Copyright Information

BibTeX

```
@inproceedings{Simetti2018decoupledsampling,
title={Decoupled Sampling-Based Motion Planning for Multiple
    Autonomous Marine Vehicles},
author={Volpi, Nicola Catenacci and Smith, Sim{\'o}n C and Pascoal,
    Ant{\'o}nio M and Simetti, Enrico and Turetta, Alessio and Alibani
    , Michael and Polani, Daniel},
booktitle={OCEANS 2018 MTS/IEEE Charleston},
pages={1--7},
year={2018},
organization={IEEE},
doi={10.1109/OCEANS.2018.8604908},
ISSN={0197-7385},
}
```

Decoupled Sampling-Based Motion Planning for Multiple Autonomous Marine Vehicles

Nicola Catenacci Volpi[1], Simón C. Smith[1,2], António M. Pascoal[3], Enrico Simetti[4], Alessio Turetta[5], Michael Alibani[6], Daniel Polani[1]

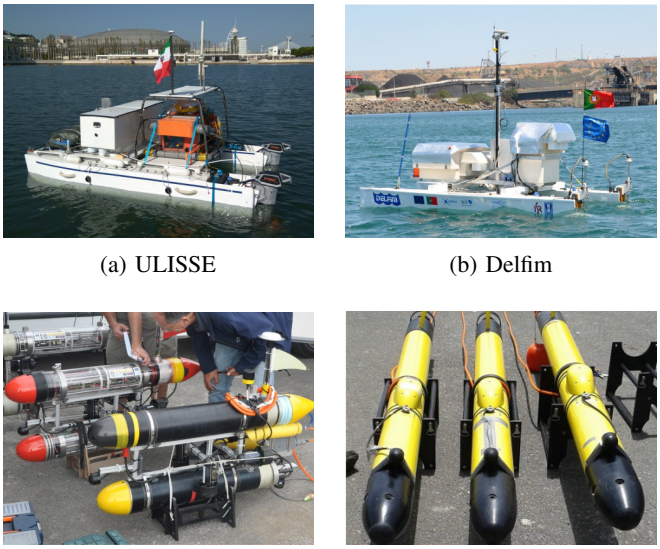
***Abstract*— There is increasing interest in the deployment and operation of multiple autonomous marine vehicles (AMVs) for a number of challenging scientific and commercial operational mission scenarios. Some of the missions, such as geotechnical surveying and 3D marine habitat mapping, require that a number of heterogeneous vehicles operate simultaneously in small areas, often in close proximity of each other. In these circumstances safety, reliability, and efficient multiple vehicle operation are key ingredients for mission success. Additionally, the deployment and operation of multiple AMVs at sea are extremely costly in terms of the logistics and human resources required for mission supervision, often during extended periods of time. These costs can be greatly minimized by automating the deployment and initial steering of a vehicle fleet to a predetermined configuration, in preparation for the ensuing mission, taking into account operational constraints. This is one of the core issues addressed in the scope of the Widely Scalable Mobile Underwater Sonar Technology project (WiMUST), an EU Horizon 2020 initiative for underwater robotics research.**

WiMUST uses a team of cooperative autonomous marine robots, some of which towing streamers equipped with hydrophones, acting as intelligent sensing and communicating nodes of a reconfigurable moving acoustic network. In WiMUST, the AMVs maintain a fixed geometric formation through cooperative navigation and motion control. Formation initialization requires that all the AMVs start from scattered positions in the water and maneuver so as to arrive at required target configuration points at the same time in a completely automatic manner. This paper describes the decoupled prioritized vehicle motion planner developed in the scope of WiMUST that, together with an existing system for trajectory tracking, affords a fleet of vehicles the above capabilities, while ensuring inter-vehicle collision and streamer entanglement avoidance. Tests with a fleet of seven marine vehicles show the efficacy of the system planner developed.

I. INTRODUCTION

A. Widely Scalable Mobile Underwater Technology

The aim of the *Widely scalable Mobile Underwater Technology* project (WiMUST, [2]) is to conceive, design, and engineer an intelligent, manageable, and distributed underwater acoustic array. The objective of this novel acoustic array is to improve the efficiency of geophysical and geotechnical acoustic surveys at sea. These include seabed mapping, seafloor characterization, and seismic exploration, which are fundamental operations for the oil and gas industry, as well as civil engineering. The main novelty of WiMUST is the introduction of a cooperative team of autonomous marine robots, acting as intelligent sensing and communicating nodes of a dynamic seismic recording network. Two types of vehicles are part of the WiMUST fleet (see Fig. 1). First, autonomous underwater vehicles (Fig. 1c and Fig. 1d) with the role of towing streamers of hydrophones of small aperture (see Fig. 2). Second, surface autonomous vehicles that work as anchor nodes for communication and carry the sparker units responsible for producing seismic acoustic signals. (Fig. 1a and Fig. 1b). The array is capable of acquiring seismic data by illuminating the seabed and the ocean sub-bottom with strong seismic waves sent by the sources installed on-board of the anchor nodes. Such configuration allows the overall system to behave as a large moving distributed acoustic antenna. The resulting operational flexibility allows improving the resolution of seabed and sub-bottom mapping and obtaining side lobe rejection at almost any frequency and



(a) ULISSE (b) Delfim

(c) Medusas (d) Folagas

Fig. 1: Seven vehicles were used in the WiMUST trials: the ULISSE (a) and Delfim (b) catamarans; three Medusas (c) and two Folagas (d) autonomous underwater vehicles.

This work was partially supported by EC Horizon 2020 programme under the project WiMUST (Grant agreement no: 645141, Strategic objective: H2020 - ICT-23-2014 - Robotics)

[1,6]Adaptive Systems Research Group, School of Computer Science, University of Hertfordshire, U.K. n.catenacci-volpi@herts.ac.uk

[2]Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh, U.K.

[3]Laboratory of Robotics and Engineering Systems (LARSyS), ISR/IST, University of Lisbon, Portugal

[4]University of Genova - ISME Node, Genova, Italy

[5]Graal Tech S.r.l., Genova, Italy

[6]Department of Information Engineering (DII), University of Pisa, Italy

for any plane. Also, WiMUST facilitates the operations at sea through the use of wireless, acoustic communication between the surface ship and the acquisition equipment. As its name suggests, scalability has been an important priority for the WiMUST project. Hence, several solutions were adopted to make the whole system scalable with the number of deployed vehicles. This paper presents how the problem of scalability was tackled in the context of motion planning for multiple AMVs.

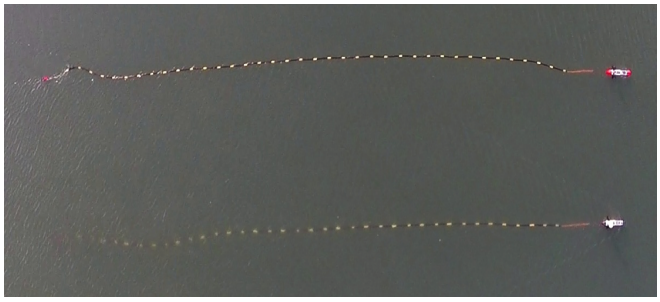


Fig. 2: Two Medusas AMVs equipped with streamers.

Go-to formation maneuver: Among the several challenges addressed in the scope of WiMUST (see [2] for a detailed list), in this paper we consider and present a solution to the problem of making a number of autonomous vehicles reach a desired geometric formation at a specified initial location, prior to starting an acoustic survey. In fact, before a survey can start, the vehicles carrying the array of hydrophones and the acoustic sources need to be in a precise formation. Hence, the objective of this phase is to bring the vehicles from random positions in the ocean to the initial formation needed for the survey. Current methods include deploying the vehicles at the desired positions or manually driving them to the required locations. Such methods suffer from poor scalability. The operation of manually placing or driving the vehicles to the desired positions is already cumbersome for a fleet of a few vehicles, and grows in complexity for every new vehicle added to the array. Sea conditions and the number of human operators are limiting factors when the number of vehicles increases. The WiMUST project aimed for the automation of the go-to formation task. When the operations at the sea start, all the vehicles are scattered randomly around the survey area. The task of the WiMUST motion planner is to drive the vehicles autonomously from their random initial positions to the location of the starting formation, limited by the physical and control dynamics of the vehicles. Additional constraints include making the AMVs arrive at the desired configuration at the same time while avoiding inter-vehicle collisions and streamer entanglement. To automatically generate the desired trajectories for each vehicle, we propose a decoupled prioritized motion planning approach that extends the rapidly-exploring random tree algorithm (RRT, [16]).

B. Multiple AMVs Motion Planning

Motion planning is at the core of advanced systems for single and multiple vehicle operations in vastly different domains that include marine, land, aerial, and space robotics. See for example [12] [3] and the references therein. There is by now a wealth of methods available for motion planning that exploit different, yet possibly complementary theoretical frameworks, ranging from artificial potential functions and roadmap path planners to cell decomposition methods and optimal control-based trajectory generation algorithms. Optimal control methods for motion planning are especially suited to take directly into account time and energy objectives, as well vehicle dynamics and ambient constraints. Direct multiple shooting [8], pseudo-spectral [17], and PRONTO [3] are but a few examples. In general, they are quite demanding in terms of computational power. Direct discretization methods (whereby the states of the systems involved, the inputs, or a combination thereof are approximated by specific classes of polynomial functions) have computational advantages, for they have the potential to cast the original problem in the form of nonlinear programming problem with a reduced search space. Examples include the use of classical polynomials [12], splines [18], and Bezier curves [7]. Sampling-based methods for motion planning algorithms, which were used in the WiMUST project, are among the methods that scale well to multiple vehicles problems [14]. To obtain smooth and dynamically-feasible paths, they must be combined with other techniques [13]. A comparison of the computation times obtained by different sampling-based strategies can be found in [9].

II. METHODS: DECOUPLED SAMPLING-BASED MOTION PLANNING

The WiMUST motion planner solves the problem of computing collision-free trajectories that the vehicles must track in order to be brought to a desired geometric formation. The latter is crucial to properly synchronize the initialization of the ensuing WiMUST cooperative mission. Meeting this objective requires quick computation of a motion plan that will allow the vehicles to arrive at the same time at their desired final configuration from any arbitrary initial configuration. In this section we define the corresponding motion planning problem and present the solutions adopted by the consortium to solve it efficiently.

A. The multi-AMVs Motion Planning problem

For AMV motion planning, the space where the vehicles trajectories are computed consists of the set of possible transformations that can be applied to the vehicles, also called the *configuration space* X (or C-space) [15]. To solve a motion planning problem, an algorithm must perform a search in the C-space. For the i-th of the N surface vehicles $A_1, \ldots, A_i, \ldots, A_N$, the C-space X_i is a manifold characterized by the set of all possible 2D rigid-body homogeneous transformations in $\mathbb{R}^2$. Since any q_i^x, q_i^y can be selected for translation, this alone yields a manifold $M_1 = \mathbb{R}^2$. Independently, rotations in yaw $q_i^\theta \in [0, \pi)$ can be applied along the z axis yielding the circle manifold $M_2 = \mathbb{S}^1$. To obtain the complete manifold corresponding to all 2D rigid-body motions we take $X_i = M_1 \times M_2 = \mathbb{R}^2 \times \mathbb{S}^1$ (i.e., a

space that "looks like" a cylinder). Hence, the resulting C-space is a three-dimensional manifold, with two dimensions deriving from translations and one more deriving from the yaw rotation. Since in the WiMUST missions seven vehicles were deployed in the environment $W = \mathbb{R}^2$ ($N = 7$), their corresponding C-spaces X_i can be combined using Cartesian products to obtain the configuration space X of the whole multiple vehicle system. Hence, the complete C-space is defined as $X = X_1 \times X_2 \times \dots X_7$.

Let $O \subset X$ denote the *obstacle region* i.e., the portion of the configuration space X that either cause the vehicles to collide with each other, with the streamers, or cause the streamers to entangle in W. A motion planning algorithm must find a path from an initial configuration to a goal configuration that traverses only the leftover space $X_{free} \subset X$, which is obtained by removing O from X i.e., $X_{free} = X \backslash O$. Hence, in WiMUST the multi-AMVs motion planning problem consists of finding a set of trajectories $\overrightarrow{\mathbf{x}}_1, \overrightarrow{\mathbf{x}}_2, \dots, \overrightarrow{\mathbf{x}}_7$ in the collision-free space X_{free} that start at a given starting configuration $x^s \in X_{free}$ (i.e., the vehicles in the vicinity of the support vessel) and that end at a desired goal configuration $x^g \in X_{free}$ (i.e., the vehicles at the geometrical formation that is required to properly initialize the autonomous geotechnical survey). The pair (x^s, x^g) is called a *query* of the motion planning problem. In the following section we present the techniques adopted to solve this problem in the context of the WiMUST project.

B. Sampling-based motion planning

In WiMUST, the solution developed for the problem described in the previous section belongs to the broad class of sampling-based motion planning algorithms [15] [13]. This section provides a brief introduction to such sampling-based techniques. The key idea behind this class of algorithms is to avoid the explicit construction of O and instead conduct a search that probes the C-space sampling randomly chosen configurations. This probing is enabled by the presence of the geometric models of the vehicles and streamers, which are used by a collision detection module that verifies the validity of each sampled configuration (i.e., if it is collision-free or not). The collision detection module used within the WiMUST Motion Planner was implemented using the *Flexible Collision Library (FCL)* [11]. The aim of this sampling process is to construct a compact graph data-structure (usually called *roadmap*) which, once searched, efficiently solves queries. The roadmap is a topological graph $G(V, E)$ in which the set of vertices V represents the sampled valid vehicles configurations and the edges E correspond to the set of paths that connect such configurations in the space X_{free}. Hence, once the roadmap is constructed, a specific query pair (x^s, x^g) is solved applying standard discrete graph search algorithms in G (e.g., depth-first search) to obtain a sequence of edges that form a collision-free path from x^s to x^g.

C. The WiMUST motion planner

The WiMUST motion planner is an extension of the *Rapidly expanding Random Tree (RRT)* algorithm [16]. It was integrated in the general WiMUST software architecture using the *Open Motion Planning Library* (OMPL) [20] running on *ROS* (Robot Operating System [19]). This library was extended to implement a prioritized decoupled motion planning approach, which includes kinematical constraints for the vehicles and geometrical constraints to avoid the entanglement of the streamers. In addition, a simple and effective solution was found to make all the vehicles arrive at the chosen final formation at the exact same time. Such extensions, together with the main challenges encountered during the project and the solutions adopted to tackle them, will be described in the following sections.

1) Kinematic constraints: In WiMUST, the imposition of differential constraints was necessary to develop a planner capable of computing collision-free trajectories from x^s to x^g that take into account the vehicles kinematics in terms of velocity profiles. The differential constraints that arise from the vehicles kinematics restrict the allowable velocities at each points in the *state space* X[1]. For the i-th vehicle, these constraints are usually expressed in terms of a state transition equation such as $\dot{x}_i = f(x_i, u_i)$ defined on the smooth state manifold X_i and with control vector $u_i \in U \subset \mathbb{R}^2$, where U is the control space, which for all the vehicles is composed by the their surge velocity v_i and yaw turning rate r_i. The state-space model f adopted to take the vehicles' kinematic constraints into account is given by

$$\begin{bmatrix} \dot{q}_i^x \\ \dot{q}_i^y \end{bmatrix} = R(q_i^\theta) \begin{bmatrix} v_i \\ 0 \end{bmatrix}$$
$$\dot{q}_i^\theta = r_i$$

where $R(\cdot)$ is the rotation matrix from body to inertial reference frames, $v_i \in [0.3, 1]$ m/s, and $r_i \in [-0.2, 0.2]$ rad/s for all vehicles.

In this context, the solution trajectory of the i-th vehicle $\overrightarrow{\mathbf{x}}_i$ is derived from a sequence of sampled motion primitives $\overrightarrow{\mathbf{u}}_i^p$ via repeated integrations of the state transition equation. These sampled $\overrightarrow{\mathbf{u}}_i^p$ are control trajectories for which a randomly chosen control input u_i is held constant for a short amount of time Δt. Given a control trajectory $\overrightarrow{\mathbf{u}}_i$ and starting from the initial state $x_i(0)$ at time $t = 0$, the WiMUST motion planner derives the corresponding state trajectory $\overrightarrow{\mathbf{x}}_i$ from $\overrightarrow{\mathbf{u}}_i$ integrating the equation $\dot{x}_i = f(x_i, u_i)$ to yield $x_i(t) = x_i(0) + \int_0^t f(x_i(\tau), u_i(\tau))d\tau$. Let us define the state trajectory $\overrightarrow{\mathbf{x}}_i$ in the time interval $T = [0, \tilde{T}] \subset \mathbb{R}$. Then, the output of the WiMUST motion planner is a control trajectory $\overrightarrow{\mathbf{u}}_i : T \to U$ for which the corresponding state trajectory $\overrightarrow{\mathbf{x}}_i$ satisfies $x_i(0) = x_i^s$, and there exists some $0 < t < \tilde{T}$ for which $x_i(t) = x_i^g$. Note that, since motions

[1]For the scope of this paper, the term "state space", usually used when differential constraints are introduced, can be identified with the term "configuration space". Hence, in the following the two terms will be used interchangeably.

are expressed in terms of sequences of motion primitives $\overrightarrow{\mathbf{u}}_i^p$, but collision detection tests must be performed in X_i, the system $\dot{x}_i = f(x_i, u_i)$ needs to be integrated frequently during the planning process. In addition, when kinematic constraints are considered, the edges E of the roadmap graph G do not represent paths, but motion primitives $\overrightarrow{\mathbf{u}}_i^p$ instead.

2) Decoupled prioritized approach to tackle the multi-vehicles' curse of dimensionality: The scalability of motion planning for multiple autonomous vehicles is a computationally hard problem. Planning in the joint configuration space of all the vehicles can lead to an intractable problem. Such intractability arises from the linear growth of the dimensions of the configuration space with respect to the number of vehicles. In Section II.A the C-space of the seven vehicles of the WiMUST fleet was defined as the Cartesian product $X = X_1 \times X_2 \times \dots X_7$. In theory, X can be considered as an ordinary C-space, hence the same standard motion planning algorithms developed for a single vehicle may be applied without particular adaptation. Nevertheless, the main concern is that the dimension of X grows linearly with respect to the number of vehicles, hence the dimension of X might not scale well for real-time applications. Take for instance the case of WiMUST, where for each of the seven vehicles $X_i = \mathbb{R}^2 \times \mathbb{S}^1$, and therefore the total dimension of X is $3 \times 7 = 21$. This large dimensionality would require too many samples to have a coverage of the C-space that would allow to quickly find proper collision-free trajectories from x^s to x^g.

An alternative approach, which was adopted to decrease the computational complexity of the WiMUST planner, is *decoupled prioritized motion planning* [21]. The idea is to sort the vehicles by priority and plan for each vehicle separately, considering higher priority vehicles first. Low priority vehicles then plan by viewing the vehicles with high priority as moving time-dependent obstacles. Suppose the vehicles are sorted by $A_1.A_2, \dots, A_7$, where A_1 has the highest priority. The prioritized planning approach proceeds inductively as follows:

- **Base case:** use the RRT algorithm with kinematic constraints to compute a collision-free trajectory $\overrightarrow{\mathbf{x}}_1$ for vehicle A_1, independently from all other vehicles of the fleet.
- **Inductive step:** suppose that the trajectories $\overrightarrow{\mathbf{x}}_1, \dots, \overrightarrow{\mathbf{x}}_{i-1}$ have been computed for vehicles $A_1, \dots, A_{i-1}$, and that these avoid vehicle and streamers collisions among any of the first $i-1$ vehicles. Let us represent the first $A_1, \dots, A_{i-1}$ vehicles as moving time-dependent obstacles in W for the vehicles with priority larger than $i-1$. In other words, for each time $t \in T$ the configurations of all vehicles $\overrightarrow{\mathbf{x}}_j(t)$, with $j \in \{1, \dots, i-1\}$, are added to the corresponding obstacle regions $O_k(t)$ of vehicles with priorities $k \in \{i, \dots, N\}$.

Since in prioritized decoupled motion planning the obstacle region $O_i(t)$ of each A_i becomes time-dependent (i.e., vehicles high in the hierarchy are seen as moving obstacles by vehicles with low priority), we need to add time as an additional dimension to the state space X. This leads to the new time-dependent state space $\hat{X} = X \times T$. Hence, a state $\hat{x}$ is represented as $\hat{x} = (x, t)$ to indicate the vehicles configurations x at time t. Thus, to use the prioritized decoupled approach, the WiMUST motion planner must perform a search in $\hat{X}$. In this regard, the obstacles' time dependency was taken into account to verify if sampled configurations are collision-free: the validity of a sampled configuration (x_i, t) of A_i at time t is verified by detecting potential collisions with its obstacle region $O_i(t)$ at time t. During the WiMUST project the OMPL was extended to take into account prioritized decoupled motion planning and, consequently, the representation of time-varying obstacles.

On one hand, in terms of computational efficiency, instead of sampling a 21-dimensional space this approach searches for 7 different solutions, with each solution sampled in in a 3-dimensional space, therefore saving a considerable amount of computational time. On the other hand, the drawback is that this approach is not complete, meaning that if the prioritization is not chosen carefully, even with a large number of samples it may not be able to find a solution, if it exists. In WiMUST, the prioritization was chosen by taking into account the inter-vehicle distances in x^s and x^g and the distances from x_i^s to x_i^g for all $i = 1, \dots, 7$.

3) To make the vehicles arrive at the same time: Another requirement for the WiMUST project was to plan trajectories that will allow the vehicles to arrive at the goal configuration simultaneously. This requirement was compulsory to initialize properly the autonomous survey. Once the time dimension is added to the state space of the planning problem, this can be achieved simply by assigning to all the vehicles the same desired time of arrival t^g to the goal configuration $\hat{x}^g = (x^g, t^g)$. This feature allows the planner to find trajectories that arrive at the required final configuration x^g at a specific desired time t^g. One of the problems tackled during the project was how to chose a reasonable t^g. In this regard, the solution adopted was the following: first, compute a motion plan for each vehicle with no desired time of arrival; then, choose t^g as the maximum time among all the times $t_1^g, t_2^g, \dots, t_7^g$ needed to arrive at the corresponding $x_1^g, x_2^g, \dots, x_7^g$ in the previously computed trajectories. As will be described in the following section, to add a desired time to the goal configuration considerably increases the computational complexity of the search performed by the motion planner.

4) To avoid streamers' entanglement: Because some of the vehicles of the WiMUST fleet are equipped with 13 meter long streamers, one of the key requirements of the WiMUST project was to avoid vehicle-streamer collisions and streamer entanglement. To adopt the classical approach of considering the system composed by a vehicle plus its own streamer as a single chain of linked bodies would be too computationally expensive. In fact, even if a complete model of such system were available (as in [4]), to represent the dynamics of a streamer as the one of a chain composed by n joints, at least five dimensions would be needed to model the state of each joint (i.e., two for position, two for velocity, and one for

angle). Hence, to represent the complete system composed by the vehicle plus its streamer would require at least $5 \times n + 3$ dimensions. For instance, with $n = 8$, this would imply covering through sampling a 43-dimensional space for each vehicle, which would require too many computational resources for the computation of an acceptable solution.

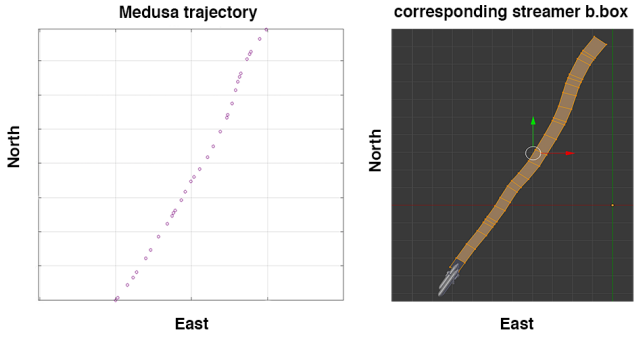


Fig. 3: The streamer bounding box built around the trajectory of a Medusa AMV.

The alternative and more practical approach adopted it the WiMUST project was to construct online a set of bounding boxes around the heuristically predicted positions of the streamers at each time t and then to add them to the corresponding obstacle regions $O_i(t)$. The main assumption behind this heuristics is that, when the vehicles speed is large enough (i.e., at least 0.3 m/s), a streamer essentially follows a trajectory that is identical to that followed by the AMV over a previous time window (this was experimentally observed during the project). To this aim, at each time step t, the WiMUST motion planner keeps a set of buffers with the time-stamped positions of the AMVs equipped with streamers, for the amount of time needed to cover the streamers' length at the current vehicles' speed. Then, for each streamer, the planner constructs a bounding box centered around the coordinates stored in the buffer of its corresponding vehicle and adds it to the obstacle regions $O_i(t)$. To illustrate the output of this method, in Figure 3 the path of a simulated Medusa is compared with the corresponding streamer's bounding box built around it.

III. RESULTS: THE WIMUST TRIALS

The planner was tested at sea at the end of 2018 in Sines, Portugal. A full survey was carried out by the WiMUST team including seven vehicles. The fleet included the ULISSE [5] and Delfim autonomous catamarans, three Medusas [1] and two Folagas [6] autonomous underwater vehicles (see Fig. 1). In the context of the go-to formation task, all the vehicles were at the surface of the sea, even though some had underwater capabilities. Figure 4 shows the initial positions of the vehicles, the calculated paths and the final required positions in one of the performed trials. In the figure, the grey squares represent the initial positions of the vehicles, q_s^x and q_s^y. The final positions q_g^x and q_g^y and heading q_g^θ of each AMV are represented by the grey triangles. The colored lines between

Vehicle	Init East q_s^x	Init North q_s^y	Final East q_g^x	Final North q_g^y	Error
Delfim	-42	-42	0	0	0.17
ULISSE	52	-42	10	0	2.13
$\text{Medusa}_{\text{BLACK}}$	-47	-58	-5	-16	0.54
$\text{Medusa}_{\text{RED}}$	-27	-62	2.5	-10	1.46
$\text{Medusa}_{\text{YELLOW}}$	37	-62	7.5	-10	1.37
Folaga_{54}	-27.5	-62	2.5	-22	0.40
Folaga_{55}	15.5	-82	7.5	-22	0.17

TABLE I: Initial and final positions of the go-to formation task in meters. The error column shows the euclidean distance between the required final positions and the calculated final positions by the planner.

the initial and final position are are the paths calculated by the planner $\overrightarrow{\mathbf{x}}_1, \overrightarrow{\mathbf{x}}_2, \ldots, \overrightarrow{\mathbf{x}}_7$. The sizes of the arrows represent the speeds along each path. The values of the initial and final positions are shown in Table I. Clearly, the short distances among the vehicles when they reach the final configuration make the problem of multiple vehicle motion planning extremely challenging. The difference between the final position of the calculated trajectory compared with the required final position represents the error in the calculated trajectories. The requirements of the WiMUST mission, the dynamics of the AMVs, and the sea conditions, were used to define an acceptance threshold for the difference between real and calculated positions of the vehicles. In Table I, the Error column shows the euclidean difference between the required and calculated final positions in meters. The arrival time t^g for all the AMVs was 106 seconds, with an error of ± 3 seconds. Note that this amount of time is small compared to the time that would be needed if the go-to formation task would have been performed manually. The final heading of the vehicles is also part of the solution space. The pointing direction towards north represents the final desired heading for each AMV. The sinuous paths of the solutions is explained by the different distances that each vehicle has to cover in order to arrive at the final configuration at the same time. Vehicles further from their final positions tend to have a more linear path than the ones closer to their final positions[2].

In the full configuration of the system, on a 3.10GHz 8 cores computer the planner took an average of 5 minutes per vehicle to compute the trajectories. The planner can compute the trajectories for all the vehicles under 5 seconds when the time for the final position is not part of the solution space. This significant difference is due to the time dimension T added to the searched space and to the non-reversible nature of the dynamics of the vehicles. On average, the planner spawned a roadmap graph G composed by $1,000,000$ vertices V per vehicle and the final trajectories were composed of 100 waypoints.

Figures 5a and 5b shows selected views of the WiMUST operational console. The console tracks the positions of the vehicles in real-time. The triangles indicate the positions

[2]The minimum speed of the vehicles is always positive in order to avoid streamers entanglement.

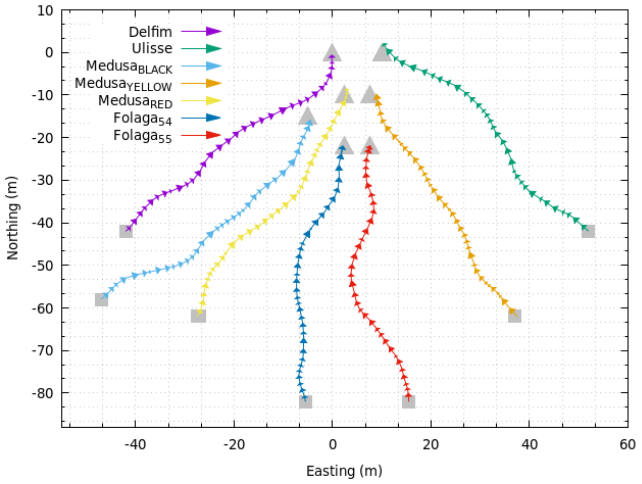
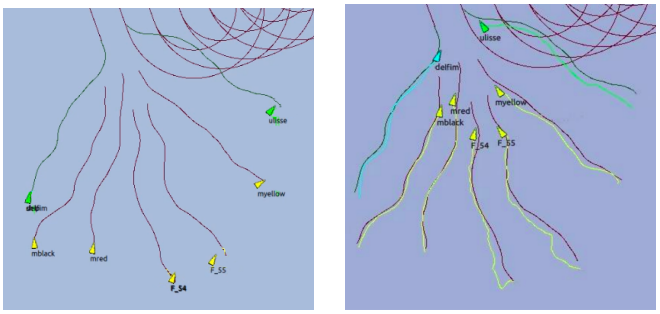


Fig. 4: Calculated paths for each vehicle in the formation. Grey squares represent the initial positions. Grey triangles represent position and heading of the final positions for each vehicle. Arrows represent the headings that the vehicles are required to follow, while the sizes of the arrows represent the speeds along each path.

and headings of the AMVs. The red and green lines are the paths calculated by the planner $\overrightarrow{\mathbf{x}}_1, \overrightarrow{\mathbf{x}}_2, \ldots, \overrightarrow{\mathbf{x}}_7$ that the vehicles must track at assigned speeds. The WiMUST system runs a trajectory tracking algorithm embedded in every vehicle. The trajectory tracker computes the vehicle control references for linear speed and yaw rate required to reduce the distance between the actual position of the vehicle and the next time-stamped position of the desired trajectory, as computed by the mission planner. Figure 5b shows the result of the tracking task. The light blue, yellow, and light green lines show the real paths followed by the vehicles. The differences between the calculated paths and the actual paths are due to several factors that include reading error from the GPS system in each AMV, sea currents, and actuators limits on the vehicles. In Figure 5b, the vehicles are about to reach the final desired positions. After all the vehicles have arrived at the desired configuration, the automated seismic survey starts. Finally, Figure 5c shows a superposition of an image taken from a drone of the real vehicles performing the go-to formation task with the path calculated by the planner.

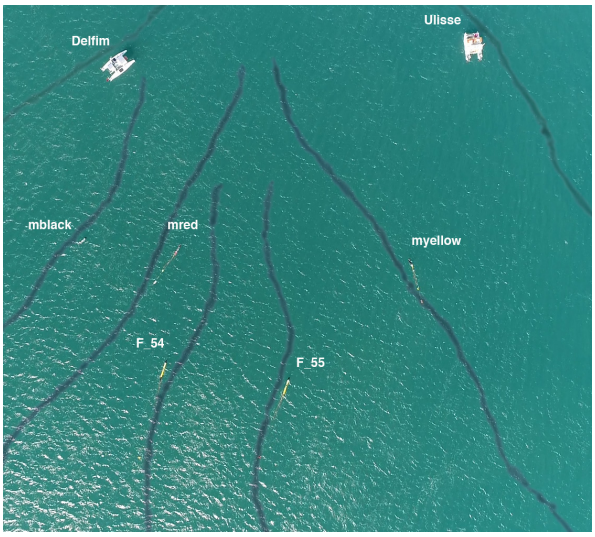
IV. Conclusions

This paper presented the motion planner used in the WIMUST project to generate reference trajectories for multiple vehicle to steer them from initial scattered positions at sea to desired target positions, prior to mission execution, while avoiding vehicle collisions and streamer entanglements. The planner uses a prioritized decoupled approach to overcome the computational cost of planning in the space consisting of the Cartesian product of all vehicle configurations. This approach allowed for the WiMUST system to plan for the motions of several vehicles, while keeping the solution search time within bounds acceptable for real-time usage.



(a) Console initial view

(b) Console final view



(c) Actual paths obtained during sea tests superimposed with the vehicles photographed by a drone during a trial.

Fig. 5: (a) and (b) show two views of the WiMUST operational console for real-time tracking. The triangles show the positions and headings of the vehicles. The lines in (a) represent the paths that the vehicles have to follow to arrive at their desired formation. (b) shows the tracking of the paths by the vehicles. (c) shows the real vehicles photographed by a drone following the superimposed trajectories.

The planner includes geometrical and kinematical constraints for the streamers and the vehicles. These constraints, together with the inclusion of time in the state space, allow for solutions that are collision and entanglement free and realisable by the AMVs in the open sea. The trials at sea showed that the WiMUST motion planner's trajectories and the effective tracking systems of the vehicles yield safe (no collisions nor streamer entanglement), controlled behavior (respecting the AMVs dynamics) and operability (all the vehicles arrived at the same time at the formation point regardless of the distance and initial orientation with respect to the goal).

References

[1] Pedro Caldeira Abreu, João Botelho, Pedro Góis, António Pascoal, Jorge Ribeiro, Miguel Ribeiro, Manuel Rufino, Luís Sebastião, and Henrique Silva. The MEDUSA class of autonomous marine vehicles and their role in EU projects. In *OCEANS 2016-Shanghai*, pages 1–10. IEEE, 2016.

[2] Pedro Abreu et al. Widely scalable mobile underwater sonar technology: An overview of the H2020 WiMUST project. *Marine Technology Society Journal*, 50(4):42–53, jul 2016.

[3] A Pedro Aguiar, Florian A Bayer, John Hauser, Andreas J Häusler, Giuseppe Notarstefano, Antonio M Pascoal, Alessandro Rucco, and Alessandro Saccon. Constrained optimal motion planning for autonomous vehicles using PRONTO. In *Sensing and Control for Autonomous Vehicles*, pages 207–226. Springer, 2017.

[4] Gianluca Antonelli. Dynamic modelling of a streamer of hydrophones towed with an autonomous underwater vehicle. In *Modelling and Simulation for Autonomous Systems: 4th International Conference, MESAS 2017, Rome, Italy, October 24-26, 2017, Revised Selected Papers*, volume 10756, page 179. Springer, 2018.

[5] Gianluca Antonelli, Filippo Arrichiello, Andrea Caiti, Giuseppe Casalino, Daniela De Palma, Giovanni Indiveri, Matteo Razzanelli, Lorenzo Pollini, and Enrico Simetti. ISME activity on the use of autonomous surface and underwater vehicles for acoustic surveys at sea. *ACTA IMEKO*, 7(2):24–31, 2018.

[6] Andrea Caffaz, Andrea Caiti, Giuseppe Casalino, and Alessio Turetta. The hybrid glider/auv folaga. *IEEE Robotics & Automation Magazine*, 17(1):31–44, 2010.

[7] Ronald Choe, Javier Puig, Venanzio Cichella, Enric Xargay, and Naira Hovakimyan. Trajectory generation using spatial pythagorean hodograph bézier curves. In *AIAA Guidance, Navigation, and Control Conference*, page 0597, 2015.

[8] Moritz Diehl, Hans Georg Bock, Holger Diedam, and P-B Wieber. Fast direct multiple shooting algorithms for optimal robot control. In *Fast motions in biomechanics and robotics*, pages 65–93. Springer, 2006.

[9] Jonathan D Gammell, Siddhartha S Srinivasa, and Timothy D Barfoot. Informed rrt*: Optimal sampling-based path planning focused via direct sampling of an admissible ellipsoidal heuristic. *arXiv preprint arXiv:1404.2334*, 2014.

[10] Andreas J Häusler, Alessandro Saccon, António Pedro Aguiar, John Hauser, and António M Pascoal. Energy-optimal motion planning for multiple robotic vehicles with collision avoidance. *IEEE Transactions on Control Systems Technology*, 24(3):867–883, 2016.

[11] D. Manocha J. Pan, S. Chitta. Fcl: A general purpose library for collision and proximity queries. In *International Conference on Robotics and Automation (ICRA)*, pages 3859–3866. IEEE, 2012.

[12] Isaac Kaminer, António M Pascoal, Enric Xargay, Naira Hovakimyan, Venanzio Cichella, and Vladimir Dobrokhodov. *Time-Critical Cooperative Control of Autonomous Air Vehicles*. Butterworth-Heinemann, 2017.

[13] Sertac Karaman and Emilio Frazzoli. Sampling-based algorithms for optimal motion planning. *The international journal of robotics research*, 30(7):846–894, 2011.

[14] Yoshiaki Kuwata, Gaston A Fiore, Justin Teo, Emilio Frazzoli, and Jonathan P How. Motion planning for urban driving using rrt. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 1681–1686. IEEE, 2008.

[15] Steven M LaValle. *Planning algorithms*. Cambridge University Press, 2006.

[16] Steven M. LaValle and J. J. Kuffner. Rapidly-exploring random trees: Progress and prospects. pages 293–308, 2001.

[17] L Ryan Lewis, I Michael Ross, and Qi Gong. Pseudospectral motion planning techniques for autonomous obstacle avoidance. In *Decision and Control, 2007 46th IEEE Conference on*, pages 5997–6002. IEEE, 2007.

[18] Tim Mercy, Ruben Van Parys, and Goele Pipeleers. Spline-based motion planning for autonomous guided vehicles in a dynamic environment. *IEEE Transactions on Control Systems Technology*, 2017.

[19] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y Ng. ROS: an open-source robot operating system. In *ICRA workshop on open source software*, volume 3, page 5. Kobe, Japan, 2009.

[20] Ioan A Sucan, Maciej Moll, and Lydia E Kavraki. The open motion planning library. *Robotics & Automation Magazine*, 19(4):72–78, 2012.

[21] Jur P Van Den Berg and Mark H Overmars. Prioritized motion planning for multiple robots. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 430–435. IEEE, 2005.