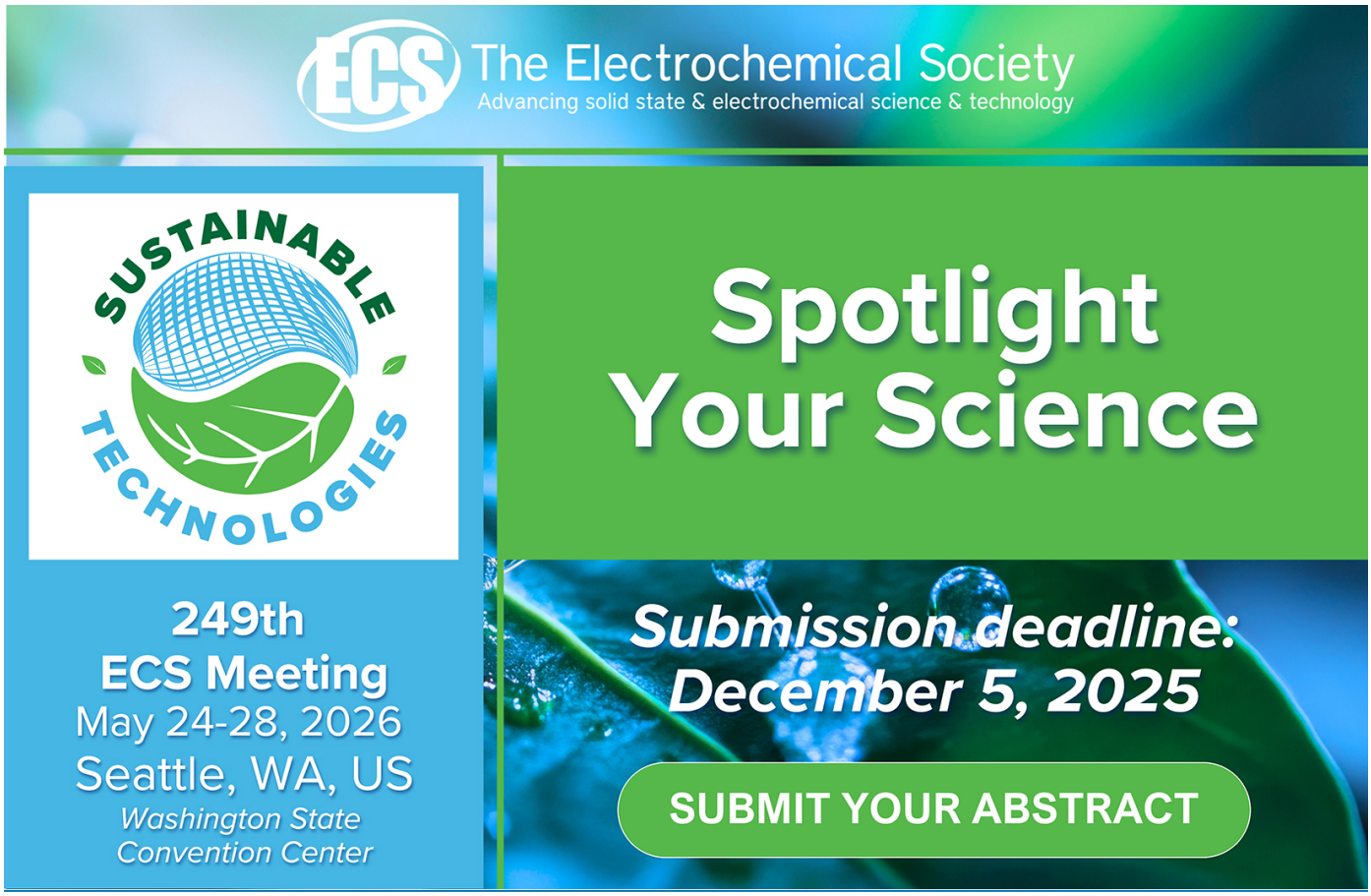
Journal of Physics:
Conference Series

PAPER • **OPEN ACCESS**

Multi-Waypoint Global Path Planning for Unmanned Surface Vehicles in Confined Environments

To cite this article: Cosmin Delea *et al* 2025 *J. Phys.: Conf. Ser.* **3123** 012023



Journal of Physics: Conference Series **3123** (2025) 012023 doi:10.1088/1742-6596/3123/1/012023

Multi-Waypoint Global Path Planning for Unmanned Surface Vehicles in Confined Environments

Cosmin Delea[1,*], Nico Zantopp[1], Mohamed G. Essa[2], Matteo Schmid[3], Johannes Oeffner[1]

[1]Fraunhofer Center for Maritime Logistics and Services CML, Hamburg, Germany
[2]Hamburg University of Technology, Hamburg, Germany
[3]Hamburg Port Authority, Hamburg, Germany
[*]Corresponding author. Email: cosmin.delea@cml.fraunhofer.de

Abstract.

Motivated by the SeaClear2.0 Project's aim to clean ports and coastal areas using aerial and marine robotics, this work focuses on the reference generation for autopilots of Uncrewed Surface Vehicles (USVs) deployed in confined areas, such as, rivers, channels, ports or harbours. In the envisioned scenario, a USV must navigate through multiple waypoints, which roughly sketch the intended route. The planned route is based on available chart data and doesn't take in consideration optimality of the route or uncharted obstacles. The study implements a global path planner consisting in an A* algorithm, extendable via Genetic Algorithm (GA), for particular scenarios, where the order of the given waypoints is not enforced. In addition, a kinematics-based path planner using the Line-of-Sight (LoS) algorithm is developed for creating the reference vector for the USV control system. The global path planner and LoS algorithms are tested both in simulation and in sea trials, using a medium-sized USV. For the simulation environment, a digital twin of the USV is used. The sea trials validate the concept in a representative environment. The entire software solution is developed under Robot Operating System 2 (ROS2) framework and implements Guidance Navigation and Control (GNC) architecture, commonly used for autonomous navigation of USVs. The global path planning algorithm efficiently handles multi-waypoint selection and finds the shortest path between each two waypoints while avoiding known static obstacles. Additionally, the LoS algorithm generates reliable references for the vessel's autopilot. The novelty of the proposed solution consists in the usage of meta-heuristic algorithms for solving a discrete optimisation problem and in the integration of nautical chart data within the global path planner. The paper highlights the potential of the robotic solution for maritime applications, beyond the chosen scenario, when coupled with a complete GNC solution.

1 Motivation

The application of USVs has seen a swift rise, driven by advancements in artificial intelligence algorithms, the computational power of embedded systems, and the availability of various software development tools which have simplified the emulation of hardware and integration of additional layers of complexity of the navigation software. Consequently, researchers are striving to leverage these advancements in the creation of USVs and harness their benefits to improve numerous maritime tasks. A fundamental problem to solve for an USV is autonomous navigation through a series of waypoints situated within a confined environment, while adhering to a specific path or trajectory dictated by a task planner, commonly called global path/trajectory planner. In a confined environment, a waterway is meant to have limited space and potential hazards, such as marine traffic or uncharted obstacles. Although the ability to obtain good tracking performance is directly impacted by the performance of the control system, the planner can further enhance it by optimising the generated paths or trajectories by considering the vessel's dynamics.

Motivated by the SeaClear2.0 Project and its primary goal of employing marine robotics to clean ports and coastal areas by removing surface and seafloor debris, an USV acts as a shuttle tender, having to autonomously navigate to a specific location, where another USVs is located. The latter is assumed to be laden with a piece of litter weighing up to 250 kg, that can be discarded from a position 1.5 m behind the USV and at least 0.5 m above the water surface. The seafloor waste collected by the shuttle tender needs to be transported and taken to shore, for further processing. Furthermore, for scalability purposes, it is assumed that there might be multiple USVs, each simultaneously laden with a litter object, that needs to be discarded. The lumped weight of all the litter objects collected simultaneously and ready to be deposited on the shuttle tender USV is not considered to exceed the overall payload of around 400 kg.
There are two types of challenges in respect to the route planning for the shuttle tender USV: firstly, determining the optimal sequence route for the shuttle tender USV to cover all locations where the laden USVs are located. Secondly, for every target location, the shortest route, while avoiding obstacles needs to be found. The primary objective of this work is to implement a path planning algorithm that manages multi-waypoint selection, identifies the shortest collision-free path between each pair of points, and ultimately produces valid control reference input. This is accomplished by deploying a global path planner that incorporates a GA for multiple waypoint selection and an A-star algorithm for shortest path identification between each pair of points, and a kinematics-based path planner that uses the LoS algorithm to generate valid control reference input. In order to be implemented in the existing systems all algorithms need to be compliant with ROS2 framework. The effectiveness of the proposed techniques was assessed through experiments conducted both in a simulation environment and sea trials using a small-size catamaran USVs.

2 Literature Review

According to the literature, authors have proposed different approaches to handle the multi-waypoints and multi-objectives problem by proposing different meta-heuristic techniques.
In [1], the authors proposed a multi-objective nonlinear dynamic swarm optimisation algorithm, to generate a path that guarantees the shortest, smoothest, and most economical path in addition to avoiding obstacles and currents. The USV can choose the best path by integrating the four objectives mentioned earlier. Furthermore, compared with different numerical solutions, the effectiveness of the proposed algorithm is confirmed.
In [2], the authors aimed to observe the environmental hazards in the Ypacarai Lake in Paraguay. They modelled the problem as a confined travelling salesman problem by deploying an USV into the lake to visit a ring of beacons placed at the lake's shores to exchange data, however, not all paths between beacons are valid. The GA is utilised to solve this problem and is evaluated concerning other types of algorithms such as brute force, randomised, and greedy algorithms. The GA exceeded the performance of the brute force and randomised algorithms by 15 % and 3 % respectively, while the greedy algorithm did not succeed in visiting all the beacons.
Furthermore, researchers utilised graph traversal algorithms to find the shortest path between the start and target point, while avoiding colliding with obstacles in global-based maps.
In [3], the authors used the Dijkstra algorithm to find the shortest path for an Autonomous Surface Vehicle (ASV) on a grid map based on practical marine environments such as Portsmouth harbour. The optimal path is evaluated in terms of computational time between the start and goal waypoints. This algorithm is compared to the Artificial Potential Field (APF) path planner in a real-time marine environment. The results proved that the Dijkstra algorithm is suitable for global path planning in environments containing static obstacles. The algorithm demonstrated high performance in terms of computational time and path length preceding the APF algorithm.
The authors in [4] proposed an A-star algorithm that exceeds the general concept of using graph-based methods, as it takes into consideration static and moving obstacles of the environment, as well as, different current intensities, and the effect of headwinds and tailwinds currents. Furthermore, the implemented algorithm maintains a circular boundary to ensure a safe margin for the USVs. An evaluation was performed in a simulation environment, where different environmental conditions were applied, and both the path length and computational time were assessed. The results showed the effectiveness of the proposed approach as a global path planner.
Eventually, reference control inputs need to be fed to the controller to allow a feasible constrained motion for the USV to follow the intended path. Therefore, a kinematics-based planner converts the desired waypoints to follow to a controllable USV state is discussed in the research as well.
The authors in [5] introduced an improved LoS guidance algorithm that alternates adaptively according to the path following error. They integrate the proposed guidance module with a non-linear backstepping controller, which is assessed for stability by Lyapunov stability analysis methods, to control the USV. Furthermore, they formulated the path following error and control problem in the Serret-Frenet coordinate frame, besides, they combined the longitudinal path following error in this frame with the improved LoS, and eventually, they introduced a virtual tangential velocity input to the

path generated, to solve the under actuation problem of USVs. Simulation experiments evaluate the performance of the proposed technique. Results show high efficiency and validity of the algorithm.
In [6] the authors presented an Integral-Differential Line of Sight (IDLOS) approach to enhance the accuracy of normal LoS algorithms in path tracking operations. The authors integrate both, an adaptive integral term to enhance the convergence rate and a differential term to decrease overshoot. MATLAB simulation results evaluate the proposed technique and compare it to normal LoS and IDLOS. The IDLOS achieved a lower convergence rate of 1.4 sec and decreased the integral absolute error of the cross-track error compared to LOS, besides, it decreased the overshoot about 13 times compared to IDLOS.
It has been noticed that many works aim to use meta-heuristic algorithms to solve NP-space problems such as waypoint order selection. For the current work, a GA from the umbrella of meta-heuristic algorithms was chosen to solve the optimisation problem. GA accounts for discrete and continuous problems compared to the particle swarm algorithm, which deals mainly with continuous problems and needs some modifications to handle discrete problems, and the Ant colony algorithm that may get trapped in local minima. Furthermore, different graph traversal algorithms such as Dijkstra and A-star have been used in other works. However, Dijkstra is computationally expensive, therefore, an A-star one has been chosen for its optimally and fast computation time. Additionally, LoS was mainly used to generate reference control inputs to the controller. It extends to different types such as the IDLOS. However, the traditional LoS implementation was used in this work.

3 Methodology

Autonomous navigation of an USV requires the development of a comprehensive navigation solution capable of processing reference points (or paths or trajectories), integrating feedback from the sensory system and generating appropriate commands for the actuation system. For aerial drones, such solutions are commonly referred to as *flight stacks*, whereas for waterborne systems, the term GNC system, as theorised by Prof. Fossen in [7], is more prevalent. The structure of a GNC system is not universally fixed, with various authors introducing modifications to the specific features of each GNC module. Such slightly modified example can be found in [8]. However, these modifications primarily affect the software architecture rather than the core functionality of the navigation solution. As depicted in Figure 1, the GNC system is mainly divided into three main modules: *Guidance*, *Control*, and *Navigation*. The *Navigation* module is responsible for providing the necessary information about the vessel's position, velocity, and pose, which is used by the other two modules. The *Control* module is responsible for generating the control commands to the thrusters based on the reference provided by the *Guidance* module. The *Guidance* module is responsible for generating this reference, in form of a vector containing the desired values of the vessel's states, calculated based on the current position of the vessel and the desired path to be followed.

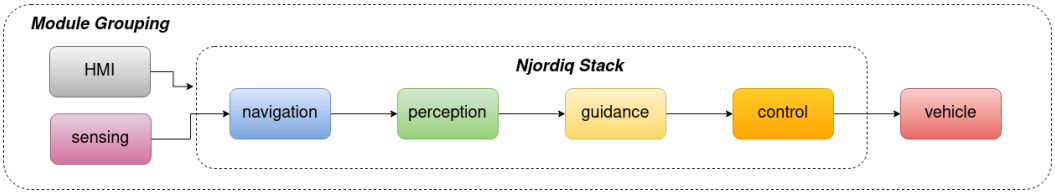


Figure 1: NJORDIQ Stack - GNC architecture for USVs

This work contributes to the development of a complete GNC solution for navigating USVs in confined areas under operational supervision. The primary focus is on the *Guidance* module, which is responsible for parsing the input itinerary (provided as a list of desired locations), generating an optimal path for the vessel to reach these locations in a specific sequence, and dynamically adjusting the active location to be followed based on the vessel's current position. While the onboard controllers are assumed to robustly handle environmental disturbances such as wind and waves, as well as compensate for sensory noise, the effects of these disturbances can be mitigated through effective planning. Proper planning minimises sharp heading changes, reduces sailing time under adverse conditions (e.g., side winds), and the overall time required to reach all objectives. The planning process is conducted at two levels: global and local. Global planning involves generating a path that accounts for a priori knowledge of the environment, such as waterways, piers, and restricted areas. Local planning, on the other hand, addresses dynamic elements such as wind, vessel traffic, and obstacles encountered in real-time. The current work focuses on the global planning aspect, while the local planning will be addressed separately. The design objectives is to complete a series of at least 4 waypoints scattered around the designated test area, described in Section 4, with a twin-screw

propeller catamaran USV, with an an average cruise speed of 1 m/sec calculated over the complete length of the reference path. The cross-track error needs to be kept within 5 m.
The desired state vector $\mathbf{x}_d$ for the vessel is defined as follows:

$$\mathbf{x}_d = \begin{bmatrix} x_d \\ y_d \\ \psi_d \end{bmatrix} \tag{1}$$

where:

- x_d and y_d represent the desired position in the global coordinate frame
- ψ_d is the desired yaw angle (heading) of the vessel

, all of them expressed using the North-East-Down (NED) convention, where the x-axis points North, the y-axis points East, and the z-axis points downwards.
In view of the envisioned operation briefly explained in Section 1, the USV receives a list of geographical locations, called waypoints, assumed to be the positions of the USVs collecting marine litter. Each location needs to be visited once for transboarding the collected waste into the litter bay, on-board the USV. For the scope of this work, there are no time or vessel's heading constraints upon reaching these locations. The only requirement is to visit all the locations in a specific order, which is determined by the global path planner. As a result, the desired velocity components (surge, sway and yaw) are omitted from the desired state vector. As depicted in Figure 2,the list of waypoints are the input for the *Guidance* module, which, in turns, provides $\mathbf{x}_d$ output for the controller located in *Control* module.

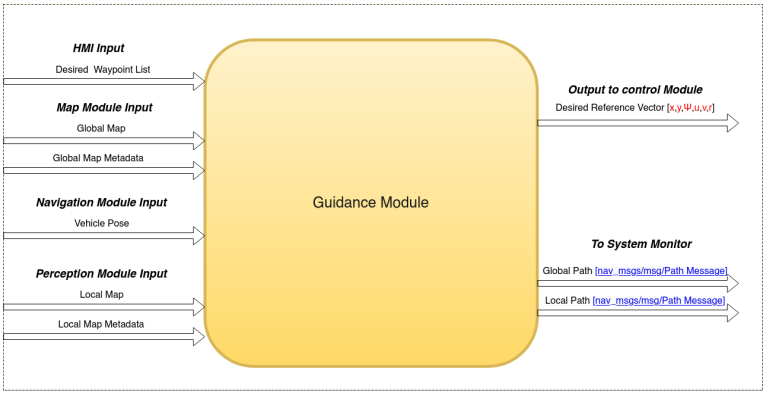


Figure 2: NJORDIQ Stack - detailed *Guidance* module

The two modules are used within a custom ROS2 GNC architecture, called *Njordiq*-stack which mostly follows the architecture proposed in [8]. As seen in Figure 2, the two modules make use of filtered navigation and perception data, which is used in each step when calculating the desired course reference. However, due to incomplete real-time integration of the Light Detection and Ranging (LiDAR)-based vessel detector (presented in a separate work), for the experiments conducted in this paper, the obstacles encountered in sea trials are identified using public satellite data, which is not completely accurate. As a result, the dynamic obstacle avoidance was only validated in simulation, while the static obstacle avoidance was handled using satellite data. In further works, the validation and integration of the vessel detector will be presented, thus enabling the validation of a real environment, with both static and dynamic obstacles. The assumption used for sea trials is that the target USVs is located within a work envelope, where there aren't any uncharted obstacles.
The path planner is implemented initially by generating an occupancy grid map, where each cell in the occupancy grid has a value representing the probability of the occupancy of that cell. Values close to 1 represent a high probability that the cell contains an obstacle. Values close to 0 represent a high probability that the cell is not occupied and obstacle free. The map grid size was set at 5.68 m, meaning that each cell of the occupancy grid map represented a $5.68 - by - 5.68$ m space, which would have a non-zero value even if it was partially occupied by an obstacle. The obstacles referred to in this work are static (e.g landmass, piers, tonnes, mooring posts etc.) and are computed using sea chart information.
This information will be fed to a global path planner consisting of GA and A-star algorithm to find the optimal sequence of waypoints to be visited, as well as the shortest path between each waypoint. The

global path planner aids the USV to avoid collision with static pre-known obstacles. Eventually, the kinematics-based planner, represented by LoS generates a valid reference vector for an existing controller.
The multi-waypoint problem is formulated as a Travelling Sales man Problem (TSP), in which the USV starts its itinerary from a depot point and needs to visit multiple waypoints and return to its depot point, as represented by the chromosome shown in Figure 3.

	$Gene_{j,0}$	$Gene_{j,k}$	...	...	$Gene_{j,n}$	$Gene_{j,n+1}$
$Chromosome_j$	$Depot$	$Point_k$	....	....	$Point_n$	$Depot$

Figure 3: Chromosome j representation

The fitness function of each chromosome is provided by the A-star algorithm, as the estimated cost function between each two pair of points is summed through the chromosome to provide the final fitness value for each chromosome as described by the following equation 2.

$$C^j_{fitnessvalue} = \sum_{0}^{k=C^j_{length+1}} A - star_{f(n))}(C^j_k, C^j_{k+1}) \tag{2}$$

where, C^j_k denotes the chromosome of the j^{th} order and the k^{th} gene in the population. $f_{(n)}$ is the A-star estimated cost function.
The elitism, crossover and mutation parameters are 10 %, 80 % and 10 %, besides 10 populations and 8 generations were used.
Afterward, inspired by LoS equations in [7], the kinematics-based path planner was implemented, which converts the desired waypoints into desired course angles that can be manipulated by the controller. The main errors defined by LoS are the cross-track and along-track errors shown in equations 3 and 4.

$$e(t) = -(x(t) - x_k)\sin\alpha_k + (y(t) - y_k)\cos\alpha_k \tag{3}$$

$$s(t) = (x(t) - x_k)\cos\alpha_k + (y(t) - y_k)\sin\alpha_k \tag{4}$$

where, $x(t)$ and $y(t)$ are the USV current position. x_k and y_k are the position of the first waypoint. While α_k is represented by equation 5

$$\alpha_k = \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \tag{5}$$

The cross-track error is the error between the USV and the path in the normal direction, and the along-track error is the error between the USV and the path in the tangential direction. The cross-track error assesses the path following performance, while the along-track error defines the switching mechanism of waypoints. Both errors can be altered by tuning two main parameters the lookahead distance Δ and the radius of acceptance R.

4 Experiment Setup

To evaluate the performance of the *Guidance* module, the complete GNC system has been tested both in simulation and in sea trials using Fraunhofer CML's SeaDragon USV (see Figure 4), which is a 6.5 m x 2.2 m catamaran, having 350 kg payload and capable of developing up to 6 kn speed. The USV actuation system consisted in a twin-screw propeller, capable of developing 6 kW power. In simulation, the digital twin of the SeaDragon USV, with similar, but not identical hydrodynamic properties has been used inside Gazebo simulator, using the VRX simulator plugin [9].
Multiple experiments have been deployed in the Gazebo simultation environment to validate the proposed algorithms. Initially, a basic experiment is conducted to test LoS algorithm, in which a 30 × 40 m rectangle path is set to assess the LoS algorithm. LoS generates reference desired course angles to allow the USV controller to follow the desired path.
Then an low-complexity experiment is conducted, in which LoS is integrated with the A-star algorithm, multiple waypoints are added arbitrarily on the global map. The A-star discretised the path into smaller segments decided by the global map resolution and fed these waypoints to the LoS to produce reference vector. The A-star should find the shortest path between each two waypoints while avoiding collision with global static obstacles.
Finally, an medium-complexity experiment is conducted by integrating the GA with A-star and LoS. Multiple waypoints are selected arbitrarily on the map as well, however, GA should be able to find the best sequence of waypoints that need to be visited.

Figure 4: SeaDragon USV, catamaran used for validating the *Guidance* module

The sea trials took place in February and October 2024 in Lotsekanal (Pilot's Channel) in Hamburg, Germany, depicted in Figure 5. The channel is situated in the southern part of the port-city, behind a lock that protects it from most of the currents of the Elbe river. As a consequence, much of the tidal effects and currents are mitigated. On average, the currents in the Elbe river are around 1.2 m/sec, or 2.3 kn. Most of the disturbances are caused by the wind, which during the sea trials has been blowing at a speed of 6 m/sec to 8 m/sec, or 11 kn to 16 kn. The wind direction was from N, NE and E, depending on the days.

Figure 5: The Lotsekanal (Pilot's Channel), situated in the Port of Hamburg, has been chosen for conducting the sea trials. The designed area, marked with dark-red colour, is approximately 200 m long and 65 m wide in its broadest point

For the sea trials the GA is integrated with the A-star search and the USV is steered to follow the waypoints set arbitrarily. The lookahead distance and radius of acceptance set for simulation experiments are 100 m and 10 m respectively. In sea trials, due to the confined space, they are set to be 10 m and 1 m respectively.

5 Results and Discussions

In this section, the results of the aforementioned experiments are evaluated. The metrics used to evaluate the path following performance are the Root Mean Square Cross-track Error (RMSE) and the sailing time taken by the USV to finish visiting all the desired waypoints. It is worth noting that the results obtained are not compared to specific benchmark reference data, as no available data fulfilling the aforementioned experiments were achieved before. Therefore, the performance of the proposed planners is compared with each other.

5.1 Simulation Results

Firstly, a simple experiments with four waypoints placed in a square sequences is conducted, with the results shown in Figure 6. It is shown that the difference between the reference path and the actual path taken by the USV through the whole journey. A maximum of nearly 5 m deviation between the actual and reference frame took place at some locations, especially at the curves. Additionally, it could be noticed that, despite the high discretisation and sharp edges of the reference path, the LoS was able to guide the USV.

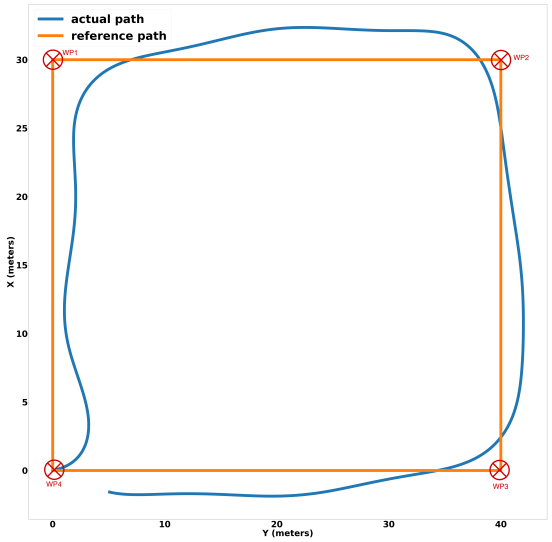
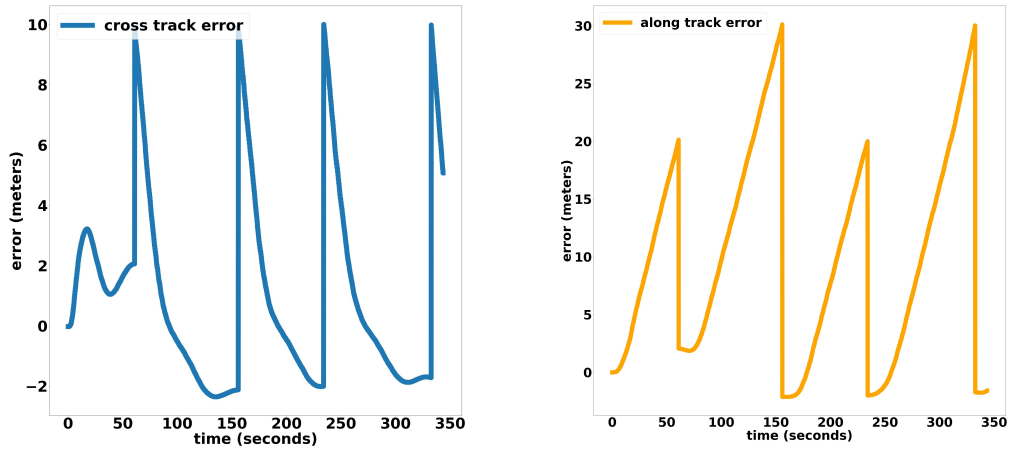


Figure 6: Simulation path following performance

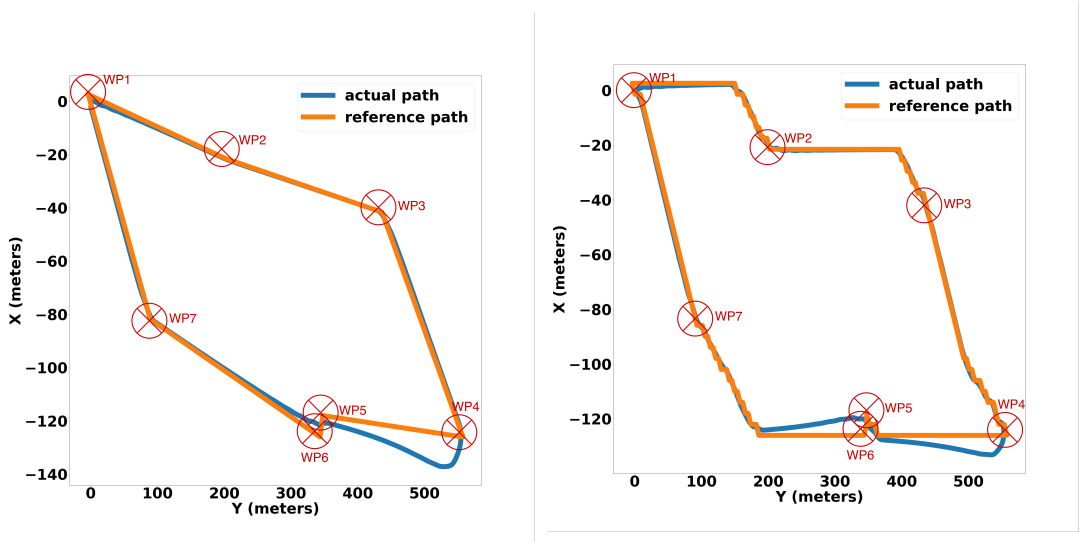
Furthermore, the cross-track error in Figure 7a highlights the USV tracking performance, also including the reference course angle. The rapid change in the cross-track error is due to the waypoints switching mechanism which is shown by the along-track error in Figure 7b. The along-track error does not evaluate the USV performance, as it only defines the accepted range of following the current waypoint. With the USV being within the user-defined radius of acceptance of the current reference point, the latter is considered as being reached and the next reference point is set as target. Consequently, the peaks of the along-track error coincide on the time axis with the peaks of the cross-track errors.
After analysing the performance, the USV was able to finish this path in 5 min and RMSE of 4.25 m.
Even though the desired performance is obtained, it is worth mentioning that the current version LoS does not allow the USV to avoid collision with static obstacles. Static obstacles are considered the points on the occupancy grid with values > 0.
Next, the A-star algorithm is integrated with LoS in a low-complexity experiment to aid static collision avoidance. The path tracked in this experiment is shown in Figure 8.
The results in this experiment show a different path following behaviour, as the LoS connects the waypoints in straight lines, however, when integrating the A-star search algorithm, the USV follows the path in a more optimal manner.
In Figure 9 the cross- and long-track errors of the two approaches in this experiment are compared. On the graphs the A-star behaviour during the rapid changes of the cross-track error and along-track error values can be observed. This is due to the discretisation property of the A-star, as it divides the large path into smaller paths based on the map image resolution, which, in this case, is 5.68 m/cell.
Table 1 shows the summary of the the path planning performance of the USV with and without the integration of the A-star algorithm. As it could be noted A-star integrated with LoS produced a lower



(a) Simulation: cross-track error

(b) Simulation: along track error

Figure 7: Simulation LoS path following error evaluation



(a) Simulation LoS path following performance

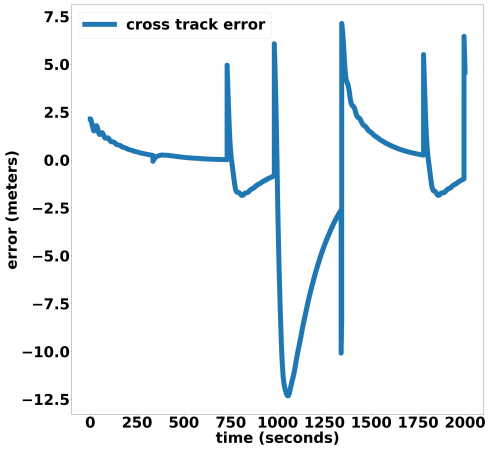
(b) Simulation A-star integrated with LoS path following performance

Figure 8: Simulation path following performance: LoS vs. A-star integrated with LoS

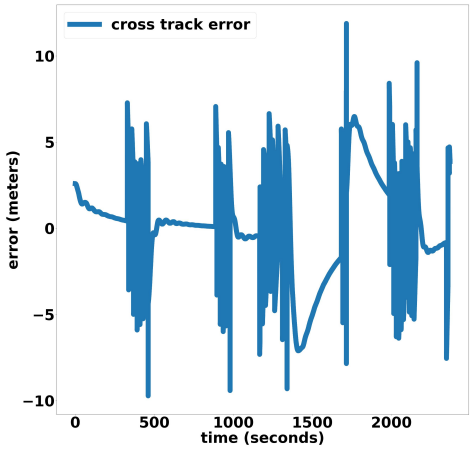
RMSE, than LoS, Furthermore, the sailing time is nearly 10 min less when incorporating A-star. However, the A-star algorithm will not select the best sequence of waypoints to be visited.

Table 1: Assessment of USV's path planning performance by evaluating the RMSE and sailing time (low complexity experiment)

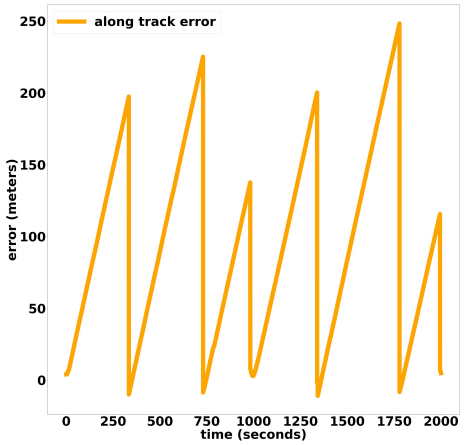
	RMSE (meters)	**Sailing time (minutes)**
LoS	**5.3**	**48.9**
A-star integrated with LoS	**3.12**	**39.5**



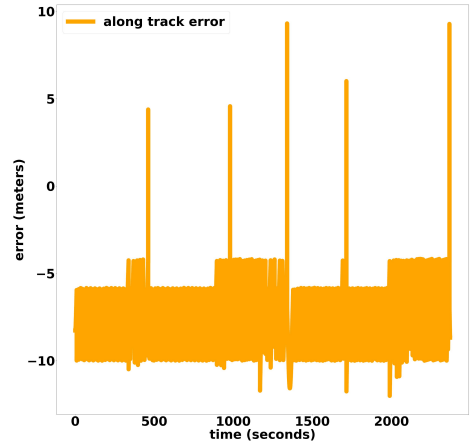
(a) Simulation LoS: cross-track error



(b) Simulation A-star integrated with LoS: cross-track error



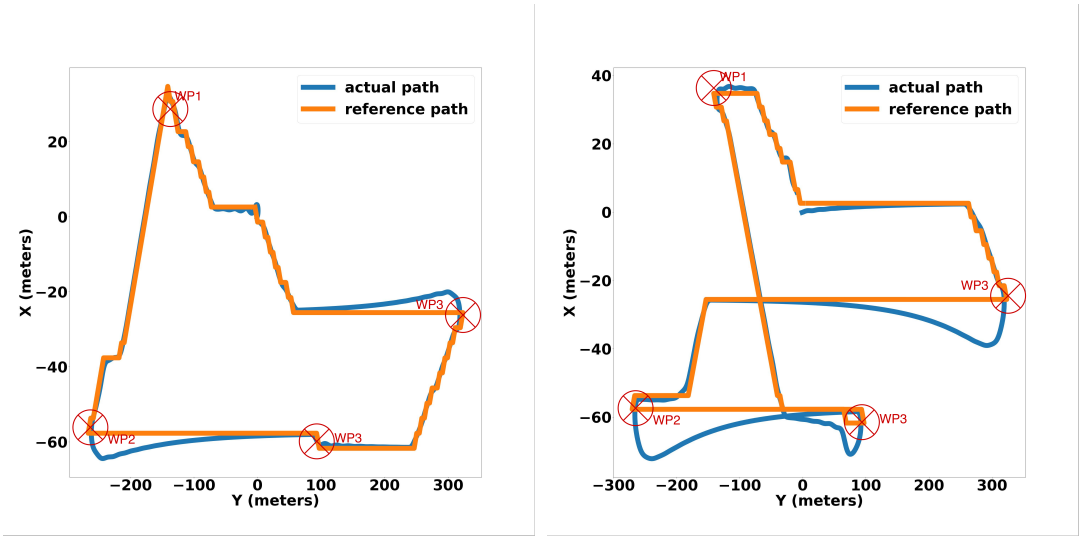
(c) Simulation LoS: along-track error



(d) Simulation A-star integrated with LoS: along-track error

Figure 9: Simulation path-following error comparison between LoS and A-star integrated with LoS: cross-track error (top row) and along-track error (bottom row).

In a medium-complexity experiment, the GA is integrated with A-star search and LoS to find the best order of waypoints to be visited, as well as, maintaining the shortest path between every two waypoints in the sequence, while avoiding collision with static obstacles. As could be observed from Figures 10a and 10b the A-star search cannot find the best order to visit the user-given waypoints. On the other hand GA is able to select the best possible sequence of waypoints which will lead to the shortest path.



(a) Simulation A-star integrated with LoS path following performance

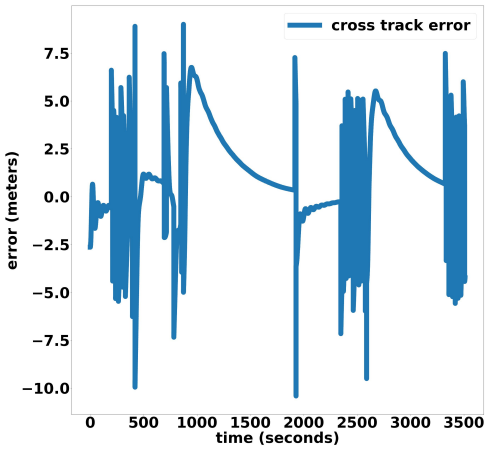
(b) Simulation GA integrated with A-star and LoS path following performance

Figure 10: Simulation path following performance: A-star integrated with LoS vs. GA integrated with A-star and LoS

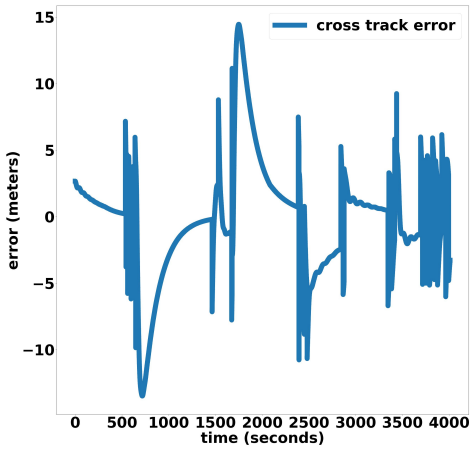
Figure 11 shows the cross-track and along-track errors of the USV while following the desired path. The same discretisation behaviour is maintained due to the integration with the A-star algorithm. Table 2 demonstrates how GA enhances the performance as the RMSE decreased nearly by half and sailing time decreased by nearly 8 min.

Table 2: Assement of USV's path planning performance by evaluating the RMSE and sailing time (medium-complexity experiment)

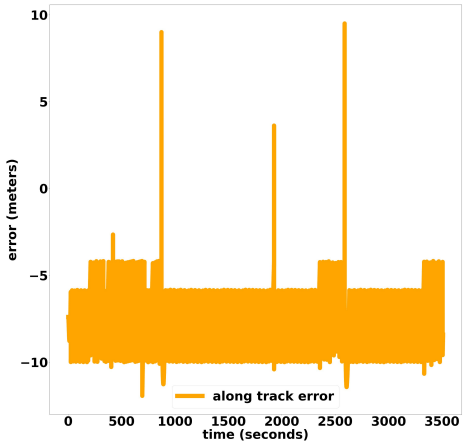
	RMSE (meters)	**Sailing time (minutes)**
A-star integrated with LoS	**4.36**	**66.8**
GA integrated with A-star and LoS	**2.92**	**58.5**



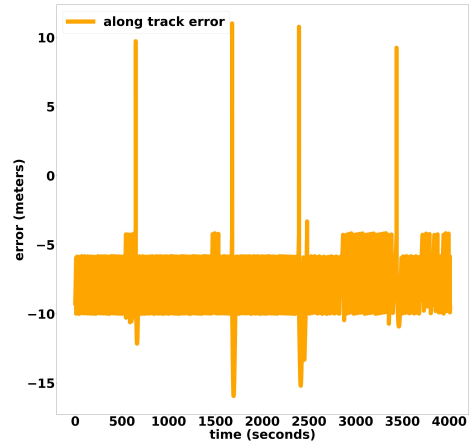
(a) Simulation A-star integrated with LoS path following performance: cross-track error



(b) Simulation GA integrated with A-star and LoS path following performance: cross-track error



(c) Simulation A-star integrated with LoS path following performance: along-track error



(d) Simulation GA integrated with A-star and LoS path following performance: along-track error

Figure 11: GA path following error evaluation in simulation (top) and real-world (bottom)

5.2 Sea Trials Results

For the sea trials, the path-following performance of the USV is depicted in Figure 12, where the vessel navigates through four arbitrarily placed waypoints. The sequence of waypoints determined by the GA integrated with A-star and LoS is also illustrated.

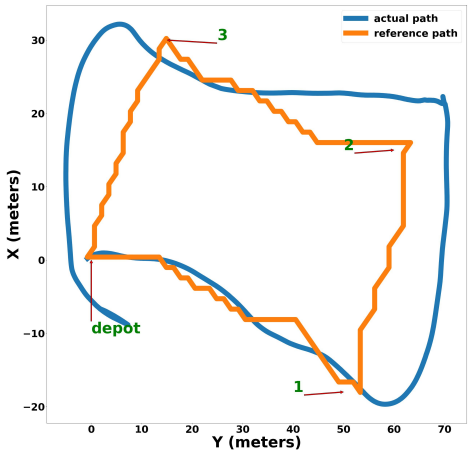
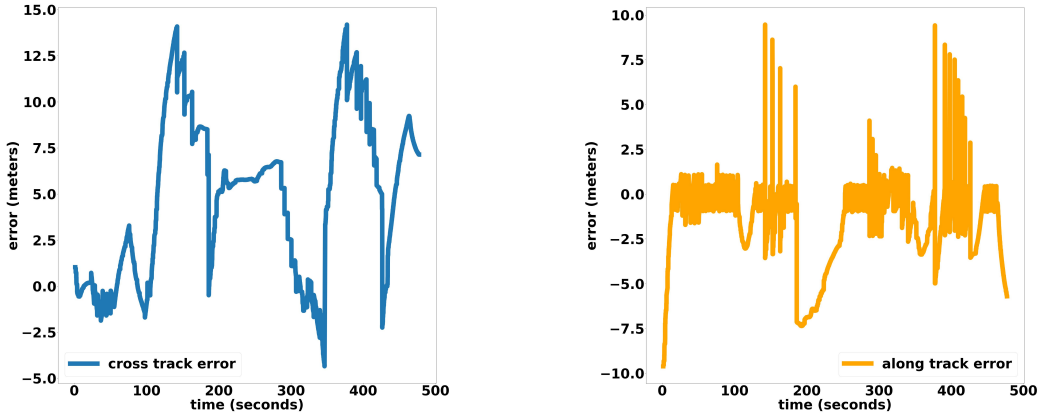


Figure 12: Sea trials path following performance

Figures 13a and 13b illustrate the USV's errors, highlighting the impact of wind and currents compared to simulation results. Both RMSE and sailing time nearly doubled, exceeding expectations. Increasing the lookahead distance and radius of acceptance could reduce the along-track error but may increase the cross-track error, which is undesirable.



(a) Sea trials GA cross-track error

(b) Sea trials GA along-track error

Figure 13: Sea trials GA path following error evaluation

The overall decrease in tracking performance within the sea trials is due to three factors: inconsistent hydrodynamical model of the USV, leading to alteration of the tuning parameters, lack of environmental data (wind, waves) and slow low-level thruster controller response. Out of the three factors, the lack of wind and current disturbances information, has, in this case, the highest impact on the overshoot and, implicitly, on the cross-track error. One straightfoward way to decrease the

cross-track error is direct measurement of the wind speed and direction and integration into the path planner. As depicted in Figure **??**, the USV was able to follow the desired path with a bearing angle as desired course angle. The bearing angle is defined as the angle between the North and the line connecting the USV position and the next waypoint. The USV was able to follow the path with a maximum cross-track error of 5 m, using same lookahead distance and radius of acceptance as in the all the sea trials, i.e., 10 m and 1 m respectively. The USV was able to finish the path in 6 min, with a RMSE of 3.5 m. This shows that the design objectives are met also on the real vehicle, however, due to the changes in expressing the desired angle, a consistent comparison between the simulation and sea trials results cannot be drawn.

6 Conclusion

In this work, a comprehensive global path planning algorithm, incorporating both genetic and A-star algorithms, has been presented. It proved to be highly effective in identifying the best sequence of waypoints and in calculating the shortest routes between them. The automatic sequencing of waypoints contributes to the SeaClear2.0 system by upscaling its capabilities for multiple marine litter collection systems, that are being serviced by a single shuttle tender USV and, thus, minimising the costs and operation time. Furthermore, in the simulation experiments, the LoS algorithm successfully provided suitable course values, ensuring the USV followed the predetermined route, while mimising the cross- and along-track errors. However, the sea trials have shown the limitations of the LoS in presence of environmental forces (i.e. wind). Failing to augment the wind into the calculation of the desired course, yielded having the bearing angle as desired course as sufficient for obtaining the desired performance. Due to the fact that the bearing angle is not necessary the optimal angle, the trade-off consisted in increased control effort for stabilising the vessel at higher turning rates. Thus, during the sea trials, only the guarantee that the shortest and collision-free path between the given waypoints has be validated, along the identification of the best sequence. The collision-free attribute was solely guaranteed via the occupancy grid generated using sea charts.

The focus of this work has been laid on the *Guidance* module, aiming to develop a range of path planning strategies to securely navigate a medium-sized USV along a designated route and fulfill the given design objectives. The scenarios presented in all experiments presented in this paper, focused more on the tracking performance for randomly placed waypoints, which could lead to sharp changes of the course angle, rather than on avoiding the obstacles seen on the occupancy grid. However, more complex scenarios, with more challenging paths have been succesfully tested and will be presented in future works.

Two considerable limitations for the tracking performance are the USV's type of actuation and the availability of environmental data. Having an underactuated vessel and no environmental data into the global path planner, means exposing the vessel to the risc of side drifting, which cannot be compensated by the control system, while keeping the reference course angle. Integral and adaptive LoS could increase the tracking performance and system robustness in high wind and current environments. In addition, Dubin curves could be integrated to generate more smoother paths. For the further works actuation of the SeaDragon USV will be changed from twin-screw propellers, to twin-azimuth propellers, thus making the USV capable of controlling also the sway.

In addition to the static obstacle data obtained a priori from sea charts, an online path planner is required to continuously optimise the route based on real-time odometry and remaining waypoints. Dynamic obstacle avoidance should be addressed through a local path planner with a limited range compared to the global planner. To balance computational efficiency, an adaptive global A-star planner with a modified search strategy should be integrated alongside the local planner.

7 Acknowledgment

The work presented has been conducted within the SeaClear2.0 Project and has been co-funded by the European Union under the Horizon Europe Programme (Grant Agreement no. 101093822).

References

[1] Yong Ma, Mengqi Hu, and Xinping Yan. Multi-objective path planning for unmanned surface vehicle with currents effects. *ISA transactions*, 75:137–156, 2018.

[2] Mario Arzamendia, Derlis Gregor, Daniel Gutierrez Reina, and Sergio Luis Toral. An evolutionary approach to constrained path planning of an autonomous surface vehicle for maximizing the covered area of ypacarai lake. *Soft Computing*, 23:1723–1734, 2019.

[3] Yogang Singh, Sanjay Sharma, Robert Sutton, and DC Hatton. Towards use of dijkstra algorithm for optimal navigation of an unmanned surface vehicle in a real-time marine environment with results from artificial potential field. *TransNav: International Journal on Marine Navigation and Safety of Sea Transportation*, 12(1):125, 2018.

[4] Yogang Singh, Sanjay Sharma, Robert Sutton, Daniel Hatton, and Asiya Khan. A constrained a* approach towards optimal path planning for an unmanned surface vehicle in a maritime environment containing dynamic obstacles and ocean currents. *Ocean Engineering*, 169:187–201, 2018.

[5] Tao Liu, Zaopeng Dong, Hongwang Du, Lifei Song, and Yunsheng Mao. Path following control of the underactuated usv based on the improved line-of-sight guidance algorithm. *Polish Maritime Research*, 24(1):3–11, 2017.

[6] Guoqing Zhou, Jinchun Lin, Jinhuang Wu, Zhexian Liu, Gongbei Wu, Dawei Zhao, Chao Xu, and Haotian Zhang. An integral-differential los algorithm for usv path-tracking control. In *International Conference on Automation Control, Algorithm, and Intelligent Bionics (ACAIB 2023)*, volume 12759, pages 326–331. SPIE, 2023.

[7] Thor I Fossen. *Handbook of marine craft hydrodynamics and motion control*. John Wiley & Sons, 2011.

[8] Anete Vagale, Rachid Oucheikh, Robin T. Bye, Ottar L. Osen, and Thor I. Fossen. Path planning and collision avoidance for autonomous surface vehicles I: a review. *Journal of Marine Science and Technology*, 26(4):1292–1306, December 2021.

[9] Brian Bingham, Carlos Aguero, Michael McCarrin, Joseph Klamo, Joshua Malia, Kevin Allen, Tyler Lum, Marshall Rawson, and Rumman Waqar. Toward maritime robotic simulation in gazebo. In *Proceedings of MTS/IEEE OCEANS Conference*, Seattle, WA, October 2019.