

ENC-based Anti-Grounding and Anti-Collision System for a Wave-Propelled USV

Alberto Dallolio¹, Thea K. Bergh¹, Pedro R. De La Torre¹, Henning Øveraas¹ and Tor A. Johansen¹

Abstract—Wave-propelled unmanned surface vehicles (USVs) rely on the forces exerted by the environment to navigate as intended. The unpredictability of the environment and the low speed and limited maneuverability that characterizes such platforms motivates the need of a continuous monitoring of both static and dynamic obstacles over a long time horizon. In this manuscript we present an automatic anti-collision and anti-grounding system that integrates digital charts and automatic identification system (AIS) messages received onboard to enhance the situational awareness perceived by a wave-propelled USV, and enable evasive maneuvers to avoid grounding and collisions with static and dynamic obstacles. The acquired information is used in a scenario-based model predictive control (SB-MPC) algorithm that computes optimal behaviors that minimize the risk of collisions, grounding and damage. Additionally, the proposed system integrates environmental factors (wind, waves and sea currents) in the SB-MPC optimization process, in order to command the safest behavior when high sea states increase the risk of drifting and grounding. The information contained in Electronic Navigational Charts (ENCs) is represented with point clouds, enabling fast software manipulation of large areas and therefore increasing the USV responsiveness when scenarios with static obstacles turn into dynamic collision avoidance scenarios. The proposed anti-collision and anti-grounding system is tested in simulations and field experiments.

I. INTRODUCTION

Unlike common marine vehicles, the propulsion of wave-propelled unmanned surface vehicles (USV) relies on the forces exerted by the environment, for example, waves, currents, and wind. The unpredictability of the environment, the low speed and the limited maneuverability that characterizes wave-propelled USVs motivate the need of monitoring continuously both static and dynamic obstacles at sea. Detailed knowledge of the surrounding environment and vessels is important in order to maximize safety and ensure risk-free operations in both the near and distant future. While the knowledge of surrounding vehicles can be acquired with the automatic identification system (AIS) and radars, marine vessels usually rely on Electronic Navigational Charts (ENC) [1] to locate themselves with respect to land and static objects present at sea.

Supporting navigation of marine vehicles with digital charts is an already investigated topic. However, most of the efforts concerning unmanned platforms focus on solving path

planning problems [2] and make use of classical techniques such as Voronoi diagrams, rapidly-exploring random trees, artificial potential fields, occupancy grids and A* algorithm. In [3], digital charts are instead used for determining whether a route is navigable or not and a method to obtain an optimized navigation area using water depth information is proposed. An interesting example of situational awareness based on Electronic Navigational Charts (ENC) is described in [4], where several features are extracted from the digital catalog and enclosed in a database. This procedure enhances safer operations by providing the USV's onboard navigation system *a priori* knowledge of the environmental features and hazards.

Our work combines the information extracted from electronic charts, and presents an extension of the simulation-based Model Predictive Control (MPC) algorithm for collision avoidance described in [5] and [6]. This algorithm computes COLREGS compliant behaviors that ensure collision-free navigation among vessels equipped with AIS [7]. The same algorithm is expanded in order to account for grounding hazards and static obstacles such as islands, buoys, beacons and permanent or temporary surface installations. This is achieved by integrating the static obstacles represented as point clouds within the field-tested MPC-based algorithm described in [5]. Information about land and static objects are stored in a database stored in the USV's onboard computer and are retrieved by the navigation software in the form of WGS84 location point clouds via fast SQL queries. In the proposed implementation, the current and future (forecast) information about the sea state (wind and sea currents speed and direction, surrounding bathymetry, waves height and frequency) are included in the optimization process, as

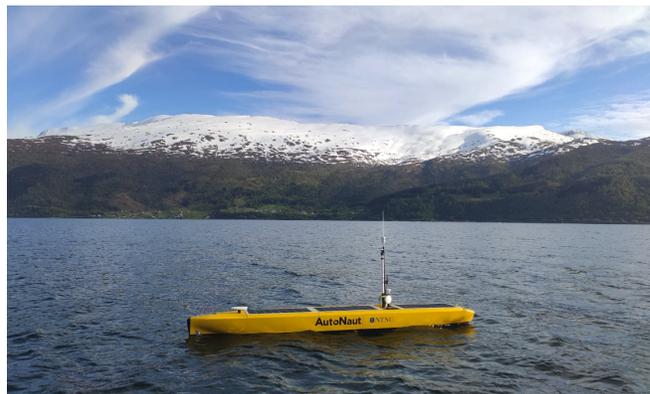


Fig. 1: The AutoNaut USV.

¹The authors are affiliated with the Center for Autonomous Marine Operations and Systems (NTNU-AMOS), Department of Engineering Cybernetics, NTNU Norwegian University of Science and Technology, 7491 Trondheim, Norway alberto.dallolio, theakbe, pedro.torre, henning.overaas, tor.arne.johansen@ntnu.no

grounding might be preferred over a collision with other vehicles in certain circumstances.

This work is specifically developed for the AutoNaut [8], a commercially available wave-propelled USV designed for long-duration field deployments (see Figure 1). The anti-collision and anti-grounding system presented in this work is part of the control architecture developed at the Norwegian University of Science and Technology (NTNU) and discussed in [9]. The speed of the AutoNaut is usually 0-3 knots and is mostly determined by surface currents, waves and wind. As its behavior is mainly determined by the environmental forces, knowledge of these is crucial to the computation of the optimal control command, during a collision scenario.

II. MPC STRATEGY FOR ANTI-COLLISION AND ANTI-GROUNDING

Model Predictive Control (MPC) is a powerful control method that can be used to compute an optimal trajectory based on predictions of the obstacles' motion, and account for their uncertainty. In this method, we formalize risk, hazard, operational constraints and objectives as a cost function in an optimization problem [5]. MPC has in fact been extensively employed for collision avoidance in automotive vehicles [10], underwater and ground vehicles [11], and aircraft traffic control [12].

The main challenges of anti-collision algorithms are related to the convergence and computational complexity of the optimization itself. In fact, complex scenarios may lead to non-convex optimization formulations which are affected by local minima. This, and the fact that real-time implementation requires low computational latency, makes it challenging to implement MPC algorithms for collision avoidance.

Higher performance and low software complexity can be achieved by limiting the optimization process of MPC over a finite number of control behaviors, based on a comparison of the cost, feasibility and risk that they involve [13], [14].

The main objective of the scenario-based MPC (SB-MPC) algorithm is to compute modifications to the desired course (χ_d) and speed (u_d) that lead to a COLREGS-compliant USV trajectory. Since AutoNaut's speed cannot be controlled, this work only considers course offsets. While the obstacle's future motion is predicted as a straight-line trajectory, this formulation focuses on a hazard minimization criterion (i.e., a cost function) that considers dynamic obstacles and COLREGS compliance. Unlike previous works that make use of the same optimization principles [15], [16], this analysis integrates anti-grounding functionalities that include land and static obstacles (e.g., buoys, beacons, islands, permanent or temporary installations). It is demonstrated that the extended algorithm provides efficient anti-collision and anti-grounding functionalities, that can operate at the same time and result in evasive maneuvers to avoid both static and moving obstacles. Finally, the integration of environmental factors allows the algorithm to choose the best control behavior that keeps the

USV far from ground in case of high sea states, minimizing the risk of damaging the vehicle.

This work is based on the algorithm proposed in [5] and implemented in [6], that computes the best control behavior associated to the worst-case hazard for each scenario, where the latter is a combination of own vehicle and predicted dynamic obstacle trajectories.

A. Own USV model

A prediction model of the USV is necessary to generate the trajectories to be evaluated by the cost function. This analysis assumes that the trajectory computed with the kinematic equations only is accurate enough. The kinematic model $\dot{\eta} = R(\chi)v$ is therefore employed [17], where $\eta = (x, y, \chi)$ denotes the position and course over ground in the earth-fixed frame, $R(\chi)$ is the rotation matrix from body-fixed to earth-fixed frame and $v = (v_x, v_y, r)$ denote the velocities in surge, sway and yaw decomposed to BODY-fixed frame. The prediction of the trajectory in a scenario k is obtained by inserting the desired values into the kinematic equation. This model is however very simplistic, since it assumes no drift due to winds and currents (i.e., $\chi = \psi$, where ψ is the heading angle). As discussed in [18], steering of the USV is very much affected by the environmental forces, making this model not suited to represent the true dynamics of the USV in some cases. Nevertheless, the applicability of this model is confirmed by [19], in which both the kinematic equation and the full 3-DOF model were tested and produced only minor differences in the simulation results.

B. Risk factors and collision costs

Essentially, the algorithm computes the best control behavior associated to the worst-case hazard for each scenario, where the latter is a combination of own vehicle and predicted dynamic obstacle trajectories. The cost function indicates the hazard evaluation criterion used in the anti-collision strategy. This work adopts the main components proposed in [5] and [6].

1) *Collision avoidance with dynamic obstacles:* According to [5], the risk factor for collision with obstacle i can be defined as

$$R_i^k(t) = \begin{cases} \frac{1}{|t-t_0|^p} \left(\frac{d_i^{safe}}{d_{0,i}^k(t)} \right)^q, & \text{if } d_{0,i}^k(t) \leq d_i^{safe} \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where t_0 is the current time and $t > t_0$ is the time of prediction. The index k denotes a scenario associated with a single course offset belonging to $\chi_{ca}^k \in [-90^\circ, +90^\circ]$ leading to a trajectory with distance $d_{0,i}^k(t)$ to obstacle i at time t under scenario k . The distance d_i^{safe} and the exponent $q \geq 1$ must be selected large enough to follow COLREGS rules (e.g., COLREGS rule 16 that demands early and substantial give-way actions). In order to prioritize avoiding collisions that are close in time over those that are further into the future, the exponent $p \geq 1/2$ indicates the inverse proportionality to time until occurrence of the event.

Thus, the collision risk factor is higher for events close in time than for events in the more distant future.

The cost associated with collision with an obstacle i is chosen to be

$$C_i^k(t) = K_i^{coll} |\vec{v}_0^k(t) - \vec{v}_i^k(t)|^2, \quad (2)$$

where K_i^{coll} is the cost of collision parameter. The relative velocity of the obstacle is included, in order to minimize the consequences of a collision if a complicated situation would occur, where a collision is unavoidable. \vec{v}_0^k is the predicted velocity of the own vehicle and \vec{v}_i is the predicted velocity of the obstacle with index i in scenario k . We observe that since the own vehicle's ground speed is typically less than 3 knots, the cost associated to a collision is mostly proportional to the obstacle's speed.

2) *Collision avoidance with grounding and static obstacles*: The risk factor for grounding and collision with static obstacles, associated with each course offset k is defined as

$$R_g^k(t) = \begin{cases} \frac{(r^k(t))^q}{|t-t_0|^m}, & \text{if } d_c^k(t) \leq d_k^{safeG} \vee d_{\pm}^k(t) \leq d_k^{safeG} \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

where $r^k(t) = r_c^k(t) + r_+^k(t) + r_-^k(t)$ and $r_c^k(t) = \frac{d^{safeG}}{d_c^k(t)}$, $r_+^k(t) = \frac{d^{safeG}}{d_+^k(t)}$ and $r_-^k(t) = \frac{d^{safeG}}{d_-^k(t)}$. The distance $d_c^k(t)$ is the distance between the own vehicle and the grounding obstacle at time t when own-ship follows the center direction $\chi_{LOS} + \chi_{ca}$, which is in the direction of the course offset that is currently being evaluated in scenario k . The terms d_-^k and d_+^k correspond the distances between the own-ship and land locations found in the -15° and $+15^\circ$ directions, i.e., $\chi_{LOS} + \chi_{ca} \pm 15^\circ$, and their projection onto the desired course path as defined in Figure 2. This is done to ensure safe navigation through narrow passages, keeping enough distance to land on the sides. Here, -15° and $+15^\circ$ are used, since this grounding data already is available for these courses. However, this can be tuned to include a wider range of directions to ensure that the distance to land in the range -90° to $+90^\circ$ is being kept larger than the safe distance to land. For collision avoidance with dynamic obstacles, the exponent factor p is used to prioritize events that are close

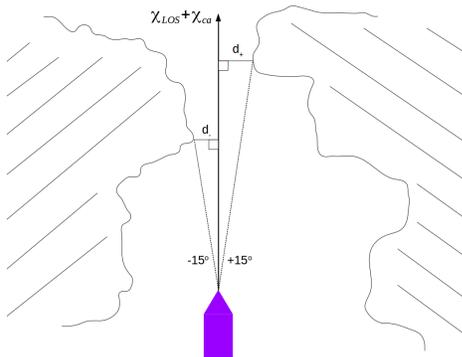


Fig. 2: Definitions of the distances used in the anti-grounding risk function.

in time over those that are further in the future and because there is an uncertainty of where the dynamic obstacle will be located in the future. For anti-grounding, since the obstacles and their locations are static the last argument is no longer valid. On the other hand, the USV will have more time to maneuver when looking further ahead. Nevertheless, the exponent m should have a smaller value for anti-grounding than for collision avoidance.

The cost associated with grounding is chosen to be

$$C_g^k(t) = K_g + K_{env} E^k(t), \quad (4)$$

where K_g is the cost of grounding, $K_{env} = [k_1, k_2, k_3, k_4]$ is a vector containing the weights for each environmental factor. The environmental factors are contained in the vector $E^k(t) = [B, H, W(\chi_{ca}), C(\chi_{ca})]^T$, that will be explained in Section III. The horizontal plane velocities of the obstacle and the own USV are not included, since static obstacles do not move and the ground speed of the USV is low.

C. Hazard evaluation

In summary, the total hazard associated with scenario k at time t_0 is

$$H^k(t_0) = \max_i \max_{t \in D(t_0)} (C_i^k(t) R_i^k(t) + C_g^k(t) R_g^k(t) + \kappa_i \mu_i^k(t) \lambda_i \tau_i^k(t)) + f(\chi_{ca}), \quad (5)$$

where the first and second terms of the cost function are the collision hazard and the grounding hazard, respectively. As described in [5], the term $\kappa_i \mu_i^k(t)$ is the cost of not complying with COLREGS, where κ_i is the tuning parameter and $\mu_i^k \in (0, 1)$ is a binary denoting violations of COLREGS rule 14 or 15. The term $\lambda_i \tau_i^k(t)$ is a COLREGS-transitional cost, where λ_i is a tuning parameter and $\tau_i^k(t) \in (0, 1)$ is described in [6]. Finally, the fifth term is the cost of deviating from the nominal course respectively. We observe that the grounding hazard term $C_g^k(t) R_g^k(t)$ is representing the term $g(\cdot)$ used in [5].

The selected control behavior at time t_0 among the scenarios $k \in 1, 2, \dots, N$ is the one with minimal hazard $H^k(t_0)$

$$k^*(t_0) = \arg \min_k H^k(t_0). \quad (6)$$

During a potential collision scenario, this minimization is done at regular intervals of 5 seconds. In [5] it is emphasized that this optimization is “brute-force” deterministic and guarantees that the global minimum is found after a pre-defined number of cost function evaluations.

III. ENVIRONMENTAL FACTORS FOR COLLISION AVOIDANCE SCENARIOS

Typical scenarios encountered by the AutoNaut are in coastal environments including fjords and archipelagos that may involve multiple dynamic and static obstacles. Situations with reduced maneuverability of the USV might occur depending on the sea state. For this reason, the collision avoidance system needs to be aware of the prevailing environmental conditions in order to let the USV navigate safely.

A. Bathymetry

The bathymetric features might differ greatly between fjords, archipelagos and other coastal areas. In this work, it is assumed that it is associated a higher cost to a steep rocky shelf than a flat sandbank because the USV would most likely suffer greater damage by colliding with the former. Information about the bathymetry and the seabed can be extracted from the ENCs. In the algorithm implementation, the bathymetry is treated as a binary term that indicates whether the shore is safe ($B = 0$, e.g., sandy) or not ($B = 1$, e.g., rocky).

B. Heave displacement

For the AutoNaut, the heave displacement and the wave height are very similar in amplitude. A large displacement in heave is associated to a higher environmental hazard because it indicates rough sea and increases the danger of damage if grounding. The heave displacement and wave height are measured directly in the onboard GNSS and/or IMU. Alternatively, the USV's pitch angle can be used. Similarly, high waves correspond to higher environmental hazards because they make the sea shallower in the troughs, and increase landing force to grounding. In the proposed implementation, the average heave displacement is named H .

C. Wind

Similarly to [20], for a scenario k the wind cost for the static obstacle i is defined as:

$$W_i^k(t) = \frac{V_w(t)^k}{d_{0,i}(t)^k} \max(0, \chi_{ca}^k \cdot \beta_w(t)^k), \quad (7)$$

where V_w^k and β_w^k are the wind speed and direction defined according to [17] and $d_{0,i}^k$ is the distance between the USV. The dot product scales the wind force contribution toward the static obstacles in any orientation around the USV. This means that the risk increases with an obstacle to the east of the vehicle if the wind is coming from the west, etc. The dot product is positive when the angle between the USV-to-obstacle vector and the wind direction vector is less than 90° . Negative dot products are however set to zero, disregarding favorable winds with respect to perceived risks.

D. Sea current

Similarly, the sea current cost C is defined as:

$$C_i^k(t) = \frac{U_c(t)^k}{d_{0,i}(t)^k} \max(0, \chi_{ca}^k \cdot \beta_c(t)^k), \quad (8)$$

where U_c and β_c are the Earth-fixed current velocity and direction.

E. Total environmental cost

The total environmental cost for a course offset χ_{ca}^k is finally defined as:

$$K_{env}E^k = k_1B^2 + k_2H + k_3W(\chi_{ca}^k) + k_4C_O(\chi_{ca}^k). \quad (9)$$

The weights in K_{env} are tuned in simulations with a trial and error approach. Moreover, the weights can be adjusted to make some terms have a greater impact on the total cost than others, mostly depending on the current scenario and onboard measurements. The impact of large values of the bathymetry term and the heave displacement factor is emphasized by adding an exponent. The bathymetry, heave displacement, and wave height factor will give a constant cost for each course offset and will therefore only affect the trade-off between collision or grounding. The wind and current term, on the other hand, are functions of the course offset and will give different costs for different course offsets, depending on the direction of the wind and the current. For this reason, these terms will also affect the choice of the optimal course offset during pure anti-grounding situations (i.e., no surrounding vessels).

IV. HYDROGRAPHIC DATA EXTRACTION AND TRANSFORMATION

The anti-grounding system developed in this work is based on digital charts (ENCs), which are vector-based electronic maps that contain all information necessary to conduct safe navigation at sea. The employed S-57 digital charts is provided by the Norwegian national hydrographic offices [21] for the International Hydrographic Organization (IHO). The S-57 ENC contains several features describing both static objects but also bathymetric information. The DEPARE object, for example, represents the depth of the queried area and is the most useful feature for navigational use, since from that one could infer the coastline and the depth of shallow waters.

To store and use the *a priori* geographic information on the USV, a local SQLite [22] database was created. The SQLite database was chosen because it does not require connection to a server, and because onboard navigation software DUNE [23] contains the necessary libraries. Another benefit is that it's a very light-weight database, since it stores the entire amount of information in a single file.

The two-dimensional grids uses a lot of storage, and searching for the desired area in the data also becomes computationally expensive. The approach of extracting only the vertices of the polygons is explored. This means that the vehicle only gets to know the boundaries of static obstacles or areas present at sea. For the DEPARE object, this means that the software onboard the USV can only observe the depth contours as it explores an area with different depths. This is considered to be enough for implementing anti-grounding functionalities. For further reducing the amount of data handled with this method, just including the depth ranges that are interesting can be done. For the purpose of anti-grounding, depths above a certain threshold can be ignored, because the USV is only interested to know if it is heading towards a shallow area.

V. ARCHITECTURE DESIGN

Figure 3 shows the designed software architecture. A navigation plan is commanded from shore based on

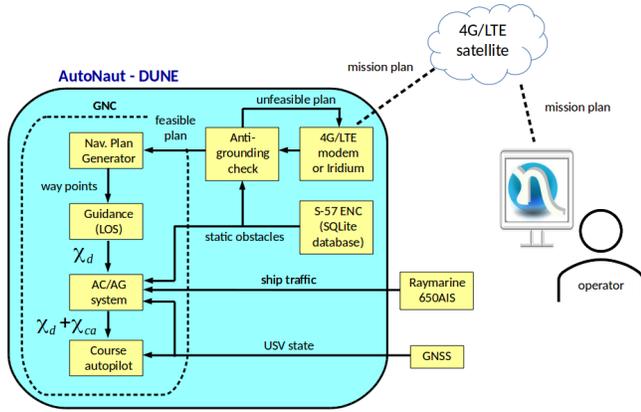


Fig. 3: Designed software architecture including anti-collision (AC) and anti-grounding (AG) functionalities.

the available communication link, is received onboard the USV and a dedicated software digests it. A plan is infeasible if, for example, the path that connects multiple desired locations crosses land, islands, or static obstacles contained in the database. Also, a navigation plan is considered infeasible if the commanded path makes the USV navigate too close to static obstacles. As it navigates, the software will periodically assess the mission safety, by computing the USV's position with respect to the queried digital information. The periodical queries extract database information in a surrounding of the USV, so new information is provided as the USV moves. This is essential, since the navigation performances of the AutoNaut are highly dependent on the sea state and, environmental forces may cause the USV to drift from the original path which was initially considered safe. This means that a mission can initially be evaluated as safe and, during its execution dangers appear and the plan is not safe anymore. Figures 4 shows the initial evaluation of mission plan commanded from shore. The plan involves three way points around the Munkholmen island (Trondheim Fjord), starting from the initial USV's location. Bathymetric information is displayed in the form of point clouds whose color is defined based on the depth of the seafloor in the considered location.

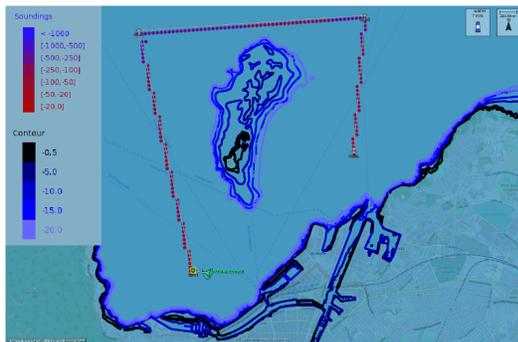


Fig. 4: Point clouds of Trondheim's coastline, depth contours around Munkholmen and depth of the commanded path.

Moreover, Figure 4 also shows that the software evaluates the safety of a commanded navigation plan by querying the depth of the path connecting the target locations. By checking the depth soundings along a commanded path, the software is able to halt the execution of a mission in which the target destinations or the paths that connect them are not in safe, navigable waters.

The results presented in this manuscript are a combination of field tests and simulations. In the first section below, pure anti-collision functionalities are presented and discussed. Then, the anti-grounding system is evaluated with a set of simulations. Finally, anti-collision and anti-grounding functionalities are tested together and the results are discussed.

VI. EXPERIMENTAL RESULTS: ANTI-COLLISION WITH DYNAMIC OBSTACLES

The collision avoidance algorithm was tested for the first time on the field in December 2019 in Børsla, 20 km southwest of Trondheim (Central Norway). The experiment lasted two consecutive days. On the first day, some virtual obstacle vessels were simulated on the operator's laptop onshore and the local network created on the laptop was shared with the USV. Doing so, the AutoNaut was perceiving the virtual obstacles as if they were actually real vehicles navigating at sea. An instance of DUNE is run on the operators' laptop, where vehicles acting as obstacles for the collision scenarios are simulated. The operators are in this way capable of simulating motored vessels. The simulated vessels are set to transmit a fake AIS message over the closed network at regular intervals, in way that emulates actual vehicles at sea. The simulated message contains both their static and dynamic information, usually encapsulated in AIS messages 1, 2, 3 and 5. Whereas dynamic information (types 1, 2, 3 messages) is used by the collision avoidance algorithm to predict the ship's state and trajectory, the notion of its size (type 5 messages) is used to let the AutoNaut determine the appropriate clearance.

The onboard software monitors the vessels transmitting AIS messages in the 5000 meters surrounding the AutoNaut. When a potential collision scenario is identified, the collision avoidance algorithm runs at an interval of 5 seconds. This is considered enough, given the slow dynamics of the USV. However, if the distance between the AutoNaut and one or more obstacles drops below a defined safety threshold, this interval is reduced.

Table I contains the parameters that are used in this initial test, which happened on 12th of December 2019.

The obstacle's and AutoNaut's states are evaluated with a time step $T = 5$ s over a prediction horizon H of 600 seconds. The software remembers the vessels in the d^{max} range up to $T_{max} = 240$ s, period after which the potential obstacles which do not update their state via AIS are discarded from the monitored list. The algorithm considers an obstacle as a potential threat when its distance to the AutoNaut drops below $d^{sbmpc} = 700$ m, meaning that

	Symbol	Value AC	Value AG
Prediction horizon	H	600 s	600 s
Time step	T	5 s	60 s
Max. obstacle surveillance range	d^{max}	5000 m	5000 m
Disappeared obstacle	T_{max}	240 s	/
SB-MPC surveillance range	d^{sbmpc}	700 m	/
Minimal safe distance to obstacle	d^{safe}	300 m	100 m
Minimum safe depth	H^{safe}	/	5 m

TABLE I: Anti-collision (AC) and anti-grounding (AG) algorithm parameters.



Fig. 5: Head-on scenario with a moving obstacle.

action will be taken within this range if a potential collision is predicted. This range was chosen given the limited width of the fjord in the test area. A larger range should be used when the USV operates autonomously in open waters: given its slow dynamics it is desirable to perform safety maneuvers well in advance.

The first presented field test is a simple head-on scenario in which the obstacle is moving towards the AutoNaut. In this experiment, the simulated obstacle has a ground speed of approximately 1.3 m/s and a course of 118° , as depicted in Figure 5. When the distance between the two vehicles drops below 700 meters, the collision avoidance algorithm commands a positive course offset $\chi_{ca} = +60^\circ$. The new desired course becomes $\chi_d = -2^\circ$. It can be noticed that a larger course offset is commanded, since COLREGS rule 14 state that in a head-on scenario a vehicle is expected to turn starboard in order to avoid the other one. Once the AutoNaut is located on the port side of the obstacle the offset is set to $\chi_{ca} = 0^\circ$ since the algorithm expects that also the other vessel is taking the necessary action. However, the obstacle does not comply with the COLREGS regulations and keeps a constant speed and course. The algorithm computes therefore a 90° course offset in order to avoid the collision and try keeping the desired minimal safety clearance to the obstacle. The minimal distance between the USV and the obstacle is approximately 200 meters. Once the obstacle has passed, the AutoNaut resumes nominal navigation towards the desired destination.

In the second experiment, the obstacle is in the port ahead sector (see Figure 6), i.e., the obstacle see the AutoNaut on its starboard side. COLREGS rule 15 states that in a crossing situation with risk of collision, if one vessel can see another vessel on its starboard side, the former has to give way. In

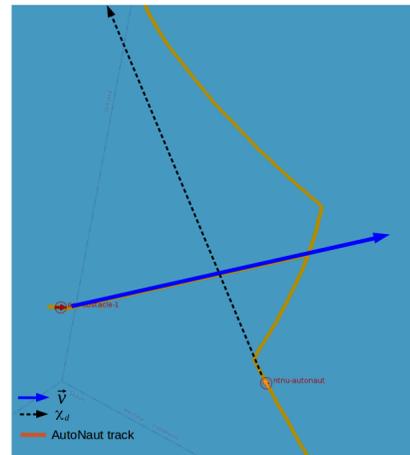


Fig. 6: Crossing from starboard scenario with a moving obstacle.

the same situation, the vessel that has the other one on its port side is the stand-on vessel. In other words, the simulated obstacle is expected to take a starboard turn in order to avoid colliding with the AutoNaut. In the test, this does not happen, and when the distance between the two vehicles drops below d^{sbmpc} , the algorithm opts for a course offset $\chi_{ca} = +45^\circ$ as shown in Figure 6. While the AutoNaut keeps navigating with the new course $\chi_d = 20^\circ$, the obstacle does not take any action to avoid collision and the collision avoidance algorithm is forced to keep the same course offset in order to avoid collision and respect the desired minimal distance. When the obstacle has passed the AutoNaut, the offset is zeroed and the USV resumes line-of-sight navigation towards the desired location. The minimum distance between the two vehicles is 330 meters.

On the second day of field trials the operational setup was further modified. In this occasion, a 5.6 m long motorboat (named Buster) was employed to act as an obstacle. Since this boat is not equipped with an AIS system, it was decided to create again a fake message and run this in the laptop's DUNE instance. Unlike the previous experiments, this time the telemetry of the simulated obstacles was obtained by the laptop's GPS and, the operator with the laptop was passenger on the motorboat. In this way, as the boat was moving on the sea, the laptop's GPS was providing telemetry data to DUNE, which was emulating the AIS on a vessel and sharing that with the AutoNaut over the closed network. In other words, the AIS of the motorboat was emulated through DUNE and the motorboat itself was acting as a vessel transmitting AIS messages. The advantage of using this setup is that the scenario can be more flexible, i.e., it is possible to steer the Buster in order to change the scenario. The first tested scenario involves two head-on situations, in which the Buster heads towards the AutoNaut. Figure 7 shows the first situation. Again, the obstacle does not comply with COLREGS regulations and keeps a constant course of approximately 115° and speed of 3.5 m/s. This time the algorithm decides to disregard COLREGS rules as

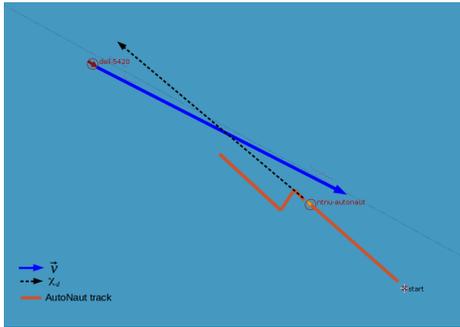


Fig. 7: Head-on scenario involving the Buster.

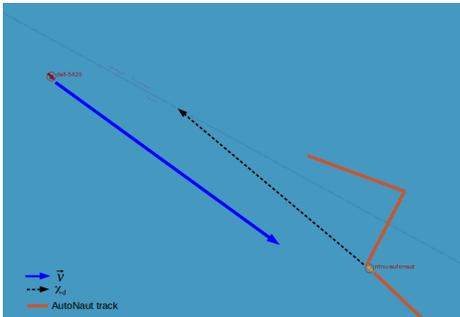


Fig. 8: Head-on scenario involving the Buster.

a starboard turn is likely to lead to a collision. The decided offset is therefore $\chi_{ca} = -90^\circ$. The minimal distance between the vehicles during this scenario is approximately 200 meters, which is the maximum clearance that the AutoNaut can impose between the two vehicles given their velocities and the provided course offsets. Once the obstacle has passed over, the USV resumes its straight-line navigation towards the desired destination. Few minutes after, the Buster is driven back to its initial location and a new head-on scenario was initiated. Figure 8 shows the second consecutive head-on situation. This time the AutoNaut finds itself on the port side of the obstacle's frame, making it easier for the algorithm to choose a positive course offset.

VII. SIMULATION RESULTS

In this section we discuss two scenarios. In the first we show how the MPC-based collision avoidance algorithm is used as a simple anti-grounding system. In the second scenario we show instead how the information retrieved from digital charts is used to achieve a combined anti-collision and anti-grounding system, in which the relative position of land is taken into account during the optimization process.

A. ENC-based anti-grounding

In this section, the results of the anti-grounding system are presented. To date, anti-grounding functionalities are only tested in simulation whose objective is to assess the feasibility and limitations of the proposed method. The anti-grounding system checks for land or static obstacles every $T_g = 60$ s, given the low speed of the USV. In this implementation, the S-57 object DEPARE (depth area) is

the only object considered since from this it is possible to retrieve the coastline as well as areas with shallow waters. Despite the database also contains information about other static objects at sea, in this analysis the main focus is the depth contours. When the system scans the USV's surroundings, a query is made to the database containing the ENC data point clouds, asking for the depth of all the data points located within a distance of $d^{max} = 5000$ m from the AutoNaut, whose depth is within a specified range. For this reason, a minimum safe depth value parameter (H^{safe}) is defined (see Table I).

Then, the algorithm iterates through the retrieved information in order to find the point that is the closest to the AutoNaut for each of the directions corresponding to the course offsets. This results in a new vector containing the closest data point in each course offset direction. Each point retrieved from the point-cloud database contains the geographic location and depth of the seafloor.

In this implementation, the SB-MPC algorithm is used for anti-grounding functionalities only, treating the land information as an obstacle that does not move. The optimization is then run as if the obstacles were moving (with zero speed). In other words, exactly as before, the algorithm iterates through all the possible course offsets and chooses the one corresponding to the lowest hazard, based on the associated distance to land along the considered offset. This is achieved by computing the risk and cost of grounding for each course offset. The chosen optimal course offset is added to the course angle from the LOS guidance law, and the modified course reference is sent to the autopilot.

In the first scenario, the AutoNaut moves straight towards the Munkholmen island with a course of approximately 90° . Figures 9a-9d show the evolution of the hazard as a function of distance and of the environmental factors. Different values for wind and current velocity and direction are tested, while the remaining terms are set to constant values: bathymetry $B = 1$, heave $H = 2$ m. The given wind and current directions are the absolute directions, and the course of the AutoNaut is kept at 87° in all simulations. The plots in Figure 9a and 9b show that when the wind and current are directed towards land, the hazard is higher. Figure 9b shows in fact that the hazard still increases, since the USV keeps moving towards the island, but it does not grow high since wind and sea current try to push the vehicle towards West. Similarly, in Figure 9c, the wind and current speeds are lower than in Figure 9b, but still the hazard is higher since the direction of wind and current push the AutoNaut towards Munkholmen. In Figure 9d, the wind and current come from the front sides of the vehicle, resulting in a smaller hazard than in Figure 9a, but a similar hazard as in Figure 9c, and higher than in Figure 9b. Note that the hazard has a magnitude in the size of 10^5 in all of the four plots shown here, confirming that the environmental factors can have a large impact on the perceived grounding hazard.

The behavior shown through these plots validates

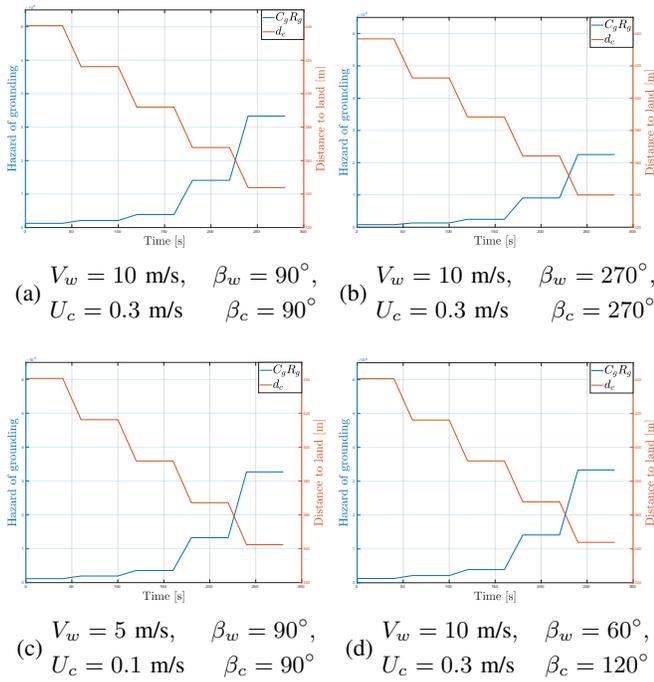


Fig. 9: Plots of the grounding hazard and distance to land with environmental factors added, when the AutoNaut moves towards land. Different wind and current speeds and directions are tested.

that the anti-grounding system behaves according to the theory presented. When the AutoNaut approaches land, the grounding hazard increases significantly. Adding the environmental factors results in a considerable increase in the grounding hazard, depending on the velocity and direction of wind and currents. A higher velocity leads to a larger hazard. Moreover, when the wind and current are directed towards land, the hazard increases.

The anti-grounding system is tested in simulation. In the first simulation, the USV is commanded to a mission that involves crossing a portion of land in order to reach the target destination. If the anti-grounding system is not activated, the vehicle would just hit land (white dashed arrow in Figure 10). Figure 10 shows the depth information extracted by the algorithm. The distance to land in the direction of the course offsets ($\chi + \chi_{ca}$) is provided to SB-MPC, which selects the appropriate offset needed to avoid grounding. The minimum safe distance to land is set to 100 meters, since the way point is located in a bay. The objective is that AutoNaut reaches it without getting closer to land than the minimum safe distance indicates. Instead, the minimum safe depth is set to 10 meters, meaning that the AutoNaut considers this depth as ground and wants to keep the minimum safe distance to this depth contour. The 10 meters depth contour is the dark blue contour shown in Figure 10.

It is important to emphasize that the anti-grounding system is a part of the reactive obstacle avoidance control system and is not intended to be used as a path planning system.

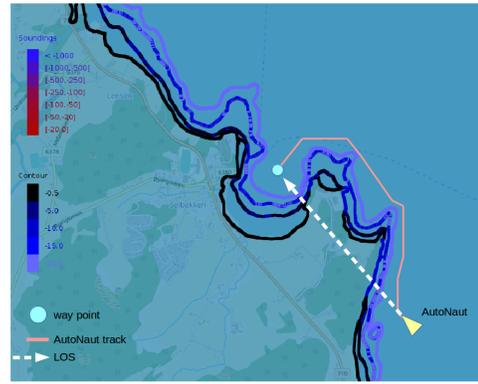


Fig. 10: The AutoNaut avoids land by keeping a safety distance from the coastline.

Therefore, it does not aim at finding the optimal path to a way point. Even though the anti-grounding system manages to find a good solution in most of the situations, it cannot be guaranteed to work optimally. Because of its relatively short time horizon, it can get stuck in narrow passages and complex areas and not be able to find the optimal path for a longer mission.

B. Combined anti-collision and anti-grounding

In this last section, the combination of both systems is evaluated and discussed with simulations. The objective is to test the proposed system when the AutoNaut operates close to land, when environmental conditions are not favorable and other vessels are present in the vehicle's surroundings. In such situations, the algorithm should balance the risk of grounding with that of colliding with a moving vessel. The tuning of the costs are essential to guide the algorithm towards the desired behaviour. Additionally, the knowledge of sea state plays a main role in the control decision since grounding may be cause more or less damages depending on the state of the sea.

Figures 11 and 12 show a head-on scenario in which a simulated obstacle navigates towards the AutoNaut, without complying with COLREGS rules, at a constant speed of 13 m/s. When the anti-grounding system is deactivated (Figure 11), the algorithm chooses a 90° starboard turn, since this is the preferred COLREGS direction as stated in rule 14. The shortest distance between the two vehicles is 142 meters, while the shortest distance between land and the AutoNaut is about 65 meters. In this head-on scenario, the obstacle maintains a high speed and approaches the AutoNaut straight ahead, which means that the AutoNaut has too little time to react, and cannot manage to keep the minimal safety distance (300 meters), although it turns 90° starboard. When the anti-grounding system is enabled (Figure 12), the AutoNaut turns to port instead, to avoid getting closer to land, at the same time as it avoids collision with the obstacle. The shortest distance between the two vehicles this time is 202 meters, and the shortest distance to land is about 240 meters, which explains why the AutoNaut does not choose to turn starboard. In this situation, the priority of avoiding grounding in com-



Fig. 11: Head-on without anti-grounding activated.



Fig. 12: Head-on with anti-grounding activated.



Fig. 13: Head-on with anti-grounding activated.

pliance with COLREGS rule 18 (responsibilities between vessels) over complying with COLREGS rule 14 (head-on).

The same simulation was repeated with a lower obstacle speed of 6 m/s. Figure 13 shows the same exact head-on simulation with the only difference being the fact that the algorithm is aware of land. It can be observed that, knowing that land is very close on the USV's starboard side, the algorithm chooses a smaller offset ($\chi_{ca} = +45^\circ$). Then in order to respect the minimal safety distance to the other vessel, the algorithm commands a $+90^\circ$ offset for a short period, before the offset is zeroed and normal navigation is restored once the obstacle has passed. Despite land is close to the USV, the algorithm chooses a starboard turn, since the obstacle's speed allows the AutoNaut to comply with COLREGS rules, while avoiding collision safely and navigate at a minimum desired distance from land. Despite the minimal distance between the two vehicles cannot be respected (approximately 150 meters), the minimum safety distance to land ($d_g^{safe} = 100$ m) is respected since the measured minimum distance is 120 meters.

Including the environmental state: In this section, the environmental factors and their additional cost $K_{env}E^k$ are added in the simulations. The simulated scenario is the same

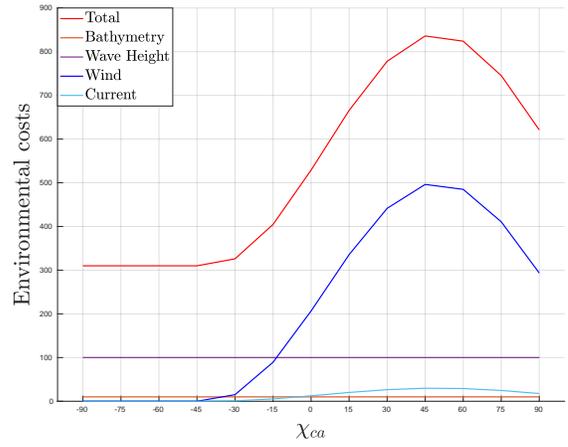


Fig. 14: The evolution of each cost depending on the course offset.

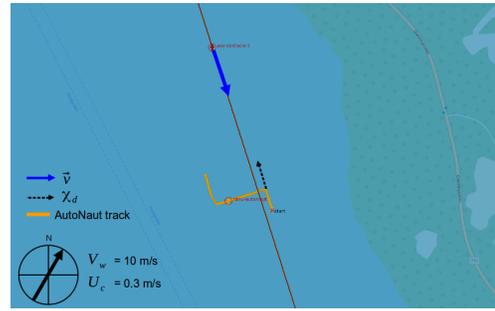


Fig. 15: Head-on with anti-grounding activated and sea state included.

as in the previous section and the speed of the obstacle is kept constant at 6 m/s. The environmental state is included, in order to understand if the environmental factors have an impact and, most importantly, the desired impact on the decisions of the collision avoidance and anti-grounding algorithm. In rough conditions, it is expected that the USV prioritizes avoiding grounding obstacles more than before and is careful about going close to land. In order to stress the importance of the sea state in the control decision, high environmental costs are considered. A change in behavior demands quite a high additional cost, which is desired since complying with COLREGS is important to ensure safe navigation at sea and should be the standard behavior. The total environmental cost can be tuned by adjusting the weights in the K_{env} factor. The values of the environmental factors used in this scenario are defined as: $[B, H, V_w, \beta_w, U_c, \beta_c] = [1, 2, 10, 30^\circ, 0.3, 30^\circ]$. The weights and the grounding cost are defined as $K_{env} = [10, 50, 50, 100]$ and $K_g = 100$ respectively. The values of the factors are set to create rough environmental conditions. The plot in Figure 14 shows the cost of each environmental term together with the resulting total environmental cost $K_{env}E^k$ for each course offset. As intended by design, the cost is highest for the course offsets that coincide with the wind and current direction. The costs from the sea current and the bathymetry

are quite low compared to the cost from wind and heave motion. This can be changed by adjusting the weights in the K_{env} vector.

Figure 15 shows that despite the obstacle speed is low enough to allow the USV to comply with COLREGS rule 14, the algorithm opts for a port maneuver. This is due to the wind and current that push the AutoNaut towards land on its starboard side. It can be noticed that the AutoNaut turns -90° to port, instead of turning $+45^\circ$ to starboard where it was considered safe to go in the previous simulation when the environmental state was not included in the optimization (see Figure 13). Because there is a quite strong wind and a current directed North-East, in addition to high waves, the total environmental cost is so high that keeping a safe distance to land is prioritized over choosing the COLREGS compliant behavior. Still, the AutoNaut manages to avoid collision with the obstacle and, at the same time, a safe distance from land is kept on its starboard side. The shortest distance to the obstacle is 118 meters, which is only a few meters shorter than before (Figure 13), while the shortest distance to land has increased to more than 200 meters. This result shows that in rough environmental conditions, the system will prioritize avoiding grounding obstacles, and that the environmental factors can have an impact on the decisions of the AutoNaut.

VIII. CONCLUSIONS

The automatic anti-collision and anti-grounding system presented in this chapter is a reactive obstacle avoidance control system intended to make the AutoNaut aware of, and avoid, the surrounding moving and static obstacles in situations where it has to diverge from its originally planned path. The simulations and experimental results presented in this paper show that the performance of the collision and grounding avoidance system can be tuned as desired. Furthermore, the full system functionalities should be further tested in the field in order to assess the validity of the theoretical assumptions.

A future work is to include the nonlinear dynamic model of the USV and use that to update the vehicle's state when winds and currents affect its navigation [18].

IX. ACKNOWLEDGMENTS

The authors would like to thank Inger Hagen for the help with software implementation and field testing. This work was supported by the Research Council of Norway (RCN) through the MASSIVE project, grant number 270959, and AMOS grant number 223254 to NTNU.

REFERENCES

- [1] International Hydrographic Bureau Monaco, "IHO Transfer Standard for Digital Hydrographic Data - Special Publication No. 57. 3.1," 11 2000.
- [2] M. Candeloro, A. M. Lekkas, and A. J. Sørensen, "A voronoi-diagram-based dynamic path-planning system for underactuated marine vessels," *Control Engineering Practice*, vol. 61, pp. 41 – 54, 2017. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0967066117300072>
- [3] Dongwoo Kang, Jungsik Jung, Sewoong Oh, and Sunyoung Kim, "Automatic route checking method using post-processing for the measured water depth," in *2015 International Association of Institutes of Navigation World Congress (IAIN)*, 2015, pp. 1–5.
- [4] S. Reed and V. E. Schmidt, "Providing nautical chart awareness to autonomous surface vessel operations," in *OCEANS 2016 MTS/IEEE Monterey*, 2016, pp. 1–8.
- [5] T. A. Johansen, T. Perez, and A. Cristofaro, "Ship collision avoidance and colregs compliance using simulation-based control behavior selection with predictive hazard assessment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, pp. 3407–3422, 2016.
- [6] I. B. Hagen, D. K. M. Kufoalor, E. F. Brekke, and T. A. Johansen, "Mpc-based collision avoidance strategy for existing marine vessel guidance systems," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 7618–7623.
- [7] D. K. M. Kufoalor, T. A. Johansen, E. F. Brekke, A. Hepsø, and K. Trnka, "Autonomous maritime collision avoidance: Field verification of autonomous surface vehicle behavior in challenging scenarios," *Journal of Field Robotics*, vol. 37, no. 3, pp. 387–403, 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rob.21919>
- [8] P. Johnston and M. Poole, "Marine surveillance capabilities of the autonaut wave-propelled unmanned surface vessel (usv)," in *OCEANS 2017 - Aberdeen*, June 2017, pp. 1–46.
- [9] A. Dallolio, B. Agdal, A. Zolich, J. A. Alfredsen, and T. A. Johansen, "Long-endurance green energy autonomous surface vehicle control architecture," *OCEANS 2019*, Seattle, Washington, accepted.
- [10] T. Shim, G. Adireddy, and H. Yuan, "Autonomous vehicle collision avoidance system using path planning and model-predictive-control-based active front steering and wheel torque control," *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering*, vol. 226, no. 6, pp. 767–778, 2012. [Online]. Available: <https://doi.org/10.1177/0954407011430275>
- [11] C. Caldwell, D. Dunlap, and E. Collins, "Motion planning for an autonomous underwater vehicle via sampling based model predictive control," 09 2010, pp. 1–6.
- [12] K. Bousson, "Model predictive control approach to global air collision avoidance," *Aircraft Engineering and Aerospace Technology*, vol. 80, pp. 605–612, 10 2008.
- [13] A. Bemporad and M. Morari, *Robust model predictive control: A survey*, 10 2007, vol. 245, pp. 207–226.
- [14] L. Chisci, J. Rossiter, and G. Zappa, "Systems with persistent disturbances: Predictive control with restricted constraints," *Automatica*, vol. 37, pp. 1019–1028, 07 2001.
- [15] E. Brekke, E. Wilthil, B.-O. H. Eriksen, D. K. Kufoalor, Ø. K. Helgesen, I. B. Hagen, M. Breivik, and T. Johansen, "The autosea project: Developing closed-loop target tracking and collision avoidance systems," 2019.
- [16] B.-O. H. Eriksen, G. Bitar, M. Breivik, and A. Lekkas, "Hybrid collision avoidance for asvs compliant with colregs rules 8 and 13–17," *Frontiers in Robotics and AI*, vol. 7, 2020.
- [17] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control, 2nd Edition*. Wiley, 2021.
- [18] A. Dallolio, H. Øveraas, J. A. Alfredsen, T. I. Fossen, and T. A. Johansen, "Design and Validation of a Course Control System for a Wave-Propelled Unmanned Surface Vehicle," 2021, Field Robotics (in press).
- [19] I. B. Hagen, "Collision avoidance for ASVs using model predictive control," Master's thesis, Trondheim, NTNU, 2017.
- [20] S. Blindheim, S. Gros, and T. A. Johansen, "Risk-based model predictive control for autonomous ship emergency management," *IFAC-PapersOnLine*, vol. 53, no. 2, pp. 14 524–14 531, 2020, 21th IFAC World Congress.
- [21] Kartverket, "Elektroniske sjøkart (ENC)." [Online]. Available: <https://www.kartverket.no/til-sjos/kart/elektroniske-sjokart-enc>
- [22] C. Newman, *SQLite*, ser. Developer's library. Sams, 2005. [Online]. Available: <https://books.google.it/books?id=Hx0iAQAAIAAJ>
- [23] J. Pinto, P. S. Dias, R. Martins, J. Fortuna, E. R. B. Marques, and J. B. de Sousa, "The lts toolchain for networked vehicle systems," *2013 MTS/IEEE OCEANS - Bergen*, p. 1–9, jun 2013. [Online]. Available: <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6608148>