# Development of a Navigation and Position Tracking System for a Remotely Operated Vehicle (ROV) – ORCA

**P Tejaswini, I Sai Deepika and Sakthivel Murugan Santhanam**

Underwater Acoustic Research Lab, Department of Electronics and Communication Engineering, Sri Sivasubramaniya Nadar College of Engineering, Kalavakkam, Chennai, India 603110

E-mail: sakthivels@ssn.edu.in

**Abstract:** Remotely Operated Vehicles (ROVs) are a class of manually operated underwater vehicles that are designed for specific applications in maritime environments. For an underwater vehicle, vigilant precautions must be ensured in designing the navigation module because of the inconsistent nature of underwater channels. This paper aims to focus on the same, by developing an apt navigation system for the Remotely Operated Vehicle, ORCA which has been equipped with sensors such as temperature, pressure, and leak detector, for monitoring the ambient conditions during deployment and an Inertial Measurement Unit for tracking and positioning of an ROV. Mission Oriented Operating Suite-Interval Programming runs as the platform for navigation. Emergency Float-Up Mechanism that allows the system to be easily retrieved during emergency failures has been built in. The authors have made a successful attempt at building a navigation module that achieves stability and accurate positioning of an ROV after certain post-processing of data.

## 1. Introduction

The future of research in maritime technology is promising in the decades to come. Everything from rivers to oceans is being thoroughly inspected with the help of fast pacing underwater robot technology instead of human probation. This is done to ensure the safety of the human workforce in such harsh and mercurial environments. A Remotely Operated Vehicle (ROV) is an underwater robot that is customized and designed to serve specific applications. It is maneuvered by the operator with a tether, to capture real-time pictures and to stream video footage. Underwater ROVs are used in a variety of industries: Search and Rescue, Military, Recreation and Discovery, Aquaculture, Marine Biology, Oil, Gas, Offshore Energy, Shipping, Submerged Infrastructure, and more. The advancement in navigation and sensing technologies opened a wide variety of solutions to be incorporated in the ROVs for efficient and smart navigation in unpredictable environments. These localization and navigation methods range from acoustic navigation, geophysical navigation, doppler-based navigation, a fusion of LBL (Long Baseline) and, doppler-based navigation to advanced and efficient technologies like dead reckoning and inertial navigation systems [1], [2], [3]. In [4], [5], position tracking systems for an ROV are discussed using a laser-based vision system and IMU.

The control of an underwater vehicle is complex because there are highly nonlinear hydrodynamic effects resulting from the interaction with the environment that cannot be quantified. Hence the development of an efficient control system for stabilized movement is essential [7], [8], [9], [15]. There are a variety of control systems that differ based on efficiency which can be developed for ROV [10], [11]. The Proportional Integral Derivative (PID) controllers are the standard controllers for industrial automation and are extremely popular due to their flexibility. The development of an efficient PID closed-loop system for control and positioning of the ROV is very crucial [6], [12], [13], [14]. The remote operation of the system is enabled by various communication strategies used in ROV telemetry. Sensors like altimeter, pressure sensor, inclinometer, magnetic and, gyrocompass are also used to help ROVs understand their environment [3], [2]. ROVs are commonly equipped with cameras for surveillance and video streaming. [16] discusses the design and implementation of Aquabot, an ROV developed with the focus of tackling real-time video streaming difficulties and other development strategies.

This paper aims to focus on the development of an efficient navigation and control system that can be integrated into ORCA for smooth maneuvering and inspection. The data acquisition and position tracking were enabled with the use of an Inertial Measurement Unit (IMU) and sensors to monitor temperature, pressure, and water ingress into the electronic chamber. The paper is divided into two sections. The first section elaborates on the hardware design and various sensors equipped in ORCA The second section highlights the MOOS-IvP software, which is the prime navigation tool used in this development. MOOS-IvP is a robust software developed by MIT, Massachusetts, to provide autonomy to underwater robots. Although this software focuses only on autonomous robots, an exclusive MOOS Application (MOOS-App) has been developed by the authors to impart remote operation for an ROV. This development in Orca helps it stand out in terms of integration of various sensors with ease and efficacious control system algorithms for stability the incorporation of the Inertial measurement unit in ORCA for stabilized navigation has also been highlighted. Finally, the results based on the practical deployment of ORCA are discussed.

## 2. ORCA Design

ORCA is a 500 x 350 x 210 mm vehicle weighing 8kgs which is equipped with sensors and a camera to serve the prime application of underwater inspection and survey. This six thruster ROV was fabricated using HDPE. It comprises two sealed acrylic enclosures on the body, to carry the electronics and the battery, respectively. The whole system is powered using the 14.8V, 18Ah Li-ion battery. The flexibility in mounting the T200 thrusters at two different angles has extended the adaptability of ORCA. In this section, the electronic system development by integrating the sensors is detailed.

### *2.1. Sensors in ORCA*

Sensors are vital for any underwater inspection and survey to happen effectively. Various sensors specific to the application are equipped in ORCA to carry out detailed underwater monitoring. These sensors also serve as a health monitor for ORCA to prevent its navigation beyond the operating range.

*2.1.1. Temperature and pressure sensor.* This ROV was equipped with a DS18B20 temperature sensor with a usable temperature of -55 to 125°C to monitor the temperature underwater for any alarming temperature variations. The DF Robot Analog pressure sensor was used to provide real-time pressure data which helps in determining the depth of the ROV. It measures a range of 0-1.6MPa of pressure and serves as an external aiding sensor for position tracking of ORCA by providing depth. The data from these sensors are serially communicated to the microcontroller which is connected to the processor. They are powered by a 5V converter. These sensors were placed along the inner walls of the ROV frame and wired into the enclosure via a slotted acrylic flange as shown in figure 1. The data obtained during the operation of ORCA was remotely monitored by the user and logged.

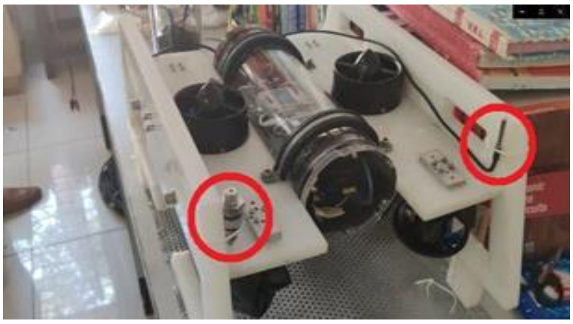


**Figure 1.** Representation of temperature and pressure sensor mounted on ORCA.

*2.1.2. Camera.* A Lowlight HD USB camera with a tilt system was mounted at the front end of the acrylic housing to provide continuous live video footage to the user and to help in smooth navigation underwater. It has an onboard H.264 compression chip where all the video compression is done onboard reducing the load on the microprocessor allowing it to support the addition of multiple cameras in the future for alternate views. It offers a Field of view of 80 degrees horizontally and 64 degrees vertically. The tilt system was PWM controlled by the controller for 180-degree movement.

*2.1.3. Safety sensors.* A safety sensor used in the system is the Blue robotics SOS leak sensor wherein four reusable sponge-tipped probes interfaced with the sensor probe host board, are placed on the corners of the enclosure which helps in monitoring from the penetrators to front and rare flanges for any water leakage into the electronic enclosure. An attractive safety mechanism incorporated in this positively buoyant ROV is the emergency float-up mechanism which involves powering off and floating the vehicle to the surface with an alarm in case of any water leakage into the improperly sealed electronic enclosure. This ensures no damage to the electronics inside the enclosure and prevents any electrical catastrophe due to leakage.

*2.1.4. Inertial Measurement unit.* The prime attraction of ORCA for realizing its efficient navigation is the presence of this industrial-grade HG1120 IMU which contains a MEMS gyroscope and accelerometer for providing compensated incremental angle and velocity data for navigation and angular rates, linear accelerations, and magnetic fields for control through a digital serial interface bus. The IMU measures angular rates(rad/sec), linear acceleration(m/sec$^2$), and magnetic fields(milli-gauss) in a body-mounted strap down configuration along the three axes which are fed to a sensor fusion algorithm to obtain the attitude and heading of ORCA with respect to the earth's frame. The navigation data involving the incremental angles and velocities along the three axes are primarily used for positioning the ROV while the control data is used in the determination of the orientation in Euler angles (roll, pitch, and yaw) and control of the vehicle. Since stand-alone gyroscope has no frame of reference in contrast to accelerometer with earth's acceleration due to gravity as the reference and magnetometer with the earth's magnetic north as a frame of reference, all the three sensors are fused to obtain the complete orientation of the sensor with respect to a navigation frame. The control data frames are filtered and sampled at 1800Hz while the navigation data frames were sampled at 300Hz. The IMU was powered with a 5V converter. It was mounted vertically on the flat tray at the center of gravity of the vehicle such that the x-axis of the IMU as shown in figure 2, coincides with the direction of the vehicle's heading. Further calibration of the IMU after placing it at the COG was done to reduce the bias and obtain precise values.

**Figure 2.** IMU embedded in ORCA.

### *2.2 Electronic circuitry of ORCA*

The electronic circuitry of ORCA is responsible for signalling to and from the control side and data acquisition during the survey. This will be securely sealed inside the electronic enclosure. The master-slave architecture is reflected in the working of Orca with the master being the user PC and the slave being the onboard processor. The T200 thrusters mounted in ORCA involve four thrusters vectored at 45° helping in the horizontal motion along with two vertical thrusters to dive. This configuration thus allows us to actively control the system in 6 degrees of freedom. The sensors required for the inspection were added as payload and integrated with the circuitry. Cat 5 Ethernet cable, having a maximum transmission rate of around 1 Gb/s was used as the tether for transmission of control signals, the camera feed, and the sensor data from the ROV.

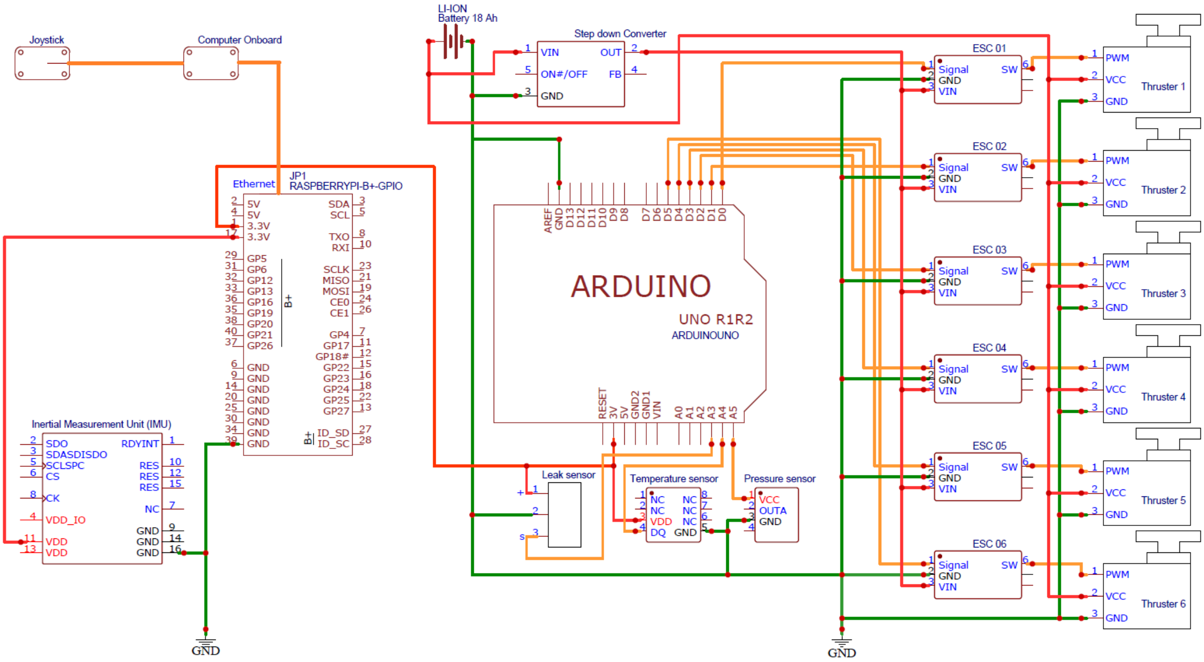


**Figure 3.** Schematic Representation of the Electronic Circuitry in ORCA.

Figure 3 presents the schematic representation of the embedded circuitry in ORCA. It primarily comprises a 64-bit 1.2GHz microprocessor (Raspberry Pi Model B+) interfaced with the inertial measurement unit connected to a 16-bit microcontroller (Arduino Uno R3). Connected to the Arduino are the Electronic speed controllers (ESCs) that are mounted on a flat tray fit inside the leak-proof enclosure. The microprocessor is pivotal in relaying control signals from the user end to the thrust controller board, Arduino. This Arduino further provides PWM signals via the Electronic Speed Controllers (ESCs) to the six T200 thrusters to achieve the desired movement as signalled by the joystick by controlling the speed and direction of the thrusters. The efficient thrust vectorization algorithm developed and embedded on the Arduino hence helps navigation of ORCA in 6 degrees of freedom namely surge, sway, heave, roll, pitch, and yaw. Various inspection sensors as shown are integrated with the Arduino aiding in the underwater survey.

## 3. Navigation System of ORCA

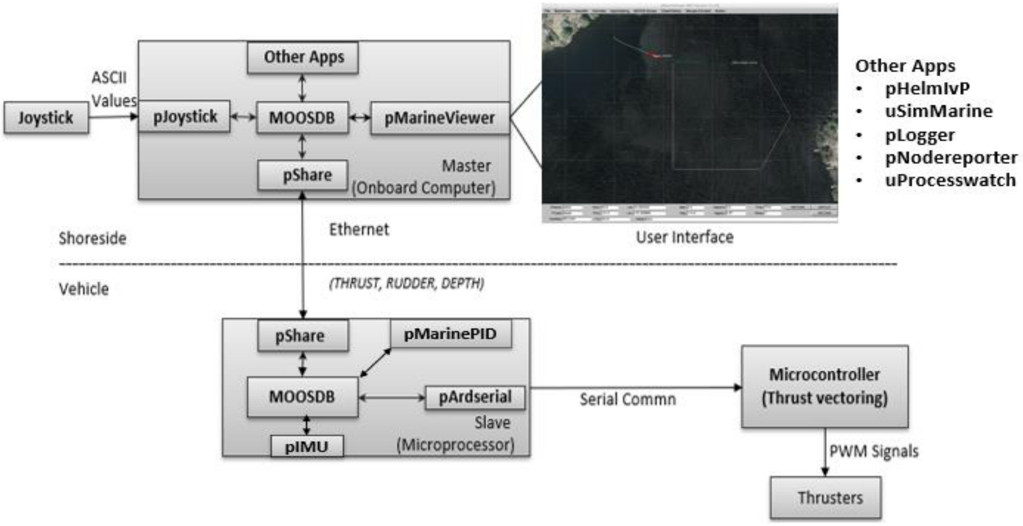


**Figure 4.** The architecture of ORCA navigation and control system using MOOS-IvP

Mission Oriented Operating Suite-Interval Programming (MOOS-IvP), a set of C++ open-source modules has been incorporated in this ROV as the major tool for navigation. Although in a general scenario, MOOS-IvP is widely used only for providing autonomy to robots, the authors have modified and developed it to drive ROVs as well. Field tests to run the vehicle in both autonomous and remotely operated modes have been performed successfully due to the adaptability in design as well as the software making it unique in its style. The layout emphasizing the relation between the different elements of a MOOS Community on the user and the vehicle side is portrayed in figure 4.

Two different MOOS-IvP communities with the "publish-subscribe" principle run on the user computer and ORCA as shown in figure 4. Various applications for running on each community are connected to the MOOSDataBase (MOOSDB) in its community forming a centralized control. The MOOSDB contains all the variables or data published by the respective applications which can be subscribed by other MOOS-Apps based on their requirement. The remote operation of the ROV was achieved by a uniquely developed MOOS Application (MOOS-App), pJoystick which publishes the axes coordinates of the input provided by the user joystick to MOOS-DB (MOOS-Database) on the master side at every clock frequency. These coordinates are mapped to the desired thrust, rudder, and depth before being relayed to ORCA. This enables on-demand movement of the vehicle as per the user's command. uSimMarine was used as the simulator tool wherein the underwater environment was created along with the vehicle's parameters before the actual hardware testing.

The IvP Helm (pHelmIvP) publishes high-level control decisions. The PID controller (pMarinePID) converts the high-level control decisions to low-level actuator decisions. Finally, the simulator (uSimMarine) reads the low-level actuator postings to produce a new vehicle position. Post achieving the navigation of the vehicle as per the user's interest using uSimMarine, the hardware implementation was tested and validated. The pMarineViewer, MOOS Graphical User Interface(GUI) application is used as a geo display which helps the authors to feed the latitude-longitude of the place of inspection along with the map as a .tiff file. Local reference coordinates were defined on the map, to track the vehicle's movement easily. The communication between both the MOOS communities to relay the user's command to the vehicle is achieved using the pShare over a UDP link. This MOOS-App has been developed to publish and subscribe messages that include the thrust, rudder, and depth flows from MOOS-DB at the master to the one at the slave, where it must be requested for subscription by another MOOS-App pArdserial. It involves serially transferring the control details to the Arduino where the thrust vectorization is done to realize the vehicle's movement.

### *3.1. IMU integration with MOOS-IvP*

The integration of the IMU was done by developing a MOOS-App, pIMU. The 9-axis IMU renders the vehicle's current acceleration, angular rate, and magnetic field along x, y, and z axes respectively w.r.t to a reference post initial calibration. A MOOS-App was designed to log the above-mentioned data and serve the current position of ORCA to MOOS as feedback to form a closed-loop system during the run-time and hence enhancing the stability of the vehicle. The logged data in a .CSV file was analyzed as shown in the results. At the time of writing this paper, the authors are currently involved in fusing the data obtained from the IMU with the AHRS algorithm to obtain the orientation of the vehicle in the navigation frame.

### *3.2. Closed-Loop Control System Algorithm in MOOS-IvP*

Highly nonlinear hydrodynamic effects resulting from the interaction with the environment occur, that cannot be quantified. Thus, an open-loop control system wherein the command is provided to the vehicle and corresponding PWM signals are supplied to the thrusters to achieve the motion seems to be ineffective. It was observed that the drift suffered by the vehicle could not be substantiated. Hence, a closed-loop PID control system as shown in figure 5 was implemented to minimize the positional errors and drift of the vehicle thus inducing stability in its movement. The basic PID controller implementation was carried on in MOOS-IvP. The application pMarinePID has three basic PID controllers operating to control heading speed and depth based on incoming desired heading, speed, and depth objectives. On tuning the respective Kp, Ki, and Kd variables using trial and error, the exact fit for achieving minimum error was obtained. Thus, the system can achieve the desired movement and navigate successfully.

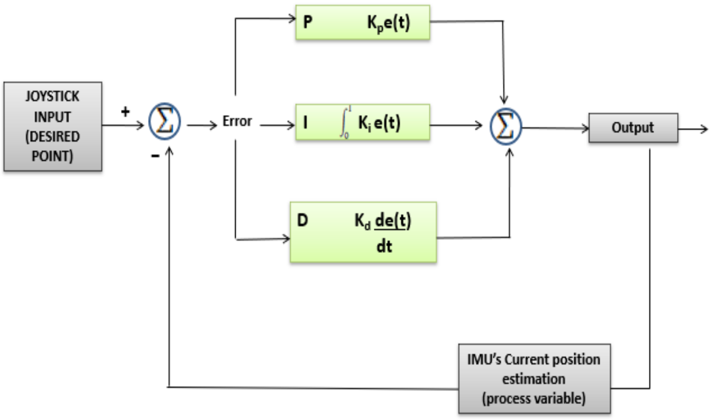
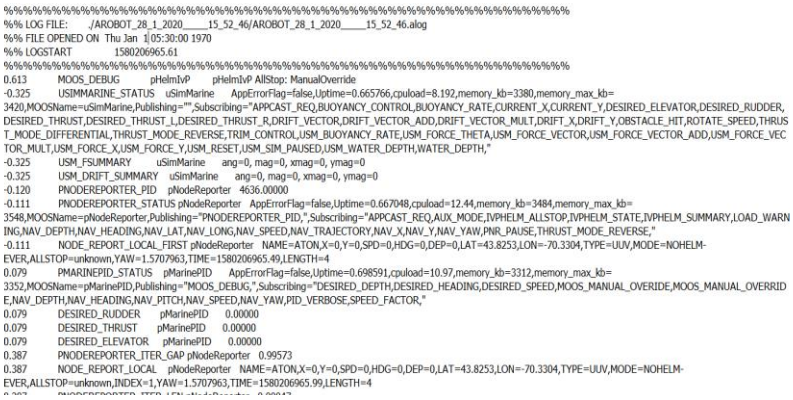

P
$K_p e(t)$
JOYSTICK INPUT (DESIRED POINT)
+
-
Error
I
$\int_0^t K_i e(t)$
Output
D
$K_d \frac{de(t)}{dt}$
IMU's Current position estimation (process variable)

**Figure 5.** Block representation of closed-loop PID controller system used in ORCA.

## 4. ORCA Navigation Test Results

A series of tests were conducted on ORCA for about more than ten days. However, 8 test deployments were made on January 11, 2020, at a pool (length - 25m, breadth - 15m, depth - 1.82m) in The Village Apartments (Lat- long: 12.742437, 80.198086) for ORCA where proper logging of data and thus further analysis based on that was carried out. pLogger MOOS-App was used for successful logging of the data involved in each process or MOOS application as shown in figure 6.

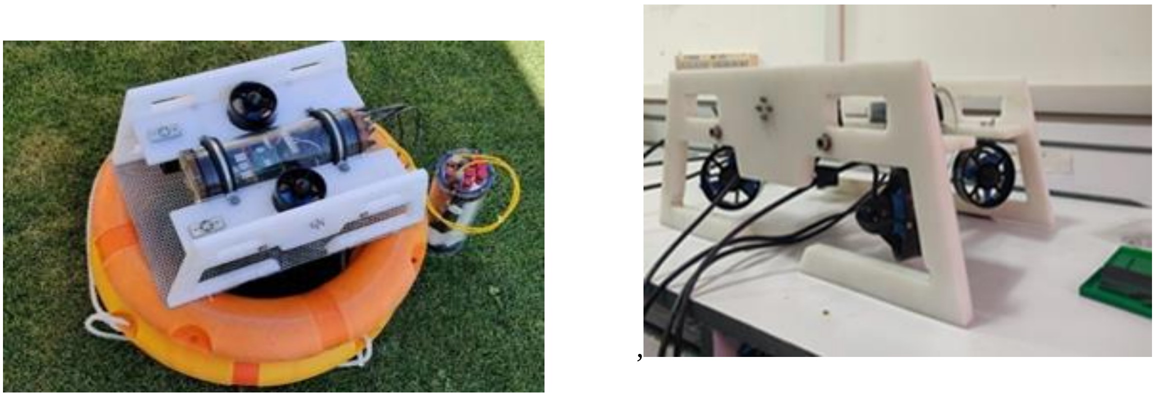


```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% LOG FILE:       ./AROBOT_28_1_2020_____15_52_46/AROBOT_28_1_2020_____15_52_46.alog
%% FILE OPENED ON  Thu Jan  1 05:30:00 1970
%% LOGSTART           1580206965.61
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
0.613         MOOS_DEBUG          pHelmIvP       pHelmIvP AllStop: ManualOverride
-0.325        USIMMARINE_STATUS   uSimMarine     AppErrorFlag=false,Uptime=0.665766,cpuload=8.192,memory_kb=3380,memory_max_kb=
3420,MOOSName=uSimMarine,Publishing="",Subscribing="APPCAST_REQ,BUOYANCY_CONTROL,BUOYANCY_RATE,CURRENT_X,CURRENT_Y,DESIRED_ELEVATOR,DESIRED_RUDDER,
DESIRED_THRUST,DESIRED_THRUST_L,DESIRED_THRUST_R,DRIFT_VECTOR,DRIFT_VECTOR_ADD,DRIFT_VECTOR_MULT,DRIFT_X,DRIFT_Y,OBSTACLE_HIT,ROTATE_SPEED,THRUS
T_MODE_DIFFERENTIAL,THRUST_MODE_REVERSE,TRIM_CONTROL,USM_BUOYANCY_RATE,USM_FORCE_THETA,USM_FORCE_VECTOR,USM_FORCE_VECTOR_ADD,USM_FORCE_VEC
TOR_MULT,USM_FORCE_X,USM_FORCE_Y,USM_RESET,USM_SIM_PAUSED,USM_WATER_DEPTH,WATER_DEPTH,"
-0.325        USM_FSUMMARY        uSimMarine     ang=0, mag=0, xmag=0, ymag=0
-0.325        USM_DRIFT_SUMMARY   uSimMarine     ang=0, mag=0, xmag=0, ymag=0
-0.120        PNODEREPORTER_PID   pNodeReporter  4636.00000
-0.111        PNODEREPORTER_STATUS pNodeReporter  AppErrorFlag=false,Uptime=0.667048,cpuload=12.44,memory_kb=3484,memory_max_kb=
3548,MOOSName=pNodeReporter,Publishing="PNODEREPORTER_PID,",Subscribing="APPCAST_REQ,AUX_MODE,IVPHELM_ALLSTOP,IVPHELM_STATE,IVPHELM_SUMMARY,LOAD_WARN
ING,NAV_DEPTH,NAV_HEADING,NAV_LAT,NAV_LONG,NAV_SPEED,NAV_TRAJECTORY,NAV_X,NAV_Y,NAV_YAW,PNR_PAUSE,THRUST_MODE_REVERSE,"
-0.111        NODE_REPORT_LOCAL_FIRST pNodeReporter  NAME=ATON,X=0,Y=0,SPD=0,HDG=0,DEP=0,LAT=43.8253,LON=-70.3304,TYPE=UUV,MODE=NOHELM-
EVER,ALLSTOP=unknown,YAW=1.5707963,TIME=1580206965.49,LENGTH=4
0.079         PMARINEPID_STATUS   pMarinePID     AppErrorFlag=false,Uptime=0.698591,cpuload=10.97,memory_kb=3312,memory_max_kb=
3352,MOOSName=pMarinePID,Publishing="MOOS_DEBUG,",Subscribing="DESIRED_DEPTH,DESIRED_HEADING,DESIRED_SPEED,MOOS_MANUAL_OVERIDE,MOOS_MANUAL_OVERRID
E,NAV_DEPTH,NAV_HEADING,NAV_PITCH,NAV_SPEED,NAV_YAW,PID_VERBOSE,SPEED_FACTOR,"
0.079         DESIRED_RUDDER      pMarinePID     0.00000
0.079         DESIRED_THRUST      pMarinePID     0.00000
0.079         DESIRED_ELEVATOR    pMarinePID     0.00000
0.387         PNODEREPORTER_ITER_GAP pNodeReporter   0.99573
0.387         NODE_REPORT_LOCAL   pNodeReporter  NAME=ATON,X=0,Y=0,SPD=0,HDG=0,DEP=0,LAT=43.8253,LON=-70.3304,TYPE=UUV,MODE=NOHELM-
EVER,ALLSTOP=unknown,INDEX=1,YAW=1.5707963,TIME=1580206965.99,LENGTH=4
```

**Figure 6.** pLogger publishing logged data to other MOOS-Apps.

### *4.1. Assembly and Preliminary Bench Tests*

The ROV was successfully assembled and proved to be leakproof as shown in figures 7(a) and 7(b). All the electronics were placed in the enclosure and sealed tightly. Preliminary bench tests as in Fig.10 were performed to realize the movement of the vehicle as per the joystick command and satisfactory results were obtained. Simultaneously, respective temperature and pressure sensor readings after calibration were logged continuously at each clock frequency.



,

**Figure 7(a) and 7(b):** Fully fabricated ORCA with Integrated Navigation System

### *4.2. Testing, Operation, and Post-Processing Stages*

Finally, the system was deployed in the pool as in figure 8 and multiple straight-line missions were accomplished with an open-loop system for 5-15 mins each to study the behaviour of the system along with the IMU connected. The data collected during these missions were visualized into graphs for comprehensive analysis of the various environmental parameters on the vehicle and to study the dynamics of it as it moves.

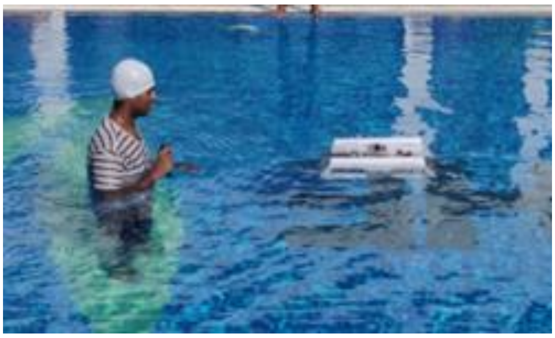


**Figure 8.** Real-time deployment in a pool.

These missions involved the movement of the vehicle in a straight line at a constant depth, hence the acceleration along the x-axis in the graph has a small value due to the forward surge, while in the z-axis it is close to 9.8 m/s$^2$, acceleration due to gravity ($A_z$ -green) as shown in figure 9(a). ROV suffered a deviation to its port side due to water circulation in the pool which explains the spike in the angular velocity in the z-direction ($G_z$ -green) as shown in figure 9(b).

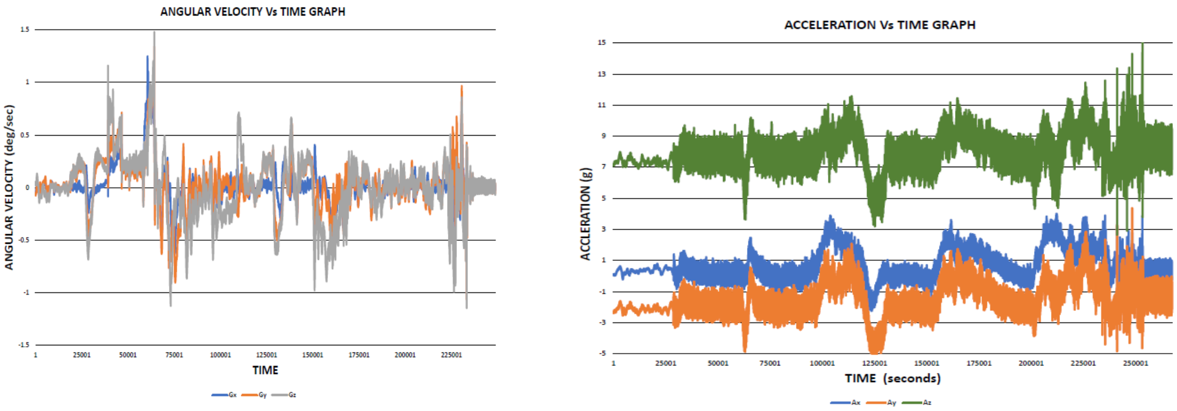


**Figure 9. (a).** IMU post-processed graphs - acceleration vs time **(b).** IMU post-processed graphs - Angular velocity vs. time

## 5. Conclusion and Future Work

This paper presents the development of a navigation and control system for an ROV, which was tested and experimented on a research class ROV, ORCA. This system is equipped with temperature, pressure, and leak sensors to monitor the ambient parameters. The ROV's navigation is aided by the IMU which is integrated with MOOS-IvP as a closed-loop system. The GUI of MOOS-IvP helps in identifying the real-time position of the ROV to precisely navigate inside the underwater environment. The major advantage of MOOS comes in its ability to run the vehicle autonomously. Trial runs in the pool with predefined missions were also carried out successfully, where the IMU data was logged and analysed. The current scope of work involves the fusion of sensor data from IMU into useful positioning information using the Attitude Heading Reference System (AHRS). Moreover, a safety protocol has been developed to aid the recovery of the vehicle in case of power failure, tether snap, or any detection of water ingress in the electronic chamber. This module, the Emergency Float-Up Mechanism eliminates the possibility of losing the vehicle in case of unforeseen circumstances. External sensors like pH, salinity, and turbidity will be incorporated into the system. The images stored in the repository will be used to develop a Map-Based Localization technique along with MOOS-IvP to enhance autonomy. This can be further extended by pre-feeding the maps of specific environments to automatically reduce the speed of the vehicle at zones prone to failure, leading to better inspection underwater**.**

**Acknowledgement**

This work is a part of a project funded by Sri Sivasubramaniya Nadar College of Engineering, Chennai, under the scheme of the Student Consortium Project. The authors would like to thank the other members of Team Starboard, Mr. Vigneshwar Veeravagu, Mr. Vishal Mohan, Ms. Raksshitha N J, Mr. Hashmat Jeelani Banday, and Mr. Guru Prasad Gupta for their constant support. The authors also wish to extend thanks to the faculty associated with this project, Dr. R Vimal Samsingh and Dr. K Murugesan for their constant mentorship.

## References

[1] Scaradozzi D Conte G, Zanoli S M and Gambella L 2006 Acoustic mapping and localization of an ROV *14th Mediterranean Conference on Control and Automation* pp 1-6.

[2] Capocci R, Dooly G, Omerdić E, Coleman J, Newe T, Toal D 2017 Inspection-class remotely operated vehicles—A review *Journal of Marine Science and Engineering* Mar **5(1)** p 13.

[3] Chutia S, Kakoty N M and Deka D 2017 A review of underwater robotics, navigation, sensing techniques and applications *ACM Int. Conf. Proc. Series*, **Jun 28** pp 1-6.

[4] Karras G C, Panagou D J and Kyriakopoulos K J 2006 Target-referenced localization of an underwater vehicle using a laser-based vision system *Oceans 2006* **Sept 18** pp 1-6.

[5] Leonardi M, Stahl A, Gazzea M, Ludvigsen M, Rist-Christensen I and Nornes S M 2017 Vision based obstacle avoidance and motion tracking for autonomous behaviors in underwater vehicles *OCEANS - Aberdeen* **Jun 19** pp 1–10.

[6] Aguirre-Castro O A, Inzunza-González E, García-Guerrero E E, Tlelo-Cuautle E, López-Bonilla O R, Olguín-Tiznado J E, Cárdenas-Valdez J R 2019 Design and Construction of an ROV for Underwater Exploration *Sensors 2019* **Jan 19 (24):5387** pp 1-25.

[7] Arnesen B O 2016 *Motion Control Systems for ROVs*.

[8] García-Valdovinos, L G, Salgado-Jiménez T, Bandala-Sánchez M, Nava-Balanzar L, Hernández-Alvarado R and Cruz-Ledesma J A 2014 Modelling, design and robust control of a remotely operated underwater vehicle *Int. J. of Adv. Robotic Syst.* **Jan 9 11(1)** p 1.

[9] Chin C S and Lau M W S 2020 *Benchmark Models of Control System Design for Remotely Operated Vehicles*. (Singapore: Springer Nature)

[10] He Y, Wang D B and Ali Z A 2020 *Measurement and Control.* (United Kingdom)

[11] Aras M S M, Sulaiman M, Keong Y E, Afif Kasno M, Kassim A M and Khamis A 2017 Performance analysis of PID and fuzzy logic controller for unmanned underwater vehicle for depth control *J. of Telecommunication, Electronic and Computer Engineering* **9** pp 59–63.

[12] Abidin A Z, Mardiyanto R and Purwanto D 2017 Implementation of PID controller for hold altitude control in underwater remotely operated vehicle *Proc. - 2016 Int. Seminar on Intelligent Technology and Its Application : Recent Trends in Intelligent Computational Technologies for Sustainable Energy* **20** pp 665–70.

[13] Rúa S and Vásquez R E 2016 Development of a low-level control system for the ROV Visor3 *Int. J. of Navigation and Observation* **2016** .

[14] Zul Azfar A and Hazry D 2011 A simple approach on implementing IMU sensor fusion in PID controller for stabilizing quadrotor flight control *Proc. - 2011 IEEE 7th Int. Colloquium on Signal Processing and Its Applications* **March 4** pp 28–32.

[15] Nagodagamage S K , Attanayaka A M D C, De Silva G S C and Perera M S R 2015 *AQUABOT : Hydro view Real-Time Video Streaming Remotely Operated Vehicle*. (SriLanka)

[16] Chin CS 2012 *Computer-Aided Control Systems Design: Practical Applications Using MATLAB and Simulink*. (Florida : CRC Press)