solenoid valve, air passes through the valves into the braking units. When it is not powered, it is off by default and air escapes from the valve into its surroundings. This pressure difference triggers a quick

			<p>exhaust valve to release air from brakes faster.</p> <p><a href="https://sea.omega.com/th/pptst/SV4100_SV4300.html">https://sea.omega.com/th/pptst/SV4100_SV4300.html</a></p>
17	Kobelt Model 3903 Quick Release Valve	2.5" 3" 1.875"	<p>Provides fast exhausting of air from the pneumatic system. Exhaust air is vented directly from large valve orifices instead of going back through the solenoid valve</p> <p><a href="https://www.kobelt.com/products/disc-caliper-brakes/list/product-detail/3903-3905-quick-release-valve/brake-controls-accessories">https://www.kobelt.com/products/disc-caliper-brakes/list/product-detail/3903-3905-quick-release-valve/brake-controls-accessories</a></p>
18	Tee Connector	1.63" 1.46" 0.55"	Splits output from either side of the quick release valve to go to the four braking units.
19	Plastic push-to-connect fitting	0.92" 0.55"	Mates pipe to quick release valve.
20	Pipes	ø 0.25" 5'	<p>Provide a pathway for pneumatic power to be delivered from the tank to the braking units, sensors, and a refill valve.</p> <p><a href="https://www.swagelok.com/en/catalog/Product/Detail?part=SS-T4-S-035-20">https://www.swagelok.com/en/catalog/Product/Detail?part=SS-T4-S-035-20</a></p>
21	Brake Attachment	4.33" 1.56" 1.97"	Attaches braking units to the bottom of the pod frame structure.
22	Kobelt Pneumatic Braking Unit	7.25" 3.04" 5.27"	<p>Provides 900N of compressive force onto the test track when depressurized, contains replaceable brake pads with 0.3 friction coefficient. Designed to apply brakes if the solenoid valve is unpowered, so it is fail-safe.</p>
23	Pneumatic system mounting sheet	23" 17.25" 5"	Protects the high voltage battery and provides a platform on which many of the pneumatic braking system's components are mounted.

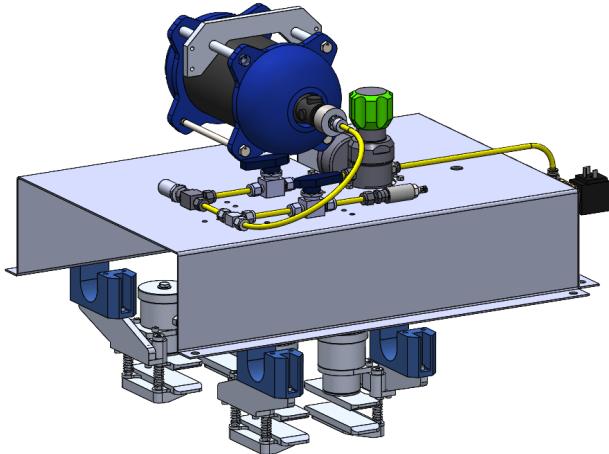


Figure B2c: Appearance of the Braking System

The air tank is suspended at the top of the pod with attachments to the carbon tube frame, while all other components are fixed onto various sheet metal pieces attached to the frame of the pod.

To operate the system, it needs to be pressurized with an air compressor before the demo, which requires an AC outlet. To demonstrate deceleration of the pod from the braking system, the pod needs to be placed onto the provided test track.

## Safety

Key elements of risk of the system include:

- Braking units: actuates when filled with air (clamp onto I beam) and releases when air is released
- Solenoid valve: it alone controls the states of the braking system (release or actuation of brakes)

Highest safety risks are:

- Air tank: a high pressure vessel rated up to 4500 psi, filled to only 1000 psi for safety. Need to prevent contact with heat or impact
- Power loss during a power loss, the solenoid valve will automatically vent towards the environment, triggering actuation of brakes and bringing the pod to a stop

In order to test the subsystem, these tests will be performed:

- Stress testing: the system will be pressurized multiple times to ensure that there is no mechanical weak points
- Leak checking: short-term leaks will be identified by applying soapy water onto joints of the pneumatic system, and long-term leaks will be identified by leaving the system pressurized for 24 hours and logging pressure inside. Any leaks will be resolved prior to the demonstration
- Vibration testing: the system will be vibrated to simulate external forces during the demonstration to check for any potential failure modes

- Live runs: the system will be tested with all other systems in a live run to ensure its functionality

### Engineering Guidelines followed

- The most important thing about the braking system is that it must be safe. All of the components except the sensors are designed to handle at least double the pressure that is being put on them.
- The underlying pneumatic logic is also designed to be fail-safe. This means that if there are any electronic glitches, the electronics can be completely shut off and the brakes will engage, safely bringing the pod to a stop.
- Pipe diameter, tank pressure, sensor limits, and all the components were spec'd around the requirements of the braking units which are the most crucial part of the system.

### FMEA

Table B2: FMEA of Braking

Process/Mechanism	Failure Mode	Effects	Causes	Prevention Control	Detection Control	Severity Rating (S)	Frequency of Occurrence (O)	Ease of Detection and Prevention (D)	Risk Priority Number
Brake pads clamp onto the rail	Solenoid Valve fail to release air	Brakes fail to actuate	particles entering and blocking path	Protect valve from environment	Pressure sensor (1*1000psi, 2*150psi)	10	1	1	10
	Spring Failure		incorrect pressure in brakes	Monitor pressure before run		10	1	1	10
Airflow Through Tank	Loose Fittings	Brakes do not open	vibrations	Check all fittings before testing		3	5	1	15
	Tank not being pressurised		human error	Run test prior to operation		2	1	1	2
Tank containing compressed air	Explosion	damaging other systems	overfilling, overheating	monitor pressure during fill		10	1	1	10
	Leaking	Brakes fail to actuate	damage	padding while handling, inspection		3	10	1	30
Pressurize brakes before run	Over pressurization	damaging brakes	Pressure regulator not tuned correctly	Check with data from other sensors		5	1	1	5
	Under pressurization	unable to fit onto tracks/ scraping during run				1	1	1	1
tube holders	fail to secure system onto pod	damaging system	bolts not tightened	tighten everything before run	Check before run	1	5	1	5

Major failure mitigation strategies aren't severely needed as all failure modes are below the RPN threshold; however, a proper standard operating procedure (SOP) will be followed to prevent any additional safety mishaps. This SOP will follow the prevention controls listed in

## **Stability**

### **General Overview**

The stability subsystem is responsible for keeping the pod in the center of the track while minimizing vibrations due to uneven surfaces. Disturbances from track misalignment are to be expected as per test track specifications; EHW 2021's I-beam has a lateral tolerance of  $+\text{-} 3\text{mm}$ . Pod I features two stability units located at the front and back of the pod as shown in Figure S1. Each unit consists of two lateral and two clamping suspension mechanisms. This system has been designed for a straight track, i.e. with no curvatures.

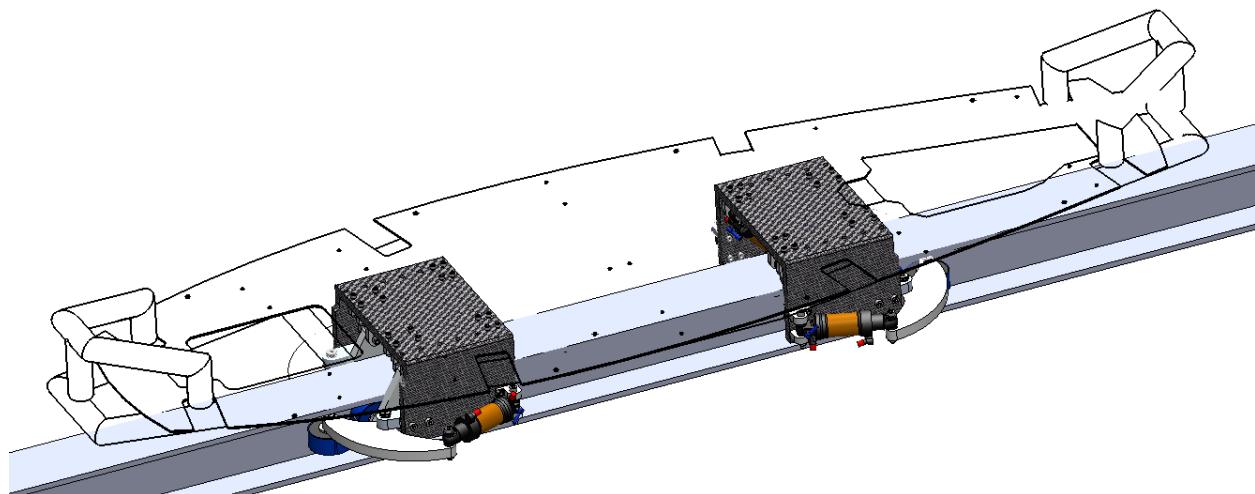


Figure S1: Location of the two stability units in the pod

Table S1 provides the main components of the stability subsystem including the quantity, size, material, and functionality. Figure S1 identifies the location of the parts described in table S1. Table S2 contains the fasteners used for assembly and their specifications.

Table S1: Stability system main components

No.	Component	Qty.	Size	Material	Functionality
1	C-arm	2	Levant	Carbon fiber and Structural foam	To withstand the large loads coming from the lateral system.
2	Air Shock Absorber	8	Eye-to-eye: 165mm Travel: 35mm	Anodized Aluminum 6061	To transform kinetic energy into thermal energy which helps resist movement caused by track imperfections.

3	Wheel	8	Diameter: 3in Width: 1.25in	Polyurethane Rubber and Iron core	To reduce friction.
4	Clamping Force Arm	4	Width: 5in Height: 4in Thickness: 0.75in	Aluminum 6061	To transfer load from the clamp wheel to the shock absorber.
5	Clamping Pivot Arm	8	Width: 2.3in Height: 2.2in Thickness: 0.5in	Aluminum 6061	To connect the clamp arm to the overarching C-arm while allowing for minimal and controlled rotation of the arm
6	Lateral Force Arm	4	Length: 10in Width: 3in Thickness: 0.75in	Aluminum 6061	To transfer load from the lateral wheel to the shock absorber.
7	Lateral Pivot Arm	4	Width: 1.5in Height: 3.7in Thickness: 0.75in	Aluminum 6061	To work as a fixed pivot point for the force arm to rotate about
8	Clamping / Lateral Clevis	16	Width: 1in Height: 1.6in Thickness: 0.5in	Aluminum 6061	To connect the shock to the C-arm

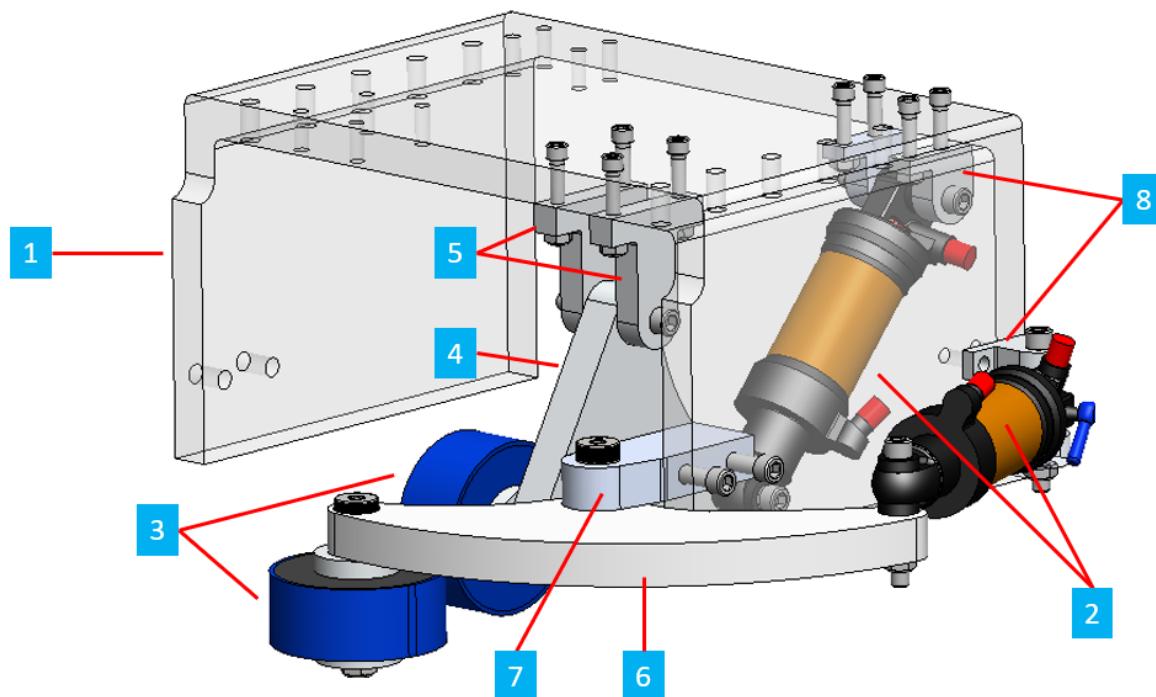


Figure S2. Main components of half of a stability unit

## C-arm

The C-arm supports both the clamping and lateral mechanisms. It is made of carbon fiber and structural foam to withstand the large loads coming from the lateral system. Its dimensions are 306x155x210mm (width x height x depth). The C-arm also facilitates the integration of the stability unit with the pod sheet metal base. It is attached by means of screws as depicted in Figure S3.

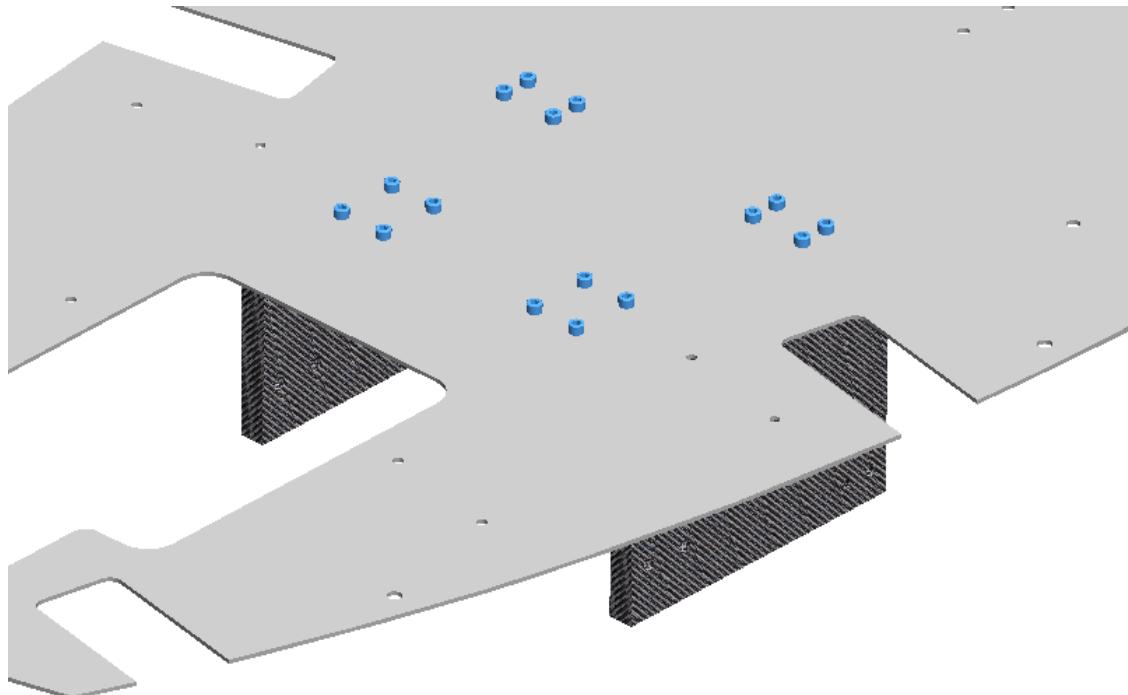


Figure S3: C-arm attachment to pod's floor. Connections highlighted in blue

## Lateral Mechanism

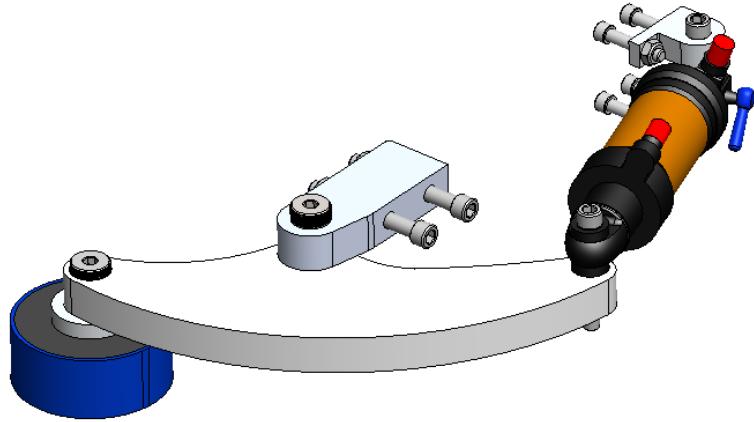


Figure S4. Lateral subassembly

The lateral mechanism, shown in Figure S4, prevents yaw, roll, and sway movement. Each stability unit has two pairs of wheels running along the web of the I-beam as shown in Figure S5. Load is transferred from the wheel into a force arm, which rotates about a fixed pivot point, and into a shock absorber mounted to the C-arm with the help of two clevises. The shocks help minimize vibrations caused by track misalignment when two I-beams are placed together.

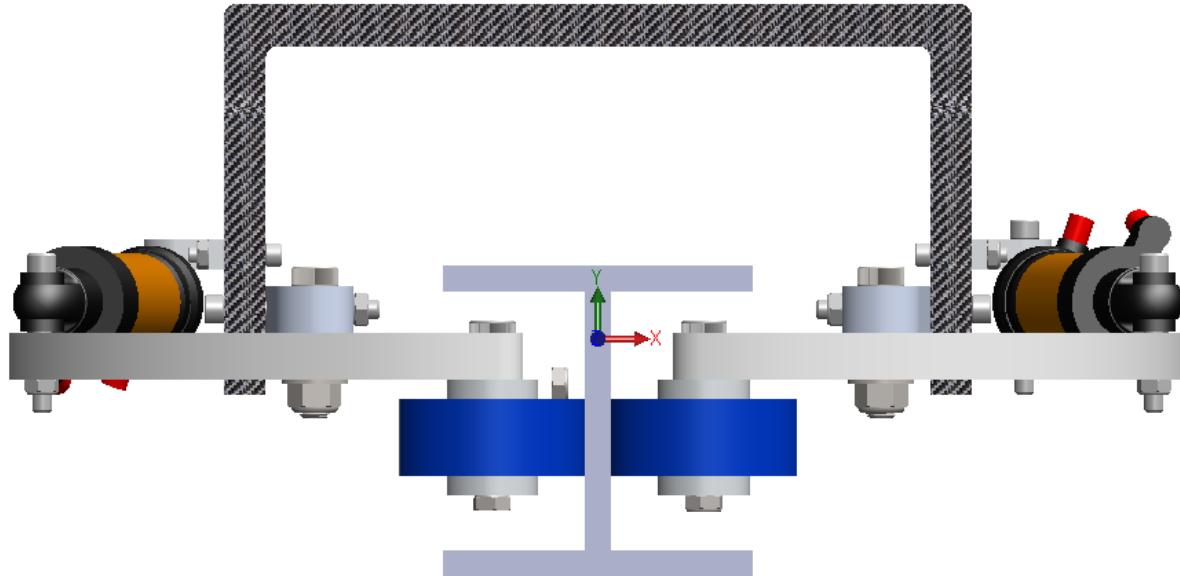


Figure S5: Close-up view of the lateral subassembly

Before operating, this system will be calibrated to ensure that the vehicle enters the track pointing in the forward direction with a tight tolerance. If the pod is not aligned with the track, the shocks will compress, creating a correcting moment and restoring the pod's original position. Figure S6 shows a free body diagram representation of the lateral system of this scenario.

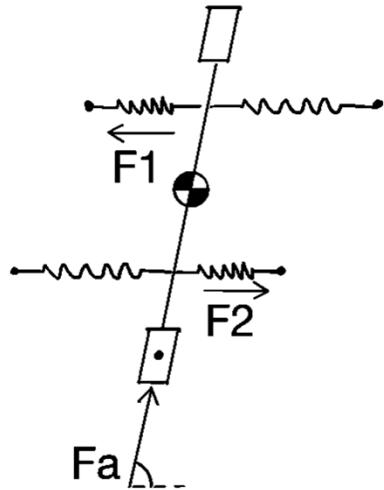


Figure S6: Free body diagram of the lateral subsystem that shows shock compression that will subsequently cause the pod to be aligned with the track

Moreover, the team performed stress analyses to find the parts limiting the performance of our mechanism and modified the geometry and materials to achieve a higher performance. The clevises, force and pivot arms are made of Aluminum 6061 T6 which has a yield strength of 276MPa. The first design iteration aimed to find the loads when the maximum von-Mises stress was approximately equal to the yield strength. A summary of the results can be found in Table S3. Figures S7 through S9 provide the results obtained in Ansys.

**Table S3: Approximate maximum load for each component**

Component	Maximum Load (N)
Pivot Arm	16500
Force Arm	25300
Single Clevis	6750

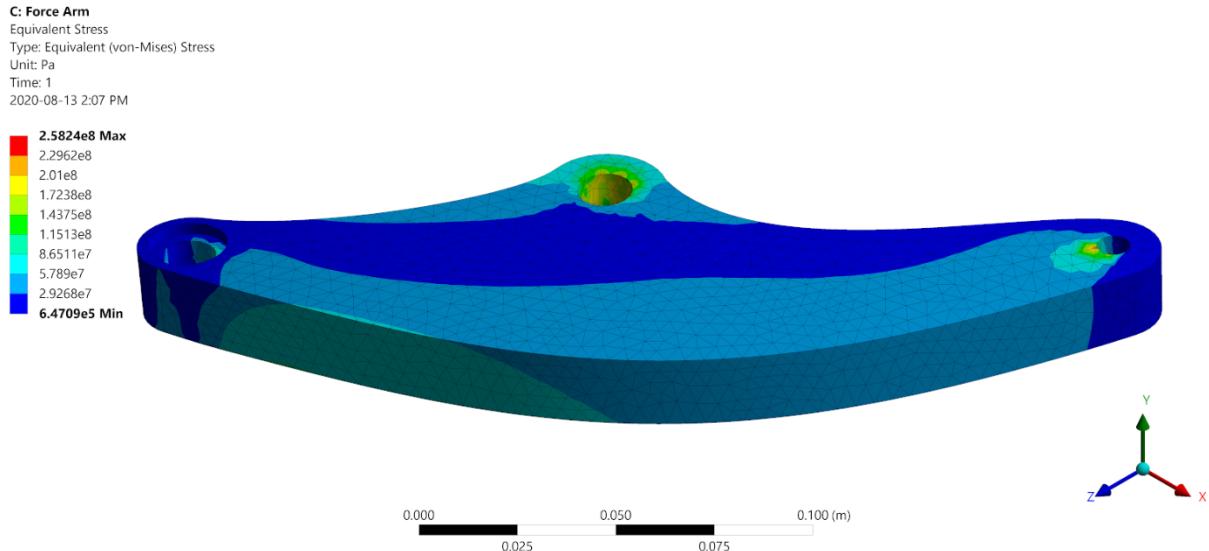


Figure S7. Stress distribution of first iteration force arm

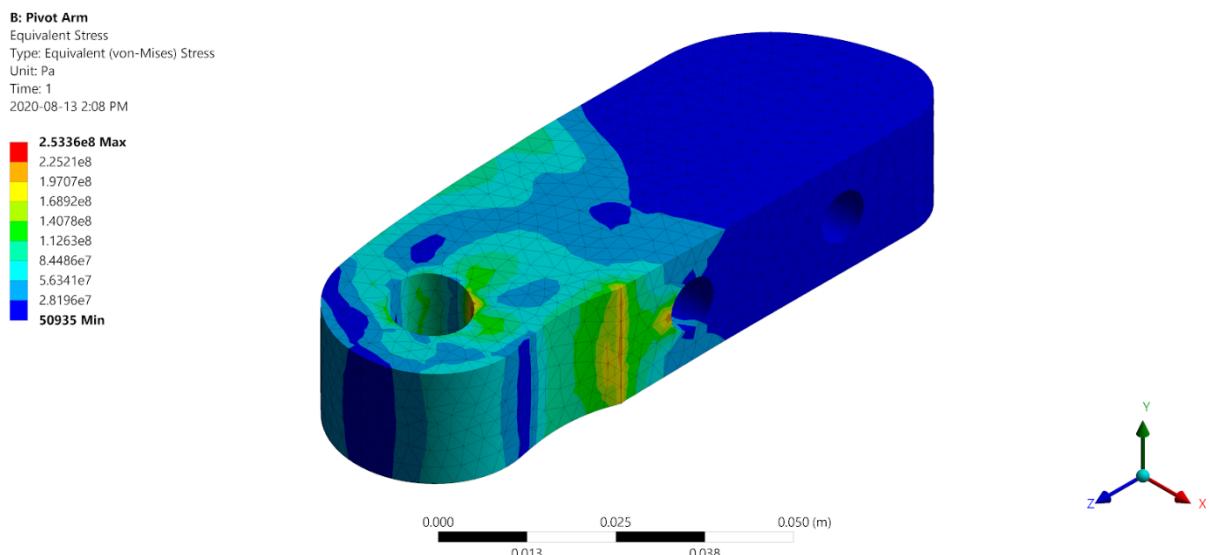


Figure S8. Stress distribution of first iteration pivot arm

## Clamping Mechanism

Similarly, each stability unit has two pairs of wheels running along the bottom of the top flange of the I-beam, as shown in Figure S10, that serves to provide control motion in two degrees of freedom: pitch and heave. The main purpose of the clamping suspension is to absorb vibrations in the direction perpendicular to the track. Our pod is relatively light for the speeds it aims to achieve, so this mechanism is necessary to prevent excessive motion in the vertical direction. A force arm bridges the clamping wheel with the shock as can be observed in Figure S11, allowing

for vibrations and irregular motion to be absorbed by the system. The force arm has freedom to rotate about its uppermost bearing hole to allow for the movement caused by track top flange imperfections. The main load that the arm experiences is a bearing load from the wheel's vertical movement. Calibration for this mechanism is similar to the lateral one.

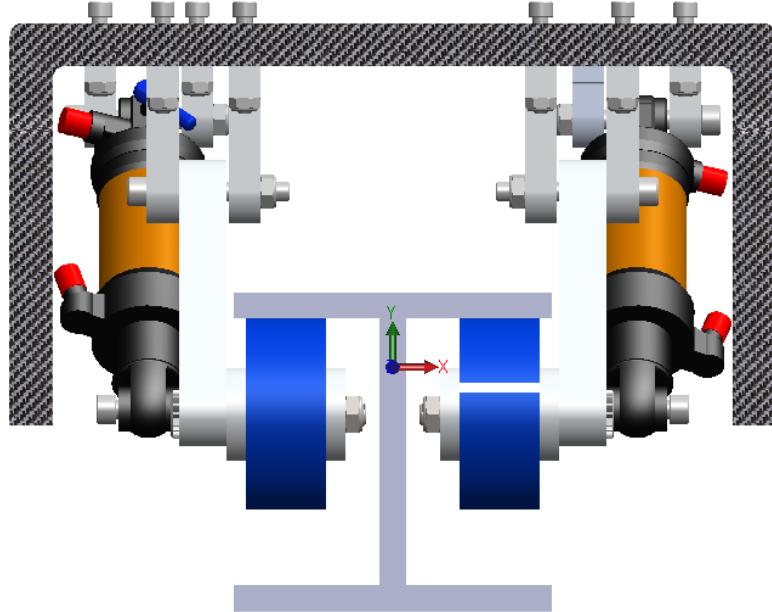


Figure S10: Close-up view of the clamping subassembly

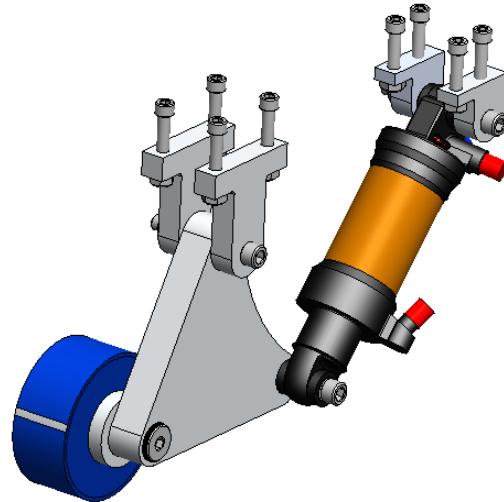


Figure S11. Clamping subassembly

Furthermore, stress analyses were also performed to optimize geometry and material selection. The pivot, arm, and clevises have been designed to withstand loads most effectively with minimal deformation. As mentioned before, the maximum loads were determined when the part approached a von-Mises stress equivalent to the yield strength of the part material. The results

have been summarized in table S4. Figures S12 through S14 provide the results obtained in Ansys.

Table S4. Approximate maximum load and deformation for each part

Component	Maximum Load (N)	Stress at Load (MPa)	Maximum Deformation (mm)
Pivot	7000x2	254.25	0.081155
Arm	12500	259.52	0.026977
Clevis	5400x2	261.25	0.016127

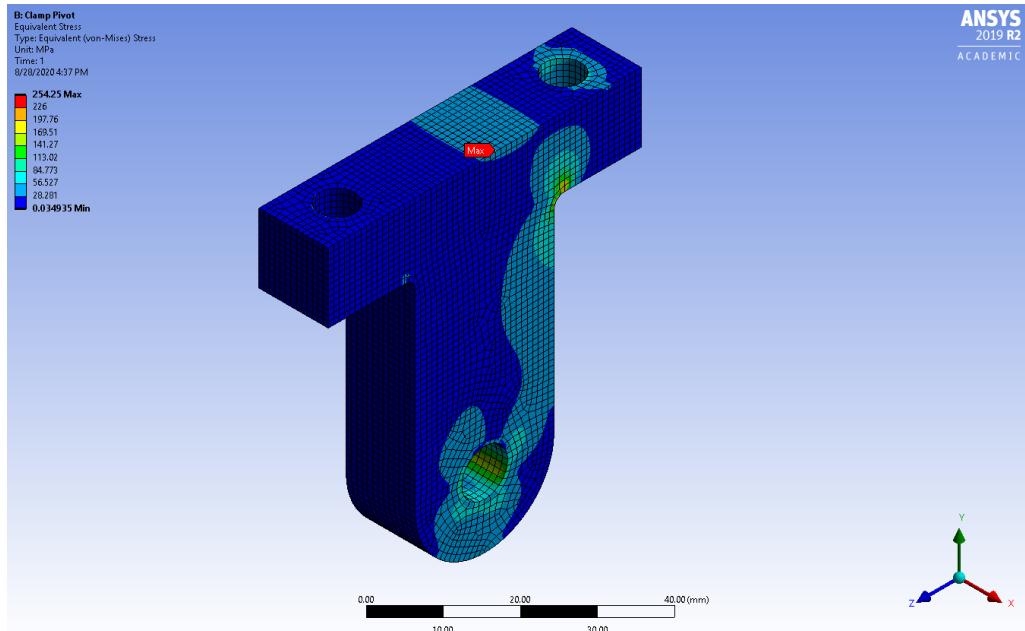


Figure S12. Stress distribution of bearing load on clamp pivot

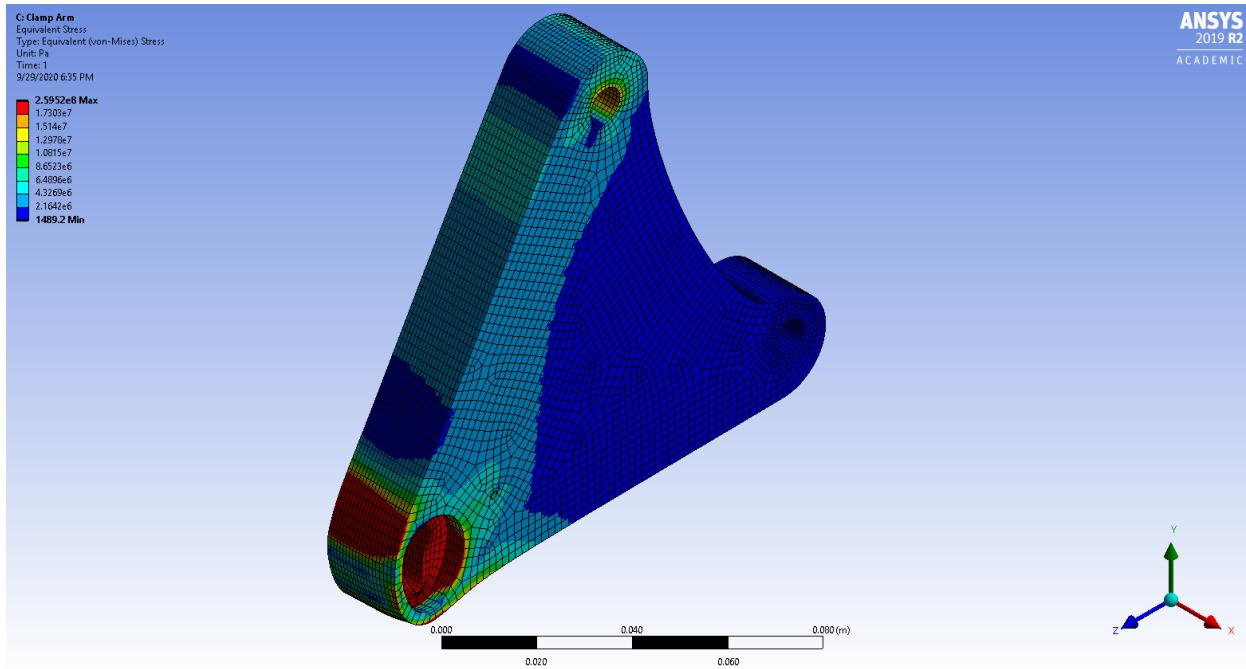


Figure S13. Stress distribution of bearing load on clamp arm

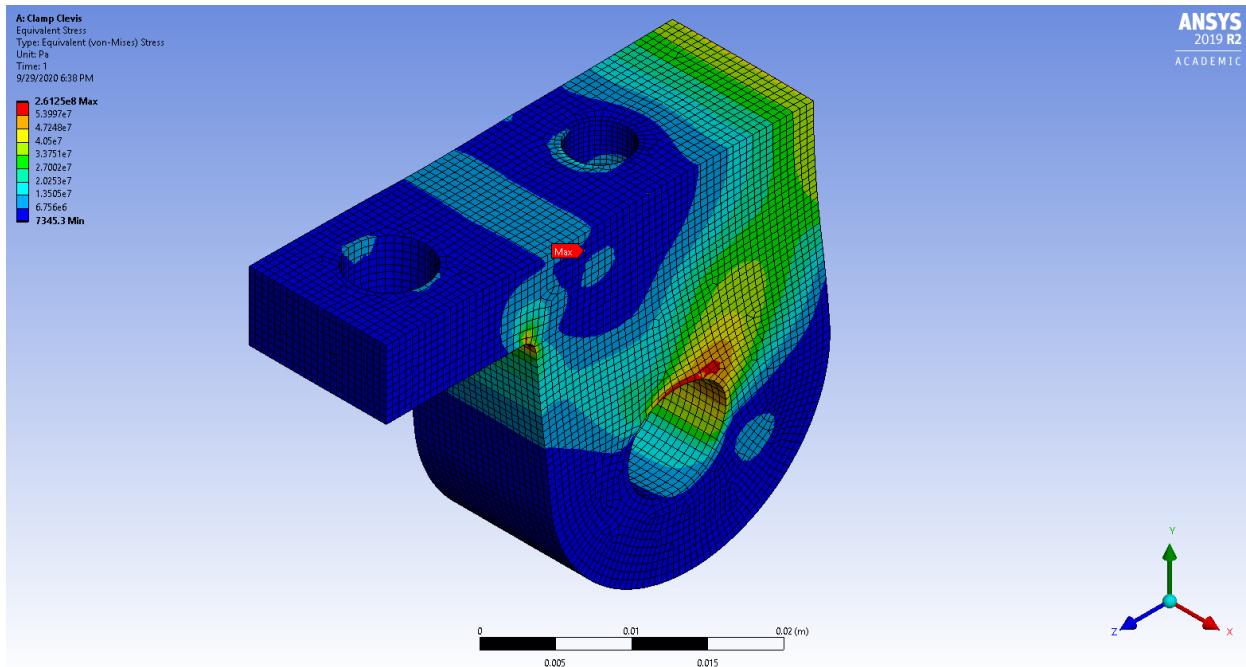


Figure S14. Stress distribution of bearing load on clamp clevis

The clamp clevis ties the shock to the overarching C-support and deals with any loads from the shock at an angle of  $48^\circ$  to the horizontal. It should be noted that this is the same part that was analyzed in the lateral mechanism section. However, the setup slightly differs and so the results are different.

## Modal Analysis

In order to see how our system would behave under dynamic motion, we looked at the mode shapes, or deformation the components would experience when vibrating at their natural frequencies. The frequency of the structure depends upon mass and stiffness. Higher the frequency and lower the period means your structure is stiff and will have less global response. The response is a relative term. One limitation of this theory is that it holds only in the elastic range.

Two frequency sources in the pod when achieving a speed of 70m/s are the motor and the beam misalignments with 73.31Hz and 23.33Hz, respectively. By comparing the pod's frequencies to the motor (source) frequency generated, it can be seen in Table S5 that the lowest mode frequency is twice that of the source frequency. Thus, for a given 70m/s speed, a safety factor of 2 is applied to avoid resonance.

Table 2.2.1: Modes and corresponding frequencies for the entire assembly

Mode No.	Natural Frequency (Hz)
1	152
2	156
3	179
4	230
5	231
6	364
7	365
8	571
9	572
10	623
11	623
12	873
13	879

## Shock Selection

Air shocks have been selected instead of coil shocks because its properties can be easily adjusted by changing the pressure of the air chambers. This flexibility is especially important because the clamping and lateral stability systems have different requirements. The guiding principle for the clamping system is that it needs a relatively small preload and low spring constant, while the lateral system requires a larger preload and high spring constant. The downside of choosing air shocks is that they require more extensive analysis and testing.

The major limiting factor was geometry. The stability system had to fit within the existing design of the frame, brakes, wheels, and shell, so the ideal shock size was 165 mm. The team also needed the ability to adjust the pressure of both positive and negative chambers easily during tests.

UTHT selected the DNM-AOY 36RC because it balances quality and affordability. This robust shock absorber has the following specifications:

- Eye-to-eye: 165mm
- Travel: 35mm
- Adjustable preload and spring constant.
- Adequate damping ratio

To characterize the stiffness behaviour of the selected shock for simulation purposes, a mathematical model of the air shocks was created based on the internal pressure and dimensions of the two chambers. Fortunately, the team found an open-source Excel calculator with extensive video documentation. The inputs to the calculator are the dimensions and initial pressures of the shock, and the output is a curve with the force excerpted by the shock shown in green in Figure S15. These results were then validated through testing. The team was not able to test 100% of the shock total travel since the testing device started to deflect at loads higher than 1000N, so the shock was only tested up to 35%. Test data is shown in orange in Figure S15 as well. The black line represents the shock equivalent coil stiffness. At 80% of spring travel, that is,  $35*0.8=28\text{mm}$ , the equivalent force is  $300\text{kgf}=3000\text{N}$ . Thus,  $i = 100 \text{ N/mm}$  as can be seen in Figure S15.

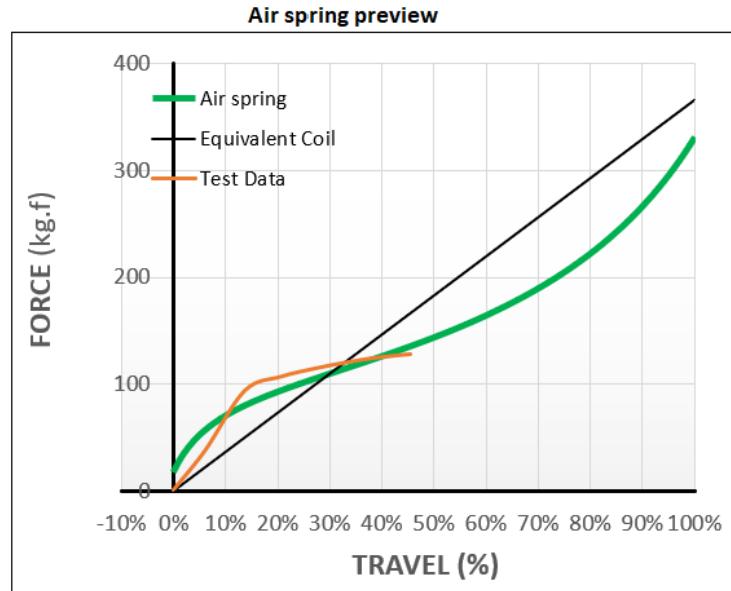


Figure S15: Shock force vs travel curve

The shock was tested by mounting it between two thick aluminum plates and compressing them with lead screws and nuts as shown in Figure S16. The force was measured by adding a scale and a third plate as shown in Figure S16. The angle of the third plate was controlled with a spirit level to ensure it stayed perpendicular to the ground throughout the tests. This device needs to be redesigned so loads higher than 1kN can be tested. The team aims to achieve a minimum of 70% of total shock travel characterization. The full assembly drawing of the shock testing device can be found in Appendix S1.



Figure S16: Shock testing device and shock

## Safety

### FMEA

Table S5: FMEA of Stability

Process/ Mechanism	Risk	Effects	Causes	Prevention Control	Detection Control	Severity Rating (S)	Frequency of Occurrence (O)	Ease of Detection (D)	Risk Priority Number
Load support/ transfer	Failure of bolts that connect the shock absorber to the C and force arms	Pod tilt and derailment. Partial disconnection of components in the stability system	Reaction force at the wheels greater than expected due to track defects/bumps/ misalignment. Bolts are too weak	Testing of subassembly to determine dynamics loads. Bolt material grade selection	Visual inspection	10	4	1	40
Lateral suspension	Shock failure	High level of vibrations leading to derailment of pod. Position of pod could not be corrected anymore	Failure of shock due to manufacturing. Improper preload and pressurization	Testing of shocks. Pressure check. Shock selection	Adjust/ Measure pressure before run	10	4	3	120
Clamping suspension	Shock failure	Undamped vibration can lead to motion of pod in the heave direction	Failure of shock due to manufacturing. Improper preload and pressurization	Testing of shocks. Pressure check. Shock selection	Adjust/ Measure pressure before run	3	4	3	36
C-arm support	Failure of C-arm due to excessive forces coming from the lateral suspension system	Bending or total failure of C-arm. Overcompressi on of shock	High stress concentration at clevis that bridges C-arm with shock	Arm to be made out of several layers of carbon fiber. FEA to find maximum deformation	Visual inspection	3	2	5	30
Stability unit components	Part failure due to bearing load	Failure of part affects overall functionality of stability system	High stress concentration due to excessive load at the connection holes	Structural analyses of parts to find maximum allowable forces. Material selection.	Visual inspection	9	1	3	27

Failure mitigation strategies will be employed by rigorously testing the shocks before use to verify design points are met and choosing fasteners with appropriate tensile strength to mount and assemble the subsystem. All prevention controls will be followed too.

### Compliance with EHW rules

The position of the wheels or any other component of the stability unit does not interfere with the I-beam keep out zones established by EHW. Figure S17 shows the position of the current stability system's wheels with respect to the web and bottom of the top flange of the I-beam.

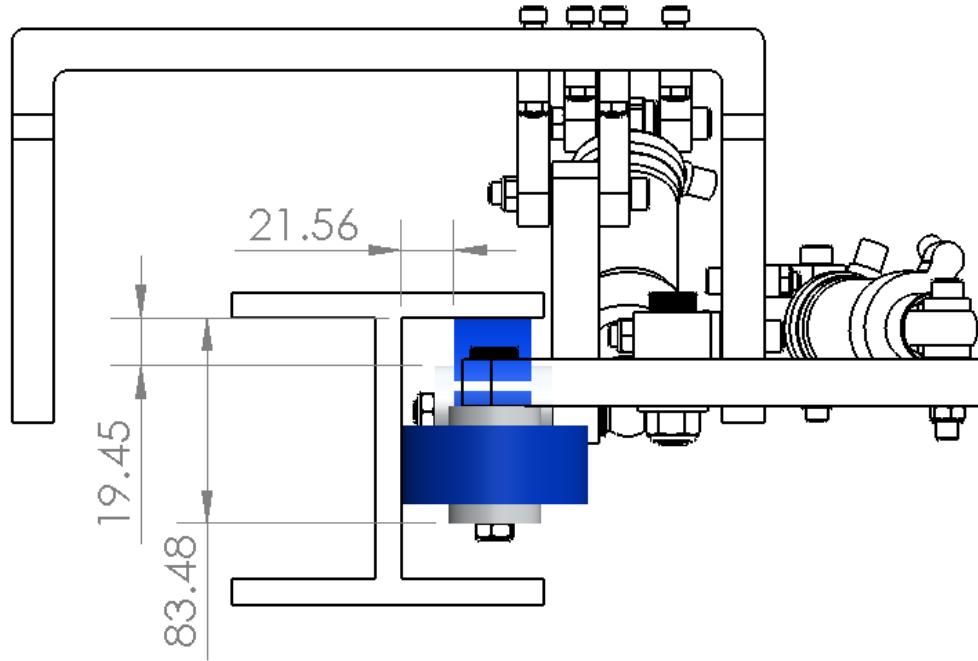


Figure S17: Position of wheels with respect to the I-beam (dimensions in mm)

### Subsystem testing

To validate modal analysis results and prevent any resonances in the system, the team plans to build a device that allows the shock to rapidly move up and down. A bar connects the shock to a rod moved by a motor while an accelerometer measures vibrations. The up and down motion is caused due to the offset created with a smaller bar connected to the shaft of the motor as seen in Figure S5. The resting bar should be fixed to the ground as a safety measurement and to minimize noise in the vibration data when running the test.

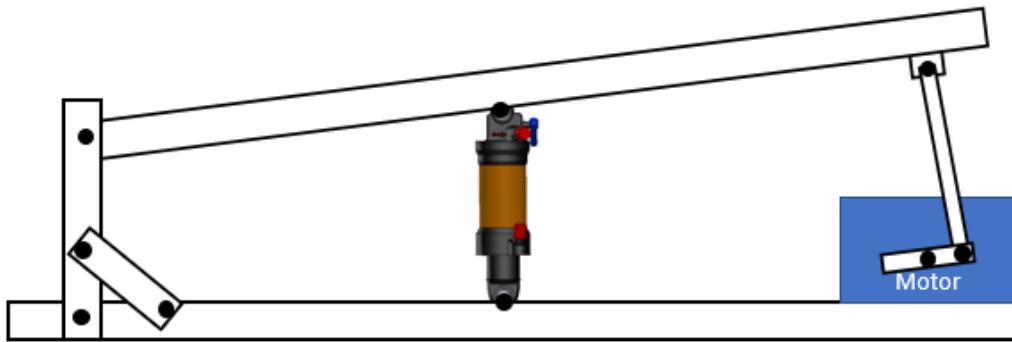


Figure S5: Frequency testing device setup

Finally, shock damping will be also tested as the team needs to complement and verify the simulation results. UTHT will use a disk of 2.5 m in diameter, rotated by a motor, to compare the wheels' movement with and without the shock. Weight will be placed on top of the system to simulate the pre-load and mass of the pod (see Figure S7).

This testing machine already exists at the University of Toronto for rolling resistance tests; UTHT will adapt it to accommodate a shock and a smaller wheel. The machine also counts with an accelerometer to measure vibrations. We will use the data collected by the accelerometer and other known variables to find the mathematical model, including the damping coefficient, that best describes our system.

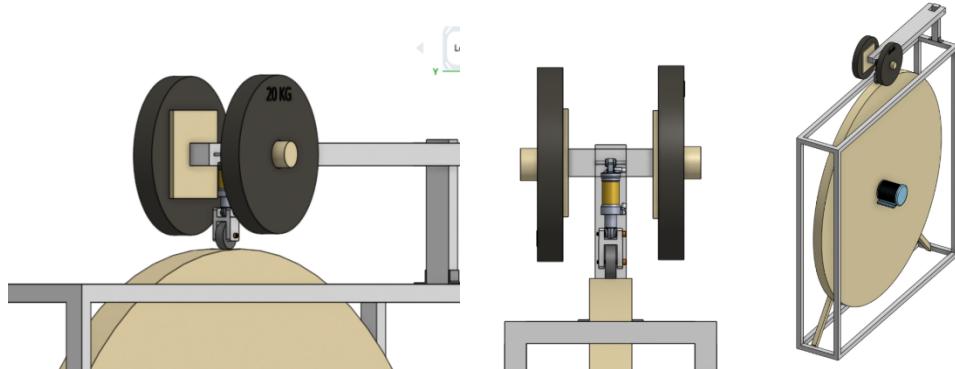


Figure S7: Close isometric, front and full views of the test setup

To see the assembly drawing of the shock testing device, see Appendix ST1.

## Energy Systems

### General Overview

The UTHT Energy Systems powers the Pod's electric motor and its various electronics. This is accomplished with two independent battery modules, as depicted in Figure ES1:

- A high voltage battery (HVB) system responsible for powering the motor
- A low voltage battery (LVB) system dedicated to powering the onboard electronics, sensors, and motor controller

The subsequent subsections will provide more in-depth explanations of the high voltage battery, motor inverter, low voltage battery, low voltage power distribution, and charging systems. Their respective testing and safety procedures will also be detailed.

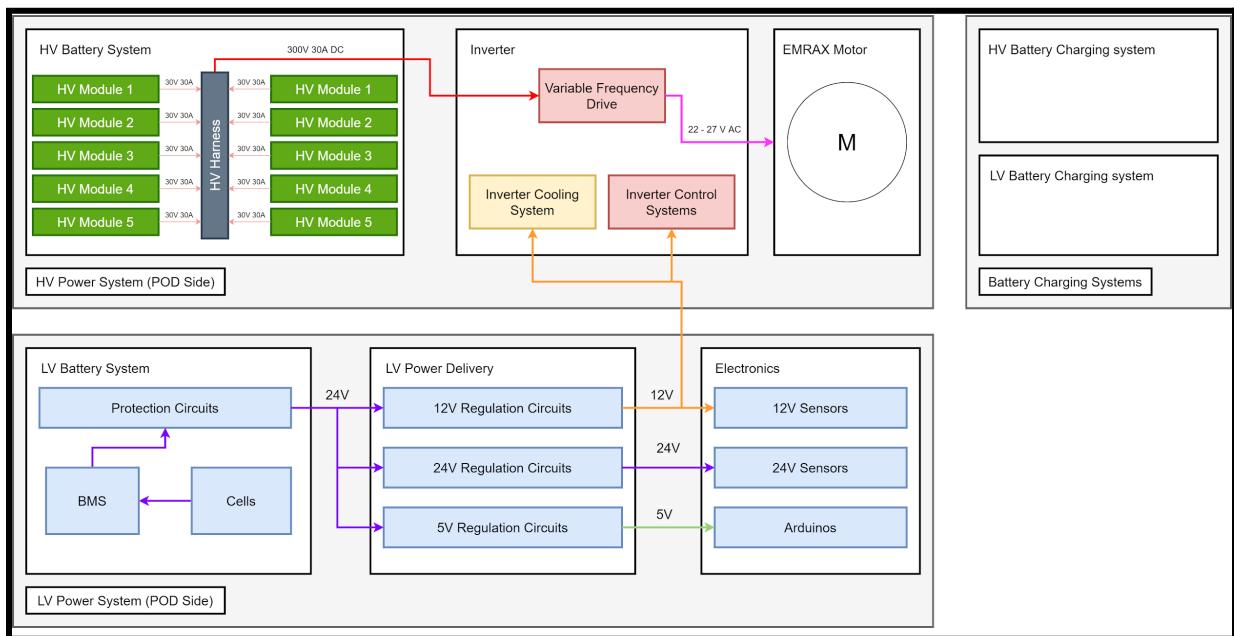


Figure ES1: A high-level block diagram of the major energy systems components

Table ES1: Summary of Energy Storage Types & Systems

Subsystem	Storage Type	Safety Precautions
HV Battery	Electrochemical Cells	Equipped with BMS, temperature sensor and controls
Motor Inverter System Capacitors	Electrostatic Charge	Capacitors will be automatically discharged once the inverter is disconnected from DC power or when the pod operation is complete
LV Battery	Electrochemical Cells	Equipped with BMS, temperature sensor and controls

## Energy Systems Safety & Testing

This section briefly discusses some safety considerations & standards that were used for the design and engineering of the Energy Systems. It also details integrated sub-system tests that would incorporate all parts of the Energy Systems working together. Tests will be carried out on Low Voltage and High Voltage sides individually before combining both sections together for safety and reliability. Unit tests are detailed in each subsection.

Table ES2: Table of Safety Measures & Standards Used in System Design

Subsystem	Safety Standard	Application
HV Battery System (HVB), LV Battery Module (LVB), LV Power distribution system (LVPD)	IPC-2221 - Generic Standard on PCB Design	Used to determine necessary PCB trace width for Arduino control & monitoring system
HV Battery System (HVB), LV Battery Module (LVB), LV Power distribution system (LVPD)	ISO 19642-4:2019(en) - Automotive Cables	Dimensioning of wire gauge necessary for battery system and HV Power distribution system on DC side
Motor Inverter System	ISO 19642-4:2019(en) - Automotive Cables	Dimensioning of wire gauge necessary for AC power transmission between inverter and induction motor
Motor Inverter System	Canadian Electrical Code Code 26-222 - Requirements of installation of all capacitors	Sizing of resistor to drain inverter capacitors, rating of contactor necessary for capacitor discharge system

Table ES3: Table of LV Power System Tests

Test Description	Purpose of Test	Expected Values
Connect a fully charged LVB to the LVPD. Connect a load to each power rail, monitor voltage of each rail.	Ensure that each of the voltage rails maintains the output voltage they are rated for as the voltage of the battery drops and the battery is being discharged under loads	$V_5 = 5V \pm 0.1V$ $V_{12} = 12V \pm 0.1V$ $V_{24} = 24V \pm 0.1V$
Connect a fully charged LVB to the LVPD. Connect all expected loads to each power	Ensure that each of the voltage rails maintains the output voltage they are rated for as the voltage of the	$V_5 = 5V \pm 0.1V$ $V_{12} = 12V \pm 0.1V$ $V_{24} = 24V \pm 0.1V$

rail, monitor voltage of each rail.	battery drops and the battery is being discharged under real and heavy loads	
-------------------------------------	------------------------------------------------------------------------------	--

Table ES4: Table of HV Power System Tests

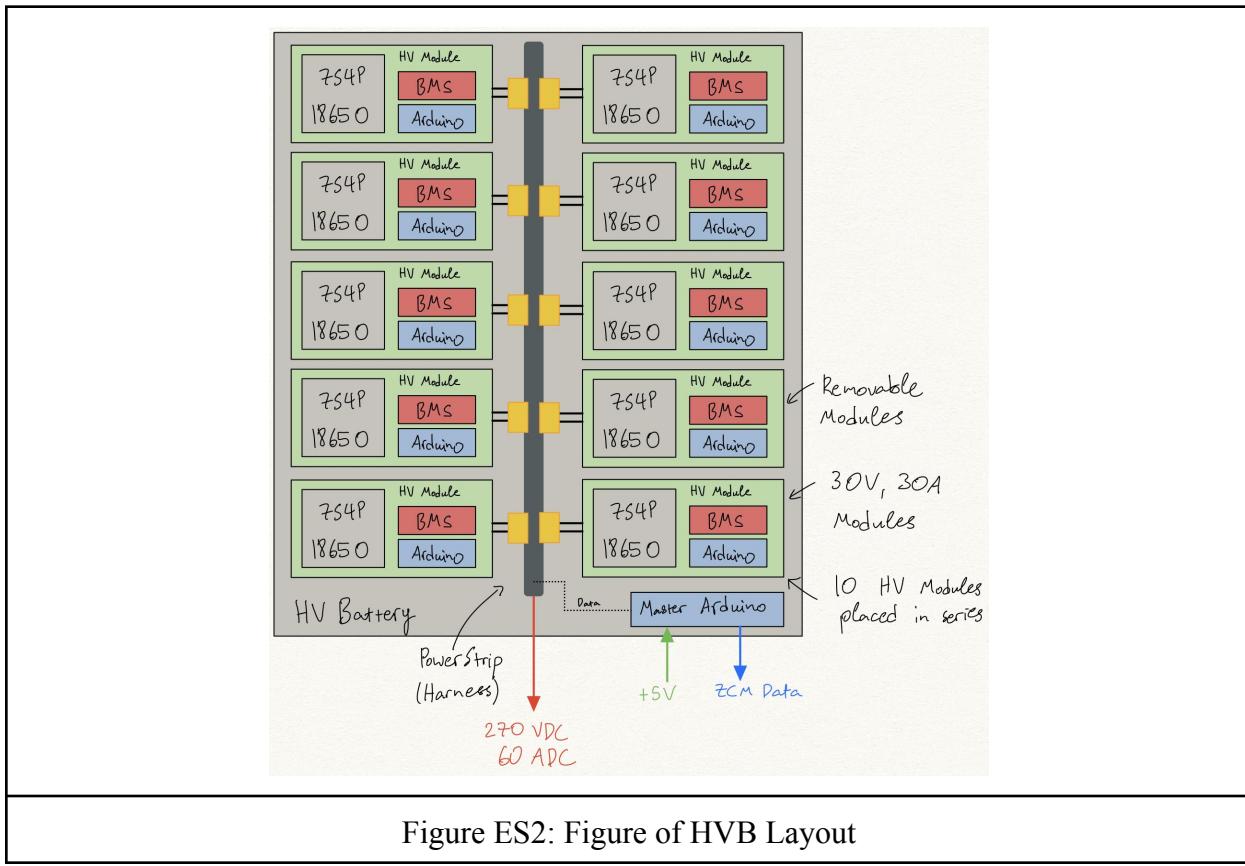
Test Description	Purpose of Test	Expected Values
Connect all fully charged battery modules together to form HVB and monitor its voltage.	Ensure that the HV Battery System harness is functioning properly	$V = 294V \pm 1V$
Connect HVB to simulated load and monitor voltage & current during continuous discharge (20A, 40A, 60A) loads.	Ensure that the system can provide the power requirements needed when connected via the harness. Allows us to monitor and compare real world expected performance of battery	Temperature of cells should not exceed threshold T of 70 C throughout tests.

Table ES5: Table of Integrated Energy Systems Tests

Test Description	Purpose of Test
Connect fully charged HVB and LVB to Inverter. Connect Inverter to motor. Connect Arduino from electronics subteam with CANBUS to inverter input. Activate pre-charge sequence & monitor HVB voltage and current.	Ensure that the entire Energy Systems is able to function together. There is a dependency on an Arduino designed by the electronics subteam to interface with the inverter. This test focuses on the ability of the pre-charge circuit to function and be activated in a safe and controlled environment.
Test motor speed control using the inverter. Monitor HVB parameters, record motor data. Continue until the HVB is drained.	Ensure that the system functions together to provide power to the motor. Integrated system should have BMS's cutoff when batteries are drained.
Open HVB contactor, activate MIS capacitor discharge circuit. Monitor inverter DC voltage.	Ensure that the shutdown sequence for the Energy Systems can be executed. The capacitors should be fully drained under 1 minute.

## High Voltage Battery System (HVB)

The HVB is made of 280 Samsung 18650 25R lithium-ion cells in a 70S4P configuration. It is made up of 10 individual High Voltage Battery Modules (HVBM) connected in series, each featuring a 7S4P layout. The modules are connected together using a HV power bar, which is fitted with 10 XT-90 connectors in series. This results in a battery pack with a nominal voltage of 252V and is capable of safely supporting current draws up to 80A (a total max power of 22 kW). However, the EMRAX 188 motor has been configured to draw less than 60A of current from the HVBM and operate at a maximum of 16 kW. This buffer, combined with the other active safety features implemented, will help minimize the risks of overheating, thermal runaway and accelerated cell degradation. The entire power bar is connected to the pre-charge circuit and feeds the power into the inverter. This setup is shown in Figure ES 2.



Each HVBM is rated at 30V and 80A max and has its own battery management system (BMS). The 7S4P cell configuration is made up of 7 Energus Li4P25RT modules connected in series, each of which are equipped with a BMS that provides overcharge, over-discharge, overcurrent protection, and balance charging functionalities. An Arduino Micro in each of the 7 Energus modules monitors the temperature of the cells through two dual-point temperature sensors. Altogether, 8 dual-point temperature sensors (or 29% of the total number of cells) are used to

ensure the HVBM does not overheat by automatically disconnecting the affected modules from the powertrain. It also provides operators with a heatmap-view of cell temperatures from the online dashboard with fairly high granularity. The Arduino also provides real time monitoring of the battery's State of Charge (SoC). The subsequent HV Battery Software section will discuss this temperature monitoring feature in greater detail.

The Arduinos within each HVBM are daisy chained with the other modules via a dedicated communications line embedded within the power bar. A master Arduino records the maximum temperatures and module SoCs from the slave Arduinos and relays this information to the Pod's main flight computer.

### HV Battery Software

Every battery module contains an Arduino Micro that monitors each cell's temperature and its' SoC. The Micro measures the temperature of each of the seven cells by receiving a voltage drop from the 2 point temperature sensor connected to each cell. It then converts the voltage value into the equivalent temperature by interpolating data points from the voltage-to-temperature conversion graph (Figure ES3). To measure state of charge of each module the Micro reads the voltage over a voltage divider and the reports the battery's state of charge according to the discharge characteristics (Figure ES4)

Table 2. Voltage-to-temperature conversion values

Temp, °C	-40	-35	-30	-25	-20	-15	-10	-5	0	5	10	15	20	25	30	35	40
V <sub>out</sub> , V	2.44	2.42	2.40	2.38	2.35	2.32	2.27	2.23	2.17	2.11	2.05	1.99	1.92	1.86	1.80	1.74	1.68
Temp, °C	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	
V <sub>out</sub> , V	1.63	1.59	1.55	1.51	1.48	1.45	1.43	1.40	1.38	1.37	1.35	1.34	1.33	1.32	1.31	1.30	

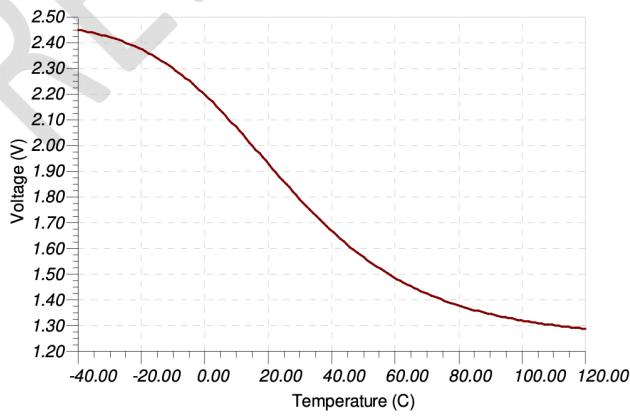


Figure 5. Temperature-Voltage response

Figure ES3: Voltage-to-temperature conversion values (From Energus)

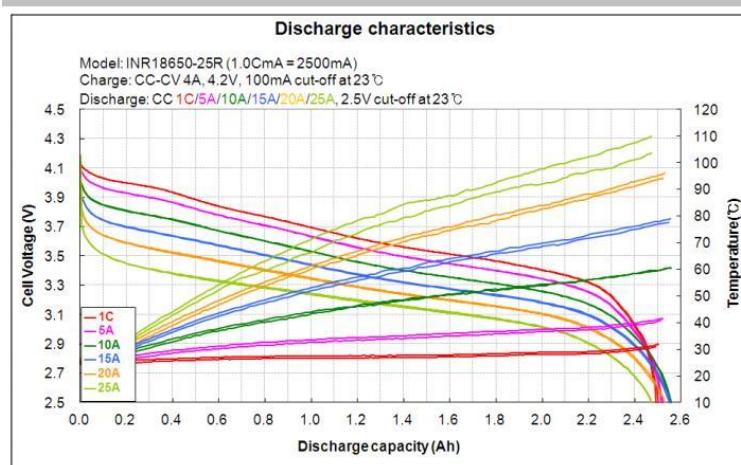


Figure ES4: Cell Discharge Characteristics

#### Temperature, SoC Communication, Current Measurement

Each of the ten Arduino Micros from the battery modules is connected to a main Arduino Due. The Arduino Micros communicate with the Arduino Due using I2C communication protocol, where the Arduino Due acts as the master and the Arduino Micros are the slaves. Each Arduino Micro communicates the highest cell temperature and the SoC of the module. The Arduino Due then sends these temperatures and the highest SoC to the flight controller, where they can be monitored. This slave-master Arduino set-up is illustrated in Figure ES2. A 100A hall effect current sensor (LEM USA HAIS 100-P) is placed around the main wires leaving the power strip to measure the total current being discharged from the 10 battery modules. The voltage from the hall effect sensor is sent to the main Arduino Due where it is converted into a human-readable current value and sent to the flight controller where it can be monitored by the Pod's operators.

#### Parts List & Sizing

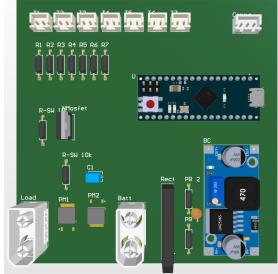
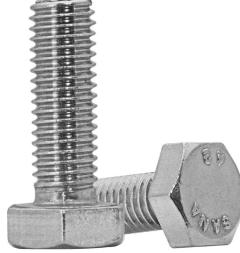
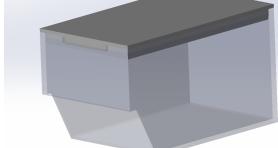
Each HVBM is around (176mm x 102mm x 101mm) and the entire HV battery when assembled is around (515mm x 367mm x 112mm). The image of the module and battery are shown in Figure ES5, and Figure ES6 respectively. There are 10 HV modules in the battery and the components of each HV module are listed in Table ES6.

Table ES6: Table of HV Battery Module Components

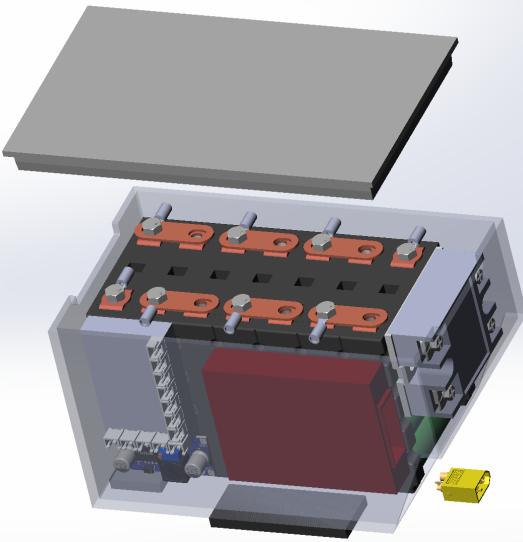
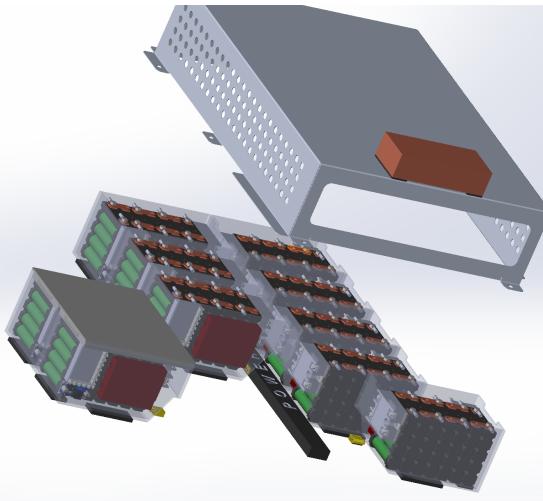
Component	Image	Description	Quantity
Li4P25RT Cell Module		A pack of 4 Samsung 25R 18650 cells connected in parallel. Each cell is double fused and there are 2 temperature probes in each module.	7

DALY 7S 24V 60A BMS		7S 24V 60A BMS. 180A peak current, rated for 7 cells in series	1
Arduino Micro		An Arduino micro A000053	1
LM 2596 DC-DC Buck Converter		30V to 5V buck converter	1
Full Wave Bridge Rectifier		Bridge Rectifier Single Phase Standard 600V Through Hole GBJ	1
P-Channel Power Mosfet		OptiMOS-P2 Power-Transistor, -30 V VDS, -80 A ID, PG-T0252-3-11, Reel, Green	2
N-Channel Mosfet, logic level		MOSFET N-CH 60V 100A TO220-3	1

0.1 $\mu$ F Capacitor		0.1 $\mu$ F Capacitor	1
10 $\Omega$ Resistor		10 $\Omega$ Resistor, 1/2w	1
100 k $\Omega$ Resistor		100 k $\Omega$ Resistor, 1/2w	1
JST-XH 2 Pin Female Port		CONN HEADER VERT 2POS 2.5MM	7
JST-XH 3 Pin Female Port		CONN HEADER VERT 3POS 2.5MM	1
XT-90 Connector Male		XT-90 Connector Male for power connections	3

XT-90 Female Connector		XT-90 Connector Female for power connections	3
JST-XH 2 Pin Male to Male Cable		JST-XH 2 pin male cable for temperature sensors	7
Arduino Controller PCB		PCB to hold Arduino, temperature monitoring, SoC monitoring and control circuits for HV Battery modules	1
M5 Bolts		M5 Bolts to secure batteries to copper bus bars	8
Copper Bus Bars		Copper bus bars used to connect cells within HV Battery Modules in series	6
3D Printed Shell		A 3D printed Shell and Lid to house the HV Battery Module components and cells	1

Hall Effect Current Sensor		LEM USA HAIS 100-P hall effect current sensor rated for 100A AC/DC and supply voltage of 5V	1
----------------------------	-----------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------	---

	
Figure ES5: HV Battery Module	Figure ES6: HV Battery (Exploded)

The HV battery system will be a part of the HV power system. It will be connected via 2 AWG cables (insulated with chemical, acid, UV, gasoline, and oil resistant EDPM sheathing) to the powertrain inverter via the pre-charge circuit system. The cables from the inverter to the EMRAX motor are also 2 AWG. When the pod is not in use, the HV battery modules can be removed individually and plugged into the ISDT Q8 balance charger as part of the charging system to be charged.

The HV battery modules will be connected together via the powerbar. The HV battery will be used to power the inverter and the EMRAX motor. It will operate at a nominal 252V and 60A during the demonstration. When the batteries have been discharged, they will require a mains power supply in order to power the charging system that will charge them.

The inverter control systems, the main flight Arduino controlling the inverter, and the software subsystems need to be in operation in order to operate the system. Otherwise, the HV battery

modules are all self-sufficient and have safety precautions built in to power down in case other supporting infrastructure malfunctions or the pod enters E-stop.

### HV Battery Safety

When using lithium ion cells at high currents, there are overcharge, overdischarge, over-current, and overheating risks which can all result in a thermal runaway. To combat these, each module is equipped with a Battery Management System (BMS). This protects each cell from being overcharged and over discharged in addition to balance charging them. The BMS also has over-current protection, which protects the cells and the system in the event of a short circuit. Each Energus Li4P25RT module is also equipped with 2 point temperature sensors, which the onboard Arduino Micro is constantly monitoring. The Micro will disconnect the cell from the circuit if the temperatures exceed software-defined thresholds. Furthermore, each cell is double fused to protect it and the rest of the system in the event of a short.

A further layer of protection has been built into the design of the HV battery in that each module can be charged individually. This allows us to charge the HV battery under much safer conditions, as each module can be charged individually at a lower voltage of 30V and only contains 500W of energy.

The Energus modules (4p configuration) will each be tested for their voltage when charged up to ensure that they are all reading the correct top voltage. Additionally, each HV module will be tested under its rated load of 30V and 60A continuously, several times, in order to ensure its reliability. During this time, temperature data and power drain of the cells will be monitored and recorded. The modules will also be assembled into the entire HV battery and will be retested under its rated load continuously until it is drained, with the same tests and monitoring being performed. Furthermore, each BMS will be tested to ensure that its overcharge, overdischarge and overcurrent protection mechanisms are all working. The Arduino system will also be tested extensively, to ensure that it cuts the battery module power when the cells reach a specified temperature limit. The unit tests performed to ensure these safety systems work are detailed in Table ES7.

### HV Battery Testing

A lot of these tests involve using the charging system. More details on the charging system are denoted in a section called “[Charging System](#)” further down below.

Table ES7: Table of Unit Tests for HVBM

Test Description	Purpose of Test	Expected Values
Measure voltage of each Energus Li4P25RT module	To ensure that the cells and modules aren't damaged. The cells are brand	$V_{cell} \approx 3.6V \pm 0.1$

when pack is first constructed	new and so there should not be much deviation between the cells. Voltage should be $V_{\text{nominal}}$ .	
Measure voltage of entire module when first constructed	To ensure that all the connections between the pack are working/connected.	$V_{\text{module}} \approx V_{\text{25V}} \pm 0.7V$
BMS undervoltage protection test. Apply small parasitic load across bms, with balance leads connected to variable voltage power supplies. Monitor current in load.	To test the undervoltage protection of BMS in ideal conditions. When the power supply voltage drops below $V_{\text{min}}$ of 2.5V, the BMS should disconnect the power from the load.	$V_{\text{cut}} = 2.5V - 2.7V$
Charge HVBM using ISDT charger at 1A & balancing, test cell and module	Make sure the cells can all be safely charged up to 4.2V at 1A. The temperature profile and voltage profile of HVBM will be recorded.	
Discharge HVBM using ISDT & BMS balance, test cell voltages and module voltage	Make sure the BMS can perform its under-voltage protection in real world application	
Charge HVBM using ISDT at 2A, monitor cell temperature through Arduino, test	Make sure the batteries and control system are able to charge under real world expected loads	
Discharge HVBM using ISDT at 3A	Make sure the batteries, BMS and control system are able to discharge under higher loads than 1A	
Stress test, discharge HVBM under artificial load at 60A continuous	Ensure all control systems and monitoring systems are working. Temperature will be monitored using Arduino and compared with values obtained from an IR thermometer. If temperatures exceed the threshold, the control system should activate.	
Stress test, discharge HVBM under artificial load at 80A peak loads with 60A continuous	Ensure all control systems and monitoring systems are working. Temperature will be monitored using Arduino and compared with values obtained from an IR thermometer. If temperatures exceed the threshold,	

	the control system should activate.	
--	-------------------------------------	--

### HV Battery Software Safety and Testing

A safety risk of the battery is that it may overheat, harming the battery and potentially causing the cells to catch fire. If the temperature of a cell within a battery module is outside of a safe range (0°C to 50°C), the Arduino will automatically disconnect it from the DC power strip. Once the temperature of every cell in the battery is within a safe range, the Arduino will automatically reconnect the module to the strip.

As an extra precaution, the Arduino's also able to disconnect the module if prompted by the flight controller. The temperature and SoC will be monitored on the flight controller to ensure that they are within range. The tests to ensure these software systems work seamlessly with the hardware are detailed in Table ES8.

Table ES8: Table of HVBM Software System Tests

Test Description	Purpose of Test	Expected Values
Compare the voltage drop of each temperature sensor using an Arduino with the voltage using a multimeter. This test will be done at different temperatures.	To ensure that the Arduinos are correctly reading the voltage.	The voltage received from the Arduino is expected to be the same as the voltage indicated on the multimeter.
Compare the SoC read by the Arduino vs the SoC read manually by reading voltage with a multimeter and converting it via the discharge characteristics graph.	To ensure that SoC monitoring is accurate.	The SoC read by the Arduino is expected to be the same as that from the multimeter reading and characteristics graph.
Compare temperatures read by the Arduino Micros from each temperature sensor with temperature given by an infrared thermometer. This test will be done at different temperatures.	To ensure that the Arduino Micros are converting the voltage input from the temperature sensors to the correct corresponding temperature.	The temperature received from the Arduino is expected to be the same as the voltage indicated on the temperature sensor.
Apply a voltage corresponding to a shut off temperature (1.59 V) to each pin to ensure that the battery	To ensure that the battery module will automatically disconnect if the temperature is out of range.	Pin D1 should be high > 3.3 V which will disconnect the battery module.

will disconnect.		
Apply a voltage corresponding to a shut off SoC to each pin to ensure that the battery will disconnect.	To ensure that the battery module will disconnect if the SoC is out of range.	Pin D1 should be high $> 3.3$ V which will disconnect the battery module.
Compare the highest temperature printed by the slave Arduinos with temperatures received from the master Arduino.	To ensure that the master Arduino is receiving the highest temperature from the slave Arduinos.	The master Arduino is expected to receive the highest temperature from the slave Arduinos.
Compare the temperature and SoC values printed by the Arduino Due with that printed by the flight controller.	To check the communication between the Arduino Due and flight controller.	The flight controller is expected to print the same values as the Arduino Due.
Disconnect each battery module using the flight controller.	To ensure that the battery module will disconnect if prompted by the flight controller.	The battery modules are expected to disconnect when prompted by the flight controller.

## **Motor Inverter System (MIS)**

The PM100DX motor inverter (shown in Figure ES7) was chosen to convert the DC power provided by the HVBM to the 3-phase AC required by the EMRAX 188 for operation. The motor inverter itself is powered with the LVBM. The MIS incorporates active thermal management and pre-charge circuit subsystems to ensure safe operation.



Figure ES7: Image of the PM100DX motor inverter used in the Pod

## Active Thermal Management Subsystem (ATMS)

The ATMS is designed to maintain safe operating temperatures for the PM100DX by removing excess heat from the system. The ATMS relies on conduction between the inverter's chiller plate and coolant (a 50/50 mix of ethylene glycol and distilled water). The system is driven by a pump (controlled with inputs from the ProSense VFS50 in-line combination temperature and flow sensor) to supply coolant to the inverter. An overflow tank filled with coolant doubles as a thermal reservoir for when the Pod is operated under near-vacuum conditions. The 5L tank is sized to be able to maintain the inverter's temperature within safe limits for a flight time of 4 minutes without the use of a radiator. The tank also features multiple auxiliary ports to which additional peripherals can be connected. A radiator loop can be connected in parallel for extended runtime and act as an additional factor of safety. The coolant tank will also be used to liquid cool the Pod's EMRAX 188 motor. Figure ES8 provides a high-level block diagram of the ATMS as it will be configured for EHW.

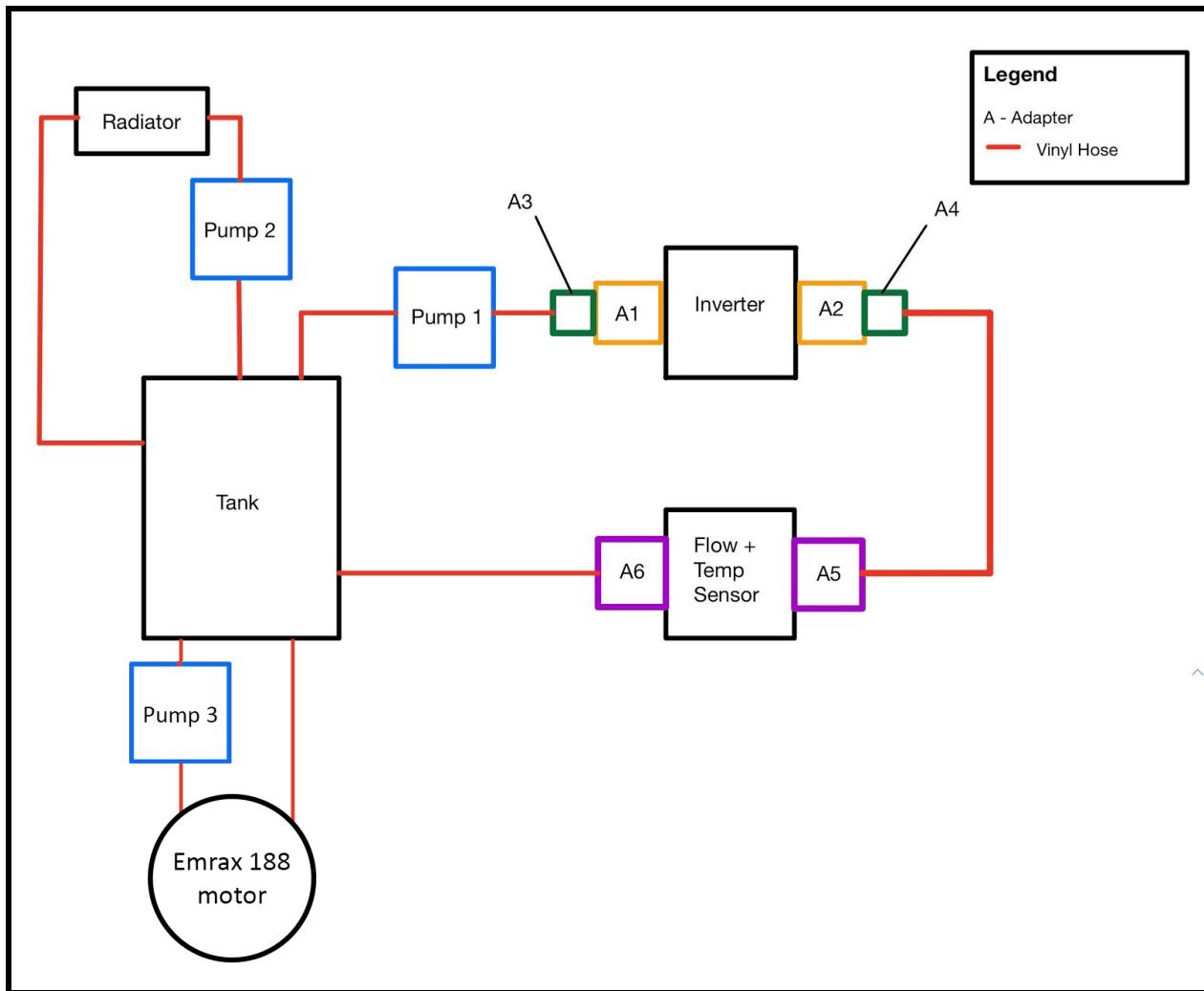


Figure ES8: The parallel radiator loop offers greater flexibility than a series system

### Testing

The ATMS' bench test bypasses the inverter by simply connecting its tube connections to each other, but maintains connections to the tank, pumps, radiators, and ProSense VFS50. Leak tests were first run with cold tap water to prevent burns in case of a leak. Once it was established that no leaks were present, the tank was filled with water boiled to approximately 80°C and the system's ability to remove heat was stress tested.

### Safety

To minimize the risk of system failures or accidents, every component is rated for the absolute maximum temperature of 80°C (the PM100DX's upper working limit) or greater and can operate under prolonged contact with water. Each adapter is properly sealed (either with teflon tape for threaded joints or properly cured contact cement for permanent connections) to prevent leaks. In the highly unlikely event of the ProSense VFS50 failing (considering it was designed for

monitoring water flow and temperatures in industrial applications) the ATMS will notify the Pod's on-board flight computers to trigger its shutdown sequence. Power to the pump will also be cut. While this may potentially damage the PM100DX as latent thermal energy within the coolant can cause the inverter to continue to heat up, this risk poses less of a safety concern than having the circulation system spring a leak near the HVBM and having no way of monitoring it. If the pump were to fail, the Pod's on-board flight computers would shut the inverter down before the upper working temperature limit is reached.

The hot coolant poses an additional safety concern, as water above 68°C will create severe burns (2nd & 3rd degree) nearly instantaneously. To prevent serious injuries, the radiator fans will continue to run until the coolant temperature has dropped to below 43°C (threshold considered safe for contact with skin). Members have been explicitly instructed to not tamper with the cooling system until the radiator fans have stopped running and the Pod's dashboard shows the temperature has dropped to below 43°C. These instructions and warnings have also been affixed to every component of the cooling system. Ethylene glycol is a known toxin that can be lethal even in small doses. MSDS warning labels and tags have also been affixed to every component of the cooling system to notify members. Members are instructed to both wear gloves and follow strict hand washing requirements every time they work with ethylene glycol to prevent cross contamination and accidental ingestion.

### Pre-Charge Circuit (PCC)

The PCC protects the inverter when a connection to the HV battery pack is made by limiting peak inrush current - thus successful integration with the MIS is key to ensuring safe operation. The DC+, J2-21, and J2-7 connections on the PM100DX must be securely connected to their respective components in the pre-charge system as per the schematic in Figure ES9. A capacitor discharge circuit is also part of this PCC. The capacitor discharge circuit is engaged at the end of the operation of the inverter, and used to discharge the capacitors within the inverter upon shutdown. This circuit is controlled by an Arduino and thus allows us to discharge the inverter whenever necessary. Table ES9 lists the PCC's major components and their specifications.

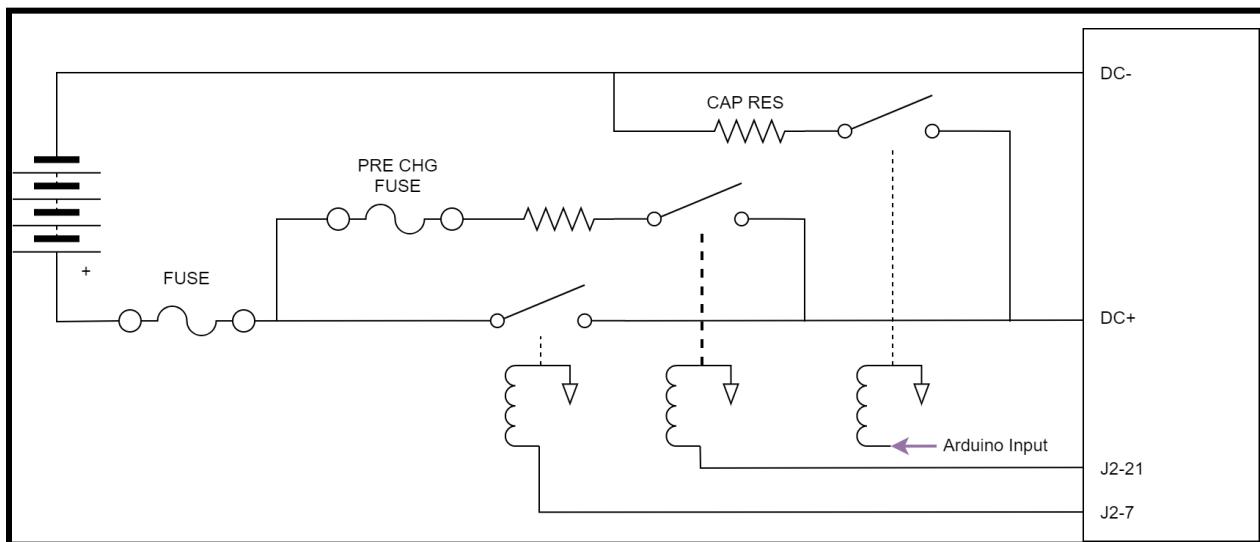


Figure ES9: The PCC schematic in relation to the HVBM and PM100DX

Table ES9: Bill of materials for PCC

Component Name	Component Description
Main fuse	400A, 700VDC, semiconductor fuse
Pre-charge fuse	5A, 500VDC
Pre-charge resistor	600 ohm, 50W
Pre-charge relay	SPST, 50A, 1200VDC, 12VDC coil
Main contactor	SPST, 500A, 900VDC, 12VDC coil
Capacitor discharge contactor	SPST, 50A, 1200VDC, 12VDC coil
Capacitor discharge resistor (CAP RES)	1KΩ, 300W

When the inverter is first powered on, it will engage the pre-charge relay to charge its internal capacitors. The pre-charge resistor is sized to quickly charge the internal capacitors, but not dissipate too much power if the inverter's input was shorted. The 5A fuse in series with the pre-charge resistor is added for additional safety protection. If the capacitors charge properly, the main contactor will engage. While the main fuse is not technically part of the pre-charge circuit, it is located in the same housing for simplicity of wiring. The PCC is relatively compact and light, measuring approximately 10" x 7.9" x 3.1" and weighing less than 1 pound. The most delicate component of the PCC is the pre-charge fuse as the ceramic cartridge may crack with excessive shock or stress. While this fuse is easily replaceable, and the other components are fairly robust, the PCC should be handled with care during transport.

### Testing

Given the critical role the pre-charge system plays in protecting the inverter, components must be tested beforehand. The main contactor can be tested by applying its coil voltage of 12V and hearing for a click, indicating the electromechanical relay has been engaged. A multimeter set to check for continuity can also be used to check if the contactor relay has closed. A similar testing procedure can be done to check the pre-charge relay. A multimeter set to measure resistance can be used to determine if the main and pre-charge fuses are faulty - a low resistance indicates good working condition while a 100% resistance state indicates the fuse requires replacement. A continuity check can also be used to test fuses. A visual inspection of all components and connections can check for signs of physical damage, misplaced connections, or loose wires. Unfortunately, there is no method of testing the fully assembled pre-charge system in isolation - short of tearing apart and reverse engineering the inverter, there is no way of replicating its capacitor charging sequence and logic with respect to the HV battery pack.

### Safety

The highest safety risk is posed when the pre-charge circuit is being connected to the HV battery pack. To prevent accidentally shorting the HV battery pack, the negative lead will be physically and electrically insulated.

## LV Battery Module (LVBM)

The LVBM is designed to support the onboard electronics, sensors, and motor inverter. It is made of 12 Samsung 18650 25R lithium-ion cells in a 6S2P configuration. This results in a battery pack with a nominal voltage of 24V and 20A of max continuous discharge current. To account for varying voltage requirements, the LVBM is connected to the Low Voltage Power Distribution System (LVPD) that provides 5V, 12V, and 24V DC of power across the Pod. These voltages will be hereinafter referred to as voltage rails. Table ES1 outlines specifications of the main components used in the LVBM. Figure ES10 shows a 3D model of the LVBM without its casing.

Table ES10: Bill of materials for LVBM

Component	Size (mm)	Description/Use	Quantity
INR 18650 25R Cells	18.33 diameter 64.85 height	- 3.6V, 20A continuous discharging	12
4-Cell Battery holders	80x76x18	- Holds 4 cells, pins are on the outside to make connections	3
Battery Management System (BMS)	34x59x9	- Balances the cells - Prevents overcharging and over discharging - Provides overcurrent and overvoltage protection	1
14-gauge copper wire (AWG)	1.63 diameter	- 20A current rating - American Wire Gauge Standards used	
Spade connectors	4.8 inner diameter Thickness: 0.35 (male) 0.5 (female)	- Used to connect balance wires to cells safely - Handles 18-14 gauge wire (by American Wire Gauge standards)	7
PCB	100x75x25	- Detailed in the PCB section below - Printed by JLCPCB	1

3D printed casing	266x96x73	- Box which encases the full LV system	1
-------------------	-----------	----------------------------------------	---

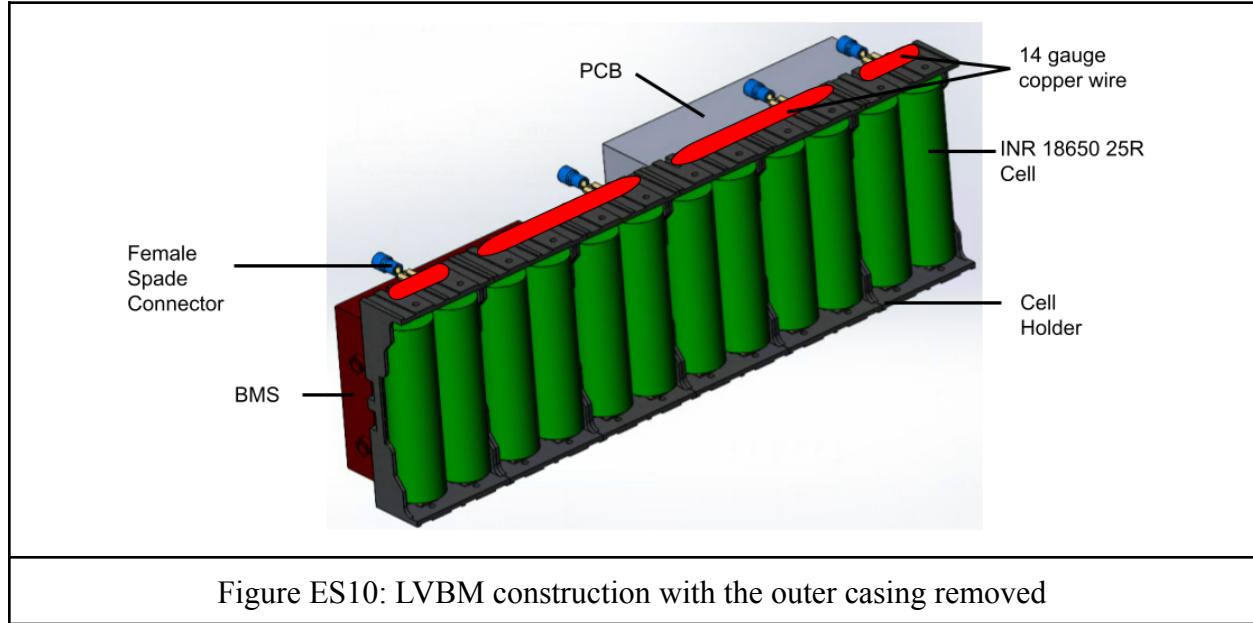


Figure ES10: LVBM construction with the outer casing removed

The schematic diagrams for the BMS' balance wire connections to the battery cells are shown in Figure ES11. The PCB used to monitor temperatures and the pack's state-of-charge (SoC) are shown in Figure ES12.

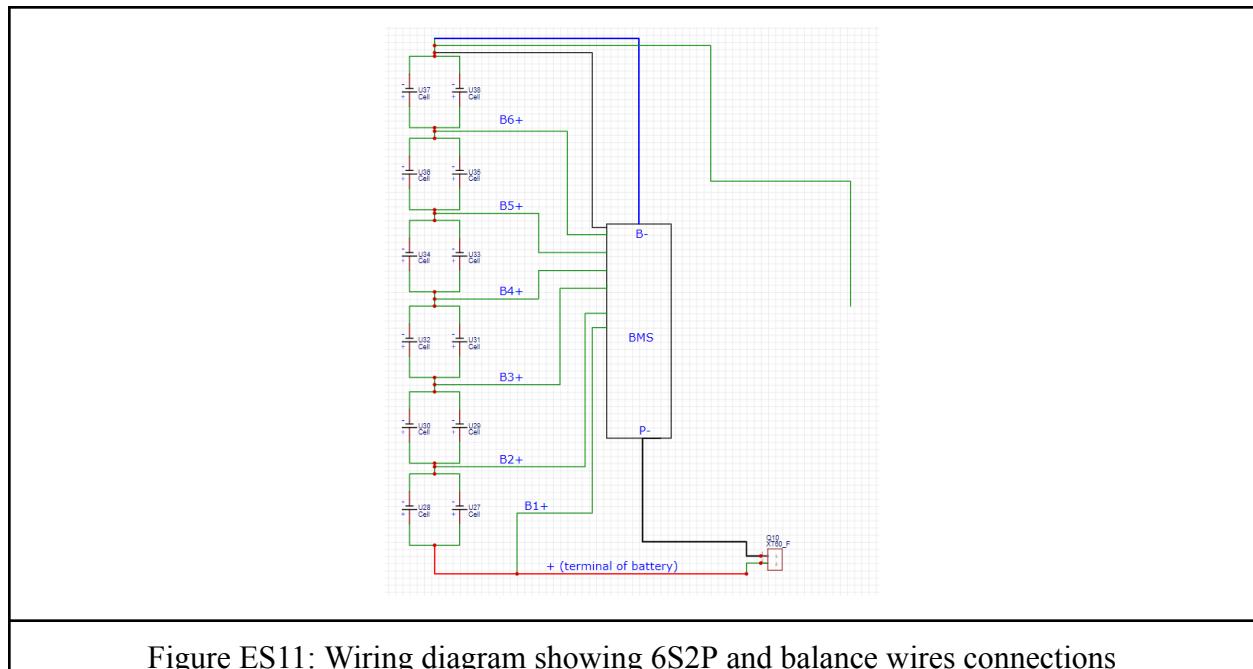


Figure ES11: Wiring diagram showing 6S2P and balance wires connections

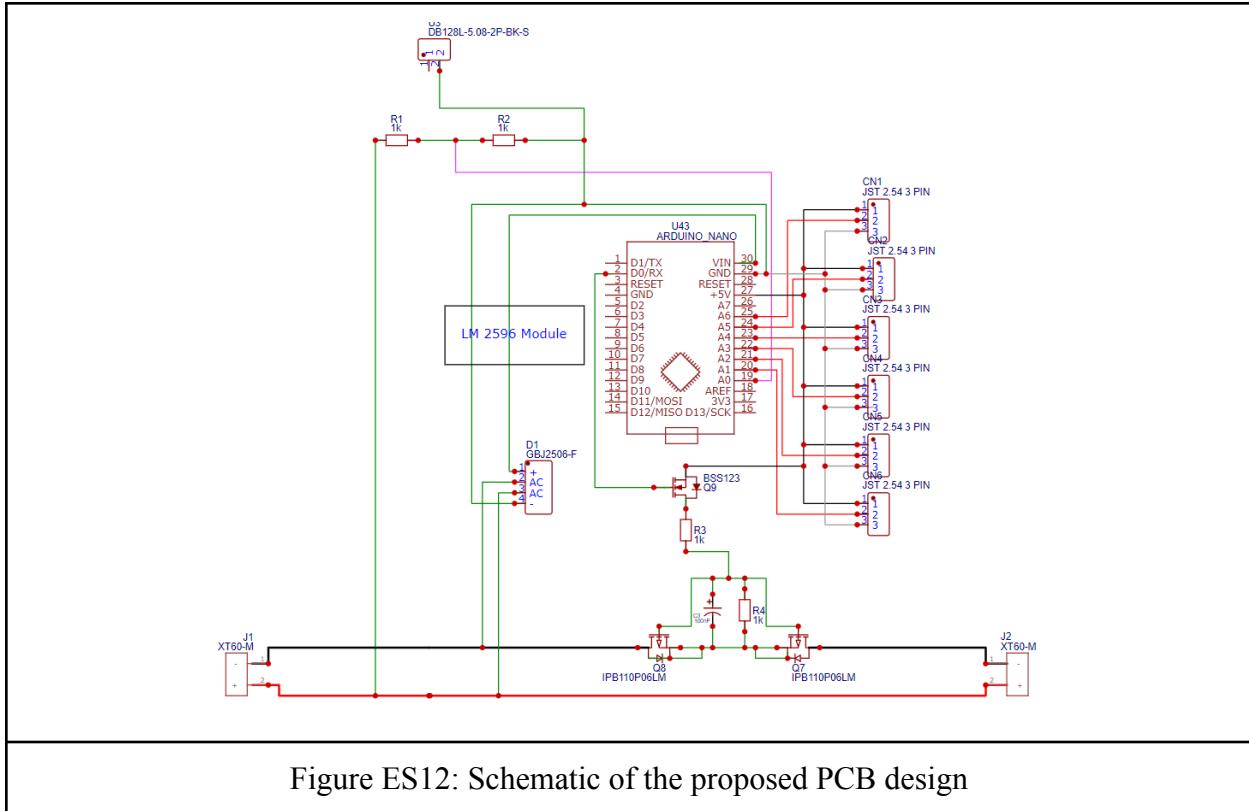


Figure ES12: Schematic of the proposed PCB design

### Battery Specifications

Table ES11 outlines the basic specifications of the low voltage battery.

Table ES11: Table of LV Battery Specifications

6S2P Config	
Series Cells	6
Nominal Cell Voltage (V)	3.6
Nominal Battery Voltage (V)	21.6
Minimum Cell Voltage (V)	2.7
Minimum Battery Voltage (V)	16.2
Maximum Cell Voltage (V)	4.2
Maximum Battery Voltage (V)	25.2
Max Current Consumed (A)	9.89

### PCB

This section focuses on the PCB of the low voltage battery, detailing each component and the layout. The PCB functions as a “shield” for the Arduino control system and streamlines the

manufacturing process of the batteries to allow us to connect all the various components required. The PCB is detailed in Figure ES13, and the components are listed in Table ES12.

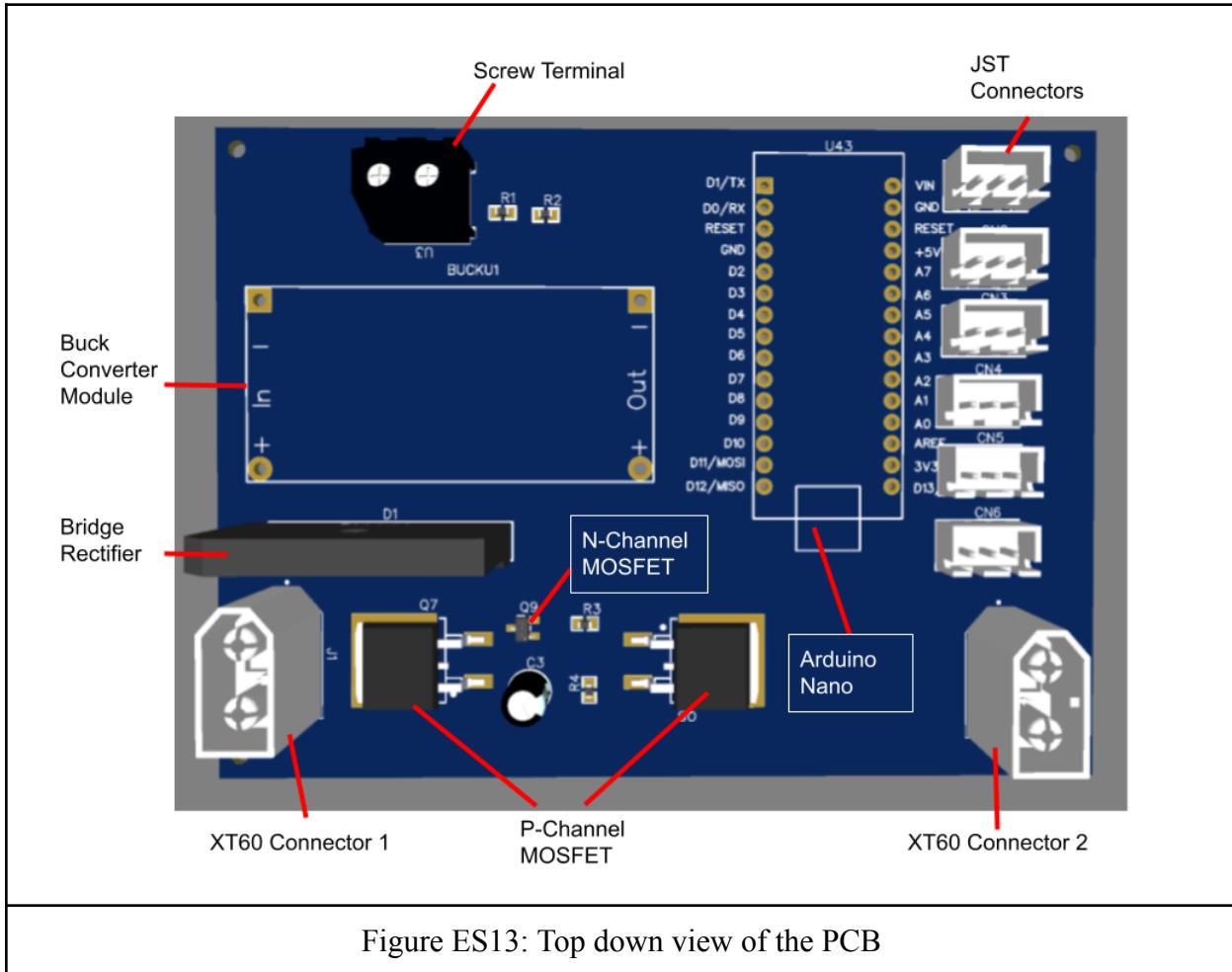


Figure ES13: Top down view of the PCB

Table ES12: Table of PCB Components

Main Components	Size (L*W*H) mm	Use
Arduino Nano	18x45	Connects to the TMP36 (Temperature measuring diodes) through the JST connectors to monitor the temperatures of the cells and ensure the prevention of overheating of the batteries
JST Connectors	10*5.75*10.4	Allow the TMP36 diodes to be connected to the Arduino.  Have a pitch of 2.5mm which matches the TMP36 diodes.
TMP36 Connectors		Measure the temperature of the battery and send

		these signals to the Arduino. These diodes are able to measure a range of -40 to 125 °C. Which is above the recommended charging temperature of 50 °C.
P Channel MOSFETs (IPB110P06LM)	10.31*15.88*4.57	<p>Power MOSFET that makes up part of the bidirectional switch that allows the Arduino to disconnect the battery when the battery is overheating.</p> <p>This power MOSFET has a high <math>V_{DS}</math> of -60V and a maximum continuous drain current of -100A; Both of which far exceeds the requirements of the Low Voltage Battery.</p>
N Channel MOSFET (BSS123)	3*2.55*1.15	<p>Logic MOSFET that connects to the Arduino and controls the on/off state of the bidirectional switch.</p> <p>As it is a logic MOSFET it does not need a very high tolerance to voltage and current. So its <math>V_{DS}</math> of 100V and continuous drain current of 0.17A is sufficient.</p>
Screw Terminal (DB128L-5.08-2P-BK-S )	10.2*10*14	<p>Allows a wire to be connected to the PCB safely.</p> <p>Is rated for 250V @ 18A which exceeds the maximum output voltage specifications of the Low voltage battery (25.224V)</p>
XT60 Connector 1	15.5*8.10*20.20	Connects the PCB to the positive terminal of the battery and the P- terminal of the BMS. Rated for 500V and 30A exceeding the output of the battery which is 24V nominal a maximum of 25.2V.
XT60 Connector 2	15.5*8.10*20.20	Acts as the common port of the battery, allowing it to be charged or discharged. The rating of 500V and 30A exceeds the charging requirements. Where the charger operates between 0.1-20 A & chargers output voltage is DC 1-34 V.
Bridge Rectifier (GBJ2506-F)	30*4.8*24.2	<p>Converts the input into a DC Current, which can be used to power the Arduino.</p> <p>It has a reverse working maximum voltage of 600V which is higher than the 25.24V maximum from the battery or the 1-34V from the charger.</p>
Buck Converter Module	45*20*14	Steps down the voltage from the bridge rectifier to 5V in order to power the Arduino.

(LM2596)		The max input voltage of 40V is greater than the battery or charger voltage. Additionally it can output a maximum of 3A which is sufficient to power the Arduino nano which draws 19mA and the 6 temperature sensors which only draw
----------	--	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

### Operation and Testing

The low voltage battery system does not require any other systems during operation. However, charging systems will be required before operation. Components within the low voltage system itself must also be functional to ensure safety. We will thus conduct tests for various components prior to operating the battery.

#### PCB Testing:

Components like the TMP36 diodes can be tested individually, where their readings of the temperature of a surface can be compared to the reading from an infrared thermometer to ensure that the diodes give an accurate reading.

To test the board itself, first perform a quick visual inspection to ensure that none of the components are damaged or if some solder has overflowed and is connecting separate pins. Next, short circuits can be found by using a digital multimeter, by using the probes to see if two components that are part of two separate circuits have a small resistance. The PCB can be connected to a power supply and a test load to verify with a multimeter if the buck converter and bridge rectifier are operational and are able to rectify and step down the voltage from the battery to power the Arduino. Finally, the PCB can be tested to see if it is able to disconnect the battery when the temperatures are too high when discharging or charging. This can be done by heating one of the temperature sensors and measuring the current at the battery to ensure there will be no current reading.

#### INR 18650 25R Cell Testing:

The voltage of each parallel set of cells (two cells) will be measured with a voltmeter. The reading of each when fully charged should be 4.2V. The capacitance of each cell will be measured using a multimeter to measure the cell's state of health. Each reading should be 2500mAh.

#### BMS Testing:

In order to test the battery management system a test will have to be done while the battery is being charged to determine if it can properly balance each of the cells. This will be done using a multimeter or a voltmeter to check the voltage of the battery through both the charge and discharge wires. If the voltage is too low, double check all main and balance connections. The

battery will also be tested under a test load. The test load can be created using resistors or a heating element to create a power draw similar to the actual load to ensure that it provides the required voltage. It will be checked for heating under operating conditions using an infrared laser thermometer. The battery is not expected to heat up too much and stay within the 35-50°C range.

### Safety

One danger of the battery is that the individual cells can charge/discharge at different rates and become unbalanced. Additionally the batteries can become damaged due to either overcharging or over-discharging. This poses safety risks as over-charging can cause cells to radiate heat and over-discharging can cause damage to cells. The BMS integrated into the system works to prevent this from occurring.

Additionally when charging or discharging, the temperature of the battery can quickly rise which reduces the useful life of the battery and can lead to the cells catching fire. To prevent this a temperature sensor will be connected to each cell. These temperature sensors will also connect to the Arduino which will monitor the temperature reading from each sensor and if this temperature exceeds 50°C, it will stop the charging/discharging.

Prior to transporting the LV system, the operator must make sure that all parts are tightly bound and no loose connections exist. During transportation of the battery, two people are required for safe lifting and moving and caution is taken to ensure no metal objects in the environment are able to make contact with the battery. If the system is to be transported to areas farther away, it must only be transported once secured inside its casing.

Maintenance of the battery pack can have safety risks to the operator due to the possibility of creating shorts if, for example, the battery terminals come into contact with each other or metal objects around the battery. One way we have significantly reduced this risk is by using spade connectors between the cells and the balance wires from the BMS. Female spade connectors are soldered to the cell holders and male spade connectors with balance wires can be easily connected/disconnected when needed. This means that if the cell holder or BMS need to be replaced, it can be done easily without having to remove soldered connections. Additionally, the use of cell holders mean that the 18650 25R cells can be replaced easily if damaged. Using cell holders also avoids the need to directly solder onto cells, which in turn minimizes direct heat to cells and potential shorts.

Furthermore, safety protocols were strictly followed during the engineering and construction of the LV and HV systems. Some of which are outlined below:

- Two ABC fire extinguishers are kept nearby while constructing, testing, and operating.
- Area has an active fire detection system.
- Appropriate PPE are worn.
- No debris (especially metal) is on a surface where any battery cells are kept.

- Surface is kept clean and a white paper is placed on top before starting to easily spot any new hazards created during construction.
- No metallic accessories were allowed around the workbench.
- Appropriate gloves were used when handling electronic components to avoid short circuits caused by sweat from hands.
- New hazards created throughout the process such as loose metallic parts were eliminated promptly.

Additional safety precautions in system design:

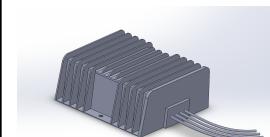
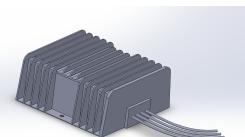
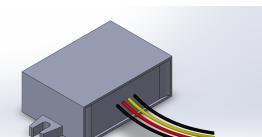
- Fish paper will be used on all exposed wire surfaces including the 14-gauge wire and the spade connectors. Additionally, a piece of kapton tape will be placed above each connection between the cells to secure them and ensure that they cannot move freely which may lead to a short.
- Kapton and electrical tape will be used to further add insulation between the BMS, PCB, and cell holders and to prevent wires or metal from escaping the enclosure.

## LV Power Distribution System

### System Description

This system is connected to the LV battery system, which then converts the 24 V from the battery to 3 different voltage rails (24V, 12V, 5V) by using boost buck converters. It is designed to provide power to all the electronics in the pod at the rated voltage that they can take. It is responsible for converting the power from the LV battery into the corresponding appropriate voltage that each electronic component can receive.

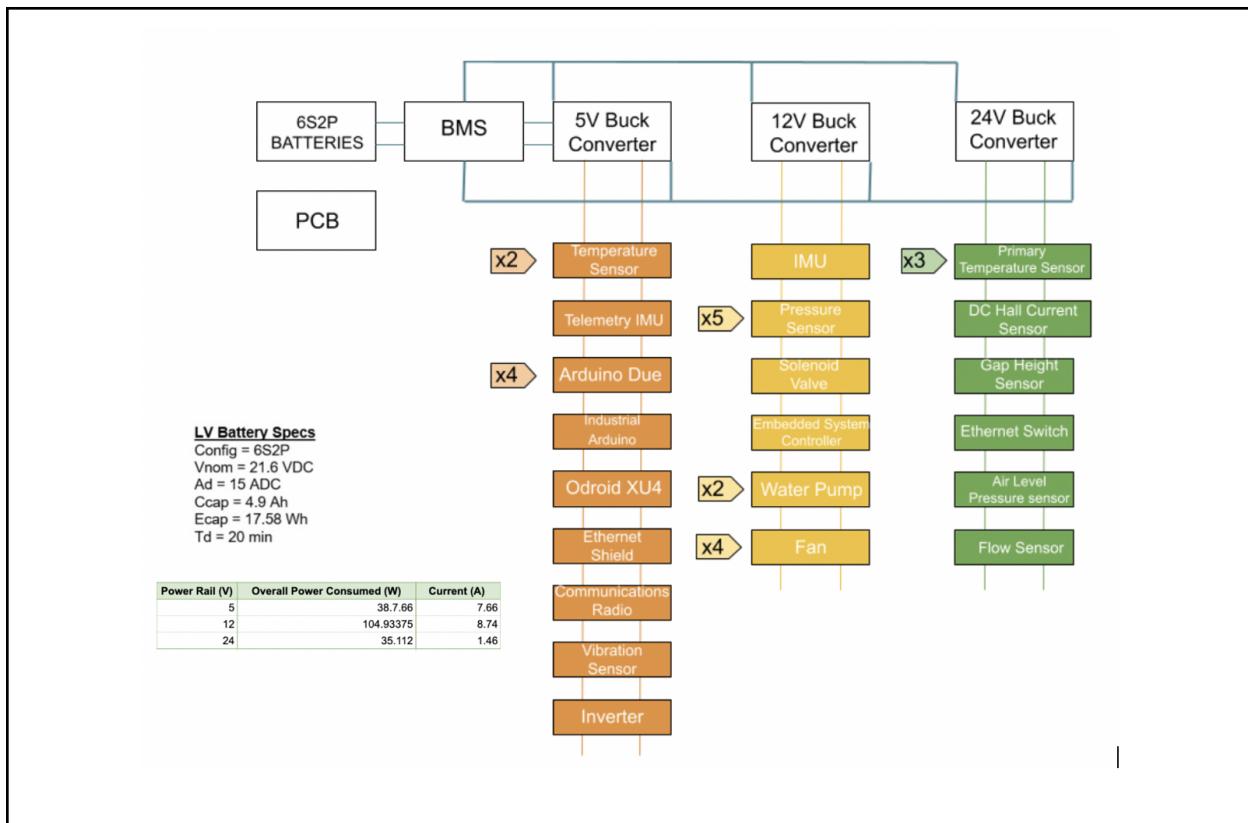
Table ES13: Converter Specifications

	24V boost/buck converter	12V buck converter	5V buck converter
Dimension	74*74*32 mm	74*74*32 mm	115.1*78*31 mm
Input Voltage Range	9-36 V	9-36 V	6-40 V
Output Current	2A	10A	20A
CAD model			

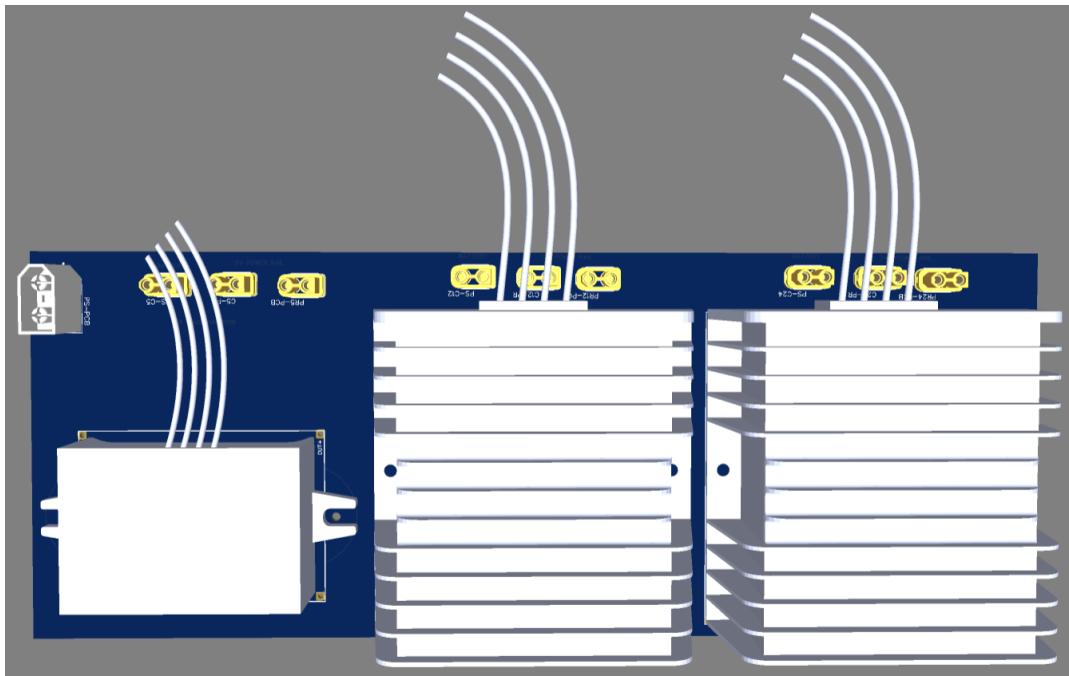
The LV power distribution system converts and distributes power to all the sensors, Arduinos, main flight controller and the inverter controller. An XT-30 connector will be used to connect the LV battery to the LV power distribution system. The connectors will have the input power wired in parallel to feed into the different converters forming three different power rails. The individual power rails will then use connectors shown in Table ES14 to connect to different downstream components. The LV battery needs to be operational in order for this system to run. Figure ES14 and ES15 shows the general layout and the 3D PCB representation of the low voltage power distribution system. An example of the PCB layout for the power rail is shown in Figure ES16. A 3D CAD model to show how the power distribution rails & connectors look is shown in Figure ES17.

Table ES14: Power Rail Connectors Specifications

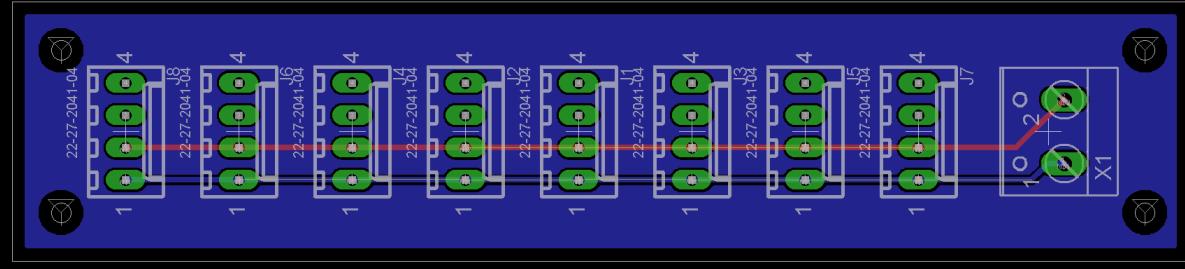
	5V Rail	12V Rail	24V Rail
<b>Series</b>	Molex Nano-Fit 105311	Molex Nano-Fit 105311	Molex Nano-Fit 105311
<b>Number of pins</b>	2 POS	3 POS	4 POS
<b>Pitch</b>	2.50mm	2.50mm	2.50mm



**Figure ES14:** General layout of the LV distribution system



**Figure ES15:** 3D Representation of the PCB Layout for Power Converting



**Figure ES16:** PCB Layout for the 24V Power Rail

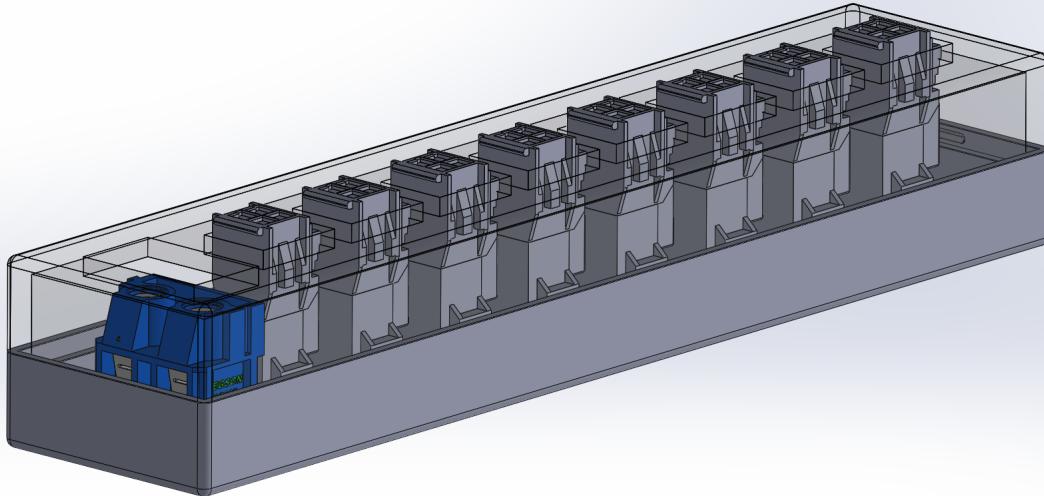


Figure ES17: 24V Power Rail Assembly (top) Exploded View (bottom)

### LV Power Distribution System Safety

Since the low voltage distribution system is responsible for delivering different power voltages to different downstream components, one of the major safety risks that may occur is that if the wires connecting the downstream component to the power rails are mismatched (i.e delivering higher voltage to components that have a lower voltage range) the downstream components could possibly be damaged. To eliminate the potential problem and ensure the safety of the downstream components, different connectors with different numbers of pins are used for each power rail making it impossible to connect the components to the wrong power rail. It is important to note that while connectors with 4 and 3 pins are used, only pin 1 and pin 2 will be used to connect to ground and power sources.

The boost-buck converters and connectors used in the LV power distribution system are off-the-shelf products that satisfy the commercial safety standards. The components were selected to ensure it can withstand the maximum current and voltage that needs to be delivered. The width of the traces for the PCB (3 mm for power rails, 6 mm for power converting PCB) are also carefully designed to make sure it can supply the maximum current required.

Although the LV power distribution system is only delivering power with low voltage and current, safety protocols should still be followed when assembling the system.

- Make sure all power source are disconnected when handling the components

- Always use insulated tools while working
- Keep all sources of water away from the working area
- Always ensure that all electronics equipment is properly grounded
- Make sure safety equipment and basic first-aid kit can be easily accessed

### Testing

As the battery system discharges, its voltage will drop causing damage to the downstream components or wrong readings to the sensors. Buck converters are therefore implemented to ensure a constant voltage supply. Performance test is outlined in Table ES14 to make sure the converters are able to deliver the desired voltage.

Table ES14: Performance tests for the converters and connectors in the LV Power System.

Component	Test	Target
Converters	<u>Efficacy Test</u> For each converter, a lower and higher input voltage will be supplied and the output of the converter would be measured to check its ability to convert the voltage to the desired values.	Voltmeter reading of 5V for 5V Buck Converter  Voltmeter reading of 12V for 12V Buck Converter  Voltmeter reading of 24V for 24V Boost-Buck Converter
	<u>Load Regulation Test</u> For each converter, at a constant input voltage, current fluctuating between a minimum of 4A to a maximum of 40A will be supplied and the output voltage would be measured to ensure these converters are capable of maintaining the desired output voltage.	<ul style="list-style-type: none"> <li>• A constant voltmeter reading of 5V for 5V Buck Converter</li> <li>• A constant voltmeter reading of 12V for 12V Buck Converter</li> <li>• A constant voltmeter reading of 24V for 24V Boost-Buck Converter</li> </ul>
	<u>Line Regulation Test</u> For each converter, an input voltage fluctuating between a minimum of 18V to a maximum of 25.2V will be supplied and the output voltage would be measured to ensure these converters are capable of maintaining the desired output voltage.	

Connectors	For each connector, a 5, 12 or 24 volt of power will be supplied to the female end and measured at the male end.	<ul style="list-style-type: none"> <li>• A constant voltmeter reading of 5V for 5V supply</li> <li>• A constant voltmeter reading of 12V for 12V supply</li> <li>• A constant voltmeter reading of 24V for 24V supply</li> </ul>
------------	------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

## **Charging System**

### **System Description**

The charging system charges the high- and low-voltage battery modules that power the hyperloop pod. After the high- and low-voltage battery modules have been discharged, removed from the pod, and cooled down to room temperature, they can be re-charged using the charging system. The charging system consists of a server power supply unit (PSU) that converts from AC to DC (detailed specifications below), and a balancing charger that uses the output of the PSU to charge the battery cells. The charger allows the user to specify the charging current and voltage and charge the battery at a constant-current, constant-voltage (CC-CV) charge profile. Moreover, if the competition location has a mains electricity and/or a socket type that differs from the PSU input requirements, a transformer and/or a plug adapter will be used in conjunction with the PSU. The charging system is equipped to be able to charge both the LV and HV battery modules using the same set-up.

Table ES15: Size, components, and appearance of the system

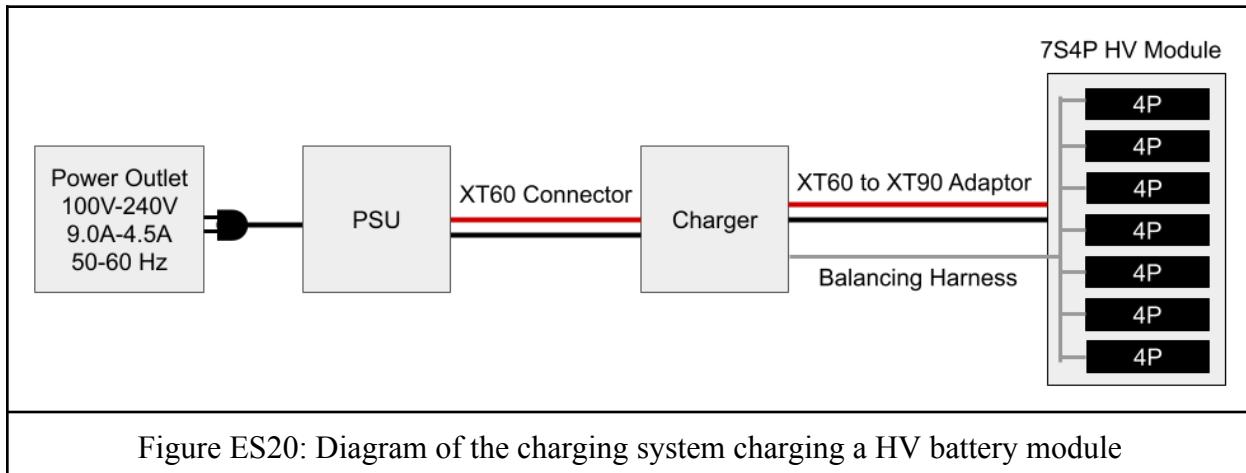
Component	Specifications	Use
Switching Power Supply Unit (PSU)	<p>Model: HP DPS-750RB A  Dimensions: 23.8 cm x 3.8cm x 8.6cm  Specifications:  Input: 100V-240V ~/ 9.0A-4.5A  Output: 750W (MAX)</p> <ul style="list-style-type: none"> <li>• +12.0V -- / 62.5A</li> <li>• +12VSB (Voltage stand by) -- 2.5A</li> </ul>	Converts current from AC to DC for charging.
Balancing Charger	<p>Model: ISDT Q8 Balance Charger  Dimensions: 8cm x 8cm x 3.3cm  Specifications:</p> <ul style="list-style-type: none"> <li>• Max input current: 20 A</li> <li>• Input voltage: DC 10-34V</li> </ul>	Balance-charges each cell of the HV and LV battery module.

	<ul style="list-style-type: none"> <li>• Output voltage: DC 1-34 V</li> <li>• Balance current: 1.5A/cell maximum</li> <li>• Charging current: 0.1-20 A</li> <li>• Discharging current: 0.1-1.5A</li> <li>• Max charging power: 500 W</li> <li>• Max discharging power: 15 W</li> <li>• Supported battery types and cell count (in series): LiFe 1~8S, Li Ion 1~8S, LiPi 1~8S, LiHv 1~7S, Pb 1~12S, NiMH/Cd 1~16S</li> </ul>	
XT60 to XT90 Adaptor	Length: 21cm <ul style="list-style-type: none"> <li>• 10 AWG 200°C wire</li> </ul>	Connects the HV battery module terminal (XT90) to the charger port (XT60).
XT60 Connector	The connector is soldered to the power supply unit. Length: 9cm <ul style="list-style-type: none"> <li>• 14 AWG 200°C wire</li> </ul>	Connects the power supply unit to the charger.
Balancing Harness	Length: 30cm <ul style="list-style-type: none"> <li>• 22 AWG wire</li> </ul>	Connects the cells in the battery module to the balancing ports of the charger.
Type-B SoCket Adaptor	N/A	Required for plugging the power supply unit into non-type-B SoCkets in Europe.

	
Figure ES18: The switching power supply unit (HP DPS-750RB A)	Figure ES19: The charger (IDT Q8 Balance Charger)

### Integration of the system into a subordinate structure/system

The charging system operates independently of other energy system components. The charging system is responsible for charging the battery modules that have been discharged, removed from the pod, and cooled down to room temperature.



### System Operation during Demonstration

The following procedure outlines the demonstration of the charging system:

1. Connect the battery module and the charger using the XT90 to XT60 connector. The battery port on the charger is labelled as “BATTERY.”
2. Connect the balancing leads on the battery module to the balancing ports of the charger. Ensure that the negative terminal (black wire) is connected to the corresponding port (labelled as “-”) on the charger.
3. Connect the charger to the power supply unit using the XT60 connector soldered onto the power supply.
4. Connect the power supply to a plug adapter, if applicable. Then, connect the power supply unit to an outlet rated for 100V-240V ~/ 9.0A-4.5A.
5. Operate the charger:
  - a. Long press the middle key to enter System Settings. Check that “Lowest Input Voltage” is set to 10.0V, and “Max Input Power” is set to 550W.
  - b. Press the middle key to enter the Task Setting menu. Enter the following parameters:

Task	Charge
Chemistry	LiIon
Condition	4.20V

Cells	7S
Current	4.0A

- c. Select “Start” to start charging, and long press the middle button to navigate to the main interface and monitor the cells while charging.
- 6. Once the charger indicates that the battery module has been fully charged, disconnect the battery module from the charger. The demonstrator may then choose to charge another battery module, or shut down the system.
- 7. To shut down the system, power off the power supply unit, then disconnect the charger from the power supply. Be careful as the power supply unit and the charger might be hot due to operation.

The charging system requires the EHW to provide a power outlet that meets the input requirements of the PSU (120-240V, 9.0A-4.5A, 60 Hz) and a Type B plug adapter, if applicable. UTHT will be responsible for the rest of the required infrastructure, including the PSU, the charger, and connectors.

### Manufacturing of the Charging System

The power supply unit and the charger are off-the-shelf products that adhere to various commercial safety standards. The connectors were securely soldered to the PSU’s output wires and insulated with electrical tape and/or heat shrink. The wire gauges were chosen with a factor of safety of 3, so that the maximum current flowing through the wire does not exceed the rated current for the wire at any point. In addition, XT60 and XT90 connectors were used to avoid potential mistakes of switching positive and negative terminals.

### Transporting the Charging System

The charging system is modular and safe to transport. The charger, the power supply unit, and the multimeter will be wrapped with sufficient cushioning to avoid damages during transportation. All the detachable connectors will be contained in a separate compartment to avoid any risk of shorting.

### Safety

The key elements of the system are the power supply, batteries, and charger. The power supply involves the highest safety risks. These safety risks are mitigated by insulating the power supply with fish paper and its terminals with electrical tape. The following guidelines are quoted directly from the manufacturer, and care must be taken to adhere to them: “The power supply should be operated in an environment without excessive amounts of dust, liquids, foreign matter, or corrosive gases; the rated operating temperature, operating humidity, and storage temperature

must be followed; the vibration and shock resistance limits must not be exceeded; the power supply must be installed away from devices that produce strong, high-frequency noise and surge; and the power supply must be stored out of direct sunlight.”

To further mitigate safety risks, the charger will also be stored in an environment without dust, humidity, rain, high temperatures, direct exposure to sunlight, intense vibration, contact with flammable objects, and contact with explosive objects. All system elements operate on surfaces that are heat-resistant, non-flammable, and insulating.

The charger and the power supply unit are off-the-shelf products that adhere to commercial safety standards. To ensure compatibility, these components were purchased after careful considerations. For the harnesses and adapters, the use of XT60/XT90 connectors was an intentional design decision to avoid potential mishaps of switching positive and negative terminals. Moreover, the adapters have a factor of safety of 3 to ensure that the operating current in the wires do not exceed the rated current for the wire gauge.

The power supply unit contains a capacitor, which stores energy during operation. Therefore, after unplugging the power supply, sufficient time must be given for the capacitor to discharge before transporting the power supply.

The charging system will be thoroughly tested prior to EHW 2021. In our testing process, we charge the LV and HV battery modules under room temperatures and under different charging currents to collect data (i.e., state of charge and temperature). We also test for the accuracy of our monitoring system by comparing the current, voltage, and temperature values measured by the monitoring system against the values measured by external instrumentation (e.g., a multimeter and an IR thermometer).

Table ES16: Charging system testing procedures

Component	Test Description	Target
Battery	Monitor the battery temperature, state of charge, and time to full charge under different values of charging currents (maximum 16A).	Determine the optimal charging current.
Current and voltage monitoring	During the charging operation, record the current and voltage values measured by the monitoring system of the charger, and measure the current and voltage with a multimeter. Compare the measured values.	The percentage error should be less than 5%.
Temperature monitoring	During the charging operation, record the temperature values measured by the Arduino and measure the cell temperatures with a IR thermometer. Compare the	The percentage error should be less than 5%.

	measured values.	
--	------------------	--

## Electronics

### General Overview

The UTHT Electronics System provides the low-level hardware and software needed to control and monitor the Pod. The electronics system accomplishes this by integrating various electronic components and sensors together, as depicted in Figure E1.

- A [sensor system](#) responsible for monitoring the current state of the pod
- A [low-level controls system](#) responsible for the low-level propulsion and brake controls
- An [infrastructure system](#) to responsible for the electrical connections between the sensors, microcontrollers, and controls system to the pod's [software system](#)

The subsequent subsections will provide more in-depth explanations of the sensors, motor inverter, brake controls, microcontrollers, and the interconnections between the electronic components. Their respective testing and safety procedures will also be detailed.

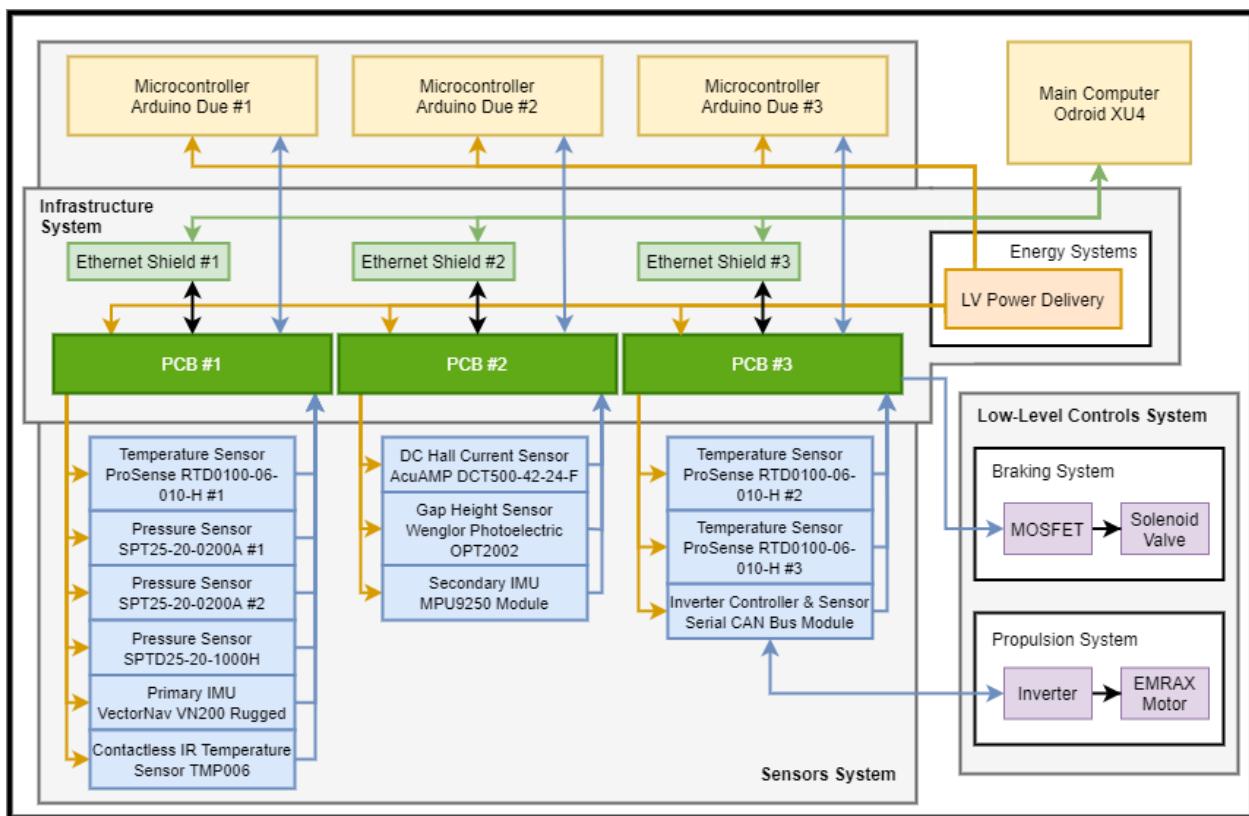


Figure E1: A high-level block diagram of the major electronics system components

## Sensor System

The sensor system is designed to monitor the state of the pod. The electronics team has worked closely with all other subsystems, ensuring that all monitoring needs of the other subsystems are met. The sensor system monitors the pod's inertial measurements, braking system, region temperatures, vibrations, high voltage DC current, and vertical movement. Three Arduino Due microcontrollers and accompanying printed circuit boards will be used to both send power to the sensors as well as collect and process their data output. [Appendix E1](#) and [E2](#) contain a detailed breakdown of all sensors and all microcontrollers respectively, about their integration with the rest of the electronics system for monitoring and control. The printed circuit boards are discussed in greater detail in the [electronics infrastructure section](#). Tables E2, E3, E4 below breakdown each microcontroller regarding which sensors it interfaces with, and the sensors' respective function towards monitoring the pod's state.

Table E2: Table of Arduino Due #1 Sensors

<b>Sensor Name</b>	<b>Communication Protocol</b>	<b>Respective Subsystem</b>	<b>Function</b>
SPTD25-20-1000H Pressure Sensor	Analog	Braking System	Monitor the pressure at the main brake reserve tank at the center of the pod
SPT25-20-200A Pressure Sensor #1 & #2	Analog	Braking System	Both monitor the pressure of the braking exhaust valves at the bottom of the pod
ProSense RTD0100-06-010-H #1	Analog	Region Temperature Monitoring	Monitor the temperature near the center of the pod, near the main electrical components
TMP006 Contactless IR Temperature Sensor	I2C Serial	Region Temperature Monitoring	Monitor the temperature around the center of the pod, the additional temperature data will be logged on the master computer for post-run internal pod temperature analysis
VectorNav VN-200 Rugged	TTL Serial	Inertial Measurement	Monitor pod's inertial measurements, such as the pod's position, velocity, and acceleration on the track

Table E3: Table of Arduino Due #2 Sensors

Sensor Name	Communication Protocol	Respective Subsystem	Function
Wenglor Photoelectric OPT2002	Analog	Vertical Movement Measurement	Track the vertical distance between the bottom of the pod and the top of the I-beam track, for suspension tuning and emergency braking
MPU9250	I2C Serial	Vibration Analysis	Monitor accelerometer data to log the data on the master computer for post-run vibration analysis
AcuAMP DCT500-42-24-F	Analog	Energy Systems	Monitor current output from the high voltage battery system to the inverter/motor for safety

Table E4: Table of Arduino Due #3 Sensors

Sensor Name	Communication Protocol	Respective Subsystem	Function
ProSense RTD0100-06-010-H #2 & #3	Analog	Braking System	Monitor the temperature near the front and back wheel of the pod
Serial CAN Bus Module	Serial (UART to CAN)	Propulsion System	Establish a connection between the microcontrollers and the inverter, and monitor the motor vitals during operation, such as its temperature, motor frequency, current and operational state. The Serial CAN Bus Module also controls the inverter, discussed in the <a href="#">low-level control systems</a> section.

The three Arduino Due microcontrollers communicate the processed sensor data to the pod's master computer (Odroid XU4), using the [electronics infrastructure system](#), and the [software communication system](#). The Odroid XU4 will monitor the sensor data, and make high-level decisions regarding the pod's movement and relaying information to the pilot computer. The high-level decision making is discussed in greater detail in the [software control systems](#). The MPU9250 and the TMP006 Contactless IR Temperature Sensor will be used as auxiliary sensors,

for the team to constantly improve our pod through a detailed analysis of the data from test runs and the sensor data will be logged onto the Odroid XU4. However, the auxiliary sensor data will not be used for any decision making and is not critical to any monitoring functionality. Table E5, illustrates which sensors are primary (a part of the Odroid XU4 decision making and will operate at a high refresh rate) and auxiliary.

Table E5: The sensors divided into primary sensors and auxiliary sensors

Primary	Auxiliary
SPTD25-20-1000H Pressure Sensor SPT25-20-200A Pressure Sensor ProSense RTD0100-06-010-H VectorNav VN-200 Rugged OPT2002 AcuAMP DCT500-42-24-F Serial CAN Bus Module	MPU9250 TMP006 Contactless IR Temperature Sensor

For the sensor system to operate properly and consistently, several other systems must run correctly in accordance with the sensors. These systems are as follows:

The sensor system relies on the [electronics infrastructure system](#) to implement the wiring connections between the microcontrollers and their respective sensors. This system must be able to consistently power the sensors and microcontrollers. In addition, it must establish reliable electrical connections, required for communication between the microcontrollers (Arduino Due's) and the master computer (Odroid XU4). These electrical connections include those embedded into the three PCB's, which act as the intermediary between the sensors and microcontrollers, as the sensors are either directly mounted on the PCB, or connected via the PCB connectors. Failure to maintain a proper communication channel may lead sensor or system malfunction, causing the pod to emergency stop.

The electrical power required to power the sensors and microcontrollers comes from the energy systems' [Low Voltage Battery Module \(LVBM\)](#). Consistent power delivery is critical to sensor operation, otherwise the data outputs may not be representative of the actual state of the pod, leading to poor or incorrect decision making.

To maximize the accuracy of the sensor readings as much as possible, the structures system handles the secure mounting methods and optimal mounting locations of the various sensors. More details are provided in the [Sensor Mounting Overview](#) section.

The sensor system depends on the software system for communications. The software communication system is required to communicate sensor data to the Odroid XU4. Failure to

maintain consistent communication will trigger the low level control systems to emergency stop the pod. Refer to the [low level control systems](#) section to more details about this safety protocol.

### Sensor Mounting

Table E7, discusses the mounting details with each sensor, including the various mounting methods, location, materials, fasteners, and installation. The mounting locations are as depicted in Figure E6.

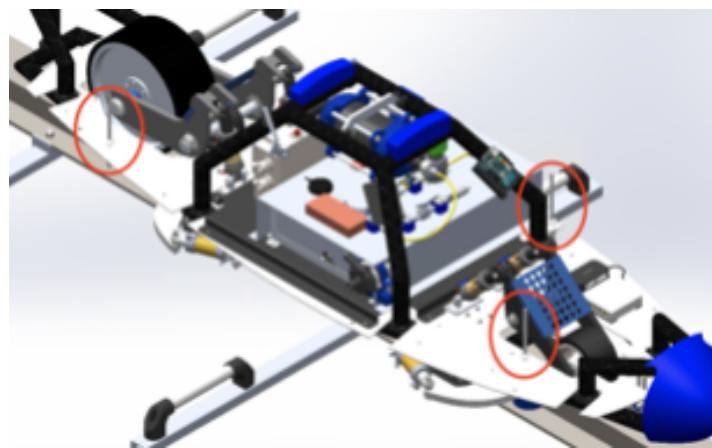


Figure E6: An example mounting render, showing the mounting locations of the three Prosense RTD0100-06-010-H sensors.

Table E7: Overview of the sensors various mounting methods, location and materials

<b>Sensor</b>	<b>Qty</b>	<b>Fabrication</b>	<b>Hardware and Installation</b>	<b>Location</b>
ProSense RTD0100-06-010-H	3	N/A	Requires a 1/2" NPT-threaded hole and CF06-25N compression fitting (see thermowell documentation).  Install (from beneath the main sheet metal component) the thermowell, then the compression fitting with sensor, then the transmitter.	Front wheel, center of the pod, and rear wheel
TMP006 Contactless IR Temperature Sensor	1	Solder header pins to breakout	Mount to PCB using 2x each standoffs and screws.	PCB

VectorNav (INS) VN200 Rugged	1	N/A	<p>SENSOR: Can mount using 2 bolts (either #4-40 standard or M3 metric, long enough for sensor, sheet metal, and nuts and/or washers/O-rings). Requires two holes ~41mm apart, large enough for said bolts (3mm).</p> <p>GPS MODULE: mounts magnetically (or via velcro, etc.)</p>	Center of the pod (or region with least vibrations)
MPU9250	1	Solder header pins to breakout	Mount to PCB using 2x each standoffs and screws.	PCB
AcuAMP DCT500-42-2 4-F	1	Cut a DIN rail to size	<p>The sensor is screwed into place through drilled holes.</p> <p>The wiring from HV will pass through a hole in the sensor. Mount on 35mm DIN rail in any position. Place wire to be measured through the aperture ensures current flow follows the label.</p>	High voltage battery housing
SPTD25-20-1 000H	1	N/A	<p>Uses a 1/4" male NPT thread only.</p> <p>Tapered for sealing</p> <p>Does not require thread sealer (DO NOT use liquid or paste thread sealer). If necessary, use a tape-style thread sealer that is compatible with the media.</p> <p>Plumbed into the brake line, probably with T-fitting. Already mated on the actual pod, however, there will need baseplate and velcro (likely) to help secure and support the sensor.</p>	Brake main reserve tank

Wenglor Photoelectric OPT2002	1	Laser cut (or, alternatively, water jet) mount from sheet metal. Bend as necessary using standard tools.	2 × M4 × 30 mm screws, 2 × M4 spring washer, 2 × washers. 2 × M4 nut  Use a mounting bracket, aim sensor to object where distance is wanted	Bottom-front of pod
Serial CAN Bus Module	1	Screw it in, connects to PCB via Grove connector, the CAN H, CAN L wires connect to the inverter CAN H and CAN L ports	Mounted to PCB using 3x each screws and standoffs	PCB
SPT25-20-200 A	2	See SPTD25-20-1000H	See SPTD25-20-1000H	Braking exhaust valves

## Sensor Safety and Testing

The main point of failure for the sensor system are miscalibrated sensors which pose a hazard of causing unexpected behaviour in control systems of the pod running on the Odroid XU4. The primary solution to prevent miscalibration are good calibration practices and frequent calibration testing. Table E8 below contains the complete detailed testing procedure for each sensor, to ensure the sensors are calibrated and working as expected. Conducting a thorough testing procedure has a direct impact on the success of the pod. For example, if the VectorNav VN-200 IMU is miscalibrated and returns the incorrect position value of where the pod is on the track, then the pod will fail to brake before the track ends. Even the smallest miscalibration could potentially end in catastrophic failure. As a result to keep the calibration practice consistent and the sensors working as expected, the principles outlined in the Handbook for the Quality Assurance of Metrological Measurements<sup>7</sup> will be employed.

The sensor system relies on various aforementioned subsystems to operate successfully. A potential risk is failure in any of the aforementioned subsystems. The biggest risk with the aforementioned subsystems comes from inconsistent electrical connections. The [low-level control system](#) is designed to handle complete power loss, complete communication loss, and complete electrical connection loss, however it is not designed to handle inconsistent electrical connections on the printed circuit boards which can lead to inaccurate data. The sensor system relies on the [electronics infrastructure system](#) to prevent this risk at the printed circuit boards level.

The additional risk with the ProSense RTD 0100-06-010-H is the mounting of the sensor. As mentioned in Table E4, two ProSense RTD 0100-06-010-H sensors are being used to measure the temperature around the front and rear wheels. The sensors must be mounted in such a way that it does not come into contact with any moving parts, like the front and rear wheels. Any contact with moving parts, like the wheels can severely damage or destroy the sensors. This risk will be mitigated with the use of the CF06-25N compression fitting and RTDTW-06-010-50N thermowell to mount the RTD 0100-06-010-H at a safe distance from the front and rear wheels. As mentioned before, the sensor system relies on the [structures system](#) to ensure the stability of the sensor mount. More details on mount details can be found in the [sensor mounting](#) section above.

Several sensors are strategically placed in specific locations of the pod in order to take the most useful measurement. The primary IMU VectorNav VN200 is positioned at the center of the pod to ensure the sensor inertial data are representative of the actual motion of the pod. The temperature sensors ProSense RTD0100-06-010-H and TMP006 are placed throughout the pod,

---

<sup>7</sup> “Calibration Procedures,” *NIST*, 31-May-2019. [Online]. Available: <https://www.nist.gov/pml/weights-and-measures/laboratory-metrology/calibration-procedures>. [Accessed: 13-Dec-2020].

at the front and back wheels as well as the center of the pod where most components are located, so the temperatures throughout the pod is always monitored. The pressure sensors SPTD25-20-1000H and SPT25-20-200A are placed at the brake reserve tank and exhaust release valves to take the most relevant pressure measurements. The gap height sensor Wenglor Photoelectric OPT2002 is positioned at the front of the pod where the change in vertical height can be measured as fast as possible.

The CAN H and CAN L wiring between the Serial CAN Bus module and the PM100 inverter poses another risk. The CAN H and CAN L wires are how the sensor system and low-level control system interfaces with the inverter. It is essential for the operation of the pod to have a reliable connection between the Serial CAN Bus module and the PM100 inverter. The electronics infrastructure system is responsible for the CAN H and CAN L wiring. An additional risk with the Serial CAN Bus Module is the difference in operating voltage between the Arduino Due microcontroller and the Serial Canbus module. The Arduino Due microcontrollers operate at 3.3 VDC and will be using the BOB-12009 Bi-Directional Logic Level Converter to interface with the Serial Canbus module which operates at 5.0 VDC. The logic level converter being used and the Serial Canbus module itself pose a risk, they must operate as documented<sup>8 9</sup>. The testing procedures outlined below Table E8 will be routinely followed to ensure the Serial CAN Bus Module and it's logic level converter operate as expected. However, in addition to testing, the electronics system will carry an additional logic level converter and the Serial Canbus module for redundancy.

The braking system is also one of the most important systems onboard. Therefore, two SPT25-20-0200A Pressure Transmitters are connected to better monitor the pressure of the exhaust release valves throughout the run. The testing procedures outlined below in Table E8 will be routinely followed to ensure the two SPT25-20-0200A Pressure Transmitters operate as expected. However, in addition to testing, the electronics system will carry two additional SPT25-20-0200A Pressure Transmitters for redundancy. Beyond that, the SPT25-20-0200A can measure up to 200 psi with a max permissible pressure of 580 psi, and it is designed for harsh conditions. All pressure sensors will be tested thoroughly in the University of Toronto Dynamic Systems Lab prior to installation. Afterward, they will be inspected regularly to ensure they are operating as expected.

At an individual sensor level, the auxiliary sensors (MPU9250 and TMP006 Contactless IR Temperature Sensor) are purely optional. The operations of the UTHT Pod are not dependent on these sensors. Any individual failure of these sensors will not put the sensor system or the pod at

---

<sup>8</sup> "BSS138", Onsemi.com, 2021. [Online]. Available: <https://www.onsemi.com/pdf/datasheet/bss138-d.pdf>. [Accessed: 14- Mar- 2021]

<sup>9</sup> "Serial CAN-BUS Module based on MCP2551 and MCP2515", Digikey, 2021. [Online]. Available: [https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/14991377\\_Web.pdf](https://media.digikey.com/pdf/Data%20Sheets/Seeed%20Technology/14991377_Web.pdf). [Accessed: 14- Mar- 2021]

risk. Of course, the data from the auxiliary sensors must be accurate for post-run analyses to yield representative conclusions. Lastly, to eliminate risks associated with energy storage, there is no type of energy storage, or any energy storage components within the sensor system.

Table E8 below, discusses the complete testing procedures to ensure the sensor system can safely monitor the state of the pod. Each sensor and microcontroller will be individually tested as per its individual tests prior to installation. After installation, the team will perform periodic inspection to the parts that pose a safety threat to the pod, and complete the testing procedures before every run of the pod.

Table E8: Complete Detailed Testing Procedures

System	Part	Testing
Control System	Arduino Due	<p><b>Test:</b> Compose Arduino test code to lower speed of PWM using divider of 1000 or more to create PWM widths of 100ms to multiple seconds. Connect PWM output pin with resistor and LED to GND.</p> <p><b>Purpose:</b> To ensure that the PWM works on each Arduino Due.</p> <p><b>Outcome:</b> Observe if the PWM pin LED flashes. If the LED flashes faster, the PWM speed is high, otherwise low. By using a stopwatch you can roughly calculate the PWM speed to conduct a quick numerical check (taking the divider into account) and after the test remove the divider.</p>
		<p><b>Test:</b> Connect each digital and analog pins used to a breadboard using male jumper wire. Check voltage levels using a multimeter once test messages are transmitted by Arduino. Also, connect LEDs with resistors to GND for each pin on the breadboard.</p> <p><b>Purpose:</b> To verify that Arduino Due supports IC2 and SPI communication, and the digital and analog pins operate as expected.</p> <p><b>Outcome:</b> Multimeter should display voltage levels once messages are being transmitted. Pin LEDs will flash indicating messages are successfully sending.</p>
Braking System	SPTD25	<p><b>Test:</b> Basic sensor pressure measurements without any additional air pressure. (A)</p> <p><b>Purpose:</b> Pressure sensor tests before and after its installation on the pod to verify that it acts as expected.</p> <p><b>Outcome:</b> With no additional air pressure, the sensor should read 0 psi.</p>

		<p><b>Test:</b> Sensor pressure test while increasing pressure by 10 psi increments until its maximum limit and/or 50% over the maximum pressure limit. This test will also be used while testing the braking system.</p> <p><b>Purpose:</b> To ensure that pressure sensor applies pressure properly.</p> <p><b>Outcome:</b> Arduino Due will read pressure sensor readings as psi value increases to maximum level and/or 50% over.</p>
Braking System	Arduino Due powering the solenoid valve	<p><b>Test:</b> Engage/disengage the brake solenoid valve with the braking system at atmospheric pressure.</p> <p><b>Purpose:</b> To check if brake solenoid valve pressure increases/decreases as expected.</p> <p><b>Outcome:</b> As the solenoid valve engages, pressure is increased, verify with pressure sensor readings. Same with disengaging the valve, pressure sensor reading will decrease.</p>
Propulsion System	CANBUS module	<p><b>Test:</b> Two Serial CAN Bus modules will each be connected to separate Arduino Dues to test the communication channels for sending/receiving data, and applying the CAN mask/filter.</p> <p><b>Purpose:</b> To first test the Serial CAN Bus Module separately from the PM100DX Inverter and pod. Basic send/recv functions must operate.</p> <p><b>Outcome:</b> Both Arduino Due power LEDs flash ON, followed by their Serial CAN Bus Module power LED turning ON. Once the Arduino CAN Bus test code is uploaded to the transmitter and receiver, the TX/RX LEDs on both Serial CAN Bus Modules will begin flashing, indicating that CAN frames are transmitting. Finally, open Arduino Due Serial Console of the receiver, and check expected print statements:</p> <ul style="list-style-type: none"> <li>• 9600 bps Baud Rate OK/FAIL message</li> <li>• CAN Mask/Filter OK/FAIL message</li> <li>• Continuously printed out CAN messages: “Received CAN message from ID:X”, followed by hexadecimal frame bits.</li> </ul>
		<p><b>Test:</b> Connect a Serial CAN Bus Module to PM100DX inverter. Continuously receive CAN messages from the inverter, and send test CAN messages to the inverter to check if its torque values change (accelerate and decelerate).</p> <p><b>Purpose:</b> To ensure Serial CAN Bus Module can send/recv CAN messages to the PM100DX Inverter and control it accordingly.</p>

		<p><b>Outcome:</b> PM100DX is powered on, followed by the Arduino Due and Serial CAN Bus Module. Their respective power LEDs turn ON, and the TX/RX LED of the Serial CAN Bus Module will begin flashing. This indicates CAN messages are being transmitted. Next, open Arduino Due Serial Console of the receiver, and check expected print statements:</p> <ul style="list-style-type: none"> <li>• 9600 bps Baud Rate OK/FAIL message</li> <li>• CAN Mask/Filter OK/FAIL message</li> <li>• Continuously printed out CAN messages: “Received CAN message from ID:X”, followed by hexadecimal frame bits.</li> <li>• Next, send a CAN message to PM100 to accelerate, observe if torque changes. Same with decelerating, check if PM100 torque changes by polling the received CAN messages from the inverter. They specify the value readings of the inverter.</li> </ul>
Propulsion System	ProSense RTD 0100-06-01 0-H	<p><b>Test:</b> The RTD will have the accompanying thermowell threaded over the compression fitting on the probe, and the temperature transmitter connected to its M12 connection. An M12 cable will send signals from the transmitter to a keyed, 4-pin GROVE connector on a PCB. This PCB will send signals to an Arduino Due that will be connected to a main computer where readings can be displayed in the Arduino serial monitor. 24VDC will be provided via the PCB and GROVE connector to the transmitter as well. Two of the connections coming from the sensor will not be used (pin 3 and pin 4).</p> <p><b>Purpose:</b> To ensure the ProSense RTD sensor successfully operates.</p> <p><b>Outcome:</b> Open Arduino Due Serial Console monitor to observe printed out temperature readings. These readings can be verified against the current output characteristics and min/max temperatures provided by ProSense’s documentation.</p>
Internal Pod environment	TMP006 Contactless IR Temperature Sensor	<p><b>Test:</b> The temperature sensors will be tested separately in a controlled environment where the temperatures are known and the readings will be compared to the real thermometer values of the room. Connect the TMP006 to the Arduino Due and upload the temperature reading test code.</p> <p><b>Purpose:</b> To check if the temperature sensors are properly calibrated.</p>

		<b>Outcome:</b> Printed temperature values will be visible on the Arduino Due Serial Monitor. Verify these values with the current room temperature.
Propulsion System	IMU - VectorNav (INS) VN-200 Rugged (x1)	<p><b>Test:</b> The IMU will be mounted securely to a “test bench” surface (any fixed, rigid surface with appropriate screw holes), and the GPS antenna and connector will be attached. The other end of the connector will be connected to 3.3V power, ground, and an Arduino Due via breadboard (as shown in the VN-200 pinout and hookup documentation). The Arduino Due will be connected to a computer via USB, and the Arduino Serial Monitor will be opened. The VN-200 testing software will be verified, compiled, and uploaded onto the Arduino Due.</p> <p><b>Purpose:</b> To ensure IMU is connected correctly to the Arduino Due and that it reads data.</p> <p><b>Outcome:</b> The Arduino Due Serial Monitor will begin by displaying POST messages, and will then move into reading and displaying the data from the VN-200.</p>
Energy System	AcuAMP DCT500-4 2-24-F	<p><b>Test:</b> Connect the 24V DC power source required by the sensor. A resistor will be used in a series of the sensor's output to calculate the current received by the analog input accurately. Connect the Arduino Due to a computer and the test code to the sensor. The computer will receive the sensor's output once you pass the DC power source through the sensor.</p> <p><b>Purpose:</b> To check if the AcuAMP powers on, and that it is properly connected to its Arduino Due. Also to ensure correct sensor readings are received by the Arduino Due.</p> <p><b>Outcome:</b> Put a few DC power sources with different currents through the sensor to check if the sensor receives the current value. Make sure the output load is no more than <math>500\Omega</math>. Check if the sensor is properly mounted on the 35mm DIN rail.</p> <p>When 24V DC is supplied to the AcuAMP, an output of 4-20 mA current will be read directly from the Arduino's analog pin. The sensor should be outputting the correct reading when connected to power and the battery pack of the pod.</p>

		<p><b>Test:</b> Before turning on the sensor, check the screw connectors for any open-wires and loose connections.</p> <p><b>Purpose:</b> To check if the AcuAMP is properly secured to its pod mount.</p> <p><b>Outcome:</b> When conducting this visual check, any and all loose screw connectors are adjusted to be appropriately tightened. Open wires are fastened.</p>
Stability System	Wenglor Photoelectr ic OPT2002	<p><b>Test:</b> Attach the OPT2002 securely to the OPT2038 mounting bracket which is mounted securely to a “test bench” surface. The M12 connector is attached to the OPT2002, and the other end will be connected to 24V power, GND, and the Arduino Due all via breadboard. The Arduino Due will be connected to a computer via USB.</p> <p>The OPT2002 has an analog output, so data can be read directly by the Arduino Due’s onboard ADC. The analog output of the sensor is 10V, however, and the Arduino Due only supports up to 3.3V, so a voltage divider is used.</p> <p><b>Purpose:</b> The OPT2002’s values will be checked from the Arduino Due for proper measurement.. Testing can be performed with a ruler and a moving object. POST/debug information can be taken from the various onboard LEDs.</p> <p><b>Outcome:</b> The OPT2002 testing software will be verified, compiled, and uploaded onto the Arduino Due. The Serial Monitor will begin displaying POST messages, and then move into reading and displaying the data from the OPT2002.</p>
Propulsion System	Arduino Due/CAN Bus Module  BOB-1200 9  Bi-Directional Logic Level Converter	<p><b>Test:</b> 5V will be hooked up to the side of the logic level converter labelled HV, 3.3V will be connected to the side labelled LV, and GND will be wired on both sides. To test that the voltage conversion works properly, connect a 10k ohm pull up resistor from LV1 to LV.</p> <p><b>Purpose:</b> A logic level converter is used to allow the Arduino Due (3.3V operating) to communicate with the CAN Bus module (5V operating), so this converter will be tested separately from the system on a breadboard to ensure proper functionality.</p> <p><b>Outcome:</b> The LV1 pin should measure 3.3V and the HV1 pin 5V. Then connect the 10k pull up resistor to HV1 instead,</p>

	<p>HV1 should measure 5V and LV1 3.3V. The resistors will be removed after the test.</p>
	<p><b>Test:</b> An Arduino Due will have its TX pin connected to LV1 and RX pin connected to LV2. The HV1 and HV2 pins will be shorted with a jumper wire.</p> <p><b>Purpose:</b> To test the bi-directional capability.</p> <p><b>Outcome:</b> The Arduino Serial Monitor will be opened so that when a message is sent through the TX pin, it should be received back in the RX pin, then the same message should be displayed in the serial monitor once it is echoed back.</p> <p>(TX&gt;LV1&gt;HV1&gt;HV2&gt;LV2&gt;RX)</p>

## Low-Level Control System

The low-level control system implements the pod's motion control, via the [brake subsystem](#) and the [propulsion subsystem](#). The low-level control system is responsible for controlling the brake solenoid valve and the motor inverter based on the commands from the [high level control system](#) on the master computer (Odroid XU4). Refer to the [high level control systems](#) section for more details about the decision making commands on the Odroid XU4. These commands are implemented using Arduino Due #3, whose wiring diagram is included below in Figure E9, the wiring schematic for Arduino Due #3 can be found in [Appendix E3.3](#).

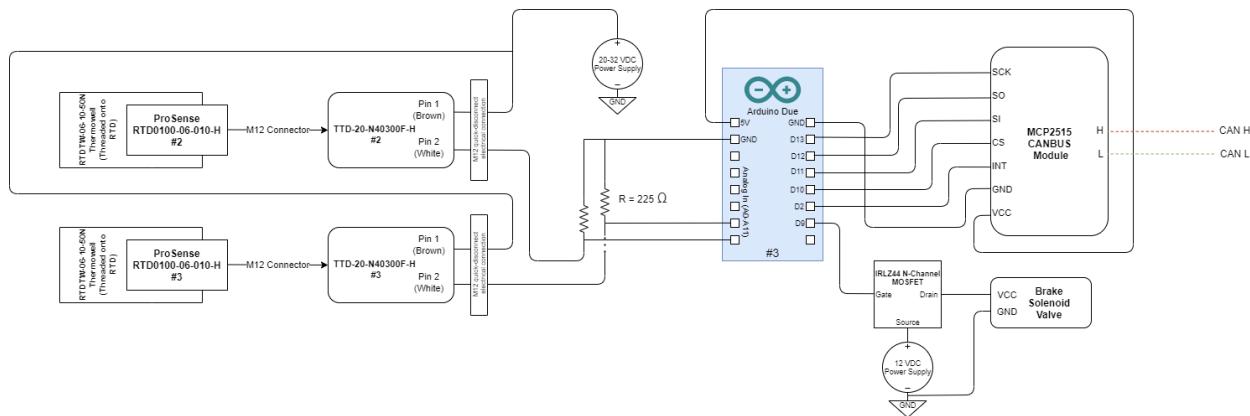


Figure E9: Wiring diagram for Arduino Due #3

Arduino Due #3 is used as a part of both the [sensor system](#) and the low level control system, as shown in the wiring diagram above in Figure E9. The Serial CAN Bus Module controls both the motor inverter and provides temperature, voltage, current, torque information and fault code data about the propulsion motor and the motor inverter. The low level controls system relies on the [electronics infrastructure system](#) to implement the wiring diagram on a printed circuit board where the electronic components will also be mounted. More details on wiring of the low-level control system are denoted in the [Printed Circuit Board # 3](#) section.

The electrical power required to power the low-level control system comes from the energy systems' [Low Voltage Battery Module \(LVBM\)](#). Consistent power delivery is critical to the control systems operation, otherwise the control system will passively trigger emergency stop. This passive emergency stop trigger upon power loss is further discussed in the [braking system](#) section further below.

The [software communication system](#) will be utilized to communicate between Arduino Due #3 and the Odroid XU4. The Odroid XU4 and the Arduino Due #3 will constantly run a handshake protocol to ensure communication is maintained between the two. In the event the handshake fails to complete, Arduino Due #3 will go into emergency mode, shut down the motor, and engage the brakes. More details on the [software communication system](#), and its underlying

infrastructure are denoted in the [software communication system](#) section. The subsequent sections provide more in-depth explanations of the brake and propulsion motor control. Their respective testing and safety procedures will also be detailed.

### Braking System

The braking system functions as both the pod's main and emergency brakes. The electronics system controls the braking system via the braking system solenoid valve. The solenoid valve controls the pressure, which when released to the braking units, engages the brakes. The solenoid valve requires power to remain closed, operating in an active low configuration. As a result, in the event of power loss the brakes will be automatically activated. The wiring diagram above in Figure E9 shows how using the STP80NF55-08AG N-Channel MOSFET the solenoid valves will be connecting to Arduino Due #3. The Arduino Due drives the gate of the MOSFET with its 3.3 VDC signal to control the power input from the Low Voltage Battery Module on the 8W 12 VDC solenoid valve. Once the pod is ready to run, Arduino Due #3 will direct power to the solenoid valve to ensure the brake pressure tank retains its pressure and the brakes are disengaged. In the event of a brake command or communication failure (Odroid XU4 handshake failure), Arduino Due #3 will disable the power to the solenoid valve which will cause it to open and the air pressure will drop in the braking units which will in turn engage the brakes.

### Braking Safety and Testing

The main risks of failure for the low-level braking control system are the STP80NF55-08AG N-Channel MOSFET, brake solenoid valve, or Arduino Due failing. Any physical damage to the STP80NF55-08AG N-Channel MOSFET poses a safety risk. The maximum voltage/current rating must never be exceeded to avoid burning, otherwise the MOSFET will not operate appropriately and as a result the solenoid valve will not be controlled properly. This risk will be mitigated by thoroughly testing the STP80NF55-08AG N-Channel MOSFET before every run. The brake solenoid valve failing to release air will prevent the brakes from engaging. The low-level controls system relies on the braking system to prevent this risk, and for the braking system to mechanically operate as expected. The risk of the Arduino Due failing will be mitigated by following the complete testing procedure before everytime the pod runs, discussed Table E6 in the [sensor safety and testing section](#). The TP80NF55-08AG N-Channel MOSFET, brake solenoid valve, and system integration tests are in Table E10 below. There is no type of energy storage, or any energy storage components within the low-level brake control system system.

Additional risks from power loss of the [Low Voltage Battery Module](#) are mitigated with the design of the braking system. Since the solenoid valve operates in an *active low* configuration, once power is lost, the Arduino Due #3 will disable the power to the solenoid valve (via the MOSFET), the solenoid valve will open and the air pressure will drop in the braking units,

engaging the brakes in an emergency stop. The same situation will transpire, if the electrical connections between the [Low Voltage Battery Module](#), Arduino Due #3, the STP80NF55-08AG N-Channel MOSFET, and the brake solenoid valve disconnect. The low-level control system relies on the [low voltage battery module](#) and the [electronics infrastructure system](#) to successfully complete a run, however with the *active low* design of the braking system in the event of power loss, communication loss, or electrical connection issues the pod will engage the brakes and come to a stop, safely ending the run.

Table E10: Braking System Tests

Test Description	Purpose of Test	Expected Outcomes
STP80NF55-08AG N-Channel MOSFET multimeter tests. First, short the gate and drain pins, and then touch the red multimeter probe to the drain pin, the black probe to the source pin.	To confirm that the MOSFET is not damaged and properly operating.	Check the source and drain pins with the multimeter, should read as “open circuit”.
Basic brake engage/disengage testing using the Arduino Due. It will apply/remove voltage to the braking solenoid valve.	To ensure that the MOSFET is properly connected and activated by the Arduino Due, and that the MOSFET controls the solenoid valve as expected..	As the Arduino Due engages the MOSFET to supply voltage to the solenoid valve, the valve stays disengaged (brakes are off). If MOSFET supplies 0V signal to the valve, the brakes engage/turn on.
Odroid/Arduino Due communication handshake protocol tests.	To check if the Odroid and Arduino Due are communicating to each other with no disruptions, and to test scenarios where communication is down, enabling emergency brake.	When Odroid and Arduino Due are powered on, Odroid successfully sends commands to Arduino to activate/de-activate MOSFET output. The Odroid will receive an acknowledgement frame that it was received successfully by the Arduino. Next, activate a test where the handshake

		protocol will break, observe that the emergency brakes activate.
Turn on LV power system, followed by the Arduino Due and then the Odroid XU4.	To check if these devices turn on correctly.	Observe if the Odroid and Arduino Due power LED is ON.

### Propulsion System

The EMRAX 188 Motor and its PM100DX Inverter are the main components of the pod's Propulsion System. The low-level control subsystem is responsible for operating, accelerating and stopping the EMRAX motor. The Odroid XU4 communicates to Arduino Due #3 via UART which is found on PCB #3, and it then sends instructions to the PM100DX inverter through a UART to CAN Bus converter module called the Serial CAN Bus Module. The design of using CAN Bus communication with the PM100DX inverter ensures fast and reliable communication using the CAN H and CAN L wires that transport CAN messages, and it is a standard in the automotive industry. [Appendix E4](#) contains more details on how the Serial CAN Bus Module interfaces with the PM100DX, including its safety risks, considerations and testing procedures. [Appendix E5](#) contains the full startup sequence for the PM100DX Inverter. Following the startup sequence in [Appendix E5](#) and the connection interface in [Appendix E4](#), the motor will reliably accelerate and decelerate and provide internal sensor readings e.g. operating temperature, motor frequency, current and the inverter's operational state.

### Propulsion Safety and Tests

The low-level propulsion control system integrates and requires many other systems in order to operate. The Energy Systems provides power to all the modules (Odroid, Arduino, PCB, Serial CAN Bus Module) and the EMRAX 188 Motor with its PM100DX inverter. The low-level propulsion control system doesn't include any energy storage components. The [Electronics Infrastructure System](#) provides the PCB layout for Arduino Due #3 and all electrical wiring connections between modules. The pod's wheels provided by the [Propulsion Team](#) must successfully operate in order for the pod to mechanically accelerate on the track, and the [Stability System](#) ensures stable acceleration and motion. Also the Structures System provides all the mounts for the motor, inverter, and PCBs. These are the many systems that must function correctly for the low-level propulsion control system to successfully accelerate and stop the pod. This highlights the important integration of software/electrical systems to the physical/mechanical systems of the pod. Along with the [Braking System](#), the [Propulsion System](#) control circuitry is neatly organized on its own PCB aside from the rest of the pod's electronics, making testing and troubleshooting easy and efficient.

The control system's effect on propulsion is to control the inverter on, pre-charge, discharge, and off state, and the rotational frequency of the motor. The delivery of the high voltage power to the propulsion system and its conversion from DC to 3-Phase AC power is handled by Energy Systems. This is discussed in the [Motor Inverter System](#) section. The modulation & control of this power is handled by the control systems.

Next, the wiring on the PCB can be faulty. Any damage to the PCB will be detrimental to the pod's performance, therefore multiple versions and models of the PCB are going to be manufactured and tested. This provides the electronics team room for improving the PCB and wiring of the systems throughout the testing period.

The Propulsion System will be tested to make sure the Odroid XU4 can use its software system to control and poll the status of the EMRAX 188 Motor through the PM100DX Inverter. Firstly a simple test code for receiving CAN Bus messages from the PM100DX in real time will be uploaded. Current, voltage, temperature and fault code information of the inverter should be visible to the Odroid once the inverter is powered on. The second test code will include sending CAN Bus messages to the inverter, setting specific torque speeds, and the EMRAX motor will be observed if it reacts. Other system tests include performing visual checks confirming that all modules are ON by checking the power LEDs of the Odroid, Arduino Due and Serial CAN Bus Module. These propulsion system tests and more are highlighted in Table E11 below. These can be performed before the pod's complete assembly and also when it is built and mounted on a test track.

Finally, in the emergency event of the pod losing power from the LV Battery Module, the [HV battery software](#) will disable power to the PM100DX Inverter, thus shutting down the ERMAX motor. When there is any communication loss with the PM100DX inverter or the Odroid XU4, the Odroid will either get an Emergency Stop message from the Arduino Due or declare an Emergency Stop itself due to failure of completing the communication handshake protocol with the Arduino Due. The Odroid XU4 will then send a message to the [HV battery software](#) to disable power and shut down the ERMAX motor. The [HV Battery Software](#) discusses how this system operates from an energy systems perspective.

Table E11: Propulsion System Tests

Test Description	Purpose of Test	Expected Outcomes
Check if the Odroid XU4 is communicating with the Arduino Due by sending test frames.	To confirm that ZCM communication between the devices is running.	ZCM handshake protocol successfully passes, frame acknowledgement messages are received by the Odroid.
Identify if Serial CAN Bus Module is operating.	To check if Serial CAN Bus Module is ON and sending/receiving CAN messages from the PM100 DX Inverter to the Arduino Due.	Serial CAN Bus Module power LED is ON, RX/TX LEDs are flashing. Also CAN messages will be continuously received by the Odroid as soon as the PM100 is powered on.
Check the PM100 Inverter's connection to the ERMAX Motor and Serial CAN Bus Module. Send CAN frames from the Odroid to the Inverter setting the motors torque.	To ensure that the PM100 is properly connected in the Propulsion System, and that the Odroid commands control the ERMAX motor.	Firstly, CAN bus frames transmitted by the Inverter will be received by the Arduino, which will be communicated to the Odroid. Next, a series of test CAN frames will be sent to the Inverter setting the motors torque. Visually observe the motor spin, accelerate, and stop properly.
Conduct a visual check of the Propulsion System on the pod before testing the pod on the track.	To confirm that all electrical wiring and PCB connections are secure and connected properly. This is mandatory to make sure there is no visual damage to the system before pod launch.	There are no damaged wire/pin connections.

## Infrastructure System

The function of the infrastructure system is to create a uniform connection framework that makes sure the connections between sensors and the microcontrollers are implemented in a simple and robust way that constantly provides the pod's control systems, as well as its operators, mission critical diagnostics and information. Therefore it was decided to design custom Printed Circuit Boards (PCB) to organize these circuit paths and connections. They create an abstraction of all the wiring between the Arduino and sensors. The wiring schematics that show how the connections are done can be found in [Appendix E3](#). The reliability of these PCBs is of major importance as they contain the connections that provide the controls of the braking system and the propulsion system. Due to this role we take special safety considerations in their designs which are discussed more in detail on the rest of this section.

The PCBs will be used to integrate power delivery and to receive readings from multiple sensors to one microcontroller which is connected to the pod's master computer (Odroid XU4). As the sensors are located throughout the length of the pod, it is important to have a central location where power delivery and connections to an Arduino Due microcontroller can be managed. The PCBs themselves are about the size of an Arduino Due, and will be connected to the Due in a shield configuration (PCB placed on top of Arduino) using header pins with a pitch of 2.54mm on the PCB, similar to the example in Figure E12. To ensure the stacked configuration is stable, we will be using screw-secured standoffs which will secure the PCBs on the Arduinos and make the stack more rigid. This will also lower the risk of the components physically disconnecting from vibrations and jerks, during the run.

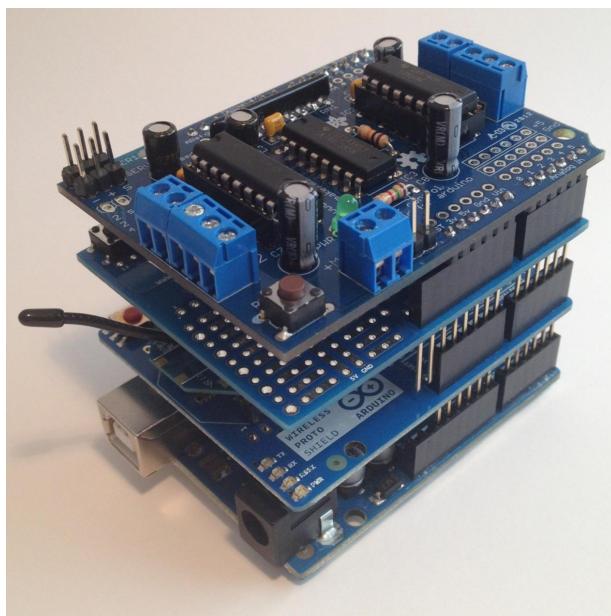


Figure E12: Stacked arduino shields

The communication infrastructure will contain the W5500 Ethernet Shield as the interface node to the Arduino, which will allow it to communicate to the main computer using SPI and an ethernet cable connection. Refer to [Appendix E2](#), for a more detailed breakdown of the ethernet shield. This shield will be using the same 2.54mm pitch header pins to interface to the Arduino and its mapping and connection will be implemented on each Arduino's respective PCB.

We will be using keyed, spring-locked GROVE connectors<sup>10</sup> (seen in Figure E13) with a variety of pin quantities to interface with almost all the sensors. These connectors were selected in order to create a physical constraint for a unique orientation to connect the sensors. This eliminates the possibility of a pin mis-alignment error when interfacing with the PCB.

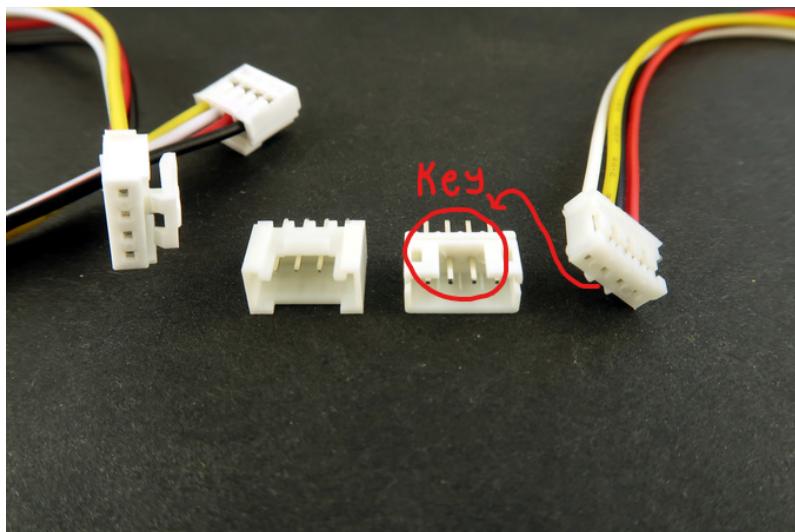


Figure E13: Example of a GROVE Connector

The PCBs also have a role as part of the system that delivers the power to the various components present in the electronics system. The necessary voltage required to power the components ranges from 3.3V to a max of 36V. [The energy systems' low voltage battery module](#) on the pod supplies the required power for these inputs, which will be connected to each PCB. The sensors and low-level control system components are powered by different power nets on each PCB, depending on what voltage they require. This can be seen of Figure E14 where J4, J3 and J6 represent the power input ports. This massively simplifies sending power to multiple sensors that need similar voltage levels. Instead of having unique power connections for each component, using PCBs gives us the ability to have singular voltage levels on them and then connecting each component to its respective voltage level on the PCB, rather than drawing multiple connections from the main power system's unit.

---

<sup>10</sup> "Grove Female Header", Seeedstudio.com, 2021. [Online]. Available: <https://www.seeedstudio.com/Grove-Universal-4-pin-connector.html>. [Accessed: 14- Mar- 2021]

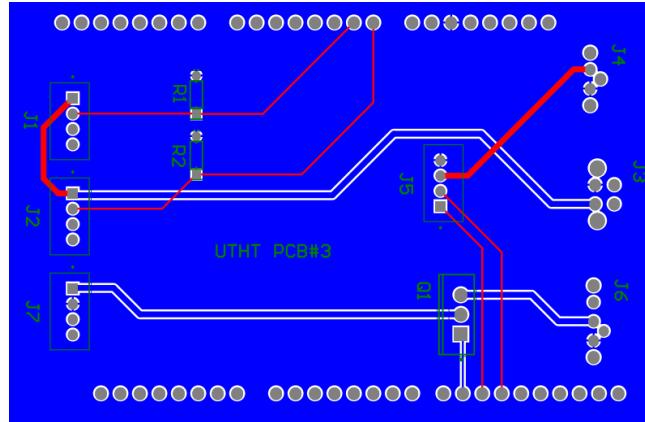


Figure E14: Layout of PCB#3 showing the physical connection traces.

The suite of sensors requires the ability to process several different forms of data including digital signals, analog signals, and serial communication data (I2C & TTL). Our Arduinos are limited by specific pins necessary for these serial communication protocols. Having three Arduinos allows us to take in sensor data from every part of the pod concurrently. The raw data coming from the sensors will be processed into useful information before being sent to the main computer, the Odroid XU4. For example, the 4-20mA analog output of a temperature sensor needs to be converted to a floating point number in degrees Celsius before the Odroid XU4 can use it. The Arduinos will be able to get the raw data and process it in order to transmit meaningful data to the main computer. This distribution of the processing load increases the efficiency of the system by lowering the load on the Odroid XU4. A good refresh to how the system comes together would be to have another look at Figure E1 from the [general overview](#) below listed as Figure E15.

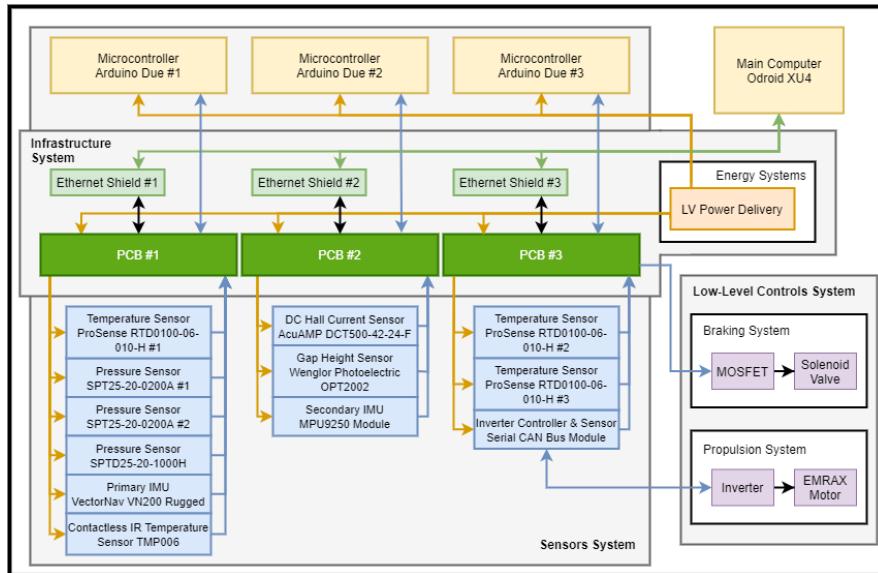


Figure E15: A high-level block diagram of the major electronics system components

## Printed Circuit Board #1

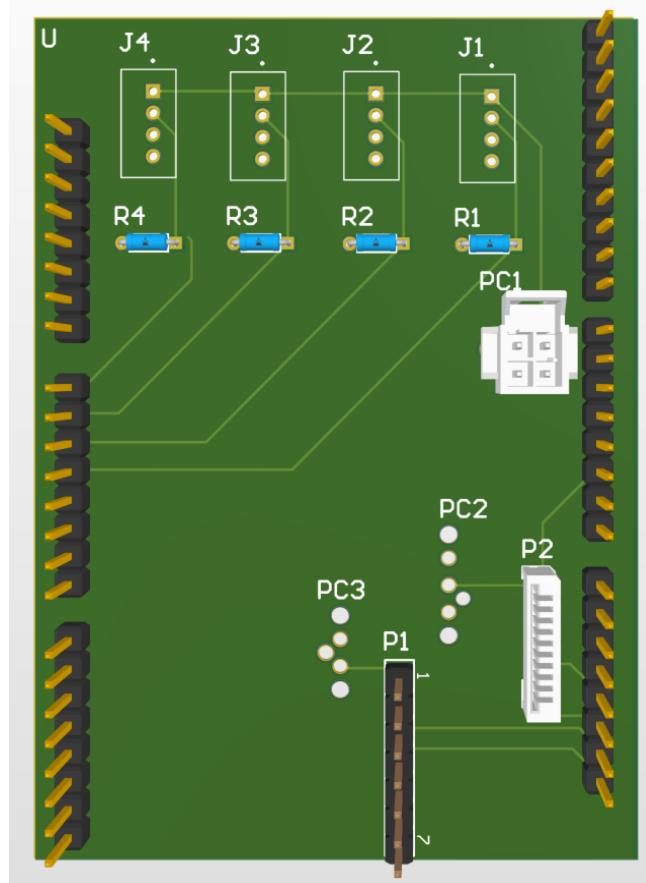


Figure E16: 3D view of PCB#1

PCB#1 is shown in a 3D view in Figure E16 above. This figure shows the vertical pins and through holes which perfectly align with the Arduino Due along with the components that are placed inside to save space. The through holes for the power supply are also displayed as PC1, PC2, and PC3 which respectively deliver 24V, 12V, and 5V. Those power supplies are used to power three different types of sensors on the pod. PC1 provides 24V of power to one SPTD25-20-1000H and two SPT25 200A pressure sensors, J1 to J3, and the Pro-Sense RTD0100-06-010-H temperature sensor, J4. PC2 provides 3.3V to the VN-200 IMU on our pod, connected using a 10 pin connector labeled P2. Finally, PC3 provides 5V to the TMP006, the contactless IR temperature sensor, which will be connected using a 7 pin connector labelled P1. Along with that, there are 4 through hole resistors for J1 to J4 which will include  $220\ \Omega$  resistors as seen on the wiring layout on Figure E16 above.

## Printed Circuit Board #2

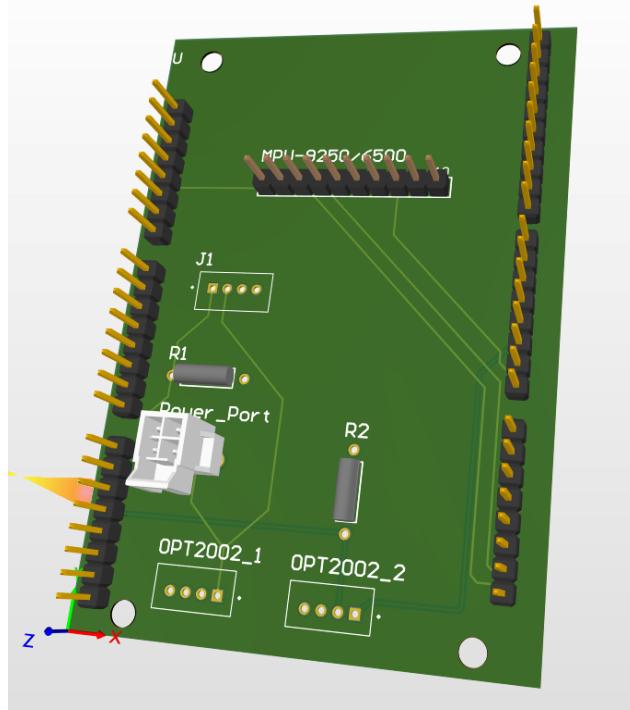


Figure E17: This is a 3D view of PCB#2

PCB#2 is shown in a 3D view in Figure E17 above. The power port is a 4 position, through hole connector header, named Power\_Port in the figure above, and it will provide a 24V power supply from the [Low Voltage Battery Module](#) in order to power the OPT2002 Wenglor Photoelectric sensor and the AcuAmp DCT500-42-24-F DC current sensor. The DC current sensor interface to the board is a 4-pin GROVE connector. Pin 2 in the GROVE connector will receive the 24V supply, while Pin 1 will provide the output of the sensor to Analog input pin 10 on the Arduino Due. Pin 1 outputs a current between 4 mA to 20 mA. In order to map it to a voltage input for the Arduino Due we use a  $220 \Omega$  through hole resistor. The Photoelectric sensor uses 2 GROVE connectors in order to interface to the board, for more details on how the pin mapping works refer to the schematic in Figure E18 below. The auxiliary MPU9250 IMU on board is provided to us as a breakout board and it connects to the PCB board using a vertical pin header with a 2.54mm pitch similar to the one used to connect to the Arduino Due.

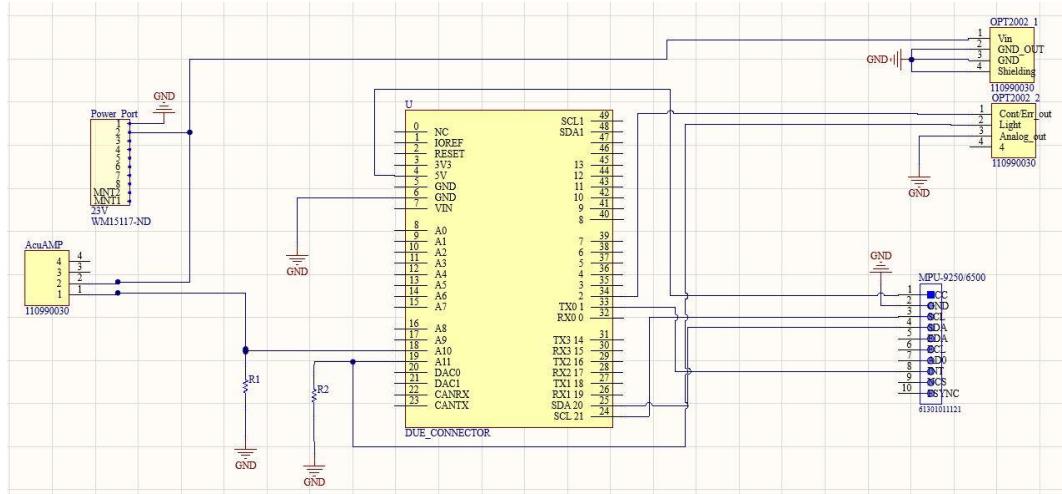


Figure E18: This is the schematic view of PCB#2

### Printed Circuit Board #3

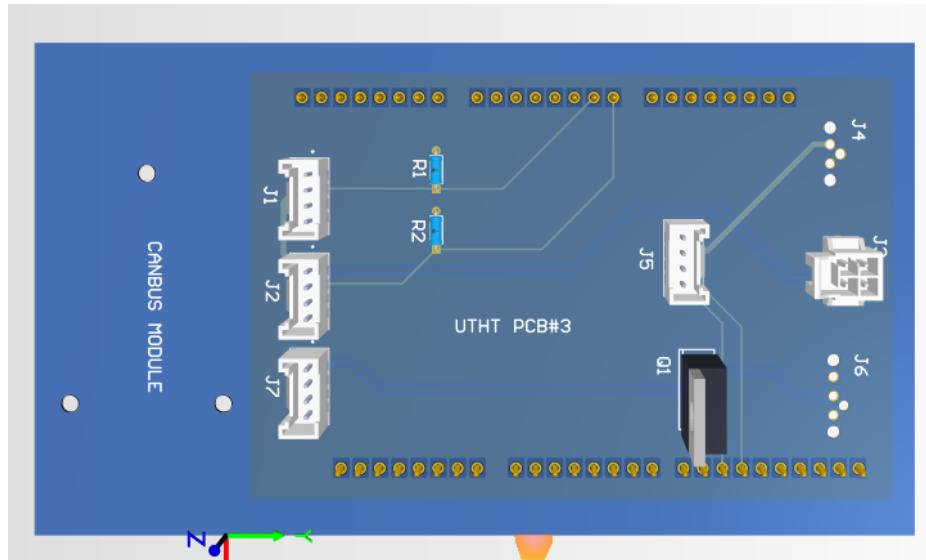


Figure E19: 3D View of PCB#3

PCB#3 is shown in Figure E19 above. Along with its connections for the Arduino, the PCB has several other connectors across its top surface, 3 of which being power connectors. Connectors J4, J6, and J3 are 2,3, and 4 pin molex connectors that provide 5V, 12V, and 24V respectively. The rest of the connectors on the board are 4 pin GROVE connectors which are used to send power to the sensors and receive their outputs for the Arduino. Lastly, the board will have a through hole MOSFET and 2 through hole resistors. The MOSFET is used to control the brake-solenoid valve from the [low level control system](#). The resistors are required to read the

current outputs of the RTD0100-06-010-H Temperature sensor into the Arduino Due microcontrollers.

### Safety and Testing

To successfully operate, all sensors must be connected via their GROVE connector, and the necessary voltage levels must be connected as well via their respective power ports as seen in the 3D PCB views in the figures above. All connectors on the board are keyed, meaning that they can only be plugged in one correct way. Sensor wires will be soldered to GROVE connector cables so that they can be connected. Lastly, the Arduino must be connected to power via its main power connector. For the system to run successfully, the [low voltage battery module](#) must be operating correctly to provide 5V, 12V, and 24V to the sensors.

All of the PCBs were designed using classic PCB design guidelines<sup>11</sup>. This helps prevent poorly-made connections and exposing bare wire. The system is secured using screw-in standoffs and keyed connectors. This means that the system can be secured easily as well as disassembled and removed from the pod easily. Safety risks on the PCBs arise from manufacturing problems and related systems that use the PCBs to operate. Due to manufacturing problems the PCBs can have connection issues in their traces. A way to mitigate this issue is to have a few spare PCBs, in our case it will be 4 spare PCBs. There are two ways to detect faulty paths on the PCBs. The first one is to use a simple circuit consisting of a battery, a resistor and an LED in series and using two wires to close this circuit. The wires can be used as probes and if they are connected to two ends of a correct path on the PCB the LED will light up, if it doesn't light up there is an issue with the path. The second method is using an Arduino. The Arduino input and output pins are connected to the ends of the paths on the PCB and the Arduino will send a signal through the output pin and expect to receive the same signal on the input pin. If any of the signals is not detected we have a faulty PCB and we replace it with a spare one that works. Another factor we have considered is the current, which won't be very high, as the current flowing in the PCB ranges from 0-250 mA, and the board itself is not crowded with circuit elements, as there will be 13 components for a board of size 60 mm x 82 mm, therefore there is very low risk of it heating up and catching fire. A more significant safety risk is if something associated with this PCB was to fail. This could range from a power source to a sensor. The Arduino will not be affected as long as it has a consistent power source. If the power source was to fail or not deliver the proper power then a sensor could break or report false measurements. There are no energy storage components in this system, so there is no risk with energy dissipation on the circuits before connection. Since the connectors are all the same for the sensors but some of the sensors have different operation voltage levels we will include protective fuses so that in a case of a misconnection the circuitry of the sensor is protected.

---

<sup>11</sup> C. F. Coombs and H. T. Holden, Printed circuits handbook. New York: McGraw-Hill Education, 2016.

Another risk we have to account for is that of poor electrical connections. This risk exists between the printed circuit boards and the following sensors: SPTD25-20-1000H Pressure Sensor, SPT25-20-200A Pressure Sensor, ProSense RTD0100-06-010-H, VectorNav VN-200 Rugged, Wenglor Photoelectric OPT2002, and AcuAMP DCT500-42-24-F. Each of those sensors interface with the PCBs using grove connectors. For all of the sensors, except the AcuAMP DCT500-42-24-F, the grove connector male header pins were soldered onto the rat-tail splice end of their M12 connector. A GROVE cable is then used to connect the male header on the sensor side and the male header on the PCB. Because of this method a risk of improper soldering exists. This can be tested and mitigated by using a multimeter to check if current is passing through or connecting an arduino directly on both ends, sending and receiving a signal.

The poor electrical connection risk with the DC Hall Current AcuAMP DCT500-42-24-F sensor comes from the screw connectors that are used for power and analog output. According to the installation manual, the screw connectors must be secure with a specific torque to the screws for the best accuracy. This sensor is used to monitor the current output from the batteries, and it is therefore connected to the controlled system. If the screw connectors corrupt the sensor's analog output and show the batteries are operating abnormally, it can potentially trigger the brake or other emergency actions. The Serial CAN Bus Module also has the same risk for its screw connectors. The Serial CAN Bus Module uses screw connectors for CAN H and CAN L to connect with the inverter. If the connections between them are corroded, it might have serious effects on the propulsion system. The electrical team will strictly follow the provided installation guideline during installation, complete with periodic checkups afterward to ensure their safe operation.

Table E20: Performance tests for the components and connectors in the electronics system

Component	Test	Target
PCBs	<p><u>Path Test</u></p> <p>We use a small battery in series with a resistor and an LED light. The circuit is open to a node and two wires are used to probe to nodes on the PCB. If the path in the PCB is not faulty then the LED will light up.</p> <p>Another method we will be using is to connect the nodes on a PCB to the input and the output pins on an Arduino and perform an automated test where the Arduino sends a specific signal from one</p>	<ul style="list-style-type: none"> <li>• We expect that any probing of two nodes connected by a path in the PCB should light up the LED.</li> <li>• We expect the input value of the Arduino to be the same as the value outputted.</li> </ul>

	<p>end of the path and receives it on the other end.</p>	
	<p><u>Temperature Test</u></p> <p>For each PCB we connect all the components and let the system run nonstop for 10 hours. During this time we monitor the temperature of the system with an IR thermometer.</p>	<ul style="list-style-type: none"> <li>• The temperature of the system should be below 50 degrees celsius at all times.</li> <li>• Pressure reading should be <math>200 \text{ psi} \pm 5\text{psi}</math> at all times.</li> <li>• The distance reading should be <math>30 \text{ cm} \pm 1\text{cm}</math> at all times.</li> <li>• The current reading should be <math>1.5 \text{ A} \pm 0.05 \text{ A}</math> at all times.</li> </ul>
	<p><u>Calibration Test</u></p> <p>Each sensor is given a preset physical input value to measure. The pressure sensors are hooked up to tanks pressurized to 200 psi. The Photoelectric sensor is directed to a board set up at a distance of 30 cm. The DC Hall sensor is connected to a wire carrying a current of 1.5 A.</p> <p>The components are connected to the PCB and the PCB to the Arduino. The whole system is left to run for 10 hours. The data coming from the sensor will be logged and analyzed.</p>	
Connectors	<p>For each connector, a 5, 12 or 24 volt of power will be supplied to the female end and measured at the male end.</p>	<ul style="list-style-type: none"> <li>• A constant voltmeter reading of 5V for 5V supply</li> <li>• A constant voltmeter reading of 12V for 12V supply</li> <li>• A constant voltmeter reading of 24V for 24V supply</li> </ul>

Table E21: In this table we discuss the risks mentioned above

Risk	Effects	Causes	Detection Control	Prevention Control
Poor soldering	PCB or components may not function	Wires not connected or damaged	Practice before doing the final PCB board soldering	Board does not function as expected, current & voltage doesn't flow as expected to each component
Placing traces too close together	PCB or components may not function	Short circuit	Follow proper PCB designing conventions, keep traces far apart	Can see traces touch or close
Component issue	No data or falsified data received	Mounting or manufacture	Test all components off the board before	N/A
Wrong connections	PCB or components may not function	Miss reading schematic and/or incorrectly installing components and/or incorrectly connecting wires	Double check wiring, use connectors that can only connect one way to prevent mistakes	Check wiring compare to the schematic
Open circuit	Operation will be erratic and unpredictable	Some components or sensors may not get the required power to function properly	Intense research on what each component can handle and testing individually and with full board	Lacking signals, no feedback or false feedback
Short circuit	Immediate damage to components or the board	Providing wrong amount of power to components could fry them causing them to fail	Have grounds, double check every other piece of PCB before powering	Could have flames, smell of burning, warm components, visible smoke

## **Software**

### **General Overview**

The UTHT Software System provides the ability to bring together all active components on the pod to ensure highly modular control and a robust communication system design. The software system enforces low-level management by multiple microcontrollers (Arduino Dues) for data monitoring and control. High level management of complex tasks is enforced by a central computer (Odroid XU4) for granular control over navigation. The idea behind this hierarchical design is to enable watchdogs, error detectors and redundancy against failures, which can be accomplished by the following subsystems:

- A [communication system](#) to allow seemingly disconnected components to work cooperatively and cohesively by ensuring efficient data and control signal transfer.
- A [high-level control system](#) to provide high-level autonomy by making informed decisions on pod behaviour based on sensor data and commands.
- An [infrastructure system](#) to provide a control panel interface to enable command of the pod and visualization of all systems and data streams.

## Communication System

The communication system connects all sensors using multiple Arduino boards to the master computer (Odroid XU4) for all high-level decision making. All Arduinos and the Odroid are connected in a network and data is shared by publishing on channels. The Odroid is subscribed to these channels to receive all the data for decision-making. Following this, the Odroid publishes its decision to Arduinos through channels, where the respective Arduino will implement the decision logic which is the system response to incoming data such as temperature or braking pressure. (Refer to Figure S1)

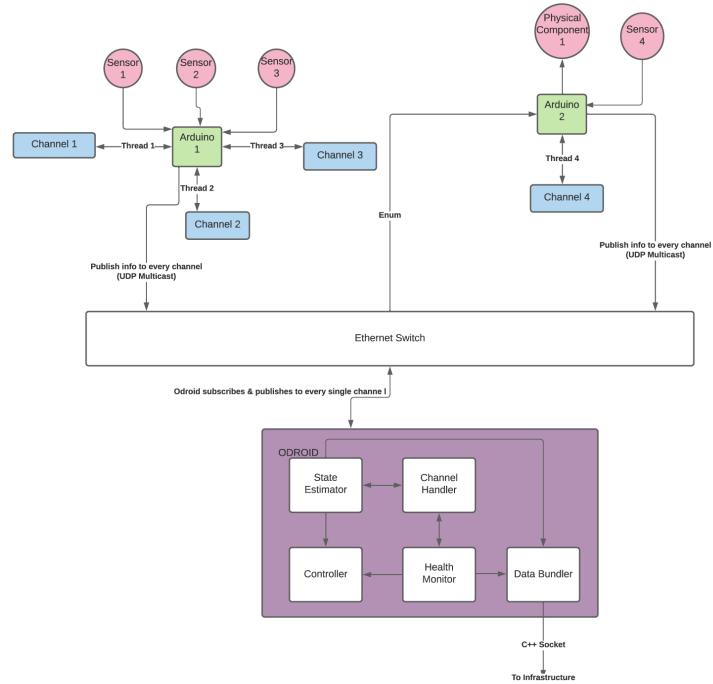


Figure S1: An illustration on how components are interconnected from microcontrollers to Odroid

All communications are handled through a single unified, broadcast-based message passing system, namely the Zero Communications and Marshalling (ZCM) system. ZCM utilizes a publish and subscribe model that uses channels to transmit information between hosts. Channels allow information to be efficiently transmitted between hosts. ZCM defines channel data by using a ZCM message type variable, which defines the primitive types that may be used within a channel for communication.

A host may publish information to a channel by defining an object as a ZCM message type variable. Another host may subscribe to the same channel and retrieve data as soon as it has been published. The publish and subscribe model of ZCM also has several additional benefits:

- Certain components in a system may choose to unsubscribe to a channel if the messages do not carry any significance to them.

- ZCM utilizes UDP multicast to efficiently transmit data to numerous hosts. This eliminates the need of sending data one-by-one to each host that is interested in the data.
- The maximum message size is 4GB which is defined as the maximum size that the UDP protocol can transmit.

### Integration with the Odroid XU4

The following describes key functions that are used for communicating with the Odroid. Refer to Figure S1 to see where in the system the functions are implemented.

Channel Class:

Data is transmitted through channels which controllers can subscribe and post messages to. Each sensor has its own channel for efficient communication of data. Table S1 describes the variables and member functions in the channel class:

Table S1: Describes the variables of the Channel Class and the function of each of them

Variable	Function
Channel Name	A unique identifier of the channel
Max/Min Values	Returns the maximum and minimum values a sensor can accommodate
Current Value	The current value returned by the channel
Semaphore	Restricts concurrent access to the channel variables to prevent corrupting channel data
validateCurrentValue	Ensures the values are within the required range.
subscribeToChannel	Allows a ZCM object to subscribe to a channel specified by the Channel Name variable
callbackHandler	Processes information published on a channel when data is received on a subscribed channel

Channel Handler:

The Channel Handler is a module located on the Odroid. The main goal of the Channel Handler is to update variables within each Channel object based on the values obtained from a channel. This allows other aspects of the Hyperloop pod to make decisions based on the obtained values. In the channel handler, a channel object will be instantiated for every sensor on the pod. Each

object includes information regarding the permissible sensor values, channel name, and validation and handler member functions.

The Channel Handler would subscribe to each ZCM channel by calling the subscribeToChannel() method function within the Channel object. This method subscribes to the channel and passes a callbackHandler() function specific to the ZCM object. When an Arduino publishes information to the channel, the ZCM object within the Channel Handler passes that information to the respective callbackHandler() function which updates the Channel object variables.

#### Health Monitor:

The Health monitor is responsible for validating each channel object and relaying whether the data received from an Arduino implies any issues in its performance. It will be used extensively in the Controller and Data Bundler subsystem. The controller will use its validation to determine the next state in its finite state machine. The data bundler will use the validation to display whether a system is performing as expected. Validation works by calling each channel's validation() function and checking its return value.

#### Tests

Multiple tests will be performed on all Communication System modules to achieve the goals of having a robust and efficient Communication System design for proper functioning and interconnection of all active components. These tests are described in Table S2:

Table S2: An outline of all tests to be performed on the Communication System modules

Test	Description	Implementation
Faulty sensor data	This checks for faulty sensor data and prevents sending it to the controller so no action is taken based on inaccurate readings.	The data will be published to a validation channel for checking before being forwarded to a subscribed channel.  The data will have a tag to identify the sensor and valid/invalid values.
Loss of communication on a channel	This checks if a sensor or Arduino disconnects from the network. This allows the channel handler to inform the state estimator and controller of this fault to take appropriate action.	A script publishing data to channels will stop publishing when a device disconnects.  Tests will then be run on each channel to specify the information to be relayed to the

		estimator ensuring it is up to spec.
Packet Loss	This ensures that some packet loss will not cause the pod to malfunction and ensure critical data is sent and received as needed.	A script has been developed to stress test the system by hitting all channels with different data types to establish a worst case scenario. Data received will be compared to the mock data that is supposed to be transmitted to assess the percentage failure if data was lost.
Failure in sending/receiving messages	This ensures that channels can publish messages or subscribe to messages, ensuring proper communications in the pod.	A script is developed to publish data to the Arduino serial monitor to ensure the channels send data.  A second script will be used to read incoming data, which will be displayed on the serial monitor.
Misconfiguration and Routing Errors	This test reads data values and traces the origin from a channel and sensor to allow for correct system response. This ensures all data is routed to the same place and that only messages which are published/subscribed to each other are sending data at the correct time. This conserves bandwidth of the network and prevents any confusion of data.	Publish test: A channel publishes to its subscriber channels some data, which is compared to ensure it remains the same. Channels not subscribed are also checked to ensure they have not received data from the publishing channel.  Subscribe test: A channel subscribes to multiple channels and the input will be checked when it attempts to receive multiple messages concurrently to ensure they are easily read.
Relating information from the Channel Handler to the controller on the Odroid	The Odroid must efficiently send data from the handler to the controller module. This helps the controller to make decisions based on data observed from various	The function call to pass data will be checked to see if it throws any error to investigate the handler to Odroid transmission.

	sensors and prevents loss of communication.	Ensure the controller module can accept data and can make meaningful decisions based on the data received. The output from the controller will be checked to test this.
Messages take too long to send/receive (delays)	The Arduinos and Odroid should send messages to each other quickly in order for Arduinos to relay the status and data from sensors to Odroid close to real-time. This helps to eliminate problems due to slow transmission of packets on a ZCM channel.	This test compares the timestamp when the message is sent from the Arduino with the timestamp when the message has been received by the Odroid. This comparison estimates the average delay in communication.
Testing the functionality of ZCM	ZCM tools will be used to analyze the data transmitted and observe any abnormal network or slow speeds. This ensures that the entire message is transmitted and message specific details can be seen.	<p>Logging Tool: All messages with the timestamps of each channel will be collected to ensure data is successfully sent.</p> <p>ZCM Spy: This tool records message specific data such as bandwidth and data values to debug any anomalies in data transmission.</p> <p>ZCM Spy Lite: This allows ZCM Spy to run on the terminal if a display environment is not accessible</p>
Testing the entire communication systems	After all unit tests (from each module), the entire system will be tested to ensure all interconnected components are operating efficiently and allow for fast and reliable data transmission.	A smaller model similar to Figure S1 will be constructed to ensure all devices are communicating by both sending and receiving messages from all channels. These components would be connected to a laptop which would have a synonymous role with the Odroid.

## Safety

The communication system provides multiple components with safety features to enable graceful handling of a variety of scenarios that can lead to failure. These features are described in Table S3:

Table S3: An outline of the safety features of components in the Communication System

Component	Safety Feature
ZCM	<p>The ZCM fingerprint ensures all components can communicate with each other and that they are compatible for data transfer. This prevents components from failing when receiving invalid data</p> <p>A safety risk arises when multiple messages are sent from multiple components concurrently. ZCM accounts for this by using a channel-based infrastructure where each sensor will communicate on separate channels. In the event of failure from a sensor, communication to other components are unaffected.</p>
Channel Handler	<p>Multiple modules on the Odroid use the information in a channel object to make decisions. The handler ensures that modules view the latest data obtained from each sensor.</p> <p>A safety risk arises when multiple modules access the same sensor data concurrently, which can cause race conditions. To account for this, a semaphore will be used to control access to each channel object which is a shared resource. The handler must obtain the semaphore for a channel before updating the channel variables.</p>
Health Monitor	<p>There is a possible race condition when the data is being updated via a ZCM channel during the health monitor's data validation. This is a high safety risk since the health monitor is integrated with the controller which makes a decision based on the data observed.</p> <p>This risk is accounted for by the locking mechanism in C multithreading. In the event the controller is trying to access data to make a decision, it will wait for the health monitor via a locking mechanism such as a semaphore to prevent race conditions.</p>

## High Level Control Systems

The control system regulates the behaviour of all the other subsystems present on the pod. It lives on an on-board single-chip computer (the Odroid XU4) and is responsible for monitoring sensor data, making high-level decisions regarding the pod's movement and relaying information to the pilot computer.

The core communications are managed by utilizing ZCM—a micro-framework for high-bandwidth, low latency message-passing and data-marshalling. The ZCM framework includes a logger that records all our ZCM traffic, and allows us to replay the recordings to perform appropriate diagnosis and data analysis after system tests.

## Closed-Loop Control System

The closed-loop control system is modelled as a Finite State Machine<sup>12</sup> (FSM), an approach which simplifies debugging, ensures modularity and uses our limited resources more effectively (See Figure S2). At any point in time, the pod is in one specific state. This state, along with inputs from the sensors and commands from the control panel, is used to make high-level decisions and move the pod to a different state. Each state transition sends a signal to the appropriate actuators.

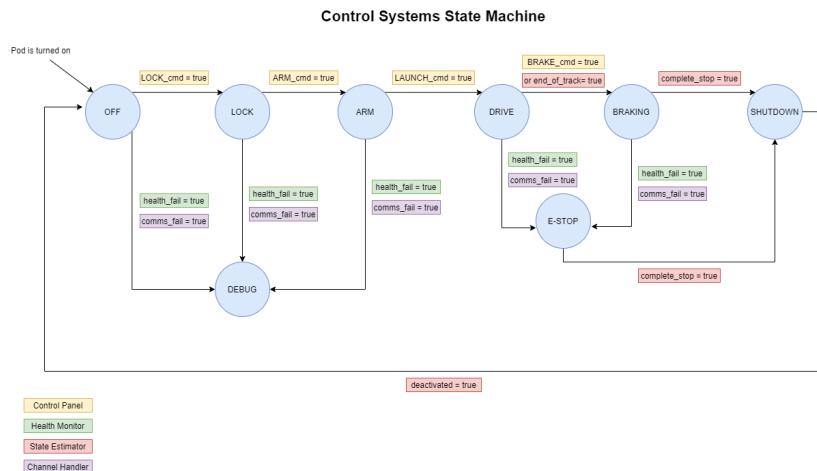


Figure S2: State transition diagram for the Finite State Machine model

The software is written entirely in the C++ language, which offers features such as object-orientated methods, template programming, and optimization. This aids our modular approach, and ensures that the system can react quickly to real-time events<sup>13</sup>. The system lives on

<sup>12</sup> A. Drumea and C. Popescu, "Finite state machines and their applications in software for industrial control", *27th International Spring Seminar on Electronics Technology: Meeting the Challenges of Electronics Technology Progress, 2004..* Available: 10.1109/isse.2004.1490370.

<sup>13</sup> C. Kormanyos, *Real-Time C++*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018.

a centralized computer on the pod - the Odroid XU4. It receives data from and sends commands to four peripheral Arduino Dues.

The FSM receives input from three sources:

**Health Monitor:** The health monitor continuously compares all sensor data against set parameters chosen by the sub-teams based on current safety standards and the needs of the system. Such sensor data include: battery temperatures, brake pressure, velocity, acceleration and lateral stability among others. If there is a discrepancy between the actual value and the desired change, an error signal is fed into the FSM.

If the pod is in motion when the error signal is received, it is moved to the E-Stop/Debug state (See Figure S2), which disengages the drive-train and vents the pneumatic brakes, bringing the pod to a complete stop quickly. Moreover, logged data is uploaded to the control panel in case of disk damage.

**State Estimator:** The state estimator receives sensor data and performs various calculations to estimate the pod's current position, velocity and tilt. As before, these values are compared against set parameters, and in case of a discrepancy, an error signal is sent to the FSM which then moves to the E-Stop/Debug state.

**Channel Handler:** While the Health Monitor and State Estimator are systems that ensure the control and safety of all the other subsystems, the Channel Handler handles the failure of the control system itself. It is the “roundabout” where all the sensor data is collected before entering the control system. It ensures valid data is supplied at a consistent rate from all sensors. Since establishing safety is almost completely dependent on reliable data, if any sensor stops providing data for a given amount of time, an error signal is fed into the FSM. Incoming data is validated to ensure correctness. If a validation check fails for several transmissions, an error signal is fed into the FSM.

In the event that the Odroid itself fails, the Arduinos on-board will detect the lack of communication from the control system and emergency brakes will be engaged, bringing the pod to a complete stop. The communication channels are monitored in both directions for this reason. The Channel Handler is also responsible for transmitting the pod's current state, sensor readings and any error messages to the control panel through web sockets.

## Control System Testing

We aim to test our software at two levels of granularity: the individual modules that make up the program through Unit Testing and the overall control system through Model-Based Testing (see Figure S3).

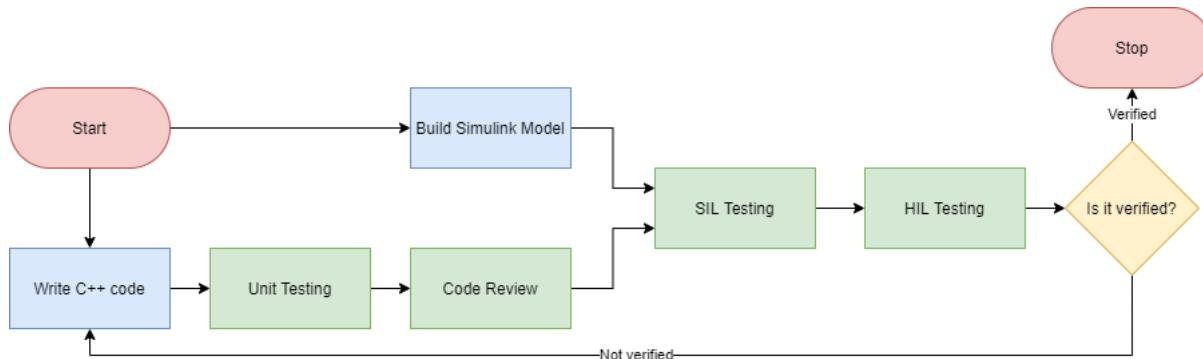


Figure S3: An overview of the testing procedure

## Unit Testing

A potential point of failure for the system is untraced bugs. To avoid this, it is necessary to isolate and test the functionality of the smaller modules that make up the program before testing the system itself. To allow for a continuous integration workflow, we aim to implement unit testing along with a strict version control system.

Staying with the modular approach, every part of the program is developed in an isolated version of the code, and once finished, goes through various automated unit tests to verify the correctness of the code. This allows us to pinpoint errors and modify points of interest with little consequence in terms of functionality elsewhere in the program. Once all the tests are passed, the code is reviewed by team members, who check for readability. The code is then merged with the stable version.

The general idea will be to individually test the decisions based on individual sensor data. Edge cases vary between sensors, unit testing ensures every respective edge case is tested thoroughly. For example, a function written to deal with temperature sensor data should immediately flag an issue if the temperature goes above a predetermined threshold.

This method allows for bug fixing early in the development cycle, and generates an efficient workflow where there is no wait time required for testing<sup>14</sup>. We will be using UnitTest++, a unit testing framework developed for C++<sup>15</sup>.

<sup>14</sup> Y. Cheon and G. Leavens, "A Simple and Practical Approach to Unit Testing: The JML and JUnit Way", Iowa State University Digital Repository, 2001.

<sup>15</sup> <https://unitest-cpp.github.io/>

## Model-Based Testing

It is critical to ensure that the control system is sensitive to all incoming data, can make appropriate decisions/state transitions and can communicate results to the output quickly. To test this functionality before the pod is actually running on a track, we use a model-based testing approach.

A model simulating the various practical situations the pod will encounter during operation will be created using Simulink<sup>16</sup>, a well-known modelling environment software. With an accurate model and using fault insertion, we can also simulate various ‘edge cases’ that may occur to increase the rigour of testing.

The testing procedure can be described in two steps:

1. Software-in-the-loop (SIL): Using a ‘C-Caller block’ in Simulink, our control system code will be integrated into the modelling environment. As our program is written using C++, it will be called using C wrapper functions. Once imported, it will be possible to see the software interacting with a real-time environment. Therefore, multiple situations and edge cases will be tested against the software and any incorrect behavior can be traced and fixed instantly, allowing for an extremely fast development cycle.
2. Hardware-in-the-loop (HIL): Here, we will replace the sensors and actuators with I/O from our model. Due to our software’s modular design and the ZCM infrastructure, it is an easy task to connect the system to our model. This will allow us to test communication between the various microcontrollers and the Odroid on the pod (see Figure S4).

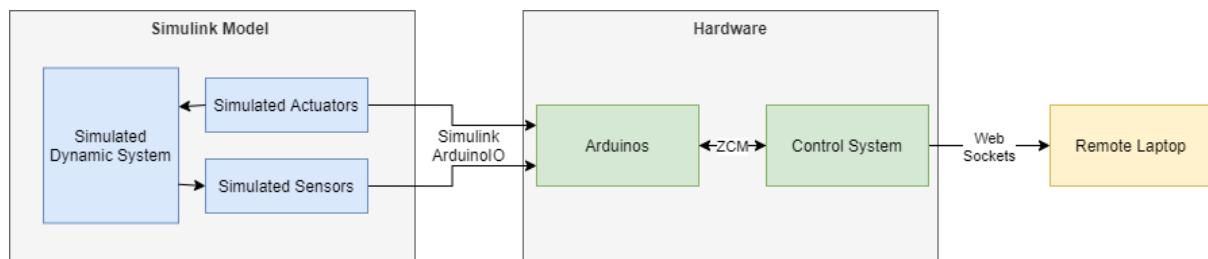


Figure S4: An overview of the HIL Testing procedure

Model-based testing has significant advantages in the development of the control system. This testing procedure requires no physical systems such as a testing track, empowering rigorous testing during the pandemic where in-person testing methods are limited. Using a simulated model is also more cost-friendly and safer, since no investments or efforts need to be made towards a possibly faulty system.

---

<sup>16</sup> <https://www.mathworks.com/products/simulink-real-time.html>

While unit testing allows to track local errors of a system, the system's response to holistic changes are more complicated to trace. For example, we can monitor how the system behaves if multiple sensor faults are detected in quick succession. The model-based testing approach shines in cases such as these. Testing with software is quicker and more versatile than a physical system.

## Infrastructure

The infrastructure sub-system provides a control-panel Graphical User Interface (GUI) for the user to interact with the pod. This allows the user to monitor live data ensuring the pod runs correctly. The system carries out data transmission from the Odroid master computer to the central backend system in the laptop.

## Components and Architecture

The central system is an intricate and comprehensive data parsing, validating, encapsulating, and storing mechanism that acts as the backbone of the entire subsystem. The validated data is sent to the frontend widgets to be analyzed by the user to detect any errors which are further relayed to other subsystems (energy, electronics, propulsion etc.). This process allows subsystems to debug their components facilitating safe operation of the pod. In the frontend, the widgets depict various graphs and charts that display sensor readings. A 2-D Bird's-eye view module is integrated to easily locate sensors with abnormal readings whose unique ID is displayed allowing for easy debugging. A backend validation module is developed for the sole purpose of verifying the Odroid's data integrity. Lastly, a scalable NoSQL database that stores JSON-format data acts as the pod's backtracking system storing all data gathered during the pods operation.

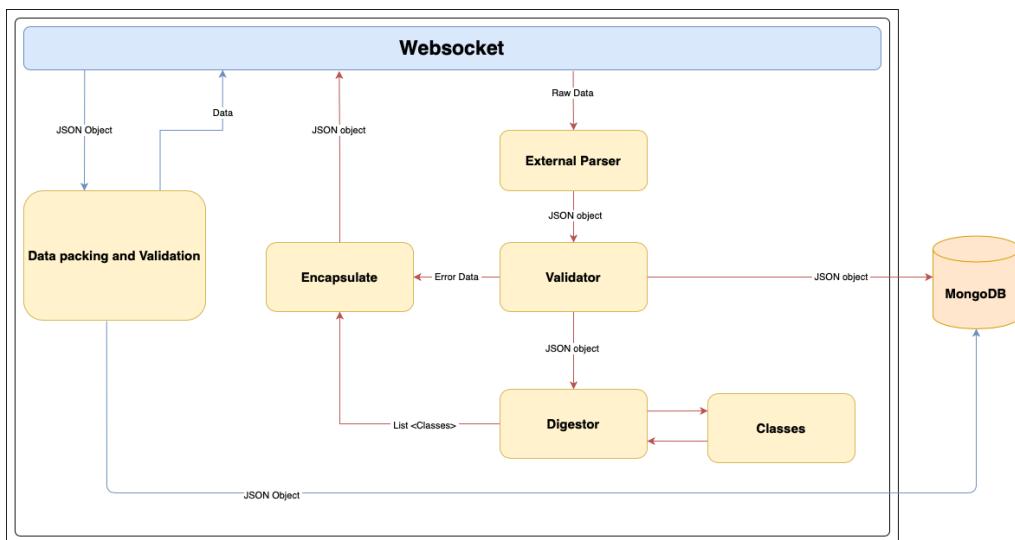


Figure S5: Data flow at the server side (backend)

## Interaction with Odroid

One of the essential components is the connection between the Odroid Master computer and the web socket on the server-side. In order to achieve that, the team created a special API that can be leveraged to send data to the server from the Odroid in a timely and efficient manner. Once the [control system](#) receives data from individual components inside the pod, it dumps the data into the API. Once the Odroid knows the web socket is still open, it then pushes the string data to the web socket.

## Web Socket

A web socket module is utilized to establish a real time connection with the odroid and GUI control panel. It is the first component that receives data from the Odroid. The web socket provides us the ability to do fast and efficient data transfer in a very short amount of time. It is evident that the HTTP benchmark peaks at about ~950 requests per second while socket.io serves about ~3900 requests per second. They can be relied upon for accurate, fast, and robust communication between the database, the pod, and any intermediary modules.

## Validation

Data that is received from the Odroid is converted to JSON data via the External Parser and then passed to the Validator module. This module verifies the content of the data received from the odroid before sending it further to the Encapsulator, Digestor, or the MongoDB database as depicted in Figure S5. The Validator module's functionality is three fold: validate incoming sensor data, generate an array of errors based on the validation failures, and push data to the subsequent modules. The error array contains error objects that can be displayed on the Frontend widgets as shown in Figure S6. Data verification comprises of two checks:

- **Data Correctness:** The Validator checks for any loss of packet or information while data is transferred from the Odroid to the web socket. Specifically, data correctness is verified using the following parameters:
  - Range check: Range check ensures the value received for each component is within the minimum/maximum range allowed. For example, the value received for speed can not be less than 0.
  - Data-type check: Data type check verifies the data value received for each component is consistent with the expected data type for that component. For example, the temperature value can not be of data type null.
- **Structural Correctness:** This module ensures the data received from the Odroid is in the correct format. Moreover, this module decides whether the data structure is consistent enough to be sent for further processing.

## Database

The team is using MongoDB, a NoSQL database, as it offers flexibility for schema creation (equivalent to tables in SQL) and allows us to directly save data in a JavaScript Object Notation (JSON) format. The database logs every command that the pod undertakes (ex. Debug Mode, Comms Check, System Check, etc.) as shown in Figure S2 . Additionally, MongoDB records a timestamp from every pod run to enable backtracking of operational history. As described in the “Testing and Error Handling” section, the Validator ensures all incoming JSON data is structurally and syntactically correct.

## Encapsulation and Digestor

The digester on the server is used to process incoming JSON data that comes in from the Odroid, after it is processed through the validator. It simply translates the sensor properties into specific custom sensor objects such as SpeedSensor, BatterySensor, etc. The digested array would then be passed through an encapsulator. The encapsulator module converts the array of sensor objects into an object with the sensor type as a key and an array of the appropriate sensor object as the value. Encapsulator also packs all the error data that are generated from the validator module.

## Dashboard and UI

The frontend system has a few components connected to the web socket. As shown in Figure S6 below, the data from the web socket is digested and validated, then stored globally. All the dashboard components access the globally stored data which then is displayed with the widgets in the control panel GUI. Additionally, there is a logging system which can be accessed in the control panel GUI, allowing the user to access previously stored data.

Figure S7 shows the current iteration of the dashboard graphical user interface. Sensor readings for temperature, battery and pressure levels of each component are displayed on gauges and charts. Velocity is displayed on a speedometer and the pod’s position, with respect to the length of the track, is displayed on a progress bar.

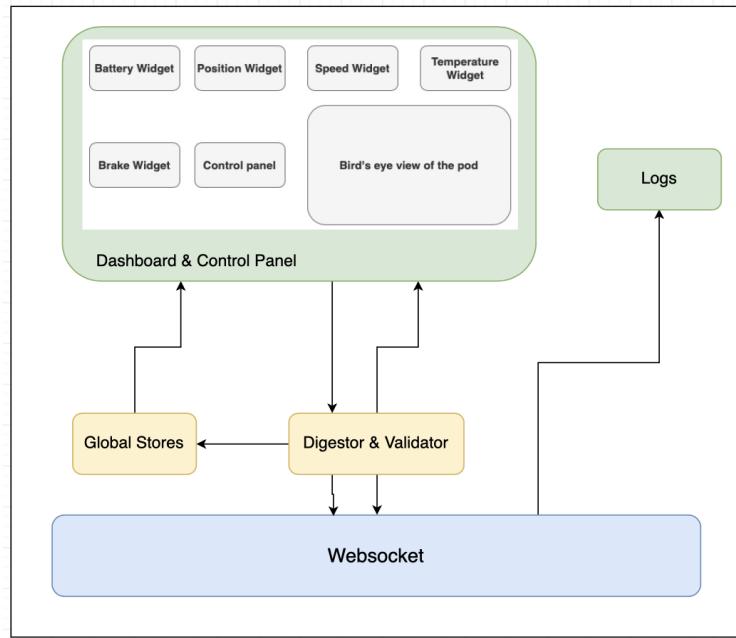


Figure S6: An illustration of the Front-end System



Figure S7: Pod Analytics Dashboard

The system contains a state diagram displaying the pod's current state. Buttons allow the user to change the pod's state and launch the pod once the pod is in the armed state. Manual brake and emergency stop buttons are provided to the user as additional safety features to override the pod.

Currently, there are lights that display sensor status but the plan is to replace this with a more visual approach with a bird's-eye view of the pod as shown in [Appendix S1](#). The dashboard relies on data input from the database in the backend system. Live data is displayed automatically and previous data stored in the database can be accessed using the log system at any point during the operation of the pod.

### Front-end digester and stores

Data from the server arrives at the front-end through a web socket angular service component. The service is used such that data can be easily shared between all components on the dashboard. Within the service implementation, an array is used as a global data store for all incoming data. When an encapsulated data object arrives to the service, it simply pushes the entire object to the array and sends out a signal to all subscribed components that data have been updated. To access data from the service, a component can simply subscribe to a Subject provided by the service. This allows the component to observe for changes in the latest data that is coming in and update its ui representation accordingly. Each component must keep track of what it needs to parse out from the entire encapsulated object which is simple since the encapsulated object already come in a digestible form with sensor values mapped to their respective keys

### Safety - Testing and Error Handling

#### Testing the database:

The database's most vulnerable component is the data it stores. Inaccurate data storage hinders the pod's performance analysis and leads to delays in debugging problems. Therefore, to eliminate such trouble, the team is using Jest and MongoDb In-Memory Server to unit test the Mongoose database. Jest simplifies creating unit tests by providing in-built exceptions, and MongoDB In-Memory Server is necessary as unit testing data is stored in memory rather than in the actual database. Examples of unit tests include simulated sensor data that is completely valid, completely invalid, and a combination of both valid and invalid.

#### Testing for Validator Module:

The *Validator* module comprises checks for every sensor and its respective data. Data verification is the highest safety risk as improper checks lead to inaccurate data transmission to the frontend widgets from which the user is unable to identify issues and debug complications. Thus, the module is tested using a combination of White Box and Black box testing techniques. White box refers to testing the module's internal structures that facilitate the various data checks by feeding it valid and invalid simulated sensor data. Black box focuses on the overall functionality by examining the error messages. Jest is used to create the unit tests for both testing methods.

## HeartBeat Mechanism:

One of the major techniques we are leveraging for error handling is *Reactive Failure Detection (RFD)*. More specifically, one of the major safety measures we are taking is implementing a heartbeat mechanism between the Odroid master computer and server-side nodejs. A periodic signal, consisting of a simple payload, is sent to the server-side from the odroid to indicate its immediate health and to make sure the connection is established. The server side returns a heartbeat to indicate to Odroid if the server is still active and working in full strength. If the server-side does not receive a heartbeat in a certain interval, and a timeout occurs, it assumes that the connection between Odroid is lost, indicating the user for immediate action. Similarly, if Odroid does not hear back from the server within a certain time interval and a timeout occurs, it assumes the server-side has malfunctioned and changes its state appropriately.

## **Macro-Integration of Subsystems**

### **Integration**

#### Structure System

The structure system is responsible for both the pod's shell and its frame. All components of the other subsystems are mounted on the frame, which is split into two subcomponents: the sheet metal base and carbon fiber tube backbone. The structural components were designed to allow sufficient space for other components, while avoiding the risk of comprising aerodynamics. The aerodynamic shell covers and aims to protect all components from any external (mainly aerodynamics) forces.

#### Propulsion System

The propulsion system's components are designed to fit within the confines of the geometry developed by the structures team, encouraging weight savings. The motor at the rear of the pod sits on the metal base as low as possible to ensure that the center of mass of the pod is as close to the track as possible. Additionally the braking system integrates into the propulsion system via two pairs of braking pads placed at the center of the pod responsible for bringing the pod to a stop at the end of each run as well as in the case of any emergency brakings.

The electric motor operates on a 3-phase AC current. This is supplied by a set of lithium ion batteries whose DC current is converted to 3-phase AC via an inverter, which is also responsible for drive frequency control. The aforementioned mechanisms are conducted by the energy system integration into the propulsion system. For optimal and safe operation, the pod's vertical and lateral distance from the track, its spatial orientation, and the motor temperature are monitored during the run. This is done via placing a set of sensors in close proximity to the wheels for distance monitoring, a gyroscopic sensor for the pod's spatial orientation, and feeding the motor temperature profile to the controls which is all done by working with the electronics system.

#### Stability System

The pod operates on a bilateral stability system: two pairs of wheels run along the bottom of the top flange of the track's I-beam and are situated at the front and back ends of the pod. This clamping system is responsible for vertical stability and uses shocks to mitigate the effects of any vertical accelerations. Additionally, two pairs of wheels also at the front and back of the pod run along the web of the track's I-beam. This second set is responsible for the lateral stability and prevents yaw and roll. These wheels as well as their respective suspension systems are classified under the Stability System and are integral in the propulsion of the pod as they mitigate energy losses and help in stabilizing the pod.

### Braking Subsystem

Two pairs of braking pads are located to the left and right of the pod's center which pneumatically lift the pod for the pads to make contact with the track and stop the pod at the end of the runs and in the case of emergency stops.

Electronics system connects solenoid valves and pressure transducers to the main control board and the energy system supplies the power needed to open the solenoid valve. Sensors send instructions to electronics whether to open or close the solenoid valve and help to monitor pressure inside the system for any anomalies.

Structures system mounts braking components to keep the system rigidity. Braking also depends on the Stability system to keep the appropriate operational gap between the brake pads and the I beam flanges.

### Energy System

The energy system will be used to power the electronics and the pods onboard motor. It is made up of 2 power systems: high voltage (HV) Power and low voltage (LV) Power systems. Both systems are equipped with batteries to provide power to their respective systems. The HV power system will be used to provide power to the inverter and motor. The LV power system will be used to power the pods electronics and communications systems. Both the LV and HV batteries will be removed from the pod for charging, and they will be plugged into the charging system. The charging system will require a main power outlet capable of handling at least 750W.

Propulsion system operates on a max 52kW continuous power electric motor, but for the EHW demonstration we will be running it at up to 15kW only. The motor uses a 3-phase AC current fed by lithium ion battery packs. The DC current drawn from the batteries in the HV power system is run through an inverter which converts it to 3-phase AC. The inverter is also responsible for drive frequency control because it converts the DC voltage to the 3-phase AC required by the motor. Both the battery pack and the inverter fall under the Energy system which is therefore closely integrated into propulsion.

The inverter controls and input for desired drive frequency are handled by an arduino which interfaces with the inverter via CAN BUS. This arduino is part of the Electronics system and the control is performed through software. Therefore the HV power system requires the software and electronics sub-system to operate. They are interconnected, and the software and electronic control systems are what will allow us to control the operation of the HV power system. The low voltage power system does not rely on any other systems to operate.

## Electronics System

For the electric motor to operate at optimal performance conditions, the drive frequency should be controlled and monitored; this is done by the Electronics system. Run initiation, braking, and emergency stops at pre-planned positions and based on the sensory system data are also conducted by the electronics system. Various metrics involved in the propulsion of the pod such as height, temperature, and pod's spatial orientation are closely monitored by the system controls. The sensing system recording and measuring these metrics is provided for by the electronics system. Pod propulsion requires the lateral and vertical distance from the track data in order to make sure they fall within the safety margin obtained by vibrational analysis. For this, sensors are installed on the pod front and rear sections measuring the distance from the track both laterally and vertically. Moreover, a set of gyroscopic sensors are installed at the center of the pod in order to provide spatial awareness. Lastly, temperature sensors feed the motor temperature profile motor to the controls system during the run.

## Tests

This section goes over the tests that ensure the individual sub-systems and their respective safety features are functioning when they are fully integrated. These tests all require that the entire pod be fully assembled, with braking tanks filled, batteries fully charged, and all electronic components connected. The tests have been separated into mechanical and electrical sections to highlight the different levels of integration in the pod. However, it should be noted there is a large overlap between them when performing tests (e.g. any integrated tests for propulsion requires the energy system's battery for operation). A full operation test on a test track (currently being developed) will also be done to test every major component of the pod and their associated design points.

## **Mechanical Systems Integrated Tests**

Mechanical systems require multiple tests to demonstrate the feasibility and reliability when all mechanical systems interact with each other. These tests are adopted from previous subsystem-level tests

Table M1: Mechanical systems general tests for a macro-integration verification

Test	Purpose of Test	Expected Result
Electric Motor	Need to verify motor provides enough force and torque to move the combined weight of the pod	Motor should be able to activate and transport the pod on an I beam
Brake Check	Non-emergency or emergency situation require the use of brakes, so a test to test this is required; mounting of components must be	Upon activation, brakes are deployed within the correct time frame; also all components will be intact

	sustained while and after braking	
Stability Shocks	Stability shock dampen movement and vibrations to components in the pod and thus must be verified in operation	Shocks from both the stability system and propulsion help in reducing any excessive vibrations to other components in the pod (e.g. no loosened screws)
Full Mechanical Operation	Verifying all mechanical systems meet their design points	Result Consistent with defined design points for each mechanical subsystem either through subsystem testing or simulations; no major flaws detected

### Electrical Systems Integrated Tests

The software systems allow for the control of the pod, while the electronics systems convert this high level control into low level, real world action. The Energy Systems provides the necessary power required for these actions, such as driving the motor. These 3 sub-systems make up the electrical systems of the pod. In doing the following tests, we ensure that all of the control systems are fully operational along with the active systems that power them.

Table M2: Mechanical systems general tests for a macro-integration verification

Test	Purpose of Test	Expected Result
Brake Check	With pod fully stationary, in the “Ready” state, test the deployment of the braking system from the control panel	The brakes should deploy right after the command is issued to the pod from the control panel
Cooling System Check	With the pod fully stationary, in the “Ready” state, test the functionality of the cooling system.	The fans & pumps should be running, the inverter temps should be at or below ambient. Flow sensor should provide positive readings.
Sensor Check	With the pod fully stationary, in the “Ready” state, test the functionality of the sensors.	The sensors should all provide readings to the dashboard consistent with
Drive-train Check	With the pod fully stationary, in the “Ready” state, test the functionality	The wheel should spin. Any change in frequency input to

	<p>of the electric drivetrain, propulsion and control systems. The pod will be held in place using mounts, and the motor will be held in place using the X frame and spun up to similar conditions to running track operations.</p>	<p>the system via the control panel should be reflected in the motor speed. Cooling system should maintain the temperature within acceptable levels.</p>
Flight Test	<p>With all the sub-systems verified functioning when fully integrated, we will test the pod running on a track over a series of tests in order to verify its ability to run down the track and operate safely.</p>	<p>The pods operating conditions (Temperature, pressure, etc) should not exceed threshold values and the pods different subsystems should function as expected when running the pod.</p>

## **Transportation of POD**

### **Overview of Carrier**

The function of the carrier is to aid in transporting the pod from storage onto the hyperloop track while minimizing risk of damage to the pod and physical injuries. The carrier must be structurally sound, must be able to support the weight of the pod, and should be light enough for it to be portable. The process of material selection was conducted using CES EduPack Eco 3, and the CAD model and simulations were done on SOLIDWORKS and ANSYS 2019 R2 respectively.

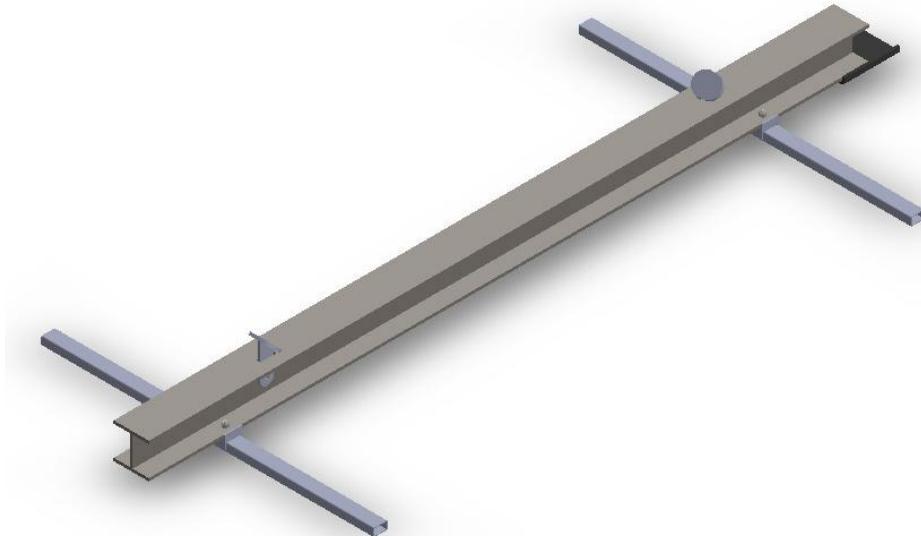


Figure C1: Carrier Assembly

## Parts List

**Table C1: Carrier BOM**

Part Name	Material	Dimension/Unit (LxWxH)	Qty	Function
<b>Carrier</b>				
I-Beam	Cold rolled steel C1003	Length:127mm Height:127mm Flange Thickness:10.5 mm	1	The Pod will be mounted on the I-Beam. The I-Beam will allow for easy transfer of the pod onto the rail.
Rectangular Rods	Cold rolled steel C1010	1200x50x30mm	2	The Rectangular Rods are used to attach the eyelets and Nylon Slings to the I-Beam, while also ensuring that the Pod remains balanced while on the carrier
Nylon Sling	Nylon fabric	914.4 mm x 25.4 mm	4	The Nylon Slings will connect the carrier and engine hoist, and support the pod on the carrier
Rubber Stoppers	Silicone	200x151x25mm	1	Align the I-Beam with the rail to ensure easy and safe transfer of the pod
Eyebolts	Zinc-plated steel	2 $\frac{7}{8}$ ", $\frac{7}{8}$ " eye diameter	4	Attach the Nylon Sling to the Rectangular Rods
5/16"-18 Nuts	Zinc-plated steel	$\frac{1}{2}$ " width, $\frac{17}{64}$ " height	4	Attach Eyebolts to Rectangular Rods
Wheel Stoppers	Mild steel	122 mm height, 100 mm face width, 75 mm face height	2	Stop the pod from rolling on the I-Beam
M12 Bolts, Rectangular Rods	Zinc-plated steel	72.5 mm length	4	Bolt the Rectangular Rods to the I-Beam
M12 Bolts, Wheel Stoppers	Zinc-plated steel	35 mm length	2	Bolt the Wheel Stoppers to the I-Beam
M12 Nuts	Zinc-Plated Steel	10 mm height	6	Secure the Wheel Stoppers and Rectangular Rods

Lifting S-Hook	Alloy steel	$\frac{1}{4}$ " diameter, 2 $\frac{3}{4}$ " length	4	Attach the Nylon Sling to the Eyebolts
----------------	-------------	----------------------------------------------------	---	----------------------------------------

**Table C2: Dimensions of all the Carrier**

Assembly	Length (maximum dimension)	Width (maximum dimension)	Height (maximum dimension)	Estimated Mass
Carrier	2600mm	1200mm	230.21mm	94.7 kg

**Full description of subsystem separated in sections by major subassemblies**

Material Selection

After conducting material selection using the appropriate MPIs and CES EduPack Eco 3, it was determined that cold rolled steel is the ideal material for the construction of the I-beam of the carrier. Cold rolled steel (C1003 Steel) has a smaller grain size than regular steel which enhances its ability to resist deformation. This reduces the chances of the beam flexing while transporting the pod. The C1003 steel is also readily available, and is a solid material, hence it meets the material requirements for the pod. Upon repeating the material selection process to find the ideal material to suspend the carrier, nylon rope is found to be the ideal material of choice. Nylon rope is less heavier than chains, able to support heavy loads, and is likely to cause damage to the surface of the pod.

Carrier Material Selection

Similar to Section ST3.1.2, the material performance index (MPI) used to determine the best material for the carrier was derived to minimize the deflection of a material. Minimal deflection is ideal for the design as the beam should be able to support the mass of the pod while experiencing minimum to no deformation. The log of the MPI was then rearranged to give a linear equation which will be used to select an ideal material for the carrier. The material of the carrier must strictly be made of solid materials, hence any fibrous materials are not considered for selection. In addition to the MPIs and the requirement for the material to be solid, several other requirements for the materials must be met, such as availability and cost. I-Beams are generally manufactured out of aluminum and steel, hence, only these materials will be

considered as choosing other materials might not guarantee the availability of an I-Beam with that material of choice.

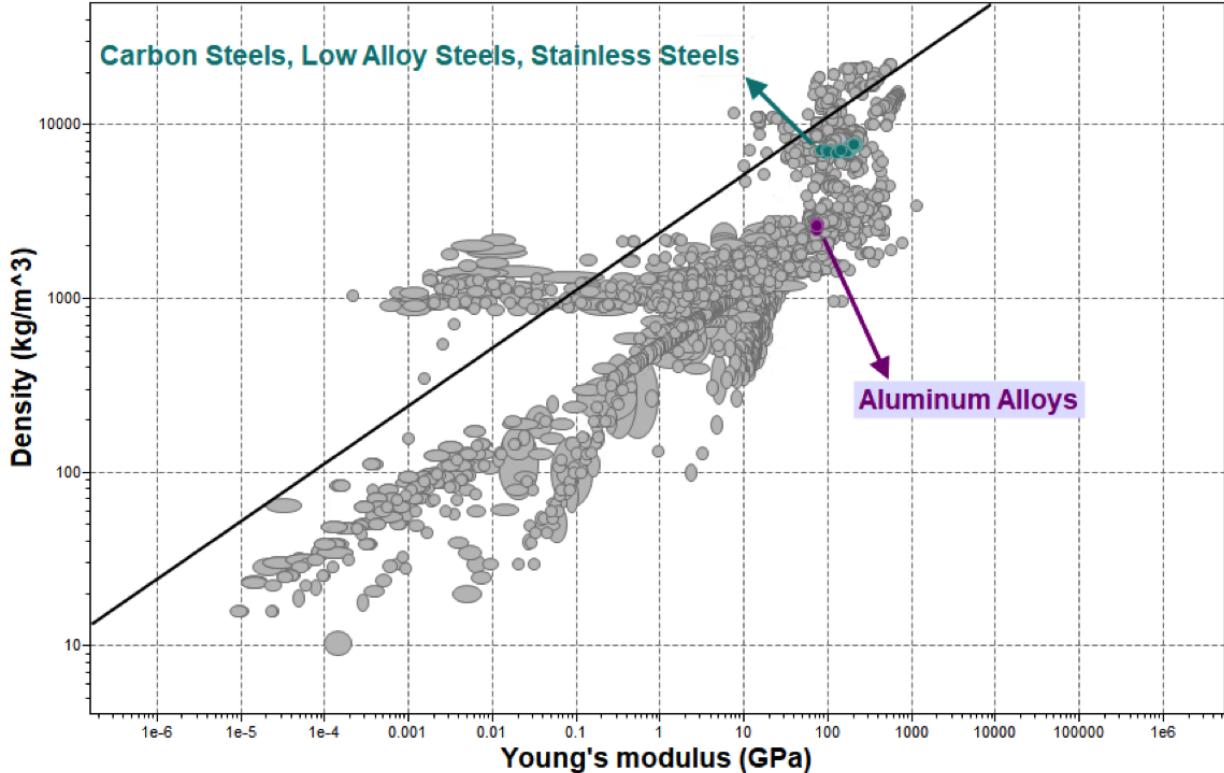


Figure C2: CES Plot for Selection of the I-Beam of the Carrier

### Carrier Design Consideration

The previous design iteration of the carrier was designed to be carried by team members, however, European Hyperloop Week regulation requires pods to be transported by an engine hoist. The new design iteration of the carrier was made to ensure that the pod can be safely lifted by an engine hoist and easily transferred onto the hyperloop track. The pod carrier consists of an I-beam, two square tubes that are bolted perpendicular to the I-beam, and the entire structure will be attached to the engine hoist by nylon ropes that connect to the square tubes. The I-Beam selected is identical to the specifications of the hyperloop rail to ensure that the pod can easily be transferred onto the test track without removing and reassembling parts. The square tubes due to their ease of drilling, ease of acquisition, and high tolerance to stresses placed perpendicular to their axis. The nylon slings will be attached on each end of the square tube, ensuring even load distribution while reducing the chances of the carrier from tilting forwards, backwards, or

sideways during transport. The nylon slings were selected to support the carrier as they exhibit excellent strength, are lightweight, and affordable.

### **Micro-Integration of Structures Assemblies**

Figure C3 shows the I-Beam of the carrier as it will be delivered. Given the difficulty of milling steel of the thickness of the I-Beam milling of the slots and drilling of the holes in the I-Beam will be performed by the supplier, who will also supply the brackets used to connect the I-Beam to the Rectangular Rods.



Figure C3: I-Beam of the Carrier

Figure C4 shows the Wheel Stopper attached to the I-Beam, both Wheel Stoppers will be milled from steel plates, assembled by welding, and bolted to the I-Beam after the pod is loaded, as the purpose of the Wheel Stoppers are only to keep the Pod in place once loaded.

The next step in micro-integration is to drill holes in the Rectangular Rods (Figure C4, both for I-Beam attachment and attachment of the Eyeloops), which can be performed with a drill press due to the 3 mm thickness of the Rods, and then to bolt the Rectangular Rods to the I-Beam with the 12M Rectangular Rod Bolts, securing their attachment with the 12M Rectangular Rod Nuts.

Next, the Eyeloops are mounted to the remaining holes on the Rectangular Rods, and secured with the  $\frac{5}{8}$ "-18 Eyeloop Nuts. The  $\frac{5}{8}$ "-18 Eyeloop Nuts will be tightened to prevent rotation, as these could loosen the nuts.

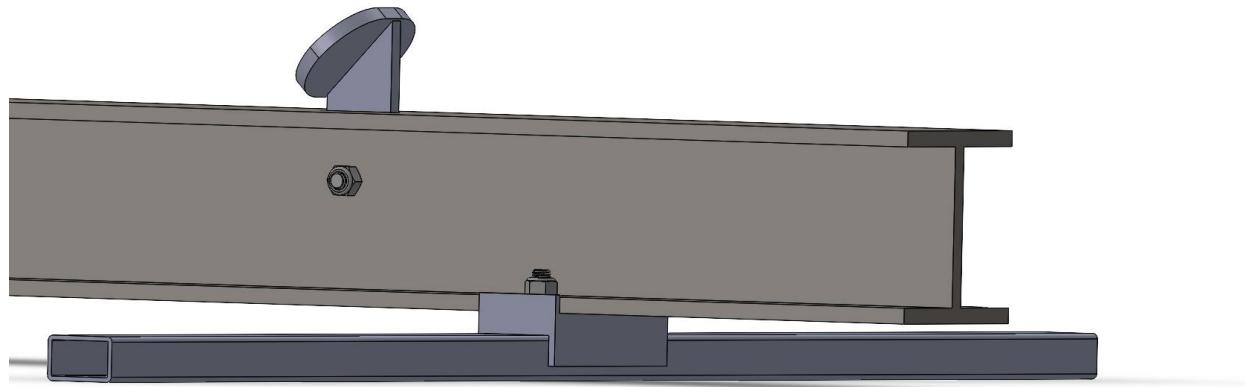


Figure C4: Rectangular Rod Mounted to I-Beam with Bracket Attachment

Finally, the Rubber Stopper (Figure C5) is epoxied to the end of the I-Beam, at roughly half its length, although excessive precision is not required in this step, as the Rubber Stopper guides itself when attaching it to the I-Beam, and so it will always ensure correct alignment with the track I-Beam if glued in place along the bottom side of the carrier I-Beam.

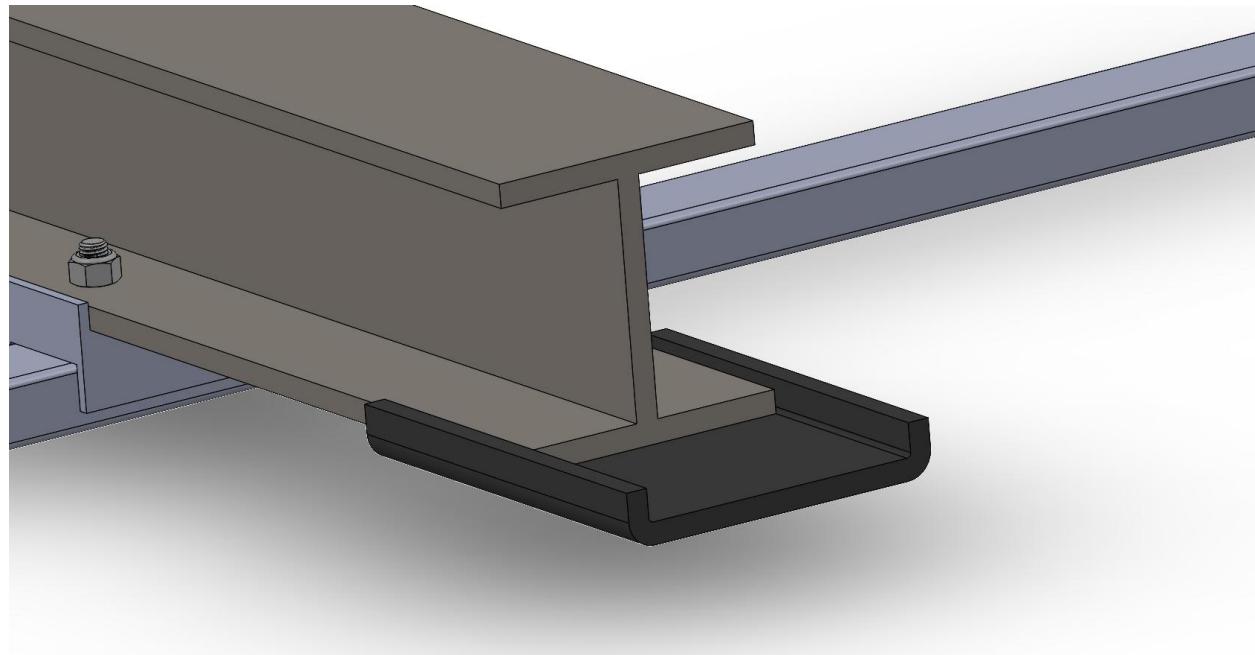


Figure C5: Rubber Stoppers to Align Rail with Carrier I-Beam

## Safety

**Table C3: FMEA Spreadsheet for the Carrier**

ID	Process/Mechanism	Risk	Effects	Causes	Prevention Control	Detection Control	Severity Rating (S)	Frequency of Occurrence (O)	Ease of Detection (D)	Risk Priority Number
1	Carrier support	Pod breaking off carrier causing damage	Pod endangered to a drop and damage to all components, particularly the wheels, suspension, and sheet metal	Critical connection point failure.	FEA analysis (static)	Cracks, signs of wear, etc	8	2	1	16
2	Carrier manufacturing	Defect or error during manufacturing	Affected component damaged or destroyed, requires replacement	Misuse of tools, use of tool in incorrect context, manufacturing defect	Training, quality control of purchased components	Cracks, signs of wear, supervision during manufacturing	4	2	1	8

## Simulations

Due to the simplicity of the carrier's design and the strength of the components with which it was designed, one simulation was conducted to verify that the carrier would withstand the load of itself and the pod. To minimize the risks presented in ID3 and ID5 of Table ST4 verification of these simulations requires a Safety Factor of at least 2.<sup>17</sup> It was already established in pre-design that the pod would not tip on the carrier due to the center of mass of the pod and carrier being equidistant from the two Rectangular Rods, and so a balance simulation was deemed unnecessary. Additionally, it was not expected that the carrier would experience dynamic forces different to those it experiences in static loading situations, as the carrier always moves slowly, and so dynamic simulations were not considered. Finally, simulations were not required for the nylon slings, as each one individually can hold the mass of both the pod and carrier, nor were they considered for the bolts and nuts, as none of the stresses involved in the carrier come close to the failure stresses of the bolts and nuts.

<sup>17</sup> "Hyperloop", *SpaceX*, 2020. [Online]. Available: <https://www.spacex.com/hyperloop>. [Accessed: 13-Dec-2020].

## Loads and Constraints

The load of the pod and the carrier (150kg + 95kg) was distributed equally between the locations where the front and back wheels of the pod sit on the carrier, resulting in a force of -1202 N on each point, or a net force of -2404 N. The ends of each rectangular rod was taken as a fixed point, as opposed to the exact eyeloop location, to add a safety margin to the simulation.

## Results

Figures C6 And C7 below show the deformation and stress plots, respectively. The deformation plot shows most deflection in the center of the carrier and the center of the rectangular rods, as these sections are most distant from the fixed points of the eyeloops. However, even the maximum deformation of  $8.7135 \times 10^{-5}$  m is considered to be insignificant, and noticeable when using the carrier. This is further demonstrated in the stress plots, where the maximum stress is  $\sim 10$  MPa. Based on the yield strength of the steel I-Beam and Rectangular Rods being  $\sim 250$  MPa, this provides a Safety Factor of  $\sim 25$ , and thus this reduces the risks present on the carrier in terms of mechanical failure, and suggests that the carrier is highly unlikely to fail.

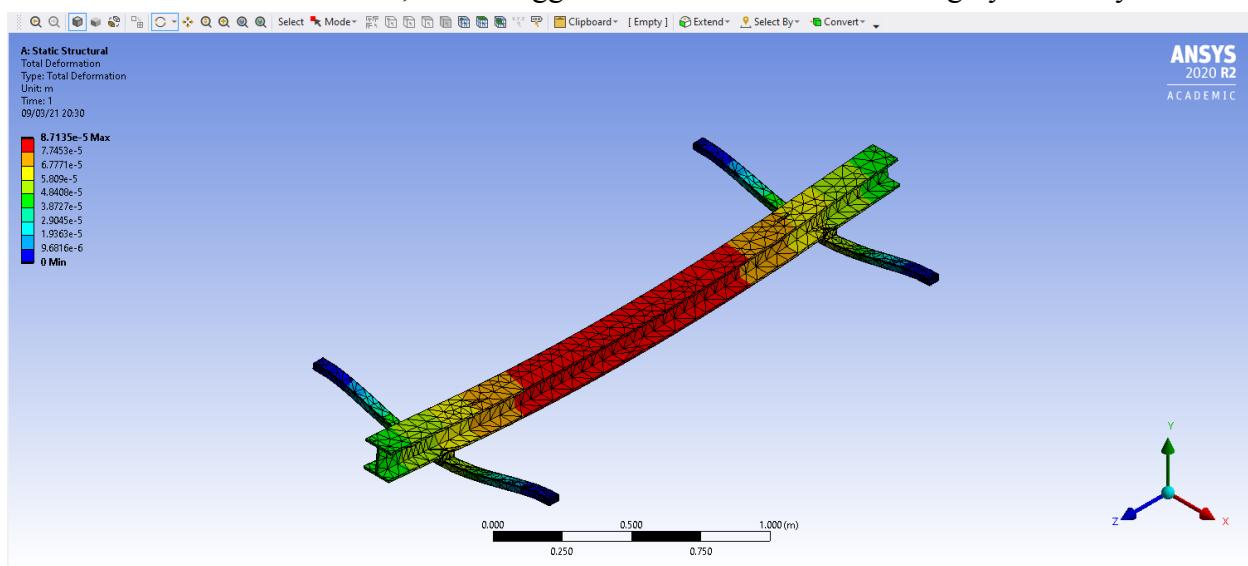


Figure C6: Deformation of the Static Carrier Analysis

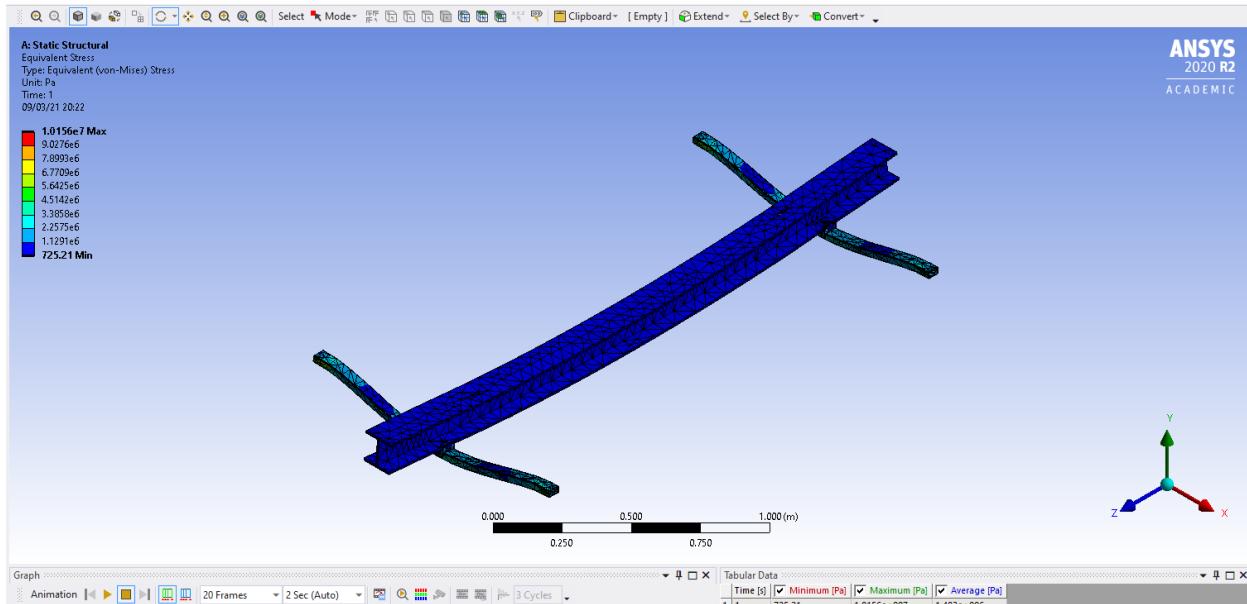


Figure C7: Von Mises Stress of the Static Carrier Analysis

## Standards

As the Carrier does not use any materials that are not already used by the Structure of the Pod (Carrier only uses steels and has minor use of rubber), all standards for material selection, machining, and bolts are the same as for the Structure of the Pod, and can therefore be found in the Structures section.

## Tests

Required testing for the carrier is minimal, as the chosen structural elements for the carrier are able to support loads multiple times greater than the total load of the carrier and pod combined. However, the assembled carrier will be tested by loading it with underslung sandbags of a mass of 140kg, to simulate the mass of the pod and some additional mass for a safety margin. The carrier will then be lifted with an engine hoist available at U of T, similar in specifications to the engine hoist that will be available at EHW, and it will be visually observed for buckling and deformation. In simulations, the carrier was demonstrated to only suffer deformations in the millimeter scale, therefore any deformations that are larger than 1 mm will signify a defect, and be grounds for remanufacturing of the carrier.

## Failure Mitigation

Failure mitigation is most important during the manufacturing stage, where loose bolts can lead to weakness in the structure of the carrier, as such failure mitigation is conducted by strict quality control and ensuring all bolts are tightened as required by the torque standards in the Structures section [IT7, ITD8].

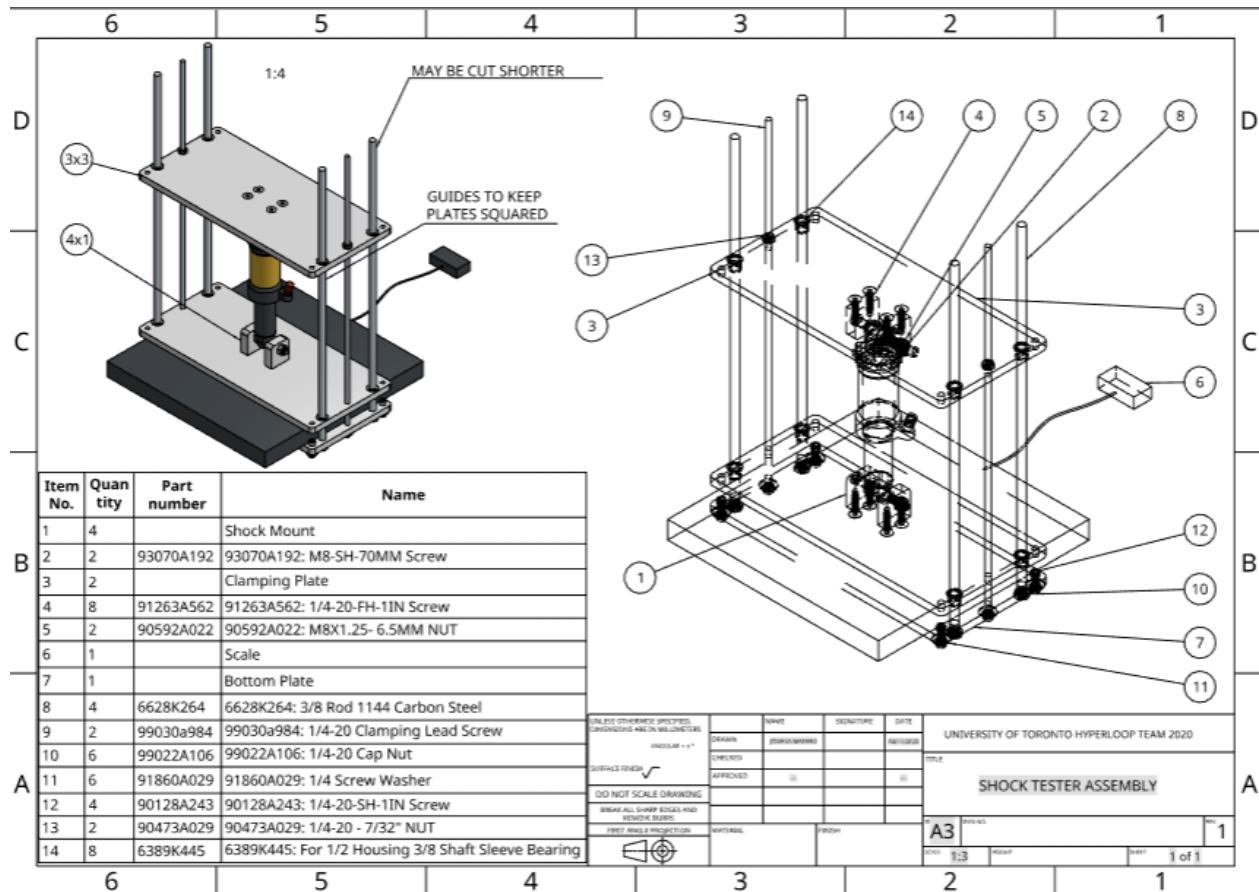
Additionally, a non-structural failure of the carrier would involve undesired movement of the carrier while being hoisted or moved by the engine carrier, this is mitigated by having three team members operating the carrier and engine hoist at the same time, so that the speed and direction of the engine hoist is controlled, and the carrier is prevented from swinging due to changes in direction and acceleration.

### **Health and Safety (during manufacturing) - ID11**

Manufacturing involves drilling of steel, which is dangerous without sufficient PPE and training. As such, team members involved in the manufacturing of the carrier will be trained in drilling steel, and will wear eye protection and gloves during the drilling process. Lifting of the carrier I-beam alone will only be performed by a team of six individuals, and after the rectangular rods are attached to the carrier, the carrier will only be manipulated by the use of the engine hoist and nylon straps.

## Appendix ST - Stability

### **ST1: Assembly drawing of the shock testing device**



## Appendix E - Electronics

### **E1: Primary and Auxiliary Sensors**

#### **E1.1 Primary Sensors**

##### **Temperature Sensor - ProSense RTD0100-06-010-H<sup>18</sup> (x3)**

###### **Inputs:**

Power: 20-32 VDC

Communication: 4 Pin M12 Disconnect

**Output:** 4-20 mA directly proportional to the reading over the range of -40 to 300 degrees Fahrenheit

###### **Dimensions:**

ProSense RTD0100-06-010-H: length 160 mm, diameter 6 mm

ProSense TTD-20-N40300F-H<sup>19</sup> (transmitter): 51.9 x 25.0 x 18.0 mm

ProSense RTDTW-06-010-50N<sup>20</sup> (thermowell): length 113 mm, diameter 9 mm

###### **Function:**

Temperature readings of critical components to ensure the pod is running under expected conditions. This will be used to determine the “health” of the pod and may influence critical operations, e.g. reducing motor power or emergency stopping if the motor overheats.

There will be three primary temperature sensors on the pod. The first one will be located near the front and will monitor temperature of the front wheel. The second will be in the center of the pod. This is where a bulk of the electronics are located including the master computer and pressure sensors. Most importantly this centrally located temperature sensor will externally monitor the batteries for abnormal temperature spikes. The last sensor will be located near the back wheel.



Figure E1.1.1: ProSense RTD0100-06-010-H

<sup>18</sup>Cdn.automationdirect.com, 2020. [Online]. Available:  
<https://cdn.automationdirect.com/static/specs/psrtddprobem12cc.pdf>. [Accessed: 13- Dec- 2020]

<sup>19</sup>Cdn.automationdirect.com, 2020. [Online]. Available:  
<https://cdn.automationdirect.com/static/specs/psttdtemptrans.pdf>. [Accessed: 13- Dec- 2020]

<sup>20</sup>Cdn.automationdirect.com, 2020. [Online]. Available:  
<https://cdn.automationdirect.com/static/specs/psthermowellm12cc.pdf>. [Accessed: 13- Dec- 2020]

**Accessories Used:**

1. Prosense isolated temperature transmitter
  - This component was recommended by the distributor to reduce noise measured by the temperature sensor due to its proximity to high voltage electrical components such as the motor and inverter. All 3 temperature sensors will be equipped with this isolation device.



Figure E1.1.2: ProSense TTD-20-N40300F-H

2. Prosense Thermowell 113mm
  - Cylindrical fitting used to protect temperature sensors. This thermowell will allow the sensor inside to withstand any acceleration or shock forces.



Figure E1.1.3: ProSense RTDTW-06-010-50N

**IMU - VectorNav (INS) VN200 Rugged<sup>21</sup> (x1)****Inputs:**

Power: 3.3 VDC, 80 mA draw @ 5 VDC. Max power consumption 220 mW  
Communication: Serial TTL

**Outputs:** Euler angles (Yaw, pitch roll), acceleration, angular rate, magnetic field, pressure filtering

**Dimensions:**

VectorNav VN200 Rugged: 33.9 x 35.9 x 9.5 mm  
M80-4861005 (connector): 12.70 x 20.00 x 6.65 mm

<sup>21</sup> "VN-200 GNSS/INS". *VECTORMAN*. [Online]. Available: [https://www.vectornav.com/docs/default-source/datasheets/vn-200-datasheet.pdf?sfvrsn=e1a7b2a0\\_5](https://www.vectornav.com/docs/default-source/datasheets/vn-200-datasheet.pdf?sfvrsn=e1a7b2a0_5). [Accessed: 13-Dec-2020].

**Function:**

Industrial grade 3 axis accelerometer, gyro, and magnetometer along with a pressure sensor and 32-bit processor. This will be used for navigation of the pod and will have an associated control system. Equipped with a 50-channel L1 GPS receiver and Kalman filtering algorithms to account for drift (integration errors) when calculating position of pod. This sensor will act as the primary location device of the pod. We will rely on the IMU data about the pod's speed, acceleration, and initial position to make decisions on accelerating and braking. In the presence of light strips similar to the SpaceX Hyperloop Competition, the light strips will be used in addition to the GPS receiver to account for drift.

**Notes:**

In the event that the pod is required to operate in a tube, additional tests will need to be done to ensure that the GPS module receives adequate signal for drift correction. If the GPS signal is weak or unusable, this device can be used as an IMU and will no longer act as the primary distance measuring sensor for the pod. Additionally, the mount for the IMU will include vibration dampening springs to minimize noise from any vibrations on the pod.



Figure E1.1.4: VectorNav VN 200 Rugged

**Accessories Used:**

1. M80-4861005 Connector
  - This is the mating connector required for serial communication with the VN200. The pins split off into wires bridging the connection between the Arduino Due and the sensor.



Figure E1.1.5: M80-4861005 Connector

<b>DC Hall Current Sensor - AcuAMP DCT500-42-24-F<sup>22</sup> (x1)</b>
<b>Inputs:</b> Power: 24 VDC
<b>Output:</b> 4-20 mA directly proportional to the reading over the range of 0 to 500 Amps DC
<b>Dimensions:</b> AcuAMP DCT500-42-24-F: 104.9 x 76.2 x 82.6 mm
<b>Function:</b> This sensor will read the DC current from the high voltage system before it is fed to the inverter. Reading current here allows the team to monitor values for safety and assess the output capabilities of the battery pack. This sensor is able to read up to 500 Amps and will produce an output of 4-20 mA that is proportional to its readings.

Figure E1.1.6: AcuAMP DCT500-42-24-F

<b>Pressure Sensors:</b> <b>SPT25-20-0200A<sup>23</sup> (x2)</b> <b>SPTD25-20-1000H<sup>24</sup> (x1)</b>
<b>Inputs:</b> Power: 8.5 - 36 VDC Communication: M12 quick-disconnect electrical connection
<b>Output:</b> 4 - 24 mA analog directly proportional to the pressure reading range
<b>Dimensions:</b> SPT25-20-0200A: length 49.9 mm, diameter 27.0 mm SPTD25-20-1000H: length 66.9 mm, diameter 18.9 mm

<sup>22</sup> “DCT Series DC Current Transducers”. *ACUAMP*. [Online]. Available:

<https://cdn.automationdirect.com/static/specs/acuampdct.pdf>. [Accessed: 13-Dec-2020].

<sup>23</sup> “SPT25 Series Pressure Transmitters”. *ProSense*. [Online]. Available:

<https://cdn.automationdirect.com/static/specs/prosensespt25transmitters.pdf>. [Accessed: 13-Dec-2020].

<sup>24</sup> “SPTD25 Series Pressure Transmitters”. *ProSense*. [Online]. Available:

<https://cdn.automationdirect.com/static/specs/prosensesptd25transmitters.pdf>. [Accessed: 13-Dec-2020].

**Function:**

These sensors will read the pressures of the pneumatic braking system. Reading current here allows the team to monitor values for safety and assess the status of the braking system. The pod's internals will not be pressurized. The sensors range limitations within the braking system are listed below:

- 4 sensors that can measure 0 to 200 psig for the individual braking units
- 1 sensor that can measure 0 to 1000 psig for the main reserve tank



Figure E1.1.7: SPT25-20-200A



Figure E1.1.8: SPTD25-20-1000H

**Gap Height Sensor - Wenglor Photoelectric OPT2002<sup>25</sup> (x1)****Inputs:**

Power: 18-30 VDC

Communication: 8 pin M12 Disconnect

**Output:** 4-20 mA directly proportional to the reading over the range of 30 to 80 mm

**Dimensions:**

Wenglor Photoelectric OPT2002: 50 x 20 x 50 mm

**Function:**

A point of data collection during flight is to check vertical movement of the pod. Having a large vertical travel distance can indicate that either the track is uneven, the suspension needs tuning, or could also be due to loose components that are moving around. This distance sensor will measure the gap between the bottom of the pod's frame to the top of the I beam. There

<sup>25</sup> "OPT Short Range (CMOS) Series Photoelectric Sensors". [Online]. Available: <https://cdn.automationdirect.com/static/specs/perectoptcmos.pdf>. [Accessed: 13-Dec-2020]

will be a control system associated with this sensor and its readings will be used to tune the pod and emergency stop the pod.



Figure E1.1.9: Wenglor Photoelectric OPT2002

### Inverter Controller & Sensor - Serial CAN Bus Module

<https://docs.longan-labs.cc/1030001/>

#### Inputs:

Power: 5 VDC

Communication: UART

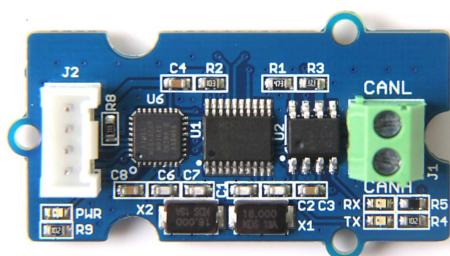
**Output:** Measurements of the inverter (CAN Messages)

#### Dimensions:

Serial CAN Bus Module: 40mm x 20mm (L x W)

#### Function:

The CANBUS module establishes communication between the microcontrollers and inverter/motor. This allows us to control the torque of the motor by passing START or OFF state signals to the inverter. In order to monitor the health of the motor during its operation, sensor values from the motor such as the temperature and current will be collected through this bus.



<https://www.antratek.com/serial-can-bus-module-with-mcp2551-and-mcp2515>

Figure E1.1.10: Serial CAN Bus Module

## E1.2 Auxiliary Sensors

### Telemetry IMU - MPU9250 Module<sup>26</sup> (x1)

#### Inputs:

Power: 3-5 VDC

Communication: I2C

**Outputs:** 9 axis measurements - accelerometer, magnetometer, gyroscope

#### Dimensions:

MPU9250 Module: 25.5 x 15.4 mm (L x W)

#### Function:

Secondary IMU - 3 axis accelerometer, gyro, and magnetometer will be used for vibration profiling for data collection. The electronics team currently plans to use this as an auxiliary sensor and not for real time decision making. By analyzing the data obtained from this sensor the team will be able to improve upon any stability issues and excessive vibrations. Mitigating vibrations is essential to ensure that readings from other sensitive sensors are as accurate as possible. In the future, the team may use the vibration data for possible emergency stop decision making.

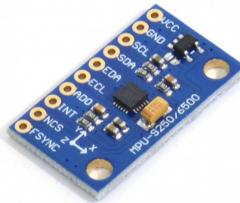


Figure E1.2.1: MPU9250 Module

### Temperature Sensor - TMP006 Contactless IR Temperature Sensor<sup>27</sup> (x1)

#### Inputs:

Power: 3-5 VDC

Communication: I2C

<sup>26</sup> "MPU-9250 Register Map and Descriptions Revision 1.6". *InvenSense*. [Online]. Available: <https://invensense.tdk.com/wp-content/uploads/2015/02/RM-MPU-9250A-00-v1.6.pdf>. [Accessed: 13-Dec-2020].

<sup>27</sup> "TMP006 User Guide". *Texas Instruments*. [Online]. Available: <https://cdn.sparkfun.com/datasheets/Sensors/Temp/sbou107.pdf>. [Accessed: 13-Dec-2020].

**Output:** Temperature reading of over the range of -40 to 125 degrees Celsius

**Dimensions:**

TMP006 Module: 20.32 x 20.32 mm (L x W)

**Function:**

This secondary temperature sensor will not be connected to control systems, but rather will be used for analysis to observe zones on the pod that may get too hot during flight.

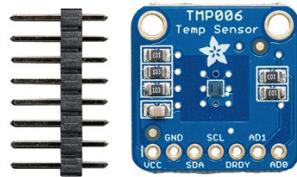


Figure E1.2.2: TMP006 Contactless IR Temperature Sensor

## E2: Microcontrollers & Master Computer

### **Microcontrollers - Arduino Due<sup>28</sup> (x4), Industrial Arduino (x1)**

#### **Inputs:**

Power: 7-12V input, 3.3V operating

#### **Output:**

54 I/O pins to read various signal values from sensors

#### **Dimensions:**

Arduino Due: 101.52 x 53.3 mm (L x W)

Industrial Arduino: variable dimensions

#### **Function:**

Microcontrollers will be spread out throughout the pod in order to connect to sensors in a localized manner. Each microcontroller will connect back to the master computer to ensure synchronization. Sensors will use the arduino to relay outputs. Power to the sensors will not be drawn from the microcontrollers, but rather the low voltage power system managed by the Energy Systems Team. This will prevent excessive load on any single microcontroller.

The industrial arduino provides a robust and modular approach to microcontroller use. Functionality wise it will operate identical to the arduino dues, making use of the GPIO pins to communicate with sensors. The difference with this arduino is that it is rugged and shock proof, making it ideal for an application like hyperloop. It has various modules like ethernet shields, power supplies, and GPIO, making any future expansions simple. This new industrial arduino by AutomationDirect has an immense amount of applications and will serve as a key component for the electronics onboard.



<sup>28</sup> “Arduino Due,” *Arduino Due | Arduino Official Store*. [Online]. Available: <https://store.arduino.cc/usa/due>. [Accessed: 13-Dec-2020].

Figure E2.1: Arduino Due

Figure E2.2: Industrial Arduino with its power supply

### Master Computer - Odroid XU4<sup>29</sup>

#### Inputs:

Power: 5V 4A power supply

#### Output: N/A

#### Dimensions:

Odroid XU4: 83 x 58 x 20 mm

#### Function:

Centralized computer connected to the peripheral arduinos. Carries out the centralized computing and interfaces with the other microcontrollers. Multiple processes carried out at once such as radio communication as well as processing sensor requests.

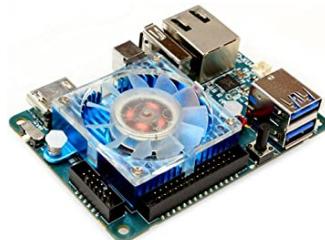


Figure E2.3: Odroid XU4

### Ethernet Shield - W5500 Ethernet Shield<sup>30</sup> (x3)

#### Inputs:

Power: 3.3V/5V from Arduino header pins

Communication: RJ-45 Ethernet connector with Transformer; SD-Card Slot; Serial Peripheral Interface with Arduino header pins

**Output:** Sensor data using Transmission Control Protocol(TCP) from RJ-45 Connector to Odroid master computer

<sup>29</sup> Magazine.odroid.com, 2020. [Online]. Available:

<https://magazine.odroid.com/wp-content/uploads/odroid-xu4-user-manual.pdf>. [Accessed: 13- Dec- 2020]

<sup>30</sup> “W5500 Ethernet Shield,” *W5500 Ethernet Shield Document Wiki*, 27-Feb-2018. [Online]. Available:

[http://wizwiki.net/wiki/doku.php/osh:w5500\\_ethernet\\_shield:start](http://wizwiki.net/wiki/doku.php/osh:w5500_ethernet_shield:start). [Accessed: 10-Mar-2021].

**Dimensions:** 68.58 x 53.31 mm

**Function:**

The W5500 Ethernet Shield will be used in conjunction with the Arduino Due to communicate with the Odroid master computer. Adding this shield in between the arduino and sensor PCBs will allow for an ethernet cable to easily be routed to the main computer to send all sensor information. Because of its shield design, connections can still be made through the Ethernet shield from the custom PCBs to the Arduino, as long as we are not using the same communication interface (SPI for this ethernet shield).

The Odroid XU4 already has an ethernet interface so connecting this shield should only require a basic ethernet cable. To use the ethernet shield in conjunction with the Arduino Due, a preexisting Ethernet 2 library will be installed and used.

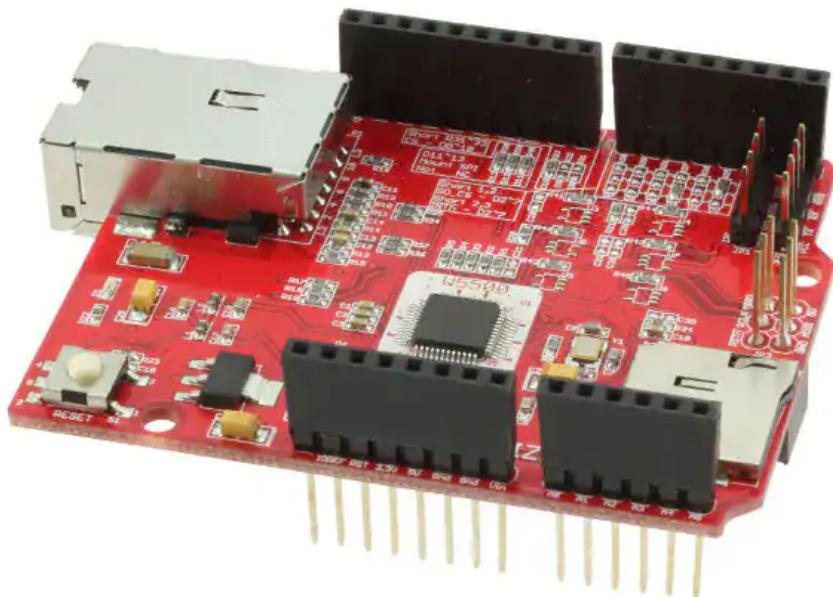
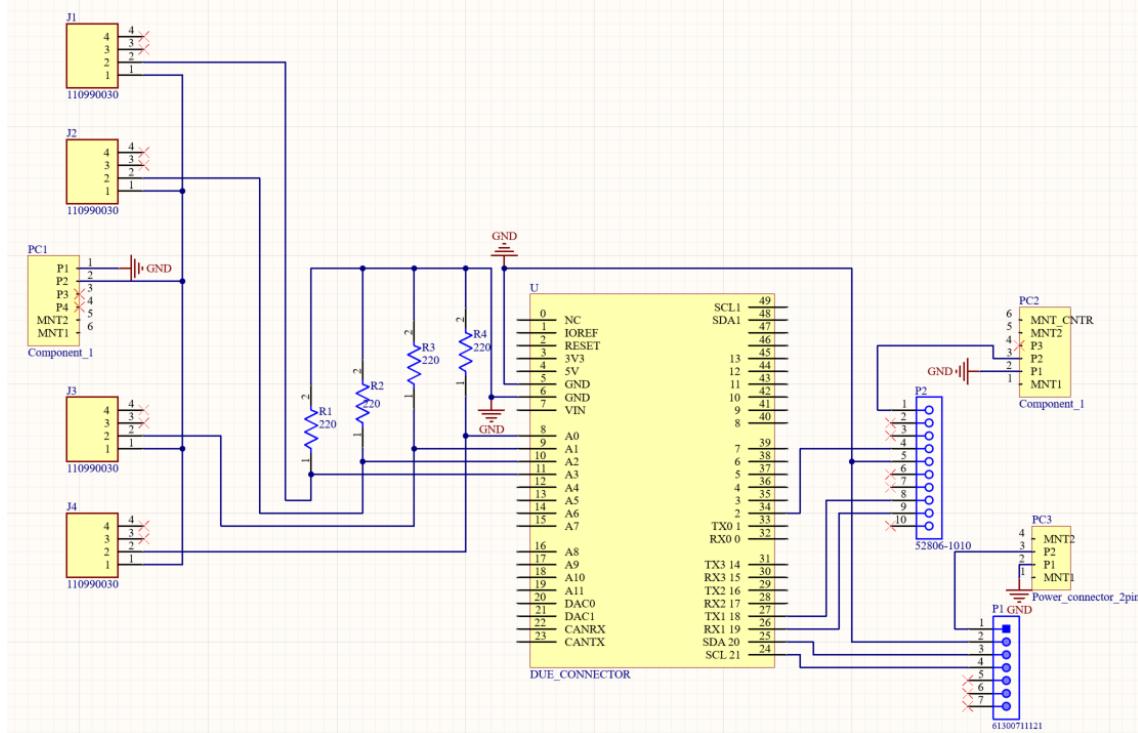


Figure E2.1: W5500 Ethernet Shield

## E3: Complete Wiring Schematics

### E3.1 Printed Circuit Board #1 Wiring Schematic



### E3.3 Printed Circuit Board #3 Wiring Schematic

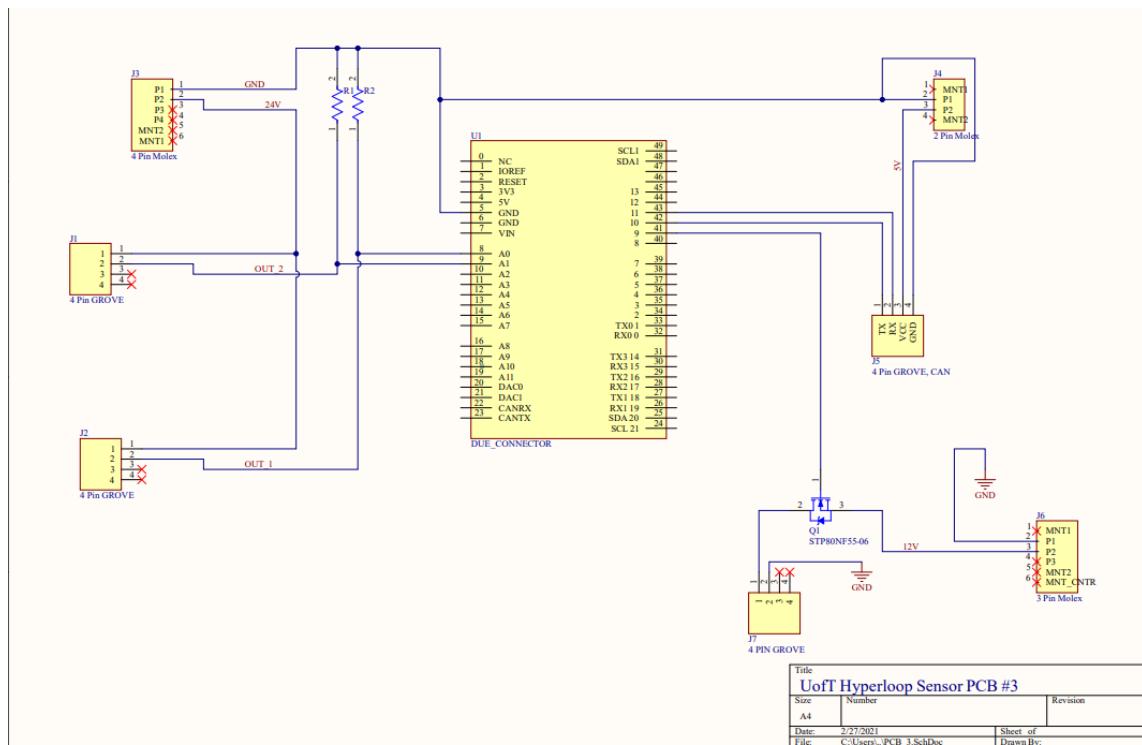
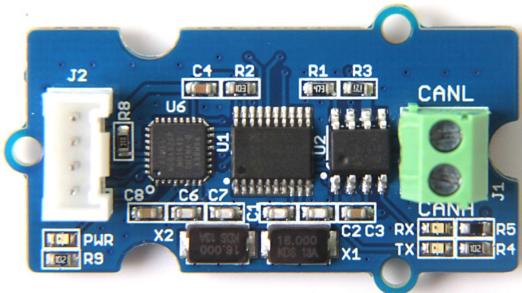


Figure E3.3: Schematic for PCB#3

#### **E4: Serial CAN Bus Module Interface with PM100DX**



<https://www.antratek.com/serial-can-bus-module-with-mcp2551-and-mcp2515>

In order to control torque, voltage, send signals, and read data from the PM100DX Inverter of the pod, the Serial CAN Bus Module was chosen as the shield to connect to an Arduino Due. It supports the CAN 2.0B protocol that uses 29 bit identifiers, thus capable of transmitting higher bit messages at fast speeds of up to 1Mbps<sup>31</sup>. This particular module was also chosen due to its small/compact footprint of 40 x 20 mm, allowing for more economical use of the pod's mounting space.

- Serial CAN Bus Components:
  - 4 pin 2.0mm Grove Connector (Vcc, GND, TX, RX)
  - CAN L, CAN H screw terminal
  - Power, TX, RX LEDs
  - MCP2515, MCP2551 (CAN Controllers)
  - ATMEGA328P (MCU)

The Serial CAN Bus Module uses UART pins for communication with the Arduino Due (TX/RX), and CAN L (Low) and CAN H (High) for communication between a transmitter (the PM100DX) and receiver node<sup>32</sup>. The CAN Bus Module's CAN H and CAN L will be wired to the PM100 Inverter's CAN A High and CAN A Low pins, creating the CANbus.

#### **Module Wiring to Arduino**

GND (CAN)-->GND (Arduino)

5V/VCC (CAN)---->3.3V/5.0V (Arduino)

<sup>31</sup> "MCP2515 CAN Bus Interface Module - ProtoSupplies", ProtoSupplies, 2020. [Online]. Available: <https://protosupplies.com/product/mcp2515-can-bus-interface-module/>. [Accessed: 13- Dec- 2020]

<sup>32</sup> "Arduino MCP2515 CAN Bus Interface Tutorial - CAN Protocol", Electronics Hub, 2020. [Online]. Available: <https://www.electronicshub.org/arduino-mcp2515-can-bus-tutorial/>. [Accessed: 13- Dec- 2020]

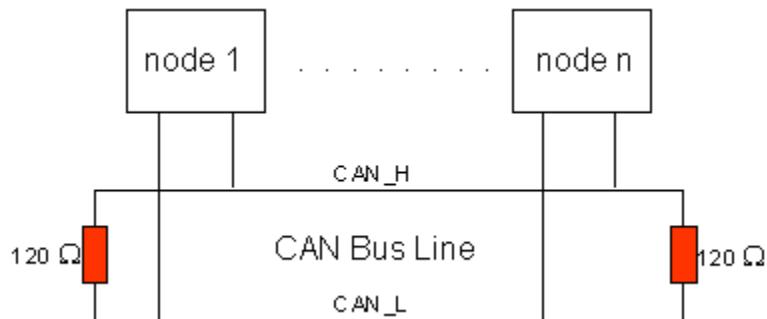
RX (CAN)---->Digital Pin 11 (Arduino)

TX (CAN)---->Digital Pin 10 (Arduino)

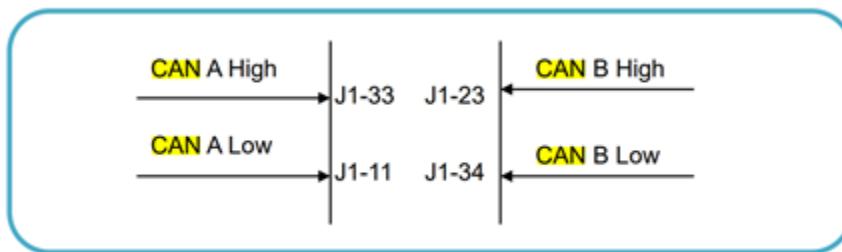
CANL (CAN Module 1)-->CANL (CAN Module 2)

CANH (CAN Module 1)-->CANH (CAN Module 2)

**Also, 120  $\Omega$  resistor must be placed between CANL & CANH wires of the module. Thus, solder P1 on the Serial CAN Bus Module to activate the internal 120 $\Omega$  resistor. The PM100DX already has this 120 $\Omega$  resistance built in for itself.**



<https://infosys.beckhoff.com/english.php?content=../content/1033/e16751/2521746827.html&id=>



CANbus pins on the PM100DX Inverter <sup>33</sup>

On the PM100DX, we will need to configure its CAN parameters to set up the CAN message bits for proper bit indexing for data extraction. This will be done during assembly and testing of the pod, through the Arduino Due. The inverter also includes CAN diagnostic parameters that displays any warning/error statuses. These warning messages will be immediately transmitted to the Arduino Due (through the Serial CAN Module) where CAN messages will be received every 1-1.5 seconds.

<sup>33</sup> Wiki.neweagle.net, 2020. [Online]. Available:

[https://wiki.neweagle.net/docs/Rinehart/PM100\\_User\\_Manual\\_3\\_2011.pdf](https://wiki.neweagle.net/docs/Rinehart/PM100_User_Manual_3_2011.pdf). [Accessed: 13- Dec- 2020]

```
03:57:13.174 -> 00000000GET DATA FROM ID: 0
03:57:13.817 -> 0x1      0x1      0x1      0x1      0x6      0x6      0x6      0x6
03:57:13.887 -> 00000000GET DATA FROM ID: 0
03:57:14.534 -> 0x1      0x1      0x1      0x1      0x6      0x6      0x6      0x6
03:57:14.569 -> 0000GET DATA FROM ID: 0
03:57:14.939 -> 0x1      0x1      0x1      0x1      0x6      0x6      0x6      0x6
03:57:14.974 -> GET DATA FROM ID: 0
03:57:15.074 -> 0x1      0x1      0x1      0x1      0x6      0x6      0x6      0x6
```

## Sample CAN Messages received by Serial CAN Bus Module, Initial Testing

The PM100 broadcasts the following messages<sup>34</sup>:

Address	Frequency	Content	CAN Active Messages (Low Word)
0x0A0	Slow/10 Hz	Temperatures #1	0x0001
0x0A1	Slow/10 Hz	Temperatures #2	0x0002
0x0A2	Slow/10 Hz	Temperatures #3	0x0004
0x0A3	Fast/100 Hz	Analog Inputs Voltages	0x0008
0x0A4	Fast/100 Hz	Digital Input Status	0x0010
0x0A5	Fast/100 Hz	Motor Position Information	0x0020
0x0A6	Fast/100 Hz	Current Information	0x0040
0x0A7	Fast/100 Hz	Voltage Information	0x0080
0x0A8	Fast/100 Hz	Flux Information	0x0100
0x0A9	Slow/10 Hz	Internal Voltages	0x0200
0x0AA	Fast/100 Hz	Internal States	0x0400
0x0AB	Fast/100 Hz	Fault Codes	0x0800
0x0AC	Fast/100 Hz	Torque & Timer Information	0x1000

The Serial CAN Bus Module allows for easy readability and indexing of these CAN messages when connected to the Arduino Due, which will be done using the SoftwareSerial.h and Serial\_CAN\_Module.h files in our Arduino software. These libraries give us the necessary functions to analyse and extract CANbus messages transmitted through the Serial CAN Bus Module. This makes it easier to create a robust CANbus communication system with the inverter, and it is easy to test its functionality.

As soon as the PM100 is turned on, CAN frames are sent indicating its status and measurements, which includes torque, temperature, voltage and current.

The Serial CAN Bus Module code will be uploaded onto its Arduino Due. It consists of calling the `recv()` function in a continuous loop so that all CAN frames from the PM100 will be received and printed on the Serial Console. In addition, the CAN mask/filter will be set using `can.setFilt()` and `can.setMask()`. As the Pod turns on and operates, the CAN Bus Module will be used to observe the PM100's measurements. Later it is necessary to accelerate/decelerate the PM100

<sup>34</sup> Wiki.neweagle.net, 2020. [Online]. Available: [https://wiki.neweagle.net/docs/Rinehart/PM100\\_User\\_Manual\\_3\\_2011.pdf](https://wiki.neweagle.net/docs/Rinehart/PM100_User_Manual_3_2011.pdf). [Accessed: 13- Dec- 2020]

inverter. This is done by using the send() CAN function that will transmit our messages instructing it to do so.

### Safety Risks:

The features of the Serial CAN Bus Module that incorporate the highest safety risks are:

- The operating requirement of 5V/3.3V, this voltage will be either supplied by the Arduino Due or an external power source. The wire connection for VCC and GND must be stable, otherwise there is risk of heating and fire.
- Also, the CAN H & CAN L wires pose a safety risk, as they transmit CAN frames using voltages up to 5.0V. Although this is a low voltage, improper wiring poses further heating and fire risks. These wire connections must be secure, and appropriate 24 AWG must be used to connect to the PM100
- Supplying the Serial CAN Bus Module with voltage greater than 5.0V poses a risk of burning/fire

The risks above are treated and accounted for by properly assembling/soldering the Serial CAN Bus Module onto its respective PCB. The PCB is modeled to only supply the module with a maximum of 5.0V, and no other power source will be connected to it. Thus eliminating the risk of improper power supply. Next, the CAN Bus wiring will be connected meticulously to the PM100, and secured precisely so that nothing comes undone.

- Avoiding Engineering Safety Risks
  - All PCBs and modules were designed to be supplied correct voltages that are lower/equal to their maximum.
  - Wiring of the module is designed to avoid clutter
- System Testing Procedures:
  - Send/Recv testing done between 2 Serial CAN Bus Modules, along with mask/filter function. Successful
  - Next, testing of Serial CAN module connected to PM100: check if Power LED turns on, check if TX/RX LEDs are flashing one PM100 is powered on
  - Open Arduino Serial Console and check CAN messages being received: should continuously print out CAN messages (Received CAN mssg from ID:X, followed by hexadecimal frame bits)
  - Check if Baud rate of 9600 is correctly set on the Serial CAN Bus Module, OK/FAIL message will be printed right away
  - CAN Mask/Filter OK/FAIL check will be printed as well

- Next, send a CAN message to PM100 to accelerate, observe if torque changes.  
Same with decelerating, check if PM100 torque changes.

### Safety Mitigation and Testing Procedures:

- Major failure mitigation strategies:
  - Complete a visual check of the Serial CAN Bus Module in the Pod, make sure power LED is ON, and TX/RX LEDs are flashing (messages are transmitting)
  - Observe if CAN messages are appearing on the Arduino's Serial Console, this must be done before start
  - Properly connect Vcc/GND to the Serial CAN Bus module, and the CAN Bus wires between the module and PM100. If any faulty connection, communication is gone
- Tests, Protocols and Procedures to ensure safety
  - During Pod assembly, do not provide more than 5V to Serial CAN Bus Module. This will burn and destroy the MCP2515 chip on the module, thus rendering it defective.
  - Connect CAN Bus wires to proper ports on the PM100, this inverter operates under much higher voltage/current than the Serial CAN Bus module, and improper connection can destroy the PM100 AND/OR the CAN Bus Module
- Safety considerations during assembly and integration
  - Do not connect the Serial CAN Bus Module to its PCB while it is powered, this can damage its chips
  - Similarly, do not connect the CAN bus wires if the PM100 is powered on, this can damage the module and inverter

## **E5: PM100DX Inverter Startup Sequence<sup>35</sup>**

Initially, the inverter starts at VSM\_State 0 when it receives a 12V power. More details on the inverter VSM\_State can be found in Table EPS1. The inverter has an internal state machine that steps through a series of actions at startup, at shutdown, and generally whenever operations “transition” from one mode or state to another. Table EPS1 is a list of the states that the inverter sequences through upon the application of power.

**Table EPS1: Table of inverter vehicle state machine**

VSM_State	Name
0	Start State
1	Pre-charge sequence initial state - Turn on the pre-charge relay
2	Pre-charge sequence active state - Waiting for the capacitor to finish charging
3	Pre-charge sequence finish state – Completes the final checks before proceeding to Wait State.
4	Wait State – waiting for activation of forward or reverse.
5	Ready State – Activates the inverter state machine to begin energizing the motor
6	Motor Running State – Normal motor running
7	Fault State – The controller has faulted
0xFFFF	Recycle Power State – This indicates that the power to the controller needs to be recycled after EEPROM Programming is complete.

The startup sequence for the inverter goes as follows:

1. VSM\_State = 0
  - a. Receive 12V power
  - b. Perform Power On Self Test (POST)
    - i. Several tests are performed in this state. Each test has an associated fault flag. The following is a list of some parameters checked:
      1. Current Sensor Offset
      2. Accel
      3. T\_PCB

---

<sup>35</sup> Wiki.neweagle.net, 2020. [Online]. Available:

[https://wiki.neweagle.net/docs/Rinehart/PM100\\_User\\_Manual\\_3\\_2011.pdf](https://wiki.neweagle.net/docs/Rinehart/PM100_User_Manual_3_2011.pdf) [Accessed: 13- Dec- 2020]

- ii. If a fault occurs, then the inverter reports back the fault through the `post_fault_hi` and `post_fault_lo` variables with the DSPGUI. If more than one fault has occurred, the number indicated will be an AND of the two faults. A full table of error faults and associated flags can be found in the PM100DX Inverter Manual.
- c. If no error flags are returned in POST, proceed to `VSM_State 1`
- 2. `VSM_State = 1`
  - a. Pre-charge sequence initial state - Turn on the pre-charge relay
  - b. Proceed to next `VSM_State 2`
- 3. `VSM_State = 2`
  - a. Pre-charge sequence active state - Waiting for the capacitor to finish charging
  - b. Proceed to next `VSM_State 3`
- 4. `VSM_State = 3`
  - a. Pre-charge sequence finish state - Completes the final checks:
    - i. If all the following are TRUE:
      - 1. `Vdc` stops rising by less than `PRECHARGE_RATE`
      - 2. `Vdc` is greater than `VIN_MIN`, `Vdc` is less than `VIN_MAX`
      - 3. Time has not exceeded 1 second
    - ii. Then do the following:
      - 1. Engage the `MAIN_OUT` output
      - 2. Turn-off the `PRE_CHG` output
      - 3. Go to the 30ms delay
  - b. Step 'a' should have taken 1s, if the total pre-charge time exceeds 1 second, then do the following checks:
    - i. If the DC BUS is greater than `VIN_MAX`:
      - 1. Turn the `PRE_CHG` output off
      - 2. Declare `VDC_OOR_HI` fault
    - ii. If the DC BUS is less than `VIN_MIN`:
      - 1. Turn the `PRE_CHG` output off
      - 2. Declare `VDC_OOR_LOW` fault
    - iii. If the `Vdc` is still rising by more than or equal to `PRECHARGE_RATE`
      - 1. Turn off the `PRECHARGE_OUT` output
      - 2. Declare `PRECHARGE_TIMEOUT` fault
  - c. Delay 30 ms
  - d. Measure `Vdc` and check:
    - i. If `Vdc` is above `VIN_MAX`, declare `VDC_OOR_HI` fault
    - ii. If `VDC` is below `VIN_MIN`, declare `VDC_OOR_LO` fault
  - e. Verify that voltage is within `VDC_MATCH_RANGE_THRESHOLD` of the first measurement. If not, declare `VDC_DATA_MISMATCH` fault.
  - f. Proceed to next `VSM_State 4`

5. VSM\_State = 4 (Wait State)
  - a. Check Key Switch Mode, based on that value, inverter can be powered to run motor
  - b. If Key Switch Mode 0
    - i. Simple on/off ignition switch functionality
    - ii. Turn ignition to ON position
    - iii. Check if brake switch is active AND ONLY one of /FORWARD or /REVERSE
  - c. If Key Switch Mode 1
    - i. Traditional switch functionality
    - ii. Turn ignition to ON position
    - iii. Check if brake switch is active
    - iv. Check off start signal pulse has been received
    - v. While keeping brakes on, only one of /FORWARD or /REVERSE needs to be activated
  - d. Proceed to next VSM\_State 5
6. VSM\_State = 5
  - a. The READY state sends out Enable Inverter Command
  - b. Wait for Inverter Ready Flag to be set
  - c. If no response within a specific amount of time:
    - i. Declare an INVERTER\_RESPONSE\_TIMEOUT\_FAULT
  - d. State automatically transitions to next state if no faults, VSM\_State 6
7. VSM\_State = 6
  - a. The motor running state
  - b. This is the normal running operation of the torque commanded field oriented controller that rides on top of the power electronics hardware. While running the drive can be switched from torque command mode to speed command mode. It may be exercised within the full operating envelope of the machine / drive combination.

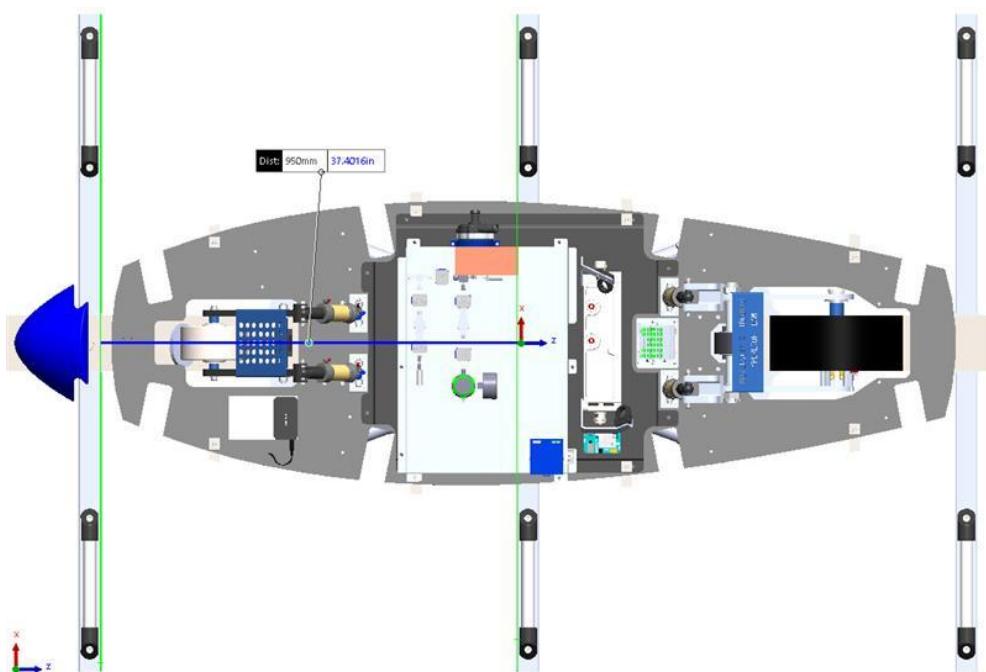
On top of the VSM\_State, the inverter has another set of built in state machines known as Inverter States and is shown by the variable inv\_mode. Most of the motor running modes in VSM\_State=6 can be accessed from this variable. A full list and description can be found in Table EPS2.

**Table EPS2: Inverter States Table**

Inverter States (inv_mode)	Description
0	Precharge, power-up state
1	Stop - Inverter is not running and is in “STOP” state.
2	Open Loop State - for testing purposes
3	Closed Loop state – normal state
4	Start Time Delay – small delay before starting the inverter
5	Current Sensor Test – flux ramp and flux regulators enabled
6	Closed Loop Torque – torque regulator is enabled
7	Torque Ramp – start torque ramp
8	Idle Run – inverter running normally
9	Idle Stop – inverter is stopped
10	Ramp Off Torque – ramps down the torque command
11	Ramp Off Flux – ramps down the flux command
12	All Ramps Off – shutoff inverter
15	Default – Stop state

## Appendix S - Software

## **S1: Two dimensional bird's eye view of the pod outlining major components like back and front wheels and sensors**



## **References - Software**

- [SR1] Drumea, Andrei & Popescu, C.. (2004). Finite state machines and their applications in software for industrial control. 25 - 29 vol.1. 10.1109/ISSE.2004.1490370.
- [SR2] Kormanyos, Christopher. Real-Time C++ Efficient Object-Oriented and Template Microcontroller Programming. Heidelberg, 2015.
- [SR3] Cheon, Yoonsik and Leavens, Gary T., "A Simple and Practical Approach to Unit Testing: The JML and JUnit Way" (2001). Computer Science Technical Reports. 181
- [SR4] UnitTest++ [Online]. Available: <https://unittest-cpp.github.io/>. [Accessed: 15-Feb-2021].
- [SR5] Simulink Real-Time. (n.d.). Retrieved February 14, 2021, from <https://www.mathworks.com/products/simulink-real-time.html>