Control Engineering Practice 61 (2017) 41–54

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

A Voronoi-diagram-based dynamic path-planning system for underactuated marine vessels☆

Mauro Candeloro[a,b,*], Anastasios M. Lekkas[a], Asgeir J. Sørensen[a,b]

[a] Department of Marine Technology, NTNU, Trondheim, Norway
[b] Centre for Autonomous Marine Operations and Systems, NTNU, Trondheim, Norway

ARTICLE INFO

Keywords:
Path-planning
Underactuated vessels
Replanning
Navigation support systems

ABSTRACT

The main contribution of this paper is the development of a rapid, dynamic path-planning system for 3-DOF marine surface vessels navigating in environments where other marine vehicles might be operating too. The method is based on the Voronoi diagram and generates the initial path while ensuring that clearance constraints are satisfied with respect to both land and shallow waters. Fermat's Spiral (FS) segments are used to connect successive straight lines, hence, resulting in curvature-continuous paths that are rapidly computed. When a ship is detected, the range of its position during a given time frame is estimated, and the path-planning system produces in real time a new safe and smooth path. The International Regulations for Preventing Collisions at Sea (COLREG) are taken into account in the replanning procedure. An indirect adaptive Line-Of-Sight (LOS) guidance algorithm from the existing literature is implemented to ensure the underactuated vessel will counteract the effects of unknown environmental forces, such as ocean currents, while converging to the safe path. Simulations show the effectiveness of the proposed approach.

1. Introduction

Considerable progress has been achieved over the last years in the field of autonomous marine vehicles and systems, with unquestionably visible positive results for a large number of applications within the ocean sector. The works by Seto (2013), Ludvigsen and Sørensen (2016), and Marino, Antonelli, Aguiar, Pascoal, and Chiaverini (2015) present an overview of advances of autonomy for marine applications, a new autonomous architecture for marine vehicles, and autonomous strategies for coordinated operations of marine robots, respectively. Increased autonomy can provide great added value in several contexts, with two of the most important ones being increased efficiency and safety of operations. Despite the efforts to increase the autonomy level of marine operations, many open issues are still to be addressed in order to make the current systems safe and efficient enough for their actual employment, especially when interaction with humans or other assets is expected. In the case of motion planning for marine vessels, route planning and replanning (as response to unexpected changes in the environment) are two important challenges that need to be tackled. These tasks become rather complex when there are dynamic elements in the environment, uncertainties in the sensors data, and safety requirements to satisfy. Planning of operations is underlined by many authors as a substantial limitation that prevents systems from reaching a higher autonomous level. According to Watson and Scheidt (2005) "*automation of planning processes has been a central problem in the field of artificial intelligence for more than 30 years [...]*". As a matter of fact, autonomous surface vessels navigating on sea, or autonomous passenger flights are still not allowed. However, new initiatives on autonomous ships are currently being taken forward, with the establishment of a part of the Trondheim Fjord, Norway, as test site for new systems, Kongsberg (2016).

Extensive research pertaining to path-planning and obstacle avoidance has been conducted in the last years. The work by Tsourdos, White, and Shanmugavel (2010) focuses on the particular challenges of cooperative path planning for UAVs, whereas LaValle (2006) and Choset et al. (2005) present a wider and more generic scope from the computer science perspective. The work by Zeng et al. (2015) gives a literature review w.r.t. to Autonomous Underwater Vehicles (AUVs), Caiti (2014) presents a complete and recent review of motion planning algorithms for marine control systems, and Antonelli, Chiaverini, Finotello, and Schiavon (2001) present a practical application of dynamic path planning and obstacle avoidance for an AUV. One classification of path-planning methods distinguishes between *global* and *local* path-planning algorithms. Global path-planning algorithms

☆ A rudimentary version of this paper was presented at the 9th IFAC Conference on Control Applications in Marine Systems, which was held in Osaka, Japan, during September 2013. This paper was recommended for publication in revised and extended form by Associate Editor "Kazuhiko Hasegawa".

* Corresponding author at: Department of Marine Technology, NTNU, Trondheim, Norway.
E-mail address: mauro.candeloro@ntnu.no (M. Candeloro).

http://dx.doi.org/10.1016/j.conengprac.2017.01.007
Received 21 May 2016; Received in revised form 27 November 2016; Accepted 15 January 2017
Available online 06 February 2017

evaluate the whole information available on a certain area in order to generate an obstacle-free path between the departure location, or initial waypoint, and the destination, or final waypoint. The generated path must satisfy certain constraints, such as the dynamic constraints of the vehicle and environment-related constraints (weather conditions, and minimum allowed distance from obstacles, for instance). A large number of methods and tools addressing these challenges have been developed in the path planning literature. A few notable examples are the rapidly-exploring random trees (Lavalle, 1998), the Voronoi Diagram (VD) (Aurenhammer, 1991), the visibility (Lozano-Pérez & Wesley, 1979), the potential field method (Khatib, 1985; Lee, Kwon, & Joh, 2004), the occupancy grid methods (Elfes, 1987), and optimization methods, such as Semi-Lagrangian methods (Falcone & Ferretti, 2013), pseudospectral optimal control (Lekkas, Roald, & Breivik, 2016) and the A* algorithm (Dolgov & Thrun, 2010; Hart, Nilsson, & Raphael, 1968; Larson, Bruch, & Ebken, 2006, 2007; Naeem, Irwin, & Yang, 2012). Local path-planning algorithms, on the other hand, take into account the surrounding space of the vehicle and are able to respond to unpredicted factors, such as unmapped obstacles etc. A few examples of local methods are the dynamic window (Fox, Burgard, & Thrun, 1997), the virtual force field (Borenstein & Koren, 1991) and its extension, the modified virtual force field (Lee et al., 2004), and the velocity obstacle method (Fiorini & Shiller, 1993). Global and local algorithms often complement each other.

Dynamic collision avoidance (CA) for marine vehicles has been widely investigated in literature. Statheros, Howells, and Maier (2008) presents a state-of-art report describing the mathematical methods utilized to model marine vehicles, the most common CA algorithms, and the navigation systems employed in CA scenarios. Campbell, Naeem, and Irwin (2012) present a review of intelligent USV collision avoidance research in terms of control, path planning and collision avoidance architecture with regards to COLREGs incorporation. Other examples of recent works on CA are the following: Naeem et al. (2012), uses the Velocity Obstacle methods to provide a CA system which respects the COLREGS (IMO, 1977). Perera, Ferrari, Santos, Hinostroza, and Soares (2015) presents a fuzzy-based CA system with experimental results on a scaled ship model. Woerner (2016) considers the problem of collision avoidance for marine vehicles by extending the traditional obstacle velocity into a multi-threshold based approach in order to represent and evaluate human ship driving practices more realistically. The current paper develops a VD-based dynamic path-planning and CA system, which incorporates the benefits of local and global approaches, as in van den Berg, Ferguson, and Kuffner (2006).

Due to its simplicity and low computational cost, the Voronoi diagram has been used consistently for path-planning purposes, including aerial operations (Bellingham, Tillerson, Alighanbari, & How, 2002; Bortoff, 2000; Chandler, Rasmussen, & Pachter, 2000) and marine operations. More specifically, Christopher Gold and co-workers have advocated the use of VDs (static and dynamic) for collision avoidance in marine applications (Gold, 1998, 2016; Gold, Chau, Dzieszko, & Goralski, 2005; Goralski & Gold, 2007). However, none of the aforementioned works give any details regarding the technical aspects of such an approach. In Marino, Antonelli, Aguiar, and Pascoal (2012), the authors addressed a harbour-patrolling problem and exploited the advantages of Voronoi tessellation's to automatically distribute a number of marine vehicles over the environment. A similar application was considered by Antonelli, Chiaverini, and Marino (2012). In these works, the VD edges were not used to assist generate the path of a vehicle, but to determine the locations of the marine vehicles so as to achieve full coverage of an area.

In Candeloro, Lekkas, Sørensen, and Fossen (2013) we considered a static environment and the first step was to generate a roadmap using the VD. Then a number of heuristic techniques was used in order to produce a set of successive waypoints in between the initial and final waypoints. All the successive waypoints were then connected using paths consisting of straight lines and Fermat's Spiral (FS) segments. The final path was curvature-continuous and the whole process had a low computational cost, thus making it a good candidate for online implementation.

In this paper, we consider a dynamic 2-D environment and the Voronoi diagram is used in both the path-planning and replanning phase, where additional heuristics allow to speed up the process and create flyable and intuitive paths. In the first phase a map is supposed to be known, while in the second phase the path is replanned locally on the basis of real-time sensor information. This paper extends the work by Candeloro et al. (2013) through the following contributions:

- Depth constraints are taken into account in order to avoid the problem of grounding during operations in shallow (for a given vehicle) waters.[1] This makes it possible to exploit in a more useful manner the information included in the nautical charts and adjust the method to a wide number of marine vehicles.
- A novel methodology for path following of paths with FS segments is developed to test the path-planning and replanning algorithms in the dynamic case, thus adding the time variable to the simulated case. In addition, the influence of unknown ocean currents is taken into account.
- A fast, VD-based replanning strategy, which is based on the same principles and procedure of the initial path-planning algorithm, is implemented in a modular way.
- The replanning algorithm generates alternative local paths which are compliant with the COLREGs. When an obstacle is detected, it is treated as a local disturbance and a deviation from the original path is computed. The new local path takes into account the COLREGS and the current dynamic constraints (vehicle velocity and estimated motion of the dynamic obstacle) so as to ensure collision avoidance and reconnection to the original path.

In Candeloro, Lekkas, Hegde, and Sørensen (2016), this work is further extended for the underwater domain, thus demonstrating the ability of our approach to tackle planning and replanning in the three-dimensional case. Due to the complexity of the problem though, Candeloro et al. (2016) considered only the geometrical aspects of the problem without taking into account temporal constraints, an aspect which is addressed in detail in the current paper.

The paper is structured as follows: Section 2 describes the maps and the preprocessing of the environmental data. Section 3 describes the path-planning algorithm. In Section 4, the path following algorithm is developed. Section 5 presents the motion control objective. Section 6 presents the simulated vessel model. Section 7 focuses on the LOS guidance algorithm. 8 defines the characteristics on the unknown obstacles and describes the obstacle tracking algorithm. 9 presents the on-line path replanning system and the actions that are taken when an obstacle is encountered. Section 10 presents the case study and simulation results. Finally, Section 11 states the conclusions and further work.

2. Geographical data sources

The data used in this paper is obtained from *Global Administrative Areas, GADM* (which is developed by the Universities of California, Berkeley and Davis and the International Rice Research Institute).[2] These maps come in vectorial and raster file formats. The *shapefile* (.shp) is a popular geospatial vector data format used by many Geographic Information System (GIS) program packages. These data need to be manipulated in a certain way to be used into the proposed planning software. In particular:

[1] In this article the term "*grounding*" refers to the collision of the vessel with the sea-bottom, that may occur in shallow-water operations, if proper precautions are not taken.

[2] The database from GADM can be found at http://www.gadm.org/

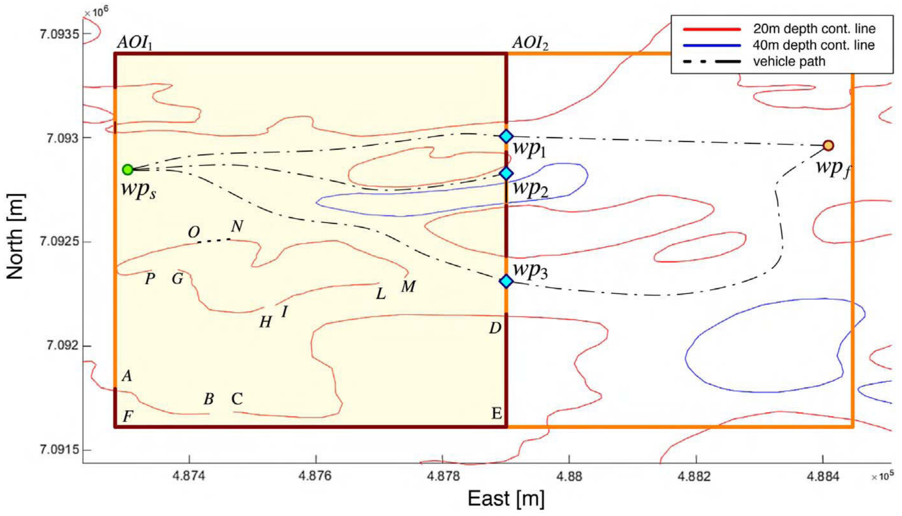


Fig. 1. Red and blue full lines represent depth contour lines. WP_s and WP_f are *starting* and *final* waypoints of the path. The AOI is partitioned in two parts to simplify the calculations. The waypoints WP_1, WP_2 and WP_3, need to be added to connect the two sub-paths defined in the two *AOIs*. Finally, the path passing through WP_1 is selected because is the shortest in the set. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

1. A rectangular portion of the map (ROI, i.e. Region Of Interest) containing the starting and ending point of the route is chosen (AOI_1 in Fig. 1, where start and final points are marked as WP_s and WP_f);
2. Only the map data in the ROI is loaded in memory;
3. Many land portions that are close to the borders of the ROI are naturally cut. These areas should be closed, so that a closed convex polygon is generated again (for example by connecting the segments $\overline{DE}$, $\overline{EF}$ and $\overline{FA}$ to the segments $\overline{AB}$ and $\overline{CD}$);
4. It is observed that lines that should represent the same pieces of land are often segmented. Moreover these structures are often missing some connection points. This should be corrected by unifying the structures and including the missing data (for example one contour lines in Fig. 1 is divided in 4 segments: $\overline{GH}$, $\overline{IL}$, $\overline{MN}$, $\overline{OP}$. By adding the segments $\overline{NO}$, $\overline{PG}$, $\overline{HI}$, $\overline{LM}$ it is possible to obtain a correctly defined polygon).

If the map is too big to be processed in a reasonable time, it can be partitioned into sub-areas, solutions for every partition can be found and subsequently merged (Fig. 1).

3. Path-planning system: initial path generation

In this section the main steps of the path-planning system are presented. Both land, water depth, and clearance constraints will be explained and considered in the following.

3.1. Land and depth constraints

The most important performance requirement that a path should satisfy is safety. The vessel must not collide with, or get too close (hence putting itself closer to a possible risky situation) to any obstacle located in the navigation space. A *minimum clearance*, i.e. the minimum vessel-obstacles distance, is defined to achieve this purpose. Moreover, surface vessels may have navigation constraints due to shallow water depths depending on their dimensions and operations. Some operations may have to be carried along the coastline, and the vessel should not encounter the risk of grounding. In this work, the obstacles (e.g. the coastlines, the depth contours or other vehicles) are represented as polygons. This allows to represent complex environments as simple geometrical structures. The number of vertexes defines the detail level and depends by the scaling and the size of the map.

3.2. Path-planning system steps

The path-planning module aims to generate the waypoints on which the path should be built upon. This algorithm provides a safe path that connects a starting with a final waypoint that respects the kinematics and dynamics constraints of the vessel. Notice that this stage is performed before the vessel starts its navigation, and provide an initial path that can be modified at any time if an obstacle or some unexpected event happens on the way. Once the map data is loaded and the obstacles are acquired as polygons, the *waypoints generation* module can be activated. Fig. 2 shows the main steps of this algorithm: the Voronoi diagram gives a raw obstacle-free roadmap (3a), and the paths that are outside the AOI are removed (3b). The shortest path among the feasible ones is chosen, and its (almost)-collinear waypoints are removed (3f) since having fewer waypoints reduces the length of the path, heading changes and, consequently, the fuel consumption. The clearance constraints is checked, and an alternative path is chosen if the safety constraints are not met (3g). Finally, *unnecessary* waypoints (these which are not almost-collinear and could be removed without breaking the clearance constraints) are removed (3h). This module produces a series of waypoints which need to be connected in a smooth way to produce a *flyable* path for an underactuated vessel, see also Candeloro et al. (2013).

An important property for the path is its curvature continuity. This requirement is important to avoid unwanted phenomena, such as discontinuities in the cross-track error. This is dealt with in the *path smoothing* module, that produces a path that is formed by straight and FS segments. A new clearance check is performed also with the final, smoothed, path.

3.3. The Voronoi diagram and resulting waypoints

A Voronoi diagram is a tool to divide a geometrical space in a number of regions determined by the distribution of the objects present in that space. This general approach allowed many scientists to use this tool in a wide number of fields and applications (Aurenhammer, 1991). In applications such as path planning (and specifically path planning for marine vehicles), the Voronoi diagram can be useful for its property of producing a roadmap with edges that are maximally distant to the generator points (Candeloro et al., 2013, 2016).

The vertexes $V_{o,i}$ of the obstacles' (modeled as polygons, as in Section 2) are selected as the generator vertexes of the Voronoi diagram. The resulting Voronoi diagram edges, are shown as black

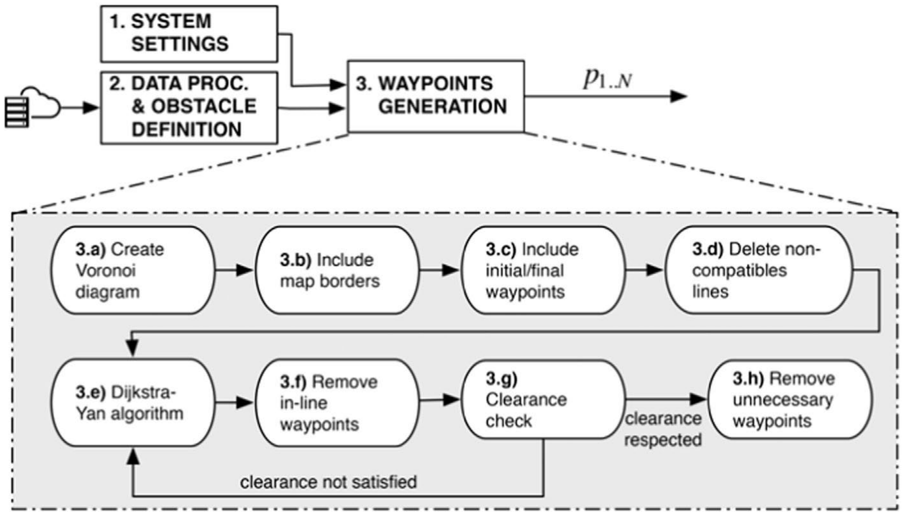

1. SYSTEM SETTINGS
2. DATA PROC. & OBSTACLE DEFINITION
3. WAYPOINTS GENERATION
$p_{1..N}$
3.a) Create Voronoi diagram
3.b) Include map borders
3.c) Include initial/final waypoints
3.d) Delete non-compatibles lines
3.e) Dijkstra-Yan algorithm
3.f) Remove in-line waypoints
3.g) Clearance check
clearance respected
3.h) Remove unnecessary waypoints
clearance not satisfied

Fig. 2. Block diagram representing the *Waypoints Generation* module of the path-planning system. From the map data a series of waypoints that respect the constraints and connect the starting waypoint with the final one, is produced.

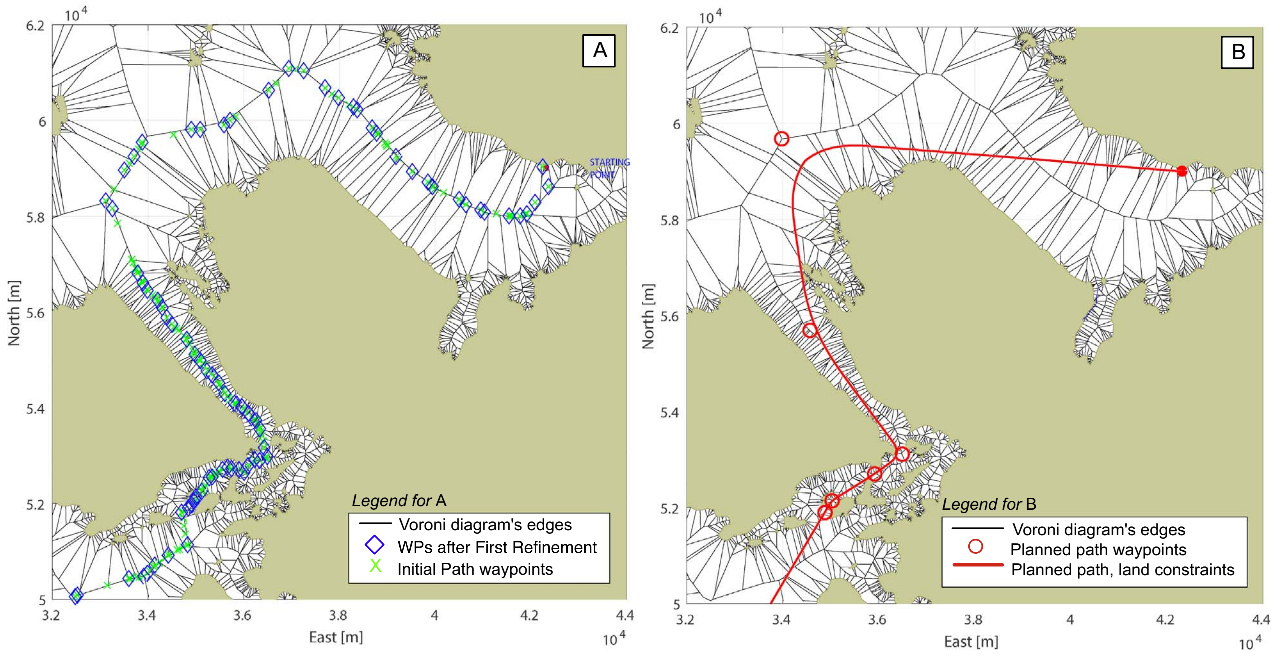


Fig. 3. (A) shows the land obstacles, modeled as polygons. Their vertexes are used as a generator points for the Voronoi diagram (black lines). The waypoints and Voronoi edges too close to the obstacles (i.e. not respecting the clearance constraints) will be eliminated producing the diagram as in Fig. 4A (in this figure the diagram is kept in its original form for clarity). The Yen-Dijkstra algorithm is able to find the shortest available path in the roadmap, and further simplification steps allows to find the minimum number of waypoints which can connect starting and final waypoints respecting the clearance constraints. In (B) a final, curvature-continuous path (constituted by straight segments and Fermat spirals) is obtained from the final waypoints.

lines in Fig. 3. Notice that, in order to limit the Voronoi diagram to the area of interest (corresponding surroundings of the production factory), a number of random vertexes are generated in the obstacle-free edges of the scenario.

Given the Voronoi diagram's edges, the candidate waypoints can be identified in the edges extreme points (Voronoi diagram vertexes). The waypoints which represent the shortest path among the ones provided by the roadmap, are selected through the Yen modification (Yen, 1970) of the Dijkstra optimization algorithm (Dijkstra, 1959). This algorithm is able to find the *k-shortest* paths. In this way, suboptimal solutions can be found if the optimal solution do not respect the preassigned clearance constraints. Notice that the Voronoi edges and vertexes which are too close to the obstacles, i.e. do not respect the predefined clearance constraints, are eliminated, originating the modified diagram which is possible to see in Fig. 4A.

The implementation detail to obtain the Voronoi diagram are not given, since they are extensively treated in the literature, see Ledoux (2007) or Candeloro et al. (2016) for a general procedure.

4. Path generation using Fermat's spiral

The idea of using FS segments for path-planning purposes was introduced and developed in Dahl (2013) and Lekkas, Dahl, Breivik, and Fossen (2013). In this paper it is presented the whole methodology necessary for a vessel to follow a path, despite to the presence of ocean currents. In this context, it is convenient to parametrize the FS segments as a bijective function of θ, so that any point on the path $P = (x, y) = f(\theta)$. Therefore, given a vessel position $P = (x, y)$, there are two possible cases: P *is* on the desired path, in which case the goal is to continue moving on the path, or P *is not* on the desired path (i.e.

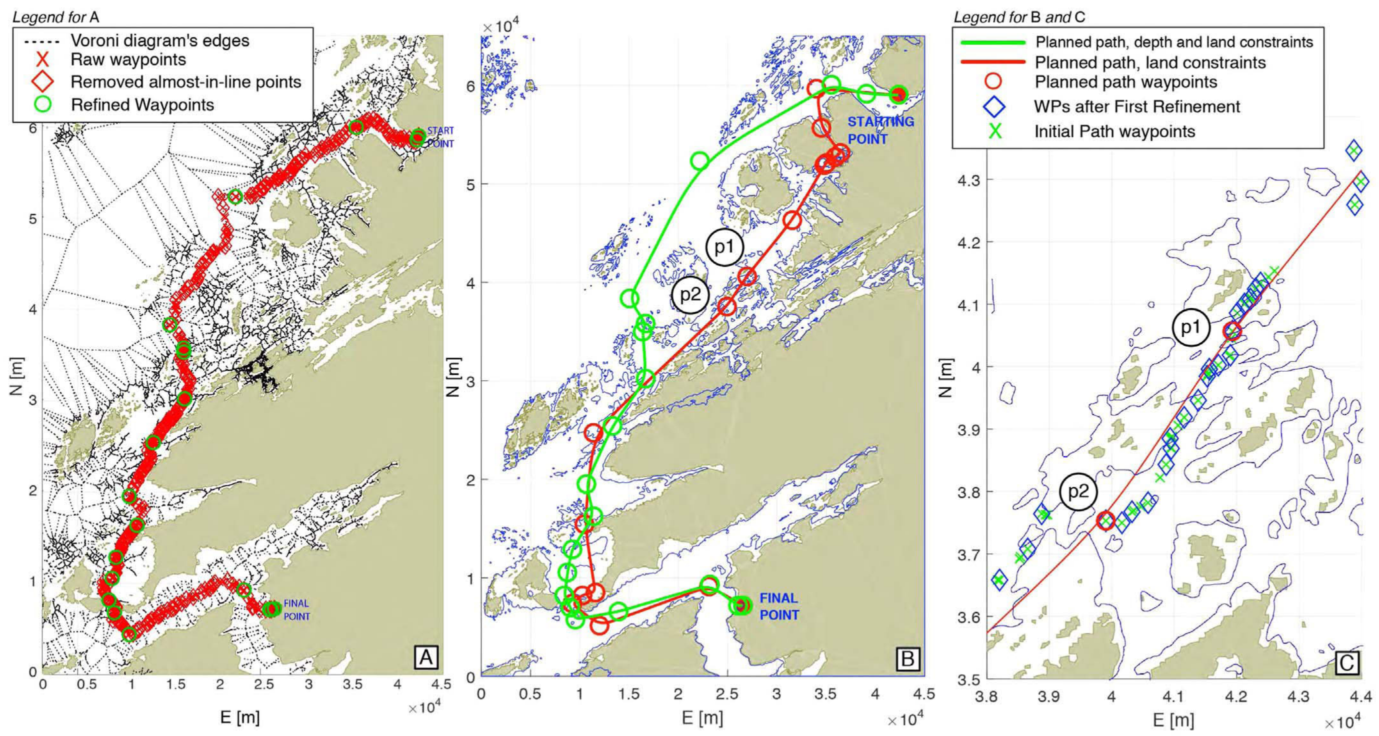


Fig. 4. Path-planning algorithm running on a real map. The area covers $45\,000 \times 65\,000\,\mathrm{m}^2$ in the region of Sør Trondelag, Norway. The (n,e) coordinates have centered in $(x_0, y_0) = (51\,5000,\ 7\,050\,000)$. The map data was processed as shown in Section 2. **A** shows the Vonomoi diagram's edges and the waypoints (WPs) that connect starting with final points respecting land and ocean depths constraints (depth contour lines are reported in **B**). The intermediate WPs are also shown, to illustrate how the implemented heuristic can simplify the problem. **B** shows the paths produced by the path-planning algorithm: the red and green paths shows respectively the one obtained considering land constraints and land plus depth constraints. Blue lines represent the contour ocean depth lines corresponding to the constraint's value. These paths have been calculated as shown in Section 3. **C** presents the detailed path between the WPs P_1 and P_2. Green crosses and blue diamonds are, respectively, the WPs after the steps 3(e) and 3(f) of Fig. 2. Red circles are the WPs after step 3(h). Red path is produced after step 4(g). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

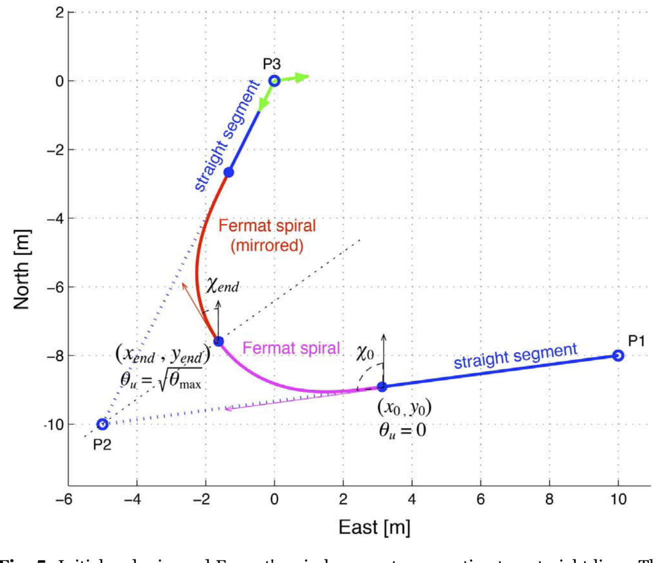


Fig. 5. Initial and mirrored Fermat's spiral segments connecting two straight lines. The resulting path is curvature-continuous. The parameter on which the mathematical formulation of the FS is based, θ, is reported, together with the course angle χ.

nonzero cross-track error), in which case the closest point on the path must be determined and fed to the guidance algorithm to lead the vessel back on the path. If external disturbances, as ocean currents, or wind, are present, it is necessary to compute the heading angle correction that will counteract them and minimize the cross-track error. An example of FS is illustrated in Fig. 5, together with some of the variables introduced in the next subsections.

Similarly to clothoids, connecting two lines using FS requires two segments: the first one starts at the end of the first line (where the curvature is zero) and reaches up to the middle of the total turn. The second segment is the mirrored version of the former one, starts where the initial segment ended, and goes all the way to the second line. The advantage of using FS instead of clothoids is that it requires much less computational time, due to the fact that FS can be described by simple parametric equations, whereas clothoids require the numerical solution of Fresnel integrals. In Dahl (2013) and Lekkas et al. (2013), FS equations in Cartesian coordinates were proposed, which introduce singularities when differentiated with respect to θ. By substituting $\theta_u = \sqrt{\theta}$ it is possible to obtain a singularity-free formulation:

$$\mathbf{p}_{\mathrm{FS}}(\theta_u) = \begin{bmatrix} x_0 + k_s\theta_u\cos(\rho\theta_u^2 + \chi_0) \\ y_0 + k_sk\theta_u\sin(\rho\theta_u^2 + \chi_0) \end{bmatrix}, \quad \text{for } 0 < \theta_u < \sqrt{\theta_{\max}}, \tag{1}$$

where $\mathbf{p}_0 = [x_0, y_0]^{\mathrm{T}}$ is the starting position of the segment, k_s denotes the scaling factor, χ_0 the initial tangent angle, and

$$\rho = \begin{cases} 1 & \text{for an anti-clockwise turn} \\ -1 & \text{for a clockwise turn.} \end{cases} \tag{2}$$

Regarding the parameter θ, it holds θ=0 at the starting point of the *initial* segment where the spiral is connected to a straight line and the curvature is zero, and $\theta = \theta_{\max}$ at the *end point of the initial segment*, which is located midway the end of the total turn.

The formulation in (5) gives the singularity-free expression for the velocity and acceleration:

$$\frac{d}{d\theta_u}\mathbf{p}_{\mathrm{FS}}(\theta_u) = k_s\begin{bmatrix} \cos(\rho\theta_u^2 + \chi_0) - 2\rho\theta_u^2\sin(\rho\theta_u^2 + \chi_0) \\ \sin(\rho\theta_u^2 + \chi_0) + 2\rho\theta_u^2\cos(\rho\theta_u^2 + \chi_0) \end{bmatrix}, \tag{3}$$

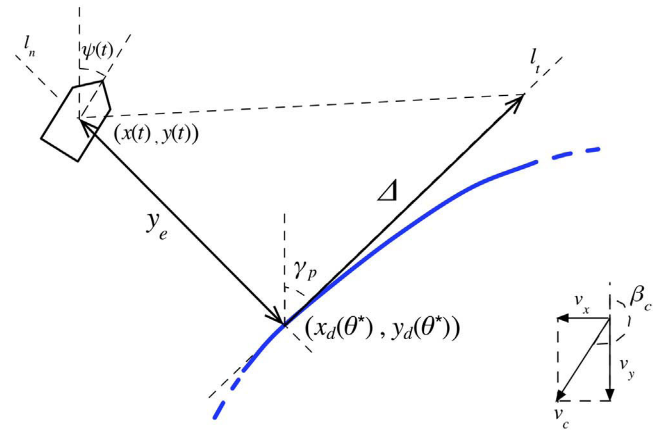


Fig. 6. Main variables concerning the LOS algorithm are reported in this figure: the *lookahead distance* Δ is directing the vessel over a point along the path tangent, oriented at an angle γ_p. The control objective tries to reduce the *cross track error* y_e.

$$\frac{d^2}{d\theta_u^2}\mathbf{p}_{\mathrm{FS}}(\theta_u) = k_s\begin{bmatrix} -6\rho u\sin(\rho\theta_u^2+\chi_0) - 4\rho^2u^3\cos(\rho\theta_u^2+\chi_0) \\ 6\rho u\cos(\rho\theta_u^2+\chi_0) - 4\rho^2u^3\sin(\rho\theta_u^2+\chi_0) \end{bmatrix}. \tag{4}$$

Similarly, for the mirrored segment will hold:

$$\mathbf{p}_{\overline{\mathrm{FS}}}(\theta_u) = \begin{bmatrix} x_{\mathrm{end}} - k_s\theta_u\cos(\rho\theta_u^2 - \chi_{\mathrm{end}}) \\ y_{\mathrm{end}} + k_s\theta_u\sin(\rho\theta_u^2 - \chi_{\mathrm{end}}) \end{bmatrix}, \quad \text{for } 0 < \theta_u < \sqrt{\theta_{\max}} \tag{5}$$

where $\mathbf{p}_{\mathrm{end}} = [x_{\mathrm{end}}, y_{\mathrm{end}}]^{\mathrm{T}}$ is the position at the end of the curve, i.e. $\mathbf{p}_{\mathrm{end}} = \mathbf{p}_{\overline{\mathrm{FS}}}(\theta_{\mathrm{end}})$, and χ_{end} is the course angle at that point. Fig. 6 depicts an illustration of how the initial and mirrored segment connect two straight lines. Equations related to the mirrored segment are not reported here for space issue. Section 5 explains why the first and second derivatives are needed when implementing a path-following scenario where no temporal constraints are present.

5. Motion control objective

For each vessel position $(x(t), y(t))$ it is necessary to determine the closest corresponding point on the path, $(x_d(\theta^*), y_d(\theta^*))$, which corresponds to a unique parameter value θ^*, see also Fig. 6. The minimum distance between the vessel and the monotonic curve between two active waypoints will be defined as the cross-track error (Lekkas & Fossen, 2014a). The tangent and normal lines through the point $(x_d(\theta), y_d(\theta))$ are given by[3]:

$$y_t - y_d(\theta) = \frac{y'_d(\theta)}{x'_d(\theta)}(x_t - x_d(\theta)), \tag{6}$$

$$y_n - y_d(\theta) = -\frac{1}{\frac{y'_d(\theta)}{x'_d(\theta)}}(x_n - x_d(\theta)). \tag{7}$$

The θ value corresponding to the path-normal that intersects the vessel is found by requiring that $(x_n, y_n) = (x, y)$. Moreover, from (7) it follows that:

$$y'_d(\theta^*)(y - y_d(\theta^*)) + x'_d(\theta^*)(x - x_d(\theta^*)) = 0. \tag{8}$$

This involves solving the roots of the third-order cubic function for θ^*. Instead of using an analytical solution a numerical solution based on Newton–Raphson method will converge quite fast. For instance,

$$\theta_{j+1}^* = \theta_j^* - \frac{f(\theta_j^*)}{f'(\theta_j^*)}, \tag{9}$$

[3] The syntax $x'(\theta)$, represents the derivative of the variable with respect of its parameter θ: $x'(\theta) = \frac{dx}{d\theta}$.

where:

$$f(\theta_j^*) = y'_d(\theta_j^*)(y - y_d(\theta_j^*)) + x'_d(\theta_j^*)(x - x_d(\theta_j^*)), \tag{10}$$

$$f'(\theta_j^*) = y''_d(\theta_j^*)(y - y_d(\theta_j^*)) + x''_d(\theta_j^*)(x - x_d(\theta_j^*)) - x'_d(\theta_j^*)^2 - y'_d(\theta_j^*)^2, \tag{11}$$

will converge in a few iterations if the initial path variable θ_0^* is taken as the last θ_i value when moving along the path between two waypoints parametrized on the interval $[\theta_1, \theta_n]$.

The normal line from the point $(x_d(\theta^*), y_d(\theta^*))$ on the path through the point (x,y) on the vessel defines the along-track and cross-track errors (x_e, y_e). Moreover:

$$\begin{bmatrix} x_e \\ y_e \end{bmatrix} = \mathbf{R}^T(\gamma_p)\begin{bmatrix} x - x_d(\theta^*) \\ y - y_d(\theta^*) \end{bmatrix}, \tag{12}$$

where $\mathbf{R}(\gamma_p) \in SO(2)$ is the rotation matrix in yaw (Fossen, 2011). In algebraic form, the equations of the along-track and the cross-track error for a given vessel position (x,y) become:

$$x_e = (x - x_d(\theta^*))\cos(\gamma_p) + (y - y_d(\theta^*))\sin(\gamma_p), \tag{13}$$

$$y_e = -(x - x_d(\theta^*))\sin(\gamma_p) + (y - y_d(\theta^*))\cos(\gamma_p), \tag{14}$$

where γ_p is the path-tangential angle:

$$\gamma_p = a\tan 2(y'_d(\theta^*), x'_d(\theta^*)). \tag{15}$$

For a path-following scenario x_e=0, which leads to the following control objective for curved path-following (Lekkas & Fossen, 2014a):

$$\lim_{t\to+\infty} y_e(t) = 0. \tag{16}$$

6. Vessel model and control system

Consider a surface vessel at the position (x,y) moving with the ground speed:

$$U = \sqrt{u^2 + v^2}, \tag{17}$$

where u and v are the velocities in surge and sway respectively. The speed U is assumed to be positive and bounded:

$$U_{\min} \le U \le U_{\max}, \quad 0 < U_{\min}. \tag{18}$$

In the following, however, the relative surge and sway velocities will be mostly considered, $u_r = u - u_c$ and $v_r = v - v_c$, where u_c and v_c will be defined below. In this way, the three DOFs horizontal dynamics of the surface vessel can be represented by three differential equations Fossen (2011):

$$\dot{u} = f_{u_r}(u_r, v_r, r, \tau), \tag{19a}$$

$$\dot{v} = f_{v_r}(u_r, v_r, r, \tau), \tag{19b}$$

$$\dot{r} = f_r(u_r, v_r, r, \tau), \tag{19c}$$

The vessel kinematic equations for horizontal plane motion can be expressed in terms of the relative surge and sway velocities according to Fossen (2011):

$$\dot{x} = u_r\cos(\psi) - v_r\sin(\psi) + V_x \tag{20a}$$

$$\dot{y} = u_r\sin(\psi) + v_r\cos(\psi) + V_y \tag{20b}$$

$$\dot{\psi} = r \tag{20c}$$

where ψ and r are the yaw angle and rate, respectively. The body-fixed ocean current velocities (u_c, v_c) and North-East current velocities (V_x, V_y) satisfy:

$$[u_c, v_c]^{\mathrm{T}} = \mathbf{R}^{\mathrm{T}}(\psi)[V_x, V_y]^{\mathrm{T}} \tag{21}$$

Notice that the pair (V_x, V_y) is constant in NED, while the body-fixed current velocities (u_c, v_c) depend on the heading angle ψ.

7. Guidance system and ocean current compensation

The LOS guidance method is widely used to generate reference trajectories for the heading angle, so that the vessel's trajectory will converge on the desired path. Consequently, a wide literature is available, see, for instance, Caharija, Pettersen, Sørensen, Candeloro, and Gravdahl (2014) and the references therein. In this work, an adaptive version of the LOS algorithm is used in order to compensate for environmental disturbances.

7.1. Indirect adaptive LOS for ocean current compensation

An additional term, α_y is added to the classic LOS algorithm with the purpose of disturbance rejection (Lekkas & Fossen, 2014b):

$$\psi_d = \gamma_p - \beta_r + \arctan\left(-\frac{y_e + \alpha_y}{\Delta}\right), \tag{22}$$

which gives the desired heading angles ψ_d. $\Delta > 0$ is the user specified lookahead distance and $\beta_r = a\tan2(v_r, u_r)$. Equation (22) results in the following cross-track error dynamics:

$$\dot{y}_e = -\frac{U_r(y_e + \alpha_y)}{\sqrt{\Delta^2 + (y_e + \alpha_y)^2}} + \theta_y, \tag{23}$$

with θ_y denoting the effect of the unknown current that needs to be compensated for, and the relative speed is recognized as $U_r = \sqrt{u_r^2 + v_r^2}$. In the same work the following adaptive observers providing estimates for y_e, θ_y were proposed (Fossen & Lekkas, 2015):

$$\dot{\hat{y}}_e = -\frac{U_r(\hat{y}_e + \alpha_y)}{\sqrt{\Delta^2 + (y_e + \alpha_y)^2}} + \hat{\theta}_y + k_1(y_e - \hat{y}_e), \tag{24}$$

$$\dot{\hat{\theta}}_y = k_2(y_e - \hat{y}_e), \tag{25}$$

where k_1 and k_2 are tuning constants, U_r is the relative speed, and Δ is the user specified lookahead distance. Choosing α_y as:

$$\alpha_y = \Delta\frac{\hat{\theta}_y/U_r}{\sqrt{1 - (\hat{\theta}_y/U_r)^2}} \tag{26}$$

minimizes the cross-track error dynamics, i.e. $y_e(t) \to \infty$, Fossen and Lekkas (2015). This indirect adaptive LOS is implemented by computing the desired heading angles according to (22) with α_y given by (26).

8. Obstacles classification and tracking

In this paper, an obstacle tracking algorithm is utilized to detect and estimate the state of the encountered obstacles. This approach introduces uncertainties in the knowledge of the obstacles state, and in the time utilized for the detection itself, allowing to create a more realistic and challenging scenario. An approach similar to the one in Karoui, Quidu, and Legris (2015) is utilized, and briefly presented in the following. Notice that two assumptions holds:

Assumption 1. Obstacles are supposed to be defined by a nearly-constant velocity vector, where changes in magnitude and direction are bounded.

Assumption 2. Moving obstacles are assumed to follow the COLREG.

Assumption 1 allows the replanning algorithm to plan a proper anti-collision maneuver based on a predictable behavior of the obstacle, while Assumption 2 gives indications on *how to* safely replan, i.e., following the COLREG.

Given Assumption 1 it is possible to use the constant velocity dynamical model described in Lerro and Bar-Shalom (1993). Karoui et al. (2015) propose an obstacle detection and identification method that could be implemented in synergy with the obstacle tracking and planning techniques here proposed. The considered model is composed by the following dynamic and measurement equations[4]:

$$\boldsymbol{x}_o^n(k) = \boldsymbol{F}\boldsymbol{x}_o^n(k-1) + \boldsymbol{G}\boldsymbol{w}_1(k), \tag{27a}$$

$$\boldsymbol{y}^n(k) = \boldsymbol{H}\boldsymbol{x}_o^n(k) + \boldsymbol{w}_2(k), \tag{27b}$$

where $k > 0 \in \mathbb{I}$ is the discrete time index, $\boldsymbol{x}_o^n(k) = [\boldsymbol{p}_o^n(k),\ \boldsymbol{v}_o^n(k)]^T \in \mathbb{R}^{4\times1}$ is the state of the moving obstacle (position and velocity) at sensor ping k in the NED frame. $\boldsymbol{\xi} = \boldsymbol{G}\boldsymbol{w}_1(k) \in \mathbb{R}^{4\times1}$ is the process noise modeled by the covariance matrix $\boldsymbol{Q} \in \mathbb{R}^{4\times4}$: $\boldsymbol{\xi} \sim \mathcal{N}(0, \boldsymbol{Q})$, $\boldsymbol{w}_1 \in \mathbb{R}^{4\times1}$ is the Gaussian measurement noise modeled by the covariance matrix $\boldsymbol{R} \in \mathbb{R}^{2\times2}$: $\boldsymbol{w}_2 \in \mathbb{R}^{2\times1} \sim \mathcal{N}(0, \boldsymbol{R})$.

A linear Kalman Filter, which includes a bias correction is implemented to obtain an estimation of the state of the obstacle and of the related error covariance, as discussed in Lerro and Bar-Shalom (1993). The resulting filter is known as de-biased converted Kalman Filter. More details can be found in Karoui et al. (2015). Assuming a Gaussian behavior of the sensor noise is a standard practice in literature. Although in reality the sensor noise might follow a different behavior, the practical application has shown that the Kalman filter is a reliable estimator. This approach is also utilized in Quidu, Jaulin, Bertholom, and Dupas (2012) and Karoui et al. (2015).

The obstacle detection and identification module is out of the scope of this paper, so the position measurements are available as soon as the obstacle enters the sensor's field of view (FOV). Notice that, in this work, the obstacle is estimated as a point. In Section 9.4 a certain, predefined shape will be associated to it. The process that has been used in this work to originate the simulated measurements is described in Section 10.

Fig. 7 shows an example of obstacle tracking. The covariance of the estimation error (provided by the de-biased converted Kalman filter) is used to generate the confidence ellipse around the estimated position of the obstacle which, in turns, are used to generate the Projected Obstacle Area (POA).

9. Path-planning system: replanning phase

In order to safely avoid the obstacles identified by the tracker, the replanning procedure adds extra waypoints and produces a flyable deviation from the original path. The algorithm also guarantees smooth re-convergence to the original path.

9.1. Timing

Safe path replanning requires accuracy in both obstacle tracking and the timing of the sequential steps. The following time variables and periods are defined (see Fig. 8):

- *Tracking time* (t_t): the obstacle enters the sensor area and it is identified. From this moment the *obstacle tracking* runs for a period T_t, until the state of the obstacle is estimated with a certain precision;
- *Replanning time* (t_r): the path replanning system starts to re-calculate the route. The replanning period T_r lasts until the algorithm produces a result;
- *Action time* (t_a): the vessel changes its course to follow the new segment.

[4] In the following, the superscript x^n represents the NED (North-East-Down) frame, the superscript x^s represents the sensor frame. For more information, refer to Fossen (2011).

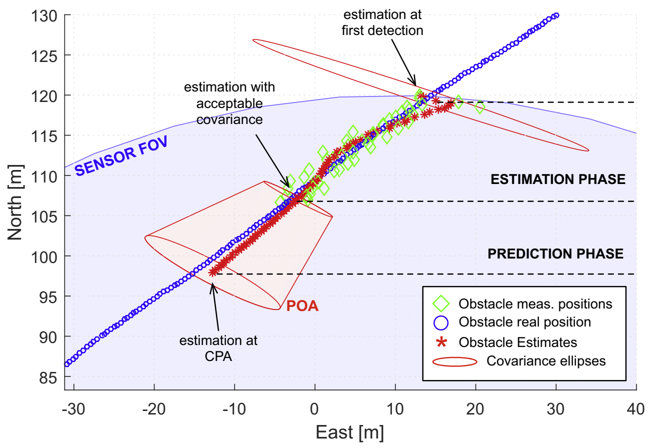


Fig. 7. An obstacle is first detected when it enters the sensor's FOV, but the tracking commences when the covariance is sufficiently low. The POA is the convex hull containing the two covariance ellipses corresponding to the first acceptable estimation and the estimation at CPA.

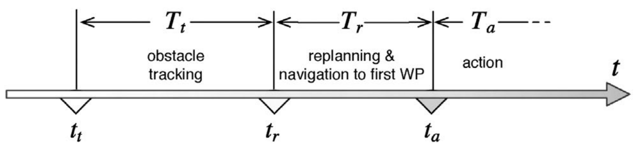


Fig. 8. Time variables and periods involved in path replanning.

9.2. Closest point of approach

The *Closest Point of Approach* (CPA) is a way to evaluate the hazard due to the minimum potential distance between the target vessel and the obstacle (Kuwata, Wolf, Zarzhitsky, & Huntsberger, 2014). In this work, the CPA is used to determine whether a replan is necessary or not, and to build the Projected Obstacle Area (POA) in Section 9.4. The CPA is described by d_{CPA}, t_{CPA}, the minimum distance and the time instant when the target vessel and the obstacle reach it, respectively:

$$t_{CPA} = \frac{(\boldsymbol{p}_v - \boldsymbol{p}_o)\cdot(\boldsymbol{v}_v - \boldsymbol{v}_o)}{\| \boldsymbol{v}_v - \boldsymbol{v}_o \|}, \tag{28}$$

$$d_{CPA} = \|(\boldsymbol{p}_v + \boldsymbol{v}_v t_{CPA}) - (\boldsymbol{p}_o + \boldsymbol{v}_o t_{CPA})\|, \tag{29}$$

where $\boldsymbol{p}$ and $\boldsymbol{v}$ are he position and velocity vectors of the target vessel and the obstacle. It is clear that if $\| \boldsymbol{v}_v - \boldsymbol{v}_o \| \to 0$ then $t_{CPA} \to \infty$ it is implied that the obstacle and target vessel are on the same course with equal speed. The replanning procedure starts if the target vessel is in a collision risk:

Definition 1. The target vessel is in a *collision risk* if:

$$(0 < t_{CPA} \le t_{risk}) \quad \wedge \quad (d_{CPA} \le d_{risk}). \tag{30}$$

9.3. Compliance to COLREG guidelines

The International Maritime Organization (IMO) established the COLREGs in IMO (1977). For the replanning phase we consider the rules found in Section 2 of the COLREG (*Conduct of Vessels in Sight of One Another*) and mainly refer to Rules 14–15 (*head-on situation* and *crossing situation*), both illustrated in Fig. 9. Another relevant rule, although not considered in this work, is Rule 13 (*overtaking situation*). In general, the target vessel should always leave the moving obstacle on its port side. In the case of a crossing situation, the target vessel may keep its course if the vehicle comes from his starboard side, but leave the obstacle on his port side otherwise (as in Fig. 9A). To determine the situation, the relative course α_r can be calculated in the following way:

$$\alpha_r = a\tan2(y_v - y_o, x_v - x_o) - \psi_o, \tag{31}$$

where x and y are the Cartesian coordinates of the target vehicle/obstacle, and ψ_o is the obstacle heading angle. The angle α_r determines the situation as defined in Fig. 9A.

9.4. Projected obstacle area

When a dynamic obstacle is detected, it is possible to predict its near-future positions, thus generating the Projected Obstacle Area (POA) (Larson et al., 2006, 2007). The size of the POA is a function of the uncertainty in tracking the obstacle (high uncertainty will result in a higher number of probable near-future positions) and the speed of the obstacle. In this paper, the KF obstacle position estimates are used to compute the POA. In particular, the covariance is used for: (a) assessing whether the obstacle tracking is reliable enough to be used in the replanning phase, and (b) building the POA. As soon as the tracking accuracy is acceptable, i.e. the estimation covariance is low enough, the prediction of the obstacle's future states starts. The last considered prediction is the one corresponding to the CPA. The convex hull containing the first and last obstacle state estimations' covariance ellipses obtained during this procedure defines the POA, as shown in Fig. 7. If $\boldsymbol{v}_o \approx 0$, the POA is approximated as a circle with a certain radius (Fig. 14). The POA is then further refined so as to exclude the paths that do not satisfy the COLREG. This is illustrated in Fig. 13, where the POA is extended towards the border of the replanning area to exclude the routes that would take the obstacle on the port side.

9.5. Path replanning system steps

In the case where replanning is needed, the same algorithm that generated the initial path is now applied to a restricted area centered on the current position of the target vessel. The size of that area is determined based on a compromise between the need for a fast solution and the need to consider a large enough area that can give valid replanning alternatives. When an obstacle is detected and tracking commences, the replanning module follows the routine below:

1. Define the replanning area (local map).
2. Compute and add the POA to the local map.
3. Compute the first WP of the replanned segment.
4. Compute the last WP of the replanned segment.
5. Generate the VD on the local map.
6. Connect the first and last WPs to the VD.
7. Compute the shortest path.
8. Simplify and smooth the path.

If no solution is found, the process is repeated after increasing the dimension of the replanning area. During the replanning period T_r, the CPA is continuously monitored and an emergency stop can be executed, should the need arise. The procedure to calculate the POA (Step 2) was given in Section 9.4 and Steps 5–8 are identical with those presented in Section 3. The procedure that follows explains in detail how the replanning area (Step 1), as well as the first and last WPs, are computed (see also Figs. 8 and 10):

1. The first waypoint WP_s of the replanned segment is obtained calculating the position of the target vessel at the time $T^* > t_c + T_r$, where T^* is a conservative estimation of the time that the replanning algorithm takes to give a solution for the selected area, and t_c is the current time;
2. $\vec{d}_{wp}$ (the versor representing the direction of the line connecting the waypoints WP_k and WP_{k+1}) is calculated. This direction defines the orientation of the replanning frame $\{r\}$. The x and y axis of this frame are respectively perpendicular and parallel to $\vec{d}_{wp}$. Its center is located on the target vessel COG. All the following variables are

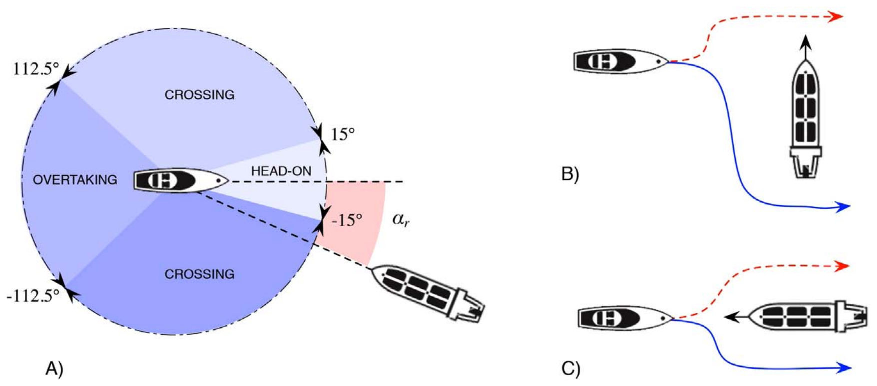


Fig. 9. The possible mutual situations between the *target vehicle* and *moving obstacle* are showed in (A). A crossing and a head-on situation are depicted in (B) and (C), respectively. Solid blue lines indicate the approved maneuver from the *target vehicle* perspective, whereas red dashed lines are maneuvers to avoid. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

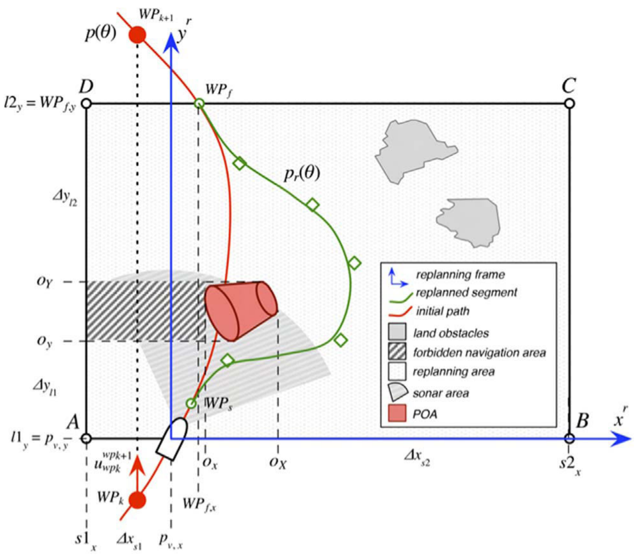


Fig. 10. Illustration of the replanning area. The replanning frame (blue axes) is oriented towards the line connecting the WP_k to WP_{k+1}. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

defined in this frame;

3. The following values are calculated: $ox = \min(o_x)$, $oX = \max(o_x)$, $oy = \min(o_y)$ and $oY = \max(o, y)$ that are the minimum and maximum coordinates of the POA. Notice that $\boldsymbol{p}_o^r = (o_x, o_y) \in POA$;
4. The replanning area $\overline{ABCD}$ is defined from the intersections of four lines parallel to the axis of the frame $\{r\}$: $l1$, $l2$, $s1$, $s2$;
5. The following values are calculated: $l1_y = v_y$, $l2_y = oY + \Delta y_{l2}$;
6. The intersection of $l2$ with the original path $r(\theta)$ gives WP_f, that is the final waypoint of the replanned segment;
7. The following values are calculated: $s1_x = min\{p_{v,x}, ox, wp_{f,x}\} - \Delta x_{s1}$, $s2_x = max\{p_{v,x}, oX, wp_{f,x}\} + \Delta x_{s2}$ define the side lines. Notice that $WP_f = (WP_{f,x}, WP_{f,y})$ and $p_v = (p_{v,x}, p_{v,y})$;
8. The following values are calculated: $\Delta x_{s1} = 100$, $\Delta x_{s2} = 2000 - |oX|$, $\Delta x_{l2} = 2000 - oY$
9. In case of moving obstacle, the *forbidden navigation area* (see Fig. 10) limited by the coordinates o_y, o_Y and $s1_x$ and $WP_{f,x}$ is added as an additional obstacle.

The values of Δ_{s1}, Δ_{s2} and Δ_{l2} are defined arbitrarily to have a replanning area of around 2000×2000 square meters, that extends in both directions for an extension of around 4 times the range of the sensor. This area can be successfully replanned in less than $T_r = 0.6$ s, so it is possible to define $T^* = 2$ s. If a solution cannot be found, the replanning area is increased and the procedure repeated again. If during this process the CPA becomes critical, the navigation is stopped. Moreover, this paper considers dynamic obstacles with realistic dynamics. In this sense it is supposed that changes in their speed or direction cannot occur instantly, and can be identified by the obstacle tracking algorithm. If the new conditions do not allow a replanning in the safety time margins, the system would notify the captain that an emergency stop may be needed.

10. Case study and simulation results

In this section, we present information related to setting-up and running the simulations. A block diagram illustrating the logic flow of the whole path-planning system and the sections within the paper that describe its elements can be seen in Fig. 11.

10.1. Map and environmental conditions

The simulations are carried out on real map data, representing a region of around 3000 Km^2 in the Sør-Trøndelag region of Norway. The data have been obtained from the sources reported in Section 2. We considered a constant (w.r.t. the inertial frame) ocean current of magnitude $U_c = 2$ m/s and direction $\beta_c = 90°$ (from West to East).

10.2. Obstacles motion

As mentioned in Section 8, we consider *dynamic obstacles* with unknown velocity, which is estimated online along with the obstacle's position. In particular, the obstacle speed is modified by adding a Gaussian noise around its average speed:

$$|u_o| \sim \mathcal{N}\left(|\overline{u}_o|, \frac{|\overline{u}_o|}{3}\right) \tag{32}$$

where u is the surge speed, that is the first element of the $\boldsymbol{v}_o$ obstacle velocity vector. The course angle evolves following a random walking process, where at every step the probabilities of changing direction is set at 0.4, giving as a result a motion with a certain variability in the course. The heading varies with steps of $\Delta_\psi = 1°$, where the period that is used for generating the moving obstacle motion is set to 0.01 s. The random walking process that generates the heading can then be defined by the following discrete probability function:

$$P[\psi_o(k+1) = \psi_o(k) + \Delta_\psi] = 0.2; \tag{33}$$

$$P[\psi_o(k+1) = \psi_o(k)] = 0.6; \tag{34}$$

$$P[\psi_o(k+1) = \psi_o(k) - \Delta_\psi] = 0.2; \tag{35}$$

This adds a certain level of unpredictability to the simulated obstacle motion, that is useful to test and challenge the assumptions stated in

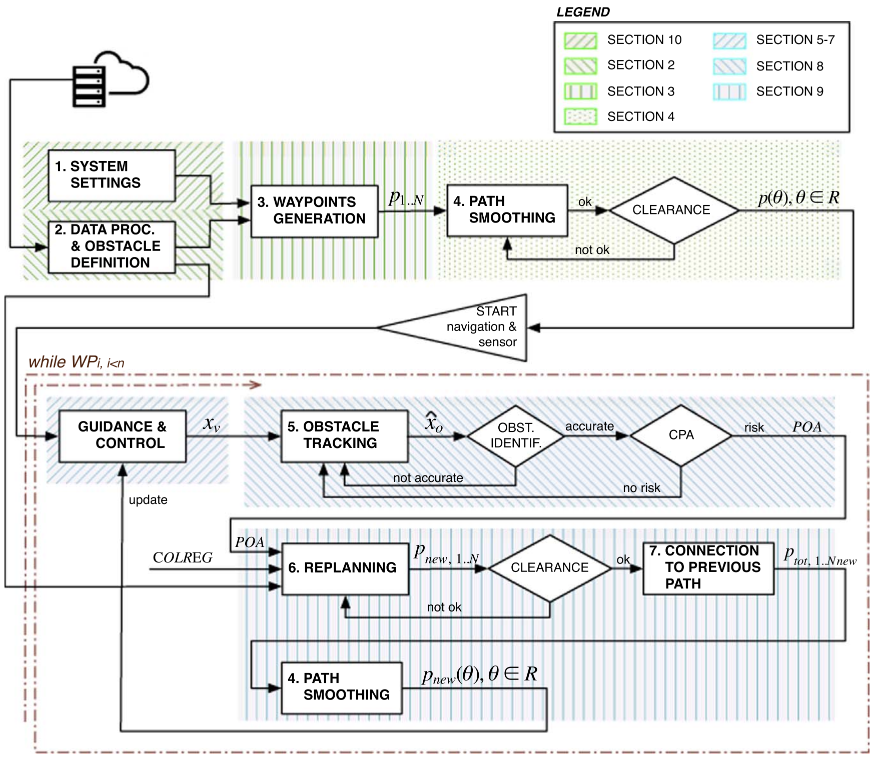


Fig. 11. Illustration of the algorithm's block elements and their correspondence with the sections of this paper. The upper part is executed offline, while the red dashed line contains the online processes. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Section 8.

10.3. Simulations parameters

The simulations were implemented using the model for *Cybership II*, a 1:70 replica of a supply ship (the model parameters can be found in Skjetne, 2005). The target vessel is moving with a relative total speed of $U_r = 5$ m/s. The LOS tuning constants from (24) and (25) were chosen as: $K_1 = 10$, $K_2 = 0.8$. Moreover, the lookahead distance of (22) and following is set to Δ=20 m.

10.4. Sensor characteristics

A variety of sensors can be used for obstacle detection, from passive-ranging cameras (in monocular and stereo configurations) to active-ranging radars, sonars and lidars. Configurations that fuse data from radars and cameras, or range finders and cameras are the most widely utilized ones (Almeida et al., 2009; Heidarsson & Sukhatme, 2011; Larson et al., 2006, Larson, Bruch, Ebken, Rogers, & Webster, 2007). This paper does not treat the sensing module itself nor the data interpretation, in fact the obstacle is detected as soon as it enters the sensor's FOV. The interested reader is referred to Quidu et al. (2012) and Karoui et al. (2015) instead, since the solution therein can be used to detect and avoid both above and sub-surface obstacles in the middle-short range by using only one sensor (sonar). A similar solution is also utilized in Heidarsson and Sukhatme (2011). In our case, the following values define the accuracy and other characteristics of the simulated sensor:

$$\sigma_d = 0.05 \text{ m}, \tag{36}$$

$$\sigma_\delta = 1.5°, \tag{37}$$

$$\sigma_v = 10^{-3} \text{ m/s}^2. \tag{38}$$

where σ_d, σ_δ are the standard deviations of the measurement errors' range and bearing, respectively, and σ_υ the standard deviation of the model error. These variables are used to tune the obstacle tracking algorithms, as in Karoui et al. (2015). Moreover, the sensor is supposed to have a range of 500 m, and to span an angle $\delta \in [-60°, \ 60°]$.

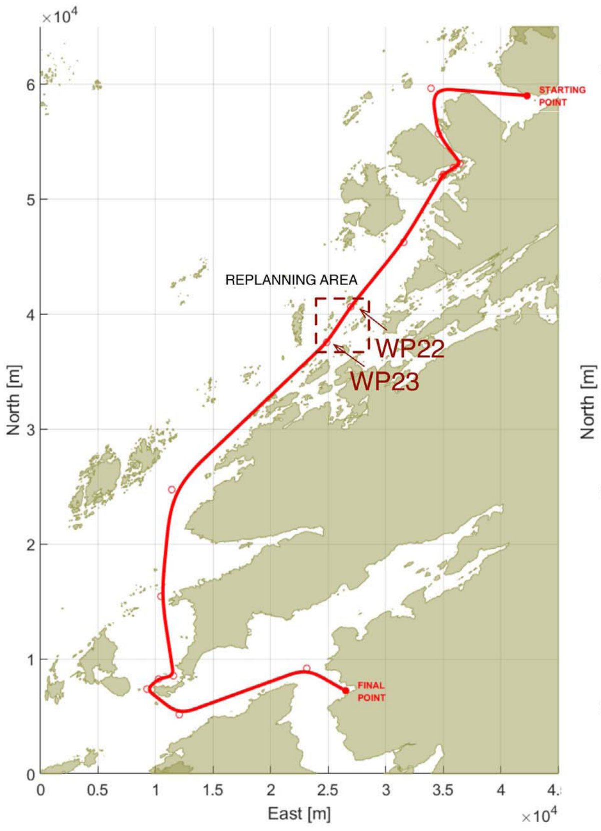
10.5. Simulation results

This section presents a simulation study where two consecutive replanning sessions are needed to avoid two obstacles. The simulation is based on the map area from Fig. 4. However, in this case study we do not consider depth constraints because the resulting paths are safer by keeping the vehicle further away from small islands, hence making the replanning phase easier to execute successfully. On the contrary, we choose to perform the dynamic replanning algorithm at a crammed area with small islands and two additional obstacles (one static and one dynamic).

The selected path segment is between WP_{22}–WP_{23}, see Fig. 12. The vehicle encounters the dynamic obstacle first and starts following the first replanned segment (or deviation). Before returning to the original path, the vehicle encounters the static obstacle and needs to perform a second replanning. Notice that, after travelling on the second replanned segment (green curve), the vehicle converges initially back to the first replanned segment (blue curve) and then to the original path (red curve), that is, it does not go directly from the green to the red path. The FOVs of the sensor in the two moments in which the obstacles are first identified are also shown. Notice that an unknown ocean current is flowing from West to East.

Figs. 13–15 represent zoomed areas of Fig. 12. They show respectively:

- The first deviation from the original path is due to the moving *obstacle* 1 (black vehicle). Notice that this case falls in the *crossing situation* of the COLREGs (from Section 9.3). The moving obstacle is at the starboard side of the target vehicle when it enters the field of view of the sensor, consequently the target vehicle should change its course in order to leave the obstacle on its port side. For this



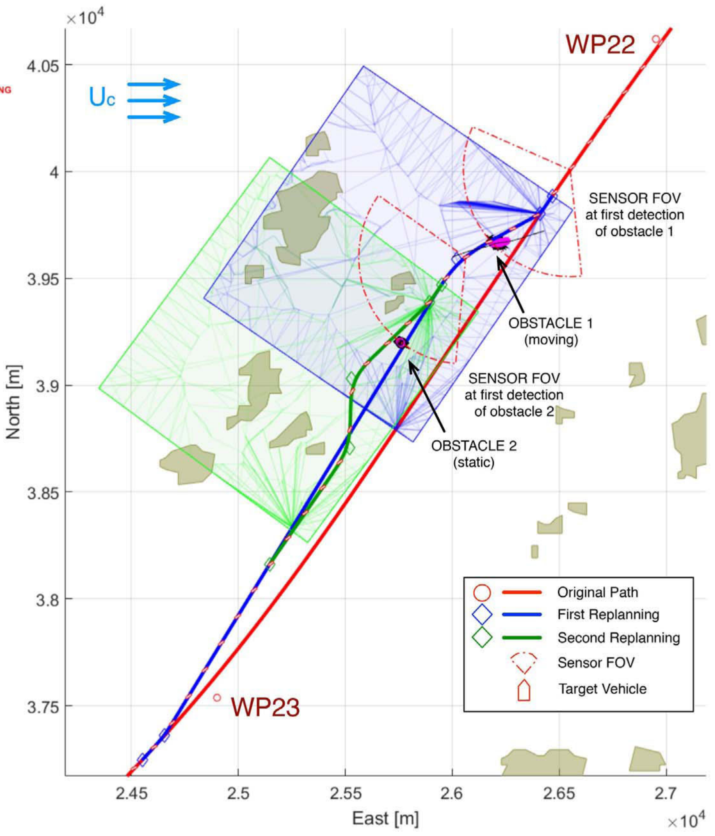
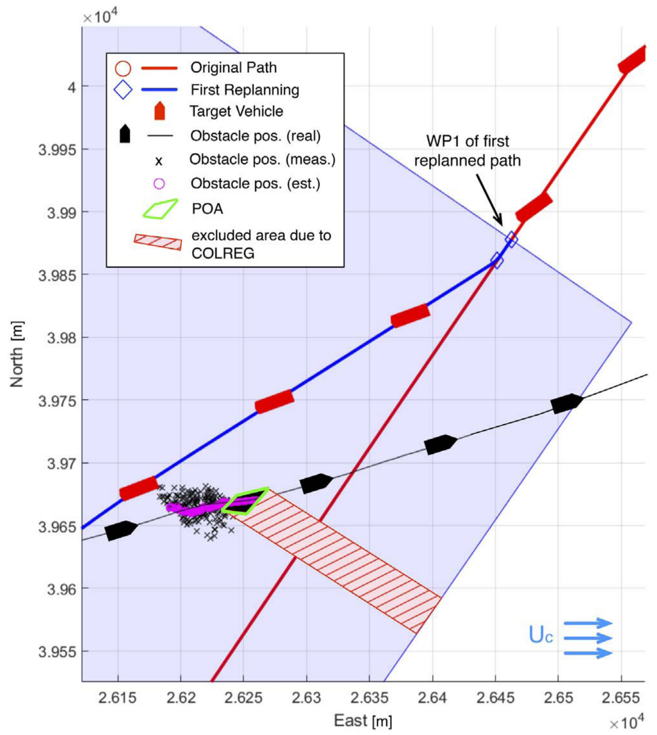


Fig. 12. Demonstration of consecutive replanning involving a dynamic obstacle followed by a static obstacle. The initial path (left plot) is generated based on land constraints. The second plot focuses on the area between WP_{22} and WP_{23}, where the two obstacles are detected and evaded. The two resulting replanning areas are highlighted with blue and green colors. The Voronoi diagram edges used for determining the replanned segments waypoints are also visible. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)



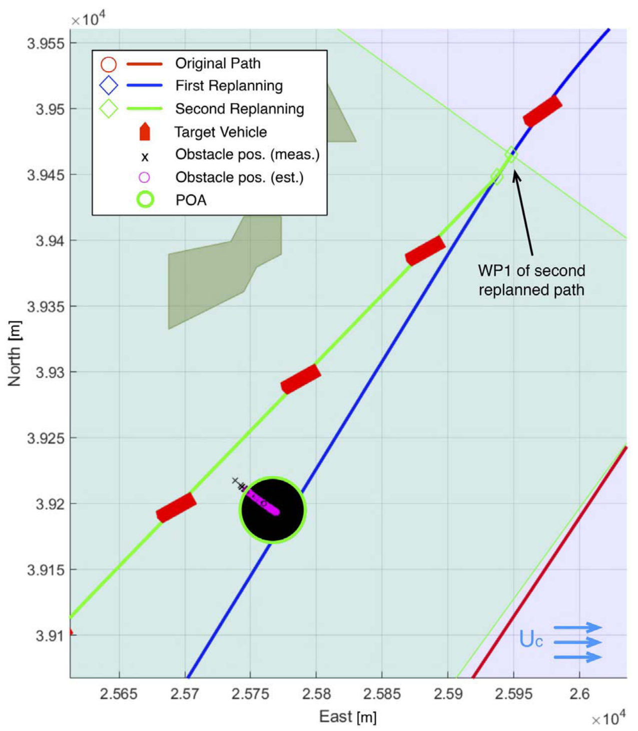


Fig. 13. Detail of the first replanning and of the tracking of *Obstacle 1*. Black crosses are the measured position of the obstacle, violet dots the estimated positions. The POA is also indicated, together with the artifact obstacle added due to the COLREG constraints. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

Fig. 14. Detail of the second replanning and of the tracking of the second obstacle. Black crosses are the measured position of the obstacle, violet dots the estimated positions. In this case the POA is approximated as a disc of standard dimensions, and COLREGs do not need to be included. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

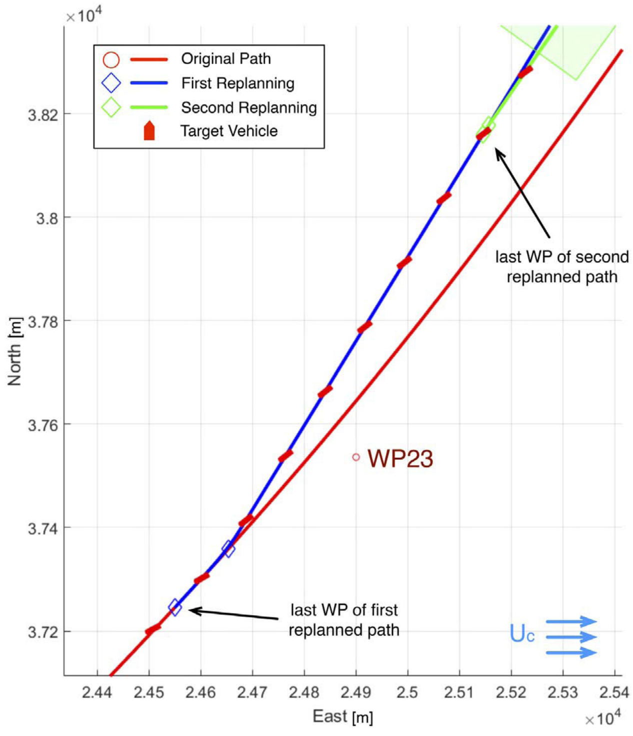


Fig. 15. Detail of path convergence: the second replanned path (green) converges smoothly on the first replanned one (blue), the first replanned path, finally, converges smoothly to the original one (red). (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this paper.)

purpose a virtual obstacle is added to the replanning area, excluding alternative paths that do not satisfy the COLREGS (light green box in Fig. 13).

- The deviation from the blue path due to the static obstacle (approximated as a circle of standard dimension, as explained in Section 9.4). Notice that in this case the POA is not modified, since the COLREGs do not apply.
- The moments when the target vessel joins again the first replanned path (blue), and the initial path (red).

In the aforementioned figure the measures positions of the obstacles (black crosses) and their estimated positions (violet circles), are also depicted. It can be seen how smoothly the vessel is deviated from the original path and joins it again after the escaped risk. The following plots show the details of the control performance along the path.

Fig. 16 shows the CTE for the simulated example. t=252 s corresponds to the *time of action* t_a, introduced in Section 9.1 and Fig. 8. The end of the plot corresponds to the moment when the vessel reach WP_{24} of the original path. Small oscillations in the CTE due to the replanning can be noticed: in these points the vessel changes its course rapidly, and the controller must adjust for the new current estimation. Despite of these aspects, the CTE is kept under 0.4 m along the whole replanned segment. The initial oscillations are due to the fact that the starting vessel's heading is not aligned with the path direction at the starting waypoint (WP_{22}).

11. Conclusions

We presented a dynamic path-planning system for underactuated marine surface vehicles based on the Voronoi Diagram (VD). Given a departure and a destination location, the algorithm first plans an initial path at *global* level (that is, considering the whole available map) by refining significantly the original VD roadmap through a number of heuristics. One more new element is that, in addition to land constraints, the sea depth data available in nautical charts are exploited in order to avoid navigating in shallow waters. Successive straight-line segments are connected using Fermat's Spiral (FS) segments, which ensures low computational cost and results in practical and intuitive paths that satisfy the dynamic constraints (curvature continuity and maximum value) of the underactuated vehicle. A methodology for FS path following was also developed and presented for the first time here.

When an obstacle (static or dynamic) is detected and its motion is tracked using the onboard sonar, the algorithm enters a *replanning phase* and generates a path deviation at *local* level that ensures collision avoidance while respecting the COLREGs. Apart from the predicted future trajectory of the obstacle itself, the replanning phase also takes into account the time intervals needed for the algorithm to replan and the vehicle to change its course, so as to produce a useful deviation. The process is very similar to the initial planning phase, but additional heuristics are developed so as to select a sensible map area size wherein to seek path deviations, and to connect this local deviation to the original path smoothly after a collision has been avoided.

The simulated case study focused on a crammed area with many small islands and two (a priori unknown) obstacles, one dynamic and one static. The locations of the obstacles resulted in consecutive double replanning of the local path. The marine vehicle avoided both obstacles and then continued smoothly onto the original path. Without any special effort to optimize the algorithm running time in Matlab, the replanning time never overcame the threshold of 0.6 s. This concerned a map area of approximatively 2000×2000 m^2, hence making this solution very promising for real applications.

Acknowledgments

This work has been carried out at the Centre for Autonomous Marine Operations and Systems (AMOS). The Norwegian Research Council (RCN) is acknowledged as the main sponsor of NTNU AMOS

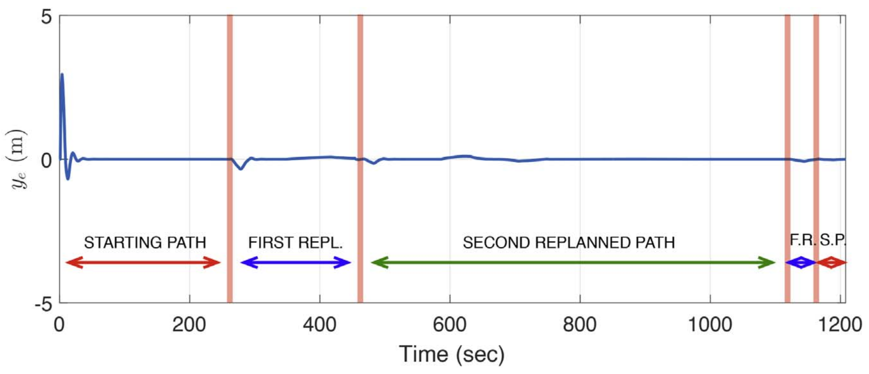


Fig. 16. Cross-track error along the path. Small disturbances (always below 0.4 m) can be observed whenever a replanning process takes place. The initial oscillations are due to the vessel's initial heading, which is not aligned with the path direction at WP_{22}.

through the Centres of Excellence funding scheme, Project number 223254. Christian Malmquist from Kartverket is acknowledged for sharing his great knowledge about maps. The second author was supported by RCN through the project *Next Generation Subsea IMR Operations*, Project number 234108.

References

Almeida, C., Franco, T., Ferreira, H., Martins, A., Santos, R., Almeida, J. M., Carvalho, J., & Silva, E. (May 2009). Radar based collision detection developments on USV ROAZ II. In *Proceedings of MTS/IEEE OCEANS*.

Antonelli, G., Chiaverini, S., Finotello, R., & Schiavon, R. (2001). Real-time path planning and obstacle avoidance for RAIS: An autonomous underwater vehicle. *IEEE Journal of Oceanic Engineering*, 26(2), 216–227.

Antonelli, G., Chiaverini, S., & Marino, A. (May 2012). A coordination strategy for multi-robot sampling of dynamic fields. In *Proceedings of the IEEE International Conference on Robotics and Automation* (*ICRA*) (pp. 1113–1118).

Aurenhammer, F. (1991). Voronoi diagrams – A survey of a fundamental geometric data structure. *ACM Computing Surveys*, 3, 345–404.

Bellingham, J. S., Tillerson, M., Alighanbari, M., & How, J. P. (December 2002). Cooperative path planning for multiple UAVs in dynamic and uncertain environments. In *Proceedings of the 41st IEEE Conference on Decision and Control* (*CDC*) (Vol. 3, pp. 2816–2822).

Borenstein, J., & Koren, Y. (1991). The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation*, 7(June (3)), 278–288.

Bortoff, S. A. (September 2000). Path planning for uavs. In *Proceedings of the American Control Conference* (*ACC*) (Vol. 1, pp. 364–368).

Caharija, W., Pettersen, K. Y., Sørensen, A. J., Candeloro, M., & Gravdahl, J. T. (2014). Relative velocity control and integral line of sight for path following of autonomous surface vessels: Merging intuition with theory. *Journal of Engineering for the Maritime Environment*, 228(March (2)), 180–191.

Caiti, A. (2014). *Motion planning for marine control systems* London: Springer London.

Campbell, S., Naeem, W., & Irwin, G. (2012). A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres. *Annual Reviews in Control*, 36(2), 267–283.

Candeloro, M., Lekkas, A.M., Hegde, J., & Sørensen, A. J. (2016). A 3D dynamic voronoi diagram-based path-planning system for UUVs. In OCEANS 2016 MTS/IEEE Monterey, 1-8.

Candeloro, M., Lekkas, A.M., Sørensen, A. J., & Fossen, T. I. (2013). Continuous curvature path planning using voronoi diagrams and fermat's spirals. *9th IFAC conference on control applications in marine systems* (Vol. 9, issue no. 1, pp. 132–137).

Chandler, P., Rasmussen, S., & Pachter, M. (August 2000). Uav cooperative path planning. In: of Aeronautics, A. I., Astronautics (Eds.), *Proceedings of the AIAA guidance, navigation, and control conference and exhibit*.

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E. et al. (2005). *Principles of robot motion: Theory, algorithms, and implementations* Cambridge, MA: MIT Press.

Dahl, A. R. (2013). *Path planning and guidance for marine surface vessels*. (Master's thesis), Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway.

Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1, 269–271.

Dolgov, D., Thrun, S., M., M., J., D. (January 2010). Path planning for autonomous vehicles in unknown semi-structured environments. *The International Journal of Robotics 29*(5), 485–501.

Elfes, A. (1987). Sonar-based real-world mapping and navigation. *IEEE Journal on Robotics and Automation*, 3(June (3)), 249–265.

Falcone, M., & Ferretti, R. (2013). *Semi-Lagrangian approximation schemes for linear and Hamilton-Jacobi equations* Philadelphia, PA: Society for Industrial and Applied Mathematics.

Fiorini, P. & Shiller, Z. (May 1993). Motion planning in dynamic environments using the relative velocity paradigm. In *Proceedings of the IEEE International Conference on Robotics and Automation* (*ICRA*) (Vol. 1, pp. 560–565).

Fossen, T. I. (2011). *Handbook of marine craft hydrodynamics and motion control* Chichester, West Sussex, U.K: John Wiley and Sons Ltd. http://dx.doi.org/10.1002/9781119994138.

Fossen, T. I. & Lekkas, A. M. (2015). Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents. *International Journal of Adaptive Control and Signal Processing*. ⟨http://dx.doi.org/10.1002/acs.2550⟩

Fox, D., Burgard, W., & Thrun, S. (1997). The dynamic window approach to collision avoidance. *IEEE Robotics Automation Magazine*, 4(March (1)), 23–33.

Gold, C. (2016). Tessellations in GIS: Part II-making changes. *Geo-spatial Information Science*, 19(2), 157–167.

Gold, C., Chau, M., Dzieszko, M., & Goralski, R. (2005). *3D geographic visualization: The marine GIS* Berlin, Heidelberg: Springer, 17–28.

Gold, C.M., 1998. The use of the dynamic Voronoi data structure in autonomous marine navigation. *Proceedings, Advanced Robotics: Beyond 2000*, pp.217-220.

Goralski, I. R. & Gold, C. M. (July 2007). Maintaining the Spatial Relationships of Marine Vessels Using the Kinetic Voronoi Diagram. In *Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering* (*ISVD*) (pp. 84–90).

Hart, P. E., Nilsson, N. J., & Raphael, B. (1968). A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics*, 4(July (2)), 100–107.

Heidarsson, H. K. & Sukhatme, G. S. (May 2011). Obstacle detection and avoidance for an autonomous surface vehicle using a profiling sonar. In *Proceedings of the IEEE international conference on robotics and automation* (*ICRA*) (pp. 731–736).

IMO, 1977. International Maritime Organization - The International Regulations for Preventing Collisions at Sea. ⟨https://goo.gl/7mPxS7⟩, accessed in November 2016.

Karoui, I., Quidu, I., & Legris, M. (2015). Automatic sea-surface obstacle detection and tracking in forward-looking sonar image sequences. *IEEE Transactions on Geoscience and Remote Sensing*, 53(August (8)), 4661–4669.

Khatib, O. (March 1985). Real-time obstacle avoidance for manipulators and mobile robots. In *Proceedings of the IEEE International Conference on Robotics and Automation* (*ICRA*) (Vol. 2, pp. 500–505).

Kongsberg (October 2016). World's first official test bed for autonomous shipping opens in Norway. ⟨https://goo.gl/oEdLk2⟩, accessed in November 2016.

Kuwata, Y., Wolf, M. T., Zarzhitsky, D., & Huntsberger, T. L. (2014). Safe maritime autonomous navigation with COLREGS, using velocity obstacles. *IEEE Journal of Oceanic Engineering*, 39(January (1)), 110–119.

Larson, J., Bruch, M., & Ebken, J. (May 2006). Autonomous navigation and obstacle avoidance for unmanned surface vehicles. Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series 6230, 623007.

Larson, J., Bruch, M., Ebken, J., Rogers, J., & Webster, R. (2007). Advances in autonomous obstacle avoidance for unmanned surface vehicles ⟨www.dtic.mil/cgi-bin/GetTRDoc?AD=ADA475547⟩.

Lavalle, S. M. (November 1998). Rapidly-exploring random trees: A new tool for path planning. Technical report, Computer Science Department, Iowa State University.

LaValle, S. M. (2006). *Planning algorithms* New York, NY, USA: Cambridge University Press.

Ledoux, H. (July 2007). Computing the 3D voronoi diagram robustly: An easy explanation. In *Proceedings of the 4th International Symposium on Voronoi Diagrams in Science and Engineering* (*ISVD*) (pp. 117–129).

Lee, S. M., Kwon, K. Y., & Joh, J. (2004). A fuzzy logic for autonomous navigation of marine vehicles satisfying COLREG guidelines. *International Journal of Control Automation and Systems*, 2(June (2)), 171–181.

Lekkas, A. M., Dahl, A. R., Breivik, M., & Fossen, T. I. (2013). Continuous-curvature path generation using Fermat's spiral. *Modeling, Identification and Control (MIC)*, 34(4), 183–198.

Lekkas, A. M., & Fossen, T. I. (2014). Integral LOS path following for curved paths based on a monotone cubic Hermite spline parametrization. *IEEE Transactions on Control Systems Technology*, 22(6), 2287–2301.

Lekkas, A. M. & Fossen, T. I. (October 2014b). Trajectory tracking and ocean current estimation for marine underactuated vehicles. In *IEEE Conference on Control Applications* (*CCA*) (pp. 905–910).

Lekkas, A. M., Roald, A. L., & Breivik, M. (2016). Online path planning for surface vehicles exposed to unknown ocean currents using pseudospectral optimal control (Vol. 49, pp. 1–7), 10th *IFAC Conference on Control Applications in Marine Systems* (*CAMS*).

Lerro, D., & Bar-Shalom, Y. (1993). Tracking with debiased consistent converted measurements versus EKF. *IEEE Transactions on Aerospace and Electronic Systems*, 29(July (3)), 1015–1022.

Lozano-Pérez, T., & Wesley, M. A. (1979). An algorithm for planning collision-free paths among polyhedral obstacles. *Communications of the ACM (CAMC)*, 22(October (10)), 560–570.

Ludvigsen, M. & Sørensen, A. J. (2016). Towards integrated autonomous underwater operations for ocean mapping and monitoring. *Annual Reviews in Control*. ⟨http://dx.doi.org/10.1016/j.arcontrol.2016.09.013⟩

Marino, A., Antonelli, G., Aguiar, A. P., & Pascoal, A. (October 2012). A new approach to multi-robot harbour patrolling: Theory and experiments. In *2012 IEEE/RSJ international conference on intelligent robots and systems* (pp. 1760–1765).

Marino, A., Antonelli, G., Aguiar, A. P., Pascoal, A., & Chiaverini, S. (2015). A decentralized strategy for multirobot sampling/patrolling: Theory and experiments. *IEEE Transactions on Control Systems Technology*, 23(January (1)), 313–322.

Naeem, W., Irwin, G. W., & Yang, A. (2012). COLREGs-based collision avoidance strategies for unmanned surface vehicles. *Special Issue on Intelligent Mechatronics*, 22(September (6)), 669–678.

Perera, L. P., Ferrari, V., Santos, F. P., Hinostroza, M. A., & Soares, C. G. (2015). Experimental evaluations on ship autonomous navigation and collision avoidance by intelligent guidance. *IEEE Journal of Oceanic Engineering*, 40(April (2)), 374–387.

Quidu, I., Jaulin, L., Bertholom, A., & Dupas, Y. (2012). Robust multitarget tracking in forward-looking sonar image sequences using navigational data. *IEEE Journal of Oceanic Engineering*, 37(July (3)), 417–430.

Seto, M. (2013). *Marine robot autonomy* New York: Springer. http://dx.doi.org/10.1007/978-1-4614-5659-9.

Skjetne, R. (2005). *The maneuvering problem*. (Ph.D. thesis). Trondheim, Norway: Department of Marine Technology, Norwegian University of Science and Technology.

Statheros, T., Howells, G., & Maier, K. M. (2008). Autonomous ship collision avoidance navigation concepts, technologies and techniques. *Journal of Navigation*, 61(January (1)), 129–142.

Tsourdos, A., White, B., & Shanmugavel, M. (2010). *Cooperative path planning of unmanned aerial vehicles* Chichester, West Sussex, U.K: Wiley. http://dx.doi.org/10.1002/9780470974636.

van den Berg, J., Ferguson, D., & Kuffner, J. (May 2006). Anytime path planning and replanning in dynamic environments. In *Proceedings of the IEEE International*

Conference on Robotics and Automation (*ICRA*) (pp. 2366–2371).

Watson, D. P., & Scheidt, D. H. (2005). Autonomous systems. *Johns Hopkins APL Technical Digest*, 26(4).

Woerner, K. (2016). *Multi-contact protocol-constrained collision: Avoidance for autonomous marine vehicles* (Ph.D. thesis). Massachusetts Institute of Technology.

Yen, J. Y. (1970). An algorithm for finding shortest routes from all source nodes to a given destination in general networks. *Quarterly of Applied Mathematics*, 27, 526–530.

Zeng, Z., Liana, L., Sammut, K., He, F., Tang, Y., & Lammas, A. (2015). A survey on path planning for persistent autonomy of autonomous underwater vehicles. *Ocean Engineering*, 110(December (A)), 303–313.