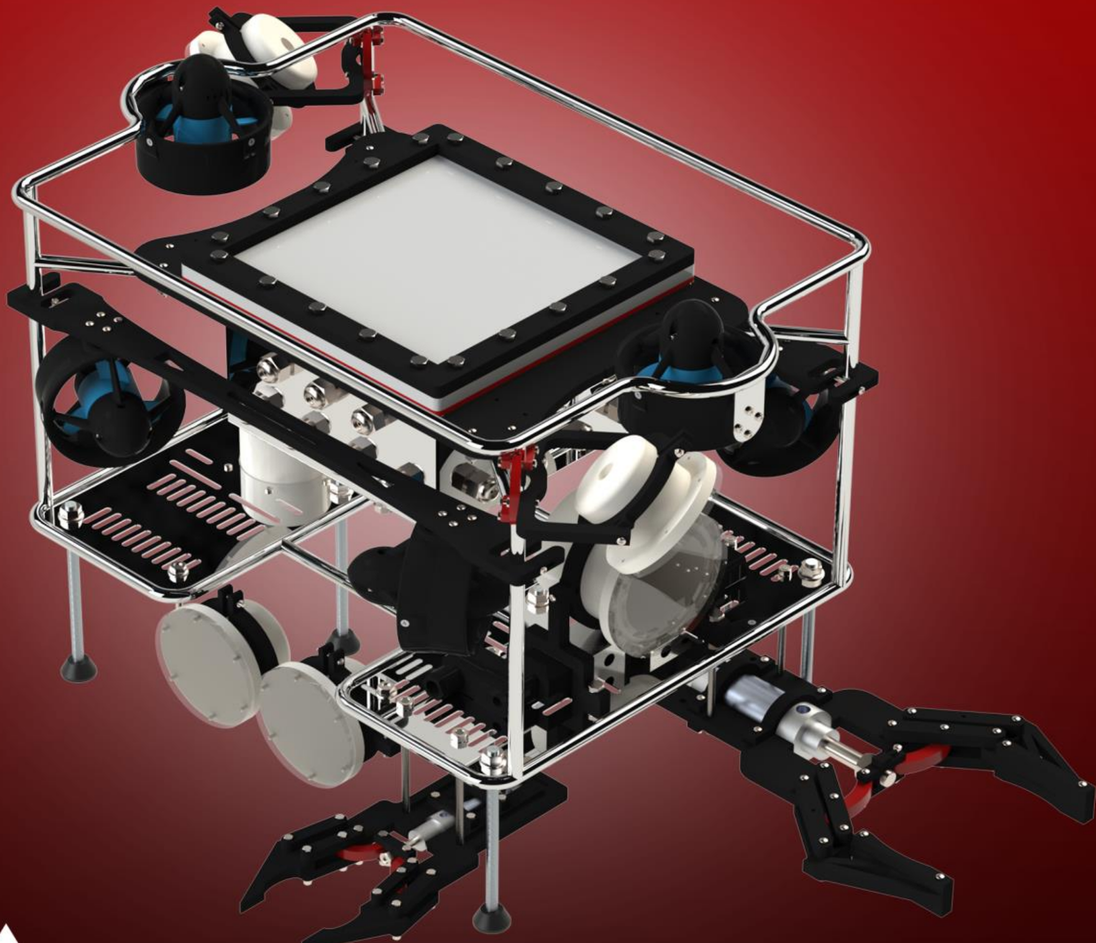

# TORPEDO PROPOSAL

2 0 2 3

# Contents

# D Software

## 1 Thrusters Vector Algorithm:

One problem we faced with our previous ROV's was heavily relying on basic motion directions (forward/backward/lateral left/right) along with rotational motion on x, y, and z axes. This made the ROV's diagonal motion hard and slow. This year, with the help of our R&D team, we found the solution to this problem in Jerzy Garus's paper which introduces a vector algorithm which enables us to move the ROV in any direction.

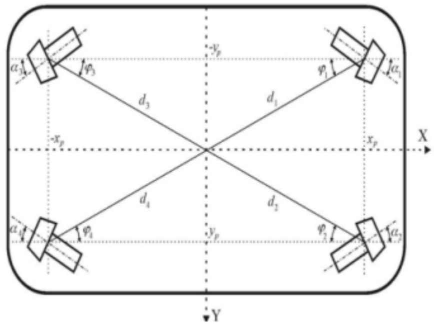

Figure 1: Polaris Horizontal Motors Configuration

## 2 Thrusters Speed

After getting the wanted speed from the pilot, this data will be transmitted to the motors board to be applied to the thrusters in the right order to enable Polaris move in a specific direction.
But assigning these speeds to the thrusters as they are, if the change in speed is large enough especially if the motors are in sleep mode before setting the speed for them, may lead to a high current draw, which may lead to the destruction of other components or affect the paths of the board itself.
At that moment comes the role of "soft start", which is to prevent the high current draw or malfunctioning of thrusters that could happen when a sudden change of the thrusters' speed occurs. We implemented our own software-based soft starter, by gradually ramping up the speed supplied to the thrusters.

# E Communication

## I Top side:

As for the communication between the TCU and Polaris, Ethernet UDP "User Datagram Protocol" was used. The communication is full duplex; hence Polaris can send and receive data simultaneously. ENC28J60 Ethernet controller was chosen to serve as an ethernet network interface for the used microcontrollers since they are equipped with SPI "Serial Peripheral Interface". On top of that, the "UIPEthernet" library was used to control the Ethernet modules


Figure 2: Ethernet Module

## II Underwater side

This year, we decided to use an ESP32 microcontroller instead of using three Arduino Nano. This reduced the error due to communication between the three Arduino Nanos. The ESP32 offered several advantages over the Arduino such as its higher processing power, its dual core architecture which allowed us to multitask and its built-in Wi-Fi which allowed us to upload different codes on Polaris' ESP32 wirelessly.

# F Vision System

## Camera Control

We have used a camera module that is made to operate and receive feeds from the cameras on the ROV. It can capture and process video feeds from various types of cameras. Moreover, it can be configured to work with either an external or local camera, depending on the specific requirements of the task. The module also provides advanced features for managing video streams, such as handling lost connections and automatically reconnecting to the camera when required

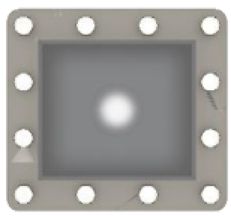

Figure 3: Polaris Horizontal Motors Configuration

Table 1: Comparison between word and latex

| Editor | Word | TeXstudio |
|---|---|---|
| Ease of use | Da keda keda el w7d msh by2dr el n3ma f3ln | NO COMMENT |
| Hacker | kol man dab w hab byktb word | Wins wesh |
| Recommended? | NO | NO (Da wna ba compile el table 3mly error msh fhmo LOL.... Y3NY EH A COMPILE TEXT) |