

Cooperative AUV Motion Planning using Terrain Information

Andreas J. Häusler^{*}, Alessandro Saccon[†], António M. Pascoal^{*}, John Hauser[‡], and A. Pedro Aguiar[§]

^{*} Laboratory of Robotics and Systems in Science and Engineering, Instituto Superior Técnico, Lisbon, Portugal
ahaesler@isr.ist.utl.pt, antonio@isr.ist.utl.pt

[†] Department of Mechanical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands
a.sacson@tue.nl

[‡] Electrical, Computer, and Energy Engineering Department, University of Colorado at Boulder, Boulder, Colorado, USA
john.hauser@colorado.edu

[§] Faculdade de Engenharia, University of Porto, Porto, Portugal
pedro.aguiar@fe.up.pt

Abstract—There is widespread interest in the deployment of fleets of marine robots with the potential to roam the oceans freely and collect data at an unprecedented scale. This calls for the development of efficient algorithms for multiple vehicle motion planning that can take directly into account the capabilities of each vehicle as well as the environmental conditions and lend themselves to seamless integration with control and navigation systems. The latter connection is for the most part eschewed in the literature, in spite of the obvious fact that in order for the vehicles to execute the planned motions they must at a later stage navigate with great accuracy and follow the trajectories using control algorithms that take explicitly into account the dynamical constraints of the vehicles involved. Among the methods available for underwater vehicle navigation, terrain-based techniques have recently come to the fore. These techniques avoid the use of overly expensive inertial-like motion sensor units and hold considerable promise for the development of a new breed of affordable long range navigation systems.

Motivated by these considerations, we tackle in the present paper the problem of multiple vehicle motion planning by taking explicitly into consideration inter-vehicle collision avoidance, together with a number of criteria that may include simultaneous times of arrival at assigned target points, energy minimization, acoustic communication constraints, and the maximization of terrain information along the vehicle paths (as measured by some appropriate criterion) for terrain-based navigation purposes.

I. INTRODUCTION

Over the past decade, there has been an exponentially increasing interest and demand for fleets of marine robots that have the capability to exploit the advantages of cooperation among small-scale, low-cost, and heterogeneous vehicles to overcome the disadvantages, and to exceed the performance, of manned submersibles and remotely operated vehicles in high-profile marine applications by roaming the oceans freely and collecting data at an unprecedented scale. This calls for the development of efficient algorithms for multiple vehicle motion planning that achieve seamless integration with control

and navigation systems, and that can take into account the requirements that are at the core of the demand for cooperative motion planning: each vehicle’s (individual and probably unique) capabilities, environmental conditions, and collision avoidance and communication constraints.

A cooperative motion planning system as envisioned here goes hand-in-hand with trajectory tracking mechanisms using control algorithms that take explicitly into account the dynamical constraints of the vehicles involved. This, in turn, demands accurate means of navigation during mission execution. Among the methods available for underwater vehicle navigation, terrain-based techniques have recently come to the fore, see for example [22], [23], [34] and the references therein. Each map-based navigation method, however, in order to achieve accurate feature matching and thus self-localization along a trajectory, can only be as good as the quality of information available to the vehicle at mission execution time. It is therefore important to already include a means of maximizing the obtainable sensor information at the level of trajectory planning. In other words, the goal is to investigate a novel planning framework that incorporates e.g. bathymetric data to plan missions for multiple AUVs such that the trajectories ensure that the vehicle is located over information-rich terrain, while still being “as optimal as possible” in accordance with a given cost criterion, e.g. total energy requirement. The central idea followed in this paper is to employ a pre-planner that uses terrain elevation information to plan a globally optimal path over a given seafloor map, before a trajectory optimization method generates a trajectory that is feasible in terms of the vehicle dynamics and subject to a global cost criterion and probably constraints.

A. Cooperative AUV Motion Planning

In [26], trajectories are designed to meet the scientific objectives of a mission, and adapted in case of vehicle failures. Instead of aiming for single-track trajectories with simultaneous arrival, the objective is rather a closed shape that can be executed repeatedly; if more than one vehicle follows such a trajectory, they need to maintain a fixed distance. Vehicle models are explicitly incorporated, but used only for position

Research supported in part by projects MORPH of the EU FP7 (grant agreement no. 288704) and CONAV of the FCT (contract number PTCDEEA-CRO/113820/2009), and by the FCT Program PEst-OE/EEI/LA0009/2011. The work of A. Häusler was supported by a Ph.D. scholarship of the FCT under grant number SFRH/BD/68941/2010.

prediction; trajectory planning is done using a simplified point-mass model.

The authors of [32] propose optimal control techniques for path planning with respect to minimum energy requirement, where the energy is computed as the mechanical power input to the propeller shaft, which, according to [19], means that the performance index does not exactly represent the energy consumed by the thrusters. By defining path constraints, the trajectories are planned in a way that obstacles are avoided. The vessel model is explicitly incorporated. This is also done in [18], but here, energy-minimal trajectories are computed by assuming the instantaneous power to be constant.

B. Terrain-Based Planning

The vast majority of terrain-based planning literature is related to path planning for unmanned ground vehicles (UGVs) such as the Mars rovers. To the best of our knowledge, the only paper that directly relates to an AUV application of terrain-based planning is [29] where the authors use a particle swarm optimization approach to enable an AUV to do what they call “terrain matching trials” for map building purposes. Terrain information is computed as a “terrain entropy metric” based on pure elevation information.

In our approach, we make use of the Cramér-Rao lower bound (or equivalently, the Fisher Information matrix), essentially taking the terrain gradient as a measure of how rich the terrain is in terms of features available to navigation methods, thus defining the “information content” of a map. This is much in the line of the “traversability index” described in [15] or the “inclination measure” of [5]. In [17], the authors only consider a binary terrain information measure (i.e. if a given patch of terrain can be crossed or not) by putting the terrain variance in relation to the diameter of a UGV’s wheels. As such, the measure is more related to terrain roughness than terrain slope (i.e. fine granular information) and not useful in our context.

Most terrain-based path planners in the literature are, at their core, based on A^* search [27] or a modification of it. In [17], the authors use a model-based evaluation process (for checking the UGV’s configuration, stability, and force conditions) on the result of A^* to determine a path’s feasibility before going into path following mode. A smooth reference trajectory for tracking by an UGV is generated from the result of A^* in [10], using geometric methods such as spline approximation. The authors of [15] use a trade-off between path length and traversability cost to influence the “aggressiveness” of A^* search, i.e. to influence the importance of terrain information to the path planner. Rapidly exploring random trees (RRTs, see [20]) are the method of choice for terrain-based planning in [5]. The authors state that the problem of their implementation of RRT is that they experience a performance decrease in the planning process in the presence of more than one local optimum, which they successfully resolve in [4] by locally invoking further instances of the RRT algorithm.

C. Contribution

The research discussed in this paper builds upon and extends the cooperative autonomous marine vehicle (AMV) motion planning results presented in [13], where we introduce a planning approach that does not only take into account the

full dynamical model of a representative AMV, but also a model of its propulsion system (energy source, motors, and propellers) so as to compute trajectories that optimize the actual energy drawn in the course of a complete maneuver. This work is extended in the current paper in three ways:

- We introduce a means of coordination space planning [20, Sec. 7.2] to ensure an already collision-free initial condition to our main trajectory optimization algorithm, thereby significantly speeding up the computation of an optimal solution to multiple vehicle problems.
- For the planning phase, the vehicle model has two first-order integrators on the system inputs that achieve a prevention of instantaneous turns with unfeasibly high thruster values as part of the resulting optimized trajectories.
- The planning framework is enhanced by adding the capability of generating desired trajectories that are globally optimal with respect to the information content of the terrain the vehicles will cover. The optimizer will, according to its weight settings, then compute an overall solution to the planning problem that is a trade-off between electrical energy spent and the terrain information content, collision-free, and feasible in terms of the vessel dynamical models.

The remainder of this paper is organized as follows: we start by giving a detailed problem formulation in Sec. II, where we also introduce the vehicle model at hand. The planning framework is described in Sec. III with a short background summary of the core trajectory optimization approach, as well as the coordination space and terrain information pre-planners. Simulation results are shown and discussed in Sec. IV, and we conclude the paper with comments and an outlook in Sec. V.

II. PROBLEM FORMULATION

The problem we aim to solve in this work is achieve collision-free trajectories for multiple AUVs, that are feasible in terms of the vehicle dynamics (and thus, executable in the sense of a trajectory tracking approach). The trajectories also have to be optimal in terms of a given cost criterion, which, in our case, is taken as the overall electrical power expected to be consumed per vehicle when executing the resulting set of trajectories. For trajectory tracking and navigation purposes, i.e. self-localization along a planned trajectory, we also want to incorporate known terrain information already at the planning stage. This information may be incorporated into the planning framework as a set of desired trajectories, whose weighted \mathcal{L}_2 distance to the vehicle trajectories may form a further part of the integral cost, and/or a cost-to-go like measure of the terrain information itself along the optimized trajectory, which is the subject of currently ongoing research.

A. Overview of the Complete Planning Process

Before going into further detail in the subsequent sections, we give a brief overview of what we envision as a complete planning framework at a more abstract level. The concept of the framework at hand is illustrated in Fig. 1: the pre-planner uses a bathymetric map to compute a trajectory that yields high terrain excitation for terrain-based navigation purposes. This

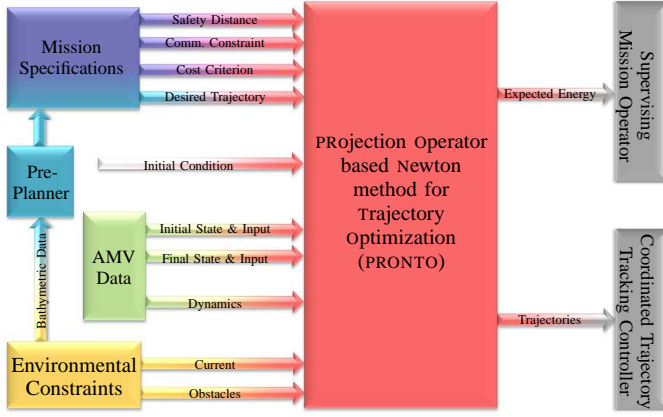


Fig. 1. The trajectory planning framework, including both the terrain-based pre-planner as well as the core trajectory optimizer. In order to ensure a cleaner presentation, the coordination space pre-planner is not included in this figure.

trajectory is the initial condition for a dynamical optimizer at the first iteration (i.e., “desired trajectory”) in the search for an optimal solution that includes tight dynamical constraints as well as collision avoidance requirements.

Further inputs to the core planner are the same initial and final values of the kinematic states as used in the pre-planner, as well as appropriate values of the additional dynamical states and system inputs, and further mission- and environment-related data such as obstacle information or collision avoidance radii. Using an appropriate cost criterion, in our case the electrical energy taken from the batteries, the trajectories are then optimized to achieve a minimum value of the cost function.

B. Vehicle Model

The MEDUSA AMV is a vessel developed at the Dynamical Systems and Ocean Robotics laboratory at ISR/IST in Lisbon. Its hull consists of two torpedo shaped tubes that lie parallel to the water surface, but with a vertical displacement, so that one tube is fully submerged at all times (see Fig. 2). In its MEDUSA_s flavor, it is an autonomous surface vehicle (ASV), whereas the recently developed MEDUSA_d has diving capability and can thus be classified as a “true” AUV.

1) *Kinematic and Dynamic Equations:* Since the MEDUSA is conceptually a semi-submersible, the mathematical model can be considered to be that of a planar vehicle. Using the notation of [6], the model is

$$\dot{\eta} = R(\eta)\nu \quad (1)$$

$$(M_{rb} + M_a)\dot{\nu} + (C_{rb}(\nu) + C_a(\nu))\nu + (D + D_n(\nu))\nu = \tau \quad (2)$$

where the kinematic states $\eta = [x, y, \psi]$ express the vessel’s pose in the inertial reference frame $\{I\}$, and the dynamic state vector $\nu = [u, v, r]$ represents the velocities in the body frame $\{B\}$. In (1), $R(\eta)$ is a rotation matrix such that

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix}$$



Fig. 2. The MEDUSA_s surface craft. The thrusters are mounted in the middle between both hull parts on the x - z -plane (body coordinates) and produce completely from the body on the x - y -plane.

The model inputs n_{ps} and n_{sb} are the rotational velocities of the MEDUSA’s actuators, the port side and starboard propeller, respectively, obtained by multiplying a percentage command with the maximally admissible rotational velocity n_{max} ,

$$n_{ps} = p_{ps}n_{max}$$

$$n_{sb} = p_{sb}n_{max}$$

Maneuvering is done using common and differential thrust, resulting in the external force vector

$$\tau = \begin{bmatrix} T_{ps} + T_{sb} \\ 0 \\ l(T_{ps} - T_{sb}) \end{bmatrix}$$

where l is the displacement of the propellers from the center of $\{B\}$, see Fig. 3. The forces T_{ps} and T_{sb} are functions of u , r , and the two propellers’ rotational velocities, n_{ps} and n_{sb} , respectively, both given in [rps].

Placing the center of $\{B\}$ at the center of mass of the vessel, the rigid body and hydrodynamic added mass matrices in the dynamics (2) can be written as

$$M_{rb} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \quad M_a = - \begin{bmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & 0 \\ 0 & 0 & N_r \end{bmatrix}$$

where m is the body mass and I_{zz} the rigid body inertia. The rigid body and hydrodynamic centripetal and Coriolis matrices are

$$C_{rb}(\nu) = \begin{bmatrix} 0 & -mr & 0 \\ mr & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad C_a(\nu) = \begin{bmatrix} 0 & 0 & -Y_{\dot{v}}v \\ 0 & 0 & X_{\dot{u}}u \\ Y_{\dot{v}}v & -X_{\dot{u}}u & 0 \end{bmatrix}$$

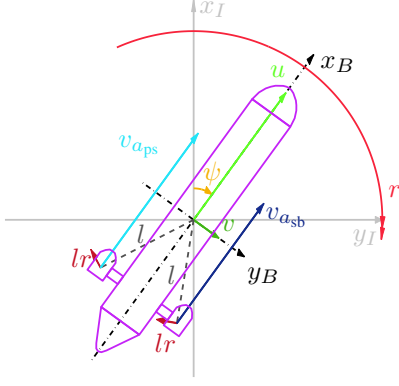


Fig. 3. Conceptual drawing of the MEDUSA as seen from above. The arrows illustrate the velocities experienced at different points of the body for a system that is undergoing a right turn while moving forward. The black coordinate axes represent the body frame $\{B\}$, the grey axes the inertial frame $\{I\}$.

and the linear and nonlinear drag is expressed as

$$D = - \begin{bmatrix} X_u & 0 & 0 \\ 0 & Y_v & 0 \\ 0 & 0 & N_r \end{bmatrix}$$

$$D_n(\nu) = - \begin{bmatrix} X_{|u|u}|u| & 0 & 0 \\ 0 & Y_{|v|v}|v| & 0 \\ 0 & 0 & N_{|r|r}|r| \end{bmatrix}$$

The hydrodynamic derivatives are known, but not given here due to space limitations, and have been obtained by scaling from a different vehicle model, a detailed explanation of which is out of the scope of this paper. Trials are planned in the close future to obtain the exact values of those derivatives for the MEDUSA.

2) *Thrust, Torque, and Input Dynamics:* Since the model inputs are the propellers' rotational velocities n_{ps} and n_{sb} , a mapping between those and the thrust T_{ps} and T_{sb} needs to be defined. Following our argumentation in [13], we use the so-called ‘‘four-quadrant propeller model’’ described in [35], which is valid in all regions of motion (i.e. ahead, back, crash back and crash ahead). The coefficients used by this model are given in terms of the advance angle β at the propeller blade, and data is available in the form of a 20th order Fourier series for various ducted propellers and nozzles [25]. A smoothing procedure that makes the coefficients more usable for Newton descent methods such as the one employed at the core of our trajectory optimization approach is described in [14].

In this propeller model, the thrust and torque equations are

$$T = \frac{1}{2} \rho c_T(\beta) (v_a^2 + v_p^2) \pi R^2 \quad (3)$$

$$Q = \frac{1}{2} \rho c_Q(\beta) (v_a^2 + v_p^2) \pi R^2 d \quad (4)$$

where d is the propeller diameter and $R = d/2$ is its radius. The advance angle can be computed as

$$\beta = \text{atan2}(v_a, v_p) \quad (5)$$

where v_a is the advance velocity of the propeller, and v_p is the lateral velocity of the propeller blade at radius $0.7R$ as a function of the rotational velocity. Figure 4, as well as [1]

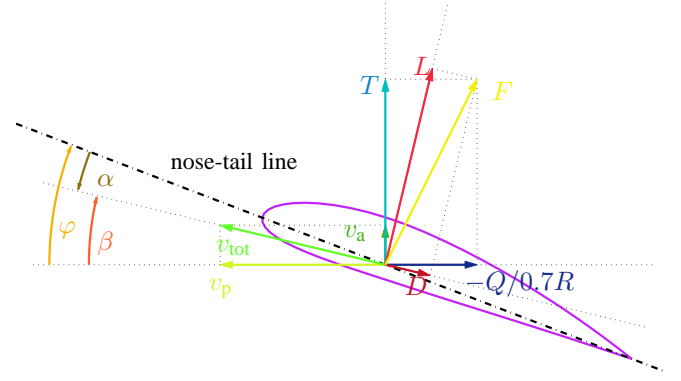


Fig. 4. A cross-section of the propeller blade at $0.7R$ showing the forces and velocities acting on the propeller blade (the force vectors are enlarged). The propeller runs counter-clockwise in order to achieve forward thrust T , moving the system upwards in this picture. To avoid overlapping arrows, the tangential component of the propeller torque Q at $0.7R$ is shown here with a negative sign. Propeller lift L and drag D and total hydrodynamic force F are only shown for reference.

and [14] give an illustration of the concepts involved. In what follows,

$$v_{ps} = 0.7R\omega_{ps} = 0.7R2\pi n_{ps}$$

$$v_{psb} = 0.7R\omega_{sb} = 0.7R2\pi n_{sb} \quad (6)$$

are the propeller lateral velocities of the port side and starboard thruster, respectively. When the vessel is rotating about the z axis, we need to account for the fact that the advance velocities at the propellers are different from each other, which results in the side-dependent expressions

$$v_{aps} = -p_y r + u$$

$$v_{asb} = p_y r + u \quad (7)$$

Here, (p_x, p_y) is the offset of the propellers from the vessel's center of mass in body coordinates, and l is their absolute distance to the center of mass (see Fig. 3). Since (6) and (7) are dependent on the vessel side, it needs to be kept in mind that the advance angle (5), and thus both the thrust (3) and the torque (4) need to be computed separately for each propeller.

The MEDUSA uses Seabotix HPDC 1507 thrusters, which run in a Type 37 Kort nozzle and have a pitch ratio of roughly $P/d \approx 1.4$. A correction factor was applied to achieve the bollard-pull conditions that Seabotix indicates for this thruster model in the following manner: from the manufacturer's specifications, the maximum propeller velocity can be obtained, which is sufficient to compute the values of thrust (3) and torque (4) at bollard-pull conditions (i.e. $v_a = 0.0 \frac{m}{s}$) for the Wageningen Ka 4-70 propeller with $P/d = 1.4$, running in a Type 37 Kort nozzle [25]. These values can be compared with the manufacturer data of maximum continuous thrust and torque at bollard-pull conditions achievable with the Seabotix thrusters, which leads to a multiplicative correction factor for c_T and c_Q . Thanks to the particular body shape of the MEDUSA (see Fig. 2), we may regard propeller-hull interactions as negligible and treat the propellers as if they were running in open water, simply employing the four quadrant model just described.

In order to prevent the optimizer from creating instantaneous turning maneuvers with unachievably high inputs for the vessel model, we do not use the system inputs directly in (3) and (4). Instead, we treat the inputs n_{ps} and n_{sb} as additional states, and instead let the optimizer treat the rate of change of these states as system inputs, defined as:

$$\dot{n}_{ps} = w_c \tanh c_{ps} \quad (8a)$$

$$\dot{n}_{sb} = w_c \tanh c_{sb} \quad (8b)$$

Here, c_{ps} and c_{sb} are the new system inputs, and w_c is a weighting factor. This achieves a transient in the real system inputs n_{ps} and n_{sb} that prevents the optimizer from generating instantaneous turns. The optimizer returns the trajectories of n_{ps} and n_{sb} as model inputs for the solution of the given optimization problem. In total, the MEDUSA vessel model is thus comprised of (1), (2), and (8a) and (8b).

C. Static D/C Motor Equations

It is a common and widely spread approach in the literature to compute energy minimal trajectories subject to the kinetic energy required for an AUV's motion, see e.g [16], [18], [19], [31], [32]. This, however, is only a crude approximation of the energy really required to propulse an AUV along a given trajectory. To that avail, we want the trajectories to be optimal in terms of the actual power taken from the batteries, somewhat in the line of [2]. To this effect, we formulate an additional set of static equations that are based on the D/C motor model for the thrusters, and which will be used as means of computing the required electrical energy along a trajectory to allow for the computation of energy-related quantities that make sense in a physically sound manner.

Since the terminal inductance of the thrusters (Seabotix HPDC 1507) is small ($L_a \approx 500\mu H$), we can neglect the fast dynamics of the electric part of the standard D/C motor equation, [7], and write the relation of the rotational velocities of the propellers to the voltage as

$$V_{ps,sb} = R_a I_{ps,sb} + K_e \omega_{ps,sb} = R_a I_{ps,sb} + K_e 2\pi n_{ps,sb}$$

where $V_{ps,sb}$, $I_{ps,sb}$ and R_a are the armature voltage, current and resistance of the (portside/starboard) D/C motor, K_e is its electrical constant, and $n_{ps,sb}$ its rotational velocity. From measurements conducted with the MEDUSA, we could determine armature current of the port side and starboard thrusters as functions of the propeller rotational velocities—see [13]. Considering the different inputs for port side and starboard propeller, the electrical power consumed by the motors thus is

$$\begin{aligned} P &= V_{ps} I_{ps} + V_{sb} I_{sb} + P_p \\ &= (R_a I_{ps} + K_e n_{ps}) I_{ps} + (R_a I_{sb} + K_e n_{sb}) I_{sb} + P_p \end{aligned} \quad (9)$$

with P_p as the constant power required by the on-board computers.

D. The Optimization Problem

Our goal is to minimize the energy spent by each vehicle when moving from a given initial to a given final configuration. Using (9), the total power consumption of the i^{th} vehicle is

$$\begin{aligned} l_{\text{pow}}(\mathbf{x}^{[i]}(t), \mathbf{u}^{[i]}(t)) &= (R_a I_{ps}^{[i]}(t) + K_e n_{ps}^{[i]}(t)) I_{ps}^{[i]}(t) \\ &+ (R_a I_{sb}^{[i]}(t) + K_e n_{sb}^{[i]}(t)) I_{sb}^{[i]}(t) + P_p \end{aligned} \quad (10)$$

where $\mathbf{x}^{[i]}(t)$ denotes the i -th vehicle's state vector. An additional cost term expressing \mathcal{L}_2 trajectory optimization can be formulated as

$$l_{\text{des}}(\mathbf{x}^{[i]}(t), \mathbf{u}^{[i]}(t), t) = \frac{1}{2} \|\mathbf{x}^{[i]}(t) - \mathbf{x}_{\text{des}}^{[i]}(t)\|_{Q_{to}}^2 + \frac{1}{2} \|\mathbf{u}^{[i]}(t) - \mathbf{u}_{\text{des}}^{[i]}(t)\|_{R_{to}}^2 \quad (11)$$

where Q_{to} and R_{to} are positive definite weight matrices that have to be chosen appropriately, and the subscript $_{\text{des}}$ denotes relation to the ‘‘desired trajectory’’.

The inter-vehicle collision avoidance constraint between vehicles i and j is

$$c_{\text{col}}(\mathbf{x}^{[i]}(t), \mathbf{x}^{[j]}(t)) = \frac{(x^{[i]}(t) - x^{[j]}(t))^2}{(2r_c)^2} + \frac{(y^{[i]}(t) - y^{[j]}(t))^2}{(2r_c)^2} - 1 \quad (12)$$

where r_c is the minimum safety distance that must be kept between two vehicles. The obstacle avoidance constraint $c_{\text{obs}}(\mathbf{x}^{[i]}(t), \mathbf{o}^{[k]})$ between vehicle i and obstacle k is can be formulated in a similar manner. (Here, $k \in \{1, \dots, N_o\}$, where N_o is the total number of obstacles in the environment.) Implementation-relevant details of how to deal with a generic number of constraints, including indexing with tensor-like constructs, are given in [28].

Obviously, (12) is based on the \mathcal{L}_2 norm, while it was shown in [24] that the Sobolev H^1 norm gives better results for multiple encounters with obstacles per trajectory in the presence of uncertainties on the vehicle and obstacle positional information. However, since we use one collision avoidance function *per obstacle* and *for each trajectory*, instead of an overall function that includes all obstacles for each trajectory, we can capture multiple encounters easily with the \mathcal{L}_2 norm.

Using the power function (10), the desired trajectory function (11), the inter-vehicle collision avoidance function (12), and the obstacle avoidance function, the vehicle trajectories are obtained by solving the optimization problem

$$\begin{aligned} \min \int_0^T \sum_{i=1}^{N_v} & \left(l_{\text{pow}}(\mathbf{x}^{[i]}(\tau), \mathbf{u}^{[i]}(\tau)) + \right. \\ & \left. l_{\text{des}}(\mathbf{x}^{[i]}(\tau), \mathbf{u}^{[i]}(\tau), \tau) \right) d\tau + m(x(T)) \\ \text{s.t. } \dot{\mathbf{x}}^{[i]} &= f(\mathbf{x}^{[i]}, \mathbf{u}^{[i]}, t), \quad \mathbf{x}^{[i]}(0) = \mathbf{x}_0^{[i]}, \mathbf{x}^{[i]}(T) = \mathbf{x}_f^{[i]} \\ c_{\text{col}}(\mathbf{x}^{[i]}(t), \mathbf{x}^{[j]}(t)) &\geq 0, \quad i \neq j \\ c_{\text{obs}}(\mathbf{x}^{[i]}(t), \mathbf{o}^{[k]}) &\geq 0 \end{aligned}$$

with $i, j \in \{1, \dots, N_v\}$ and $k \in \{1, \dots, N_o\}$, and $\mathbf{x}_0^{[i]}$ and $\mathbf{x}_f^{[i]}$ are the initial and final condition on the i -th vehicle.

III. THE PLANNING FRAMEWORK

Our trajectory generation and optimization is shown as conceptual overview in Fig. 1. The execution flow can be categorized in three main stages, the last of which may be repeated in the presence of optimization constraints, e.g. inter-vehicle collision avoidance as in (12). Those stages are:

Stage 1— A desired trajectory is generated for each vessel separately as part of a terrain-based pre-planning

step. These paths are usually not feasible in terms of the vessel dynamics and other optimization constraints.

Stage 2— To obtain a valid initial condition for the main planning process, the paths are then projected onto the trajectory manifold by virtue of (17). Furthermore, in case of conflicting paths from Stage 1, the optimization process of Stage 3 can be significantly sped up by computing a timing law that ensures temporal deconfliction [12].

Stage 3— The main trajectory optimization is done within the PRONTO toolkit, iteratively computing an optimal solution to the planning problem by use of Newton’s descent method. If required, this last stage can be repeated after changing parameters related to the constraint functions, thus achieving a tighter fit to the energy-minimal optimization criterion.

A. Terrain-Based Path Planning

In order to explicitly incorporate terrain-based information in the process of multiple vehicle motion planning, we avail ourselves of results available in the literature that emerge out of examining the Cramér-Rao lower bound (or equivalently, the Fisher Information matrix), that yield measures of how accurately the position of a vehicle along a trajectory can be obtained with any unbiased estimator [36]. Stated in simple terms, the larger the magnitude of the gradient of the terrain along a given trajectory, the better the accuracy with which its position can be estimated. In line with this observation, we propose to solve the multiple vehicle motion planning problem in its first stage with a pre-planning mechanism based on a simplified kinematic vehicle model, and using a graph-based search method that operates on a terrain-related cost criterion to be defined. The solution of that planning step is then projected onto the manifold of feasible system trajectories defined by the full dynamical model(s) of the vessel(s).

In the current implementation, this first stage involves a version of A* search [3] in an 8-neighbor implementation [8], adequately modified to take into account positional and angular information in relation to a geophysical map, in order to obtain, for each vehicle, a path between the vehicle’s initial and final kinematic boundary conditions that maximizes a cost criterion related to the integral of the magnitude of the terrain gradient along its trajectory. The terrain information is represented as point cloud, i.e. a 2D array whose cells contain elevation data (the z value of the coordinate triplet (x, y, z)). The row and column numbers of each cell correspond to the values of x and y in a localized coordinate frame (and can easily be mapped to e.g. UTC, if desired); terrain information content is then computed as [33]

$$M_I(x, y) = \frac{1}{\sigma^2} \frac{\|\nabla M_B(x, y)\|}{\max_{x', y'} \nabla M_B(x', y')} \quad (13)$$

where we assume local coordinates $x \in [1, 2, \dots, x_{\max}]$ and $y \in [1, 2, \dots, y_{\max}]$, and where $\nabla M_B(x, y)$ denotes the gradient of the bathymetric map. The original terrain map is given as $M_B(x, y)$, probably with an associated standard deviation σ , and the information map is $M_I(x, y)$; low values of $M_I(x, y)$ are related to a low information content of the terrain. We chose $\sigma = 1$ in (13), which is exactly what is done in [5]

for their “inclination measure”, defining some global “obstacle-ness” of the terrain.

In a second step, using a desired grid resolution r_x and r_y in x and y (it usually makes sense to select $r_x = r_y$), the terrain information map M_I can be downsampled to get what we call the *excitation map* by using the arithmetic mean of an area of size $r_x \times r_y$ as

$$M_E(x, y) = \frac{1}{r_x r_y} \sum_{\substack{x' \in [\lfloor \frac{x}{r_x} \rfloor r_x, \lceil \frac{x}{r_x} \rceil r_x \\ y' \in [\lfloor \frac{y}{r_y} \rfloor r_y, \lceil \frac{y}{r_y} \rceil r_y]} M_I(x', y') \quad (14)$$

A final step in our approach is the computation of a “cost map” M_C defined as

$$M_C(x, y) = \cos\left(\frac{\pi}{2} M_E(x, y)\right)$$

based on the observation that high values of $M_E \in [0, 1]$ are associated with high values of terrain information content, whereas, for a proper cost function, we want to have the contrary. In addition to that, the mapping of M_E to terrain information is linear, and it might be useful for the cost to exhibit a nonlinear behavior, i.e. being high for a wider range of “undesirable” terrain excitation, and then decreasing quickly to zero for high terrain excitation. The cosine function has both desired properties on the interval $[0, \frac{\pi}{2}]$. A* search on $M_C(x, y)$ leads to a set of globally optimal (in terms of terrain information content) paths for a given multiple AUV planning problem.

The Euclidean distance between the current and the final position is in this context chosen as heuristic function for A*, and the data represented by the map is properly scaled so that the Euclidean distance becomes an admissible and consistent heuristic [27]. This is simply possible by adding an offset to a scaled version of $M_C(x, y)$:

$$M_C(x, y) = w + w M_C(x, y) \quad (15)$$

where $w \geq \sqrt{2}$. To achieve a more accessible range of values, we chose $w = 10$.

It is relevant to point out that the trajectories generated in stage 1 are not necessarily twice differentiable, which precludes their use in a normal second order descent dynamical optimization method. By virtue of the underlying Projection Operator based Newton method for Trajectory Optimization (PRONTO, see the last part of this section), however, the trajectories are simply used as “desired trajectories” and feasible ones (closest to the original in the \mathcal{L}_2 sense) that verify the dynamical equations of motion of the vehicles are obtained by projecting the trajectories onto the so-called trajectory manifolds of the different vehicle dynamical models.

B. Temporal Deconfliction on the Coordination Space

To achieve an initial condition to the trajectory optimization algorithm that is already free of collisions, we follow [9] in making use of the coordination space formulation. For an in-depth treatment of this concept, we refer the reader to [20, Sec. 7.2]; in a rough outline, the approach can be described as planning on a N_v -dimensional space whose dimensions represent the spatial or temporal coordinates of the trajectories in

question (N_v is the number of vehicles involved). The original set of paths is a (hyper-)line that crosses this “coordination space” diagonally. Collisions among pairs of vehicles (either in space or in time) are represented as (hyper-)cylinders crossing that space—see the excellent illustration in [20, Fig. 7.9] for an extended treatise of that idea.

A planning algorithm can then be used to find a path through the coordination space connecting the two diagonal extrema, but without traversing through any (hyper-)cylinder. The solution of this planning problem is a new trajectory traversal law that guarantees that the vehicles are coordinated in such a manner that collisions are avoided.

Currently, we use a weighted A^* search on the coordination space to find the globally optimal solution to the temporal deconfliction problem. We again chose the Euclidean distance as heuristic for this planning problem, where $w \geq \sqrt{N_v}$. Weighting the A^* search speeds up the problem solving to be efficiently enough for up to 4 vehicles and trajectories of a length of 60 seconds with a gridding of 0.01 seconds. Since there is no necessity of the deconfliction law to be the globally optimal solution, however, one can easily think about speeding up the process further by the use of R^* [21], [37] or a bidirectional RRT approach [20], for instance. Because there is only one call to the deconfliction pre-planner, this is currently not a highly prioritized issue.

C. Projection Operator Trajectory Optimization

In the third stage, the main trajectory optimization algorithm takes over control of the optimization process, refining the trajectories of the vehicles by taking explicitly into account their dynamics, together with appropriate constraints imposed by collision avoidance and communication requirements (in a simplified set-up, the latter may for example capture the fact that the distance between selected pairs of vehicles should not exceed a fixed value). This stage is firmly rooted in solid dynamical optimization theory and borrows from previous collective work by the authors. In the optimizer developed, not only do we take into account the full dynamical model of a representative AMV, but also a model of its propulsion system (energy source, motors, and propellers) so as to compute trajectories that optimize the actual energy drawn in the course of a complete maneuver [13].

In the remainder of this section, we give a quick introduction to the concepts involved. Our approach to the solution of optimal control problems is a *projection operator* method that allows expressing the dynamically constrained optimization problem

$$\begin{aligned} & \text{minimize} && \int_0^T l(x(\tau), u(\tau), \tau) d\tau + m(x(T)) \\ & \text{subject to} && \dot{x}(t) = f(x(t), u(t), t), \quad x(0) = x_0 \end{aligned} \quad (16)$$

as an unconstrained one, and use Newton’s method to find an optimal solution. This is centered around the realization that a trajectory tracking controller defines a function space *operator* that maps a desired trajectory (a *curve*) to a system *trajectory* (an element of the trajectory manifold). Composing the optimization objective (a *functional*) with the (trajectory tracking) *projection operator* converts the dynamically constrained optimal control problem into an essentially unconstrained optimization problem.

Suppose that $\xi(t) = (\alpha(t), \mu(t))$, $t \geq 0$, is a bounded curve (e.g., an approximate trajectory of f) and let $\eta(t) = (x(t), u(t))$, $t \geq 0$, be the trajectory of f determined by the nonlinear feedback system

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t), t), & x(0) &= x_0 \\ u(t) &= \mu(t) + K(t)(\alpha(t) - x(t)) \end{aligned}$$

This feedback system defines a continuous, nonlinear *projection operator*

$$\mathcal{P} : \xi = (\alpha(\cdot), \mu(\cdot)) \mapsto \eta = (x(\cdot), u(\cdot)) \quad (17)$$

This allows us to formulate the following algorithm for infinite-dimensional optimization, similar to the Newton method for optimization of a function $g(\cdot)$, e.g., in finite dimensions:

PROJECTION OPERATOR NEWTON METHOD

- 1 **Init** initial trajectory $\xi_0 \in \mathcal{T}$
- 2 **for** $k = 0, 1, 2, \dots$
- 3 **do** design feedback $K(\cdot)$ defining \mathcal{P} about ξ_i
- 4 search direction
- 4 $\zeta_i = \arg \min_{\zeta \in \mathcal{T}_{\xi_i}} \mathcal{T} Dh(\xi_i) \cdot \zeta + \frac{1}{2} D^2 g(\xi_i) \cdot (\zeta, \zeta)$
- 5 step size $\gamma_i = \arg \min_{\gamma \in (0, 1]} g(\xi_i + \gamma \zeta_i)$
- 6 update $\xi_{i+1} = \mathcal{P}(\xi_i + \gamma_i \zeta_i)$

where \mathcal{T} is the trajectory manifold, $\xi \in \mathcal{T}$, and $g(\xi) := h(\mathcal{P}(\xi))$ with

$$h(\xi) := \int_0^T l(\alpha(\tau), \mu(\tau), \tau) d\tau + m(\alpha(T))$$

Since our problem formulation demands having additional collision avoidance constraints, we have in fact a constrained optimal control problem

$$\begin{aligned} & \min \int_0^T l(x(\tau), u(\tau), \tau) d\tau + m(x(T)) \\ & \text{s.t. } \dot{x}(t) = f(x(t), u(t), t), \quad x(0) = x_0 \\ & \quad c_j(x(t), u(t), t) \geq 0, \quad t \in [0, T], j \in \{1, \dots, k\} \end{aligned} \quad (18)$$

We incorporate the constraints $c_j(\cdot)$ using the barrier functional method introduced in [11]. The method requires the approximate log barrier function $\tilde{\beta}_\delta(\cdot)$, $0 < \delta \leq 1$ defined as

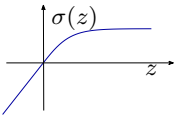
$$\tilde{\beta}_\delta(z) = \begin{cases} -\log z & z > \delta \\ \frac{k-1}{k} \left[\left(\frac{z-k\delta}{(k-1)\delta} \right)^k - 1 \right] - \log \delta & z \leq \delta \end{cases} \quad (19)$$

where $k > 1$ is an even integer. This allow us to express (18) in the shape of (16) as

$$\begin{aligned} & \min \int_0^T \left(l(x(\tau), u(\tau), \tau) \right. \\ & \quad \left. + \epsilon \sum_{j=1}^k \tilde{\beta}_\delta(c_j(x(\tau), u(\tau), \tau)) \right) d\tau + m(x(T)) \\ & \text{s.t. } \dot{x}(t) = f(x(t), u(t), t), \quad x(0) = x_0 \end{aligned}$$

where ϵ and δ express the “sharpness” of the barrier and are, over successive calls to the trajectory optimization algorithm (where the previously attained locally optimal result is used as initial condition for the next iteration of the problem solving process), are simultaneously reduced to force the trajectories into the valid region of the optimization space.

We found it necessary to slightly modify (19), since it assumes (unbounded) negative values for $z > 1$. This may result in a domination of the negative part in the cost integral, thereby, in relation to our application, effectively putting a *reward* on staying away from the other vehicle (or obstacle) as far as possible in addition to the cost for avoiding collisions. Since this is not desirable in minimum energy scenarios, and counterproductive in scenarios involving desired trajectories, we extended (19) by forming a composition with the \mathcal{C}^2 -smooth “hockey stick” function

$$\sigma(z) = \begin{cases} \tanh(z) & \text{if } z \geq 0 \\ z & \text{otherwise} \end{cases}$$


defining the new barrier functional

$$\beta_\delta(z) := \tilde{\beta}_\delta(\sigma(z))$$

that behaves as the standard barrier function $\tilde{\beta}_\delta$ for small and negative z , but goes to zero for $z \rightarrow \infty$ [28]. Using this barrier functional, we can express the constrained optimization problem in Sec. II-D as an unconstrained problem with the integral cost

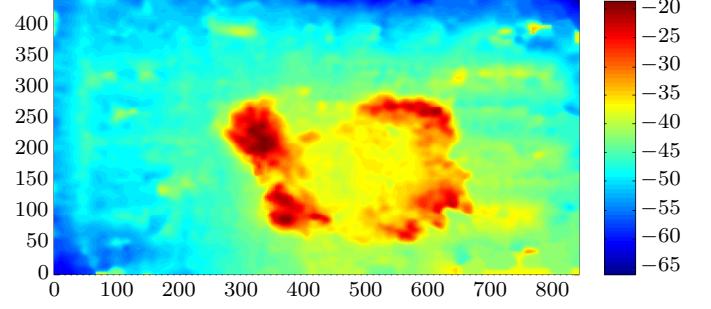
$$\begin{aligned} & \sum_{i=1}^{N_v} l_{\text{pow}}(\mathbf{x}^{[i]}(t)) + \\ & \sum_{i=1}^{N_v-1} \sum_{j=i+1}^{N_v} \epsilon_{\text{col}} \beta_{\delta_{\text{col}}} \left(c_{\text{col}}(\mathbf{x}^{[i]}(t), \mathbf{x}^{[j]}(t)) \right) + \\ & \sum_{i=1}^{N_v} \epsilon_{\text{obs}} \beta_{\delta_{\text{obs}}} \left(c_{\text{obs}}(\mathbf{x}^{[i]}(t)) \right) \end{aligned}$$

with a probably different set of ϵ and δ for collision and obstacle avoidance.

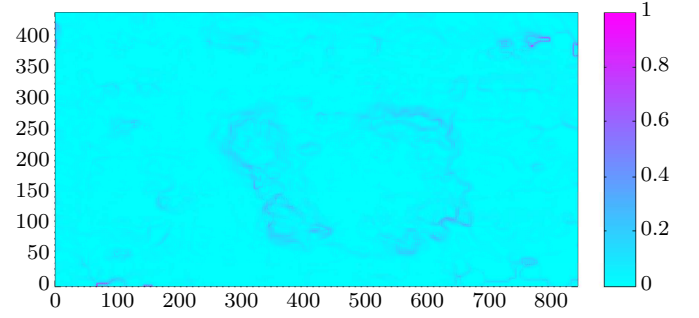
IV. SIMULATION RESULTS

In what follows, we give an overview of the different stages of the planning process using a typical planning scenario, in which two vehicles have to cross an area where the “information content” is highly concentrated along a underwater ridge, while other areas of the map are relatively featureless. The area in question is the D. João de Castro seamount close to the Azores islands in the Atlantic ocean, see Fig. 5a.

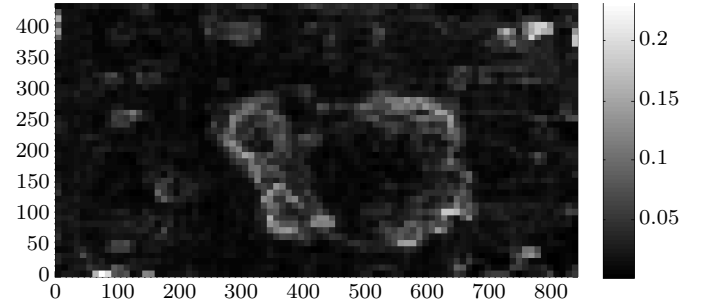
After computing the information map, the downsampled excitation map, and finally the nonlinear cost map (see Fig. 5), A^* is used to compute a set of globally optimal paths. Figure 6, shows the solution to the energy-optimal trajectory planning problem for two AUVs going from initial to a final pose while keeping its actual path closest to the one that yields good terrain-based information for navigation purposes. The result of the path pre-planner is indicated as pink colored lines. Although a portion of about 100 meter length of both paths is initially overlapping in space as well as in time, the coordination space pre-planner ensures temporal deconfliction: the green vehicle slows down and always stays behind the blue one.



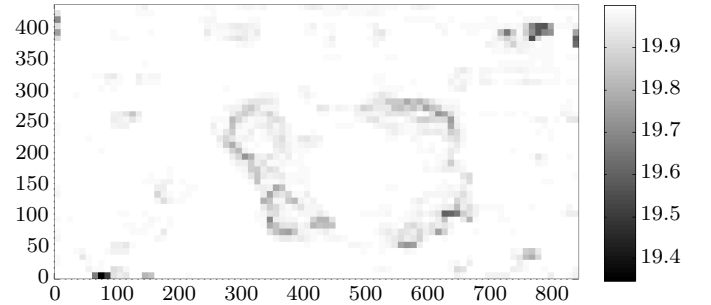
(a) The D. João de Castro seamount used as $M_B(x, y)$.



(b) The obtained information map $M_I(x, y)$.



(c) The excitation map $M_E(x, y)$ after downsampling.



(d) The nonlinear cost map $M_C(x, y)$.

Fig. 5. The stages of pre-processing the terrain map to achieve a map that can be used for terrain-based path planning. The cost map $M_C(x, y)$ shown here is already scaled as described in Sec. III-A, using a value of $w = 10$, thus essentially representing a “scaled shift” of the original range of values by one order of magnitude.

The values of x and y in these maps are in local coordinates; the colored values of the seafloor map represent actual depth. The units are meters.

V. CONCLUSION

This paper introduced a multiple AUV trajectory planning approach, taking explicitly into account the vehicle dynamics, and achieving trajectories that are minimal in terms of energy usage with a trade-off that ensures that the vehicles stay over terrain that yields high sensor excitation and thus aids in terrain-based navigation along the planned trajectories. In comparison to previous work, we enhanced the approach in three ways: 1) a terrain-based pre-planner computes paths that are globally optimal with respect to a terrain information measure and can be used as desired trajectories in the trajectory optimization part of the approach; 2) a coordination map planner guarantees a collision-free initial condition to the trajectory optimizer and thus significantly speeds up computing a solution to the multiple AUV trajectory planning problem; and 3) added thruster dynamics ensure that the planner achieves feasible model inputs. The problem can of course be enriched by considering that navigation may be done by resorting to geophysical data such as that related to the geomagnetic field of Earth.

Our future research will be directed towards several different goals:

- Terrain information so far is only used in the pre-planning step of our approach. For the future, we aim to include the terrain information content in the core optimization approach as well, thus achieving a finer level of control over the trade-off between maximizing the terrain information content and minimizing the energy used along the trajectories. In addition to that, the inclusion of communication network data could also lead to trajectories where one vehicle is going over terrain with sufficiently high information content, while the other might e.g. be crossing the D. João de Castro map in the relatively information-less middle by virtue of localization information exchanged among the vehicles.
- Regardless whether we discretize the terrain information content by (14) or not, we essentially are working on non-continuous “elevation plateaus”. This is of no further importance to the pre-planner, but if the terrain information is to be used within PRONTO, we do have to think about approximating the terrain (or the excitation map) in e.g. a similar manner as [30], i.e. a cubic B-spline mesh.
- In our current implementation, we use the discretization grid (14) to shape a “terrain graph” $\mathcal{G} = (V, E)$ upon which A^* acts, with the set of vertices V defined as

$$V = \left\{ p_{ij} \mid i \in \left[0, \left\lceil \frac{x}{r_x} \right\rceil + 1 \right], j \in \left[0, \left\lceil \frac{y}{r_y} \right\rceil + 1 \right] \right\}$$

and edges E defined as

$$E = \left\{ (p_{ij}, p_{kl}) \mid \begin{aligned} & i \in \left[0, \left\lceil \frac{x}{r_x} \right\rceil \right], \\ & k \in \left[1, \left\lceil \frac{x}{r_x} \right\rceil + 1 \right], j \in \left[0, \left\lceil \frac{y}{r_y} \right\rceil \right], \\ & l \in \left[1, \left\lceil \frac{y}{r_y} \right\rceil + 1 \right], i + 1 = k, j + 1 = l \end{aligned} \right\}$$

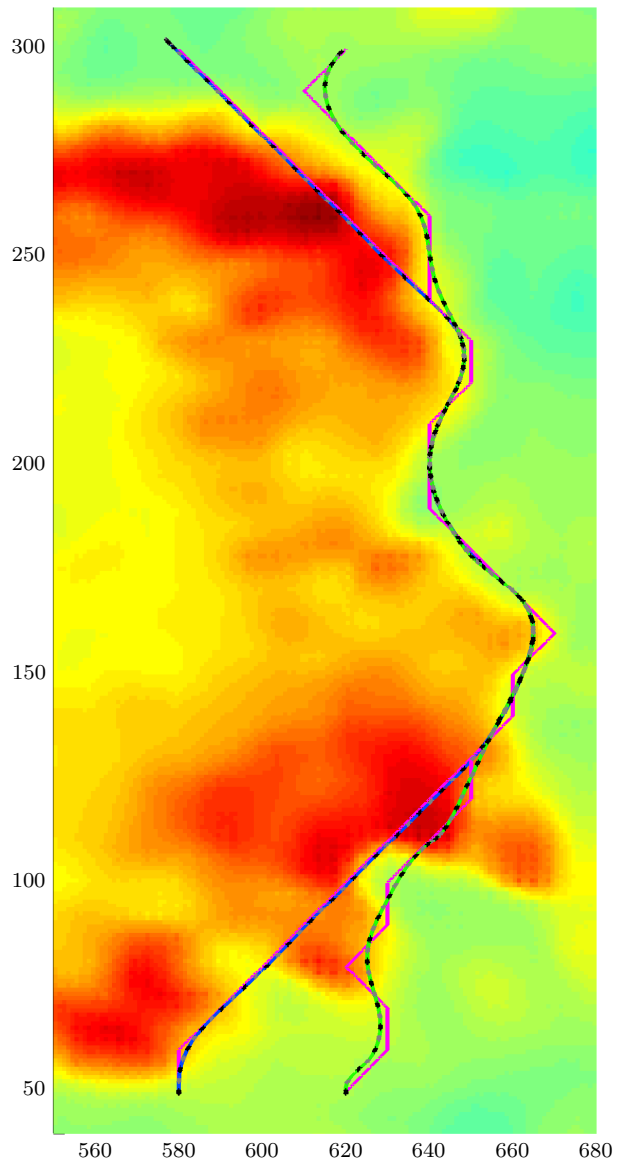


Fig. 6. Result of the trajectory optimization process for two vessels going from south to north over the eastern rim of the D. João de Castro seamount. The initially specified minimum energy problem would have the vessels go in straight, parallel lines. Making use of the terrain information lets the vehicles go over more “information rich” terrain.

and the edge weight function w defined as

$$w((p_{ij}, p_{kl})) = \hat{T} \left(\frac{i+k}{2} r_x, \frac{j+l}{2} r_y \right)$$

which means that the graph represents an evenly spaced grid with horizontal, vertical, and diagonal connections between adjacent nodes, and an edge weight that is the excitation value of the patch of M_E that is crossed by that edge. In other words, this weight function currently assigns the excitation value of adjacent cells of $M_E(x, y)$ to horizontal and vertical edges, not the value of the crossed terrain (15).

This is a more or less minor issue, depending on the resolution of the excitation map (14), but with a “crude enough” approximation (that might be required with big maps), this might become an issue.

- Further work will aim at including extra operational constraints in line with the work pursued in [38].
- The avoidance of static obstacles, especially determining if they better be circumnavigated on the relative “right” or on the relative “left”, might be made easier by including the obstacles at the pre-planning phase already, as done e.g. in [30], or by an infinite terrain cost as e.g. in [5].

Finally, a major step aimed for in the close future is the implementation and test of the proposed algorithm on the MEDUSA platform during sea trials.

REFERENCES

- [1] R. Bachmayer, L. L. Whitcomb, and M. A. Grosenbaugh, “An Accurate Four-Quadrant Nonlinear Dynamical Model for Marine Thrusters: Theory and Experimental Validation,” *IEEE Journal of Oceanic Engineering*, vol. 25, no. 1, pp. 146–159, 2000.
- [2] M. Chyba, T. Haberkorn, R. N. Smith, and S. K. Choi, “Autonomous Underwater Vehicles Development and Implementation of Time and Energy Efficient Trajectories,” *Ship Technology Research*, vol. 1, no. 55, pp. 36–48, 2008.
- [3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009.
- [4] A. Ettl and H. Bleuler, “Randomised Rough-Terrain Robot Motion Planning,” in *International Conference on Intelligent Robots and Systems*, 2006, pp. 5798–5803.
- [5] —, “Rough-Terrain Robot Motion Planning based on Obstacle-ness,” in *9th International Conference on Control, Automation, Robotics and Vision*, 2006, pp. 1–6.
- [6] T. I. Fossen, *Marine Control Systems: Guidance, Navigation, and Control of Ships, Rigs, and Underwater Vehicles*. Trondheim and Norway: Marine Cybernetics, 2002.
- [7] G. F. Franklin, J. D. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 4th ed. Upper Saddle River and NJ: Prentice Hall, 2002.
- [8] D. B. Gennery, “Traversability Analysis and Path Planning for a Planetary Rover,” *Autonomous Robots*, vol. 6, no. 2, pp. 131–146, 1999.
- [9] Y. Guo and L. E. Parker, “A Distributed and Optimal Motion Planning Approach for Multiple Mobile Robots,” in *International Conference on Robotics and Automation*, 2002, pp. 2612–2619.
- [10] Y. Guo, L. E. Parker, D. Jung, and Z. Dong, “Performance-Based Rough Terrain Navigation for Nonholonomic Mobile Robots,” in *29th Annual Conference of the IEEE Industrial Electronics Society*, vol. 3, 2003, pp. 2811–2816.
- [11] J. Hauser and A. Saccon, “A Barrier Function Method for the Optimization of Trajectory Functionals with Constraints,” in *45th Conference on Decision and Control*, 2006, pp. 864–869.
- [12] A. J. Häusler, R. Ghabcheloo, A. M. Pascoal, and A. P. Aguiar, “Multiple Marine Vehicle Deconflicted Path Planning with Currents and Communication Constraints,” in *7th Symposium on Intelligent Autonomous Vehicles*, 2010.
- [13] A. J. Häusler, A. Saccon, A. P. Aguiar, J. Hauser, and A. M. Pascoal, “Cooperative Motion Planning for Multiple Autonomous Marine Vehicles,” in *9th Conference on Maneuvering and Control of Marine Craft (MCMC)*, 2012.
- [14] A. J. Healey, S. M. Rock, S. Cody, D. Miles, and J. P. Brown, “Toward an Improved Understanding of Thruster Dynamics for Underwater Vehicles,” in *Symposium on Autonomous Underwater Vehicle Technology*, 1994, pp. 340–352.
- [15] A. Howard, H. Seraji, and B. Werger, “A Terrain-Based Path Planning Method for Mobile Robots,” in *7th International Conference on Automation Technology*, 2003.
- [16] P.-M. Hsu and C.-L. Lin, “Optimal Planner for Lawn Mowers,” in *9th International Conference on Cybernetic Intelligent Systems*, 2010, pp. 1–7.
- [17] K. Iagnemma, F. Genot, and S. Dubowsky, “Rapid Physics-Based Rough-Terrain Rover Planning with Sensor and Control Uncertainty,” in *International Conference on Robotics and Automation*, 1999.
- [18] T. Inanc, S. C. Shadden, and J. E. Marsden, “Optimal Trajectory Generation in Ocean Flows,” in *American Control Conference*, 2005, pp. 674–679.
- [19] R. P. Kumar, A. Dasgupta, and C. S. Kuma, “Real-Time Optimal Motion Planning for Autonomous Underwater Vehicles,” *Ocean Engineering*, vol. 32, no. 11-12, pp. 1431–1447, 2005.
- [20] S. M. LaValle, *Planning Algorithms*. Cambridge and New York: Cambridge University Press, 2006.
- [21] M. Likhachev and A. Stentz, “R* Search,” in *23rd Conference on Artificial Intelligence*, 2008.
- [22] P. Maurya, F. Teixeira, and A. M. Pascoal, “Complementary Terrain/Single Beacon-Based AUV Navigation,” in *Workshop on Navigation, Guidance, and Control of Underwater Vehicles*, 2012.
- [23] D. Meduna, S. M. Rock, and R. McEwen, “Closed-Loop Terrain Relative Navigation for AUVs with Non-Inertial Grade Navigation Sensors,” in *Autonomous Underwater Vehicles*, 2010.
- [24] S. Nair and M. Kobilarov, “Collision Avoidance Norms in Trajectory Planning,” in *American Control Conference*, 2011, pp. 4667–4672.
- [25] M. W. C. Oosterveld, “Wake Adapted Ducted Propellers,” Ph.D. dissertation, Delft University of Technology, Wageningen and The Netherlands, 1970.
- [26] D. Paley, F. Zhang, and N. Leonard, “Cooperative Control for Ocean Sampling: The Glider Coordinated Control System,” *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, pp. 735–744, 2008.
- [27] S. J. Russell, P. Norvig, and E. Davis, *Artificial Intelligence: A Modern Approach*, 3rd ed. Upper Saddle River: Prentice Hall, 2010.
- [28] A. Saccon, A. P. Aguiar, A. J. Häusler, J. Hauser, and A. M. Pascoal, “Constrained Motion Planning for Multiple Vehicles on SE(3),” in *51st Conference on Decision and Control*, 2012.
- [29] J. Shen, J. Zhang, and H. Li, “Route Planning for Underwater Terrain Matching Trial based on particle Swarm Optimization,” in *2nd International Conference on Computational Intelligence and Natural Computing*, 2010.
- [30] Z. Shiller, “Obstacle Traversal for Space Exploration,” in *International Conference of Robotics and Automation*, 2000, pp. 989–994.
- [31] I. Spangelo and O. Egeland, “Computing energy-optimal trajectories for an autonomous underwater vehicle using direct shooting,” *Modeling, Identification and Control: A Norwegian Research Bulletin*, vol. 13, no. 3, pp. 163–174, 1992.
- [32] —, “Trajectory Planning and Collision Avoidance for Underwater Vehicles Using Optimal Control,” *IEEE Journal of Oceanic Engineering*, vol. 19, no. 4, pp. 502–511, 1994.
- [33] F. Teixeira, “Terrain-Aided Navigation and Geophysical Navigation of Autonomous Underwater Vehicles,” Ph.D. dissertation, Instituto Superior Técnico, Lisboa (Portugal), 2007.
- [34] F. Teixeira, A. M. Pascoal, and P. Maurya, “A Novel Particle Filter Formulation with Application to Terrain-Aided Navigation,” in *Workshop on Navigation, Guidance, and Control of Underwater Vehicles*, 2012.
- [35] W. P. A. van Lammeren, J. D. van Manen, and M. W. C. Oosterveld, “The Wageningen B-Screw Series,” *Transactions of SNAME*, vol. 77, pp. 269–317, 1969.
- [36] H. L. van Trees, *Detection, Estimation, and Modulation Theory: Part I: Detection, Estimation, and Linear Modulation Theory*. John Wiley & Sons, Inc., 2001.
- [37] P. Vernaza, M. Likhachev, R. Bhattacharya, A. Kushleyev, D. D. Lee, and S. Chitta, “Search-Based Planning for a Legged Robot over Rough Terrain,” in *International Conference on Robotics and Automation*, 2009.
- [38] J. S. Willcox, J. G. Bellingham, Y. Zhang, and A. B. Baggeroer, “Performance Metrics for Oceanographic Surveys with Autonomous Underwater Vehicles,” *IEEE Journal of Oceanic Engineering*, vol. 26, no. 4, pp. 711–725, 2001.