RoboSub 2023 Technical Design Report

San Diego State University: SDSU Mechatronics

Brendan Dennish, Cayton Larmer, Conner Sommerfield, Connor Larmer, Joseph De Vico, Ken Ishio, Tran Ly

***Abstract* – For RoboSub 2023, SDSU Mechatronics will be using Scion our new AUV for our Competition Goals. The only task goal is to pass the entrance gate and qualify. As a stretch goal, we intend to attempt other tasks as time allows. Our competition goal this year is to focus on designing a completely new software and electrical infrastructure to become the foundation for future success. In general, physical designs and software architecture are focused on becoming more integrated, smaller, and standardized.**

I. Team Competition Goals

For this year's RoboSub 2023, the goals for San Diego State University Mechatronics are straightforward. Our goal is to qualify and successfully enter the semi-finals by entering the starting gate with one vehicle, our new autonomous underwater vehicle (AUV) named Scion. We have decided to focus on creating a new foundation in hopes that the next generation of students can build upon it and be in the best position to succeed. We have started working on completely new software and electrical framework; in addition to introducing a new AUV mechanical design that has never been used at RoboSub. That said, we are well into the development of our systems and have confidence that our goals will be met. We have chosen reliability over complexity by focusing on this one task. To complete our goal of the validation gate, we are relying on our computer vision (CV), control systems, and navigation.Focusing on making these systems reliable translates to better success rates for all other tasks, so we will not be quick to add complexity before making the gate task very reliable.

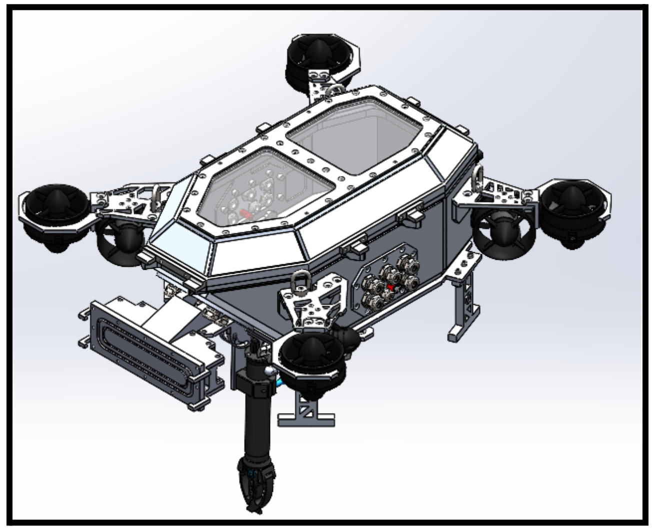

Fig. 1: CAD Rendering of Scion

Our new AUV with relatively new software and electrical architecture will not be developed enough to make a perfect course run judging by the previous year's competitor performances. As such, we are approaching the course with reasonable goals such as passing the gate. As a stretch goal, we aim to attempt more tasks this year after careful system testing with our remaining development time. This includes intersub communication which will call upon our older AUV Perseverance once again to swim together with Scion.

II. System Design Strategy

Our design strategies for passing the entrance gate are split into three different categories: Software (A), Mechanical (B), and Electrical (C)

A. Software Design Strategy

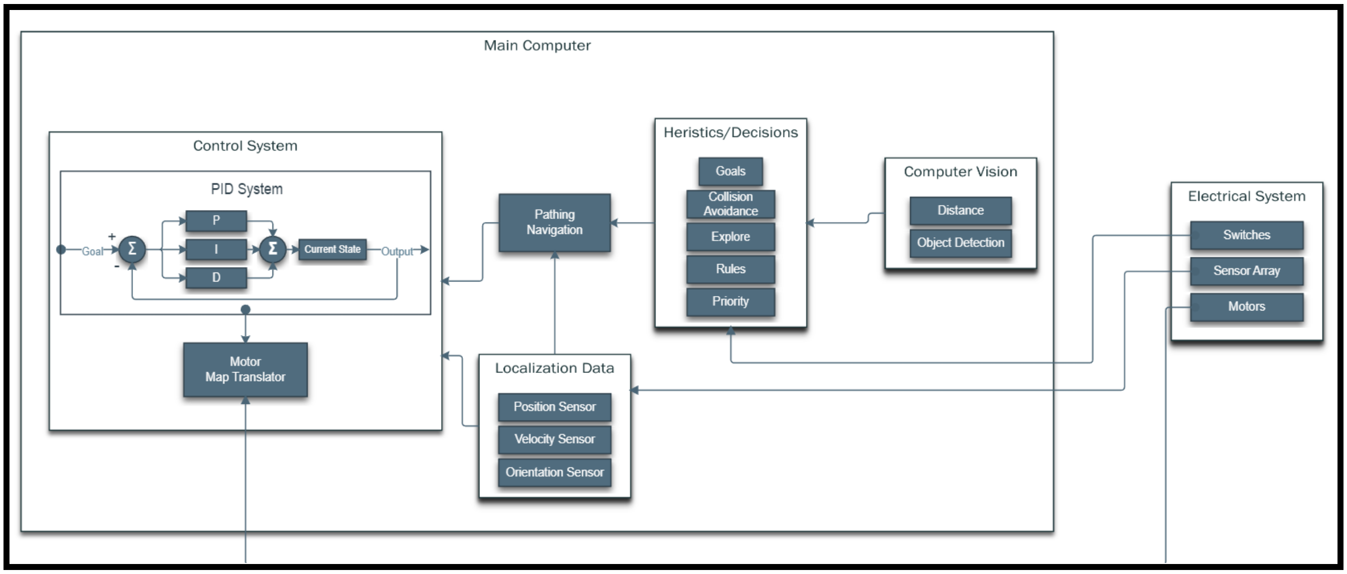

Fig. 2: Software Architecture Block Diagram

We believe the RoboSub competition heavily relies on computer vision (CV), control systems, and localization. We will be using machine learning (ML) based object detection architecture called "you only look once" (YOLO) version 5 to reliably detect the gate in real-time with high accuracy. Especially with the gate, being able to localize the gate will be the first step to passing through it. We have also added a stereo camera that can act as a visual distance sensor device to help calculate navigation through the gate. More importantly, the stereo camera enables more efficient collision avoidance by telling us the closest object and how far it is from the vehicle. Headlamps were added to increase available light to the camera sensors to increase the reliability of vision-based sensors.

A control system using Proportional Integral Derivative (PID) will be implemented to keep the AUV stable. The algorithm corrects the physical error between a desired state and the current state provided by the sensors by determining the correct amount of thrust to be given to the motors to hold or change position, orientation, or speed.

The sensors we are depending on are currently a Doppler velocity logger (DVL) which gives velocity vector data and an attitude and heading reference system (AHRS) which gives orientation data. By combining DVL and AHRS data we can calculate relative position. We are using an underwater pressure sensor to measure depth position.

For our infrastructure, we placed a greater focus on the integration of ROS2 with the submarine's core systems. We selected ROS2 primarily because of the plethora of communication, development, and debugging tools it provides. Our software architecture consists of various interdependent 'nodes', each being responsible for a specific task. These nodes are able to communicate to one another through ROS2– i.e. the PID controller node is able to receive commands from the ‘brain’ node, which makes decisions based on sensor data sent from various sensor nodes. This model allows for granular control over how the submarine interprets and responds to input; it also reduces individual node complexity, as a node should be designed to perform only one task. Overall, ROS2 provides our software with greater flexibility and robustness, as nodes can be developed and tested independently from one another.

The decentralized design of ROS2 is beneficial for the higher-level abstraction and decision-making layer of our software. However, in the past, decentralization has harmed the low-level communication between software and hardware. At this level, data reliability and integrity are key. We ultimately opted for a centralized software-to-hardware ‘bridge’ for communicating sensor, motor, and configuration data. Given these requirements, we adapted the Controller Area Network (CAN) protocol to act as a high-speed link between the main computer and the embedded system. Through this design, almost all sensor and motor data is passed through the embedded system prior to reaching the main computer. This allows for low-level processing of critical information prior to it reaching ROS2.

At a higher level, we opted for a single modular ROS2 node to handle all embedded communication. The main motivations for this design were: reducing build-time and complexity, encouraging compliance with our custom protocol, and maintaining a clean codebase. A CAN driver acts as a two-way translator between ROS2 and hardware. The CAN driver receives, sorts, converts, and publishes sensor data and commands to and from ROS2. We designed it to be configurable, allowing for certain functions of the driver to be disabled without recompiling the node. This flexibility was implemented to speed up the development and debugging of our software.

B. Mechanical Design Strategy

The primary focus of mechanical development this season has been on updating and finishing the AUV Scion. Scion was originally designed during the 2020 season as a successor to the AUV Perseverance, with a focus on reduced weight and improved maneuverability; however, due to process errors during the 2021-2022 season, the vehicle was not completed in time. Scion now serves as the team's primary vehicle, with Perseverance acting as an accessory vehicle for the intersub-communication stretch goal.

Scion and Perseverance are fundamentally very similar vehicles. Scion presents a significant weight improvement over Perseverance due to its smaller volume, facilitated by the improved density of electronic components. This reduced weight contributes to lower thruster power consumption, prolonging battery life and reducing heat emissions inside the volume of the vehicle. These factors enable Scion to move swiftly between tasks and contribute to optimizing runtime; meanwhile, Perseverance's larger internal volume makes it better suited for rapid prototyping and testing purposes.

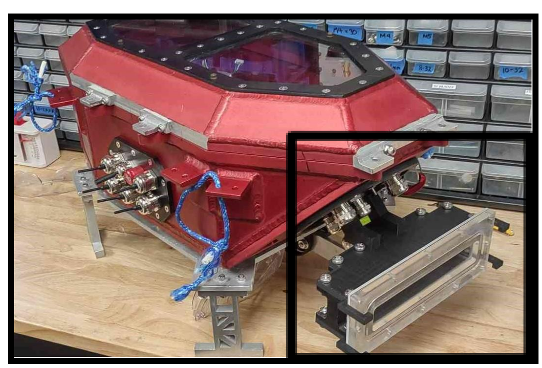

Fig. 3: Discrete Camera Enclosure

Reliable computer vision is a requirement for completing all tasks at the competition. To this end, Scion was retrofitted with a new external camera enclosure to accommodate the use of the Zed 2i Stereo Cameras (See Fig. 3). Scion was originally designed to use an incomplete gimbal-mounted camera by a previous team. However, we found that the stereo cameras are quicker to implement and provided many benefits the old system did not. While Perseverance was originally designed with a stereo camera and the geometry of the front of Scion did not suit the addition of the camera. Rather than undergo an expensive and risky machining operation on the hull of the vehicle, it was instead decided to custom-build a discrete enclosure to hold the camera. This permitted the team the multiple cycles of design and prototyping required to produce a successful component, and ultimately the use of an external camera enclosure stands as one of Scion's greatest upgrades compared to the original design.

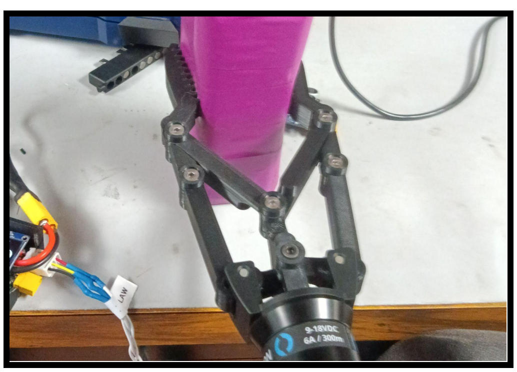

Fig. 4: Newton Subsea Gripper with Modified Jaws.

Our stretch goal of manipulating objects underwater is an important part of this year's competition; With the Bin and Octagon tasks being worth a large number of points. As a consequence, one focus of this year's mechanical effort was improving the vehicles' manipulator. Previously, a stock Newton SubSea Gripper was used over a custom build because it provided simplicity and repeatability for software development; however, one recurring pain point was the poor closing range and angle of the Gripper jaws. This made the gripper tasks more difficult for the vehicles, as they were required to be precisely positioned around the target objects in order to interact with them. Past experience shows that designing a manipulator from scratch is a taxing ordeal, so our solution this year was to replace the jaws of the Gripper to accommodate a wider range of motion (See Fig. 4 above). This improves our consistency at the gripper tasks without sacrificing the simplicity and repeatability that made the Newton attractive, allowing the team to target multiple high-point value tasks with a greater degree of confidence.

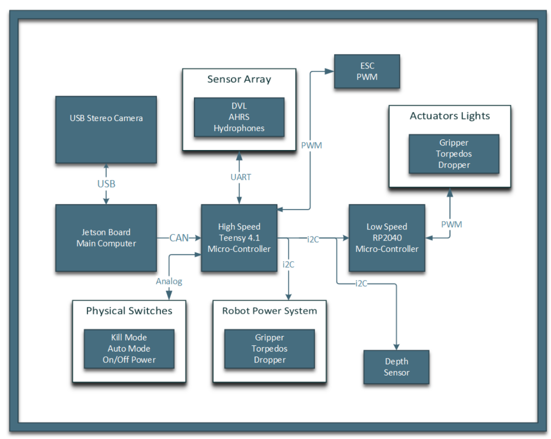

Fig. 5: Electrical Block Diagram

C. Electrical Design Strategy

The totality of the electrical system has been overhauled, with a focus on safety, ease of use, reliability, reduction in size, and future expandability. In addition to passing the entrance gate, our vision is to have an electrical system that is highly integrated in order to reduce the size, increase reliability, and reduce the development time of future vehicles.

Our Robot Power System (RPS) is a custom-designed circuit board that will be used to power robots such as Scion using two batteries. We selected 99-watt-hour batteries to ensure we can transport them on airplanes and still have enough capacity to last more than 20 mins during competition runs. The system's main function is to balance and split the load of the robot on both batteries ensuring that batteries stay balanced and healthy by providing real-time feedback to the main computer about the current state of the batteries and the power system so it can be detected preventing damage to the AUV. The RPS system enables hot swapping of depleted batteries without turning off the robots, decreasing development time. The circuit is capable of handling more than 90A peak load using 40mm PCB traces and 2 oz thick copper layers. The system will include integrated triple high-output DC-to-DC converters. This is to ensure Scion and any electrical additions will have the required power to operate the power-hungry thrusters, computers, and various components.

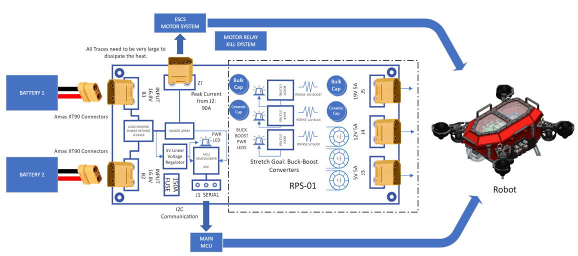

Fig. 6: Robot Power System (RPS)

Designed in tandem with our RPS, our embedded systems boards were designed to fit compactly together, reducing problems with too many wires and overly large systems. The embedded system boards act as the motherboard for data processing and communication before reaching the main computer. Although not specially designed for the gate task, the embedded system indirectly affects success by improving development time, reliability, and control systems communication of sensors.

Following the rest of the electrical system overhaul all embedded processing, control, and interfacing have been developed from the ground up. The system is divided into 2 modules or "nodes": the high-speed T4, and slow-speed RP2040, which can effectively divide the tasks of motor control, data collection, data processing, and data delivery in an effective and efficient manner. Both nodes have been architected such that rapid system expansion can be supported, and new sensors or outputs can be added with minimal development time.

Interfacing with the system is done through a standard CAN 2.0A bus operating at 500kHz. A fully in-house developed protocol standard is in place, which has allowed the cross-development of applications and embedded software to occur in parallel with little confusion.

Another major focus is safety. In the event of a control loss, program stall, communications error, or other adverse condition all moving devices will return to a safe state. This is ensured in totality by an absolute ban on blocking statements and strict control over ISR priority levels and flow. Watchdog timers are present to maintain both overall system states, and ensure valid controls arrive within a set time.

Deterministic data accesses are provided by an emulated virtual memory space (VMMIO) provided by the high-speed T4 embedded node and accessible to the applications processor via the CAN bus. The VMMIO provides massive benefits for development as time goes on, which has already assisted greatly as new technologies have been added to Scion over the year.

Utilization of parametric nested jump tables allows new hardware to be added and integrated into the memory space in minutes, instead of days or weeks. With a suitable and reliable base hardware and real-time firmware system in place, coupled with ease of integration for the applications software provided by a robust in-house developed kernel driver, pool testing has been made much more productive. It is our view that providing a robust base not only allows this year's vehicles to do well meeting at least the basic requirements to qualify, but also supporting future team goals necessitating refinements rather than rebuilding systems.

III. System Testing Strategy

Our testing strategy to pass the entrance gate follows the idea of the more time you spend testing, the better your product will be. We have secured about 8 to 20 hours a week of pool reservation in order to maximize our testing time. To validate that we pass the gate with high success rates. We need to perform full system testing and subsystem testing leading up to it.

Mechanically, we need to verify the vehicle does not ingress water at 16ft. The vehicle will be held at 16 ft underwater for 30 mins to assure no leaks will arise. With all electrical components inside, the vehicle must be slightly positively buoyant and relatively balanced. This is done by watching the sub float to the surface on its own with its full payload. Our method of correction is trial and error using weights and buoyancy foam.

Electrically, we need to verify that the kill button works properly, the software is receiving sensor data, and the motors are receiving commands from the software. This can be done out of water. Simply pressing the Kill Button should cut off power to spinning motors. Since the kill switch is a direct connection, we do not have to worry about software issues. Verifying electrical and software communications is as simple as running the motors.

Software needs to test control systems, computer vision, and navigation. Ideally, all tests should be in the water to emulate competition conditions as closely as possible. Since software heavily depends on mechanical and electrical systems. Software tests will be the system test to fully validate success autonomous completion of the entrance gate task.

Starting with the vehicle control system, the vehicle must hold a position reliably. We will apply external forces and the vehicle returns to its original position. Computer vision should detect the gate and give it an identification number, distance, and orientation that is visible on our sensor feedback graphical display. Decision making (brain node) and navigation together should bring the vehicle through the gate without touching it. If the vehicle can perform this task at a 90% or higher success rate, we will consider the vehicle validated and tested to fulfill our goal of passing through the entrance gate.

IV. Acknowledgments

The Mechatronics team would like to thank the SDSU Engineering Department, SDSU Division of Student Affairs, and Campus Diversity. The team would also like to thank faculty advisors Theresa Garcia, technical advisor Dr. Sungbum (John) Kang, and finance coordinator Craig Winton for their outstanding administrative support. We extend a heartfelt thank you to Mike Lester of the SDSU machine shop for his ongoing assistance and advice on fabrication matters. We would also like to thank our corporate sponsors: Altium, Solidworks, and SDSU CESC.

Appendix A: Component List

COMPONENTS	VENDOR	MODEL	SPECS	CUSTOM/ PURCHASED?	COST (TOTAL)	YEAR OF PURCHASE
Vehicle Hull	Metal Masters (Welding)	Custom	N/A	Custom	$7500	2022
Submersible Cord Grips	McMaster Carr	Brass, ½"NPT for 0.25" diameter wires	IP68	Purchased	$50 ($750)	2022
O-rings	McMaster Carr	NBR 70 Durometer	N/A	Purchased	($200)	2020
Stat-O-Seals	Parker Hannifan	600 Series ⅞"	N/A	Purchased	N/A	2023
PassThrough	Blue Robotics	Blue Robotics PassThrough Light	N/A	Purchased	N/A	Before 2020
Fastening Hardware	McMaster Carr	Flat head hex drive M4 bolts and nuts	N/A	Purchased	N/A	Before 2020
Non-Actuating Buttons	Other	N/A	N/A	Purchased	N/A	Before 2020
2x ESCs	HolyBro	Tekko32 F4	4x 30AMP	Purchased	$200	2023
T200 Thrusters	Blue Robotics	T200	N/A	Purchased	N/A	Before 2020
Speed Controller	Custom	Custom	N/A	Custom	0	2023
High Level Control	Custom	Custom	N/A	Custom	0	2023
Manipulator	Blue Robotics	Newton Subsea Gripper	N/A	Purchased	590	2022
Battery	Hoovo	4s	91.76Wh	Purchased	$116	2023
Custom Power System	N/A	Custom	N/A	Custom	$250	2023
Main Computer	Nvidia	Jetson Orin	N/A	Purchased	$2370	2023
External Comm Interface	Custom	Custom	N/A	Custom	Custom	Custom
Programming	C++, C, Python	C++, C, Python	N/A	C++, C, Python	C++, C, Python	C++, C, Python

Language						
Doppler Velocity Log and Inertial Measurement Unit (IMU)	Waterlinked	A50	N/A	Purchased	$7000	2023
Cameras	Stereolabs	Zed 2i	N/A	Purchased	$550	2022
Sonar	N/A	N/A	N/A	N/A	N/A	N/A
Hydrophones	Aquarian Hydrophones	AS-1	N/A	Purchased		
Manipulator	N/A	N/A	N/A	Custom	N/A	N/A
Algorithms: vision	Yolo V5	Yolo V5				
Algorithms: acoustics	Custom	Custom	Custom	Custom	0	2023
Algorithms: localization and mapping	Custom	Custom	Custom	Custom	0	2023
Algorithms: autonomy	Custom	Custom	Custom	Custom	Custom	Custom
Open source software	Open Robotics	ROS2	N/A	N/A	0	2023

V. REFERENCE

No References