TERI-Underwater-Biorobot (Turtle in water)

Design Optimization and Bill of Materials

In the mechanical and hardware optimization of the TERI robot, practical feasibility and system integration were prioritized based on the initial conceptual design. Given the limited internal space and budget constraints, hardware selection required a careful balance between performance, size, cost, and ROS compatibility. The resulting bill of materials [1] reflects a well-integrated, fully functional, and cost-effective design (approximately £332). Mechanical optimization focused primarily on the internal layout. Component placement was carefully planned to ensure balanced weight distribution and to prevent hardware interference—such as obstruction of the LiDAR sensor.

Motion Control and Autonomous Navigation

TERI's underwater mobility is enabled through a combination of precise motion control and intelligent navigation algorithms. The robot employs the Joint-Position-Controller plugin, which uses a PID control algorithm to accurately regulate joint angles, ensuring controlled fin oscillations. However, due to limitations in the Hydrodynamics plugin's ability to simulate fin-generated propulsion, the Thruster plugin was integrated to replicate the reactive forces from water. During early testing, unbalanced reaction torques caused unintended roll-axis rotation, which was mitigated by inverting the thrust coefficient on one side, achieving torque equilibrium and stable motion.

In parallel, TERI's autonomous navigation system combines LiDAR-based perception, obstacle avoidance, and state machine logic to enable reliable movement in complex underwater environments. Sensor data is processed in real time within a callback function, where point cloud data is filtered based on horizontal and vertical scanning thresholds to detect nearby obstacles. A state machine mechanism controls the robot's rotational behavior, allowing it to escape from symmetrical trap scenarios and resume straight-line navigation once obstacles are cleared. Extensive simulation tests demonstrated the robot's ability to independently traverse the environment and avoid collisions (Navigation video is available at GitHub [1])—highlighting the system's robustness and adaptability in underwater exploration tasks.

Object Identification

The TERI robot achieves real-time visual recognition of red spherical targets within the Gazebo simulation environment. The system captures images using a camera and transmits the data via ROS 2 topics. The object_recognizer node employs CvBridge to convert ROS image messages into OpenCV format for processing, with annotated detection results displayed in real time (Fig 1). The recognition pipeline includes HSV color space conversion (using two hue ranges to accurately detect red), image denoising (via morphological opening and closing operations to eliminate noise and reconnect fragmented regions), and contour detection (filtering targets based on area and circularity). Once a red circular object is detected, the system publishes the results to /red_circle_detected and /red_circle_info topics, enabling interaction with the motion control module—causing the robot to stop and park next to the target. This closed-loop system of image acquisition, recognition, and communication significantly enhances the robot's perception capabilities in complex underwater environments. Object detection video is available at GitHub [1].

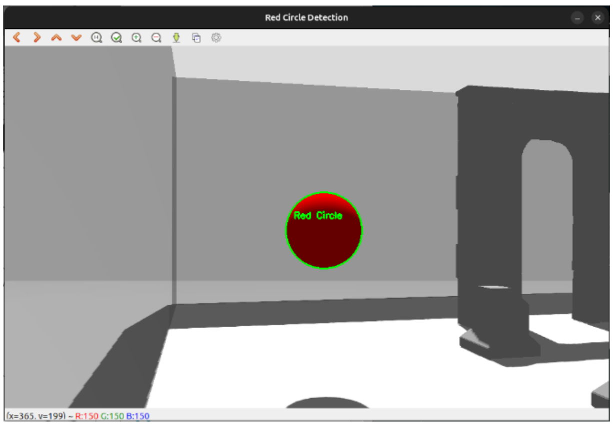

Fig 1. Detection result

World Environment and SLAM

To support the testing and development of the biomimetic underwater robot TERI, a custom simulation environment named Atlantis.world was created in Gazebo. This underwater world replicates realistic conditions using buoyancy, hydrodynamics, and physics plugins, and contains diverse obstacles such as shipwrecks, stairs, spheres, and cubes. These elements were modeled in SolidWorks and imported as STL files. The varied obstacle layout provides multiple challenges for TERI's navigation, object detection, and mapping capabilities.

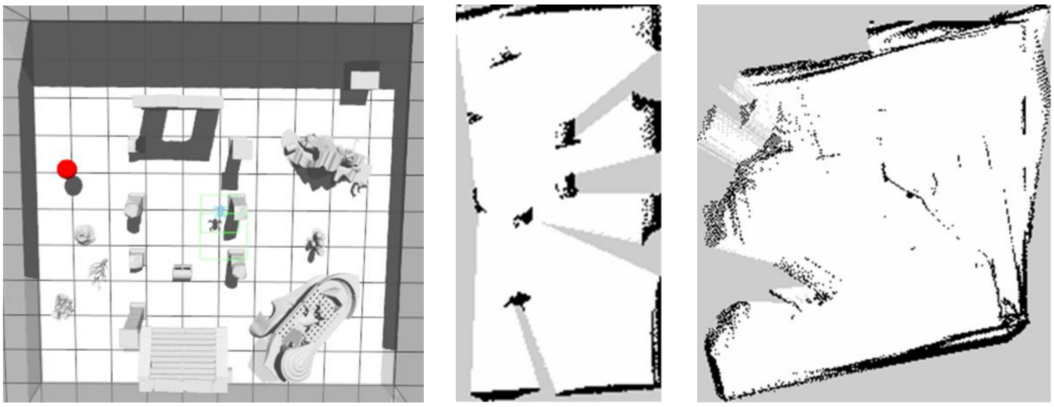

Fig 2. Comparison of actual world small area map and larger area map

TERI has the ability to build a map of an unknown environment of its surroundings (SLAM video is available at GitHub [1]). Simultaneous Localisation and Mapping (SLAM) is used to construct a 2D occupancy grid while simultaneously tracking its position within the environment.

- **Map Generation** – SLAM toolbox was configure in asynchronous mode to process real-time LiDAR data and produce a dynamic map. The system successfully updates in simulation however mapped areas are overwritten or blurred.
- **Testing Scenario** – Different obstacle layouts and starting points were tested, allowing the mapping performance to be assessed. When placed in open areas with clear LiDAR visibility the mapping performance improved. Starting near walls or obstacles led to scan distortion and increased mapping errors.
- **Accuracy of Generated Maps** – The maps created do closely match the simulation environment, initially, however as larger areas are mapped the accuracy of the map degrades (Fig 2). Scan artefacts degrade the overall accuracy.

Contribution

Qi Cheng: Developed the robot's motion control and navigation algorithms; implemented object vision recognition with OpenCV; led mechanical and hardware design with BOM.

Mia Hornett: Developed the SLAM system; configured sensors and data flow for accurate simulation.

Vincent Tirta: Created the simulated underwater world in Gazebo; configured physics and buoyancy plugins for realistic motion.

Aurora Brissenden: Introduced motion control and navigation algorithms algorithms in bench

Reference

The project-related code, CAD, detailed report, video, and Bill of Materials (BoM) can be found in the following GitHub repository.

[1] Cheng, Q., Hornett, M., & Collin, V. (2025). ELEC330_TurtleInSea: Biomimetic Underwater Robot Simulation. GitHub repository. Available at: https://github.com/cq20021111/ELEC330_TurtleInSea