Trajectory Optimization for Underwater Vehicles in Time-Varying Ocean Flows

Miguel Aguiar*, João Borges de Sousa*, João Miguel Dias‡, Jorge Estrela da Silva†,
Renato Mendes‡§ and Américo S. Ribeiro‡
*LSTS, FEUP & §CIIMAR, UP, Porto, Portugal. Email: {m.ag, jtasso}@fe.up.pt
†ISEP, Porto, Portugal. Email: jes@isep.ipp.pt
‡NMEC, CESAM, DFis, UA, Aveiro, Portugal. Email: {rpsm, americosribeiro, joao.dias}@ua.pt

***Abstract*—We consider the problem of generating optimal planar trajectories for marine vehicles taking into account the effect of time-varying ocean currents. This is motivated by the need for economic trajectory generation in long-duration operations, and it becomes even more important in scenarios where the magnitude of the ocean current speed is comparable to the maximum vehicle speed, as illustrated in our experiments. We propose a solution based on a simple motion model, dynamic programming and high-resolution ocean models. Optimal trajectories are generated under the assumption that the vehicle is capable of reaching the destination even when the matching condition is violated in some segments of the path. Numerical results show that the method is appropriate for practical generation of feasible trajectories in spatially constrained areas with strong currents.**

I. Introduction

The problem of finding safe energy optimal trajectories for given ocean and weather forecasts has been receiving significant attention in the shipping industry, especially for the long shipping routes. These problems are typically formulated with coarse spatial and temporal resolutions, not only due to computational considerations but also because of the low resolution of available ocean and weather models. The problem is different for unmanned marine vehicles because of the need for higher spatial-temporal resolutions, especially for operations in constrained areas.

Mission requirements for unmanned marine vehicles are becoming more challenging in part because the reliability and capabilities of these vehicles have improved significantly over the last decade. Improvements in reliability, navigation, control, and communications enable, for example, operations in areas with currents that may overcome, for limited periods, the motion capabilities of marine vehicles. Areas of interest include estuaries or narrow channels between islands. However, ocean currents can also significantly improve mission duration, especially for underwater operations in which the structure of the flow may change dramatically with depth.

This paper presents a formulation of a problem of optimal motion planning for marine vehicles evolving in planar flows and describes a new solution approach based on high-resolution flow models and numerical methods for solving the Hamilton-Jacobi-Bellman (HJB) equation resulting from the application of the dynamic programming principle. This study is an interdisciplinary approach at the intersection of ocean modeling, robotics, and optimal control. From ocean modelling the approach builds on high-resolution models developed for constrained ocean-river interfaces, specifically for the estuarine flow of the Sado river, in Portugal. From robotics, the approach incorporated approximation techniques for motion models and the evaluation of feasible paths. From optimal control, the approach used dynamic programming techniques for solving a family of optimal motion planning problems, as well as numerical methods for solving the associated HJB equation. The spatial and temporal resolutions of the flow models are consistent with the discretization of the continuous motion models. The optimal trajectories obtained with simple motion models used in this approach for the given flow models are feasible trajectories for more accurate motion models of the marine vehicles under consideration.

The approach applies to 3-dimensional flow models and different cost functions. Approximation results and new numerical methods specially adapted to these problems are under development.

II. Problem Statement

Let $\boldsymbol{x} \in \mathbf{R}^2$ be the position of an underwater vehicle moving at constant depth in an ocean flow described by the vector field $\boldsymbol{v}(t, \boldsymbol{x}) \in \mathbf{R}^2$. The vehicle starts at some position $\boldsymbol{x}_0$ at time t_0, and the subsequent motion is modelled by the differential equation

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{u}(t) + \boldsymbol{v}(t, \boldsymbol{x}(t)), \tag{1}$$

where $\boldsymbol{u}(t) \in \mathbf{R}^2$ is the velocity of the vehicle due to control forces, satisfying $\boldsymbol{u}(t) \leq \boldsymbol{u}_{\max}$, $\boldsymbol{u}_{\max} \in \mathbf{R}_{>0}$.

The problem is then to find a trajectory $\boldsymbol{x}$ from any given $(t_0, \boldsymbol{x}_0)$ to a target set $\Omega \subset \mathbf{R}^2$ minimizing the integral performance index

$$J := \int_{t_0}^{T} g(\boldsymbol{x}(t))\mathrm{d}t. \tag{2}$$

where g is a positive function associating a cost to each possible position and T is the time of arrival at the target.

We note that some of interesting problems of AUV operations can be seen as particular cases of this one, for instance:

- To travel from an initial position to a final destination in minimum time avoiding undesired areas or obstacles;

- Given N possible final destinations and an initial position, choose and plan a trajectory to the 'best' (in some sense) final position.
- Given a set A_{init} of possible starting points, find the 'best' (in some sense) initial position and time to start in A_{init};

III. Related Work

The first major application of numerical methods for Hamilton-Jacobi equations in the context of robotic path planning was presented in [13], where no external field is considered (the so-called isotropic control problem), enabling the simplification of the HJB equation to a more tractable one, the eikonal equation. A numerical method for solving these equations inspired by Dijkstra's method is derived from a control-theoretic perspective. In [11] a different Dijkstra-inspired method is presented, based on a direct approximation of the eikonal PDE, and is now known as the fast-marching method. These two methods led to further research on numerical methods for Hamilton-Jacobi equations arising from control problems where the velocity may depend on the direction of motion (anisotropic problems), see e.g. [3], [12].

In [8] a similar approach to the one followed here is used, but the HJB equation is solved using an extremal field algorithm, while we opt for an Eulerian approach. The value function is obtained on an irregular grid with a resolution that is adjustable, albeit only indirectly. The paper focuses on target sets consisting of a single point, while arbitrary target sets are handled only by computing the value function separately for each connected component of the target. Computation times seem to be on the order of those of our method, but the method is inherently parallelizable.

In [4], [5], a level set method to find the minimum time from a particular point to every other point in the state space is derived. The motion model is the same as the one we have adopted here. Extensions to multiple vehicles are considered. Although the method was derived for the minimum-time-from-source problem, it suffices to reverse time to obtain a solution of the minimum-time-to-target problem. Only point sources are discussed, but arbitrary source sets should be achievable by simply changing the initial condition of the level set function. The method considers that the departure time from the source is fixed, so the value function must be computed separately for each possible departure time.

In [2], [14] the same model is used in conjunction with a numerical optimal control solver to generate single (i.e., open-loop) trajectories for an autonomous underwater glider. It is shown through numerical results that one can obtain approximately optimal trajectories using Lagrangian Coherent Structures. Open-loop optimal control techniques can produce trajectories that are only locally optimal, while the HJB equation is a sufficient condition for global optimality. Furthermore, trajectories are computed for a single initial position and departure time. On the other hand, if one is only interested in a single trajectory, the computation time is much shorter.

IV. Approach

We first convert the model to a time-invariant one by simply introducing a state variable τ describing time:

$$\begin{aligned} \dot{\boldsymbol{x}}(t) &= \boldsymbol{u}(t) + \boldsymbol{v}(\tau(t), \boldsymbol{x}(t)) \\ \dot{\tau}(t) &= 1 \end{aligned} \tag{3}$$

In this way $\tau(0)$ encodes the actual initial time while t is time relative to this initial time, so we always take initial conditions at $t = 0$.

We now let $\boldsymbol{x}(\cdot, t_0, \boldsymbol{x}_0, \boldsymbol{u}) : [0, +\infty) \to \mathbf{R}^2$ be the trajectory of the vehicle satisfying $\boldsymbol{x}(0, \tau_0, \boldsymbol{x}_0, \boldsymbol{u}) = \boldsymbol{x}_0$ under the control $\boldsymbol{u} : [0, +\infty) \to \mathbf{R}^2$ and define

$$\begin{aligned} T(\tau_0, \boldsymbol{x}_0, \boldsymbol{u}) &= \inf\{t \geq 0 \mid \boldsymbol{x}(t, \tau_0, \boldsymbol{x}_0, \boldsymbol{u}) \in \Omega\} \\ J(\tau_0, \boldsymbol{x}_0, \boldsymbol{u}) &= \int_0^{T(\tau_0, \boldsymbol{x}_0, \boldsymbol{u})} g(\boldsymbol{x}(t, \tau_0, \boldsymbol{x}_0, \boldsymbol{u}))\mathrm{d}t \end{aligned} \tag{4}$$

Additionally, let $\mathcal{U}(\tau_0, \boldsymbol{x}_0)$ be the set of measurable $\boldsymbol{u} : [0, +\infty) \to \mathbf{R}^2$ satisfying $|\boldsymbol{u}(t)| \leq \boldsymbol{u}_{\max}$ almost everywhere and such that $T(\tau_0, \boldsymbol{x}_0, \boldsymbol{u})$ is finite (i.e., the set of admissible controls).

The value function $V : \mathbf{R} \times \mathbf{R}^2 \to [0, +\infty]$ is defined as the cost of the optimal trajectory starting from $(\tau_0, \boldsymbol{x}_0)$, i.e.,

$$V(\tau, \boldsymbol{x}) = \inf\{J(\tau, \boldsymbol{x}, \boldsymbol{u}) \mid \boldsymbol{u} \in \mathcal{U}(\tau, \boldsymbol{x})\}. \tag{5}$$

By a standard argument using Bellman's principle of optimality, V satisfies the HJB equation (denoting partial derivatives by subscripts)

$$\min_{|\boldsymbol{w}| \leq \boldsymbol{u}_{\max}} \{g(\boldsymbol{x}) + (V_{\boldsymbol{x}}, V_\tau) \cdot (\boldsymbol{w} + \boldsymbol{v}(\tau, \boldsymbol{x}), 1)\} = 0 \tag{6}$$

on $\mathbf{R} \times \left(\mathbf{R}^2 \setminus \Omega\right)$, which may be rewritten as

$$\boldsymbol{u}_{\max}|V_{\boldsymbol{x}}| - (V_\tau + V_{\boldsymbol{x}} \cdot \boldsymbol{v}(\tau, \boldsymbol{x})) = g(\boldsymbol{x}). \tag{7}$$

Additionally, there is the boundary condition

$$V(\tau, \boldsymbol{x}) = 0 \tag{8}$$

for all $(\tau, \boldsymbol{x}) \in \mathbf{R} \times \partial\Omega$. The optimal control action at $(\tau, \boldsymbol{x})$ is the minimizing $\boldsymbol{w}$ in (6). Hence, the optimal trajectory from $(\tau_0, \boldsymbol{x}_0)$ is the solution of the ordinary differential equation

$$\dot{\boldsymbol{x}}(\tau) = -\boldsymbol{u}_{\max} \frac{V_{\boldsymbol{x}}(\tau, \boldsymbol{x}(\tau))}{|V_{\boldsymbol{x}}(\tau, \boldsymbol{x}(\tau))|} + \boldsymbol{v}(\tau, \boldsymbol{x}(\tau)) \tag{9}$$

with the initial condition $\boldsymbol{x}(\tau_0) = \boldsymbol{x}_0$.

Remarks.

1) As shown in the first example of Section V, the value function is not necessarily everywhere differentiable, so one must take V to be the viscosity solution of (7) [1].
2) We have limited ourselves to the case where the target set is of the form $\mathbf{R} \times \Omega$ when viewed in the extended state space $(\tau, \boldsymbol{x})$. However, we could just as easily consider arbitrary sets in $(\tau, \boldsymbol{x})$ space (e.g., arrival time constraints).
3) Although we have considered only two spatial dimensions, the method applies to three-dimensional flows just as well.

To solve (7), we first convert it to a front expansion problem, following Osher [7]. This is done by considering a function $\varphi : [0,\infty) \times \mathbf{R} \times \mathbf{R}^2 \to \mathbf{R}$ such that the zero level set of $\varphi(s,\cdot,\cdot)$ coincides with the s-level set of V, that is

$$V(\tau,\boldsymbol{x}) = s \iff \varphi(s,\tau,\boldsymbol{x}) = 0 \tag{10}$$

If we choose $\varphi(0,\tau,\boldsymbol{x})$ to be zero on $\mathbf{R}\times\partial\Omega$, negative on the interior of Ω and positive elsewhere (we assume that the target is a union of compact regions delimited by smooth closed curves so that this is feasible), it can be shown that φ satisfies the Hamilton-Jacobi equation

$$\varphi_s + \frac{\boldsymbol{u}_{\max}|\varphi_{\boldsymbol{x}}| - (\varphi_\tau + \varphi_{\boldsymbol{x}}\cdot\boldsymbol{v}(\tau,\boldsymbol{x}))}{g(\boldsymbol{x})} = 0, \tag{11}$$

which can be interpreted as describing the expansion of a wavefront of surfaces: let $\Gamma(s) = \{(\tau,\boldsymbol{x}) \mid \varphi(s,\tau,\boldsymbol{x}) = 0\}$ be the wavefront at time s, with $\Gamma(0) = \Omega\times\mathbf{R}$, the target surface in the extended space $(\tau,\boldsymbol{x})$. Each point in the wavefront moves with velocity

$$\begin{aligned}\frac{\mathrm{d}\boldsymbol{x}}{\mathrm{d}s} &= \frac{\boldsymbol{u}_{\max}\boldsymbol{n}_s(\tau,\boldsymbol{x}) - \boldsymbol{v}(\tau,\boldsymbol{x})}{g(\boldsymbol{x})}\\ \frac{\mathrm{d}\tau}{\mathrm{d}s} &= -\frac{1}{g(\boldsymbol{x})}\end{aligned} \tag{12}$$

where $\boldsymbol{n}_s(\tau,\boldsymbol{x})$ is the outward unit normal to the curve $\{\boldsymbol{\xi} \mid (\tau,\boldsymbol{\xi}) \in \Gamma(s)\}$ at $\boldsymbol{x}$. In other words, each point of the wavefront moves backwards in time along an optimal trajectory.

We can solve (11) using level set methods. A computational region containing Ω is chosen and discretized, and the equation is solved on the resulting grid. Note that s takes the role of a 'time' variable when we are integrating (10), so we must integrate from $s = 0$ to some $s_{\max}$ which defines the maximum allowable cost (trajectories with higher cost will not be recoverable).

The publicly available TOOLBOXLS[1] for MATLAB® [6] allows easy implementation of a solver, and provides a routine for obtaining V from φ using (10).

V. SIMULATION EXAMPLES

We introduce two examples to illustrate key aspects of the proposed approach. The first one is about an idealized flow and the second one is about a model of an estuarine flow driven by tidal currents.

A. Idealized flow

Consider a simple flow defined by the equations

$$\begin{aligned}v_1(t,x_1,x_2) &= A(x_2)\cos(2\pi t)\\ v_2(t,x_1,x_2) &= 0\end{aligned} \tag{13}$$

where

$$A(x_2) = 1.1\exp\left(-\left(\frac{x_2}{0.3}\right)^2\right). \tag{14}$$

Assuming a vehicle velocity of $\boldsymbol{u}_{\max} = 1$ unit of distance per unit of time and a target set consisting of a ball centered at $\Omega = \{(1,0)\}$ with a radius of twice the width of a grid cell, the value function for the minimum time problem was computed on the region $\{(t,x_1,x_2) \mid 0 \le t \le 2.5,\ 0 \le x_1 \le 1.2,\ -0.3 \le x_2 \le 0.3\}$ over a rectangular grid with $250\times120\times60$ points.

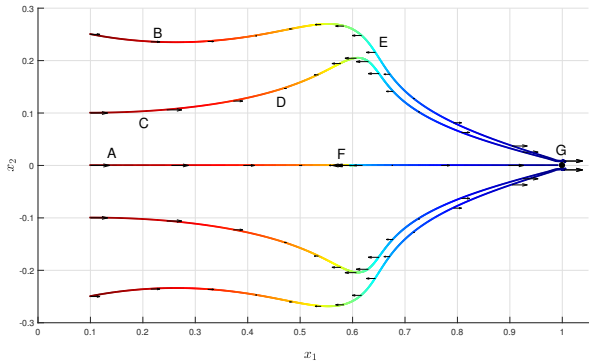

Fig. 1. A few time-optimal trajectories for a vehicle with unit speed under the influence of the flow (13)

Fig. 1 depicts five optimal trajectories – arrows indicate the flow magnitude and direction at equally spaced instants of time, and the magnitude of the value function at each point of the trajectory (i.e., the value $V(t,\boldsymbol{x}(t))$) is color-coded.

The optimal trajectories were obtained by using a second order Runge-Kutta method. The gradient of the value function was calculated on the grid points using a fourth order finite difference formula and a linear interpolation was used for the other points. The values of the flow were calculated using expression (13) (although the value function is calculated using only the values of formula (13) at the grid points).

The computation was performed on the HPC cluster Grid FEUP[2] using a single node with 16 cores and 8 GB of memory, taking approximately four hours of CPU time (resulting in one hour of wallclock time).

Next, we discuss features of the optimal trajectories plotted in Fig. 1 (cf. the labels on the figure):

A - For an initial condition with $x_2 = 0$, the trajectory stays on the x_1-axis, but neighbouring trajectories will move away from this axis to take advantage of weaker flows away from this axis. Optimal trajectories departing from x_1 axis are not unique because of the symmetry of the flow – the value function is nonsmooth along this axis.

B - A trajectory departing from a region where the flow is weak moves toward a region where there is a stronger flow in the direction of the target.

C–D - A trajectory departing from a region where the flow is strong in the direction does not move further toward the x_1-axis where it would experience more favorable flow, expressing the trade-off between moving toward regions of stronger, favorable flow and the time taken to move away from these regions when the direction of the field changes. We see that in fact the vehicle immediately starts moving away from the x_1-axis, in order to take advantage of the

[1]https://www.cs.ubc.ca/~mitchell/ToolboxLS/

[2]https://grid.fe.up.pt/

weaker flow when the flow opposes the optimal direction to the target.

E - As the flow starts changing direction again the vehicle starts moving toward the x_1-axis in order to approach the target.

F - Since the flow is stronger along the x_1-axis than the maximum vehicle speed, when the flow reaches its maximum amplitude in the direction opposite to the target, the vehicle is pushed away from the target.

G - All trajectories arrive at a neighborhood of the target point.

B. Hydrodynamic numerical model of the Sado estuary

Here, we consider a numerical model of the Sado estuary. This model outputs the horizontal and vertical components of the flow velocity defined over a nonuniform 4-dimensional grid (time, latitude, longitude and depth in sigma coordinates). The hydrodynamic simulation is performed using the Delft3D-Flow modelling system, adapted from the application developed by Ribeiro et al. [9]. This platform was setup with a curvilinear irregular grid with a mean resolution of 100 m in the area of interest (Sado estuary and inlet). The main difference relies on the numerical bathymetry, which was updated with the most recent topo-hydrographic surveys of this region.

The tidal forcing on the model open boundaries was imposed from Oregon State University (OSU) TOPEX/Poseidon Global Inverse Solution 8.0. Transport conditions were gathered from hourly results obtained from the Iberian Biscay Irish Ocean Analysis and Forecasting system. The surface boundary condition was also imposed hourly, using data from the Weather Research Forecasting (WRF) model[3] with a spatial resolution of 12 km. An ocean heat transport model was applied, taking into account air temperature, relative humidity and ocean cloudiness. The modelling time step was 60 seconds, and the bottom roughness was estimated with a standard coefficient of friction [10].

The simulation for September of 2018 was setup from a spin-up of one month, and provided a continuously 3-day forecast with a temporal resolution of ten minutes.

For the purposes of this computation only the horizontal velocity was considered, and it was assumed that the AUV would perform the trajectory at a relatively low depth (approximately 2 m), so that only the values of the velocity corresponding to the sea surface were used.

In order to use the flow data for our purposes in TOOLBOXLS, it is necessary to compute values of the flow on a regularly spaced grid. To this end, after sub-selecting the values of the flow corresponding to a desired operational area, the spatial grid is first converted to Universal Transverse Mercator (UTM) coordinates (in meters). The flow is then interpolated to a regular grid using linear interpolation.

The operational area was chosen as an approximately 13 $\times$ 9.5 km rectangular region (see Fig. 4). The target is the point with coordinates (38^{o}N 28' 37.73", 8W 52' 12.57"), approximated by a sphere of radius equal to twice the width of a grid cell. In this example the AUV starts in the lower region of the estuary and enters the inner area through the narrow channel. For this reason the flow values from the 13th of September 2018 at 23:00:00 to the 14th of September 2018 at 11:00:00 corresponding to a full tide period starting at flood tide were used. The grid had a resolution of 50 m in each spatial dimension and 10 minutes in the temporal dimension. The vehicle velocity $\boldsymbol{u}_{\max}$ was taken to be 1 m/s.

In Fig. 2 and 3 one can see part of the interpolated flow vector field corresponding to two different times, at flood and ebb tide, respectively.

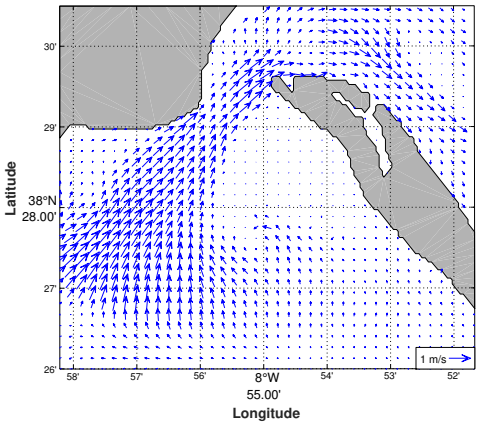

Fig. 2. Current field on the 13th of September 2018, 00:00:00

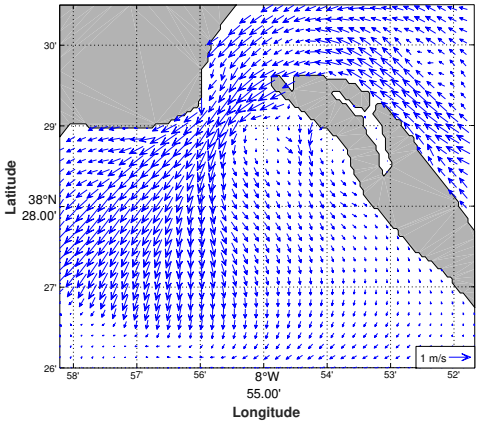

Fig. 3. Current field on the 13th of September 2018, 6:30:00

In order to prevent the trajectories from entering shallow areas or the navigation channel of the Setúbal harbor we introduced a constraint penalization in the cost function:

$$g(\boldsymbol{x}) = g_{\text{mt}}(\boldsymbol{x}) + g_{\text{const}}(\boldsymbol{x}) = 1 + g_{\text{const}}(\boldsymbol{x}), \qquad (15)$$

To construct g_{const}, a high resolution land mask adapted to the grid was obtained using the Generic Mapping Tools[4], which uses data from the GSHHG high resolution geography database[5]. To mask low-depth regions and the navigation

[3]https://www.mmm.ucar.edu/weather-research-and-forecasting-model

[4]http://gmt.soest.hawaii.edu/

[5]http://www.soest.hawaii.edu/wessel/gshhg/

channel, polygonal approximations of these were constructed employing a nautical chart of the region. After obtaining a binary mask of the whole forbidden set, a Gaussian filter was applied to smooth the transitions between allowed and forbidden regions. The function g_{const} is defined on the grid points as this smoothed version of the mask multiplied by an adjustable constant.

The computation was performed on the HPC cluster Grid FEUP using a single node with 16 cores and 8 GB of memory, taking approximately twenty hours of CPU time (resulting in five hours of wallclock time).

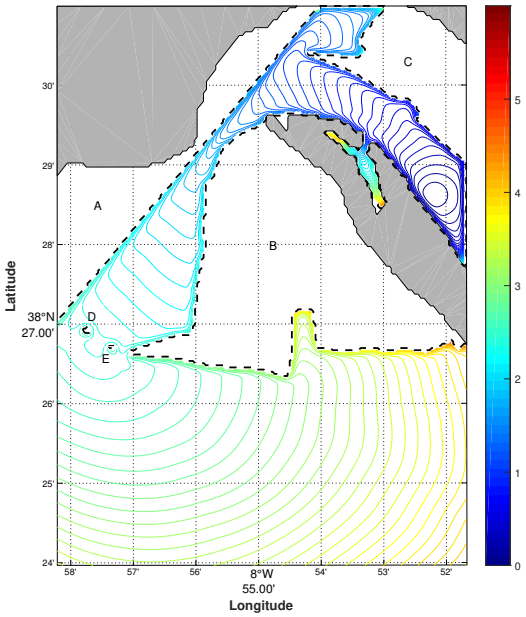

Fig. 4. Contours of the value function for trajectories departing at 13/Sep/2018, 23:00:00

In Fig. 4 the contours of the value function restricted to $t = 23{:}00{:}00$ are plotted. The color indicates the cost of each initial condition in hours. The boundaries of the forbidden regions are indicated by the dotted lines and labelled. Region A was blocked out to prevent the vehicle from entering the navigation channel, which would risk collision with a ship, and shallow regions to the left of the channel. Regions B, C, D and E are low-depth or land regions.

An optimal trajectory departing at 13/Sep/2018, 23:00:00 is shown in Fig. 5. The initial position and target point are indicated by the red and black circles, respectively. The arrows indicate the value of the flow field at equally spaced instants of time. The color indicates the evolution of the value function along the trajectory. The travel time is approximately 3 hours and 1.8 minutes and the arc length of the trajectory is approximately 17.2 km, leading to an average velocity of 1.58 m/s. The minimum radius of curvature of the trajectory is approximately 211 m, which indicates that the trajectories obtained by this method should be feasible for a torpedo-shaped AUV (although the method does not guarantee this). This seems to imply that the simple holonomic model we have

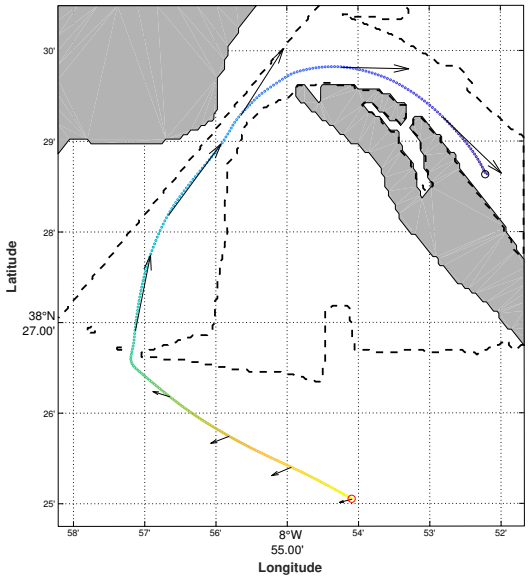

Fig. 5. An optimal trajectory and the corresponding flow values

used is a good approximation of actual AUV dynamics. Future research will include investigation of a precise statement of this property.

Four other optimal trajectories are shown in Fig. 6, with the initial position indicated by a red circle, all departing at 23/Sep/2018 23:00:00. The radius of curvature of each of these trajectories is bounded below by 190 m.

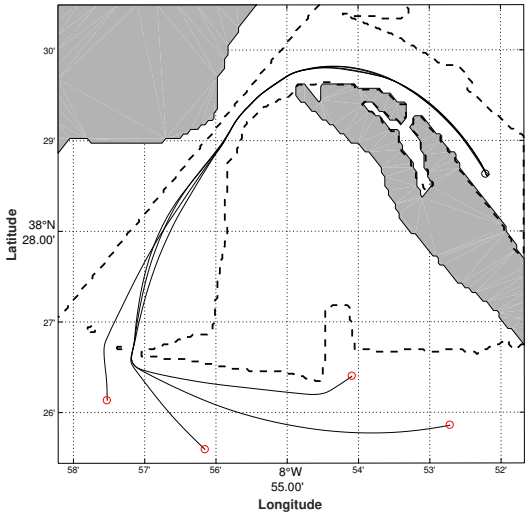

Fig. 6. A few optimal trajectories for the second example

VI. Conclusions and Future Work

The numerical results show that the proposed approach can be used to generate trajectories in practical operational scenarios. Feasible trajectories were obtained for navigating

through spatially constrained areas, and a gain in the vehicle velocity was observed. Publicly available software tools allow for easy implementation of a solver. The computation times are acceptable for offline precomputation of the value function, from which optimal trajectories can be quickly calculated at the time of deployment and used as reference for the motion control software. The optimal trajectory can also be obtained efficiently via online feedback control, but the discussion of that subject is precluded due to space limitations.

Future work will include experimental tests, results using 3-dimensional flow models, investigation of guarantees on the feasibility of the generated trajectories and approximation results.

Acknowledgments

M.A. would like to thank the LSTS team, in particular José Pinto and Paulo Dias, for insightful discussions on AUV operations. R.M. benefits from a Post-Doctoral grant (SFRH/BPD/115093/2016) given by the Portuguese Science Foundation (FCT). This article is a result of the project 'ENDURANCE – Sistema baseado em veículos autónomos para observação oceanográfica de longa duração', supported by Norte Portugal Regional Operational Programme (NORTE 2020), under the PORTUGAL 2020 Partnership Agreement, through the European Regional Development Fund (ERDF). Thanks are due for the financial support to CESAM (UID/AMB/50017 - POCI-01-0145-FEDER-007638) to FCT/MCTES through national funds (PIDDAC), and the co-funding by the FEDER, within the PT2020 Partnership Agreement and Compete 2020.

References

[1] M. Bardi and I. Capuzzo-Dolcetta, *Optimal Control and Viscosity Solutions of Hamilton-Jacobi-Bellman Equations*. Birkhäuser Boston, 1997. DOI: 10.1007/978-0-8176-4755-1.

[2] T. Inanc, S. C. Shadden, and J. E. Marsden, "Optimal trajectory generation in ocean flows," in *In Proc. American Control Conf. 674–679*, 2004. DOI: 10.1109/acc.2005.1470035.

[3] C. Y. Kao, S. Osher, and J. Qian, "Lax–Friedrichs sweeping scheme for static Hamilton–Jacobi equations," *Journal of Computational Physics*, vol. 196, no. 1, pp. 367–391, May 2004. DOI: 10.1016/j.jcp.2003.11.007.

[4] S. V. T. Lolla, "Path planning in time dependent flows using level set methods," Master's thesis, Massachusetts Institute of Technology. Dept. of Mechanical Engineering., 2012. [Online]. Available: http://dspace.mit.edu/handle/1721.1/7582.

[5] T. Lolla, M. P. Ueckermann, K. Yigit, P. J. Haley, and P. F. J. Lermusiaux, "Path planning in time dependent flow fields using level set methods," in *2012 IEEE International Conference on Robotics and Automation*, IEEE, May 2012. DOI: 10.1109/icra.2012.6225364.

[6] I. M. Mitchell and J. A. Templeton, "A toolbox of Hamilton–Jacobi solvers for analysis of nondeterministic continuous and hybrid systems," in *Hybrid Systems: Computation and Control*, Springer Berlin Heidelberg, 2005, pp. 480–494. DOI: 10.1007/978-3-540-31954-2_31.

[7] S. Osher, "A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations," *SIAM Journal on Mathematical Analysis*, vol. 24, no. 5, pp. 1145–1152, Sep. 1993. DOI: 10.1137/0524066.

[8] B. Rhoads, I. Mezic, and A. Poje, "Minimum time feedback control of autonomous underwater vehicles," in *49th IEEE Conference on Decision and Control (CDC)*, IEEE, Dec. 2010. DOI: 10.1109/cdc.2010.5717533.

[9] A. S. Ribeiro, M. C. Sousa, J. D. L. e Silva, and J. M. Dias, "David and Goliath revisited: Joint modelling of the Tagus and Sado estuaries," *Journal of Coastal Research*, vol. 75, no. sp1, pp. 123–127, Mar. 2016. DOI: 10.2112/si75-025.1.

[10] L. C. van Rijn, "Unified view of sediment transport by currents and waves. I: Initiation of motion, bed roughness, and bed-load transport," *Journal of Hydraulic Engineering*, vol. 133, no. 6, pp. 649–667, Jun. 2007. DOI: 10.1061/(asce)0733-9429(2007)133:6(649).

[11] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proceedings of the National Academy of Sciences*, vol. 93, no. 4, pp. 1591–1595, 1996, ISSN: 0027-8424. DOI: 10.1073/pnas.93.4.1591.

[12] J. A. Sethian and A. Vladimirsky, "Ordered upwind methods for static Hamilton–Jacobi equations: Theory and algorithms," *SIAM Journal on Numerical Analysis*, vol. 41, no. 1, pp. 325–363, Jan. 2003. DOI: 10.1137/s0036142901392742.

[13] J. N. Tsitsiklis, "Efficient algorithms for globally optimal trajectories," *IEEE Transactions on Automatic Control*, vol. 40, no. 9, pp. 1528–1538, 1995. DOI: 10.1109/9.412624.

[14] W. Zhang, T. Inanc, S. Ober-Blobaum, and J. E. Marsden, "Optimal trajectory generation for a glider in time-varying 2D ocean flows B-spline model," in *2008 IEEE International Conference on Robotics and Automation*, IEEE, May 2008. DOI: 10.1109/robot.2008.4543348.