RoboSub 2022: VantTec Technical Design Report

Andres Sanchez, Edison Altamirano, Abiel Fernandez, Saul Bermea, Sebastian Martinez,
Diego Martinez, Jair Barrera, Ian Duran, Kevin Vega, Gabriela Alvarez, Ariana Rodriguez, Sicilia Pompa,
Andres Estrada, Andrea Marisol, Victor Fuentes, Jorge Vazquez, Pamela Morales, Laura Luna,
Emiliano Wabe, Humberto Tamayo, David Guerra, Rodrigo Romo, Claudia Torres, Sofia Martinez,
Gabriel Bermudez, Adrian Hernandez, Francisco Aguilar, Jesus Arteaga, Nadia Garcia, Andrea Rangel,
Adrian Leon, Aleksandra Stupiec, David Martínez, Alejandro Gonzalez, and Leonardo Garrido
Tecnologico de Monterrey
Monterrey, Mexico
vanttec@servicios.tec.mx

***Abstract*—The strategy for RoboSub 2022 competition, developments, improvements, and results are presented in this report. The final strategy relies on manufacturing the new VTec U-IV vehicle, and testing hardware and software systems. A navigation system based on octomap cluster detection is addressed, a localization method based on a linear kalman filter is implemented, and results on pinger signal detection are presented. Finally, challenge approaches were validated in simulation, as a physical platform was not ready.**

***Index Terms*—RoboSub, Unmanned Underwater Vehicle, robotics, autonomy, GNC system, computer vision, perception, artificial intelligence.**

I. Competition Strategy

Health regulations established by local governments to face the COVID-19 pandemic started to soften by this competition season. Nevertheless, months of lockdown during the second half of 2021 were still enforced. In retrospective, by comparing the team performance achieved during this competition season, with the results obtained by the RoboSub 2021 team, a greater level of maturity was reached, as major areas of opportunity were deviced to work on to strengthen the UUV autonomous capabilities and overall performance.

During the second half of 2021 focus was given to three main aspects of the project: to improve the mechanical and electronics systems design, to improve and implement a custom state estimation method, and to develop a navigation algorithm. No physical development was made during this period as access to Campus facilities was still limited.

At the beginning of 2022, it was decided that we would not attend the physical competition for budget constraints, as some hardware components were not bought yet, nor tested, and priority was given to the RoboBoat competition, as it is an established project.

In february of the present year, return to activities was restored, and we were able to start working physically on our projects once again. We then decided to change our focus slightly, by prioritizing the manufacture of the mechanical and electronic systems, and the hydrophone's related tasks, as the subproject has been in existence for almost three years with few results, and a desire for its completion was present within the team.

II. Design Creativity

A. Mechanics

1) Thruster configuration: the main problem with last years six-thruster configuration (VTec U-III model) was that the two vertical thrusters were not horizontally aligned; fully-actuation was provided, but stability in roll and pitch was hard to achieve. With this limitation in mind, the thruster configuration was modified, by aligning the two vertical thrusters and tilting the frontal and rear ones. With the mentioned changes, stability in pitch and roll was easier to obtain, preserving fully-actuation, while prioritizing motion in surge.

2) Design modifications: compared to last year's mechanical design, the size of the main cylindrical enclosure was increased, from six to eight inches, as

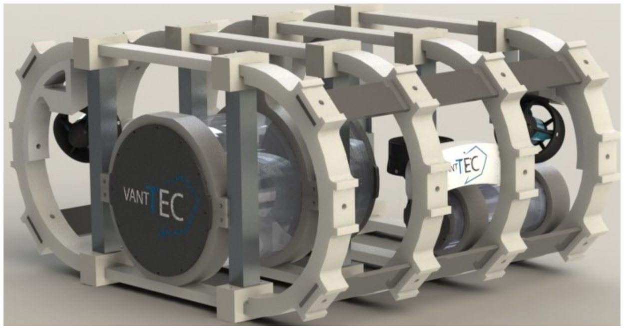

Fig. 1. VTec U-IV

not enough holes in the enclosure lid. The change in enclosure dimensions, added to the change of the thruster configuration, led to the necessity of adding a fourth rib so the vertical thrusters. The current UUV, the VTec U-IV, is shown in Fig. 1.

B. Electronics & Embedded Systems

The electronic and PCB design was made considering reusability both for our USV and UUV in terms of core components present in both systems. As an evolution from last year's design, we designed and manufactured two new PCBs. One of these, is the motor controller PCB seen in Fig. 2. This PCB receives motor commands from the Jetson TX2 via CAN bus. A STM32 [1] microcontroller is used, which processes sensor data, and generates PWM signals for the thruster motor ESCs and mechanism servos. By utilizing FreeRTOS [2], a real-time operating system and our CAN bus protocol, we can ensure that we have real-time robust control over our motors and mechanisms. The second PCB we designed is the power distribution PCB (Fig. 3), that uses relays to kill-switch the devices with a digital signal or through the emergency stop button. It also distributes power and provides over-current protection for each output channel.

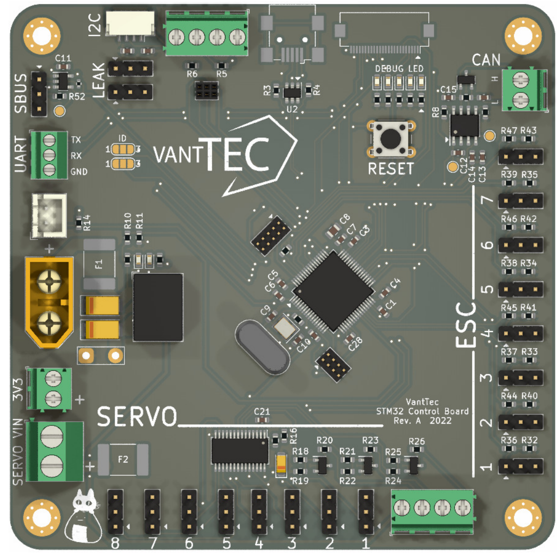

Fig. 2. Motor controller PCB.

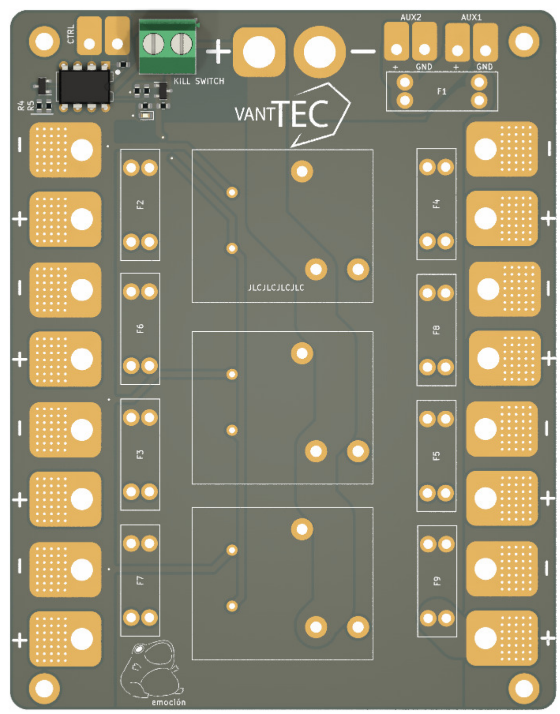

Fig. 3. Power Distribution PCB.

1) Hydrophones: Work was made to find hardware and software solutions to analyze hydrophone signals. As the pinger's signal can lose intensity due to environment factors and travel distances, a set of OpAmps was designed to amplify the signal. Also, it was deemed necessary to filter the signal using software, to reduce environmental noise. Hence, the pre-processing of the signal allowed to select and detect the same working-frequency of the pinger, which is achieved by analysing the signal Fourier Transform in the frequency domain.

The algorithm used is Direction of Arrival (DOA), which is a technique that correlates signals from the same source at different moments of time. Using a three hydrophone array as signal receivers, the angle (in 2 dimensions) relative between the vehicle and the source is obtained through geometrical identities [9].

C. Software Architecture

The software architecture for the VTec U-IV is built upon Robot Operating System (ROS). A migration from ROS Kinetic to ROS Melodic was required, as Ubuntu 16.04 is no longer supported. ROS actions were created to simplify the use of the controllers at the moment of developing challenge solution algorithms, as they allow a client-server communication with no program flow block, and also informing on the petition status. The actions implemented for navigation are: rotate action, which receives an angle in radians and rotates in yaw until the UUV reaches the desired angle; walk action, which makes the UUV navigate 6 meters in a straight line; and the GoTo action, used to navigate the vehicle to a specific 3D point. Also, for clustering, an action was implemented to to make mapping request at any arbitrary moment, so the mapping method would not consume excessive computation resources.

D. Mapping and Navigation

In simulation, the vehicle navigates from a starting position that serves as the coordinate system's origin. The vehicle detects adjacent landmarks as it travels and maps them with an uncertainty that includes fixed variation with growing pose uncertainty. The system was tested in different simulation environments created in Gazebo, and we used an open-source system called Octomap [4] to create volumetric 3D models of environments. The mapping method is based on octotrees and employs a probabilistic occupancy estimation. It expressly represents not just the habitable space, but also the open and unknown space. Fig. 4 and Fig. 5 depict the output of a cutting-edge SLAM method based on constraint graph analysis and a layered search for the optimal data correlation. Once we have as much knowledge about our surroundings as feasible, we may begin to establish exploration locations where there is a higher cloud density of points. When employing clustering algorithms, deciding how many clusters to utilize is always a challenging task. The group's inertia, or the sum of squares value, reveals how cohesive the various groupings are.

Fig. 4. Virtual world created in Gazebo

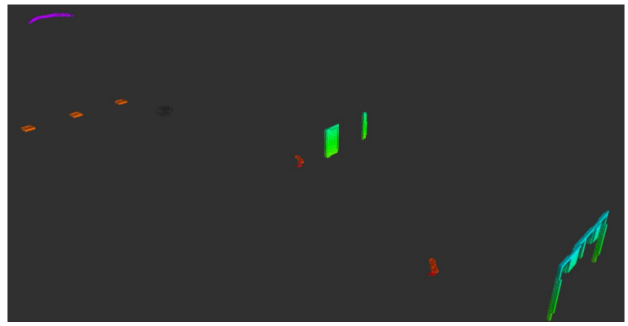

Fig. 5. Point cloud capture in global map

E. Control

Last year, a 4-DOF dynamic model was used for simulations and control design was used, but in order to provide a more realistic performance, the model was improved to 6-DOF model. Based on the mentioned model, an Adaptive Sliding Mode Control [5] scheme was applied for the motion control of the UUV, replacing last year's PID controllers. The sliding surface ensures robustness against external perturbances and model uncertainties, and the adaptive law avoids overestimation of control efforts while minimizing chattering.

F. Perception

Object detection based on instance segmentation is a demanding task in terms of computational processing, since not only an object must be identified, but also its associated 3D coordinate [6], as well as a mask indicating the shape of said object. Therefore, algorithms that have high prediction effectiveness are sought. This is the case of the YOLACT (You Only Look At CoefficienTs) detection algorithm [7]. The model trained uses the repository with the same name, with the Pytorch framework and

Resnet 101 pre-trained weights. YOLACT fully observes the image at the time of testing, so its predictions are based on the global context of the received image. This is divided into regions, so you can create bounding boxes and start predicting probabilities within each region. In the end, these results are weighted, making this prediction method significantly faster than other models and having the same detection effectiveness [8].

G. Localization

As GPS do not work underwater, and only using an IMU for position estimation leads to cumulative errors over time, a DVL was deviced necessary to acquire precise readings. In that sense, a vectornav VN-200 IMU, a barometer and a DVL are intended to be the odometry sources for the vehicle systems, with a linear Kalman filter serving as a baseline for future state estimation methods to be developed. The method was initially developed in matlab, considering only acceleration readings from an IMU. Afterwards, a python basic node was implemented in simulations, but further work needs to be required to integrate the barometer and DVL readings.

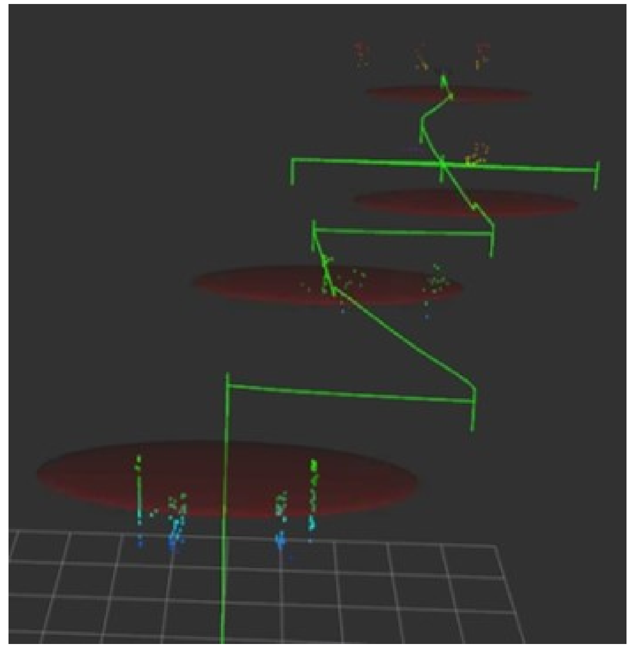

Fig. 6. Trayectory of a submarine using our exploratory strategy

III. Results

A. Discovery method

In Fig. 6, five exploring zones with a high likelihood of finding the competition's tasks are shown, representing the effort to divide a large number of point clouds into scan points using clustering techniques. In fact, the data is correct, as the algorithm correctly identified the gate, buoys, bins, and octagon missions. The red ellipse shows the region the submarine is evaluating as a zone with a high likelihood of discovering a mission, while the green route depicts the UUV's trajectory during the search of new missions.

B. Task approaches

1) Choose your Side: A state machine was developed to complete the tasks. Particularly in "choose your side" a vision based detection was implemented to recognize both g-man and bootlegger images and make a decision to pass through the selected target. In case a class is not recognized, the UUV runs the aforementioned search algorithm, inspired by clustering methods, such as k-means, which is an unsupervised machine learning algorithm. This is the quickest and most effective in grouping data points into categories.

2) Path Marker Following: the suggested detection method is locating the path marker using the bottom camera's image. The first step is for an algorithm to use numerous OpenCV filters to reduce the ambient noise. The Canny edge detector and the *findContours* function are then used to determine how many borders the marker has. The marker rotation angle is afterwards provided by the *minAreaRect*, which is then utilized to establish a waypoint in the same direction.

3) Make the Grade: to accomplish the "Make the grade" task, the UUV uses the bottom camera to identify the path marker direction. Then the craft proceeds to explore the closer point-cloud cluster arrangement generated by the K-means method. Using the prior mentioned work in perception, our vision systems recognize both buoys and decide which one should be approached.

4) Collecting: Following the generated clusters for exploration, the submarine moves towards the next one. The distance to each identified cluster is calculated using an Euclidean distance function, which measures distance between data points. Thus, it prevents the combination of clusters, by deciding

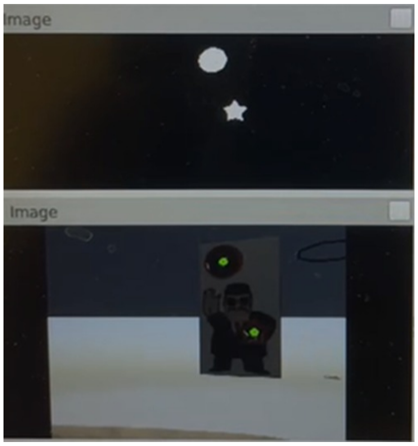

Fig. 7. Shape detection for shooting a torpedo

which data points belong to the same cluster. Once the submarine reaches the bins zone, it hovers over them, and proceeds to deploy a path marker.

5) Survive The Shootout: Thanks to our pinger localization and vision exploration systems, the shooting zone can be reached. Once the vehicle identifies the props, it decides which one to shoot. Using shape detection, a waypoint is generated in the centroid of each opening as shown in Fig. 7, which helps the UUV to positionate itself and target.

6) Octagon Challenge: While underwater, the UUV route is shifted towards the center of the hexagon by tracking the signals of the pinger. Once the vehicle is below the octagon, the location is saved, then, a bottle is grasped and the vehicle ascends to the hexagon's center on the water surface. The vehicle then finds the matched table, then it places the bottle on top of it. Next, the vehicle returns to the hexagon's center and submerges, to repeat the procedure until there are no more bottles.

IV. Conclusions

In this technical design report, the competition strategy for RoboSub 2022 was presented. For this competition season, focus was given to four aspects: to improve and implement the mechanical and electronics systems, to develop a custom state estimation method, to develop a navigation algorithm, and to finish the pinger localization system.

The mechanical design was slightly modified, adding a fourth rib to provide more space to align the vertical thrusters, and to rearrange the space due to the larger main cylindrical enclosure used. New PCBs were designed that could be used for team's USV and UUV, as some components are common in both vehicles. The software architecture was improved based on last year's efforts. ROS actions were created to simplify the use of the controller when designing challenge solutions. The octomap approach was implemented as a mapping approach to identify clusters where challenges could likely be located, and used for navigation purposes. A new controller based on adaptive sliding mode control was implemented, and the previous 4-DOF dynamic model was improved to a 6-DOF one. Yolact was implemented as an instance segmentation system for objects detection, although it was not completely implemented, and YOLOv3 from last years competition was used again, but with several improvements in detection speed. A basic linear kalman filter started development, altough further work is still required. Finally, the discovery and navigation method is described, followed by explanations for the task approaches used.

Acknowledgments

This work was supported by: Techmake, SBG Systems, Google, IFM efector, RoboNation, Velodyne LiDAR, NVIDIA, Akky, ZF Group, Güntner, and Siemens. Finally, VantTec appreciates the support from the university, Tecnologico de Monterrey.

References

[1] "STM32F405XX Datasheet " www.stm.com [Online]. Available: https://www.st.com/resource/en/datasheet/stm32f405rg.pdf. [Accessed: 05/23/2021].
[2] "What is An RTOS? " www.freertos.org [Online]. Available: https://www.freertos.org/about-RTOS.html [Accessed: 05/23/2021].
[3] Robert, B., "CAN Specification, Version 2.0" [Online], 1991. Available: http://esd.cs.ucr.edu/webres/can20.pdf. [Accessed: 23-May-2021].
[4] Armin H., Kai M.W., Maren B. and Cyrill S., Wolfram B., "OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees", Autonomous Robots, 2013. Available: https://octomap.github.io.
[5] Gonzalez-Garcia, A. and Castañeda H., "Guidance and control based on adaptive sliding mode strategy for a usv subject to uncertainties", 2021, IEEE Journal of Oceanic Engineering, 1–11.
[6] Gurney, K. (2003). An Introduction to Neural Networks. Taylor & Francis.
[7] Bolya, D., Zhou, C., and Xiao, F, and Lee, Y.J. "YOLACT: Real-time Instance Segmentation", 2019, arXiv.
[8] Bolya, D. (2019). YOLACT++ Better Real-time Instance Segmentation. https://arxiv.org/pdf/1912.06218.pdf

[9] Kumar Eranti, P. and D. Barkana B., ”An Overview of Direction-of-Arrival Estimation Methods Using Adaptive Directional Time-Frequency Distributions”, 2022, MDPI Journal of Electronics.
[10] Fahmi Amri, M. and Hilmi Ismail, Z., ”Passive Acoustic Detection Unit for Autonomous Underwater Vehicle Subsystem Based on Hydrophone Sensor”, 2017, IEEE Interational Conference on Underwater Systems Technology.
[11] Chen, L. ”Towards autonomous localization and mapping of AUVs”, 2015, Emerald International Journal of Intelligent Unmanned Systems.

APPENDIX A: COMPONENT SPECIFICATIONS

See Table I.

TABLE I
COMPONENT SPECIFICATIONS

Component	Vendor	Model	Specifications	Quantity	Cost (USD)
Frame	Own design	VTec U-IV	Nylamyd XL	1	-
Waterproof enclosure	Blue Robotics	8” series	-	1	265
Waterproof enclosure	Blue Robotics	3” series	-	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	STM32F405RG		1	20
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	Connect Tech	Quasar	-	1	488
Internal Comms Network	-	-	CAN Bus 2.0b	-	-
External Comms Network	-	-	TCP/IP over Ethernet	-	-
Programming Language	-	-	C/C++/Python	-	-
IMU	VectorNav Technologies	VN-200	-	1	4000
Camera	Stereolabs	ZED 2i	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: Perception	-	-	Yolo Tiny V3 and 3D Computer Vision internal development	1	0
Algorithms: Localization	-	-	Linear kalman filter	0	0
Algorithms: Control	-	-	Adaptive Sliding Mode Controller	1	0
Open Source Software	-	-	OpenCV	1	0
Open Source Software	-	-	Point Cloud Library	1	0
Open Source Software	-	-	ROS Melodic	1	0
Open Source Software	-	-	FreeRTOS CMSIS V1.0	1	0
Open Source Software	-	-	Eigen (C++ Library)	1	0
Team Size	-	-	-	31 members	0
HW:SW Expertise Ratio	-	-	-	9:12	0
Testing time: simulation	-	-	-	300h	0
Testing time: in water	-	-	-	0h	0