RoboBoat 2025: Technical Design Report
NAVIER USN

1st Arve Solberg*, 2nd Don Bullecer*, 3rd Lise Trehjørningen*, 4th Nora Nicolaisen*, 5th Sølve Kjelseth*
University of South-Eastern Norway
Navier USN
Horten, Norway
*Emails: 248583@usn.no, 262684@usn.no, 253819@usn.no, s374184@oslomet.no, 274153@usn.no

***Abstract*—**
Developing an autonomous surface veichle (ASV) is a significant challange, with countless potential solutions. This technical design report describes Navier's approach for RoboBoat 2025, focusing on the first year of its two-year strategy, consisting of observation and learning, with the second year dedicated to improvement and completion. The design strategy features mechanical, hardware, and software components with a modular architecture, making it adaptable, reliable and consistent with our competition strategy. It builds upon previous experience, with the hull constructed in-house, and a propulsion system designed to enhance maneuverability. The integration of advanced sensing systems such as LiDAR and GNSS/INS, combined with robust navigation algorithms, ensures the ASV achieve precise and reliable control in dynamic enviroments. This years effort prioritize refining these core systems through simulation and physical testing, laying the foundation for continued development in the second year.

I. INTRODUCTION

Fig. 1: The Navier team after defending their title in the Autodrone 2024 Championship

Navier USN is a multidisciplinary student organization at the University of South-Eastern Norway that designs and builds autonomous surface vehicles to compete with teams worldwide. The team secured first place in Autodrone in 2023 and 2024 and a respectable fourth place in the Njord Challenge 2023, with the legacy ASV Navier-1. Last year, Navier USN introduced Triton. Their new and improved ASV, earning first place in Njord Challenge 2024. This year, the team is ready to tackle new challenges.

Triton features a custom design with considerations for deployment, ease of maintenance, stability, and fluid dynamics. It utilizes two T200 and two T500 thrusters to achieve three degrees of freedom in motion. The ASV uses GPS with RTK to track its global position and employs sensor fusion between LiDAR and camera to detect and track objects. The software is developed in-house on the ROS2 framework using Python and C++.

Since the organization's inception in 2022, the team has grown to almost 20 members working in various disciplines, including mechanical, electrical, and data engineering, as well as economic, marketing, administrative, and nautical roles. The result is a highly functional team that advances new technologies and solutions to greater heights while also serving as the front-facing organization of USN to recruit new students into the university. This technical report will showcase the mechanical, electrical, and software solutions the team has deployed to develop the new ASV, Triton.

II. COMPETITION STRATEGY

A. Vision

The dream for all teams is to win, this requires good planing and careful consideration of many aspects. Navier USN has therefore strategized a two-year plan, this year the main goal is to complete a recon mission. This consists of the team trying out Triton in all tasks it is physically equipped for, even though a win is unlikely. In addition, the team will observe the competition, collect data and get inspiration. In 2026 Navier USN will return with a refined Triton, optimized for the challanges ahead and better prepared to secure victory.

B. Strategies

The main strategy for RoboBoat 2025 is to try, to push limits and to observe where limits are hit and what corrections are needed. This includes testing how the GPS system and cellular network connection will work internationally. Testing how well the computer model recognizes important objects. Gather data, including photos of the objects is very important, as making our own dataset has given the best results in other competitions.

C. Trade-offs

The main obstacle this years is limited preperation time. Which is why reliability, documentation and testing are the primary focus. After completing RoboBoat 2025, the team will have more time to prepare for 2026 and bring back valuable experience. The plan is to make necessary additions and modifications to Triton to ensure it is able to complete all tasks. This will give Navier USN the best chance to secure first place.

III. VESSEL DESIGN

A. Hull Design - Intro

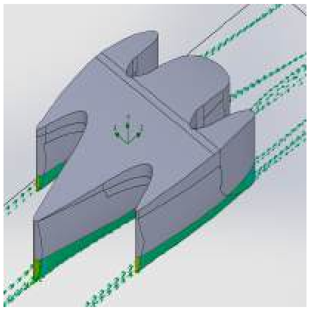
The hull for Triton has undergone significant modifications compared to the legacy design, driven by Navier USN's advancements in autonomy and boat engineering. In the autumn of 2023, our mechanical engineering team redesigned the hull as part of their "Product Design" course. During this course, we collected valuable feedback and information from users of the legacy hull, incorporating their suggestions for future improvements. Additionally, we enhanced our knowledge of hull design, stability, and buoyancy. The primary objectives were to develop a larger, modular hull capable of component interchangeability and concept testing, and to enable 360-degree maneuverability. Concurrently, we aimed to maintain the key attributes of the previous hull, including stability, watertightness, structural strength and a conventional look.

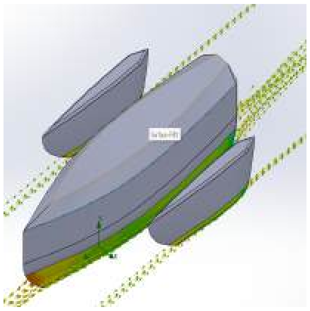

(a) Legacy hull


(b) Triton

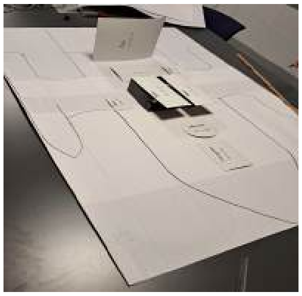
B. Design overview

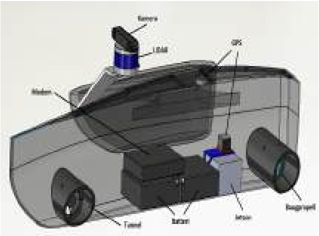
The new hull is a trimaran, which leverages the characteristics of both monohull and catamaran designs to provide superior buoyancy and hull stability. Simulations were conducted in SolidWorks to test different hull designs and their effects on buoyancy. Iterative adjustments were made to optimize performance without compromising its appearance. These simulations were complemented by real-world water tests.


(a) Design 1


(b) Design 2

Additionally, buoyancy calculations and stability simulations were performed to ensure optimal hull performance. Compared to the previous hull, which was under-dimensioned and had components placed haphazardly, the new hull accommodates all larger and heavier components internally. This design offers enhanced protection against water ingress and improved stability in the water. The final hull dimensions are 140 cm in length, 90 cm in width, and it weighs 20 kg without components.


(a) Early general arrangement idea



(b) General arrangement

C. Materials and manufacturing

Initially, the hull was intended to be 3D printed in ASA (Acrylic Styrene Acrylonitrile) due to its strength and water resistance. However, ASA is challenging to print with, requiring higher and more stable temperatures, which led to the breakdown of the 3D printer used. Consequently, it was decided to print the hull in PLA and reinforce it with fiberglass, leveraging the group's prior experience with this method.


(a) 3D print assembly


(b) Fiber glass application

D. Propulsion system

The ASV's propulsion system relies on thrusters, essential for maneuverability, stability, and efficiency. Specifically, we employ T500 thrusters for primary propulsion and T200 thrusters as tunnel thrusters. Although both thrusters can operate with more power at higher voltages, powering them at 12V ensured a simpler electrical system, as other equipment would run at the same voltage [1].

E. Batteries

The ASV is powered by three 12V Skanbatt Bluetooth Lithium batteries, two 12V 20Ah batteries with 30A BMS and one 12V 50Ah battery with 50A BMS. This configuration ensures robust and reliable power management [2]. The batteries are designed to balance capacity and weight, and because of the high energy density of LiFeP04 cells long runtimes are experienced without fear of losing power. the Bluetooth feature allows for easy monitoring and management through a mobile application without needing to unscrew the lid [3].

F. Sensors

Our ASV employs a suite of advanced sensors to achieve its autonomous capabilities. These sensors, including an enhanced GPS system, camera, and multiple LiDAR units, collect vital data on the boat's environment and position. By integrating GPS data, LiDAR data, and images, we construct a detailed digital representation of our boat's surroundings, facilitating precise autonomous navigation and obstacle avoidance.

1) Camera: Stereolabs' ZEDX camera [4] is used in combination with the YOLOV8 framework [5] to detect and identify objects. This camera provides state-of-the-art computer vision capabilities, capturing stereo images to generate detailed depth maps. By utilizing its advanced image processing algorithms, the ZEDX enhances object detection and situational awareness.

2) LiDAR: Our ASV is equipped with two LiDAR sensors to optimize detection and navigation:

- **ROBIN E LiDAR:** This LiDAR sensor features a 120-degree field of view (FOV) and a 0.1-degree resolution, offering superior detection accuracy and extended range. The Robin E LiDAR works in conjunction with the ZEDX camera. It helps determine the distance to detected objects by overlaying its laser beams onto the camera's image and transforming them into the same coordinate system [4].
- **Velodyne VLP-16 LiDAR:** Serving as a general 360-degree collision avoidance sensor. With 16 laser channels, the VLP-16 provides a comprehensive 360-degree field of view and a range of up to 100 meters, enabling detailed environmental mapping [6].

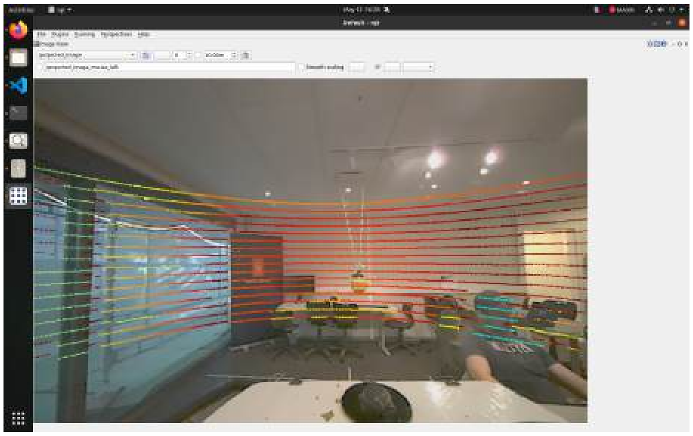


Fig. 6: LiDAR points projected onto the camera image, demonstrating the integration of depth and visual information.

3) XSENS: The XSENS MTi-680G RTK GNSS/INS [7] provides smaller than 1 cm locational accuracy, allowing us to move the boat with high precision [8]. It combines GNSS and inertial sensing to provide exact positioning, velocity, and orientation information, supporting various demanding navigation and control applications [9].

G. Network

The ability to communicate with the ASV remotely is done with a Celerways Stratus 5G. It facilitates high-speed network connectivity, enabling real-time data transmission and remote control capabilities [10].

H. Computation

The powerhouse running all software is the NVIDIA Jetson Orin AGX. It acts as the central processing unit, handling data from sensors and running complex algorithms for autonomous operations [11].

IV. SOFTWARE

A. Software design

The software infrastructure of the ASV is designed to efficiently handle sensor data, support navigation decisions, and manage the propulsion system in real time. Utilizing Python for its ease of use and rapid development capabilities, and C++ for performance critical tasks. The choice of programming languages was driven by the operational requirements of the ASV and the development team's expertise.

B. ROS 2 Humble Hawksbill

ROS 2 Humble Hawksbill is a version of the Robot Operating System (ROS 2), chosen for its improved performance, security features, and flexibility. This version of ROS supports a wide range of hardware and software, and has strong community support, making it a key tool for developing complex and scalable robotics applications. For more details, refer to the official documentation [12].

C. NVIDIA Isaac ROS

The NVIDIA Isaac ROS, a component of the NVIDIA Isaac Software Development Kit (SDK), provides powerful AI tools and computational capabilities for robotics. Its integration with existing ROS frameworks and strong support from developers makes it an optimal choice for enhancing tasks like image processing and real-time decision-making. NVIDIA Isaac ROS is specifically designed to leverage the advanced capabilities of NVIDIA Jetson hardware, maximizing the performance of computational tasks and efficiently utilizing the hardware resources available in the NVIDIA Jetson AGX Orin. [13].

D. Architecture

Our ROS2 system architecture, as shown in appendix B, primarily relies on publishers, subscribers, action servers, and service servers, with a behavior tree acting as the central client. The behavior tree is responsible for all decision-making processes within the system, delegating tasks to other modules as needed. It communicates with these modules via action servers and service servers, ensuring bidirectional communication.

Publishers and subscribers manage the flow of data, particularly for processing sensor signals used by system modules or for actuating thrusters. This architecture emphasizes modularity and scalability, making it straightforward to add new features to the system. Additionally, we have consciously avoided using flags to minimize system dependencies, which can otherwise lead to increased complexity and reduced manageability.

The diagram illustrates our system architecture, with the behavior tree serving as the central client. Positioned centrally, the behavior tree coordinates the various components of the system. On the outer rim, the system interacts with sensors and actuators through drivers and controllers. This configuration ensures efficient communication and control across the entire system.

E. Interaction Between Software Modules

The modules are implemented as ROS2 nodes using Python and C++. These nodes communicate across the ROS2 system utilizing both standard message types and custom message types that we have developed.

1) Behaviour Tree: The behavior tree functions as the system's task delegator and primary client. It handles all decision-making processes and delegates tasks accordingly. This client, written in C++, facilitates bidirectional communication with every module within the system. An example is shown in appendix C.

Additionally, the behavior tree serves as the mission planning and execution component, tracking the current state and the tasks to be accomplished using XML code. This XML code is compiled at runtime, allowing for the design, tweaking, and testing of different missions and behaviors without the need to recompile the C++ client.

The behavior tree is a powerful tool for state management, providing fallback tasks and handling increasingly complex logic. Unlike finite state machines, a behavior tree can manage multiple states, complex sequences, and decisions without increasing dependencies, making it a highly scalable solution.

2) Autopath Action Server: The autopath action server is our primary method for managing driving logic in the autonomous system. This module acts as an action client, enabling the behavior tree to send goals that consist of an array of coordinates. The module then utilizes a PD controller to sequentially navigate to each waypoint in the goal.

Due to its asynchronous nature, the Autopath action server allows the rest of the system to perform other operations concurrently. Throughout its task, the module provides feedback on the progress of the pathing and returns a status upon completion. At any point, the behavior tree can cancel this action if the vessel needs to address a different task or situation.

3) Keep Position Server: The keep position action server is responsible for holding the vessels locked state. Using three PD controller for controlling the vessels linear x and linear y positions and the vessels heading. The vessels current position and heading is given by the GPS Imu Fusion Publisher, and the error is calculated based on either a set parameter or the vessels start state.

4) Relative Service Server: The relative location service server is used to calculate the distance to a waypoint, and difference in angle between the headpoint and the vessels current angle. It is used to enable the PD controller for the auto path action server, and returns this information when requested.

5) Lidar Simplifier Publisher: The Lidar Simplifier Publisher utilizes clustering of Lidar data to simplify object detection for use in various logical operations within the system.

This functionality enables the system to avoid collisions by monitoring the distance to incoming objects in a 360-degree radius around the vessel. It is employed for a range of tasks, from avoiding collisions with incoming vessels to managing docking behavior.

6) Lidar Fusion Publisher: The Lidar Fusion Publisher calibrates and synchronizes the camera feed with Lidar data. It utilizes YOLO bounding boxes derived from Isaac ROS YOLOv8 to identify objects, determine their color IDs, and pinpoint their locations. This processed data is then published and stored in the register for future use.

7) GPS Imu Fusion Publisher: The GPS Imu Fusion Publisher is responsible for accurately tracking the vessel's position by integrating data from the GPS and IMU drivers. The GPS coordinates are converted from longitude and latitude to local coordinates, with the vessel's position set to (0,0) at system startup. This conversion simplifies development and troubleshooting by allowing the team to work with smaller, more manageable numbers.

This module is crucial for our navigation and control systems. By using a GPS with RTK (Real-Time Kinematic) fix, the vessel can track its position with up to 1 cm accuracy.

8) Manual/Auto MUX Publisher: The Auto/Manual MUX Publisher subscribes to two different ROS2 topics, receiving control signals from both the autonomous system and the RC controller. By using a switch on the RC controller, the MUX can switch between these signals, determining which one is sent to the thrusters. This capability allows for manual control of the vessel at any point, ensuring flexibility and safety during operations.

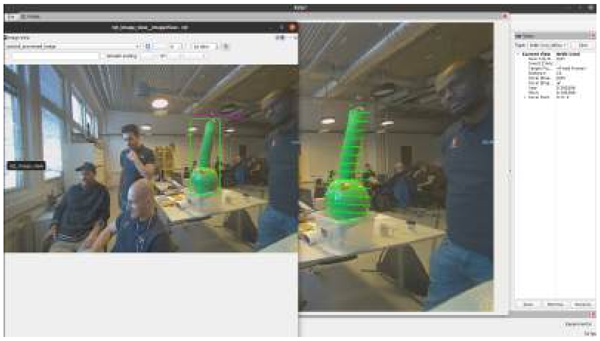


Fig. 7: LiDAR points overlaid on top of a bounding box.

V. TESTING

A. Vessel Testing

A 3D model of the hull was made using Solidworks to evaluate buoyancy and stability in regards to various loading conditions and to maximize its capacity. Following production of the hull it would be tested for its predicted properties in a small tank of water. In the tank it was loaded with weights that were placed in varying places in the hull to see how stability and waterline were affected.

Fig. 8: Pool testing

Next, the propulsion system was mounted, and the hull placed back in the tank to confirm that it remained watertight, and to test thrust capacity.

B. Hardware Testing

Each hardware component was individually tested for functionality and resilience under various conditions. Safety measures included adding fuses and waterproofing non-IP68-rated components, followed by outdoor tests in rainy and rough sea conditions. Long-term exposure tests ensured components could withstand the boat's operational demands, with waterproofing preventing electrical failures.

After verifying individual components, the entire hardware system was tested to validate combined functionality. Integrated testing involved all components, including those modified for waterproofing, under real-world scenarios to ensure seamless operation even in challenging conditions. This thorough approach highlights our commitment to delivering a resilient and high-performing ASV, ensuring reliability through fuses, waterproofing, and rigorous outdoor testing.

C. Software Testing

To accelerate development time, we have utilized digital twinning of the Triton in a Unity-based ROS2 simulator. This approach has enabled us to safely test and develop high-level controllers, propulsion systems, LiDAR and camera sensors, information flow, and mission handling. Each software module in our system's architecture was tested in this simulated environment, including the boat and buoys, as seen in Figure 9.

The use of a digital twin has allowed multiple team members to develop and test different parts of the system concurrently, significantly enhancing our development process. This innovative method has provided a risk-free environment to refine our designs and improve the overall reliability of the Triton ASV before physical deployment. Additionally, the simulator has facilitated better team collaboration and more efficient troubleshooting, ensuring a smoother and faster development cycle.

Fig. 9: Illustrates Digital Twin of the Triton in a Unity based ROS2 simulator.

D. *Integrated testing*

1) Detection Testing: Testing of the integrated system ensured seamless vessel operation, including the creation and deployment of buoy models. Extensive on-water tests validated the machine learning model's accuracy in detecting and interacting with buoys.

Fig. 10: Detecting buoys using different models in YOLO.

Various YOLOv8 models, from nano to x-large, were tested using a video from the boat's perspective to mimic real-world buoy detection at different distances. The models showed consistent detection capabilities, with larger models offering a slight increase in range but up to 3x longer inference times. The marginal range increase did not justify the higher computational demands [5].

To optimize performance, models were converted to TensorRT format, enhancing efficiency and responsiveness on our NVIDIA Jetson hardware. The nano and medium models were chosen for their balance of detection range and computational efficiency, fully utilizing the NVIDIA Jetson platform's capabilities [14].

VI. CONCLUSION

Integrating the ZED X camera and LiDAR into the Triton Autonomous Sea Drone required strategic planning to ensure effective communication and data synchronization with the Nvidia Jetson AGX Orin processing unit. Synchronizing data streams from the camera, LiDAR, and IMU/GNSS was essential for accurate detection and positioning. The `message_filters::sync::ApproximateTime` policy was used to match messages with different timestamps, ensuring temporal alignment for precise spatial mapping. Initial synchronization issues between the IMU/GNSS and other equipment were resolved to enhance overall system accuracy.

A significant challenge was ensuring real-time processing of high-volume data from both the ZED X camera and the LiDAR sensors. This was addressed by optimizing the data pipeline using NVIDIA Isaac ROS to leverage the computational power of the Nvidia Jetson AGX Orin. This platform is well-suited for handling complex algorithms required for real-time navigation and obstacle avoidance. Streamlining data processing and utilizing hardware acceleration features allowed the system to manage large data streams and maintain necessary performance levels for autonomous operations.

A. *Software and Hardware Coordination*

Coordinating hardware components and software systems, particularly with ROS2 Humble Hawksbill and NVIDIA Isaac ROS, required careful calibration. Using ROS2 wrappers for the different sensors streamlined the integration process, minimizing latency and maximizing data transfer reliability. Fine-tuning communication protocols ensured seamless operation of all components. Standardized ROS2 interfaces facilitated smoother interactions between the sensors and the processing unit, ultimately enhancing overall system performance.

ACKNOWLEDGMENT

We want to give a special thanks to all of our sponsors for helping us during this project. We appreciate your support, and we wouldn't be able to accomplish what we have without your interest in our project.

- Kongsberg Defence & Aerospace
- University of South-Eastern Norway
- Telia
- Sprout
- The Norwegian Mapping Authority
- Link Nordic
- Emcom
- Celerway
- Ocean Autonomy Cluster
- E3
- RS Noatun

REFERENCES

[1] B. Robotics, *T200 thruster*, Accessed: 2024-06-13, n.d. [Online]. Available: https://bluerobotics.com/store/thrusters/t100-t200-thrusters/t200-thruster-r2-rp/.

[2] *SKANBATT Bluetooth Lithium Batteri 12V 20AH 30A BMS*, en. [Online]. Available: https://skanbatt.no/product/757840/skanbatt-bluetooth-lithium-batteri-12v-20ah-30a-bms (visited on 06/14/2024).

[3] Skanbatt, *Datablad for skanbatt lith-blue1220b30*, Battery Datasheet, 2023. [Online]. Available: https://skanbatt.no/files/public/ea252ed0-7178-4a95-af23-8871f41c4e36/71e538e19c0d1fc213fc20c968fc85f8eff82ade/Datablad%20Skanbatt%20LITH-BLUE1220B30.pdf.

[4] Stereolabs, *ZED-X datasheet*, Online, Available: https://cdn2.stereolabs.com/assets/datasheets/zed-x-datasheet-march-2023.pdf, Mar. 2023. [Online]. Available: https://cdn2.stereolabs.com/assets/datasheets/zed-x-datasheet-march-2023.pdf.

[5] *Yolov8 documentation*, Online, Available: https://docs.ultralytics.com/, 2024. [Online]. Available: https://docs.ultralytics.com/.

[6] V. Lidar, *Puck datasheet*, Accessed: 2024-04-27, 2019. [Online]. Available: https://velodynelidar.com/wp-content/uploads/2019/12/63-9229_Rev-K_Puck-_Datasheet_Web.pdf.

[7] XSENS, *Mti 600-series datasheet*. [Online]. Available: https://www.xsens.com/hubfs/Downloads/Leaflets/MTi%20600-series%20Datasheet.pdf (visited on 06/13/2024).

[8] Kartverket, *Cpos dokumentasjon*, (sjekket 01-29-24). [Online]. Available: https://www.kartverket.no/til-lands/posisjon/cpos-dokumentasjon.

[9] N. S. Inc., *Gnss and geodesy concepts*, 2023. [Online]. Available: https://northsurveying.com/index.php/soporte/gnss-and-geodesy-concepts.

[10] Celerway. "Stratus 5g." (2022), [Online]. Available: https://www.celerway.com/products/stratus-5g (visited on 06/16/2024).

[11] Mouser and Nvidia, "NVIDIA Jetson AGX Orin Developer Kit — Datasheet," en, [Online]. Available: https://developer.nvidia.com/sites/default/files/akamai/Jetson_AGX_Orin_Developer_Kit_RG.pdf (visited on 06/16/2024).

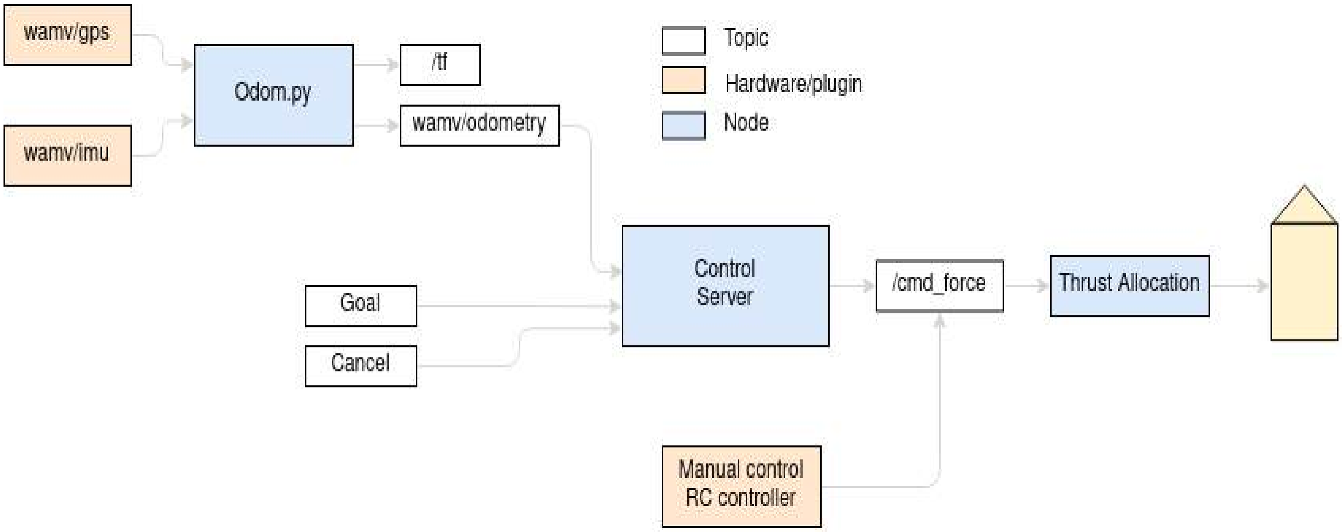
[12] Open Robotics, *ROS 2 Humble Hawksbill Documentation*, https://docs.ros.org/en/humble/index.html, Accessed: 2024-01-30, 2023.

[13] NVIDIA Corporation, *NVIDIA Isaac ROS Documentation*, https://nvidia-isaac-ros.github.io/, Accessed: 2024-06-20, 2023.

[14] *NVIDIA Deep Learning TensorRT Documentation*. [Online]. Available: https://docs.nvidia.com/deeplearning/tensorrt/ (visited on 05/15/2024).

VII. APPENDIX

A. Control diagram



B. Software diagram

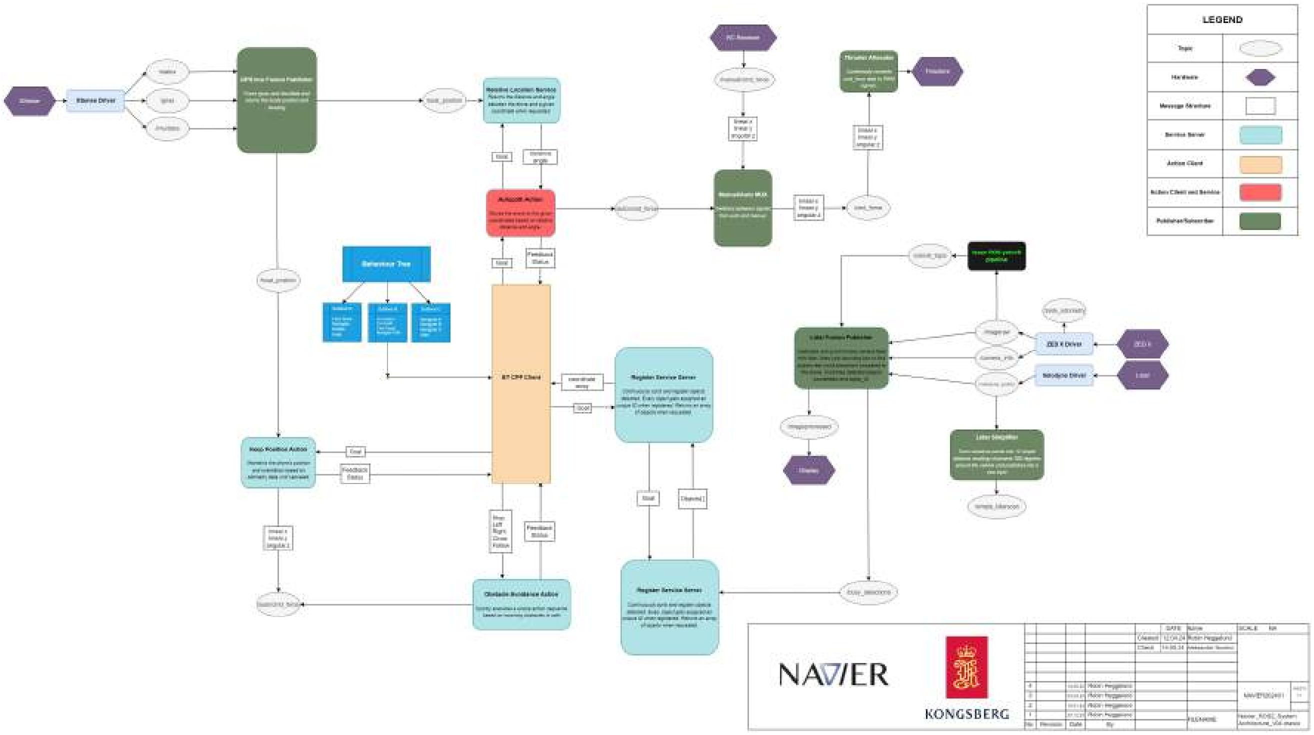


Fig. 11: Navier USN ROS2 system architecture.

C. Behaviour Tree

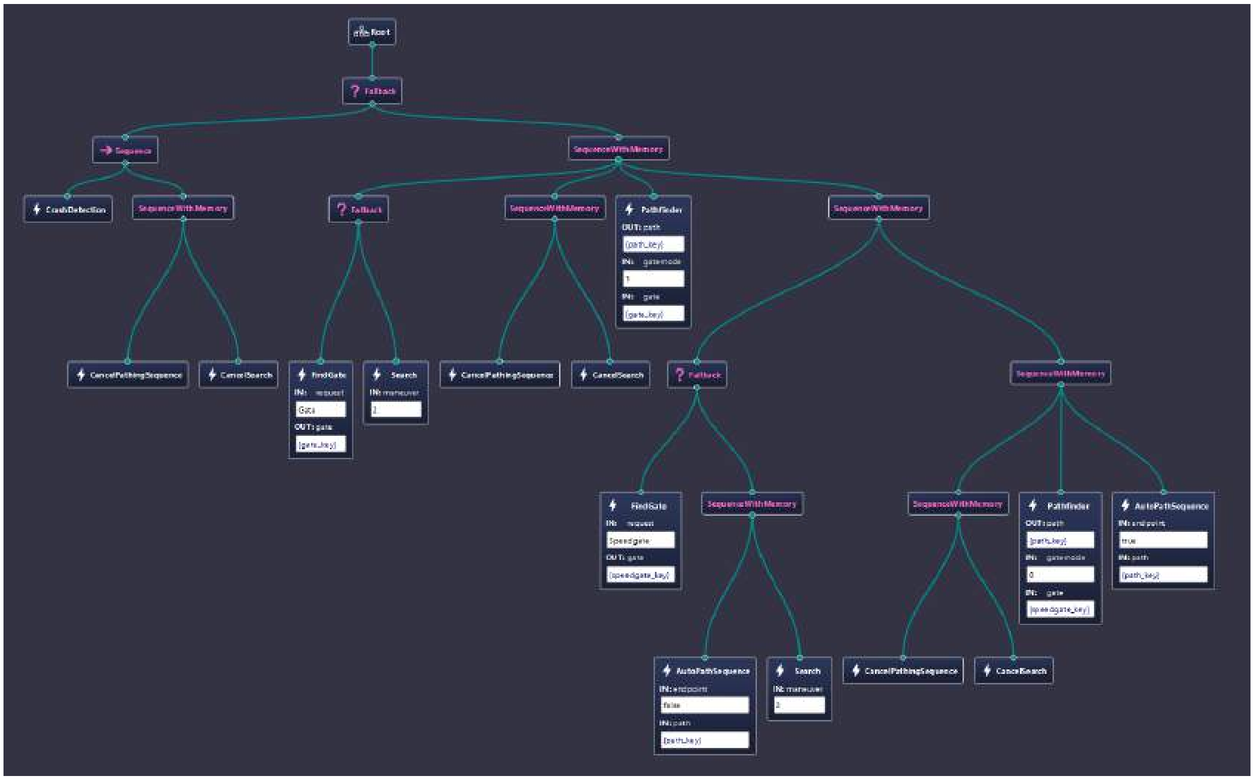


Fig. 12: The figure illustrates an example mission in the ROS2 Behaviour Tree.