Parametric UUV Design Tool

A Major Qualifying Project (MQP) Report Submitted to the Faculty of

WORCESTER POLYTECHNIC INSTITUTE

In Partial Fulfillment of the Requirements of the Degree of Bachelor of Science

Submitted By:
Joshua Barney, Belkys Felix Nova, Emma Gilroy & Keelan Smith

Project Sponsor:
General Dynamics Electric Boat

Project Advisor:
Professor Ahmet Can Sabuncu

Date Submitted:
April 25, 2024

Table of Contents

List of Figures

List of Tables

List of Acronyms

Abbreviation	Definition
ASW	Anti-Submarine Warfare
AUV	Autonomous Underwater Vehicle
CFRP	Carbon Fiber Reinforced Polymer
CN3	Communication / Navigation Network Node
DVL	Doppler Velocity Log
ESWBS	Expanded Ship Work Breakdown Structure
FLS	Forward-Looking Sonar
GPS	Global Positioning System
GUI	Graphical User Interface
ID	Inspection / Identification
INS	Inertial Navigation System
IO	Information Operations
ISR	Intelligence, Surveillance, and Reconnaissance
LTA	Lighter Than Air
MCM	Mine Countermeasures
ROUV	Remotely Operated Underwater Vehicle
SAS	Synthetic Aperture Sonar
SLS	Side-Looking Sonar
TCS	Time Critical Strike
UUV	Unmanned Underwater Vehicle

1. Introduction

Unmanned Underwater Vehicles (UUVs) have existed for upwards of 60 years, with the first UUV being deployed in 1957 (Gafurov & Klochkov, 2015). Since then, UUV designs have become increasingly complex, as well as in-demand in both the civilian and military space. With these advancements comes a requirement for rapid design to support the changes in the abilities and needs of prospective users. Due to this, our project will focus on creating a design tool that will allow engineers to quickly weigh a variety of desires to find effective UUV design solutions.

UUVs, as the name implies, are underwater vehicles that do not have human pilots in them. There are two main types of UUVs, namely ROUVs and AUVs. ROUV stands for Remotely Operated Underwater Vehicle and is something that is directly piloted by a human, usually from a surface vessel with an attached cable, or from a separate submarine. While these are certainly interesting, and a worthy subject of discussion, for the sake of this paper we will focus primarily on AUVs. AUV stands for Autonomous Underwater Vehicle, and they are vehicles that can be given a set of instructions and carry them out without the need for direct human interaction or monitoring (Nichols et al., 2020).

There are a variety of designs and design considerations for UUVs, with some of the most important being shape and power system. The most common shape is the cylindrical design, which resembles the shape of a submarine. Alternatives include a box-like design that has a relatively rectangular cross-section, an open frame design, and a biometric design. Additionally, the differences in how thrust is created are key. The traditional design with propellers in the back is referred to as a screw-driven design, with alternatives that can harness the changes in buoyancy in water to gain thrust, as well as biometric designs that generates thrust like a fish would, mimicking nature. Since the combination of cylindrical and screw-driven is the most common design, and has the most data, our analysis will focus on designs that have those two characteristics (Puzaia et al., 2016).

Our goal is to create a parametric design tool to assist with the UUV design process. To develop our tool, we will utilize basic engineering principles including understanding relationships between parameters and certain hydrostatic requirements. The tool will allow users to input ranges for various parameters, including length, speed, range, payload capacity, and more, to synthesize viable solutions.

2. Literature Review

2.1. Introduction

Literature directly involving the creation of a UUV design is somewhat sparce. Research was done by each member of the team to collect information regarding the design and development of an Unmanned Underwater Vehicle.

The process of creating a design tool similar to the one we are developing has been documented in several papers. This can be seen in Literature Reviews 1: “Developing a parametric design tool for lighter than air vehicles” and 2: “Development and Validation of a Conceptual Design Program for Unmanned Underwater Vehicles.” These reviews describe the design processes for parametric tools and programs involving lighter than air and unmanned underwater vehicles.

Design and optimization of UUV hull shapes and multi-rotor vehicles is also a subject matter essential to the completion of our tool. Matters relevant to hull design or optimization are covered in Reviews 3: “Parametric design of sailing hull shapes”, and 4: “Parametric Design and Optimization of Multi-Rotor Aerial Vehicles”.

2.2. Developing Parametric Design Tool for Lighter Than Air Vehicles

There are many ways and software to develop a parametric modeling tool. In this conference paper, a tool was created to design a lighter than air (LTA) vehicle to meet its mission requirements. The tool will allow users to easily obtain specifications of the vehicle by inputting certain requirements. To design this tool, Microsoft Excel was used due to the convenience of being easily accessible on Windows systems. Microsoft Excel allows easy user interaction and quick results with minimal coding.

During the development of the tool, a series of testing was involved. A prototype was created utilizing as many functions as possible to allow further testing. Then, if any alterations needed to be made, they were incorporated in the following cycle of development. This allows the tool to frequently improve along with the observation of individual functions.

After defining the exact requirements of the tool, the parameters that will be inputted by the user will need to be determined. Since there are many ways to go about designing an LTA vehicle, a list of inputs and outputs was produced based on the vehicle characteristics that were inputted by the user.

The tool was then divided into four sections based on characteristics of mission requirements. The first section allows the user to input the desired average wind speed, temperature, fitness ratio, and maximum altitude. From this, the related aerodynamic parameters were calculated. These include air density, surface area, drag, and volume. The second section requires users to select the type of engine and the length of the flight. This solves for the amount of fuel needed as well as the horsepower output. The third section dictates the LTA weight based on the payload, envelope weight, fuel, power systems, auxiliary systems, avionics, propulsion systems and communication systems. In addition, fins and rudders, wing power cars, gondola, and rear power cars are other factors that are essential for flight. The last section uses the angle of elevation of the LTA vehicle to calculate the field area.

An additional section was created to only contain user input variables and calculated output variables so that results are obtained faster and in one screen.

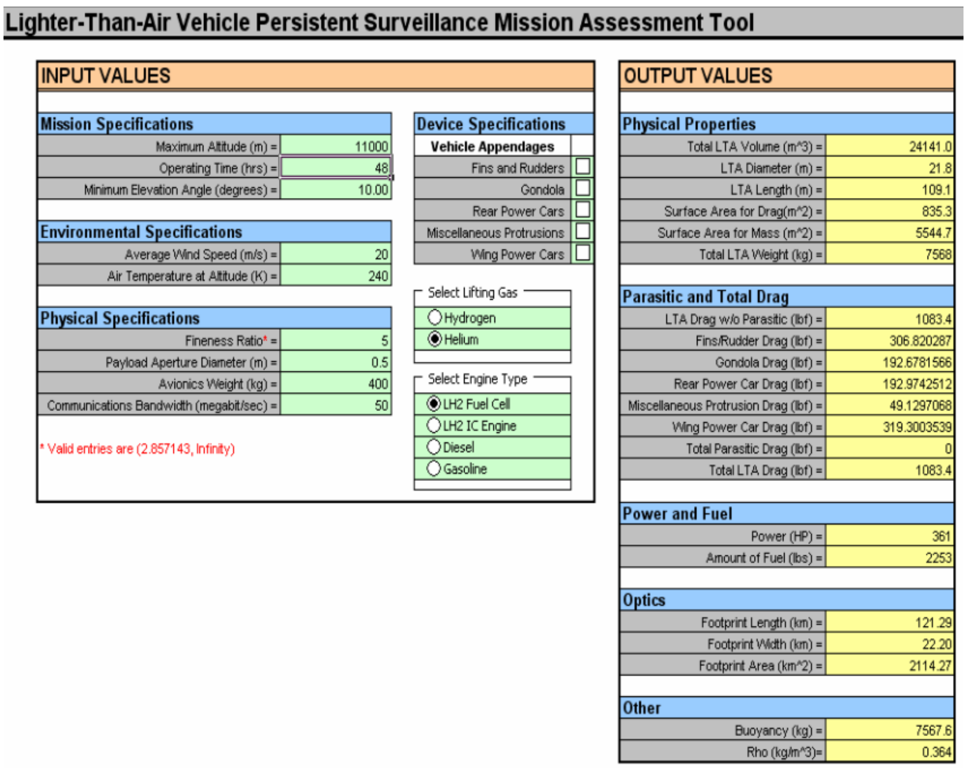

Figure 1. User Interface Based on LTA Design, Performance, and Equipment (Fomin, et al., 2005).

A User Documentation file was produced to supply information on each section including equations and how to assess the given hypothetical mission specifications. The information provided in this file can be utilized to conduct a tradeoff analysis on the cost and performance of an LTA vehicle based on user inputs.

Although the tool is efficient and helpful, there are a few limitations to the tool. The maximum altitude is 11,000 meters and the maximum wind speed is 45 meters per second.

Inputting values higher than the maximum causes some variables to exponentially increase and causes Microsoft Excel to generate an overflow error. This causes the user to be limited with certain LTA vehicle missions as Microsoft Excel will not be able to make calculations if they are passed the maximum. Despite a few limitations, Microsoft Excel is a quick and user-friendly tool to develop a parametric tool for LTA vehicles (Fomin, et al., 2005).

2.3. Development and Validation of a Conceptual Design Program for Unmanned Underwater Vehicles

This thesis also developed a parametric tool to assist with the design process of a UUV. UUVs can assist in numerous missions including Intelligence, Surveillance, and Reconnaissance (ISR), Mine Countermeasures (MCM), Anti-Submarine Warfare (ASW), Inspection / Identification (ID), Oceanography, Communication / Navigation Network Node (CN3), Payload Delivery, Information Operations (IO), and Time Critical Strikes (TCS).

The tool's intention is to focus on where modern technology is lacking in design and development. This parametric tool is expected to retrieve early information to assist in the design and development of a UUV while still being time and cost friendly. Based on the desired mission, the tool will determine the dimensions and performance of the UUV design.

A conceptual design process was created to apply the user requirements for the UUV mission and the related characteristics. The conceptual design helps understand the different factors when developing a UUV including customer specifications and the technical practicability. The process begins with user specifications containing the maximum vessel speed, combatant capability, and vessel endurance. Those specifications are then transformed into parameters related to subsystems of a ship.

The conceptual design is a MATLAB script containing the following modules: mission module, resistance module, hull module, battery module, and pressure vessel module. Each module will assist in creating a UUV design. The user has multiple Microsoft Excel spreadsheets where they will be able to directly input information and the design will be produced from one MATLAB graphic user interface. With that information the tool will provide an Expanded Ship Work Breakdown Structure (ESWBS) spreadsheet, a pdf of a 2D image of a UUV, and multiple files for design. Utilizing both Microsoft Excel and MATLAB allows for a variety of different users to use the tool. A regular user will use the Microsoft Excel sheets and a more advanced user will be able to alter the MATLAB modular script. Below is the design approach for the tool.

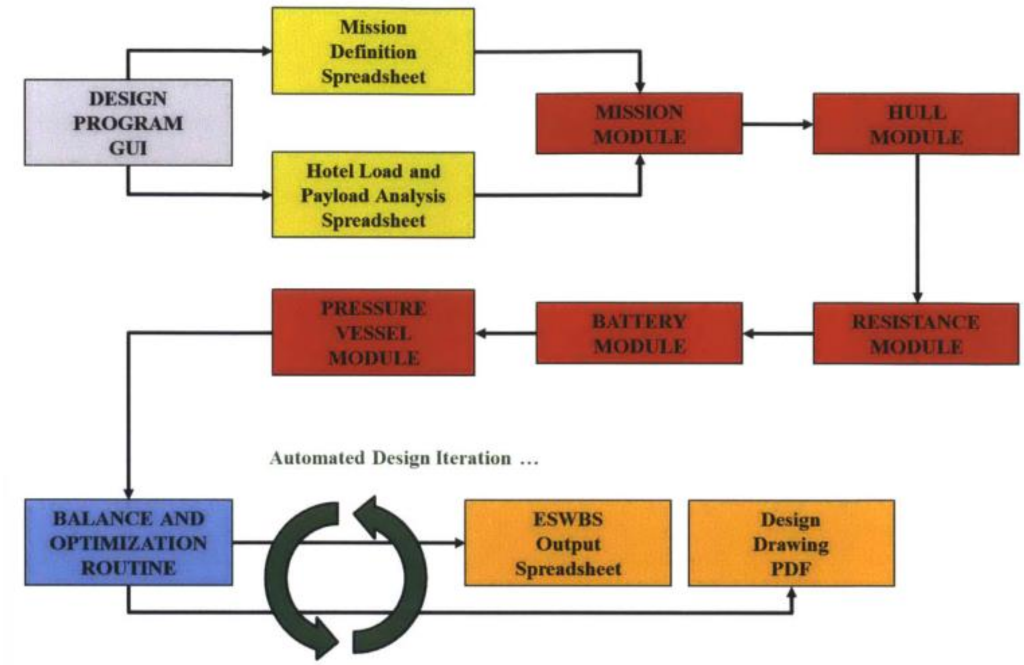

Figure 2. Tool Design Approach (Laun, 2013).

The first module, the mission module, obtains input user data from two Excel spreadsheets, the Mission Definition Spreadsheet and the Hotel Load and Payload Analysis spreadsheet, that are then converted to variables in MATLAB. In the Mission Definition Spreadsheet, the average operating speed and track distance is inputted by the user, as well as the operation type. In the Hotel Load and Payload Analysis spreadsheet, the user defines the desired shipboard quantity for each of the nine components being, global positioning system (GPS); a pressure sensor; a forward-looking sonar (FLS); a side-looking sonar (SLS); a synthetic aperture sonar (SAS); a vehicle controller; an autonomy controller; a payload controller; a variable ballast system; a combined inertial navigation system (INS), and Doppler velocity log (DVL). The user will also have the option to turn any system off if needed. The next module, the hull module, solves the general geometry of the UUV hull. This module also provides a table with offsets and other characteristics including the volume of the envelope, prismatic coefficient, sectional area, and wetted surface area of the UUV hull. In the Resistance module, the total resistance of the UUV system is calculated. The battery module applies the information given from the resistance and mission module to solve the volume and weight of the battery system. The pressure vessel module allows the user to specify the maximum operating depth to then generate a hard body with the maximum allowable pressure.

After all the modules are completed, the next step is the Balance and Optimization routine. All previous modules are applied to produce the ESWBS spreadsheet and the pdf of the design drawing.

To ensure that the validity of the resistance module is accurate, five different test models that were previously done were chosen. Each model test data investigated the resistance for many different hull bodies. This data was then utilized to test and compare the actual design of the test model to the generated design from the tool. After confirming the validity of the information output by the program, the MATLAB code was then finalized (Laun, 2013).

Hull Length (ft)	34.70
Hull Diameter (ft)	4.34
Total UUV Displacement (lb)	20,400
Total Envelope Volume (ft^3)	445
Usable Internal Volume (ft^3)	260
Total Internal Volume Required (ft^3)	173
Usable External Volume (ft^3)	188
Total External Volume Required (ft^3)	172

ESWBS Group	Weight (lb)
Group 100	6,391
Group 200	480
Group 300	7,124
Group 400	917
Group 500	2,279
Group 600	1,425
Group 700	100
Margin	1,872
Full-Load	20,588

Fitzgerald Limiting Depth Value (ft)	2,218
ABS Limiting Depth Value (ft)	1,881

Figure 3. Example of Conceptual Design (Laun, 2013).

2.4. Parametric Design of Sailing Hull Shapes

An article by Mancuso (2006) describes the process of creating a design tool to rapidly generate a hull shape for a sailboat. Designing a sailboat hull typically requires numerous iterations and entails a degree of randomness in the design choices. Mancuso streamlines the process by providing the program with a rough approximation curve of the desired hull shape then allowing each point of the hull to vary within a set range of that curve. The program can quickly converge to a successful design by iteratively varying each point and evaluating the result against predefined mathematical criteria. Mancuso uses equations that define sailboat hull parameters such as volume, and values for these parameters can be calculated for each generated design. They are then compared with optimal real-world values and, when taken together, provide the basis for determining the effectiveness of a design. Constructing the calculated design in a computational environment also allows for further predictive analysis of that hull's behavior.

An important design tool feature that Mancuso includes is the use of pre-set limits on the generated parameters. Geometric constraints are defined within the program to limit the possible outcomes and prevent infeasible solutions, such as a hull that crosses itself or a zero-thickness geometry. This or an analogous step is essential to include in projects of this nature because it ensures that a design tool generates solutions that not only satisfy all mathematical relations but are also feasible in real-world applications. This is highly relevant for our own project because we must ensure that the design specifications generated by our design tool are achievable with current UUV technology.

2.5. Parametric Design and Optimization of Multi-Rotor Aerial Vehicles

This article by Ampatis and Papadopolous (2014) focuses on optimizing an aerial vehicle design using off-the-shelf components. Specifically, equations are developed to describe a number of relevant parameters including weight, power, and so forth. The authors use a combination of available tested data from previous studies and their own derived mathematical formulas to model all relevant parameters. A particularly important relation that the authors generate is an estimation of the power output of different electrical motors based on their dimensions. They also use curve-fitting techniques to estimate the design specifications of equipment based on manufacturers' published data. The equations and available data are then input to an optimization procedure. The authors chose to use the "fmincon" function in MATLAB, which allows them to find the design input variable values that will cause minimums in their parameters of choice.

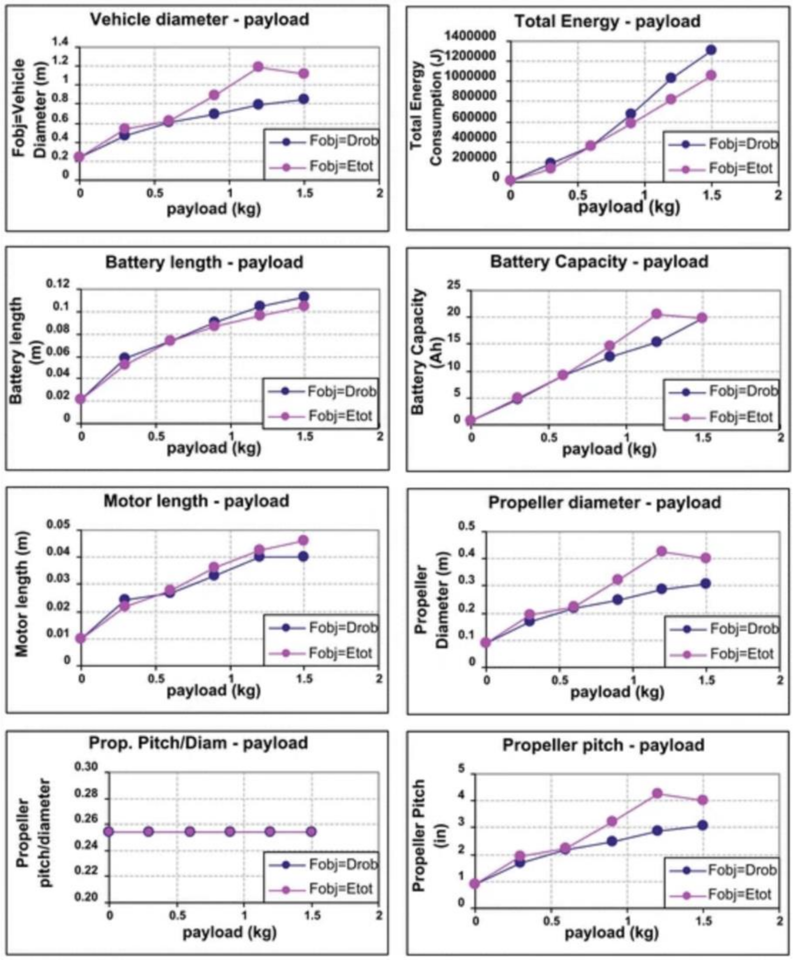

Figure 4. Graphs of Different AUV Variables With Only One Parameter Being Modified (Ampatis and Papadopolous, 2014).

The study investigates the effect of changing one design input while holding all others constant and then creates graphs that clearly display these relations, as shown in Figure 4. Recreating similar graphs between UUV parameters would not be a strictly necessary part of our project. However, such graphs would help provide context to users on the interrelation between parameters and help them visualize the impacts of changing parameters. These graphs appear to have been generated using MATLAB, but unfortunately the techniques used in this process were not described in detail.

2.6. MATLAB GUI for GOCE Sat Data

This system used the MATLAB App Designer to create a graphical user interface (GUI) for gradiometric data processing. The contents of the study are of no relevance to this project, but the GUI ability of MATLAB App Designer is.

Within their work, there is a good range of options within App Designer. This team was able to produce an app that included a variety of different graphs in quality detail as can be seen in Figures 5 and 6 below. Additionally, this app included selectable options such as tick boxes, toggles, alternative tabs, and status lights, as can be seen in Figure 5. The visual design is also high quality and it appears easy to utilize the tool to create similar options. App Designer includes a large number of options, including date pickers, buttons, drop-downs, fields, sliders, knobs, and others. It includes the option to quickly change between design view and code view, allowing visual modifications to be quickly integrated with a variety of calculations. This all seems to point to MATLAB App Designer being a fantastic option for designing our app, and the primary option we should use (Mamagiannou et. al, 2022).

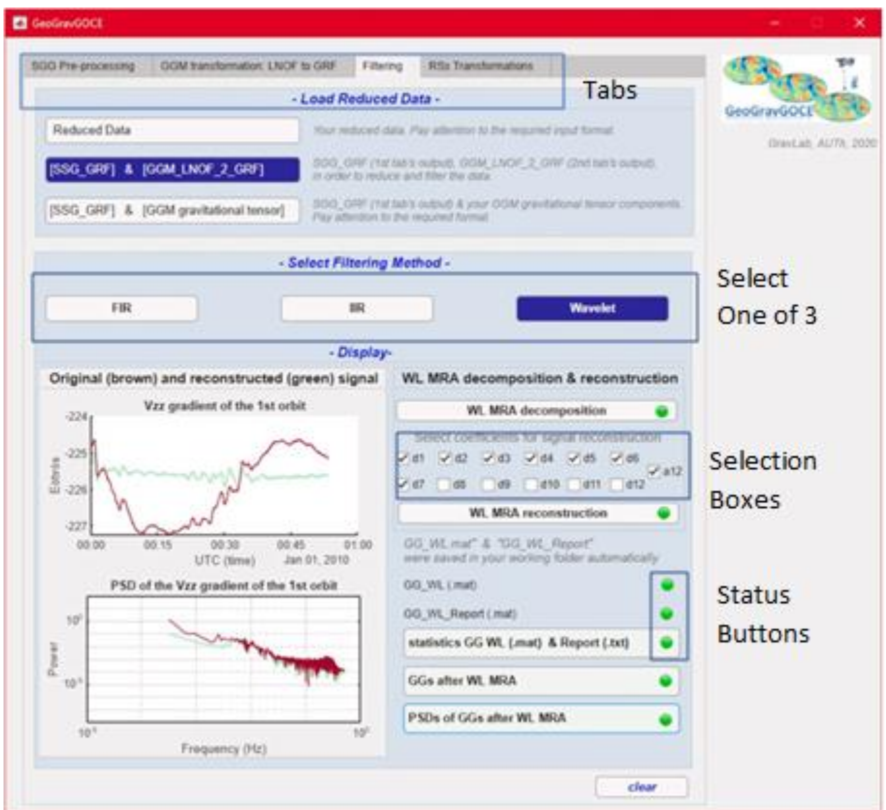

Figure 5. Example of App Designer Display (Mamagiannou et. al, 2022)

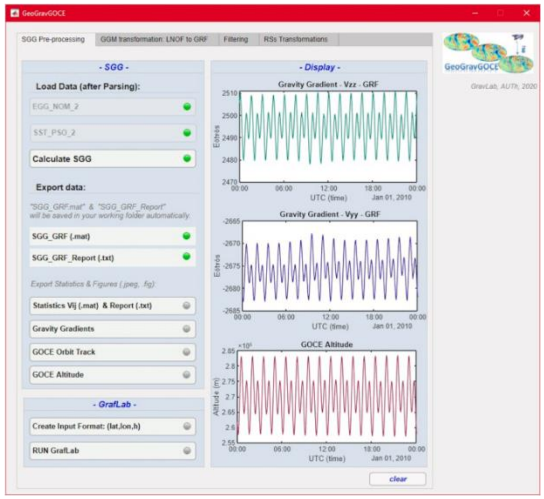

Figure 6. Output Graphs (Mamagiannou et. al, 2022)

2.7. Development of IEEE Compliant Software 'Economical Substation Grounding System Designer' Using MATLAB GUI Development Environment

Substations are some of the most important infrastructure in the United States. Failure to connect electricity through these substations results in a failure to deliver electricity to consumers, a failure which cripples our way of life. There are strict regulations about grounding systems within these substations, ensuring the safety of the system protecting the workers in the substation. Within these regulations, however, there are opportunities to be more efficient and save money by shifting grounding points, and removing excess points (Vyas et al, 2012).

The team used the regulations inherent to substations to serve as the background of the calculations for their system. They use parameters such as touch voltage, step voltage, mesh voltage, metal to metal touch voltage, and transferred voltage, to define whether a design is safe or not. The team used two different systems for analysis. The first was a performance analysis of a proposed system. This focused on determining whether the system was safe, calculating key performance indicators and safety criteria such as tolerable voltages, actual mesh and step voltages, and other key numbers. Those results are then displayed for the user to review.

The second system was an optimization function, which allowed users to select from several key inputs. From there, the system calculates key performance values. The system then cycles through all possible combinations of conductors and ground rods, saving the information of all feasible options. From there, it calculates the option that is most economical, and outputs the results and surrounding report. This method of handling development could be a potential roadmap for our project and is laid out in Figure 7 below.

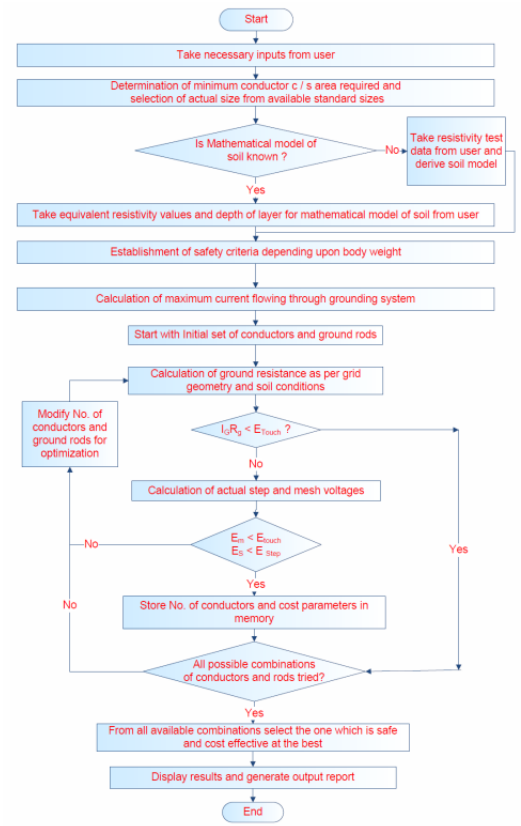

Figure 7. Procedure for Optimizing Designs (Vyas et al, 2012)

2.8. Modular Designs for Customizable UUVs and Sensor Networks

The tool that this project is based on is purposefully customizable. According to this paper, work is ongoing that would make UUVs and their design process more modular and allow thrusters or sensors to be swapped into different hull shapes as needed. Normalizing the weight ratios of these sections will also allow preservation of balance within the UUV. Figure 8 below demonstrates a possible view of this system and how it will work.

Figure 8. Modular Sections Within a UUV (Ehlers et al, 2020)

This process is interesting and could be a valuable addition to our tool. Having flexible positions for sensors and thrusters allows for a more flexible design and simple calculations. The authors of this paper simulated a UUV in a basic form with a basic sensor network arrays. From a central database, individual applications can interact with different stored variables. When a variable is eventually updated by the sensor network, a message is sent to the main board where it is decoded and converted into a readable source.

This system can be modified in such a way to include modular UUV design. Pre-coded limits to each sensor will allow the vehicle to move almost autonomously and reach maximum depth or pressure without hardware failure. Figure 9 below shows an example of the sensors reaching their limit and broadcasting that information back to a user.

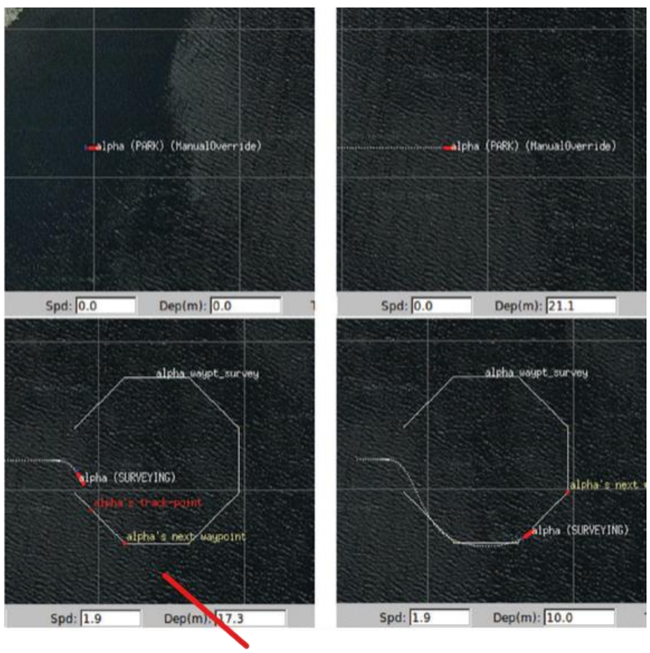

Figure 9. Sensors Broadcasting to the User (Ehlers et al., 2020)

2.9. Depth Control Methods for Variable Buoyancy UUV

Tests on variable buoyancy for UUVs are widespread. The team in this journal resolved several issues involving variable buoyancy UUV design by using numerical or quantitative data. Design and testing were performed at the Institute of Far Eastern Branch of the Russian Academy of Sciences. In the bow and the stern of the vehicle are two different modules, each containing a solid oil and hydraulic fluid tank. The UUV’s displacement can be changed by pumping fluids between these two tanks. Figure 10 below shows the designed and tested variable buoyancy UUV.

TABLE I
THE MAIN TECHNICAL SPECIFICATIONS OF AUV "CH-2".

Dry weight, [kg]	87.5
Buoyancy range, [kg]	-2.5…2.5
Length [m]	2.33
Diameter [m]	0.59
Max depth, [m]	150
Autonomy, [hour]	32
Max speed, [m/s]	2.5

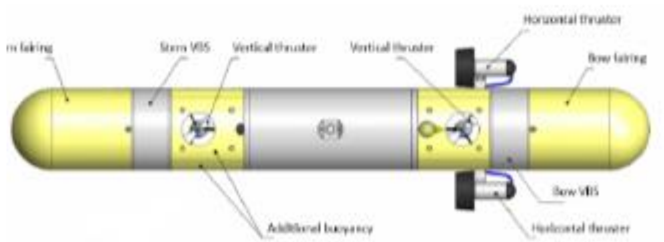

Figure 10. Variable Buoyancy UUV (Medvedev et al., 2017)

Different methods for controlling the depth of a UUV are propulsion, buoyancy, or a combination of the two. The propulsion method is the most common method for depth control. It negates the need for several complex problems within variable buoyancy. The propulsion method uses thrusters and battery power to move the vehicle vertically. The drawback of this method is that it uses a lot of power and it is rather noisy in the water, which can be highly detrimental in some UUV applications.

The second method for depth control is the buoyancy method. This method is much more complex as it involves pumps and more energy intensive work. The pumps themselves are often slow and can be unreliable. Combining both methods means that both buoyancy and propulsion techniques are used to maintain depth control in the UUV. A general control loop of this method is given below in Figure 11.

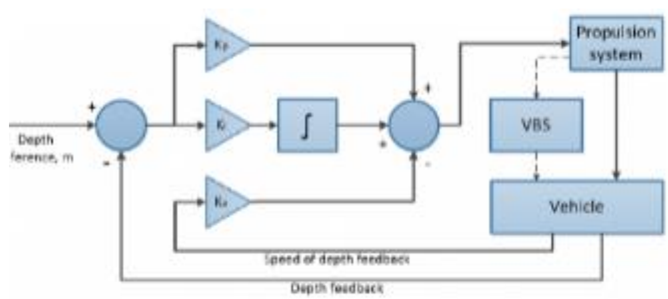

Figure 11. General Loop of the Combined Method of Control (Medvedev et al., 2017)

3. Methods

After considering the project requirements, our research, and feedback from our project advisor and sponsors, we formulated the following steps to complete our project:

1. Receive and understand the sponsor's requirements.
2. Determine the design functions of the tool.
3. Develop concept sketches of the tool's layout and functionality.
4. Decide which UUV parameters we will include in our tool.
5. Define relationships between parameters.
6. Decide what software we will use to create our tool.
7. Create a small-scale prototype focused on functionality.
8. Create a user interface and integrate it into the prototype.
9. Expand our prototype to include all parameters.
10. Test tool validity using real life data or examples.

Our first step will be to communicate with our sponsors and understand their requirements for the tool. These requirements will dictate the course and end goals for the entire project, so clear communication will be essential. We next must use their input to set the tool's design functions. These functions include the tool's inputs, outputs, features, and functionality. These elements form the core of our project, so they are crucially important and we must ensure that they meet our sponsors' needs. We will then form several conceptual layouts for our tool that visually describe potential methods of operation and ways to meet the project requirements. We will select one or more concepts for further development or combine beneficial elements across multiple concepts. Our next step will be to decide the UUV design parameters to include in our tool. We will focus our study on the variables that have the greatest impact on UUV design, as well as the ones most important to the project's sponsors. We will find these relationships by researching academic and technical sources, such as fluid mechanics textbooks and published scientific literature. We also plan to determine these relationships through computational or analytical methods if needed. After establishing the relations between UUV parameters, we will determine what software we will use to perform our calculations and run our tool.

After defining the relationships between variables and choosing an initial software, we will develop a prototype tool that considers only a selection of our chosen parameters – likely no more than 5. This prototype will focus on functionality, not on presenting a polished user interface. We will next work to integrate an intuitive and polished graphical user interface with this prototype. Simultaneous to adding a UI to our prototype tool, we will also gradually expand the functionality of our tool to include the full selection of parameters. Once our tool is fully completed, we will perform in-depth testing on it. Similarly, to Laun (2013), we will compare our tool's calculated, hypothetical results to data from real-world UUVs to ensure accuracy. We will test our tool with engineers and non-engineers to ensure both technical rigor and ease of use. We will perform testing of every component of our tool throughout the design and prototyping process, but our most in-depth and thorough testing will occur once the tool is fully completed.

4. Approach

Below is a listing of the variables that we researched equations for and used in our project.

Variable	**Definition**
A_n	Nose surface area
A_m	Main surface area
A_t	Tail surface area
A	Total surface area
B_{Net}	Net Buoyancy, usually displayed in mass units
C_{form}	Drag coefficient due to UUV form
$C_{friction}$	Drag coefficient due to skin friction
C_d	Total drag coefficient
D	UUV External Diameter
d	Diameter to UUV's Neutral Axis
E	Elasticity Modulus
E_c	Energy Capacity
e	Specific energy
L	Length of UUV
L_n	Nose length
L_m	Main length
L_t	Tail Length
M_{disp}	Mass of the displaced water
M_{shell}	Shell mass

M_{bat}	Battery mass
$M_{payload}$	Mass of the payload
M_{motor}	Mass of the motor
M_{total}	Total mass
m	Inverse Poisson's Ratio
P_h	Hotel Load
p_e	External Pressure on the UUV
R	Range
r_n	Nose radius
r_m	Main radius
r_t	Tail radius
t_{Al}	Shell thickness for an Aluminum UUV
t_{CFRP}	Shell thickness for a CFRP UUV
V_n	Nose volume
V_m	Main volume
V_t	Tail volume
V	Total volume
v	Velocity
v_p	Poisson's Ratio
ρ_{water}	Density of water
ρ_{shell}	Density of Shell

4.1 UUV Data and Mathematical Formulas

Hull Material

UUV hulls can be composed of various materials. The material selection process of a UUV is dependent on the UUV's mission and depth requirements (French, 2010). UUVs have been made of stainless steel, titanium, aluminum, acrylic, carbon fiber, fiberglass, and many more. Each type of material has advantages and disadvantages. Plastic materials are a good option for a lightweight UUV but are not comparable to metal alloys for great depths. Titanium can handle great pressures but is challenging to machine and not cost-efficient compared to other materials (Villalba Herreros, 2021). Aluminum possesses many advantages compared to other UUV hull materials. It is lighter than steel, is easily accessible, and easier to manufacture (Good, 1989).

We chose to include Carbon Fiber Reinforced Polymer (CFRP) and Aluminum as the material options in our tool. Our tool required the density, modulus of elasticity, and Poisson's ratio of a material to perform its calculations, and these values for our chosen materials are as follows:

CFRP

Density: $1800\ \frac{kg}{m^3}$

Modulus: 141 GPa

Poisson's ratio: 0.28

Data sourced from Liu et al. (2021a) and *Technical Data Sheet* (n.d.).

6061 Aluminum

Density: $2700\ \frac{kg}{m^3}$

Modulus: 68.9 GPa

Poisson's ratio: 0.33

Data sourced from *Aluminum 60601-O* (n.d.).

Hull Thickness

The thickness of a UUV's hull is dependent on many factors, including material, depth, dimensions and geometry, and desired safety factor. Additionally, the thickness required at the geometrically complex ends of our UUV model is especially difficult to calculate. The exact thickness required for a given UUV is a design decision that is difficult to predict exactly, but we have provided a reasonable estimate of it through our research. The pressure required to collapse a long, thin tube is found from Saunders & Windenburg (1931) as the following equation:

$$(1)\ p_e = 2\frac{m^2 E}{m^2-1}\left(\frac{t}{d}\right)^3$$

The pressure external to the hull is found from the underwater depth of the UUV. We fixed this depth at 600m, a typical depth for commercial UUVs, and the pressure is found from the equation:

$$(2)\ p_e = h \cdot \rho_{water} \cdot g = 600m\left(1000\,\frac{kg}{m^3}\right)\left(9.81\,\frac{m}{s^2}\right) = 5.886\ MPa$$

The diameter to the neutral axis is the diameter of the middle of the UUV shell, given by the equation:

$$(3)\ d = D - t$$

The inverse Poisson's ratio is given by:

$$(4)\ m = \frac{1}{v_p}$$

Combining these separate equations and adding a safety factor of 3 leads to the following finalized equation:

$$(5)\ 3 \times 5.886 \cdot 10^6 = 2\frac{\left(\frac{1}{v}\right)^2 E}{\left(\frac{1}{v}\right)^2 - 1}\left(\frac{t}{D-t}\right)^3$$

This equation gives thickness as an implicit function of the UUV diameter, as well as the material's elasticity modulus and Poisson's ratio. We then entered this equation into Desmos and solved for thickness given a chosen diameter, as demonstrated in Figure 12.

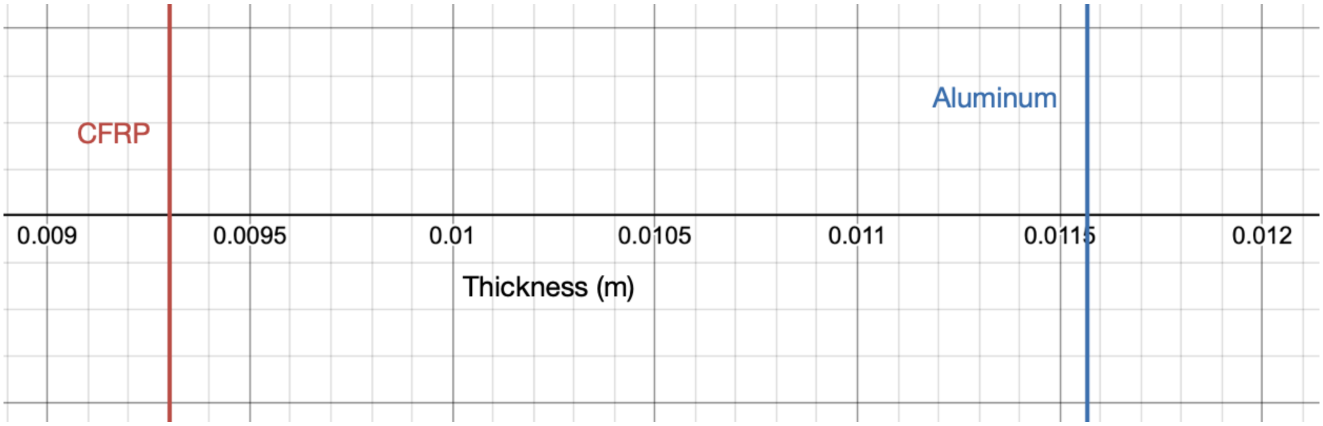

Figure 12. Necessary Hull Thickness for CFRP and Aluminum with a Diameter of 0.5 Meters

This equation was solved across a variety of UUV diameters and the generated data points are listed in Table 1.

Table 1. Diameter and Thickness Data for Our Chosen Materials

CFRP		**Aluminum**	
Diameter (m)	Thickness (m)	Diameter (m)	Thickness (m)
0.1	0.00372	0.1	0.004627
0.3	0.01116	0.3	0.01388
0.5	0.018603	0.5	0.02313
0.75	0.0279	0.75	0.0347
1	0.0372	1	0.04627
1.25	0.0465	1.25	0.0578
1.5	0.05588	1.5	0.0694
2	0.07441	2	0.09254

These data points were then graphed, and lines of best fit were generated. These lines are shown in Figure 13.

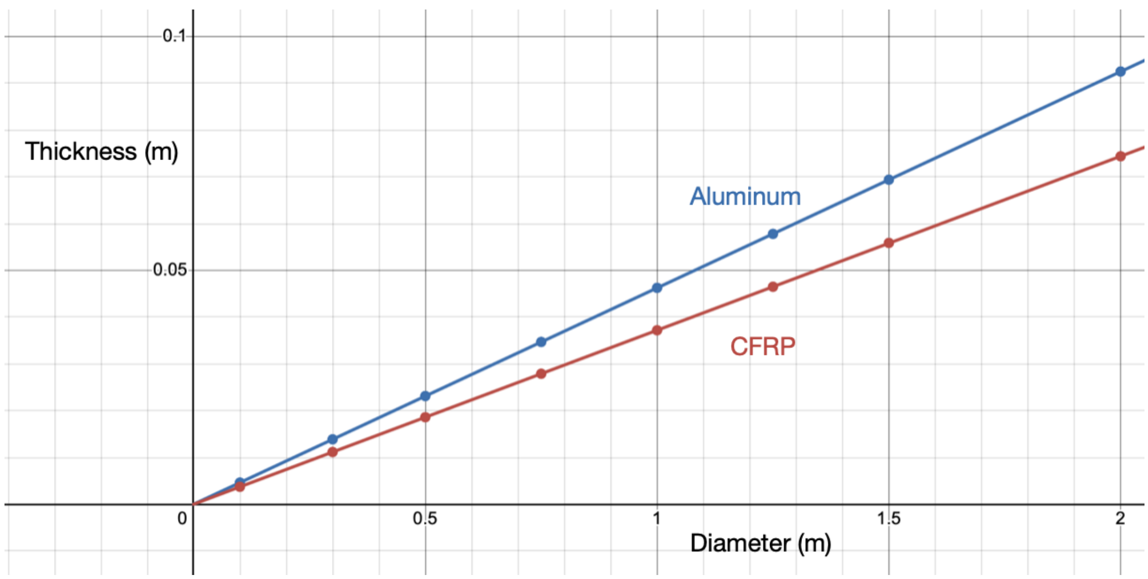

Figure 13. Data Points and Fitted Curves for Thickness as a Function of Diameter

These fitted lines have the following equations:

$$(6)\ t_{Al} = 0.0462 \cdot D$$

$$(7)\ t_{CFRP} = 0.0372 \cdot D$$

Battery

A UUV can be powered by different power sources including a variety of batteries. The battery of a UUV contributes to the range, performance, and weight of the overall vehicle. Though they have some drawbacks including safety risks, Lithium-ion batteries are the most common choice for powering UUVs because they offer a high specific energy and cycle life with low maintenance (Mendez et al., 2014; Gitzendanner et al., 2004). Figure 14 displays a comparison between the energy data of different battery types, clearly showing how Lithium-ion batteries offer superior performance to the other batteries listed (Cai et al., 2007).

	VRLA	Ni-Cd	NiMH	Li-ion	***ZEBRA***
Specific energy (Wh kg^{-1})	34	45	65	120	120
Energy density (Wh L^{-1})	120	110	135	230	140
Specific power (W kg^{-1})	75	120	90	220	180
Self-discharge per month (%)	8	10	30	5	None
Nameplate cycle life	500	1000	500	>1000	2000
Efficiency (%)	70	80	80	85	90
Operation temperature (°C)	-15~45	-40~70	-30~70	-20 ~60	275~350
Cost (£ kW h^{-1})	105-175	200-300	250-350	250-1000	70-270

Figure 14. Comparison Between Battery Types (Cai et al., 2007)

The energy data of Lithium-ion batteries can vary widely between specific batteries, with different sources giving specific energy ranges of 165-207 and 80-250 $\frac{Wh}{kg}$, respectively (Mendez, 2014; Šimić et al.,2021). Table 2 shows examples of data from specific Lithium-ion batteries used in current UUVs across the industry.

Table 2. Battery Data from Commercial UUVs

Battery Type	**Energy Capacity**	**Weight**	**Specific Energy**	**Source**
EaglePicher LP33333	1.54 kWh	17.24 kg	89.3 $\frac{Wh}{kg}$	*28 V/55 Ah Lithium Ion*, (n.d.).
Teledyne 1.2 kWh	1.2 kWh	14.5 kg	82.8 $\frac{Wh}{kg}$	*Battery Module*, (n.d.).
Bluefin 1.5 kWh	1.5 kWh	14.3 kg	104.9 $\frac{Wh}{kg}$	*Bluefin 1.5 kWh Subsea Battery*, (n.d.).
SubCTech 260 mm	2.5 kWh	14 kg	178.6 $\frac{Wh}{kg}$	*Vehicle Batteries*, (n.d.).
SubCTech 310 mm	3.76 kWh	18 kg	208.9 $\frac{Wh}{kg}$	*Vehicle Batteries*, (n.d.).
SubCTech 416 mm	7.1 kWh	32 kg	221.9 $\frac{Wh}{kg}$	*Vehicle Batteries*, (n.d.).

Given all the above data, we decided to offer two options for specific energy in our tool: a lower estimate of $100 \frac{Wh}{kg}$ and a higher estimate of $200 \frac{Wh}{kg}$. We also offered a lead-acid battery option with a specific energy of 40 $\frac{Wh}{kg}$ to demonstrate the impact of available energy on UUV performance (Šimić et al., 2021, p. 56).

Shape

A study of relevant literature shows that UUVs can be described by governing equations with similar forms and varied levels of generalization. We chose to use a teardrop general UUV shape, as given by Liu et al. (2021b, pg. 2). This shape most closely matched the description given by our project sponsors as well as the commercial UUVs that we encountered in our research. A picture of this shape is shown in Figure 15.

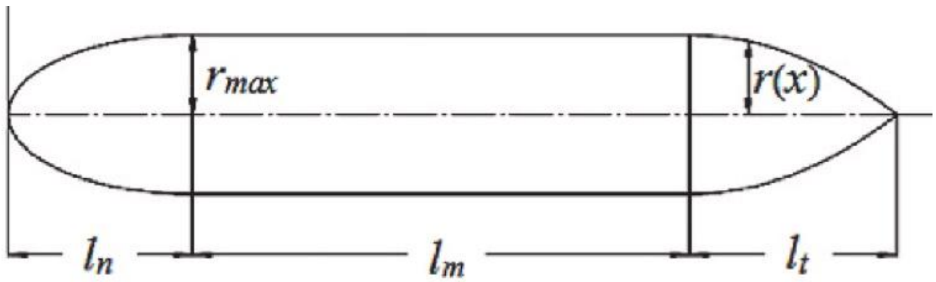

Figure 15. A Teardrop-Shaped General Hull Form of UUV

Liu et al. (2021b) provide the following equations that define the radius of the nose, main, and tail sections, respectively.

$$(8)\ r_f = \left(\frac{D}{2}\right)\left(1 - \left(\frac{L_f - x}{L_f}\right)^{n_n}\right)^{\frac{1}{n_n}} \{0 \le x < L_f\}$$

$$(9)\ r_c = \frac{D}{2}\{L_f \le x < L_f + L_m\}$$

$$(10) \qquad r_a = \left(\frac{D}{2}\right)\left(1 - \left(\frac{x - L_f - L_m}{L_a}\right)^{n_t}\right)\{L_n + L_m \le x \le L\}$$

In these equations, n_n and n_t are factors used to control the curvature of the front and rear UUV sections. Liu et al. (2021b) state that these factors have values that typically range from 1 to 2. We desired fixed values for these parameters to simplify our tool, so we chose 1.7 for the front curvature and 1.9 for the tail curvature.

It is convenient to define the length of the front and tail sections as a function of the UUV diameter to maintain the desired form factor and limit the required inputs. Liu et al. (2021b) states that the front and rear sections have lengths that typically land within the following ranges:

$$(11) \qquad 0.75 < \frac{L_f}{D} < 2 \qquad 1 < \frac{L_a}{D} < 2.25$$

To simplify our tool and reduce the number of parameters, we chose to keep these values constant. The UUV model will therefore have fixed form factors for the front and rear sections, with the length of these sections changing as a function of diameter. We chose to use the average values of the ranges given in equation (11), which are:

$$(12) \qquad L_f = 1.4D \qquad L_a = 1.6D$$

We omitted control surfaces, propulsion equipment, antennas, and other external additions to the base shape due to a lack of data on these surfaces. We used the equations listed above as the basis for our UUV throughout the tool and made a CAD model of the UUV in SolidWorks according to these equations and assumptions. The model inputs can be quickly changed to modify the shape. It is shown in Figures 16-18 below with a diameter of 0.5 m and total length of 6 m.

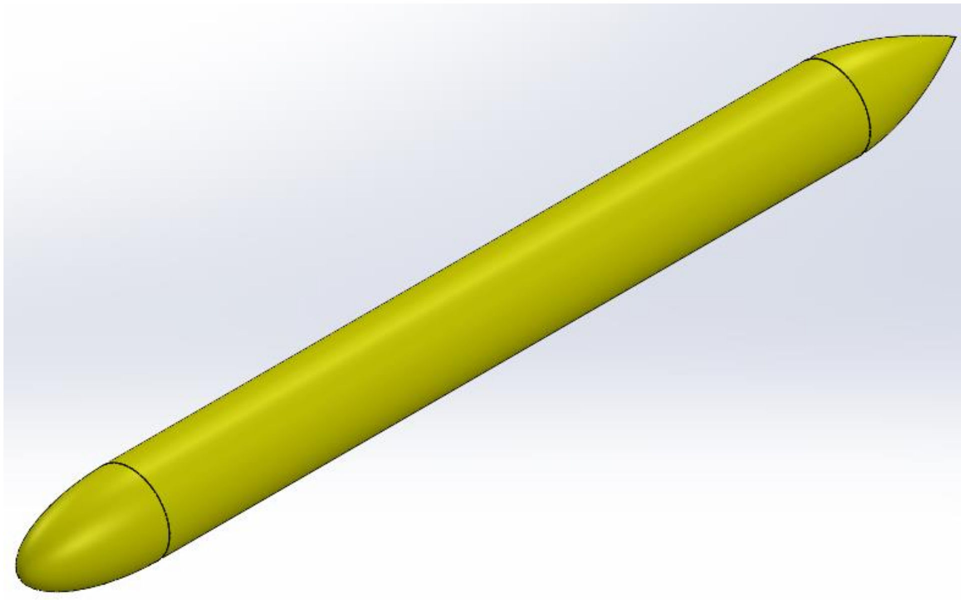

Figure 16. SolidWorks Model of the UUV Shape Used in Our Tool

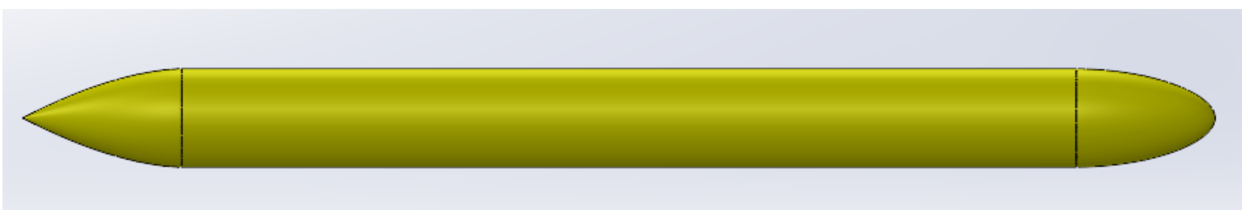

Figure 17. Side View of the SolidWorks Model

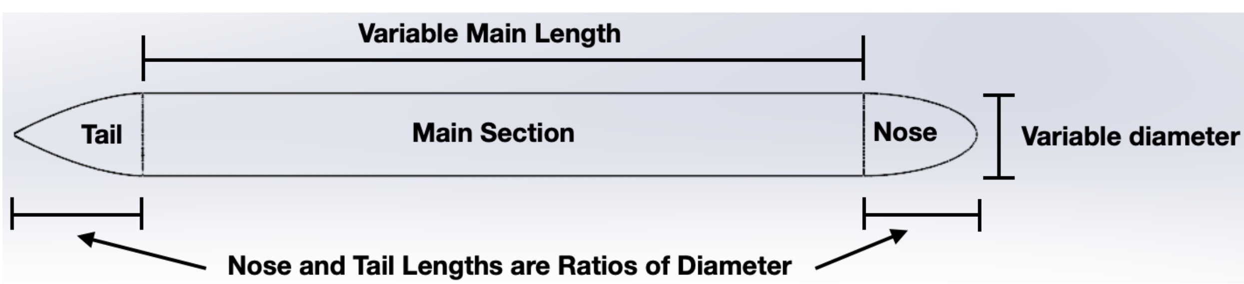

Figure 18. Diagram of the Model Showing Changeable Parameters

Drag Force and Coefficient

Liu et al. (2021b) provides equations for the Reynolds number, drag coefficients, and drag force of the teardrop shaped UUV. The Reynolds number must first be calculated as shown:

$$(13) \qquad Re = \left(10030\,\frac{m\cdot s}{kg}\right)(\rho_{water})(v)(L)$$

The equations below are given in Liu et al. (2021b) and give the form and friction drag coefficients for the teardrop shaped UUV:

$$(14) \qquad C_{form} = \frac{20.72\left(\frac{L_f}{D}\right)^{-1.218}+54.31\left(\frac{L_a}{D}\right)^{-3.428}+3.496\,(\log(Re))^2-56.29\log(Re)+273.03}{1000\left(\frac{L}{D}+0.9165\frac{L_f}{D}+1.258\frac{L_a}{D}-0.5394\right)}$$

$$(15) \qquad C_{friction} = \frac{\left(0.053136\frac{L_f}{D}-0.1581\frac{L_a}{D}+2.5744\right)\frac{L}{D}-5.2072\frac{L_f}{D}+1.1499\frac{L_a}{D}+54.59}{1000(\log(Re)-3.87)}$$

Plugging in our chosen values of $\frac{L_f}{D} = 1.4$ & $\frac{L_a}{D} = 1.6$ and simplifying where possible yields the following finalized equations that we used in our project:

$$(16) \qquad C_{form} = \frac{297.626441+3.496(\log(Re))^2-56.29\log(Re)}{1000\left(\frac{L}{D}+2.7565\right)}$$

$$(17) \qquad C_{friction} = \frac{2.39583\frac{L}{D}+49.13976}{1000(\log(Re)-3.87)}$$

These expressions are summed together to give the total drag coefficient:

$$(18) \qquad C_d = C_{form} + C_{friction}$$

These equations yield volumetric drag coefficients. Liu et al. (2021b) provides the equation to obtain drag force from a volumetric drag coefficient:

$$(19) \qquad F_{drag} = \frac{\rho_{water} v^2 V^{\frac{2}{3}} C_d}{2}$$

Volume

The volume of the UUV could be calculated in three sections: the front, middle, and end. The volume of the main section is that of a cylinder, given by the simple equation:

$$(20) \qquad V_m = \frac{\pi D^2}{4} L_m$$

The forward and aft sections have more complex geometries, however, that do not yield simple analytical volume equations. We therefore found these equations numerically. We calculated the volume of the front and back sections of our SolidWorks model and generated the following data. Note that the lengths of these sections are a function of diameter, so the volume of these sections varies only with changing diameter.

Table 3. UUV Model Diameters and Corresponding Volume Data

Diameter (m)	**Forward Volume (m^3)**	**Aft Volume (m^3)**
0.25	0.01	0.01
0.5	0.09	0.08
0.75	0.3	0.26
1	0.71	0.61
2	5.68	4.89
3	19.16	16.51
4	45.41	39.14
5	88.7	76.44

This data was entered into Desmos and graphed, then lines of best fit were generated. The datapoints and their fitted curves are shown in Figure 19.

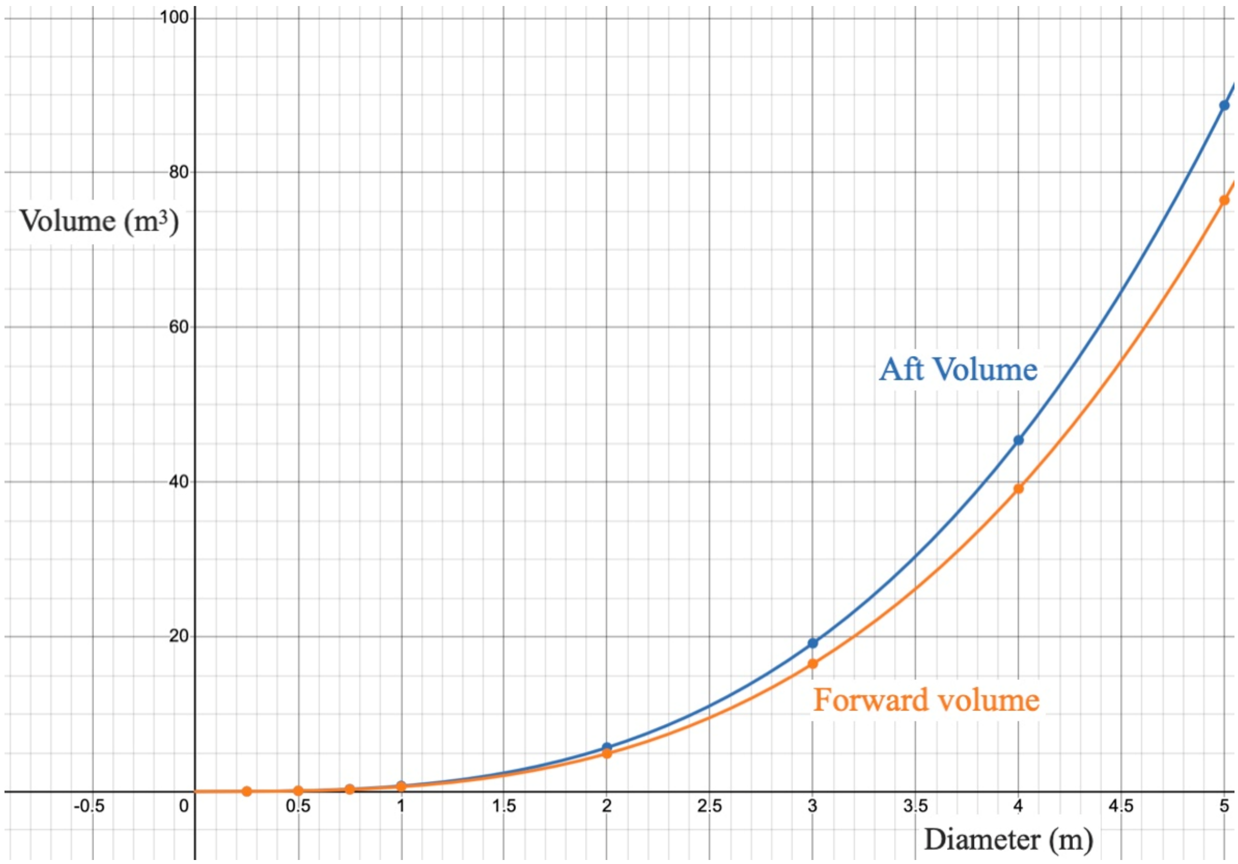

Figure 19. Data Points and Fitted Curves for Volume as a Function of Diameter

The curve fitted lines have the following equations:

$$V_n = 0.70954D^3 \tag{21}$$

$$V_t = 0.611528D^3 \tag{22}$$

Surface Area

Equations for the surface area of the UUV were found in a similar fashion to the equations for volume. The center section of the UUV is a cylinder and has a simple equation for surface area:

$$A_m = 2\pi D L_c \tag{23}$$

The area equations of the more complex nose and tail sections are found from the CAD model in the same way as the volume equations. We collected surface area values from SolidWorks for a range of diameters, as shown in Table 4.

Table 4. UUV Model Diameters and Corresponding Surface Area Data

Diameter (m)	Forward Surface Area (m^2)	Aft Surface Area (m^2)
0.25	0.22	0.21
0.5	0.89	0.82
0.75	1.99	1.85
1	3.54	3.28
2	14.18	13.13
3	31.9	29.55
4	56.72	52.53
5	88.62	82.08

These datapoints were entered into Desmos and graphed, then curves of best fit were formed, as shown in Figure 20.

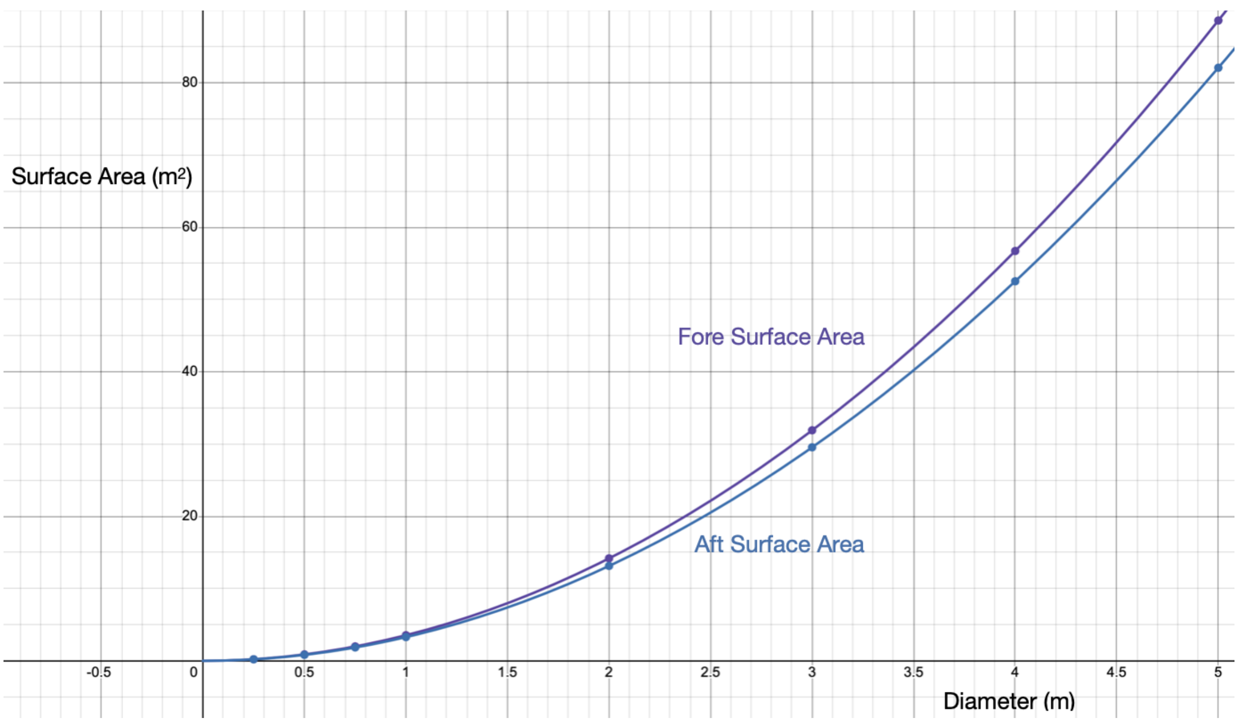

Figure 20. Data Points and Fitted Curves for Surface Area as a Function of Diameter

The curve fitted lines have the following equations:

$$(24) \qquad A_n = 3.544D^2$$

$$(25) \qquad A_t = 3.283D^2$$

Energy Capacity

A UUV's energy consumption can be divided into two categories: energy used to propel the ship and energy used for other purposes, called hotel load. The hotel load is dependent on the sensors, communications equipment, and other electronics in the UUV. The hotel load is given as power, or an energy consumption rate, so it must be multiplied by time. The time of the UUV's mission can be given as its range divided by its velocity. The propulsion energy can also be calculated from known parameters. The basic equation for energy is a force multiplied by a distance. The force in this case is the UUV drag force, given in equation (19), and the distance is the UUV range. We must also account for inefficiency in the motor. All motors consume wasted energy that is not used to propel the ship, signified by an efficiency coefficient η . Combining all of these elements gives the following equation:

$$(26) \qquad E_c = P_h\left(\frac{R}{v}\right) + R\left(\frac{F_{drag}}{\eta_{motor}}\right)$$

5. Prototype Development

5.1. Software Selection

After examining the different software available to us, we selected Microsoft Excel for our prototypes due to its ease of use and its computational ability. Figure 1 gives an example of a simple, intuitive Excel program that helped provide a strong basis for a similar prototype tool of our own (Fomin et al., 2005). For our final tool, we decided that MATLAB would be the best choice due to its popularity among engineers, positive feedback on the proposal from our sponsors, and its use in many of the similar works that we researched (Laun, 2013; Ampatis and Papadopolous, 2014; Mamagiannou et al., 2022; Vyas et al., 2012). For the transition from Microsoft Excel to MATLAB, we utilized the App Designer tool to create our user interface. App Designer provides a range of features and functionality that integrated seamlessly with our MATLAB code and was simple to work with.

5.2. Microsoft Excel Prototypes

The initial development path for the system was through linear calculations in Microsoft Excel. The prototypes developed here would later be used to form the basis of the calculations for the final system that was implemented into the MATLAB user interface. A series of prototypes were developed, each one a development or improvement on the previous one.

The first prototype developed laid the groundwork in several ways. First, it set the visual scheme of the excel prototypes. The simple layout allowed for a clear display of all values. Computationally, this first prototype was very simple, only taking the inputs of desired speed, range, hotel load, and drag coefficient. The prototype then solved for the energy capacity needed for the system. As previously stated, this was exceptionally simple and was primarily a proof of concept for the Excel prototypes.

The second prototype developed was more complex and built to a net buoyancy value. This was done by assuming the walls of the UUV served as a pressure hull. From there, the displaced mass was calculated as the product of water density and the vessel's volume. This volume was calculated very rudimentarily, assuming two hemispheres connected by a cylinder. The mass of the boat was calculated by summing the battery mass with a nebulous “Other Mass” meant to represent the shell, motor, and payload masses in aggregate. The battery mass was calculated by dividing the desired energy capacity by the specific energy. Finally, the net buoyancy was found by subtracting the total mass of the boat from the displaced mass of water. This system allowed us to introduce the conditional formatting function inherent to Excel. This was used to highlight the box as green in the case of a positive buoyancy (the UUV can float), or red in the case of a negative buoyancy (the UUV will sink). This proved useful in future cases, including testing if an output value is within a prescribed acceptable range.

When preparing for the implementation of the Excel Prototypes into MATLAB, we reorganized the inputs we had created. As the MATLAB versions would focus on calculating net buoyancy and comparing it to permissible ranges, the final Excel prototype also included net buoyancy as the calculated value. As shown in Figure 21, we calculated several intermediate values that would be valuable when creating the MATLAB tool.

INPUT	
Net Boyancy (kg):	-331.42
Range (m):	160000
Length (m):	5
Speed (knots):	5
Payload Mass (kg) :	10
Depth Capacity (m): (Not curretly Used)	
Diameter (m):	0.32
Battery Type	Lead-Acid
Material	Composite
Front Curvature (Not Used)	1
Back Curvature	2.7
Hotel Load (W):	1000
Motor Mass (kg):	10
Payload Volume (m^3) : (Not Used)	
Mass Volume (m^3): (Not Used)	

Constants	
Density of Water @ 20 °C (kg/m^3)	1025
Specific Energy of Lithium-Ion (J/kg)	540000
Specific Energy of Lead-Acid (J/kg)	144000
Specific Energy of Alternative Li (J/kg)	360000
Density of Iron (kg/m^3)	7800
Density of Steel (kg/m^3)	7800
Density of Composite (kg/m^3)	2710
Water viscosity @ 20°C (Ns/m^2)	0.00109

Intermediate	
Forward Length (m)	0.198400000
Central Length (m)	4.20960
Aft Length (m)	0.59200
Forward Volume(m^3)	0.01
Central Volume(m^3)	0.34
Aft Volume(m^3)	0.029315
Total Volume(m^3)	0.38
Forward Area (m^2)	0.19
Central Area (m^2)	4.23
Aft Area (m^2)	0.45
Total Area (m^2)	4.87
Displaced Mass (kg)	387.97
Shell Density (kg/m^3)	2710.00
Thickness (m) (Currently Incorrect)	0.02
Shell Mass (kg)	263.76
Speed (m/s)	2.57
Drag Coefficeint (Currently Set)	0.50
Energy Capacity	62732188.62
Specific Energy	144000.00
Battery Mass (kg)	435.64
Total Mass (kg)	719.40
Reynold's Number @ 20 °C	188139.40

Figure 21. Final Reorganized Prototype

6. MATLAB User Interface Development

The user interface shown in Figure 22 displays our initial stages utilizing MATLAB App Designer. The interface is a linear design with a few modifications to show the capabilities of MATLAB.

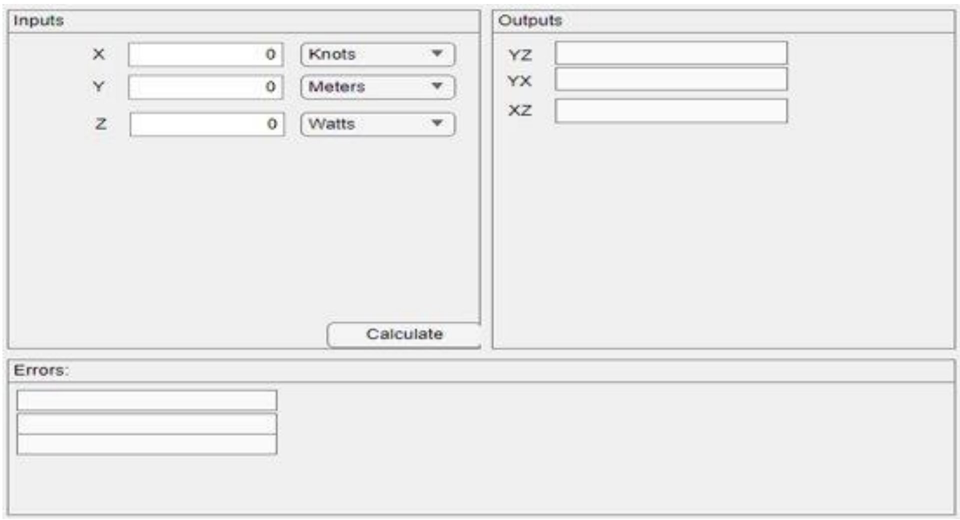

Figure 22. First UI Prototype

As shown in Figure 22, the design of the user interface is extremely lackluster but served its purpose of showing the capabilities of a MATLAB UI to the team. The second prototype, however, is much more complex. Once MATLAB was thoroughly understood, the team was able to improve on the first prototype greatly. Shown below in Figure 23 is the second and final MATLAB UI prototype.

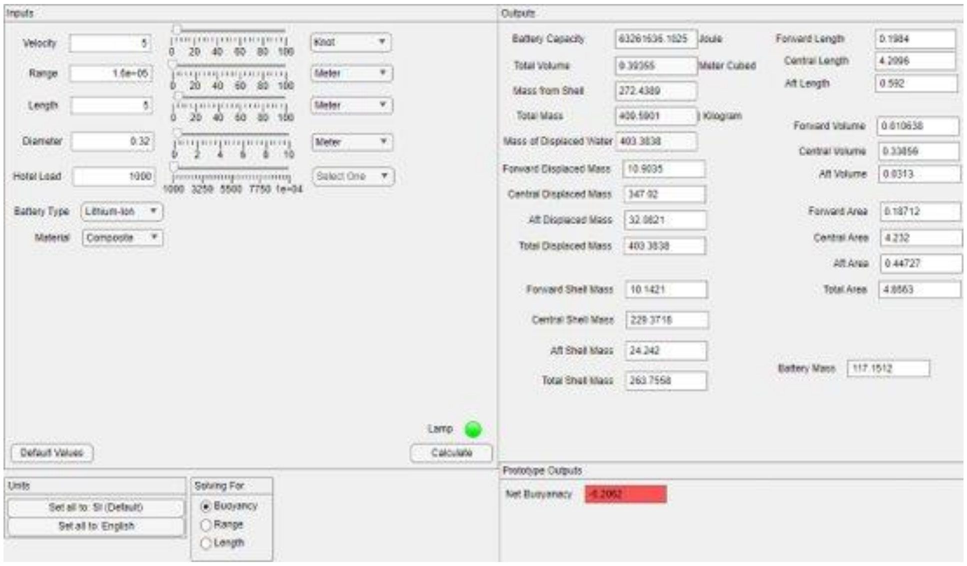

Figure 23. Second UI Prototype

The direct transfer from Excel to MATLAB meant that this UI is very similar to the Excel prototype layout. App Designer offers pre-defined code segments for visual features such as buttons and slider, allowing many tedious coding processes to be accomplished faster and more easily. Figure 24 shows an example of the code behind App Designer, showing how its features can be combined and modified to create the intended result.

```
if app.DiameterDropDown.ValueIndex == 2
    Diam = (app.DiameterEditField.Value*0.3048);
elseif app.DiameterDropDown.ValueIndex == 3
    Diam = app.DiameterEditField.Value;
else
    app.DiameterDropDown.ValueIndex = 3;
    Diam = app.DiameterEditField.Value;
end
global length;
global totvol;
function TotVol(~,~)
    if app.LengthDropDown.ValueIndex == 2
        length = (app.LengthEditField.Value*0.3048);
    elseif app.LengthDropDown.ValueIndex == 3
        length = app.LengthEditField.Value;
    else
        app.LengthDropDown.ValueIndex = 3;
        length = app.LengthEditField.Value;
    end
    totvol = (length-Diam)*(Diam^2)*(pi/4)+(pi*((Diam^3)/6));
    if totvol <= 0 || isnan(totvol)
        app.TotalVolumeEditField.BackgroundColor = ("red");
    else
        app.TotalVolumeEditField.BackgroundColor = ("white");
    end
    app.TotalVolumeEditField.Value = num2str(totvol);
end
TotVol();
%thick in meters. temp value. DensShip in kg/m^3 also temp
%value
thick = .02;
if app.MaterialDropDown.ValueIndex == 2 || app.MaterialDropDown.ValueIndex == 3
    DensShip = 7800;
```

Figure 24. A Snippet of Code from MATLAB

Each equation in the code is running through separate functions to prevent errors and allow easy cleanup and repair. Something that should be improved on in this version of the code is the number of global variables. Global variables can provide unseen errors that are relatively difficult to diagnose, especially when many lines have already been written. The use of comments within the code is also imperative for efficient development and troubleshooting.

7. Final Design and Validation

7.1. Tool Functionality

7.1.1. Input Parameters

The final tool resulted in nine user input parameters: buoyancy, range, length, speed, hotel load, payload mass, diameter, and motor mass, battery type, and material type. Battery and material type are displayed as checkboxes, as can be seen in Figure 25. All other inputs are displayed as shown in Figure 26 below, with some slight differences for hotel load and motor mass, as shown in Figure 27.

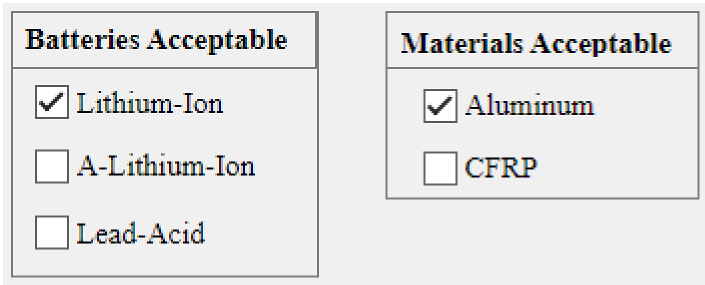

Figure 25. Battery and Material Checkboxes

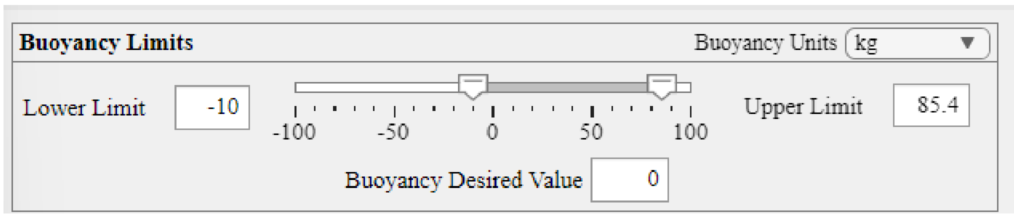

Figure 26. General Input Layout

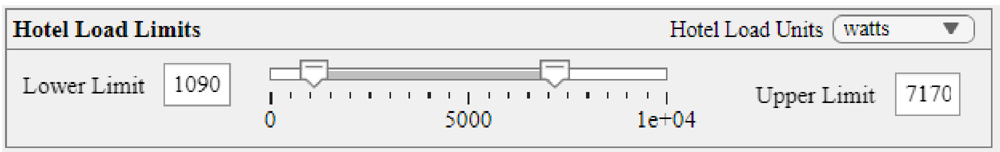

Figure 27. Hotel Load and Motor Mass Layout

These inputs allow the user to set permissible ranges, desired units, and the desired value to sort by for certain parameters. Additionally, by utilizing the MATLAB App Designer features, the tool can update in real time, displaying the calculated number of iterations. The user interface also moves the upper or lower limits as needed to ensure that the upper limit is never less than the lower limit. Together, the input panels form a robust base to run the program from.

7.1.2. Equations and Calculations

The key to the program is the net buoyancy calculation, which is built on the equations formulated in Section 4.1. As shown in Section 7.1.3., net buoyancy is the final value that is checked in comparison to the allowable ranges from the user. We can find the net buoyancy in equation (27) below. From there, we can break it into its component parts, and work until we find values with equations from Section 4.1. First, the displaced mass can be calculated as seen in equation (21) below. Following that, the volume is shown in equation (29) as the simple sum of the component volumes of the boat. That solves for the displaced mass, and we focus on the total mass of the boat in equation (30). That is found from the sum of several different values. First, the payload and motor masses have given values in each any iteration. Equation (31) gives the mass of the battery by dividing the total energy capacity needed for the boat by the specific energy of the battery being used. The energy capacity equation itself can be seen as a combination of equations (19) and (26) with drag force substituted in and the variables reorganized to create equation (32). As this sufficiently explains the battery mass equation, we can focus on the shell mass equation. The shell mass is found by multiplying the total volume of the shell, which is the area times the thickness, by the density of the shell, as shown in equation (33). The thickness varies with the chosen material according to equations (6) and (7). The density of the shell is determined by the type of material used for that iteration. Similarly to volume, the surface area of the shell is found by summing the component parts to generate equation (34). Together, this explains how to tool calculates a net buoyancy value for each iteration. Several constants are used. The density of water is assumed to be $1025\,\frac{kg}{m^3}$, and the motor efficiency is assumed to be 0.75.

(27) Net Buoyancy: $B_{Net} = M_{Disp} - M_{Total}$

$$B_{Net} = M_{Disp} - M_{Total} M_{Disp} = \rho_{water} \cdot V \tag{28}$$

$$V = V_f + V_m + V_a \tag{29}$$

$$M_{Total} = M_{shell} + M_{bat} + M_{payload} + M_{motor} \tag{30}$$

$$M_{bat} = \frac{E_c}{e} \tag{31}$$

$$E_c = R \cdot v^2 \cdot \left(\frac{V^{\frac{2}{3}} \cdot \rho_{water} \cdot C_d}{2 \cdot \eta_{Motor}} + \frac{P_h}{v^3} \right) \tag{32}$$

$$M_{Shell} = A_{Total} \cdot \rho_{shell} \cdot t \tag{33}$$

$$A_{Total} = A_f + A_m + A_a A_{total} = A_f + A_m + A_a \tag{34}$$

7.1.3. Computational Processes

Once the user presses the calculate button, the program converts the user inputs into metric units for ease of calculation. From there, it calculates using a series of nested loops to ensure that each permutation is met. The procedure is outlined visually in Figure 28. Once the variables are in metric, the system can calculate net buoyancy. This is accomplished through the equations and process described in Section 7.1.2. Once net buoyancy has been calculated, the program can compare that calculated value to the permissible buoyancy range as set by the user. If within the range, the combination is deemed a success and saved. If not, the combination is deemed a failure and discarded. From there, if there are more combinations to test, the program will test them. Once all combinations have been tested, the program sorts the table of successes by the selected parameter. From there, all successes are written to an excel file, and the top 5 values are displayed in the user interface.

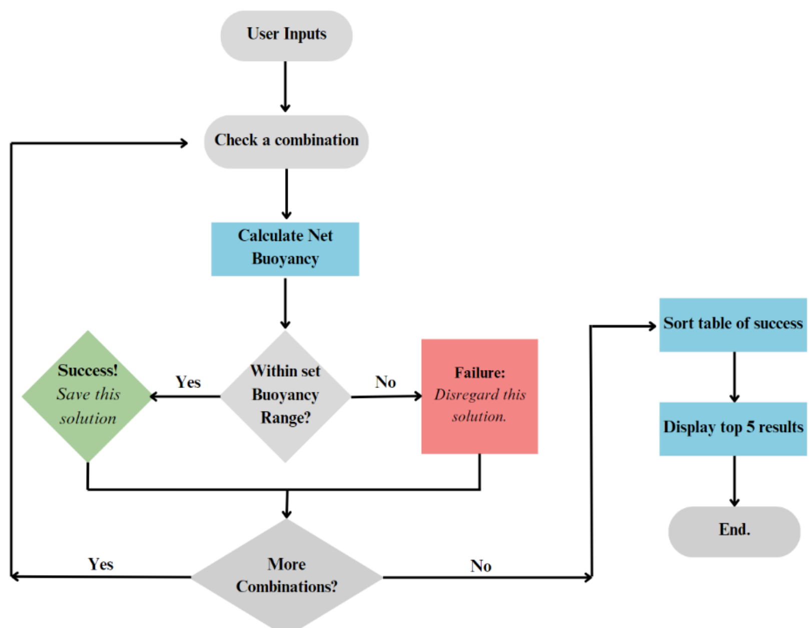

Figure 28. The Tool's Computational Process

7.1.5. User Accessibility

This tool includes user accessibility features specific to people not familiar with the tool and the input variables. These features include a 'Help' box, 'Progress Bar', 'Elapsed Time', and 'Re-Sort' functions in the UI.

Help Box

The help box provides various informative resources relating to the buttons and features included in the UI.

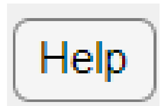

Figure 29. The Help Button

This button, located near the bottom of the tool, aids the user by providing explanations of the tool's functionality. The display that appears when the button is pressed is shown below in Figure 30.

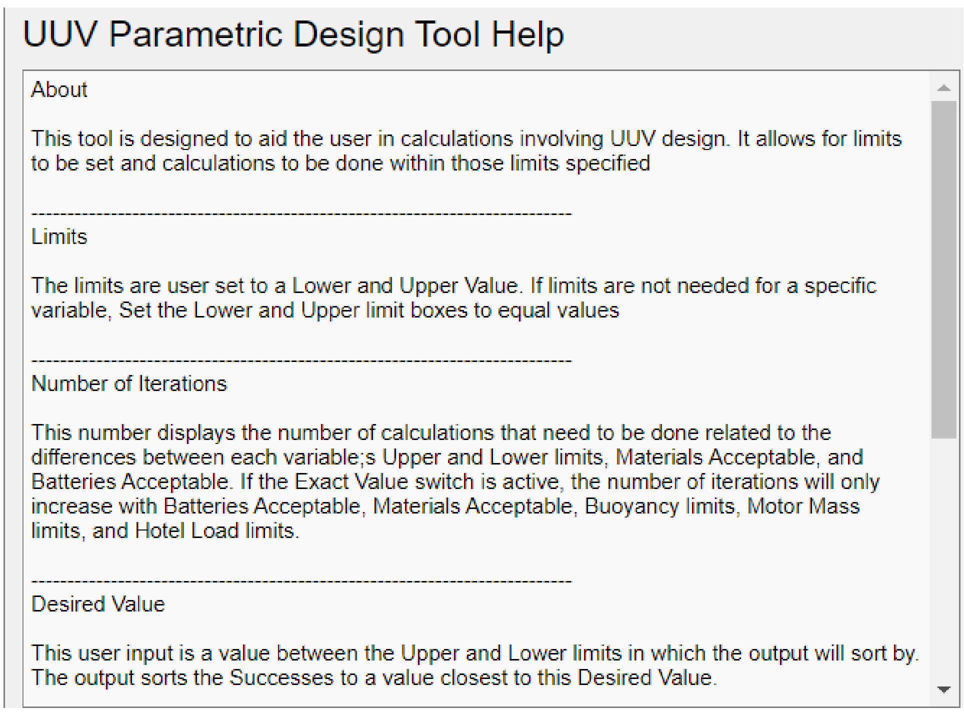

Figure 30. Contents of the Help Box

The contents of the help box summarize the tool's functionality and important features. These descriptions are intended to help inexperienced users who may be unsure how to use the tool.

Progress Bar

The progress bar provides the users with a visual cue to indicate that the program is running after the 'Calculate' button has been pressed. This progress bar disappears when all the calculations are completed.

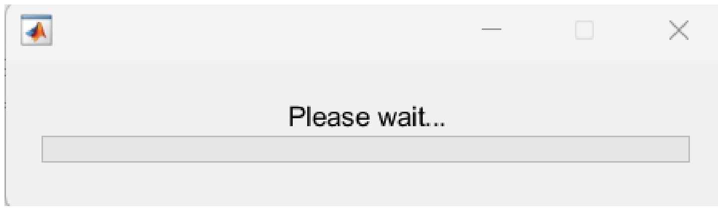

Figure 31. First Phase of the Progress Bar

Figure 31 shows the first message displayed to the user after the calculations begin. This indicates the beginning of the iteration calculations.

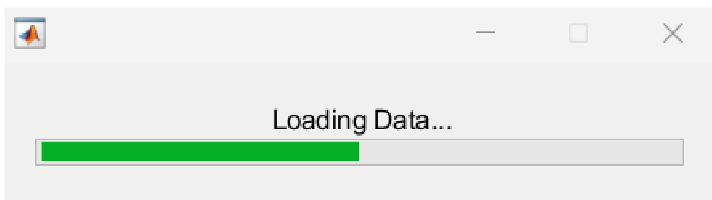

Figure 32. Second Phase of the Progress Bar

After approximately one tick of the first message, the users will see the bar change to Figure 32. This message remains up until the calculations hit the final loop, when it changes to Figure 33 below.

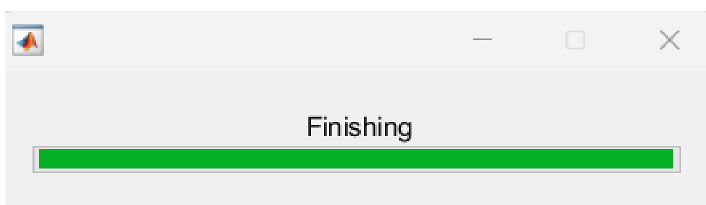

Figure 33. Third Phase of the Progress Bar

Elapsed Time

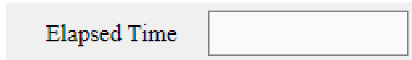

Figure 34. Elapsed Time Box

Figure 34 shows the 'Elapsed Time' box. This box displays a timer that starts as soon as the 'Calculate' button is pressed. This timer allows the user to see exactly how much time the program required to complete the required number of iterations on their specific computer. This is a quality-of-life and accessibility feature that allows the user to see whether their hardware would be able to efficiently handle a higher number of iterations. During testing, none of the team's computers could handle above roughly one million iterations.

Re-Sort

This function allows the user to re-sort the data after the calculations have been completed. The data is re-sorted through a file that gets saved to their computer at the end of the calculations, and this file is deleted and recreated once the 'Calculate' button is pressed again.

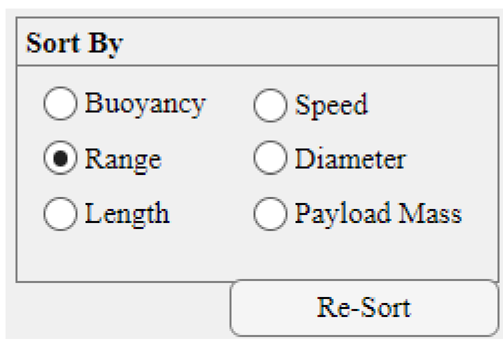

Figure 35. Sort By and Re-Sort Boxes

This button gives users control over the results they see based on their design priorities. The 'Re-Sort' button appears only after calculations have been completed at least once but the 'Sort By' button set is always displayed, allowing users to choose the value to sort by before beginning their initial calculations.

7.2. Validation & Testing

7.2.1. Comparison to UUV Data

We tested the accuracy of our tool by comparing its results to commercial UUV designs that are already known to be successful. If the tool works properly, it should generate feasible designs when the inputs are set to existing UUV data. We first gathered data on commercial UUVs from publicly available manufacturer data sheets. We researched two UUV designs and recorded data such as diameter, range, speed, battery capacity, etc. Our data collection was imperfect because some information is not published by manufacturers for proprietary or security reasons, and some of the tool inputs do not align with the data typically published by manufacturers. For example, UUV motor mass is not commonly published but is one of the parameters of our tool. Total UUV battery capacity is normally published and is accounted for in our tool's calculations but is not visible to the user. Therefore, we were unable to perfectly match the tool's inputs to the collected data. We entered as much data as possible and made reasonable approximations where possible, as outlined in the following sections on individual UUVs.

Bluefin-9

Table 5. Data Collected on the Bluefin-9 UUV

Model	**Range**	**Length**	**Speed**	**⌀**	**Battery Capacity**	**Weight**
Bluefin-9	24 nm	2.48 m	2 kts	≈ 9.9''	1.9 kWh	155 lb.

This UUV is manufactured by General Dynamics and all data in Table 5 is researched from public datasheets on their website (*Bluefin-9 Unmanned Underwater Vehicle (UUV)*, n.d.). The UUV uses several sensors and electronics, including a Sonardyne Solstice sonar with a power draw of 18 W (Sonardyne, n.d.). We were unable to find data on the other sensors used in the UUV, so we approximated the total hotel load as being from 36 W to 50 W. The UUV appears to be made from some form of composite, so we chose CFRP as the material. We assumed the design was within 15 kg of net buoyancy because most UUVs aim for net buoyancy. Unknown parameters, including motor mass and payload, were given large ranges in our tool's inputs. Table 6 shows the output from our tool that appears closest to the commercial UUV.

Table 6. Our Tool's Results When Replicating the Bluefin-9.

Buoyancy	**Range**	**Length**	**Speed**	**⌀**	**Batt. Type**	**Payload**	**Material**	**Hotel Load**	**Motor Mass**
10.32 kg	24.2 nm	2.48 m	2 kts	9.9"	Lithium-ion	98 lb.	CFRP	36 W	42 lb.

The results meet all the input criteria, but the weight values of the UUV appear optimistic. With the given battery capacity and our estimated Lithium-ion energy density, the battery weighs about 42 lb. With the payload and motor mass calculated above, the UUV weighs about 182 lb before adding the weight of the hull. This weight exceeds the 155 lb total weight given for the Bluefin-9, so there is some discrepancy in our results. This could come from a variety of sources, and we are not sure what caused it. It is possible that one of the assumptions or simplifications we used does not approximate UUVs of this shape, weight class, etc. well.

Remus-100M

This UUV is manufactured by HII and all data is drawn from public data sheets on their website (*Remus UUVS*, n.d.).

Table 7. Data Collected on the Remus-100M UUV

Model	**Range**	**Length**	**Speed**	**⌀**	**Battery Capacity**	**Hotel Load**
Remus-100M	67 km	1.85 m	3 kts	0.19 m	1.5 kWh	95.3 W*

We attempted to estimate the weight and hotel load of the sensors and payload used on the Remus-100M, but we were unable to access the complete data. We found that the sonar requires up to 11.5 W, the communications equipment requires 60 W, the location equipment draws 12 W, and the transducer uses 11.8 W (*Sea Scan® ARC*, n.d.; *Micromodem*, n.d.; *Phins Compact C3*, n.d.; *Tasman DVL*, n.d.). This is a total of 95.3 W. This does not account for all the equipment in this UUV, but the sensors and other equipment listed here also likely do not run constantly, so we used this number as the hotel load value in our tool. We did not find what material the Remus-100M is made from, so we left the tool free to use Aluminum or CFRP. The two results that appeared the most reasonable and closest to the Remus-100M are shown in Table 8.

Table 8. Two of Our Tool's Results When Replicating the Remus-100M

Buoyancy	**Range**	**Length**	**Speed**	**⌀**	**Battery Type**	**Payload**	**Material**	**Hotel Load**	**Motor Mass**
-0.83 kg.	67 km	1.85 m	3 kts	0.19 m	Alternate Lithium-ion	21 lb.	CFRP	95.3 W	42 lb
-0.83 kg	67 km	1.85 m	3 kts	0.19 m	Alternate Lithium-ion	43 lb,	CFRP	95.3 W	20 lb

The tool appears to have recreated this UUV well. The UUV designs shown are approximately neutrally buoyant, meet all known criteria, and appear to have reasonable payload and motor mass values. The main difference between the two results is how the weight is distributed between the payload and motor, reflecting the design freedom inherent in UUVs.

7.2.2. User Testing

User testing was done by the entire team, with each member testing at least 5 people who may or may not have experience with UUVs. This testing was performed far enough before the completion of the project to allow the team to incorporate the results into the finalized tool. After each person completed testing, they were asked to complete a survey with several categories: “Ease of Use”, “Bugs Encountered”, and “Thoughts and Feedback/Additions you would Make”.

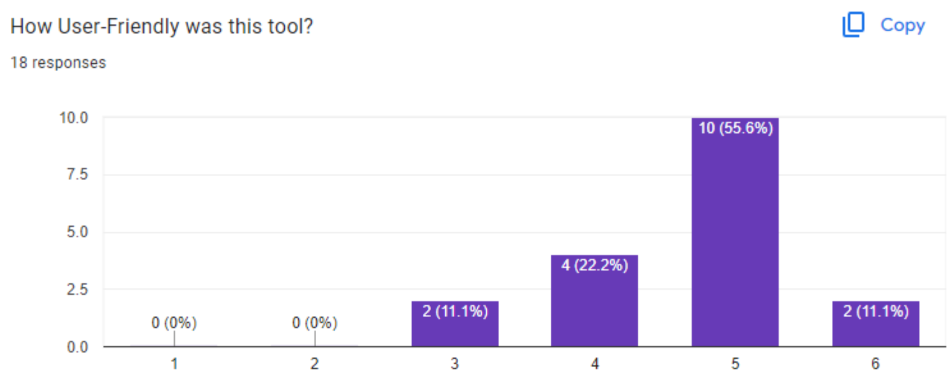

Figure 36. Graph of Results from the Survey

According to the responses gathered, the user friendliness of the tool was above the median and averaged around a 4.44. Most of these responses come from individuals who do not know exactly what a UUV is and do not have experience with the considerations necessary to design one. Given that the tool is intended for our sponsors, who have extensive experience with UUVs, concerns over whether users will have enough prior knowledge to use the tool successfully are minimal.

The survey also provides short answer questions. The second question reads, "Is there anything that you would add or change to improve this tool?" and had the following statements as the most common responses:

- Color Coding the Results
- Errors that occurred/possible solutions
- The 'Help' Button location and content
- Tooltip help

The third question read, "Did you encounter any bugs? If no, write N/A. If yes, please explain more." And had the following statements as the most common responses:

- Zero solutions were found, giving an Error.
- One success found: no results given.
- Once calculate was pressed, inputs were reset to default.

The fourth question read, "Were you able to break this tool? If yes, please explain how." and had one common answer:

- Too many iterations caused it to break.

The fifth question allowed the testers to provide additional opinions about the tool and the survey. This survey showed that the tool was satisfactorily user-friendly. The number of issues reported was relatively low, and we resolved one of the commonly reported ones by adding a message box that appears if no feasible designs can be generated. This box helps prevents the user from being confused in the event that the tool does not output any results.

8. Results: Use Case Analysis

This section demonstrates a hypothetical case in which a user needed a UUV that met certain requirements. Any parameters for which the user had no desired value were left free. The user needed the diameter to be exactly 0.5 meters, the speed to be between 2 knots and 4 knots, the length to be between 1.5 meters and 3.5 meters, the motor mass to be between 10 kilograms and 25 kilograms, and the range to be at least 100 miles. The UUV also needed to accommodate at least 20 W of hotel load. Approximate neutral buoyancy was desired, but ± 10 kg was allowed. Figure 37 shows the exact inputs of the user based on their UUV's mission.

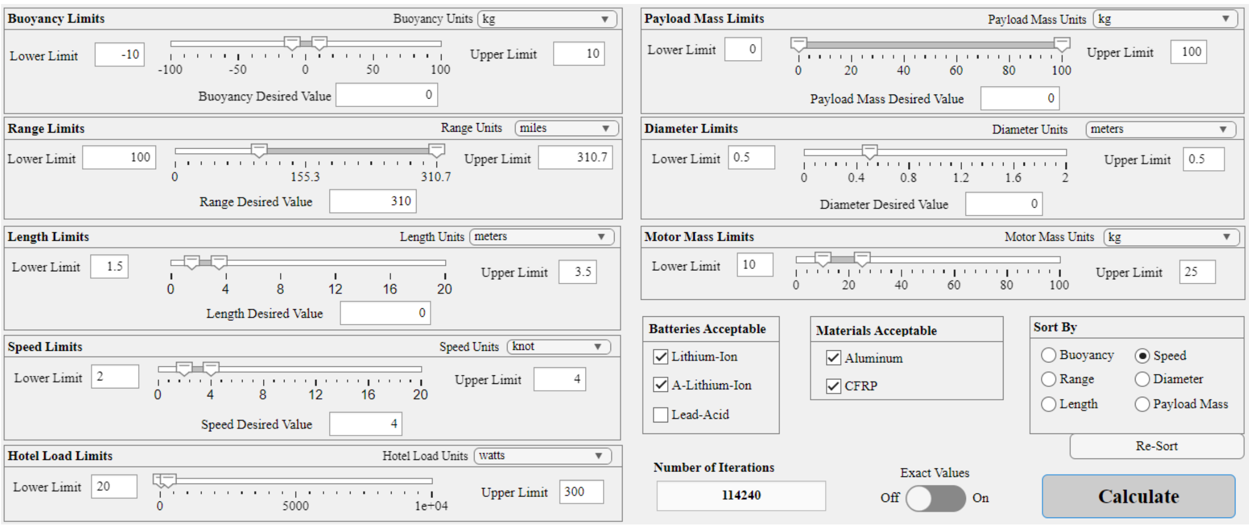

Figure 37. Exact User Inputs for Case Analysis

The tool generated 3144 successes, or designs that are feasible and meet all specified criteria. Figure 38 shows five of these results that have speeds at the maximum input value because the tool was set to sort by the speeds closest to four knots. All the results have parameters at the desired values or within the desired range. The user would then be able to select the design that best suits the desired purpose, choose to re-sort the results to view other possibilities, or run the tool again with different inputs.

Buoyancy: kg	Range: miles	Length: meters	Speed: knot	Payload Mass: kg	Diameter: meters	Battery Type	Material Type	Hotel Load: watts	Motor Mass: kg
7.482	100	1.5	3.9439	30	0.5	LithiumIon	Aluminum	20	20
2.482	100	1.5	3.9439	35	0.5	LithiumIon	Aluminum	20	20
5.1455	100	1.5	3.9439	35	0.5	ALithiumIon	Aluminum	20	20
7.482	100	1.5	3.9439	40	0.5	LithiumIon	Aluminum	20	10
-2.518	100	1.5	3.9439	40	0.5	LithiumIon	Aluminum	20	20

Figure 38. Five Generated Outputs with the Maximum Speed

9. Limitations

Despite the capabilities of our tool, it has several limitations. Three major limitations are the team's assumptions, the limited resolution of the tool's computations, and the simplifications we made to UUV shape.

9.1. Assumptions

To develop the tool, we completed extensive research to ensure accuracy. However, there is limited data on UUVs since much of the relevant technical information is kept confidential. This limited the accuracy of our tool and required us to make certain assumptions. UUV characteristics with limited data included the hull thickness, drag coefficient, and speed. Our tool assumes constant speeds and the shape of the battery was not accounted for. The user is limited to three battery types, two kinds of Lithium-ion and a lead-acid variety. In addition to limited batteries, the user only has two hull material options, Aluminum and CFRP, and the chosen material is uniform across the entire UUV hull.

9.2. Calculation Resolution

An additional limitation to our MATLAB tool is its resolution. Because the program increases by finite step sizes while iterating through the possible parameter values, as described in Section 7.1.3, it will not generate combinations with values that lie between these step values. Decreasing the step size would increase the quality of the results but also would increase the calculation time. The user could modify the code to increase or decrease the step size if needed, but there is no option to modify the step size through the UI. The default step sizes can be seen below in Table 9.

Table 9. Our Tool’s Step Sizes

Input	Step Size	Units
Range	10000	Meters
Length	0.5	Meters
Speed	1	Meters/Second
Payload Mass	5	kg
Diameter	0.1	Meters
Hotel Load	500	Watts
Motor Mass	5	kg

9.3. Shape

Another limitation to our tool is its geometric model. UUVs can be made in a variety of shapes and contain unique features designed for their specific missions. Our tool confines users to the generic, simplified shape researched from Liu et al. (2021b) and only allows modification to the total length and diameter. This limited changeability does not reflect the range of possible UUV designs, and the tool does not allow users to modify or add to the shape to match all their specific needs. If the tool is used to predict the design characteristics of UUVs that have drastically different form factors, the accuracy of the results will be decreased. However, the form factor that we used appears to be a highly generic one that should provide reasonably accurate results for many UUVs.

10. Discussion

10.1. Broader impacts

When creating a product such as the design tool described within this report, it is important to consider a variety of factors not directly included in the design process. Considered below are a variety of factors including the ethics of the project, its potential impact on the world, and its potential economic impacts.

10.1.1. Engineering Ethics

When considering the code of ethics of engineers, specific considerations must be given to Fundamental Principle I, Fundamental Canon 1, and Fundamental Canon 8. Both Principle I and Canon 1 relate to public safety, health, and welfare. We believe this project lives up to the standards set, but further consideration is offered in Section 10.1.2. Fundamental Canon 8 relates to environmental impact and sustainability, which is considered in more detail in Section 10.1.3. Other than that, none of the principles or canons are called into question for this project.

10.1.2. Societal and Global Impact

There are important questions to be asked about the global impact of this tool. The tool we built has the goal of designing UUVs, which have both civilian and military applications. With that in mind, we must discuss the advantages and disadvantages of the proliferation of this type of vehicle in both sectors.

UUVs can be very useful for civilian use. There are numerous options for UUV implementation, but seafloor exploration can be used as an example. Use of a manned submarine can be prohibitively expensive and time consuming. It would require a large crew and the boat would not necessarily be nimble enough to fit within desired spaces. UUVs, on the other hand, may be able to explore the ocean's floor cheaply (Ehler et al, 2020). This is due to the dramatic reduction in crew capacity and production costs. Additionally, UUVs drastically reduce the risk to human personnel compared to conventional manned submersibles. Because of this, our UUV tool could hopefully be used to enable a beneficial increase in civilian underwater capacity.

UUVs also have a wide array of military applications. They have been used for purposes including Intelligence, Surveillance, and Reconnaissance (ISR); Mine Counter-Measures (MCM); and Anti-Submarine Warfare (ASW) (French, D. W., 2010). Considering the benefits and negatives to United States and global safety is interesting and complex. It is true that some UUV applications like MCM are almost singularly defensive. Those can be seen as increasing the security of the United States and its citizens. In comparison, ISR can be used for either defensive or offensive purposes, and ASW is a primarily offensive tactic. The impact of this tool in military applications is yet to be determined and would depend on its exact usage within the military. If UUVs are only used defensively, the tool would most likely be seen as increasing global and United States safety. However, the use of UUVs for offensive tactics is more nuanced. In a mutually assured destruction setting UUVs could be seen as creating peace, but a more realistic interpretation would be that UUVs used for offensive purposes would contribute to an increased capacity for damage to people and property around the country and world. If our tool increases the capacity to produce such machines, it will also increase the potential concerns that come with UUV development.

Overall, our tool, and the potential increase in UUVs due to it, can be seen in two ways. On one hand, they may be seen as increasing the underwater capacity of humanity and contributing to the defense and safety of the world population. Others may view it as a potential increase in global destructive capacity. We believe and hope that the first interpretation is closest to the truth and have hope that humanity and governments around the world will use this technology to augment our capacity for good.

10.1.3. Environmental Impact

The environmental impacts of this tool will be minimal but worth discussing. If we assume that this tool will be able to make the design process of UUVs more efficient, and therefore the creation of new UUVs cheaper and more prevalent, there are concerns that the number of UUVs will increase and disrupt marine ecosystems. With this said, the marginal increases in efficiency offered by this tool are small compared with the total costs of UUV production. Therefore, the corresponding increase in seaborne vessels will also be relatively small. Additionally, we hope that this tool will help increase the operational effectiveness of future UUV designs, hopefully counterbalancing the potential increase in UUVs because fewer would be needed to achieve the same goals.

10.1.4. Code and Standards

There are no specific codes or standards for the type of work done during this project.

10.1.5. Economic Factors

This product will hopefully increase the economic potential of its users. The intention is that the tool will decrease the time and man-hours it takes to design UUVs. If that is the case, it will allow manufacturers to offer them at a reduced price because the associated costs have decreased. Alternatively, it would allow those same workers or companies to increase their wages or profit respectively, as the workers have become more efficient with their time.

10.2. Recommended Future Work

10.2.1. Variable Speed Calculations

An interesting possible improvement to the tool could deal with a more complex speed regime. Currently, the program assumes that speed is constant during the entire mission. More detail in how the speed varies over time would help gauge the true energy capacity needed for the vehicle. This could be done in a few ways. One way is allowing a few preset patterns, which could be modified multiplicatively by the user. Alternatively, you could ask the user to describe what percentage of time the UUV would operate at various speeds. These all complicate the calculations but could allow for increased accuracy if implemented.

10.2.2. Improved and Simplified Code

Our group's limited expertise in MATLAB has led to inefficiency in our code. Our code began as a simple prototype before being gradually expanded over four months. The code is therefore very lengthy and likely contains unnecessary lines. Simplifying the code will help the owner comprehend each line and ensure they can debug any section. In addition, there are many MATLAB functions that can condense some lines of our code. Utilizing certain functions will shorten the code and can help reduce run times. In conclusion, it is suggested that the code is simplified and improved to enhance tool functionality.

10.2.3. Increasing Shape Complexity

As described previously, the model used to define our tool's shape characteristics is simplified and does not include fins, antennas, a propellor, or other realistic features. Adding additional realism to the model would increase the accuracy of the tool's results. The difference would likely be most pronounced in the vehicle's drag coefficient and the calculations that depend on it. It may be possible to add correcting factors to the drag coefficient to account for the omitted UUV components, but that would require additional research, likely including experimental testing, that was beyond the scope of this project.

Additionally, our tool uses a single pre-defined form factor, and the shape of the UUV varies only in its diameter and main section length. We would ideally add the ability to calculate the volume, payload capacity, and drag coefficient across a wide range of UUV form factors and shapes. We would then allow the tool to iterate through a range of the values that define the form factor (e.g. curvature or cross-sectional shape) in addition to iterating through the UUV's diameter and main length, as it already does. This addition would help to more closely reflect the wide range of designs possible in actual UUVs. For example, a low-drag form factor is best if the user requires high speed and low payload, while a higher drag design could work well for lower speed but higher payload requirements. Introducing this variability would allow the tool to optimize UUV shape, further expanding the possibilities and increasing the likelihood of reaching a successful design for a given set of parameter ranges. However, more data would have to be found through research or experimentation to accurately calculate the necessary results for a larger range of UUV shapes.

10.2.4. Exploring the Addition of AI

There is great potential in implementing some form of AI into the tool. The benefits are clear to see. For example, deep learning can greatly shorten computation time and create bespoke designs. However, there are also major challenges with using AI in this type of computational tool. One problem is that generative AIs require a large data set to train on. There are no such datasets of viable UUVs, which makes it difficult to train a traditional generative AI for this specific purpose. Alternatives may still be possible but require more research.

10.2.5. User Guidance when Calculations Fail

Our program provides the user with feasible UUV designs if the parameters are set to suitable ranges. It also allows the user to input exact values for all parameters and evaluate whether such a UUV is feasible. However, the tool does not give good guidance to the user on how to obtain a feasible design if the chosen parameter ranges do not yield any successful designs, or if the exact inputs are not feasible. Under its current design, the user should increase the parameter ranges if the tool outputs no feasible designs, but the tool would ideally provide guidance on which parameter(s) should be changed and how far in order to most quickly reach a feasible design.

11. Conclusion

This project was an overall success that fulfilled its initial goals. We accomplished our objective of creating a tool that could generate successful UUV designs through parametric inputs and mathematical modeling. Reaching the final design was a long and difficult process that required numerous iterations, improvements and modifications. However, the result is a tool with a polished design and a smooth user experience, as well as extensive research behind it. Most importantly, it was produced under the guidance of our sponsors, was formed around their specifications and requirements, and achieved their approval.

This project posed many challenges to us, from the open-ended nature of the design prompt and the need to formulate our own operational method to our lack of expertise in MATLAB and its many features. We frequently encountered setbacks that required us to adapt by changing our methods or adjusting our objectives. The project challenged us to increase our knowledge and proficiency across a range of topics in order to overcome the difficulties it posed. We capitalized on the opportunities offered by this project, and not only fulfilled the project requirements but also gained valuable knowledge and experience while doing so.

We recognize the high potential for future development with this project. We are confidant what we have delivered a tool with a strong and versatile core functionality that will easily accommodate increased capabilities. We hope that future work will build on the results we have produced and further increase innovation within UUV design.

References

Aluminum 60601-O. (n.d.). MatWeb. Retrieved April 18, 2024, from https://www.matweb.com/search/datasheet.aspx?MatGUID=626ec8cdca604f1994be4fc2bc6f7f63

Ampatis, C., & Papadopoulos, E. (2014). *Applications of Mathematics and Informatics in Science and Engineering* (1st ed.). Springer. https://link.springer.com/chapter/10.1007/978-3-319-04720-1_1

Battery Module. (n.d.). Teledyne Marine. Retrieved April 12, 2024, from https://www.teledynemarine.com/en-us/products/Pages/gavia-battery-module.aspx

Bluefin-9 Unmanned Underwater Vehicle (UUV). (n.d.). General Dynamics. Retrieved April 12, 2024, from https://gdmissionsystems.com/products/underwater-vehicles/bluefin-9-autonomous-underwater-vehicle

Bluefin 1.5 kWh Subsea Battery. (n.d.). General Dynamics. Retrieved April 12, 2024, from https://www.teledynemarine.com/en-us/products/Pages/gavia-battery-module.aspx

Cai, Q., Browning, D. J., Brett, D. J., & Brandon, N. P. (2007). Hybrid fuel cell/battery power systems for underwater vehicles. In Proc. 3rd SEADS STC Technical Conference, Edinburgh.

Ehlers, Frank. (2020). *Autonomous Underwater Vehicles - Design and Practice - 13.8.2 Modular Design for Customizable AUVs.* (pp. 21,22). Institution of Engineering and Technology (The IET). https://app.knovel.com/hotlink/pdf/id:kt012DXNI1/autonomous-underwater/modular-design-customizable

Fomin, P., McCarron, M., Menz, S. M., Scott, C. A., Donohue, S., Louis, G. E., Smith, M. C., Gambhir, G. S., Harmon, J., Lane, D. K., Ott, D., & Spetch, T. (2005). Developing a paramagnetic design tool for lighter than air vehicles. 2005 IEEE Design Symposium, Systems and Information Engineering, 270–276. https://doi.org/10.1109/SIEDS.2005.193268

French, D. W. (2010). Analysis of Unmanned Undersea Vehicle (UUV) Architectures and an Assessment of UUV Integration into Undersea Applications (p. 149). NAVAL POSTGRADUATE SCHOOL. https://apps.dtic.mil/sti/citations/ADA531528

Gafurov, S. A., & Klochkov, E. V. (2015). Autonomous Unmanned Underwater Vehicles Development Tendencies. *Procedia Engineering*, *106*, 141–148. https://doi.org/10.1016/j.proeng.2015.06.017

Gitzendanner, R., Puglia, F., Martin, C., Carmen, D., Jones, E., & Eaves, S. (2004). High power and high energy lithium-ion batteries for under-water applications. Journal of Power Sources, 136(2), 416–418. https://doi.org/10.1016/j.jpowsour.2004.03.032

Good, M. (1989). Design and Construction of a Second Generation AUV [NAVAL POSTGRADUATE SCHOOL]. https://apps.dtic.mil/sti/citations/ADA230667

Laun, A. W. (2013). Development and validation of a conceptual design program for unmanned underwater vehicles [Thesis, Massachusetts Institute of Technology]. https://dspace.mit.edu/handle/1721.1/81608

Liu, Y., Yu, Z., Z, L., Tiantian, L., Dengxue, F., Zhang, J. (2021a). A fine drag coefficient model for hull shape of underwater vehicles. *Ocean Engineering. 236*(15). https://www.sciencedirect.com/science/article/pii/S0029801821007770?casa_token=TXEuNoh_ImIAAAAA:M8KfnzPY7dBohBBT6naRAxuljwo4Esrn-8kz_C1KJrih6E0jEAEcQpjEL2HZXmaAElkOd6J2fw

Liu, J., He, B., Yan, T., Yu, F., Shen, Y. (2021b). Study on carbon fiber composite hull for AUV based on response surface model and experiments. *Ocean Engineering. 239*. https://www.sciencedirect.com/science/article/pii/S0029801821012014?fr=RR-2&ref=pdf_download&rr=878aca95de7338a3

Mamagiannou, E., Pitenis, E., Natsiopoulos, D. A., Vergos, G. S., & Tziavos, I. N. (2022). GeoGravGOCE: A standalone MATLAB GUI for processing GOCE satellite gradient data. *Computers & Geosciences*, *166*, 105184. https://doi.org/10.1016/j.cageo.2022.105184

Mancuso, A. (2006). Parametric design of sailing hull shapes. *Ocean Engineering, 33*(2), 234-246. https://www.sciencedirect.com/science/article/pii/S0029801805001320

Medvedev, A. V., Vladimir, K., & Tolstonogov, A. (2017). *Depth Control Methods of Variable Buoyancy AUV*. 5. https://www.researchgate.net/publication/315382615_Depth_Control_Methods_of_Variable_Buoyancy_AUV

Mendez, A., Leo, T.J., Herreros, M.A. (2014). Current state of technology of fuel cell power systems for autonomous underwater vehicles. *Energies, 7*(7), 4676-4693. https://www.mdpi.com/1996-1073/7/7/4676

Micromodem. (n.d.). Acoustic Communications Group. Retrieved April 24, 2024, from https://acomms.whoi.edu/micro-modem

https://www.rockwestcomposites.com/downloads/T700SDataSheet.pdf

Nichols, P. R. K., Ryan, J. J. C. H., Mumm, H. C., Lonstein, W. D., Carter, C. M., Shay, J., Mai, R., Hood, J. P., & Jackson, M. (2020). *Chapter 7 Principles of Naval Architecture Applied to UUVs [Jackson]*. https://kstatelibraries.pressbooks.pub/nicholsproject/chapter/chapter-7-principles-of-naval-architecture-applied-to-uuvs-jackson/

Phins Compact C3. (n.d.). iXblue. Retrieved April 24, 2024, from https://www.ixblue.com/north-america/store/phins-compact-c3/

Puzaia, N. M., Ayob, A. F., & Arshadb, M. R. (2016). A Review on Recent Advancements in Unmanned Underwater Vehicle Design. *Science and Engineering*, *31*.

Remus UUVS. (n.d.). HII. Retrieved April 24, 2024, from https://hii.com/what-we-do/capabilities/unmanned-systems/remus-uuvs/

Saunders, H., & Windenburg, D. (1931). Strength of Thin Cylindrical Shells Under External Pressure. *Journal of Fluids Engineering, 53(2)*, 207-216.

Sea Scan® ARC Scout Mk II. (n.d.). Marine Sonic Technology. Retrieved April 24, 2024, from https://marinesonic.com/wp-content/uploads/2021/04/ARC-SCOUT-MKII-Spec-Sheet.pdf

Šimić, Z., Topić, D., Knežević, G. Pelin, D. (2021). Battery energy storage technologies overview. *International journal of electrical and computer engineering systems*, *12*(1), 53-65. https://hrcak.srce.hr/257521

Tasman DVL. (n.d.). Teledyne Marine. Retrieved April 24, 2024, from https://www.teledynemarine.com/en-us/products/Pages/Tasman.aspx

Technical Data Sheet. (n.d.). Torayca. Retrieved April 23, 2024 from https://www.rockwestcomposites.com/downloads/T700SDataSheet.pdf

28 V/55 Ah Lithium Ion. (n.d.). Eagle-Picher. Retrieved April 12, 2024, from https://www.eaglepicher.com/sites/default/files/LP33333%200623.pdf

Vehicle Batteries. (n.d.). SubCTech. Retrieved April 12, 2024, from https://subctech.com/ocean-power/vehicle-batteries/

Villalba Herreros, A. (2021). Direct Methanol Fuel Cell-based power plant for Autonomous Underwater Vehicles. Endurance increase and Carbon Dioxide management in a confined Environment [PhD Thesis, Universidad Politécnica de Madrid]. https://doi.org/10.20868/UPM.thesis.69261

Vyas, K. A., & Jamnani, J. G. (2012). *Development of IEEE Complaint Software "Economical Substation Grounding System Designer" Using MATLAB GUI Development Environment*. 335–346. https://www.proquest.com/docview/1323044648?forcedol=true