

WetExplorer: Automating Wetland Greenhouse-Gas Surveys with an Autonomous Mobile Robot

Jose Vasquez and Xuping Zhang*

Abstract—Quantifying greenhouse-gases (GHG) in wetlands is critical for climate modeling and restoration assessment, yet manual sampling is labor-intensive, and time demanding. We present *WetExplorer*, an autonomous tracked robot that automates the full GHG-sampling workflow. The robot system integrates low-ground-pressure locomotion, centimeter-accurate lift placement, dual-RTK sensor fusion, obstacle avoidance planning, and deep-learning perception in a containerized ROS2 stack. Outdoor trials verified that the sensor-fusion stack maintains a mean localization error of 1.71 cm, the vision module estimates object pose with 7 mm translational and 3° rotational accuracy, while indoor trials demonstrated that the full motion-planning pipeline positions the sampling chamber within a global tolerance of 70 mm while avoiding obstacles, all without human intervention. By eliminating the manual bottleneck, *WetExplorer* enables high-frequency, multi-site GHG measurements and opens the door for dense, long-duration datasets in saturated wetland terrain.

I. INTRODUCTION

Wetlands occupy barely 1% of the Earth’s surface, but store about 20% of all carbon in the ecosystem [1]. When these systems are drained, restored or otherwise disturbed, they can emit sizable amounts of CO₂, CH₄, and N₂O [2]. Capturing such swings requires repeated and spatially extensive measurements. This data is crucial to understand carbon cycles and its absence can limit the evaluation of costly restoration efforts. However, researchers can only obtain a few plots before labor, safety, and access constraints become a problem. Closing the gap between the scientific need for frequent sampling and the practical limitations of manual fieldwork is essential for data-driven wetland management.

A. Field Conditions and Manual-Sampling Burden

Our target wetlands in Denmark consist of ankle-deep mud, soft peat, knee-high vegetation, and shallow ponds (Fig. 1a). The terrain is predominantly flat but with small vegetation clusters and peat irregularities, which typically introduce bumps of up to 10 cm in height. Carrying heavy measurement devices, such as two gas analyzers weighing 10 kg each and a 4.3 kg gas collection chamber, across such terrain every two hours is physically exhausting and considerably slows progress.

The current protocol (Fig. 1b–d) requires researchers to press a PVC collar into the ground (Fig. 1b), then seat the chamber on the collar and then attach the tubing while holding the portable analyzer (Fig. 1c), and remain for the five-minute measurement procedure before moving to



Fig. 1: Field sampling workflow in Danish wetlands. (a) Overview of the sampling area. (b) A PVC collar is pushed and fixed into the ground. (c) A chamber is seated on the collar for gas collection. (d) Two portable gas analyzers carried by the researchers complete the GHG readings.

the next sampling point (Fig. 1d). Repeating this sequence twelve times per day per collar limits most studies to only two or three plots and leaves large temporal gaps in the sampling analysis.

Automating this workflow with a robotic platform would eliminate the human bottleneck, enabling continuous, high-resolution GHG measurements across dozens of wetlands.

B. Our Contribution

We developed *WetExplorer*, the first robotic system that fully automates chamber-based greenhouse-gas sampling in wetland terrain. Specifically, we:

- (i) integrate a low-ground-pressure tracked chassis and design a lift mechanism capable of centimeter-level chamber placement;
- (ii) implement a multisensor state-estimation stack that fuses dual-antenna RTK-GPS, IMU, and encoders for robust navigation;
- (iii) add a mission planning layer that combines a Hybrid-A* planner with a Pure-Pursuit controller for global motion and a precise position controller for tool alignment; and
- (iv) deliver a containerized control architecture that supports unattended operation and data logging.

Corresponding author, Department of Mechanical and Production Engineering, Aarhus University, Aarhus, Denmark.

Email: xuzh@mpe.au.dk

Section II covers related sampling and mobile-robot systems; Section III details the mechanical and electronic design; Section IV explains navigation, perception, and mission software; Section V reports localization, computer vision and chamber-placement accuracy as well as a general evaluation of the system with a full-mission field trial; and Section VI discusses limitations and future work toward greater terrain adaptability and system robustness.

II. RELATED WORK

A. Technologies for Outdoor Autonomous Vehicles

Outdoor robots can localize themselves with many techniques: GNSS/RTK for global fixes, UWB or other radio beacons for local ranging, LiDAR- or vision-based SLAM for map-centric accuracy, and odometry or RFID/QR markers for short-range updates. A recent review shows that their performance spans 0.1m for RTK-GNSS, centimeters for LiDAR-vision fusion (at high compute cost), and decimeters or worse for stand-alone odometry or low-cost radio, with each method carrying its own cost and environmental limits [3]. Because no single sensor is fault-proof, most field robots combine complementary sources, for example, RTK-GPS, IMU, LiDARs, radars, depth cameras and wheel encoders to achieve the sub-decimeter accuracy reported on agricultural and wetland platforms.

Typical outdoor navigation couple a cost-aware global planner with a reactive path follower. Global planners such as grid-based A* and Hybrid-A* give quick, kinematically feasible routes on structured maps; sampling methods like RRT* and BIT* explore cluttered terrain more freely; and optimization or learning-based planners, including CHOMP and value-iteration networks, refine trajectories when vehicle dynamics or ride comfort matter [4]. Once the path is set, geometric followers like Pure Pursuit or Stanley provide lightweight steering, whereas model-predictive and adaptive sliding-mode controllers add dynamic awareness for tighter tracking [4].

Despite their maturity, these algorithms have not yet been validated under the slip, vegetation occlusion, and micro-topography unique to the Danish wetlands.

B. Autonomous Robots for Wetlands

Tracked or amphibious ground robots have been deployed for vegetation mapping, sediment coring, and pollution surveillance in marshes and rice paddies [5], [6]. Wide tracks, paddle wheels, or oversized tires give these platforms low ground pressure, letting them traverse soft, saturated soils without sinking. Most payload modules, however, carry cameras or water samplers rather than mechanisms capable of making direct soil contact.

Existing aquatic robots highlight this limitation: the 5m Wivenhoe ASV carries an open-path optical-methane detector that “sniffs” ebullition plumes from above the water surface, while the lighter, multi-robot Inference fleet lowers a small gas-sampling chamber into the water for stationary flux checks [7]. Both automate methane sensing on reservoirs, but neither can press a sealed chamber against saturated soil,

leaving ground-level gas measurements unsolved for mobile platforms.

Existing systems either automate gas sampling at a single fixed point or mobilize sensors without soil contact. No prior work combines low-ground-pressure locomotion, compliant sealed placement on the target, and fully autonomous multi-plot scheduling in wetlands, the integrated capability targeted by *WetExplorer*.

III. SYSTEM OVERVIEW

Fig. 2a highlights the tracked chassis capable of traversing through the rough terrain, while a screw-ball lift precisely lowers the 4.3 kg sampling chamber onto each collar ring. Navigation relies on a dual-antenna RTK-GPS and two depth cameras: a forward-facing, broad-view sensor for terrain perception and a narrower target-view sensor for pinpoint ring localization. All sensor streams are handled by a computing module that houses the main processing and communication electronics.

Fig. 2b showcases the computing stack. An Intel NUC serves as the central computer, interfacing with most sensors and the chassis controller, while an STM32-based microcontroller drives the lift actuator. An IMU and the RTK-GPS module are fused for state estimation. A network router links the main computation modules, including the central computer, an AI-accelerated perception computer, and the gas sensor computers, completing the modular mechatronic architecture.

A. Tracked Chassis

Wetland soils are very soft and frequently inundated, which rules out wheeled bases and most legged platforms. Tracked locomotion spreads the vehicle mass over a large contact area, resulting in a low average ground pressure, well below the one produced by researchers [8]. Among the commercial platforms surveyed, the Robodyne MAXII offered the most favorable weight-to-payload ratio (60 kg weight; 90 kg payload) and a ground clearance of 18.4 cm, making it suitable for the targeted site.

Each track is driven by a brushed DC motor coupled to a 1:30 spur gearbox. Closed-loop velocity control is managed by a Roboteq MDC2230 controller, which integrates a quadrature encoder and provides a serial interface for commanding and monitoring the chassis from an external computer. A 24 V, 35 Ah Li-ion battery powers the drive system for approximately 4.5 h of stop-and-go operation.

B. State-Estimation Sensors

The navigation stack fuses an inertial measurement unit (IMU) with a dual-antenna RTK-GPS receiver for robust, centimeter-grade state estimation, as well as the quadrature encoders from the motors.

The chosen IMU corresponds to a PhidgetSpatial Precision 3/3/3 because it combines a low $0.0015^\circ/\text{s}$ bias gyro with a 250 Hz data rate and on-board AHRS computation, thereby offloading quaternion math from the main CPU [9].

In addition, the absolute position provided by the RTK-GPS is essential to revisit ring collars after multi-hour gaps.

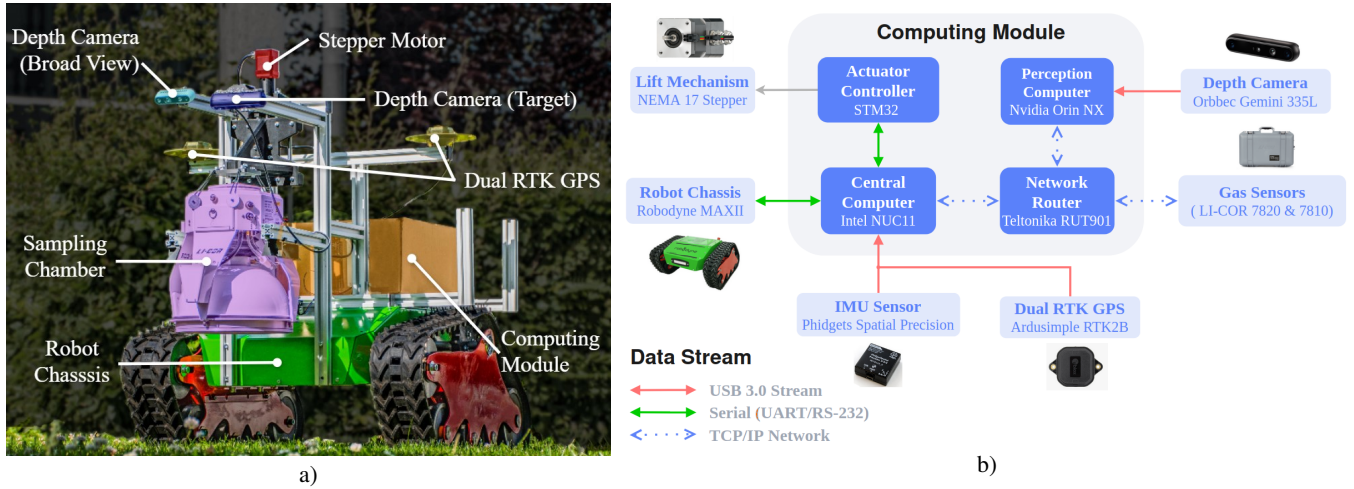


Fig. 2: WetExplorer overview. a) Main hardware components. b) Mechatronics diagram of the computing module.

The dual-antenna ArduSimpleRTK2BHeading Kit provides sub-degree heading and centimeter-grade positioning without magnetometers [10], which can saturate in iron-rich peat soils or be disturbed by the robot’s high-current actuators. The module uses 2.4 GHz radios connected to an in-house base station, delivering RTK corrections without relying on NTRIP services and maintaining reliable operation in remote areas.

C. Perception Sensor

Stereo vision enables outdoor depth perception without the range-aliasing limitations of time-of-flight sensors under sunlight. The Orbbec Gemini 335L (IP65) was selected after the Intel RealSense D435 showed good performance but lacked waterproofing. Offering similar depth quality in a sealed housing, the Orbbec meets the requirements for 3D sensing in wetlands.

D. Compute Stack

An Intel NUC11 Pro (Core i5-1135G7, 16GB RAM) serves as the central computer. All navigation, localization, and mission state machines run on this host at around 30Hz. A separate NVIDIA Jetson Orin NX 8 GB executes learned perception tasks (e.g., YOLOv11 ring detection, point-cloud registration) at up to 117 TOPS, freeing the central computer for deterministic control loops and debugging.

E. Networking and Telemetry

Field sites lack Wi-Fi, so a Teltonika RUT901 industrial LTE router provides 4G connectivity and a local network that interconnects the NUC, Jetson and Li-Cor gas analyzers. The router runs a Wi-Fi network that allows users to connect an external device to display the data of the robot computers.

F. Lift Mechanism

The 4.3 kg Li-Cor chamber is lowered by a self-locking screw-ball actuator (Fig. 3a); consequently, power is drawn mainly while the lift is moving. The stepper motor runs open-loop and re-homes at the beginning of each cycle via a limit switch located at the top of travel.

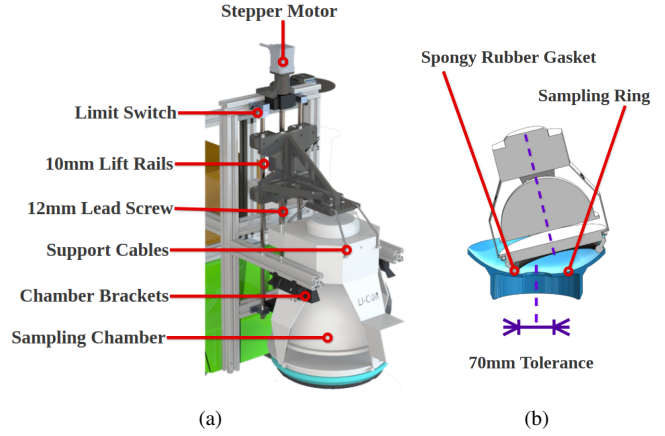


Fig. 3: Chamber Manipulator Design. a) Lift mechanism diagram. b) Section View of the Chamber - Ring coupling.

When the robot is in motion, two locating chamber brackets lock the chamber in place to prevent swinging. Once the lift is lowered, the brackets are no longer holding the chamber and two cables act as a passive gimbal, providing a few degrees of rotational and lateral compliance as the chamber settles onto the collar.

The mechanism is driven by a Drylin® D7 stepper-motor driver and we developed a STM32-based controller, which converts distance commands into step pulses for the driver and manages calibration and limit-switch safety.

The collar’s spherical profile ensures a proper seal regardless of small angular misalignments, mechanically tolerating up to 70 mm of radial placement error while compensating for minor navigation or computer vision offsets (Fig. 3b). A foam rubber gasket mounted on the chamber compresses under its own weight to form an airtight contact. Consequently, any successful mechanical coupling between the chamber and the collar implies that the placement error remained within this 70 mm tolerance, which serves as the acceptance metric for evaluating the system’s accuracy.

IV. AUTONOMOUS NAVIGATION AND MISSION EXECUTION

Fig. 4 illustrates the end-to-end software workflow of the autonomous sampling mission. The process begins with Mission Setup (Fig. 4a), where a map is generated and the required sampling points are defined. Precise Positioning (Fig. 4b) establishes a closed state-estimation loop to maintain accurate localization. During Navigation to Staging Pose (Fig. 4c), the robot autonomously travels to the vicinity of each collar. Precise Targeting (Fig. 4d) refines the detected ring pose and guides the sampling chamber into alignment. Finally, Sample Collection (Fig. 4e) lowers the chamber and records gas measurement data. The following section details the algorithms and implementation strategies that enable this fully unattended operation.

A. Kinematic Model and Odometry

The tracked base is modeled as a differential-drive platform with equivalent sprocket radius r and track separation b . The inverse-kinematics equations convert the commanded linear and angular velocities $(V_x, \dot{\theta})$ into left and right sprocket angular speeds (ω_L, ω_R) :

$$\omega_L = \frac{V_x - \frac{1}{2}\dot{\theta}b}{r}, \quad \omega_R = \frac{V_x + \frac{1}{2}\dot{\theta}b}{r}. \quad (1)$$

On the other hand, the forward kinematics equations convert the measured motor speeds (reported in r/min) to rad/s and reconstruct the robot's twist:

$$V_x = \frac{v_L + v_R}{2}, \quad \dot{\theta} = \frac{v_R - v_L}{b}. \quad (2)$$

where v_L and v_R are the left and right linear track speeds, respectively.

The resulting odometry stream is fused with inertial and RTK-GPS data, as described next.

B. Localization through Sensor Fusion

Using a single sensor is insufficient in wetlands: IMU integration drifts, wheel odometry slips, and low-rate GPS misses fast maneuvers. We therefore adopt a loosely coupled Unscented Kalman Filter (UKF) to fuse data from all sensors, employing the widely used and robust *robot_localization* implementation in ROS2 [11].

Velocity Sensor Fusion: combines IMU accelerations and angular rates with odometry, using manufacturer-based IMU covariances and adaptive odometry noise that increases with speed to account for slip [12].

Position Sensor Fusion: augments this state with dual-antenna RTK-GPS fixes at 5 Hz. The GPS position ($\sigma \approx 1.5\text{cm}$) dominates the update, while velocity constraints smooth the trajectory and eliminate jumps in raw GPS data, producing a reliable 20 Hz localization stream for navigation.

C. Navigation to the Staging Pose

Mission way-points (ring collars) are allocated in Fig. 4c. Each transfer employs:

Global planner: Smac Hybrid-A* is selected because it produces kinematically feasible paths for non-holonomic vehicles while still replanning fast enough for receding-horizon use on embedded CPUs [13]. The solver was configured for our tracked base by imposing a minimum turn radius of 0.20 m, a cost penalty of 3.0, a reverse penalty of 2.0, a non-straight penalty of 1.0, and a change penalty of 5.0. These parameters penalize zig-zag motions, yielding smoother trajectories that are easier for the controller to track.

Wetland “no-go” areas (vegetation, water ponds, neighboring collars) are converted into terrain masks and inflated on the custom costmap; the higher cell costs, together with the penalties above, steer the solution away from surfaces that could lead to collisions or sink-ins.

Local controller: Regulated Pure Pursuit (RPP) is adopted instead of the standard pure-pursuit tracker because it keeps the same geometric steering but adds built-in speed regulation for tight corners and crowding obstacles, a decisive advantage for our heavy, slip-prone tracked base [14]. The controller follows the Hybrid-A* spline while continuously lowering the commanded speed when curvature exceeds a set radius or obstacles enter a safety band, two effects that matter on the cluttered service routes of our test site.

The principal controller parameters, identified through simulation, are a *lookahead time* of 1.5 s, a *desired linear speed* of 0.20 m/s, and a *rotate-to-heading* protocol that engages whenever the robot's orientation deviates by at least 45° from the path. At nominal cruise speed these settings yield a *lookahead distance* of roughly 0.6 m, which is equal to the 0.6 m wheelbase.

D. Ring 6-DoF Pose Estimation

Our ring-pose estimation pipeline is carried out by a three-stage pipeline that remains light enough for online execution on the Jetson Orin NX. First, a fine-tuned *YOLOv11-n-seg* network detects the ring at 15 fps and returns a binary mask, offering a good speed-accuracy trade-off on edge GPUs [15]. The mask is then applied to the depth image so that only valid 3D points are retained; then a centroid computation supplies a fallback pose whenever the subsequent alignment step is unsuccessful. Finally, the cropped point-cloud is aligned to a CAD template that holds the reference of the object with Predator, a learning-based point-cloud matcher tolerant of low surface overlap [16]. A single ICP refinement reduces the error and increases the final overlap, delivering a camera-to-ring transform that easily satisfies the robot's positioning tolerance [17].

The target object is placed in an open area where it is rarely occluded by surrounding elements, and its orientation remains relatively stable under different circumstances. Varying illumination conditions, including low light, direct sunlight, and shadowed areas, were considered during dataset preparation. The object detection module was trained on 100 positive and 100 negative RGB samples, which were augmented using brightness variation, Gaussian blur, and noise

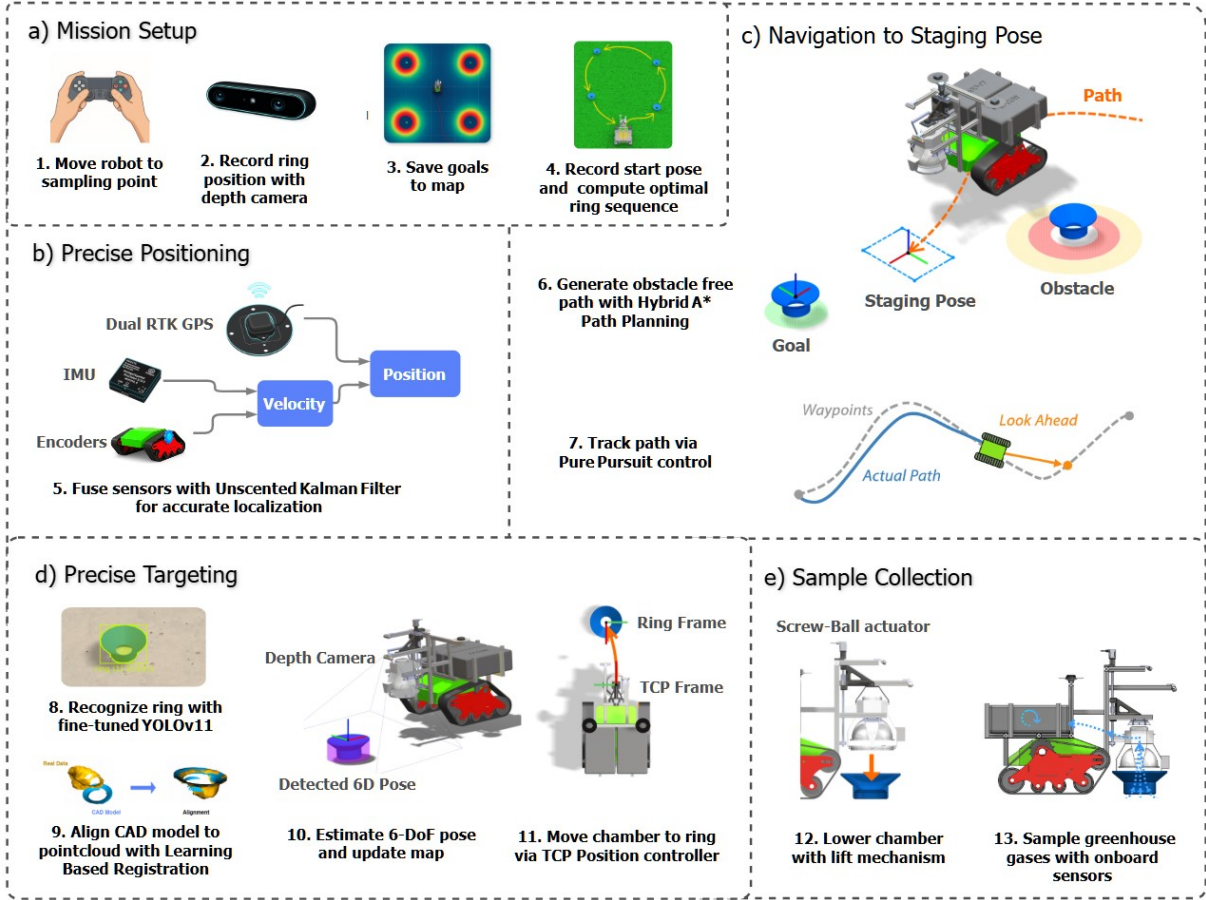


Fig. 4: Autonomous sampling workflow. a) Mission setup: the operator drives the robot to each ring location and stores the GPS goals. b) Precise positioning: dual-antenna RTK-GPS, IMU, and encoders are fused by two UKFs for sub-centimeter level localization. c) Navigation to staging pose: Hybrid-A* and Pure-Pursuit guides the robot next to the collar while avoiding obstacles. d) Precise targeting: YOLOv11 and point-cloud registration estimate the 6-DoF ring pose. e) Sample collection: the lift lowers, seals, and the analyzer records CO_2 , CH_4 , and N_2O .

injection, resulting in a dataset of 600 images that combined indoor and outdoor scenes under both sunny and cloudy conditions. For the point-cloud alignment stage, 50 primarily outdoor point-cloud pairs were collected and augmented to 250 pairs using random spatial transformations, added noise, and simulated partial observations to improve robustness in real-world deployments.

E. Chamber Alignment

The tool center point (TCP) is guided towards the ring by a planar PI controller (step 11, Fig. 4c). The control loop is inspired by a classic mobile robot feedback control while leaving the final orientation unconstrained [18]. Let $\mathbf{p}_{\text{ring}}$ and $\mathbf{p}_{\text{TCP}}$ be the positions of the ring and the TCP in the map frame. The in-plane error vector is

$$\Delta \mathbf{p} = (\Delta x, \Delta y)^T = \mathbf{p}_{\text{ring}} - \mathbf{p}_{\text{TCP}}.$$

a) Error signals:

$$e_p = \sqrt{\Delta x^2 + \Delta y^2}, \quad (3)$$

$$e_\alpha = \text{atan2}(\Delta y, \Delta x) - \psi, \quad (4)$$

where ψ is the current heading angle of the robot.

b) Control law:

$$\mathbf{v} = K_{P_p} e_p + K_{I_p} \int_0^t e_p(\tau) d\tau, \quad (5)$$

$$\omega = K_{P_\alpha} e_\alpha + K_{I_\alpha} \int_0^t e_\alpha(\tau) d\tau, \quad (6)$$

with $\mathbf{v}$ the linear and ω the angular velocity commands. These speeds are finally transformed from the TCP frame into the robot base frame before being forwarded to the low-level motion controller.

c) *Tuning*: The proportional and integral gains $(K_{P_p}, K_{I_p}, K_{P_\alpha}, K_{I_\alpha})$ were first tuned in the Gazebo simulation and then adjusted on the real robot to compensate for inaccuracies of the models.

F. Digital-Twin Simulation

All navigation software was first validated in Gazebo Harmonic. A xacro-based robot model includes mass/inertia for each link; sensor plug-ins replicate IMU drift, GPS precision, and depth-cameras, enabling hardware-in-the-loop tests before field deployment. The tracks are simulated using a kinematic approach that computes the ground forces

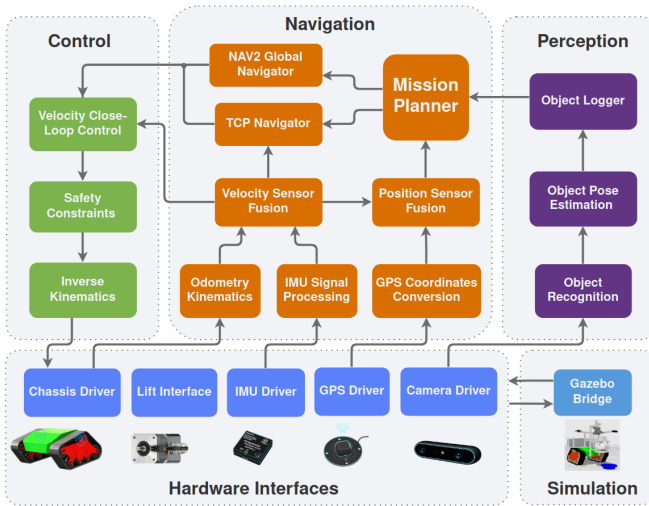


Fig. 5: ROS2 Software Architecture

exerted by the track surfaces to achieve the desired velocity. Instead of simulating the tracks' physical movement along pulleys, this method applies virtual forces that push the robot. This approach is computationally efficient and delivers accurate results, making it the most suitable implementation in Gazebo for track-based robots [19].

G. Software Implementation

The robot's software stack is built on ROS2 for its modular architecture, visualization tools and sensor drivers available. As illustrated in Fig. 5, the nodes are divided into four logical layers: a hardware-interface layer that handles the motor driver, IMU, and RTK receiver; a control layer implementing inverse and forward kinematics together with PID velocity loops; a navigation layer responsible for sensor fusion, cost-map planning, and path planning; and a perception layer that combines YOLOv11 object detection with point-cloud alignment. All workflows are packaged as Docker images, enabling one-command deployment and reproducible builds.

The complete source code is available as an open-source repository at [20].

V. EXPERIMENTAL VALIDATION

A. Localization Benchmark

The localization tests were conducted outdoors under good GPS coverage, similar to the open-field conditions expected at the deployment sites. The robot operated on a grassy area following a closed-loop trajectory of approximately 7 m at a constant speed of 0.5 m s^{-1} . Four pose estimates were logged simultaneously (Fig. 6). The resulting 2D trajectories, shown in the first graph, provide an overview of the motion smoothness and the performance differences among the methods. To complement this, Table I compares the 3D position estimates against the ground truth and summarizes the mean errors for each method.

Centimeter-level *RTK-GPS* consists in the ground truth, revealing the magnitude and nature of the errors in the other localization outputs. The first estimate, generated by pure

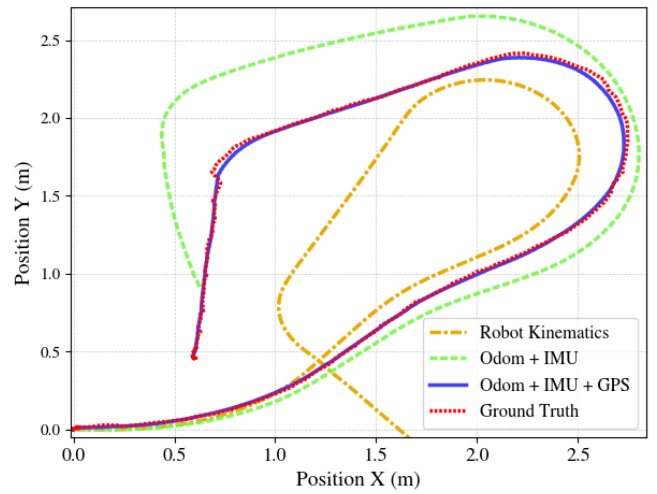


Fig. 6: Planar (X-Y) trajectories recorded during the localization benchmark. Yellow: wheel-encoder odometry that accumulates drift over time; blue: velocity estimate obtained by fusing IMU and encoder data; green: position estimate produced by integrating the fused velocity and GPS fixes; red: high-accuracy ground truth from the RTK-GPS receiver.

forward kinematics (yellow trace), integrates only wheel-encoder ticks and therefore exhibits cumulative drift caused by the uncertainty in the differential-drive model's angular-speed estimation. The second estimate, the *Velocity UKF* (green trace), fuses encoder-derived wheel speeds with the 250 Hz IMU using a velocity-space unscented Kalman filter. Although this approach effectively captures yaw rates and mitigates short-term slip, it cannot compensate for lateral drift during turns, resulting in long-term trajectory deviation. Finally, the *Position UKF* (blue trace) fuses RTK-GPS, IMU, and odometry to produce the navigation solution used in the field. This estimate closely follows the ground truth while maintaining the smooth motion of the Velocity UKF, effectively eliminating the jumps inherent in raw GPS data.

The experiment was repeated across three outdoor trajectories, and the resulting errors are summarized in Table I. Forward kinematics and velocity-space fusion methods accumulated larger errors over time, yet the results clearly show that the sensor fusion stage improved both heading and position accuracy, reducing errors from 0.5 m to 0.17 m and from 8.3° to 2.6° . The final fused estimate achieved an average 3D position error of 1.71 cm and a heading error of 0.18° . These values fall within the precision reported by the RTK-GPS (2.36 cm, 0.94°), confirming that the fusion model matches GPS accuracy while providing smoother and more consistent motion estimates. This shows that position is effectively estimated and that the navigation stack benefits from fast and accurate localization.

B. Computer Vision Validation

Validation focused on the registration stage, as it most directly determines pose accuracy. To thoroughly assess robustness, the experiments were designed to include challenging scenarios, such as the target object being partially

TABLE I: Localization Accuracy Across Sensor Fusion Levels

Method	Pos. Err. (m)	Std. (m)	Head. Err. (°)	Std. (°)
<i>Ground Truth</i>	-	0.0236	-	0.954
<i>Kinematics</i>	0.509	0.403	8.27	6.12
<i>Odom + IMU</i>	0.171	0.113	2.61	2.59
<i>Odom + IMU + GPS</i>	0.017	0.011	0.19	0.54

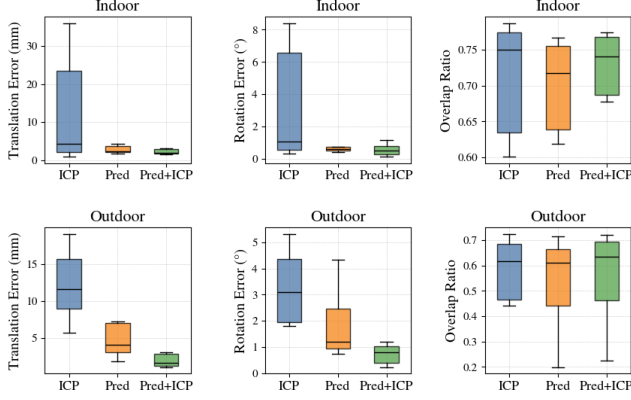


Fig. 7: Box plots showing the evaluation of the Object Localization in the indoor and outdoor testing sites.

occluded by the gas chamber or cropped by the camera’s field-of-view limits. Multiple observations of the object were collected at various positions within the camera’s view under both indoor and outdoor lighting conditions.

Each algorithm was evaluated on ten indoor and ten outdoor setups, executed ten times per setup. Fig. 7 compares ICP, Overlap Predator, and Predator + ICP, showing that Predator significantly reduces variability in translation and rotation, while the final ICP refinement further improves precision and overlap.

The mean values corresponding to these plots are summarized in Table II. Under indoor conditions, standard ICP achieved an average translational error of 8.16 mm, whereas the proposed method reduced it to 1.53 mm, also yielding a higher overlap ratio. Outdoor experiments proved more challenging due to lower point-cloud quality; however, the proposed approach maintained robust performance, achieving a mean positional error of 7.19 mm and an orientation error of 3.17°. It effectively doubled the accuracy of standard ICP and achieved a 100% success rate with at least 0.2 overlap, compared to 90% for ICP.

These results confirm that the proposed method provides sub-centimeter object-localization accuracy and remains reliable under outdoor and partially occluded conditions, meeting the precision requirements for accurate ring positioning on the robot.

C. Demonstration: Multi-Ring Mission Indoors

The navigation algorithms were tested in two sites: an indoor environment for algorithm tuning and an outdoor area for system validation under realistic conditions (Fig. 8). Navigation was divided into two stages: (i) reaching the staging

TABLE II: Registration Performance Across Environments

Env.	Method	Pos. (mm)	Rot. (°)	Overlap	Succ. (%)
Indoor	ICP	8.16	4.27	0.69	90
	Predator	2.69	0.98	0.70	100
	Pred + ICP	1.53	0.60	0.73	100
Outdoor	ICP	16.33	6.32	0.48	90
	Predator	9.08	3.04	0.47	90
	Pred + ICP	7.19	3.17	0.49	100

pose and (ii) accurately placing the sampling chamber on the ring. In total, 12 indoor and 6 outdoor missions were performed across a 5m × 5m grid.

The lack of GPS of indoors led to localization drift and reduced obstacle-avoidance reliability (91.7%). The robot occasionally collided with obstacles due to IMU heading drift and kinematic model simplifications that assume ideal rotations. Mapping errors also caused staging poses to deviate from the target rings. A recovery routine using computer-vision feedback was added, allowing the robot to correct its pose and ensure reliable TCP placement. The average time to reach the staging pose was 14.2 s (Table III), indicating efficient path planning. Since sampling missions occur roughly every two hours, time performance was not a limiting factor.

Outdoor tests were conducted on grass terrain at the university. Although less demanding than wetlands, they revealed several design challenges. With stable GPS reception, localization improved, but the robot initially struggled with slopes and chamber placement; the hanging chamber changes its final position according to the orientation, affecting accuracy. Localization was therefore extended to 3D, and the TCP placement algorithm was modified to predict contact based on gravity. Slippage and traction issues were mitigated by adding a velocity control loop to enforce motion commands. After these adjustments, the system achieved consistent obstacle avoidance and goal-reaching performance (Table III), with slightly longer travel times due to rougher terrain.

TCP placement used only odometry and IMU data, as GPS noise caused small jumps. The short 45 cm distance between staging pose and target minimized drift. Indoor tests achieved 5.5 mm placement accuracy and outdoor trials 12.7 mm, both well within the 70 mm tolerance required for a sealed coupling. The higher outdoor variation reflected terrain irregularities.

These experiments validated the integration of localization, perception, and navigation, showing that missions can be executed autonomously and repeatedly in realistic environments. The results also establish a solid baseline for evaluating and comparing future navigation and perception methods under similar terrain conditions. Remaining challenges such as slippage, soft ground, standing water, and reflections will be addressed in future iterations to achieve fully autonomous operation in natural wetlands.

TABLE III: Navigation and Placement Validation Targets

Metric	Indoor	Outdoor
Obstacle Avoidance Success	91.7%	100%
Average Time to Staging Pose	14.2s	18.6s
TCP Placement Accuracy	5.51 mm	12.67 mm
TCP Repeatability σ	2.40 mm	4.11 mm



Fig. 8: Multi-ring demonstration testing zones. a) Indoor Testing Area. b) Outdoor Testing Area.

VI. CONCLUSION AND FUTURE WORK

Conclusion—This paper has presented the development and testing of *WetExplorer*, a fully integrated ROS 2 robotic platform that unifies perception, navigation, and control. The design accommodates the specialized sensors required for wetland gas sampling, delivers robust traversability, and positions the collection chamber within a 70 mm tolerance. Outdoor experiments confirmed that the sensor-fusion stack achieves centimeter-level localization accuracy with smoothly updated estimates at 20 Hz. Indoor trials validated the vision pipeline, yielding an average positional error of 7 mm and an orientation error of 3.2° while operating in real time. Navigation mission indoor and outdoor testing demonstrated consistent reliability: the system planned collision-free paths and placed the chamber on the target rings well within the 70 mm tolerance, effectively transforming a labor-intensive manual protocol into a single-command robotic operation.

Future Work—We will extend the system’s robustness in harsher, wetland-like conditions by (i) mechanically hardening the chassis and lift to tolerate deeper inundation and floating debris; (ii) adapting the control architecture to the specific terrain characteristics of the field site; (iii) augmenting the software stack with a terrain-traversability analysis to detect sinkholes and obstructing vegetation; and (iv) validating long-duration missions across larger, heterogeneous peatland sites. These upgrades will advance *WetExplorer* from controlled test plots to fully autonomous, high-resolution greenhouse-gas monitoring in real-world wetlands.

ACKNOWLEDGMENT

The authors would like to thank Dr. Shubiao Wu and his team for their assistance on wetland sampling, and the Carlsberg Foundation for the financial support that made this project possible.

REFERENCES

- [1] R. J. M. Temmink, L. P. M. Lamers, C. Angelini *et al.*, “Recovering wetland biogeomorphic feedbacks to restore the world’s biotic carbon hotspots,” *Science*, vol. 376, no. 6593, p. eabn1479, 2022.
- [2] A. M. R. Petrescu, A. Lohila, J.-P. Tuovinen, D. D. Baldocchi, A. R. Desai, N. T. Roulet, T. Vesala, A. J. Dolman, W. C. Oechel, and B. Marcolla, “The uncertain climate footprint of wetlands under human pressure,” *Proceedings of the National Academy of Sciences*, vol. 112, no. 15, pp. 4594–4599, 2015.
- [3] J. Semborski and A. Idzkowski, “A review on positioning techniques of mobile robots,” *Robotic Systems and Applications*, vol. 4, no. 1, pp. 30–43, 2024.
- [4] N. Wang, X. Li, K. Zhang, J. Wang, and D. Xie, “A survey on path planning for autonomous ground vehicles in unstructured environments,” *Machines*, vol. 12, no. 1, p. 31, 2024.
- [5] I. AI for Good, “From research to rescue: Amphibious robots transforming environmental monitoring,” <https://aiforgood.itu.int/from-research-to-rescue-amphibious-robots>, 2024, accessed 2025-07-21.
- [6] A. Cohen and D. Zarrouk, “Design, analysis and experiments of a high-speed water hovering amphibious robot: Amphistar,” *IEEE Access*, pp. 1–1, 2023.
- [7] M. Dunbabin and A. Grinham, “Quantifying spatiotemporal greenhouse-gas emissions using autonomous surface vehicles,” *Journal of Field Robotics*, vol. 34, 2016.
- [8] L. Bruzzone and G. Quaglia, “Locomotion systems for ground mobile robots in unstructured environments,” *Mechanical Sciences*, vol. 3, pp. 49–62, Jul. 2012.
- [9] P. Inc. (2024) Phidgetspatial precision 3/3/3. Accessed 2025-07-31. [Online]. Available: <https://www.phidgets.com/?prodid=1205>
- [10] ArduSimple. (2024) Arduimplertk2b heading basic starter kit ip67. Accessed 2025-07-31. [Online]. Available: <https://www.ardusimple.com/product/simplertk2b-heading-basic-starter-kit-ip67/>
- [11] T. Moore and D. Stouch, “A generalized extended kalman filter implementation for the robot operating system,” in *Advances in Intelligent Systems and Computing*, 2016, vol. 302, pp. 335–348.
- [12] N. Strawa, D. I. Ignatyev, A. C. Zolotas, and A. Tsourdos, “On-line learning and updating unmanned tracked vehicle dynamics,” *Electronics*, vol. 10, no. 2, 2021. [Online]. Available: <https://www.mdpi.com/2079-9292/10/2/187>
- [13] S. Macenski, M. Booker, and J. Wallace, “Open-source, cost-aware kinematically feasible planning for mobile and surface robotics,” *arXiv preprint*, 2024. [Online]. Available: <https://arxiv.org/abs/2401.13078>
- [14] S. Macenski, S. Singh, F. Martín, and J. Ginés, “Regulated pure pursuit for robot path tracking,” *Autonomous Robots*, vol. 47, pp. 685–694, 2023. [Online]. Available: <https://arxiv.org/abs/2305.20026>
- [15] R. Khanam and M. Hussain, “Yolov11: An overview of the key architectural enhancements,” *arXiv preprint*, 2024. [Online]. Available: <https://arxiv.org/abs/2410.17725>
- [16] S. Huang, Z. Gojcic, M. Usvyatsov, A. Wieser, and K. Schindler, “Predator: Registration of 3d point clouds with low overlap,” *arXiv preprint*, 2021. [Online]. Available: <https://arxiv.org/abs/2011.13005>
- [17] L. Li, X. Yang, R. Wang *et al.*, “Automatic robot hand-eye calibration enabled by learning-based 3d vision,” *Journal of Intelligent & Robotic Systems*, vol. 110, p. 130, 2024.
- [18] R. Siegwart and I. Nourbakhsh, *Introduction to Autonomous Mobile Robots*, ser. Intelligent Robotics and Autonomous Agents. Cambridge, MA: MIT Press, 2004, “Mobile Robot Kinematics,” pp. 83–88.
- [19] M. Pecka, K. Zimmermann, and T. Svoboda, “Fast simulation of vehicles with non-deformable tracks,” *arXiv preprint*, Mar. 2017.
- [20] J. P. Vázquez-Rojas, “WetExplorer—Autonomous Tracked Robot (main),” <https://github.com/josepablov/WetExplorer-Autonomous-Tracked-Robot>, 2025, accessed 2025-07-21.