TECHNICAL DESIGN REPORT ASUQTR RoboSub 2026

Université du Québec à Trois-Rivières (UQTR) Autonomous Underwater Vehicle Team (ASUQTR)

Louis Emond
Embedded Control
Electrical Engineering
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada
louis.emond2@uqtr.ca

Pietro Lacommande
Power Electronics
Informatics Engineering
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada
pietro.lacommande@uqtr.ca

Elliot Labrie-Emard
Digital Twin and Imaging
Electrical Engineering
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada
elliot.labrie-emard@uqtr.ca

Émile Lahaie
Control Modelling
Electrical Engineering
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada
emile.lahaie@uqtr.ca

Lucas Martineau
Dashboard and Mission Behaviours
Informatics Engineering
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada
lucas.martineau@uqtr.ca

Louis Lavallée
Control PCB
Electrical Engineering
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada
louis.lavallee@uqtr.ca

Jean-Michel Farlay-Marchand
IMU Integration and Filtering
Electrical Engineering
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada
jean.michel.farley.marchand@uqtr.ca

Loic Pouget
Software Architecture
Mechatronics Engineering
Université du Québec à Trois-Rivières
Trois-Rivières, Québec, Canada
loic.pouget@uqtr.ca

Abstract

The ASUQTR autonomous underwater vehicle (AUV) is developed for the RoboSub 2026 competition. The platform combines a modular ROS2 software architecture, redesigned electronics, sensor fusion, model-based control, and simulation-driven validation. Vehicle state estimation is performed using an Extended Kalman Filter (EKF) integrating IMU, DVL, and pressure sensor measurements. A Linear Quadratic Regulator (LQR) controller is used for 6-DOF stabilization and navigation. The

project also includes EMI mitigation strategies, Unity digital twin simulation, ROS bag replay workflows, and integrated safety systems. Current development focuses on improving robustness, validating real-time implementation, and preparing the platform for autonomous RoboSub missions.

1. Competition Strategy and System Evolution

The ASUQTR RoboSub strategy focuses on developing a reliable and scalable autonomous underwater platform capable of progressively increasing mission complexity. Instead of maximizing feature count, the project prioritizes robustness, repeatable testing, hardware reliability, and modular subsystem integration.

Previous RoboSub participations revealed limitations related to unstable autonomous control, electromagnetic interference affecting the IMU, integration complexity under ROS1, and fragile power distribution. The 2025-2026 development cycle therefore focused on redesigning critical subsystems rather than only tuning the controller.

Current development targets reliable autonomous qualifications, stable hovering, waypoint navigation, torpedo alignment, and future manipulation tasks. Long-term continuity and maintainability were considered throughout the redesign process.

Figure 1 – ASUQTR Autonomous Underwater Vehicle

Overview of the current AUV platform.

2. Mechanical, Electrical and Safety Architecture

The vehicle uses a modular frame supporting rapid maintenance, subsystem accessibility, and future upgrades. Eight BLDC thrusters provide the maneuverability required for underwater station-keeping and navigation.

The electrical architecture is currently being redesigned. The power distribution PCB is still under development, while the control PCB remains under programming and testing. These updates aim to improve wiring organization, maintainability, monitoring, and integration with the vehicle’s safety systems.

The safety architecture currently includes a magnetic kill switch that can be physically removed to disable motor power during testing or emergency situations. Additional safety features such as leak detection, current monitoring, and thermal monitoring are used.

Special attention is being given to electromagnetic compatibility, since previous tests showed that motor drivers and power electronics could disturb IMU yaw estimation. Improved wiring layout, grounding, shielding, filtering, and component placement are being considered to improve navigation reliability.

Figure 2 – Thruster Configuration and 6-DOF Motion (Dashboard)

Thruster placement and vehicle force generation.

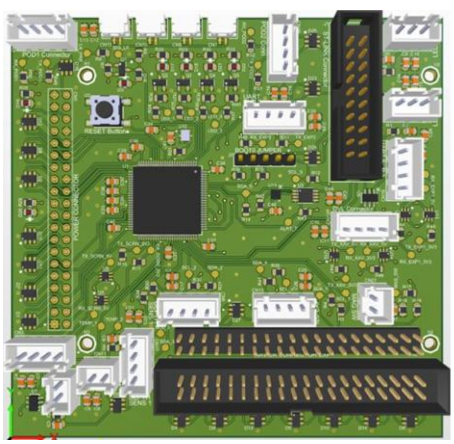


Figure 3 – Redesigned Power and Control Electronics

Integrated power distribution and PCB architecture.

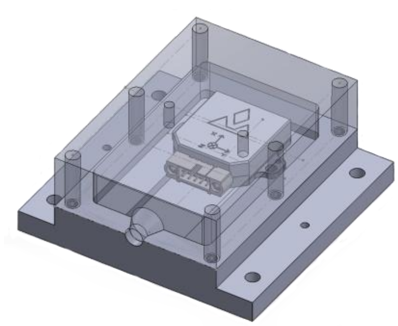


Figure 4 – EMI Mitigation and IMU Shielding

Measures implemented to reduce sensor disturbances.

3. Software Architecture and State Estimation

The software architecture is based on ROS2 and follows a modular node-based design. Responsibilities are separated between state estimation, control, actuator management, mission logic, diagnostics, and simulation.

The migration from ROS1 to ROS2 improved modularity, reliability, Quality of Service (QoS) management, and future scalability. Docker containers are used to package the software environment on the NVIDIA Jetson Xavier platform.

Vehicle state estimation is performed using robot_localization and an Extended Kalman Filter (EKF). Measurements from the VectorNav IMU, DVL, and pressure sensors are fused to produce a coherent 6-DOF estimated state.

Validation efforts focused on TF2 frame consistency, sign conventions, covariance tuning, actuator mapping, and alignment between the estimated state and the control model.

4. Guidance, Navigation and Control

The control architecture uses a model-based Linear Quadratic Regulator (LQR). The vehicle is represented using a 6-DOF state-space model:

$$x = [X,Y,Z,roll,pitch,yaw,u,v,w,p,q,r]^T$$

MATLAB and Simulink are used to generate and validate the linearized system matrices. The controller computes generalized vehicle efforts which are converted into individual thruster forces through a thrust allocation matrix.

Validation efforts focused on consistency between MATLAB-generated matrices, runtime ROS2 implementation, actuator mapping, coordinate-frame conventions, and force-to-PWM conversion.

Simulation studies were used to evaluate trajectory tracking, stability margins, sensor-frequency effects, and pole stability under different operating conditions.

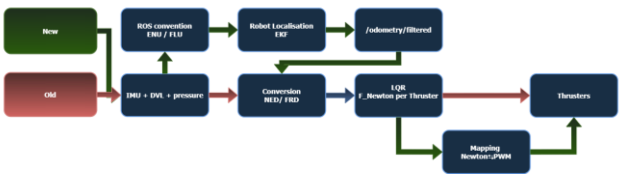


Figure 6 – LQR Control Pipeline

Complete EKF-to-actuator control loop.

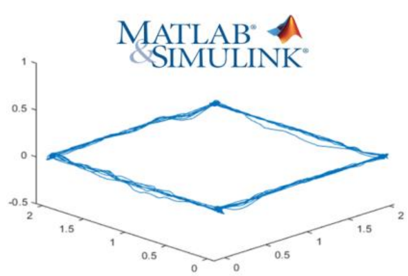


Figure 7 – MATLAB Validation Results

Trajectory tracking and stability analysis.

5. Simulation, Diagnostics and Validation

A Unity-based digital twin was developed to support simulation-driven development and reduce dependency on physical pool access. The simulator integrates vehicle physics, virtual sensors, ROS communication, and mission testing.

The Unity environment communicates with ROS through rosbridge and supports validation of waypoint navigation, control logic, and autonomous behaviors before real-world deployment.

ROS bag recording is used extensively for debugging and replay-based validation. Sensor and actuator data can be replayed offline to reproduce identical test conditions and validate modifications without requiring pool access.

Additional validation tools include telemetry dashboards, runtime diagnostics, actuator testing utilities, and systematic data recording workflows.

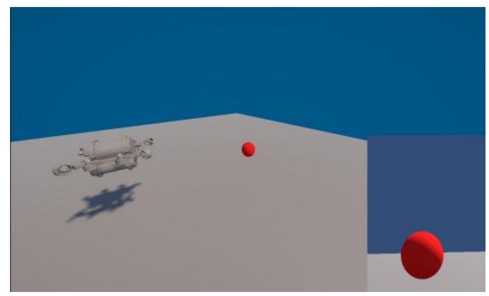


Figure 8 – Unity Digital Twin

Simulation environment used for autonomous validation.

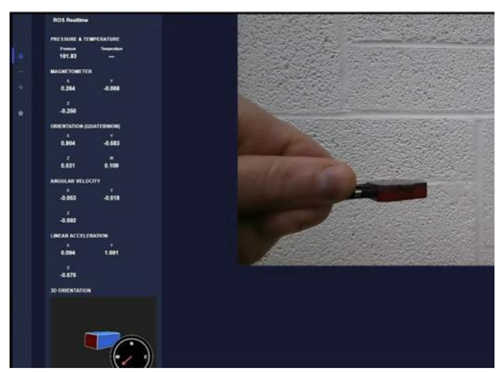


Figure 9 – Telemetry and Diagnostics Dashboard

Runtime monitoring and debugging tools.

6. Current Limitations and Future Work

Although the current architecture shows progress toward subsystem integration, additional development is still required before full competition deployment. The power distribution PCB redesign is not yet fully completed, and the control PCB remains under programming and testing. These hardware elements must be finalized and validated before extended autonomous operation.

Future work includes experimental model validation, improved thruster characterization, extended autonomous pool testing, real-time timing analysis, and expanded autonomous mission behaviors. Once the platform is stable, improvements such as gain scheduling, feedforward control, integral augmentation (LQI), and deterministic real-time control implementation may be considered.

Conclusion

The ASUQTR RoboSub platform demonstrates a modular and engineering-driven approach to autonomous underwater vehicle development. Through the integration of ROS2, sensor fusion, redesigned electronics, model-based control, digital twin simulation, and hardware protection systems, the project establishes a strong foundation for reliable underwater autonomy and long-term team continuity.

References

T. I. Fossen, Handbook of Marine Craft Hydrodynamics and Motion Control, Wiley, 2011.

ASUQTR Internal ROS2 and Control Documentation, 2026.

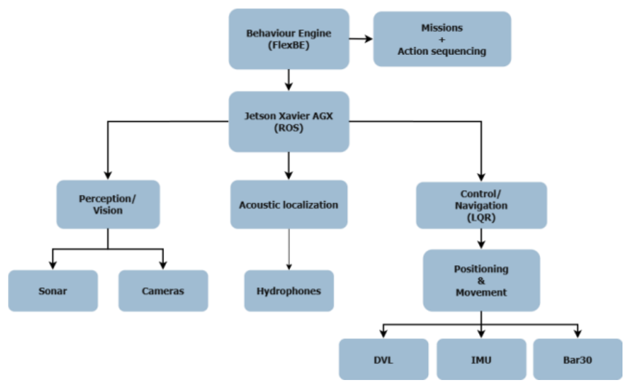
G. Holm, Optimal Control Model for an Autonomous Underwater Vehicle, KTH, 2020.

C.-J. Wu, 6-DoF Modelling and Control of a Remotely Operated Vehicle, Flinders University, 2018.

APPENDIX A – ROS2 Control Architecture

The current control pipeline is divided into four major subsystems: state estimation, frame management, LQR control, and actuator management. Sensor measurements from the IMU, DVL, and pressure sensor are fused using Robot Localization before being processed by the LQR controller.

TF2 transformations ensure that all measurements and references are expressed in consistent coordinate frames. Validation efforts focused on frame consistency, sign conventions, actuator saturation, and force-to-PWM conversion.

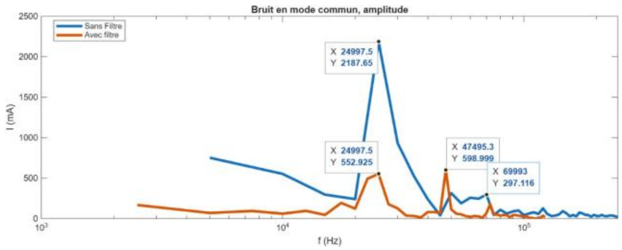


Appendix Figure A1 – ROS2 Control Flow

Detailed ROS2 control chain.

APPENDIX B – EMI Mitigation

Electromagnetic interference generated by motor drivers and switching power electronics previously caused instability in IMU yaw estimation. Shielding, filtering, and improved grounding were implemented to improve sensor stability.



Appendix Figure B1 – EMI Shielding Results

Shielding and filtering results.