RoboSub 2020: VantTec Technical Design Report VTec U-III

Pedro Sanchez, Alejandro Gonzalez, Melissa Sanchez, Ivana Collado, Alejandra Monsivais, Alonso Ugarte, Pedro Gamez, Gabriel Leal, Fernando Resendiz, Cristina Garcia, Diego Martinez, Saul Bermea, Gabriel Bermudez, Gerardo Berni, Andres Sanchez, Edison Altamirano, Penelope Ceron, Roberto Mendivil, Sebastian Martinez, Rogelio Salais, Nadia Garcia, Jorge Medina, Ernesto Monroy and Leonardo Garrido
Tecnologico de Monterrey
Monterrey, Mexico
vanttecmty@gmail.com

***Abstract*—The overall strategy, vehicle design elements, and experimental result from VantTec for RoboSub 2020 are presented. The strategy was constructed as a two-year plan, and from experience from RoboBoat competitions. The higher priority was given to have a robust and reliable hardware and software system to use as a base for future editions, and to develop autonomy capabilities such as object and sound detection and localization, and path-following control. Hence, the Gate and Buoys challenges are the priority tasks to tackle, as they do not require any additional hardware components than the base system. Members were divided in mechanical, electrical, and software areas, with a balance between software or hardware experience. Moreover, the main mechanical and electrical design elements are the frame, enclosures, PCBs, and sensors and actuators, which bring the desired robustness and reliability. Likewise, the system architecture design connects the required components for hardware and software communications, achieving determinism and task prioritization. The software architecture design allows for modular integration of the required software components. Next, computer vision and path-following control algorithms allow the VTec U-III unmanned underwater vehicle to perceive obstacles and navigate through planned waypoints. Furthermore, as the COVID-19 pandemic grew, the proposed strategy adapted to exchange manufacturing time for design time, and all of the challenge-oriented hardware components were designed. Similarly, the course approach was fully implemented in simulation, where results show successful completion of path-following control, and the Gate, Buoys, and, additionally, Torpedo competition tasks. Finally, results indicate the computer vision algorithm is capable of detecting the proposed RoboSub 2020 obstacles.**

***Index Terms*—Unmanned underwater vehicle, robotics, autonomy, computer vision**

I. Competition Strategy

This section discusses the team member organization, the course strategy, and the engineering design strategy.

A. Team Organization

VantTec has participated in RoboBoat competitions since 2017, and the team decided to participate in the RoboSub 2020 competition in the Fall 2018 semester, as a two-year project. During the first year of the project, only 5 members composed the team, using that year to research and develop a systems engineering strategy. Thus, the team could learn about the required hardware (given the lack of experience in the underwater environment), and finish the first design phase, beginning manufacturing and prototyping in the Fall 2019 semester. At the beginning of Fall 2019, sub-teams were created for mechanical, electrical, and software areas, composed by members from mechatronics engineering, digital systems and robotics engineering, mechanical engineering, and computer science majors. Finally, sub-team leaders have the higher experience in their respective areas, and the number of team members with hardware or software expertise is balanced. Additionally, VantTec, as an established robotics team, already has members responsible for group management, finances, and media.

B. Course Strategy

The two-year preparation allows the team to attempt a strategy capable of solving all of the challenges. However, as a new RoboSub team, and with previous experience as a rookie team in similar competitions, the strategy instead attempts reliability and robustness as a basis. This approach arises from the fact that VantTec has worked better while attempting a technically-and-time-feasible strategy, rather than overloading the members with responsibilities to try and tackle all of the competition tasks. Thus, even while the intention to design and manufacture all of the necessary components, and to think of solutions for every challenge, exists, higher-priority is given to the UUV platform and tasks that do not require additional actuation (apart from UUV motion control).

Hence, taking into account the RoboSub 2020 task ideas, the higher-priority challenges are Gate and Buoys. Thus, the goal of the first RoboSub participation for VantTec is to successfully complete two challenges, with a reliable hardware and software base system. This system will work as a starting point for future RoboSub editions.

Moreover, a simulation environment was planned to be built in the Spring 2020 semester to test the algorithms and software architecture. The simulation environment became a priority as the COVID-19 pandemic grew, allowing the team to attempt additional courses. The Torpedo task is also attempted in simulation, and the time selected for manufacturing and in-water testing was instead occupied by the mechanical team to further work on the design of the additional challenge-oriented hardware subsystems (torpedo launcher, marker dropper, and robotic arm gripper).

C. *Engineering Design Strategy*

The design engineering strategy relied on a year of research for members to learn and gain experience on the submarine hardware requirements. During this year, ideas were developed for the overall systems engineering approach, including full-design concepts, frames, required sensors and actuators, and challenge-oriented subsystems.

To follow the team strategy, reliability implies the UUV design protects the electronics from water, motion is robustly achievable with thrusters, and the hardware and software architectures can include different subsystems without damaging other subsystems. Hence, the priorities were selected as follows: reliability in UUV design, automatic control implementation, computer vision object detection and localization implementation, and hydrophone system implementation. Thus, with the selected priorities, the team can focus on specific design tasks to successfully accomplish the proposed objectives.

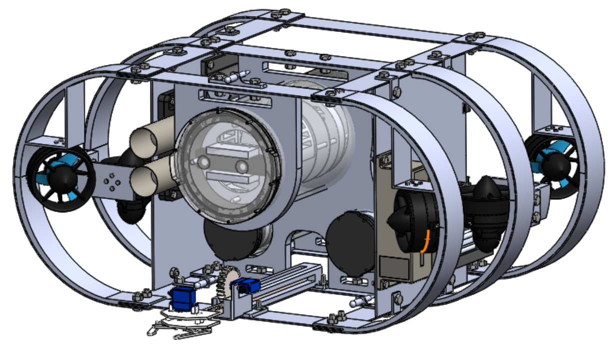

Fig. 1. CAD rendering of VTec U-III mechanical design.

II. Vehicle Design

A. *Mechanical*

The VTec U-III unmanned underwater vehicle (UUV) mechanical design is based on robustness, modularity and ease of manufacturing, which bring reliability into the system. The design also takes into account aspects of hydrostatics and buoyancy. The component distribution considers a weight balance configuration to facilitate modeling analysis. The materials and components are found in Appendix A.

1) Frame: The frame consists of structural pieces that support three acrylic electronic enclosures, and additional systems such as the torpedo launcher, the marker dropper, and the gripper. The frame also has the function of protecting the thrusters in case of collision, and of dealing with the static and dynamic loads. An open-frame with external oval supports, and inner transversal and longitudinal supports was designed. Aluminum 6061 is used for the exterior frame of the vehicle, as well as for the inner lateral supports. In the case of the central supports, ABS filament is used due to the durability ABS provides, and the complex shape manufacturing capabilities of 3D-printing.

2) Enclosures: The enclosures house the electronic boards, sensors and batteries. Three enclosures are categorized as one main enclosures

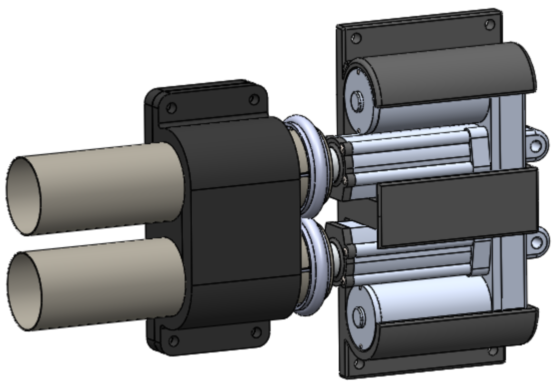

Fig. 2. CAD rendering of torpedo launcher system.

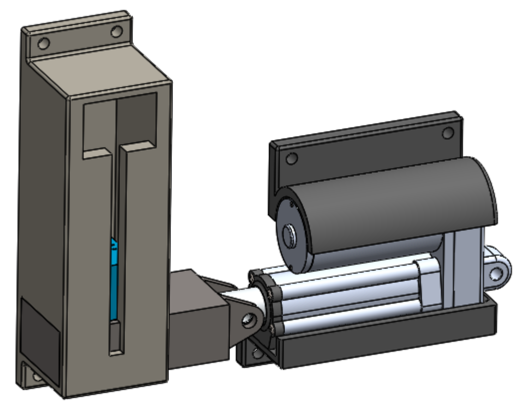

Fig. 3. CAD rendering of marker dropper system.

and two secondary enclosures. The main enclosure contains the PCB boards, sensors, and processors. Due to RoboSub rules size requirements, a 6-inch-diameter cylinder enclosure was selected, along a clear end cap to assist camera location and visualization. A custom-made PLA rack allows easy-removal, organization and safety of the internal electronics. Both of the secondary enclosures are 3-inch-diameter cylinder enclosures, each holding an independent battery. The three enclosures have aluminum end caps with holes to route cables for the actuators. Finally, the enclosures have silicone sealant to avoid leaks, as well as plugs in the end caps either to route cables safely or seal the unused holes.

3) Actuators: A six-thruster configuration was selected to actuate the system, using BlueRobotcs T-200 thrusters. Four thrusters are placed in the corners at a 45º angle, providing an X-shaped configuration. Two additional thrusters are faced upwards along the transversal axis, maximizing the movement along the normal axis. To secure stability and control, the thrusters were located in the farthest possible ends along the longitudinal and transversal axis.

The torpedo launcher (Fig. 2) is composed by two linear actuators and a torpedo-container. The torpedoes are custom-made and 3D-printed using PLA with 100% infill, achieving a roughly neutrally buoyancy.

The marker dropper (Fig. 3) is divided in three parts: a container, a sliding platform and a linear actuator. The linear actuator is attached to the sliding platform located inside the container. Once the actuator extends, it releases one of the stored markers. The designed markers are 3D-printed PLA squares with the extruded logo on the top surface.

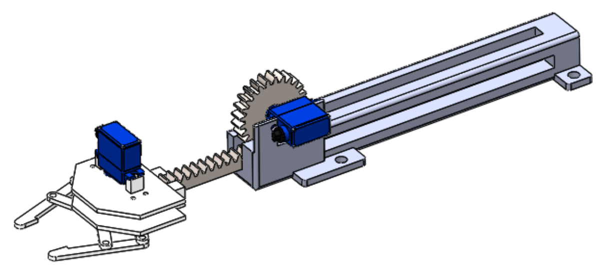

Fig. 4. CAD rendering of arm and gripper system.

The selected design features for the robotic arm were simplicity and reliability, thus a linear one-degree-of-freedom arm was designed. The gripper, shown in Fig. 4, is 3D-printed and modular to be easily changed depending on the competition rules of the year. The gripper is actuated by a HS5086 servo motor, which controls the open-close motion. The arm is conformed by a rack and pinion located in a fix base, also controlled by a HS5086 servo motor. Helical gears are selected since they have higher load bearing capacity.

B. Electrical and Electronics

The electrical and electronics design was driven by the requirements stemming from the challenges from previous editions of the competition, the need for deterministic firmware behavior and precise data acquisition, high computing power for algorithm execution and, finally, internal component reliability and safety. To tackle these requirements, a decentralized architecture was devised, distributing the low and high-level tasks between an electronic control unit (ECU) and a central processing unit (CPU). Additionally, two power supplies were included (11.1V and 14.8V) with their respective circuitry to accomplish the different voltage requirements of the components (3.3V, 5V, 12V and 16V). The following sections address these design choices in depth.

1) Sensor Selection: A high-quality, reliable Inertial Measurement Unit (IMU) is a crucial part of any autonomous vehicle, specially in underwater scenarios, where the lack of a GPS signal constraints the UUV to rely only on accelerometer and gyroscope signals to obtain velocity and position values. Doppler Velocity Log (DVL) and/or underwater GPS technologies could assist the state-estimation problem, but adding the disadvantage of significantly increasing cost. Therefore, the Vectornav VN-200 IMU was chosen for the VTec U-III, since other models present an advantage only when the use of GPS signal is available, and it is of previous VantTec ownership. Furthermore, to provide an accurate way of measuring depth (other than Z-axis accelerometer data), the BlueRobotics 30bar pressure-based depth sensor was added to the system.

Additionally, since challenge locations are identified using acoustic pinger signals, the inclusion of hydrophones on the submarine is necessary. For this, due to budget constrains, one high-end Teledyne RESON TC-4013 and two lower-end Aquarian H1C models were chosen to provide enough information for target position triangulation purposes.

Object detection and identification are critical activities in all of the challenges, as well as providing additional support when it comes to navigation. A StereoLabs Zed Mini stereo camera system was proposed over other computer-vision-oriented sensors for front-facing view because of its capability to both detect objects using camera images, and localize the objects with the depth perception ability. A Raspberry Pi Camera Module is also part of the architecture, providing a bottom-facing view from the vehicle perspective. Both cameras also fit the enclosure internal space constraint.

For system safety, a BlueRobotics SOS Leak Sensor was selected to provide an emergency stop signal for the internal electronics in case of sealing failure. Finally, a visible, accessible mechanical kill-switch is included to cut off power to all of the vehicle subsystems for emergency scenarios.

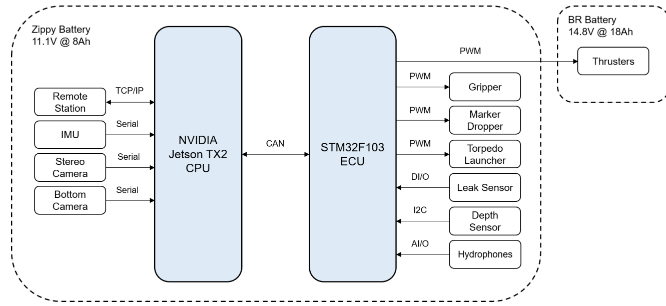

Fig. 5. Hardware architecture with power distribution.

2) System Architecture Design: Leak and depth sensing, thrust actuation, as well as hydrophone signal sampling were designated as critical tasks of the UUV, since they are involved in navigation and safety mechanisms. Thus, to additionally accomplish clock frequency and connectivity requirements, an STM32F103C8T6 was chosen as the ECU, which is in control of actuator manipulation and most of the data acquisition. To ensure determinism and task prioritization, the ECU runs on a Real-Time Operating System (RTOS).

In contrast, a CPU with high-processing power is required because of the guidance, navigation and control (GNC) algorithms. Thus, an NVIDIA Jetson TX2 was selected over other controller models, due to the integrated Graphical Processing Unit (GPU), which optimizes computer vision algorithm performance.

To allow for data transfer between the ECU and the CPU, multiple communication protocols were explored. Ultimately, in view of its expansion capabilities, robustness and existing supporting infrastructure on both Linux and the STM32 HAL library, Controller Area Network (CAN) was determined as the vehicle internal bus protocol. Specific data frames were designed in concordance with the byte size of the variables of interest, while their respective message identifiers were assigned following the CAN 2.0b standard, described in [8]. Sensor and actuator distribution between both controllers was driven by the connectivity needs of each component, as depicted on Fig. 5.

3) Power Distribution: The power demand of the T-200 thrusters (16V @ 24A) is comparably higher than that of the controllers (5V @ 1A for the ECU, 16V @ 1.2A for the CPU), actuators (ranging from 5V to 12V @ max 1.8A) and sensors (ranging from 3.3V to 5V @ max 1A). In consequence, a thruster-dedicated battery was included in the design (BlueRobotics Li-Ion 14.8V @ 18Ah), both as a safety measure and to ensure vehicle autonomy. For the rest of the subsystems, a Zippy Li-Ion Battery (11.1V @ 8Ah) was selected. Additional circuitry was designed to protect sensitive components and to provide the different voltage levels required for each element.

C. Software

The UUV software components development was accomplished following a methodology partially based on the Software Development Life Cycle defined in the ISO-12207 Standard [9]. The resulting protocol included the following tasks, listed in order of execution: Requirements Analysis, Architecture Design, Unit Design, Unit Development, Unit Testing, Integration and, finally, Integrated System Verification and Validation. The coming sections present the results of the proposed development process, as well as a description of the different subsystems.

1) Software Architecture Design: The architecture design was driven by two main objectives: to exploit the potential of the different available controllers and peripherals, and to accomplish the software requirements involved in the solution of the competition challenges. After an in-depth analysis of these objectives and related literature, such as the GNC system scheme proposed by Fossen in [6], the minimum required software components were defined, depicted in Fig. 6.

For the ECU components, a *communications bridge* module for data transfer between the ECU and the CPU; an *actuator control* module for the thrusters, gripper, torpedo launcher, and marker dropper; a *sensor monitor* module to sample the leak and depth sensors, as well as the signal from the kill switch. These applications run on a *Real-Time Operating System (RTOS)*, which manages the priorities of the tasks and ensures that the sampling rates, as well as the information processing and actuation, remain deterministic.

Similarly, the Ubuntu 16.04-based CPU runs *Robot Operating System (ROS) Kinetic*, where the various software components interact between them as ROS Nodes. These modules are: an *ECU bridge*, to provide a link between the low-level ECU functions and the resulting commands from the CPU algorithms; a *controller* module, which receives desired velocity, depth and heading references, and outputs desired thrust values; a *guidance* module, whose inputs are waypoint lists, while its outputs are desired controller values; a *mission manager* module, in charge of triggering the user-requested missions and sending commands to both the guidance and controller modules; a *master node*, which decodes user input and provides the according signals directly to the controller or triggers pre-programmed missions; a *perception* module, for processing camera input; and an *odometry* module, which receives accelerometer and gyroscope information from the IMU, processes it, and publishes velocity and pose data.

Additionally, the ground station also runs *ROS Kinetic* and is connected via Ethernet to the UUV. Currently, the only module running on the ground station is the *keyboard reader* module, which forwards key presses to the master node.

2) Modelling and Identification: The mathematical model of the UUV took a higher priority when the COVID-19 pandemic caused restrictions on the availability of the hardware. Thus, what once was supposed to be a supporting tool, became the only solution to allow the development of the vehicle to continue. The mathematical model is based on the methodology proposed by Fossen in [6], while the identification of the parameters was done based on the procedure developed by Eidsvik [7].

3) Guidance and Control: The motion controller development focused on the essential degrees of freedom (DOF): surge u, sway v, heave w, and yaw ψ. For surge and sway, speed controllers were developed, whereas heave and yaw have position controllers, since this provides depth and heading control for the vehicle. The four DOF use non-linear Proportional, Integral, and Derivative (PID) controllers with feedback linearization, similar to [10], which were hand-tuned using the previously obtained mathematical model. For path-following control capabilities, a the Line-of-Sight (LOS) guid-

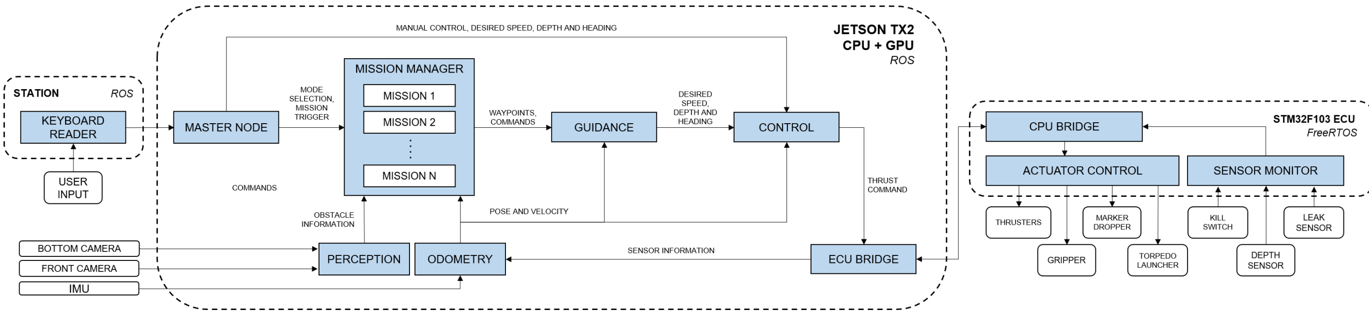

Fig. 6. Software architecture for the complete UUV system.

ance law presented by Fossen in [6] was developed. This feeds the controller the desired heading values to navigate to a specific point in space. The same ROS node selects speed and depth references.

4) Computer Vision: The computer vision system of the UUV is composed of 2 sensors: a ZED stereo camera and a Raspberry Pi camera module. The custom trained Convolutional Neural Network (CNN) YOLO-Tiny 3 [1] uses the RGB image obtained from the ZED for buoy detection. The stereo point cloud is used separately to calculate the distance, size, and orientation of buoys and other obstacles, such as the gate and the octagon. The Raspberry Pi camera is used as a down-facing camera, and the RGB image from this camera is used for detecting directional markers to different challenges.

The Convolutional Neural Network (CNN) YOLO was chosen as the detection strategy due to previous experience and positive results obtained in other VantTec projects, such as RoboBoat [5]. For training the CNN, images from past RoboSub competitions were taken and edited as a representation for RoboSub 2020 buoy images. To reduce the amount of images that had to be manually captured and edited, a data augmentation library was created. The created library introduces variation in qualitative elements like color, illumination, distortion and orientation. The data augmentation strategy has been previously proven effective for obstacle detection in [2].

Image labeling is a time consuming and monotonous task. Thus, to lower the amount of images that had to be manually labeled, an automatic labeling strategy was implemented. First, the MATLAB Image Labeler Toolbox was used to label a small amount of images of each object class. Then, an Aggregate Channel Features (ACF) detector [3] was trained for each class using MATLAB. Afterwards, each detector was applied on the entire data set and labels where manually verified and modified if necessary. Lastly, MATLAB image label objects were converted to YOLO format, and YOLO-Tiny 3 was trained using the open source neural network framework Darknet [4].

5) Simulation and Testing: The existence of a simulation environment allowed the software team to verify and validate the software architecture and its modules while the UUV was not available, effectively increasing the efficiency and speed of the development process. As the COVID-19 pandemic came into the picture, the relevance of the simulation environment became more evident; therefore, a considerable part of the work of the software team went into creating a more robust simulation.

Using the previously-obtained mathematical model, the states of the vehicle under the current inputs could be obtained. These values were then output to RViz, where the 3D CAD model was used to display the current pose of the vehicle inside the simulation space. Additionally, a 120 degree Field of View (FOV) cone with a radius of 3 meters was also depicted inside the simulation, as seen in Fig. 7.

Different software modules were developed to build the test environments for each of the challenges, shown in Fig. 8. These modules serve as a mock-up of what the perception module would output in a real-world scenario, since they take into account the pose of the UUV to only publish the objects that are inside the conical FOV.

For the 3D models of the challenge elements,

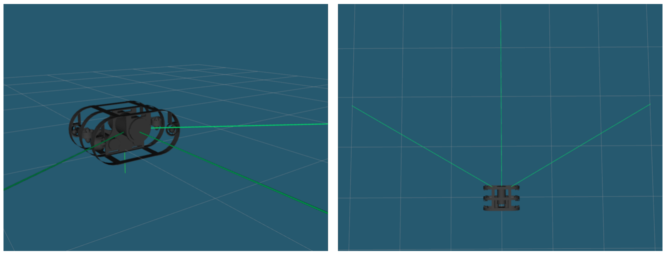

Fig. 7. 3D model of the UUV with FOV representation in RViz.

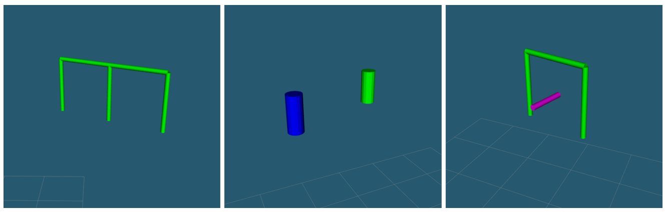

Fig. 8. 3D models for the gate (left), buoy (center) and shootout (right) challenges.

Fig. 9. Data Augmentation Examples

Fig. 10. YOLO detector Examples

simplified depictions were designed. Visual elements were discarded, such as the images of the police and gangster for the gate and shootout challenges. It is assumed that the vehicle already has the identification information that would be generated by the computer vision algorithms. Nonetheless, in case of the buoys and the shootout board, visual aids such as color differences and orientation indicators were included to facilitate the verification of the algorithms.

III. Experimental Results

In-water experiments were not possible due to COVID-19. Thus, all of the challenges were solved using the previously mentioned simulation environment. Computer vision algorithms were tested only with pre-existing data sets. The following sections discuss the solution approaches and results.

A. *Computer Vision*

Data augmentation results expanded the data set from 1095 to 1459 total images, and allowed object class balancing for impartial detector training. Some examples of data augmented images are shown in Fig. 9. Examples of detected objects using the trained YOLO CNN are shown in Fig. 10. Table I shows the performance of the trained object detector. The results obtained outperform previous work with YOLO-Tiny 3 from previous RoboBoat project results [5], demonstrating the benefits of the data augmentation strategy. As mentioned before, the computer vision strategy also relays on the detection of obstacle size and orientation using the ZED Mini stereo point cloud, however this has not been tested due to the COVID-19 pandemic.

B. *Waypoint Navigation*

As a first validation of the guidance and control subsystems, a set of pre-defined waypoints were input to the guidance module. The goal was to command the vehicle to follow a specific path, with varying changes of heading and depth, to ensure that the algorithm presented no mathematical errors and that the vehicle would be capable of reaching any point the missions would input. Fig. 11 depicts the

TABLE I
PERFORMANCE OF YOLO TINY 3 DETECTOR

gate (AP%)	police (AP%)	gangster (AP%)	gun (AP%)	badge (AP%)	marker (AP%)	mAP (%)	IoU (%)	F1 Score
96.67	94.08	94.58	97.86	98.78	99.93	96.98	82.98	93.0

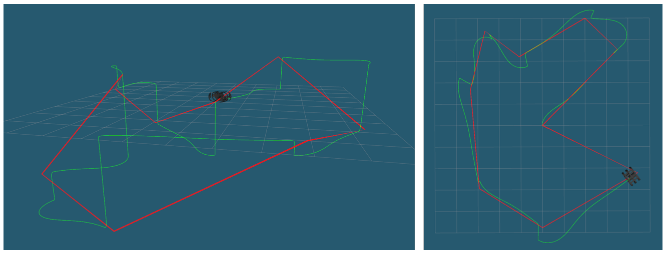

Fig. 11. Waypoint navigation simulation.

simulated waypoint navigation from two different perspectives, demonstrating the capability of the UUV to follow a path in three dimensions.

C. Gate Challenge

The gate challenge solution consisted of a state machine, whose inputs are the detected objects around the UUV, the ego vehicle pose, and the selected side to follow, be it gangster or police officer. Once the mission is triggered, the UUV will execute a search algorithm, which consists of a yaw sweep to locate any obstacles that in range. If the obstacles are any other than the gate, it will advance to a point in front of it before starting to sweep once more. Said displacement alternates directions between positive and negative sway to maximize the search area. Once the gate is located, it is assumed that the computer vision algorithm successfully detects which side of the gate corresponds to which figure. Here, a new reference frame is created with the origin located in the center of the gate. Then, three waypoints are calculated according to the pose of the gate and the desired side, placing one in front, one in the middle and one in the back, so that the UUV crosses and passes the gate completely. Results produced by this algorithm are shown in Fig. 12.

D. Buoy Challenge

Tackling the buoy challenge required more creativity. Since the buoys can be oriented differently, the identification of the image becomes more difficult. Therefore, a two-step detection and identification process was the selected approach for this challenge. Using the same search algorithm from the gate challenge, the UUV searches and tracks the position of each of the buoys. Once it has already found both of them, the euclidean distance between them is used to obtain a midpoint, which serves as the center of a circle, whose radius is proportional to the distance between the buoys. This circle is discretized into a waypoint array, which is fed to the guidance module. The vehicle then orbits around the buoys facing always toward the center point, in an attempt to identify the image that corresponds to each of the previously registered buoys. After assigning an ID to the buoys, the UUV approaches the one that matches the previously assigned side. A depiction of the generated trajectory and overall execution of the algorithm can be seen in Fig. 13.

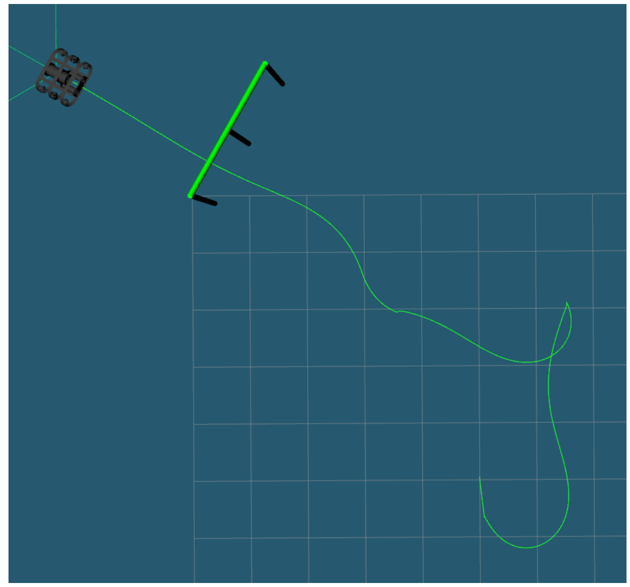

Fig. 12. Gate challenge simulation.

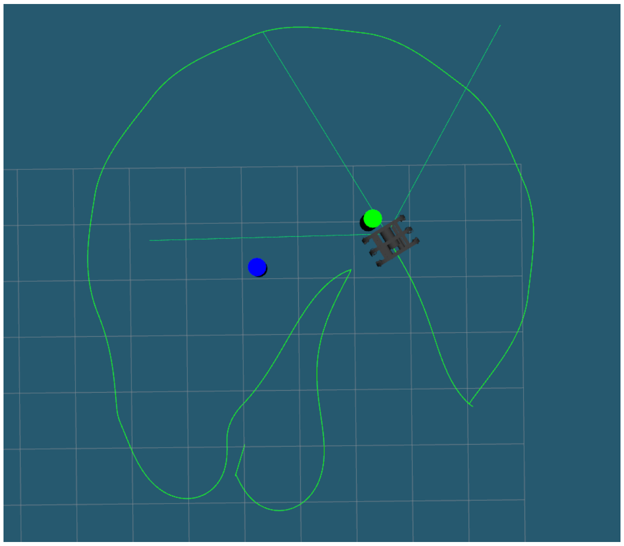

Fig. 13. Buoy challenge simulation.

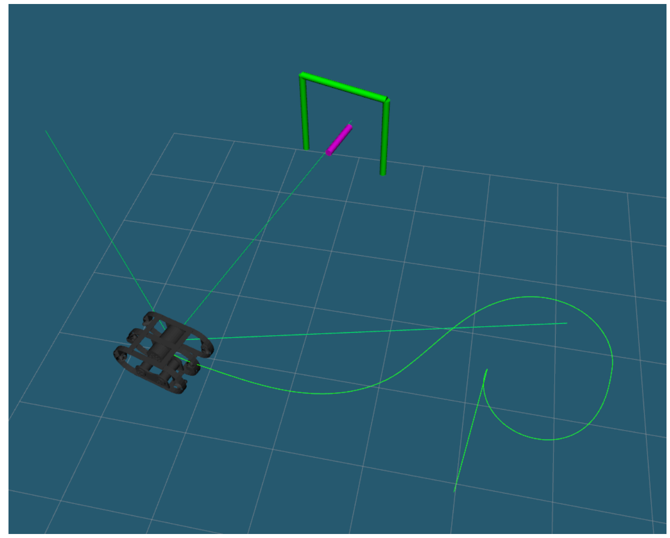

Fig. 14. Shootout challenge simulation.

E. Shootout Challenge

For the shootout challenge, it was assumed that there would be one board with two sides. Since the aiming system was not ready for this competition due to complications with algorithm testing due to COVID-19, the simulated challenge focused solely on aligning the vehicle to the center of the board, so that the aiming algorithm could then take over. Starting with the search procedure described for the gate challenge, the UUV looks for the board. Once it is located, its center point is used as a reference to create a circle with a radius of 2.5 meters, to allow for the camera to have a clear view of the board itself. The vehicle orbits the board, looking for the desired image (gangster or police officer) and once the orientation of the image matches that of the UUV, it stops. Finally, the heading of the vessel is aligned with center point of the board (or where the whole would be), to prepare to shoot.

IV. Conclusions

The presented team organization and competition strategy for the RoboSub 2020 competition involve taking advantage of the multidisciplinary nature of the team, while prioritizing those challenges which do not involve actuation other than that of the thrusters. The design of the vehicle was done from a systems engineering perspective, taking into account the influence of the mechanical, electronics and software aspects on the performance and the ability to accomplish the desired requirements to successfully execute the competition challenges. Furthermore, a digital twin of the UUV was developed to enable virtual tests of the motion control and path following algorithms, as well as the solutions to each of the priority challenges: gate, buoys and shootout. The impact of the COVID-19 pandemic constrained the available resources of the team, preventing further physical development of the submarine. Nonetheless, the results from the simulated environment show that the proposed solutions are technically feasible, since all three challenge simulations were completed successfully.

Acknowledgments

This work was supported by: Hacsys, VectorNav, Google, ifm efector, Uber ATG, RoboNation, Velodyne Lidar, NVIDIA, Skysset, Akky and Greenzie. Finally, VantTec appreciates the support from the university, Tecnologico de Monterrey.

References

[1] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement,“ Technical Report.

[2] I. Navarro, A. Herrera, I. Hernandez, L. Garrido, "Data Augmentation in Deep Learning-based Obstacle Detection System for Autonomous Navigation on Aquatic Surfaces", Mexican International Conference on Artificial Intelligence, 2018.

[3] Dollar, P., R. Appel, S. Belongie, and P. Perona. "Fast Feature Pyramids for Object Detection." Pattern Analysis and Machine Intelligence, IEEE Transactions. Vol. 36, Issue 8, 2014, pp. 1532–1545.

[4] J. Redmon, “Darknet: Open Source Neural Networks in C,” 2013-2016, https://pjreddie.com/darknet/.
[5] A. Gonzalez, et al., “VantTec Technical Design Report 2019”, RoboBoat 2019, 2019.
[6] T. I. Fossen, “Handbook of Marine Craft Hydrodynamics and Motion Control”, John Wiley and Sons, Ltd., 2011.
[7] O. A. Eidsvik, “Identification of Hydrodynamic Parameters for Remotely Operated Vehicles.”, Norwegian University of Science and Technology, 2015.
[8] Robert Bosch GmbH, “CAN Specification, Version 2.0”, 1991, http://esd.cs.ucr.edu/webres/can20.pdf.
[9] ISO/IEC/IEEE, “ISO/IEC/IEEE-12207 International Standard - Systems and software engineering - Software life cycle processes”, IEEE, 2017.
[10] Vervoort J.H.A.M., “Modeling and Control of an Unmanned Underwater Vehicle”, University of Caterbury, 2009.

APPENDIX A
COMPONENT SPECIFICATIONS

See Table II.

APPENDIX B
OUTREACH ACTIVITIES

A. Conexion Tec

Conexion Tec is an event organized by Tecnologico de Monterrey’s School of Science and Engineering which highlights the best engineering projects each semester. VantTec participated with the technology and researched developed for RoboBoat and RoboSub in both spring and fall semesters.

For more information please visit: https://www.facebook.com/conexiontec/

B. Global AI on Tour MTY

Microsoft Student Partners organized this event to promote technologies related to artificial intelligence (AI). VantTec gave a virtual talk during this event, showcasing the history of the group and the AI applications for autonomous systems.

For more information please visit: http://globalaimty.net/

C. Expo Manufactura

In partnership with sponsor ifm efector, a mobile robot was developed to demonstrate the capabilities of ifm efector sensors. This robot showcased at the Expo Manufactura 2020, serving as a platform to attract clients and promote technology to students at the Expo.

For more information please visit: https://expomanufactura.com.mx

D. INCmty

INCmty is the largest festival of entrepreneurship in Latin America, based in Monterrey, Mexico. In November, VantTec took part representing the university’s School of Science and Engineering.

For more information please visit: https://incmty.com/

E. Rice University GCURS

Rice University organizes the Golf Coast Undergraduate Research Symposium, which allows young researchers to present their work, meet Rice faculty, and receive valuable feedback. VantTec members gave presentations on the work and research made by the group.

For more information please visit: https://gcurs.rice.edu/

TABLE II
COMPONENT SPECIFICATIONS

Component	Vendor	Model	Specifications	Quantity	Cost
Buoyancy Control	Blue Robotics	Buoyancy Foam	16in x 8in x 1in	1	35
Frame	Own design	VantTec 3.0	Aluminum 1060	1	206
Waterproof enclosure	Blue Robotics	6” series	11.75”	1	265
Waterproof enclosure	Blue Robotics	3” series	8.75”	2	204
ROV Tether	Blue Robotics	Fathom	35m	1	158
Plug	Blue Robotics	Leak proof plug	-	23	26
Penetrator	Blue Robotics	Leak proof plug	-	23	61
Thruster cable	Blue Robotics	Thruster Cable	-	6	20
Cable penetrator	Blue Robotics	M10 Cable Thruster	8mm	6	10
Thruster	Blue Robotics	T-200	-	6	1048
ESC Controller	Blue Robotics	Basic	-	6	150
High Level Control (ECU)	STMicroelectronics	STM32F103C8T6		1	20
Linear actuator	Hydroworks	2in stroke	12V - 4.6A	3	80
Servomotor	HiTEC	HS5086WP	4.8V - 6.0V, 181°	2	60
Kill switch	Blue Robotics	Kill Switch	-	1	14
Battery	Blue Robotics	Lithium-ion Battery	14.8V - 18Ah	1	289
Battery	Zippy	Lithium-ion Battery	11.1V - 8Ah	1	220
Step down	Pololu	5V - 5A	-	1	15
CPU	NVIDIA	Jetson TX2	GPU and 8 GB memory	1	600
CPU Carrier	NVIDIA	Quasar	-	1	488
Internal Comms Network	-	-	CAN Bus 2.0b	-	-
External Comms Network	-	-	TCP/IP over Ethernet	-	-
Programming Language	-	-	C/C++	-	-
IMU	VectorNav Technologies	VN-200	-	1	4000
Camera	Stereolabs	ZED Mini	1080p Resolution	1	450
Camera	Raspberry Pi	Camera Module V2	8 Mega Pixel Resolution	1	450
Hydrophone	Telodyne	RESON TC 4013	-	1	1200
Hydrophone	Aquarian	H1C	-	2	318
Depth/Pressure sensor	Blue Robotics	30 Bar	-	1	80
Leak sensor	Blue Robotics	Leak Sensor	-	1	26
Manipulator	Own Design	Own Design	3D Printed Gripper	1	5
Algorithms: vision	-	-	Yolo Tiny V3	1	0
Algorithms: localization and mapping	-	-	Reference frames and 3D Computer Vision internal development	1	0
Algorithms: autonomy	-	-	Line-Of-Sight Guidance	1	0
Algorithms: autonomy	-	-	Non-Linear PID Motion Control	1	0
Open Source Software	-	-	OpenCV	1	0
Open Source Software	-	-	ROS Kinetic	1	0
Open Source Software	-	-	FreeRTOS CMSIS V1.0	1	0
Open Source Software	-	-	Eigen (C++ Library)	1	0
Team Size	-	-	-	21 members	0
HW:SW Expertise Ratio	-	-	-	9:12	0
Testing time: simulation	-	-	-	300h	0
Testing time: in water	-	-	-	0h	0