Uninterrupted Path Planning System for Multi-USV Sampling Mission in a Cluttered Ocean Environment

Somaiyeh MahmoudZadeh[1], Amin Abbasi[2], Amirmehdi Yazdani[3], Hai Wang[3], Yuanchang Liu [4]

[1]School of Information Technology, Deakin University, Geelong, VIC 3220, Australia,
[2]Department of Electrical Engineering, Azad University of Khoemeinishar, Esfahan, Iran,
[3]College of Science, Health, Engineering and Education, Murdoch University, Perth, WA 6150, Australia,
[4]Department of Mechanical Engineering, University College London, Torrington Place, London WC1E 7JE, UK

Abstract- This paper presents an uninterrupted collision-free path planning system that facilitates the operational performance of multiple unmanned surface vehicles (USVs) in an ocean sampling mission. The proposed uninterrupted path planning system is developed based on the integration of a novel B-Spline data frame and particle swarm optimization (PSO)-based solver engine. The new B-spline data framing structure provides smart sampling of the candidate spots without needing full stop for completing the sampling tasks. This enables the USVs to encircle the area smoothly while simultaneously correcting the heading angle toward the next spot and preventing sharp changes in the vehicle's heading. Then, the optimization engine generates optimal, smooth, and constraint-aware path curves for multiple USVs to conduct the sampling mission from start point to the rendezvous point. The path generated incorporates controllability over the vehicles' velocity profile to prevent experiencing zero velocity and frequent stop/start switching of the controller. To achieve faster convergence of the optimization routine, a suitable search space decomposition scheme is proposed. Extensive simulation studies emulating a realistic ocean sampling mission are conducted to examine the feasibility and effectiveness of the proposed path planning system. This encapsulates modelling a realistic maritime environment of Indonesian Archipelago in Banda Sea including ocean waves, obstacles, and no-fly zones and introducing several performance indices to benchmark the path planning system performance. This process is accompanied by a comparative study of the proposed path planning system with a well-known state-of-the art piecewise, rapidly exploring random tree (RRT), and differential evolution-based path planning algorithms. The results of the simulation confirm the suitability and robustness of the proposed path planning system for the uninterrupted ocean sampling missions.

Keywords- Ocean sampling mission, Unmanned surface vehicle, B-Spline data frame, Uninterrupted path, Bounded velocity profile, Multi-agent systems

Nomenclature			
$\mathcal{V}_c^{\mathbb{V}}$	Two-dimensional ocean wave velocity vector	ϑ	Spline control points
u_c, v_c	The horizontal profile of wave vector	m	Number of control points
$\mathcal{O}$	The centre of ocean turbulence	$B_{i,\mathcal{K}}$	Curve blending function
$\mathscr{r}$	The radius of ocean turbulence	$\mathcal{K}$	Smoothness coefficient,
$\mathfrak{I}$	The strength of the turbulence	$\mathfrak{B}_{\vartheta}^{i}$	The predefined bounds for control points
$\mathbb{V}$	2D spatial space	$\mathcal{L}_{\vartheta}^{i}$	Lower bound of the respective search regions
ψ_c	The direction of wave profile in the horizontal plane	$\mathcal{U}_{\vartheta}^{i}$	Upper bound of the respective search regions
$\mathscr{o}_c$	The update rate of wave profiles	$\vec{\mathscr{t}}_q$	The query-time vector
$\mathcal{T}^n$	Sampling task	$\vec{T}_s$	Time-sample of the spline points
n	Number of spots to be sampled	$\vec{\mathcal{P}}$	Local intermediate control points
$t^{\mathcal{T}^i}$	Task completion time	l	The number of subsets local spline points
$\mathcal{Q}$	Number of the USVs	$\mathbb{L}_{\mathscr{p}}$	The path length
$c_{\mathcal{T}^j}$	The j^{th} cluster of sampling spots	$\vec{\Theta}$	Collision boundary of the forbidden area
$\mathcal{V}_j$	Water reference velocity of the $\mathcal{USV}_j$	$\alpha_{\Theta}^{\mathscr{p}}$	Collision violation function
x_j, y_j	Vehicle's position in the horizontal plane	$\alpha_{\mathcal{V}}^{\mathscr{p}}$	Penalty functions for violation of the USV resultant velocity
u_j, v_j	Vehicle's surge and sway velocities	$\chi_i, \mathscr{Y}_i$	The particle's position
ψ_j	The heading angle of the vehicle	$\vec{\mathfrak{V}}_{\chi}^{i}$	The particle's velocity vector
$\mathbb{P}$	The probability distribution matrix	$\mathcal{W}$	The inertia weight to balance PSO local and global search
$\vec{x}_{\mathrm{px}}, \vec{y}_{\mathrm{px}}$	The coordinate vectors of the pixels/grids in the contour map	c_1, c_2	Acceleration coefficients
$\mathscr{p}_j$	The path curve for $\mathcal{USV}_j$ From start point to the rendezvous point	ℓ_h, ℓ_v	Length and width of the confined of horizontal and vertical axis

1 Introduction

In recent years, USVs are increasingly being used in a broad range of applications from water sampling to military operations [1, 2]. Exploiting USVs to perform dangerous missions, such as maritime patrolling and coastal guarding, has remarkably reduced the risk of the marine accidents associated with human error [3]. Autonomous marine vehicles can also be utilized to monitor and sample the oceanic environments, which is pivotal for long-term planning and securing water resources [4]. The water-sampling missions follow different purposes such as biological investigations [5], measuring the amount of pollution [6], measuring and sampling mineral material and chemical qualities [7], observing physical phenomena like water columns or warming trend, as well as studying the sediment in seabed [8]. However, the navigation of USVs has been always a complex task as it deals with uncertainties and disturbances associated with dynamic surface wave, wind, and water-current involved in the maritime environment. In particular, the major challenge, posed by multiple USVs' operations in a complex ocean environment, is to provide safe navigation and autonomous obstacle avoidance while guaranteeing conflict-free missions in the presence of other maritime traffic [9]. To ensure the safe and reliable operation and quick response to the environmental changes, the motion/path planning problem should be addressed properly.

Path planning is a key enabling element in USV's software development and autonomy enhancement. It facilitates the successful and reliable USV's voyage by generating optimal paths encapsulating the vehicular and environmental constraints. Substantial optimality criteria such as path length, safety margin, operation and rendezvous time, are usually embodied in the path planning problem statement and formulation. In general, the path planning systems and methods can be categorized into two classes: offline and online. In the offline methods (known as pre-generative methods), the path and corresponding waypoints are generated prior to the USV mission based on the pre-collected information of operation field and mission scenario. Whereas, in the online path planning (known as reactive methods as well), the path and corresponding waypoints are generated based on the instantaneous perception of environment and information provided by situation awareness modules while the USV is en route [10-12].

There exists numerous research works in the state-of-the art proposing both offline and online motion/ path planning and collision avoidance systems for a single USV employed in different maritime missions. Examples of these methods include but not limited to A* and Dijkstra algorithms [13, 14], artificial potential field (APF) methods [15, 16], fast marching algorithms [3, 11, 17], graph-based path planning [18, 19], evolutionary algorithms-based path planning [20-23], and extended state observer (ESO) and virtual ship (DVS) principle-based planner [24, 25]. For instance, in [26], the USV path planning was formulated as a multi-objective nonlinear optimization problem and enabled a USV to achieve the shortest, safest, and most economical path while considering static water disturbances and scattered obstacles. This study applied a multi-objective PSO algorithm to find the optimal path from the Pareto optimal solutions subject to the collision avoidance and velocity boundaries. However, in this work a little attention was paid to environmental effects and the computational efficiency of the algorithm was not analyzed. In [19], the USV motion planning was implemented using a topology-based method, in which the obstacles' and the USV's topological position relationship were utilized to accommodate a general map for the safe navigation. In this study, the dynamic constraints of the USV were considered and a decomposition model which simplified the search process of the path planning system was used to help the algorithm to avoid trapping in local optimal trajectories. In [27], a modified Theta* algorithm integrated with a limit-cycle circle set was proposed for USV path planning incorporating the heading angle. In this work, the limit cycle circle set calculated the minimum turning radius of a USV to consequently generate a configuration space of an occupancy grid map. The simulation results confirmed the effectiveness of this algorithm that can incorporate the 3-D kinematic state of the USV for global path planning. To further improve the computational effectiveness and optimality of the path length for global path planning, an improved version of the Theta* algorithm, called angular rate-constrained Theta* (ARC-Theta*) considering both heading rate and heading angle of the USV was proposed [28]. This path planning system bridged the gaps of algorithms performed on SE(2) and SE(3) grid map in that the size of graph must be expanded to facilitate operatable angular resolution. Moreover, constant angle at each orientation node as well as the concern of real-time operation was addressed.

To leverage performance efficiency, operation persistency, and robustness, multiple USVs are usually employed for undertaking various maritime missions. Therefore, significant research works have been devoted in developing different coordinated, cooperative, and distributive planning and control mechanisms enabling reliable and cost-effective operations of multiple USVs [29-31]. In this regard, in [32], a cooperative motion planning system including two layers of offline and online planning was developed for swarm of USVs operating in a realistic maritime environment. The proposed planning system employed a constrained A* approach (to produce an optimal path in offline planning) and virtual target (VT) framework combined with swarm aggregation technique to provide global and local collision avoidance of the USVs. In [11], a multi-layered fast marching technique for solving USVs path planning problem was developed. In this study, a time-variant maritime setting incorporating the repelling and propelling surface current forces was considered. The proposed path planning system in this work was examined in a well-designed simulated operating field and generated optimal mission length as well as minimum energy consumption for the USVs. In [20], a modified version of the genetic algorithm (GA) was proposed for the path planning of USVs in a submarine searching mission, where the algorithm's search mechanism and the convergence speed were enhanced by adaptively controlling the evolution direction and mutation amplitude. Although the modified algorithm performed better stability compared to the conventional GA, the testing environment was not precisely modeled and uncertainties of environmental conditions were not considered in the simulation. In [33], a deep reinforcement learning-based path planning and collision avoidance system was developed for the USVs operation in uncertain environments. In this work, the integration of deep Q-learning (DQN) and visually simulated environment was used to train the agent with the best strategies. Furthermore, the APF was employed to improve the action space and the reward function in the DQN network. The performance of this method was examined in the simulated environment while considering the International Regulations for Preventing Collisions at Sea. Similarly, in [34], a DQN path planning approach was developed for safe and collision-free motion of USVs in maritime environments. The proposed system was utilized the semi-Markov decision making and deep neural network to recognize and analyze the ambiguous collision encountering situations. The path planning system was verified through the experimental tests and the results confirmed the suitability of the method in handling multi-ship collision avoidance problem. However, the computational speed of the path planning system is highly under mercy of the visual image information used as the input for the DQN. In [35], a multi-layer system for motion planning and control of a group of surface vehicles in a complex maritime environment was designed. In this study, the steering system was designed by the vector-field guidance algorithm, and a PID-based feedback controller established the vehicles' speed control. This study conducted the fast-marching method for the vehicle's path planning, where avoiding the collision with static and dynamic obstacles was handled using a fuzzy-logic-based approach. In [36], a self-organized mapping method was used for a multi-USVs task assignment problem, where the path planning problem was addressed using the fast-marching method. The main consideration of the proposed framework was to avoid non-traversable areas in the maritime environment, as well as reducing energy consumption and keeping the vehicles within the communication range. In [37], a swarm control strategy was used for path planning of multiple USVs, where the APF algorithm was employed to handle obstacle avoidance in the presence of dynamic disturbances. The guidance system was designed using the swarm center position steering algorithm, and the solution was modified using neural network approximators. Finally, in [38], a two- layer path planning framework was proposed for cooperative motion planning of the swarm of USVs. In the primary level of the framework, a safety distance constrained A* algorithm was used to produce a collision free path. This solution was smoothened in the later stage using the spline approach. The control-points along the path were smoothly connected by spline interpolation with the aim of avoiding obstacles, where the environmental model was simplified using a binary map method. The proposed multi-layer approach was assessed using a variable number of USVs; however, the environmental modelling was too simple to emulate the real-world situations.

In the mentioned works above, often, the path planning system in the large-scale environment focuses on one optimality objective and its performance is limited just for generation of the position coordinate of the USVs with no consideration of velocity and heading angle profiles. This is quite important in some maritime mission such as water spot sampling that the USVs have some controllability over the velocity and heading profiles to maneuver around the target area based on smooth transition of the velocity vector and without the need for frequent switching on and off the USV's control system. In addition, some methods such as as Dijkstra, A*, and DQN have been proven computationally expensive and unpractical in dealing with large-scale and complex environments. Considering constraints associated with the USV's payload and operation endurance, it is quite essential to benefit from a motion/ path planning system that engage all USVs optimally and adaptively in handling the tasks of mission (e.g., sampling) and thus avoiding extra workload and longer execution time by providing parallel and uninterrupted operations in complex and dynamic environment. However, these aspects have been less considered and addressed in the previous works.

Motivated by the mentioned gaps and unaddressed USV's path planning aspects, the main contribution of this research is to develop an uninterrupted path planning system that facilitates mission of multiple USVs in the ocean sampling operation. The proposed path planning system receives the information of sampling spots in form of a series of ordered tasks and then uses a novel B-spline data frame and the PSO engine to generate optimal, smooth, and constraint-aware path curves. The proposed B-spline data frame enables the path planning system to apply more controllability over the USVs' velocity profile to prevent zero velocity experiencing and frequent stop/start switching of the controller. This enables USVs to encircle the task area smoothly while simultaneously correcting the heading angle toward the next task spot and preventing sharp changes in the vehicle's heading angle. As a result, an artifact of singular arch on the velocity, heading, and heading rate profiles is obviated and USVs' controller can guarantee the USVs' persistent maneuverability in practice. To enhance the performance of the PSO search engine and thus achieving faster convergence rate, a suitable search space decomposition scheme is also proposed. The path planning system of this study consequently enables USVs in handling multiple performance objectives such as path length, safety margin, and smoothness of the velocity and heading profiles while computationally is suitable for real-time implementation. Moreover, it is robust enough in dealing with wave and current disturbances and capable of handling irregular operation fields and immediate obstacle avoidance.

To evaluate the performance of the proposed path planning system, extensive simulation studies based on a benchmark simulated model of Indonesian Archipelago in Banda Sea including ocean waves, obstacles, and no-fly zones are conducted. This encompasses the comparative performance assessment with benchmark piecewise and RRT path planning systems together with the quantitative performance analysis in terms of different optimization criteria to demonstrate the effectiveness and robustness of the proposed USVs planning system for sustainable and efficient ocean sampling missions.

The rest of the paper is organized as follows. Section 2 describes the problem statement and environmental modelling and distribution of the sampling spots. Section 3 provides details of the uninterrupted path planning system. Section 4 emphasizes on implementation of the PSO algorithm for constraint multi USVs path planning. Section 5 demonstrates extensive simulation studies and performance evaluation of the proposed algorithm in qualitative, quantitative and comparison scenarios. Finally, section 6 concludes the paper.

2 Problem Statement and Formulation

This study aims to employ multiple USVs to conduct the operation of water quality sampling in Indonesian Archipelago in Banda Sea. The rationale behind using multiple USVs is to increase the persistence and efficiency of a mission and decrease the cost of operation and deployment. Figure 1 illustrates the captured snapshot of the chosen operating field near Archipelago in Banda Sea with the latitude of ⟨1.76° S, 119.90° E⟩ to ⟨1.76° S, 127.42° E ⟩ and longitude of ⟨7.86° S, 119.90° E ⟩ to ⟨7.86° S, 127.42° E⟩. The operation field encompasses forbidden operation zones, coordinates of start and endpoint, and distribution and coordinates of the sampling areas which should be mathematically defined prior to the implementation of the motion/path planning system. The operating conditions of Banda Sea demands to perform the sampling task without interruption. Thus, the USVs employed in this study should be capable of safe and uninterrupted maneuverability with a trustable degree of freedom on the motion profile; i.e., velocity profile. Each sampling task should be completed within a specific time frame and consequently the mission should be terminated in a pre-defined operation time.

The optimal order of visiting the water sampling areas is obtained using the recently developed heuristic fleet cooperation (HFC) algorithm [39] and given to the USVs a priori. In addition, a probabilistic map of the Banda Sea area (see Section 2.1) is given to the USVs prior to the mission. The USVs use the probabilistic map and their on-board navigation aids for localization, situational awareness, and planning purposes. The USVs should be capable of maneuvering around the sample spots and collect the data while simultaneously keeping the safe distance with the reef and obstacle margins and preventing water-turbulence areas defined as the no-fly zones. Finally, they should approach to a pre-determined rendezvous point once the water sampling mission has been completed.

2.1 Model of the Operating Environment

In maritime missions, providing a precise offline map is rarely attainable; however, the USVs can benefit from any partially constructed map to have a rough estimation of the operating field. As shown in Figure 1, the South-West area of the map is the shoreline, and numerous small islands and cliffs spread over the water-covered area, where the vehicles should always keep the safety margins from these forbidden zones for operation.

On the other hand, the ocean wave data can be obtained from remote devices (e.g., satellite observations) or numerical estimation models. Many ocean waves predictive models have been developed and used for ocean-based research and explorations [40-43]. The following is the two-dimensional ocean turbulence model in which the water circulation gradually changes over the time following the same pattern:

$$\mathcal{V}_c^{\mathbb{V}}:(u_c,v_c)=f(\mathcal{O},\mathfrak{I},\mathscr{r})$$

$$\mathcal{V}_c^{\mathbb{V}}=\begin{cases} u_c=\dfrac{(\mathfrak{I}y-\mathfrak{I}y_0)\left(e^{-(\mathbb{V}-\mathcal{O})^2\mathscr{r}^{-2}}-1\right)}{2\pi(\mathbb{V}-\mathcal{O})^2} \\ v_c=\dfrac{(\mathfrak{I}x-\mathfrak{I}x_0)\left(1-e^{-(\mathbb{V}-\mathcal{O})^2\mathscr{r}^{-2}}\right)}{2\pi(\mathbb{V}-\mathcal{O})^2} \end{cases} \tag{1}$$

$$\begin{aligned} u_c &= |\mathcal{V}_c^{\mathbb{V}}|\cos\psi_c \\ v_c &= |\mathcal{V}_c^{\mathbb{V}}|\sin\psi_c \end{aligned} \tag{2}$$

where $\mathcal{V}_c^{\mathbb{V}}$ represents two-dimensional ocean wave velocity vector, u_c and v_c are the horizontal profile of wave vector in the geographical coordinate (NED) ; $\mathcal{O}$ and $\mathscr{r}$ are the centre and radius of ocean turbulence; $\mathfrak{I}$ corresponds to the strength of the turbulence in a 2D spatial space $\mathbb{V}$. The ψ_c corresponds to the direction of wave profile in the horizontal plane. To calculate the subsequent circulation patterns, a recursive Gaussian function is applied to the $\mathcal{O}$, $\mathfrak{I}$, and $\mathscr{r}$ parameters of the wave as follows:

$$\begin{aligned} \mathcal{O}_t &= \begin{bmatrix} \mathcal{O}_{t-1} & 0 \\ 0 & \mathcal{O}_{t-1} \end{bmatrix} + \begin{bmatrix} \sigma_c(t).\mathbb{X}_{t-1}^{\mathcal{O}_x} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \sigma_c(t).\mathbb{X}_{t-1}^{\mathcal{O}_y} \end{bmatrix} \\ \mathscr{r}_t &= \begin{bmatrix} \mathscr{r}_{t-1} & 0 \\ 0 & \mathscr{r}_{t-1} \end{bmatrix} + \begin{bmatrix} \sigma_c(t).\mathbb{X}_{t-1}^{\mathscr{r}} \\ 0 \end{bmatrix} \\ \mathfrak{I}_t &= \begin{bmatrix} \mathfrak{I}_{t-1} & 0 \\ 0 & \mathfrak{I}_{t-1} \end{bmatrix} + \begin{bmatrix} \sigma_c(t).\mathbb{X}_{t-1}^{\mathfrak{I}} \\ 0 \end{bmatrix} \end{aligned} \tag{3}$$

where $\sigma_c(t)$ is the update rate of wave profiles in time t, and $\mathbb{X}_{t-1}^{\mathcal{O}_x}\sim\mathbb{N}(0,\sigma_{\mathcal{O}_x})$, $X_{t-1}^{\mathcal{O}_y}\sim\mathbb{N}\left(0,\sigma_{\mathcal{O}_y}\right)$, $\mathbb{X}_{t-1}^{\mathscr{r}}\sim\mathbb{N}(0,\sigma_{\mathscr{r}})$, $\mathbb{X}_{t-1}^{\mathfrak{I}}\sim\mathbb{N}(0,\sigma_{\mathfrak{I}})$ are Gaussian distributions of ocean wave components.

To convert the map to the mathematical model, the contour mapping method based on the satellite captured snapshot is utilized. The contingency for distribution of sampling fields is modeled using a numerical estimation model of the field and translated into a probability distribution matrix. Information of coastlines and islands locations are known priori and included

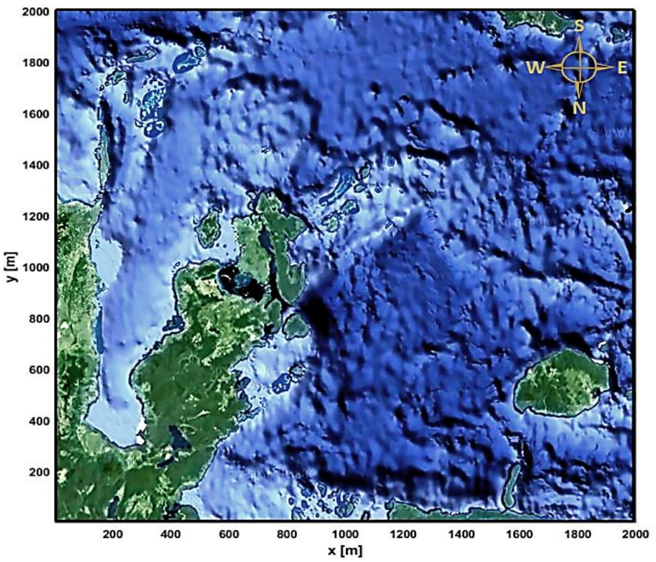


Fig.1 A snapshot of selected operation zone from Indonesian Archipelago in Banda Sea with the Latitude of ⟨1.76° S, 119.90° E⟩ to ⟨1.76° S, 127.42° E ⟩ and longitude of ⟨7.86° S, 119.90° E ⟩ to ⟨7.86° S, 127.42° E⟩.

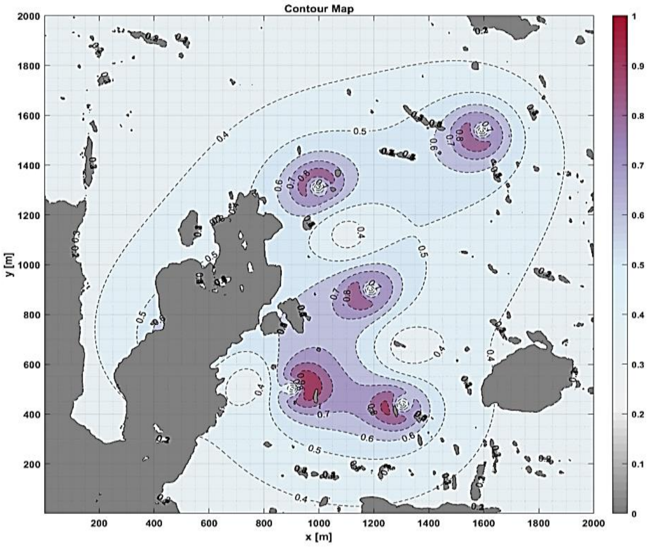


Fig.2 Contour map model of Indonesian Archipelago in Banda Sea.

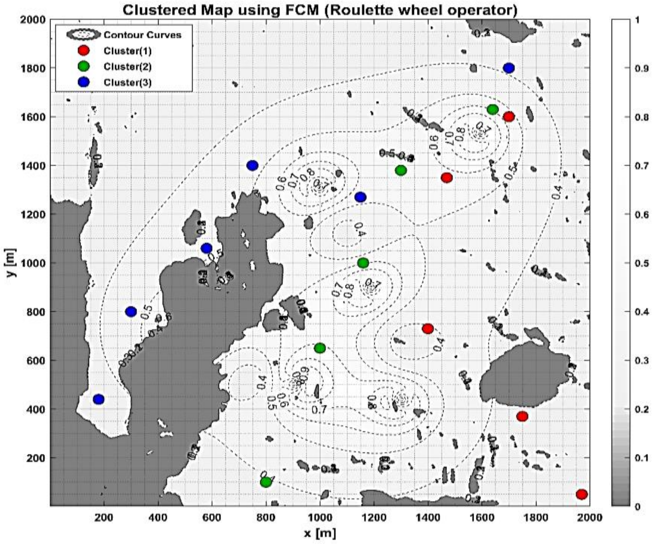


Fig.3 Clustering, ordering, and distribution of sampling spots as several tasks in the operating field.

in the map. The strength of the water turbulence along with the distribution of the sampling areas in the $\mathbb{V}_{2D}:\langle 2000 \times 2000\ m^2\rangle_{x-y}$ area is depicted by Figure 2.

The size of the contour map is considered as $2000\ m \times 2000\ m$, and each pixel/grid in the map is specified by a value between 0 and 1 in probability distribution matrix, where the value of zero corresponds to the forbidden areas of operation such as shore lands and surface objects on the map. Hence, the gray areas on the given contour model in Figure 2 are non-traversable for surface vehicles. Any value in the range of (0, 1] represents the intensity of wave (turbulence) on the map. Accordingly, as the value approaches the 1, the area on the contour map gets reddish that means the area is hazardous for deployment and should be avoided.

2.2 Water Sampling Mission

To perform the water sampling mission efficiently and robustly, the respective sampling spots are defined as a series of tasks and optimally distributed between the USVs. This can be achieved by organizing, clustering, prioritizing, and accomplishment of tasks in the best possible order. In the first stage, a *clustering* process needs to be applied to categorize the most similar tasks in groups of confined areas. In this section, the initial task population (sampling spots) is generated where each individual comprises a random sequence of tasks with uniform probability, given in (4):

$$\begin{aligned}&\mathcal{T}^n \in \mathbb{U}(n_{min} = 10, n_{max} = 100)\\&\mathcal{T}^i \in \mathcal{T}^n = \{\mathcal{T}^{n_{min}}, \dots, \mathcal{T}^i, \dots, \mathcal{T}^{n_{max}}\}\\&\forall \mathcal{T}^i, \exists\ t^{\mathcal{T}^i} \sim \mathbb{U}(60,70)\ (sec)\end{aligned} \tag{4}$$

where, $\mathcal{T}^n$ denotes the sampling task, n is the number of spots to be sampled; each task $\mathcal{T}^i$ associated with a completion time $t^{\mathcal{T}^i}$ depends on the vastness of sampling spot. Assuming there are $\mathcal{Q}$ number of identical USVs in the fleet to be deployed in the operating field, the task data should be partitioned into $\mathcal{Q}$ clusters (exclusive groups of tasks) to avoid multi vehicle mission overlap, which is given by (5):

$$\begin{aligned}&\mathcal{USV} = \langle \mathcal{USV}_1, \dots, \mathcal{USV}_{\mathcal{Q}}\rangle\\&\forall \mathcal{USV}_j \in \mathcal{USV}, \exists\ c_{\mathcal{T}j} \subset \langle c_{\mathcal{T}^1}, \dots, c_{\mathcal{T}^{\mathcal{Q}}}\rangle\end{aligned} \tag{5}$$

where, $\mathcal{USV}_j$ is the *j^th^* vehicle in the fleet of $\mathcal{Q}$, $c_{\mathcal{T}j}$ is the *j^th^* cluster of sampling spots (set of tasks) out of $\mathcal{Q}$ clusters assigned to $\mathcal{USV}_j$. This study takes the advantages of fuzzy clustering method (FCM) to partition a finite collection of n elements $\mathcal{T}^n = \{\mathcal{T}^{n_{min}}, \dots, \mathcal{T}^i, \dots, \mathcal{T}^{n_{max}}\}$ into a collection of $c_{\mathcal{T}j}$ fuzzy clusters with respect to the distribution and characteristics of the tasks $\mathcal{T}^n$. The FCM attempts to minimize an objective function (6):

$$\begin{aligned}&\underset{c_{\mathcal{T}j}}{\operatorname{argmin}} \sum_{p=1}^{n} \sum_{q=1}^{\mathcal{Q}} w_{pq}^m \| \mathcal{T}^p - c_{\mathcal{T}^q} \|^2\\&w_{p,q} = \sum_{k=1}^{c_{\mathcal{T}}} \left(\frac{\| \mathcal{T}^p - c_{\mathcal{T}q} \|}{\| \mathcal{T}^p - c_{\mathcal{T}k} \|} \right)^{\frac{-2}{m-1}}\end{aligned} \tag{6}$$

Given a finite set of data, the FCM passes a list of $c_{\mathcal{T}}$ cluster centers and a partition matrix of $\mathbb{I}_{w_{p,q}} \in [0,1]$, where each element, $w_{p,q}$, is the degree membership to which element, $\mathcal{T}_p$, belongs to cluster $c_{\mathcal{T}^q}$. The $m \geq 1 \in \mathbb{R}$ is a fuzzifier to determine fuzziness level of the cluster. The FCM method offers number of $c_{\mathcal{T}}$ membership values to each task; therefore, each task belongs to all the $c_{\mathcal{T}}$ clusters, where the degree of each task's membership ($w_{p,q}$ for $\mathcal{T}^p$) depends on its distance from the center of the clusters. This approach is taken from the previous work [39], where the HFC was proposed for optimal clustering and allocation of tasks for multi-agent systems. The same approach is utilized to cluster the tasks among the three USVs, where FCM method is applied to exclusively divide the randomly distributed tasks in a non-uniform environment into three identical groups. In Figure 3 the red, green and blue bullets correspond to the sampling spots marked for the vehicles, where each color belongs to one vehicle. It should be noted that this method can contribute to creating quantity balance among clusters when the density of the distributed tasks is non-uniform.

3 The Uninterrupted Path Planning System

In this section, details of developing the proposed uninterrupted path planning system including USV modeling, the essence of USV path planning problem, space decomposition scheme, as well as PSO-based optimization engine are explained.

3.1 Mathematical Model of Multiple USV Operation

Having prior knowledge of the USV's maneuvering capabilities is an essential prerequisite for appropriate modeling of the path planning problem. The motion of the USV is considered as a rigid body model on the horizontal plane to reflect the vehicle's maneuverability in three degrees of freedom (horizontal plane motion in x-, y-, and yaw-direction).

Assumption 1- In this study, motions on the z-direction and changes of, roll and pitch components are considered to be negligible [44, 45].

Ocean waves constantly affect the USVs' motion; therefore, each vehicle's position and heading should be adjusted simultaneously to ensure the vehicle is on the desired track despite the wave forces. The following equation represents the USV's heading model:

$$\forall\ \mathcal{USV}_j\ \text{ at }\ t\text{:}\ \exists\ \langle x_j(t), y_j(t), \psi_j(t)\rangle \tag{7}$$

$$\psi_j(t) = \tan^{-1}\left(\frac{\Delta y_j(t)}{\Delta x_j(t)}\right) \qquad (8)$$

where, the $\mathcal{USV}_j$ denotes the *j^th^* vehicle in the fleet of $\mathcal{Q}$ USVs, x_j, y_j denote the vehicle's position in the horizontal plane, and ψ_j is the heading angle of the vehicle (in the NED coordinate) in time step t. Applying the 2D wave velocity of $\mathcal{V}_c^{\mathbb{V}}\colon \{(u_c, v_c),\ \psi_c\}$, given by (2), the translational velocities of the $\mathcal{USV}_j$ are obtained using (9):

$$\begin{cases} u_j = |\mathcal{V}_j| \cos\psi_j(t) + |\mathcal{V}_c^{\mathbb{V}}| \cos\psi_c \\ v_j = |\mathcal{V}_j| \sin\psi_j(t) + |\mathcal{V}_c^{\mathbb{V}}| \sin\psi_c \end{cases} \qquad (9)$$

where, $|\mathcal{V}_j|$ is the resultant water reference velocity of the $\mathcal{USV}_j$, u_j is the vehicle's surge velocity (along x_j), and v_j is the sway velocity (along y_j).

3.2 Path Planning using B-Spline Curves

The USVs are required to consistently visit the sampling spots and collect sufficient data of the sampling spots in the assigned operation time, while moving toward the rendezvous point and keeping the safety margin with the collision boundaries at all times. Besides, the shore lands and the strong water turbulence areas in the contour model should be avoided by the vehicles. Given the combinatorial nature of the USV's route-task allocation/path planning problem, there should be a compromise among optimizing multiple objectives, including the mission available time, travel length for each vehicle, number and order of the sampled spots, on-time mission completion, and reaching to the predefined destination. This is a combination of a discrete and a continuous optimization problem and categorized as a Non-deterministic Polynomial-time (NP) hard problem. On the other hand, considering the NP-hard complexity of the multiple vehicles path planning problem under the given conditions, obtaining the accurate optimal solution is only applicable to completely known environments without uncertainty, while the modeled environment in this research corresponds to highly uncertain and dynamic terrains.

The path planner should generate time optimal collision-free path curve $\mathscr{p}_j$ (shortest path) from start point to target position passing through the obstacles without collision and visiting the ordered sampling spots, while considering the strong water turbulences. The resultant path should be safe and feasible. The path curve of each vehicle is generated by spline method, which establishes a polynomial interpolation between the control points. Spline curves are captured from a set of control points like $\vartheta = \{\vartheta^1, \dots, \vartheta^m\}$ in the problem space with coordinates of $\vartheta^1\colon (\vartheta_x^1, \vartheta_y^1), \dots, \vartheta^m\colon (\vartheta_x^m, \vartheta_y^m)$, where m is the number of corresponding spline points. These control points play a substantial role in determining the optimal path. The mathematical description of the B-Spline coordinates in a 2-dimensional workspace is given by:

$$\begin{cases} x(t) = \sum_{i=1}^{m} \vartheta_x^i B_{i,\mathcal{K}}(t) \\ y(t) = \sum_{i=1}^{m} \vartheta_y^i B_{i,\mathcal{K}}(t) \end{cases}$$
$$\mathscr{p}_j = [x_j(t), y_j(t)] \approx \sum_1^{|\mathscr{p}_j|} \left(\vartheta_{x,y}^{i+1} - \vartheta_{x,y}^i\right) \qquad (10)$$

where, the $x_j(t)$ and $y_j(t)$ are the coordinated of the $\mathcal{USV}_j$ along the path $\mathscr{p}_j$ at time t, $B_{i,\mathcal{K}}(t)$ is the curve blending function, and $\mathcal{K}$ is a smoothness coefficient, where larger values of $\mathcal{K}$ correspond to smoother curves. $\vartheta_{x,y}^{i+1}$ and $\vartheta_{x,y}^i$ corresponds to the coordinates of the ith and its subsequent control points. All control points should be located in respective search regions constrained to the predefined bounds of $\mathfrak{B}_\vartheta^i = \left[\mathcal{L}_\vartheta^i, \mathcal{U}_\vartheta^i\right]$. Equation (11) shows how the control points in x-y coordinate are generated.

$$\begin{aligned} &\mathfrak{B}_\vartheta^i = \left[\mathcal{L}_\vartheta^i, \mathcal{U}_\vartheta^i\right] \\ &\vartheta^i \in \mathfrak{B}_\vartheta^i \Longrightarrow \mathcal{L}_\vartheta^i \le \vartheta^i \le \mathcal{U}_\vartheta^i \\ &\vartheta_x^i = \mathcal{L}_{\vartheta_x}^i + Rand_x^i \left(\mathcal{U}_{\vartheta_x}^i - \mathcal{L}_{\vartheta_x}^i\right) \\ &\vartheta_y^i = \mathcal{L}_{\vartheta_y}^i + Rand_y^i \left(\mathcal{U}_{\vartheta_y}^i - \mathcal{L}_{\vartheta_y}^i\right) \end{aligned} \qquad (11)$$

Then, general form of spline using a set of spline points $\vec{\vartheta}\colon (\vec{\vartheta}_x, \vec{\vartheta}_y)$ can be calculated from (12):

$$\begin{aligned} \vec{x} &= \text{spline}(\vec{\vartheta}_x, \vec{T}_s, \mathscr{t}_q) \\ \vec{y} &= \text{spline}(\vec{\vartheta}_y, \vec{T}_s, \mathscr{t}_q) \end{aligned} \qquad (12)$$

In (12), $\vec{x}$ and $\vec{y}$ are the path's coordinate vectors containing spline points of $\vec{\vartheta}_{x,y} = \left\{\vartheta_{x,y}^1, \dots, \vartheta_{x,y}^m\right\}, \vec{T}_s$ and $\vec{\mathscr{t}}_q$, where $\vec{T}_s$ contains the corresponding time-sample of the spline points, and $\mathscr{t}_q$ is the query-time vector. The set of spline points $\vec{\vartheta}_{x,y}$ also involves the coordinates of the included tasks on the path that is a vector of $\vec{\mathcal{T}}_{x,y} = \{\mathcal{T}_{x,y}^1, \dots, \mathcal{T}_{x,y}^{\mathscr{k}}\}$. The interval between two consecutive tasks $\mathcal{T}^i$ and $\mathcal{T}^j$ also contains a subset of local control-points $\vec{\mathcal{P}}_{x,y} = \{\mathcal{P}_{x,y}^1 \dots, \mathcal{P}_{x,y}^l\}$, where l is the number of subsets local spline points involved between $\mathcal{T}^i$ and $\mathcal{T}^j$. Thus, the abstract form for the spline vectors of x- and y- coordinate is defined as:

$$\begin{aligned} &\vec{\vartheta}_x\colon \langle \vec{\mathcal{T}}_x, \vec{\mathcal{P}}_x \rangle \\ &\vec{\vartheta}_y\colon \langle \vec{\mathcal{T}}_y, \vec{\mathcal{P}}_y \rangle \end{aligned} \qquad (13)$$

Considering the defined variables, the final form of the spline vector $\vec{\vartheta}_{x,y}$: $(\vec{\vartheta}_x, \vec{\vartheta}_y)$ is as following:

$$\vec{\vartheta}_x = \left\{\vartheta_x^1, \mathcal{P}_x^{11}, \dots, \mathcal{P}_x^{1l}, \mathcal{T}_x^1, \dots, \mathcal{P}_x^{\mathcal{k}1}, \dots, \mathcal{P}_x^{\mathcal{k}l}, \mathcal{T}_x^{\mathcal{k}}, \mathcal{P}_x^{(\mathcal{k}+1)1}, \dots \mathcal{P}_x^{(\mathcal{k}+1)l}, \vartheta_x^m\right\}$$
$$\vec{\vartheta}_y = \left\{\vartheta_y^1, \mathcal{P}_y^{11}, \dots, \mathcal{P}_y^{1l}, \mathcal{T}_y^1, \dots, \mathcal{P}_y^{\mathcal{k}1}, \dots, \mathcal{P}_y^{\mathcal{k}l}, \mathcal{T}_y^{\mathcal{k}}, \mathcal{P}_y^{(\mathcal{k}+1)1}, \dots \mathcal{P}_y^{(\mathcal{k}+1)l}, \vartheta_y^m\right\} \quad (14)$$

Given the mathematical relations and the final spline vectors, $\mathcal{k}$ is number of the tasks, l is the length of each local spline set, while the number of local spline sets is $\mathcal{k}+1$. ϑ_x^1 is the vehicle's initial point, ϑ_x^m indicates the rendezvous point in the spline vector, and $\vec{\vartheta}_x$ and $\vec{\vartheta}_y$ hold the whole spline points with size of m.

In order to assign a time sample to each spline point, one way is to simply split the total time interval into m equal parts, where m is the length of $\vec{\vartheta}_x$ or $\vec{\vartheta}_y$.

$$t_s^i = \frac{i}{m} T_{\text{end}} \quad , \quad i = 0,1,\dots,m \qquad (15)$$

where, T_{end} is the available mission time, which is determined according to the vehicles' battery capacity. According to (12) -(15), the data frames of the three vectors $\vec{\vartheta}_x$, $\vec{\vartheta}_y$ and $\vec{T}_s$ are developed as shown in Figure 4.

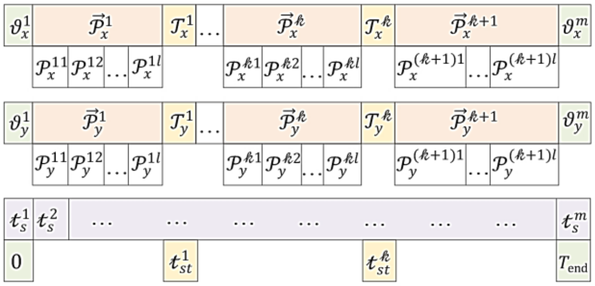
As illustrated in Figure 4, l is the maximum number of local splines placed between every two consecutive tasks in vectors $\vec{\vartheta}_x$ and $\vec{\vartheta}_y$, and $\vec{T}_s$ contains their corresponding time samples obtained by equation (14) and (15). Some of the time samples in $\vec{T}_s$ specifically belong to the task samples, which are also indicated as $\{t_{st}^1, t_{st}^2, \dots, t_{st}^{\mathcal{k}}\}$. Assuming one-second sampling period, the final forms of query-time vector $\vec{t}_q$ will be:

$$\vec{t}_q = \{1,2, \dots . T_{\text{end}}\} \qquad (16)$$

Now, all the vectors are defined for the spline operator, and the path coordinates $\vec{x}$ and $\vec{y}$ are obtained from equation (12). The final path coordinate vectors take the following form:

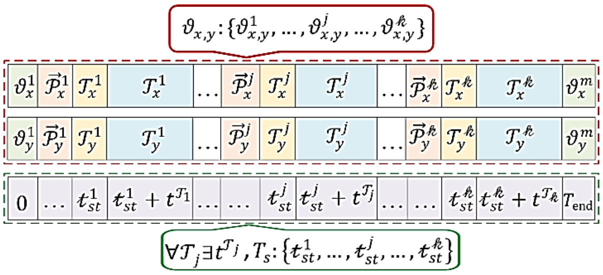
$$\vec{x} = \{x_1, x_2, \dots\ x_{\text{end}}\}$$
$$\vec{y} = \{y_1, y_2, \dots\ y_{\text{end}}\} \qquad (17)$$

The size of $\vec{x}$ and $\vec{y}$ is equal to query-time vector $\vec{t}_q$, and $(x_{\text{end}}, y_{\text{end}})$ is the final point of the path, which should be the rendezvous point.



ϑ_x^1
$\vec{\mathcal{P}}_x^1$
$\mathcal{T}_x^1$
…
$\vec{\mathcal{P}}_x^{\mathcal{k}}$
$\mathcal{T}_x^{\mathcal{k}}$
$\vec{\mathcal{P}}_x^{\mathcal{k}+1}$
ϑ_x^m
$\mathcal{P}_x^{11}$ $\mathcal{P}_x^{12}$ … $\mathcal{P}_x^{1l}$
$\mathcal{P}_x^{\mathcal{k}1}$ $\mathcal{P}_x^{\mathcal{k}2}$ … $\mathcal{P}_x^{\mathcal{k}l}$
$\mathcal{P}_x^{(\mathcal{k}+1)1}$ … $\mathcal{P}_x^{(\mathcal{k}+1)l}$
ϑ_y^1
$\vec{\mathcal{P}}_y^1$
$\mathcal{T}_y^1$
$\vec{\mathcal{P}}_y^{\mathcal{k}}$
$\mathcal{T}_y^{\mathcal{k}}$
$\vec{\mathcal{P}}_y^{\mathcal{k}+1}$
ϑ_y^m
$\mathcal{P}_y^{11}$ $\mathcal{P}_y^{12}$ … $\mathcal{P}_y^{1l}$
$\mathcal{P}_y^{\mathcal{k}1}$ $\mathcal{P}_y^{\mathcal{k}2}$ … $\mathcal{P}_y^{\mathcal{k}l}$
$\mathcal{P}_y^{(\mathcal{k}+1)1}$ … $\mathcal{P}_y^{(\mathcal{k}+1)l}$
t_s^1 t_s^2 … t_s^m
0
t_{st}^1
$t_{st}^{\mathcal{k}}$
T_{end}

Fig.4 Data frames of the spline vectors.



$\vartheta_{x,y}: \{\vartheta_{x,y}^1, \dots, \vartheta_{x,y}^j, \dots, \vartheta_{x,y}^{\mathcal{k}}\}$
ϑ_x^1 $\vec{\mathcal{P}}_x^1$ $\mathcal{T}_x^1$ $\mathcal{T}_x^1$ … $\vec{\mathcal{P}}_x^j$ $\mathcal{T}_x^j$ $\mathcal{T}_x^j$ … $\vec{\mathcal{P}}_x^{\mathcal{k}}$ $\mathcal{T}_x^{\mathcal{k}}$ $\mathcal{T}_x^{\mathcal{k}}$ ϑ_x^m
ϑ_y^1 $\vec{\mathcal{P}}_y^1$ $\mathcal{T}_y^1$ $\mathcal{T}_y^1$ … $\vec{\mathcal{P}}_y^j$ $\mathcal{T}_y^j$ $\mathcal{T}_y^j$ … $\vec{\mathcal{P}}_y^{\mathcal{k}}$ $\mathcal{T}_y^{\mathcal{k}}$ $\mathcal{T}_y^{\mathcal{k}}$ ϑ_y^m
0 … t_{st}^1 $t_{st}^1 + t^{\mathcal{T}_1}$ … … t_{st}^j $t_{st}^j + t^{\mathcal{T}_j}$ … … $t_{st}^{\mathcal{k}}$ $t_{st}^{\mathcal{k}} + t^{\mathcal{T}_{\mathcal{k}}}$ T_{end}
$\forall \mathcal{T}_j \exists t^{\mathcal{T}_j}, T_s: \{t_{st}^1, \dots, t_{st}^j, \dots, t_{st}^{\mathcal{k}}\}$

Fig.5 Data frame of spline vectors involving task points.

3.3 New B-Spline Data Frame for Uninterrupted Path Planning

To generate uninterrupted continuous path curves from start points to the rendezvous point, without needing full stop for completing the sampling tasks, a new B-Spline data frame is developed. Let us suppose each sampling spot $(\mathcal{T}^i)$ requires $t^{\mathcal{T}^i}sec$ to be completed, meaning that the motion of the USVs in the task location needs to be interrupted for $t^{\mathcal{T}^i}$ seconds. In order to accomplish sampling tasks without making any interruption in USVs motion, the path curves need to be modified. Hence, a new element is introduced to the spline vectors to provide smooth motion for USVs and avoid speed overshooting. Figure 5 shows the new data frame of the spline vectors. As discussed earlier, the spline vectors are created as the combination of task points and local spline control points. In order to form the spline vectors, the first step is to identify the task points in the spline vectors. Then, extra spline samples should be defined right next to the task samples with exactly equal value of the corresponding task sample. This results in having no gap between task point and newly generated spline points. Now, new time samples ought to be assigned to the new spline points. Similarly, in the vector $\vec{T}_s$, the place of new samples should be next to the samples that belong to the task points, while taking the value of $t^{\mathcal{T}^i}$ seconds ahead of their corresponding time samples.

As explained earlier, only the fixed points (tasks) are considered to generate the new spline vectors that are modified by new framing to involve task points in the path planning process. Therefore, the data frame focuses on $\mathcal{T}_x^j$, $\mathcal{T}_y^j$ and t_{st}^j samples which are associated with task points in the three spline vectors. While the local splines have no contribution to the task completion process and remain untouched in all three vectors. As shown in Figure 5, some extra splines equal to $\mathcal{T}_{x,y}^j$ are inserted next to the task samples in the data frame vectors meaning that the sampling points of $\mathcal{T}_x^j$ and $\mathcal{T}_y^j$ are duplicated immediately after visiting these points. Accordingly, the new added points are associated with an extra time in $\vec{T}_s$ vector for task completion given by $t_{st}^j + t^{\mathcal{T}^j}$. It is obvious that the first and last points do not need to be involved in this process as the first task is the mission start point, and the last point is the rendezvous point. It should be noted that the newly inserted points are also considered as fixed spline points in the path planning procedure, and the algorithm does not relocate them over the iterations.

3.4 Path Optimization Criteria

The following criteria are introduced and used for the optimization of path curves produced based on the new B-Spline method:

3.4.1 ***Path length minimization:*** the main objective is to minimize the path length $\mathbb{L}_{\wp}(x, y)$, which is given in (18):

$$\mathbb{L}_{\wp}(x, y) = \sum_{i=1}^{|\wp|=T_{\text{end}}-1} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \tag{18}$$

3.4.2 ***Safety margins to avoid collision:*** the resultant path indicated by $\wp$ should be safe and feasible. The environmental constraints are associated with the forbidden zones of the map or intersecting any obstacle, and the vehicle must avoid colliding the obstacles boundary. In the contour mapping method, each point in space is specified by a particular number, representing the depth of the water in the probability distribution matrix, where the elements of the matrix are normalized between 0 and 1.

Remark 1- As shown in Fig.3, the Southeast area of the map includes cliffs and rocks and numerous small barriers spread over the operation area. The collision barriers and rocks' locations are known priori and included in the map. The geographical map is considered as $2000_x \times 2000_y\ [m^2]$ grid and associated with an occupancy matrix $\left([\mathbb{P}]_{xy}\right)$ of the same size. Each pixel in the map indicates an occupancy value of $\mathbb{P}: \left(\vec{x}_{\text{px}}, \vec{y}_{\text{px}}\right) = [0, 1]$, where $\mathbb{P} = 0$ means the pixel is definitely occupied by an obstacle so that $\vec{\Theta} = 1$ and $0 \cdot 8 < \mathbb{P} < 1$ corresponds to plausible risky areas such as water turbulence that must be avoided during the path planning process.

The following definition describes the obstacle-defined pixels/grid in the contour map:

$$\left\{\vec{x}_{\text{px}}, \vec{y}_{\text{px}} \mid \mathbb{P}\left(\vec{x}_{\text{px}}, \vec{y}_{\text{px}}\right) = 0 \text{ or } \mathbb{P}\left(\vec{x}_{\text{px}}, \vec{y}_{\text{px}}\right) > 0 \cdot 8\right\} \subseteq \vec{\Theta}\left(\vec{x}_{\Theta}, \vec{y}_{\Theta}\right) \tag{19}$$

Where $\vec{\Theta} : \left(\vec{x}_{\Theta}, \vec{y}_{\Theta}\right)$ is the vector containing the point-based coordinate of obstacles (including no-flying zones) in the contour map.

The path violates the collision constraint in the i^{th} point, if any point on the path x_i, y_i is located on or inside the safety margin $\vec{\Theta}_j\left(x_{\Theta_j}, y_{\Theta_j}\right)$. The collision penalty function (α_{Θ_j}) for each arbitrary point i on the path $\wp$ is defined by (20):

$$\alpha_{\Theta_j}^{i \in \wp} = \text{size}((x_i, y_i) \in \vec{\Theta}_j(x_{\Theta_j}, y_{\Theta_j})) \tag{20}$$

where the function "size" represents the number of path points located in the obstacle pixels/grids which is denoted as $\alpha_{\Theta}^{\wp}$.

To calculate the accumulated violation of the path $\wp$ from colliding a particular forbidden area of Θ_j, we can capture the mean value of all violations as follows:

$$\alpha_{\Theta}^{\wp} = \sum_{j=1}^{q} \text{mean}\left(\left[\alpha_{\Theta_j}^{i}\right]\right) \tag{21}$$

where, $[\alpha_{\Theta_j}^{i}]$ is the vector which includes the violation of all points for Θ_j, and q is the total number of obstacles. This procedure repeats for all obstacles to obtain the overall collision violation for the path $\wp$.

3.4.3 Violation of transitional velocities: Strong waves may cause drift on the vehicle's motion and push it to the unwanted direction. Hence, another environmental constraint for the path planner is to detect and avoid crossing ocean turbulences. Therefore, the USV's transitional velocities of surge/sway should be confined to the boundaries of u_{max} and v_{max} in all states along the path. The absolute transitional velocity of the USV $|\mathcal{V}|$ in position (x_i, y_i) along the generated path curve $\wp$ can be obtained as follows:

$$\left|\mathcal{V}^{i \in \wp}\right| = \sqrt{\left(\frac{dx_i}{dt_q}\right)^2 + \left(\frac{dy_i}{dt_q}\right)^2} \tag{22}$$

where $\mathcal{V}^{i \in \wp}$ is vehicles absolute velocity in i^{th} point along the path $\wp$. The dx_i, dy_i and dt_q are first-order derivatives of the path curve in x_i y_i w.r.t t_q, respectively. To improve the spline-time vector $\vec{T}_s$, the time assigned to each spline point is calculated based on the distance travelled to reach that point. The linear distance between the spline points before reaching the i^{th} point and the corrected time samples are obtained through (23):

$$\mathbb{L}_i = \sum_{j=1}^{i-1} \sqrt{(x_{sj} - x_{s,j+1})^2 + (y_{sj} - y_{s,j+1})^2}$$
$$t_{si} = \frac{\mathbb{L}_i}{\mathbb{L}_{\wp}} T_{end}, \quad i = 0,1,2, \ldots, m \tag{23}$$

where $\mathbb{L}_i$ is the linear distance to reach the i^{th} spline point, and m is the number of spline points; t_{si} corresponds the time-sample of the i^{th} spline point. The maximum velocity value is determined for the USVs to make the vehicle move within a certain velocity range and avoid surge in velocity profile. Violating the maximum velocity exerts penalty to the cost function. Violation of the USV's transitional velocities of surge and sway in i^{th} point on the path can be achieved as follows:

$$\alpha_u^{\wp} = \text{mean}[\max(0; u(t) - u_{max})]$$
$$\alpha_v^{\wp} = \text{mean}[\max(0; |v(t)| - v_{max})] \tag{24}$$

where, u_{max} and v_{max} are surge and sway constraints associated with the USV's kinematic restrictions, $\alpha_u^{\wp}$ and $\alpha_v^{\wp}$are the penalty functions for violation of path from the given surge and sway limits. The maximum transitional velocity $\mathcal{V}_{max}$ is defined by user, and the violation of each velocity-sample is obtained using (25).

$$\alpha_{\mathcal{V}}^{\mathscr{p}} = \text{mean}\left[\max(0; 1 - \frac{\mathcal{V}_{max}}{|\mathcal{V}^{i \in \mathscr{p}}|})\right] \quad (25)$$

where $\alpha_{\mathcal{V}}^{\mathscr{p}}$ is the violation index of velocity signal from the given maximum transitional speed limit $\mathcal{V}_{max}$.

3.4.4 ***Augmented cost function:*** the path cost function is defined as a combination of the performance indices along with the weighted penalty functions to encounter the overall violation of the path $\mathscr{p}$. Thus, the augmented cost function $(\hat{\mathcal{Z}}_{\mathscr{p}})$ is formulated as follows:

$$\hat{\mathcal{Z}}_{\mathscr{p}} = \mathbb{L}_{\mathscr{p}} + \left(\gamma_1 \alpha_{\Theta}^{\mathscr{p}} + \gamma_2 \alpha_{u}^{\mathscr{p}} + \gamma_3 \alpha_{v}^{\mathscr{p}} + \gamma_4 \alpha_{\mathcal{V}}^{\mathscr{p}}\right) \quad (26)$$

The γ_1, γ_2, γ_3, and γ_4 are corresponding weighting factors for the defined constraints respectively and highlight the impact of each constraint violation on the total cost $\hat{\mathcal{Z}}_{\mathscr{p}}$ calculation.

4 Optimization Engine for USVs' Path Planning

In this section, first, the optimization engine utilized for the proposed USVs' path planning problem is introduced. Then, a search space decomposition technique to speed up the convergence rate of the optimization engine towards optimal solutions is presented.

4.1 PSO Algorithm

In this study, the PSO algorithm is adopted as an optimization engine to attain a proper location of splines' control points so that optimal, smooth, and collision-free paths satisfying all mentioned constrains are achieved. The argument for adopting PSO in solving NP-hard problems is strong enough due to its superior scalability with complex and multi-objective problems [46, 47]. PSO originally operates in a continuous space and is well-adapted to multidimensional spaces(due to its stochastic optimization nature) and hence is an appropriate method for solving path planning problems. Similar to the other evolutionary frameworks, the components of the problem should be determined for the optimizer.

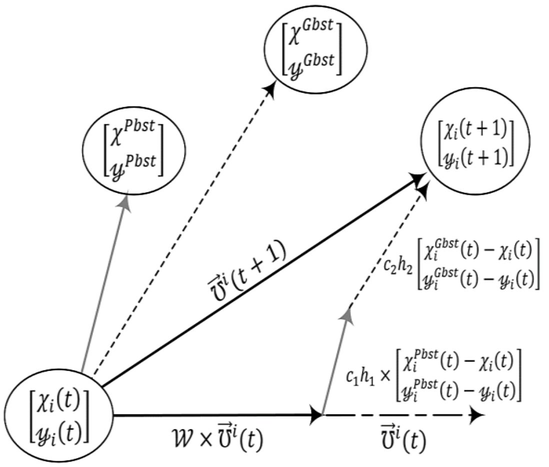
To generate the initial population, individuals should be created for the algorithm. The i^{th} individual in population is a set of spline point, defined as $\vec{\vartheta}^i : (\vec{\vartheta}_x^i, \vec{\vartheta}_y^i)$. As mentioned earlier, the local spline vectors $\vec{\mathcal{P}}_{x,y}$ comprise several sets of splines $\{\mathcal{P}_{x,y}^1 \dots, \mathcal{P}_{x,y}^l\}$, where each set belongs to the path between two consecutive tasks of $\mathcal{T}_{x,y}^i$ and $\mathcal{T}_{x,y}^{i+1}$. The task points $(\mathcal{T}_{x,y}^i)$ in $\vec{\vartheta}^i$ vectors are defined as the fixed points in all particles, while the local splines $(\vec{\mathcal{P}}_{x,y})$ are randomly generated for each of individuals to bypass the objects and forbidden deployment zones. The PSO starts to process with initializing a population of particles using spline points of $\vec{\vartheta}^i : (\vec{\vartheta}_x^i, \vec{\vartheta}_y^i)$, where each particle comprises a position $\chi_i, \mathscr{y}_i$ assigned with $\vec{\vartheta}_x^i, \vec{\vartheta}_y^i$ and velocity components of $\vec{\mho}^i : \left(\vec{\mho}_\chi^i, \vec{\mho}_{\mathscr{y}}^i\right)$ in the search space. The particle's position and velocity get updated iteratively using (27) and (28), although position of the task points cannot be affected by the algorithm's update vector. The new particles are evaluated according to the cost function given in (26). Each particle has a memory to preserve its experience best position of the $\langle \chi^{Pbst}, \mathscr{y}^{Pbst} \rangle$and the global best position of $\langle \chi^{Gbst}, \mathscr{y}^{Gbst} \rangle$ from the previous states.

$$\begin{cases} \chi_i \coloneqq \vec{\vartheta}_x^i \\ \vec{\mho}_\chi^i(t+1) = \mathcal{W}\vec{\mho}_\chi^i(t+1) + c_1 h_1 \left[\chi_i^{Pbst}(t) - \chi_i(t)\right] + \ c_2 h_2 \left[\chi_i^{Gbst}(t) - \chi_i(t)\right] \\ \chi_i(t+1) = \chi_i(t) + \vec{\mho}_\chi^i(t+1) \end{cases} \quad (27)$$

$$\begin{cases} \mathscr{y}_i \coloneqq \vec{\vartheta}_y^i \\ \vec{\mho}_{\mathscr{y}}^i(t+1) = \mathcal{W}\vec{\mho}_{\mathscr{y}}^i(t+1) + c_1 h_1 \left[\mathscr{y}_i^{Pbst}(t) - \mathscr{y}_i(t)\right] + \ c_2 h_2 \left[\mathscr{y}_i^{Gbst}(t) - \mathscr{y}_i(t)\right] \\ \mathscr{y}_i(t+1) = \mathscr{y}_i(t) + \vec{\mho}_{\mathscr{y}}^i(t+1) \end{cases} \quad (28)$$

Where, $(\chi_i, \mathscr{y}_i)$ and $\vec{\mho}^i : \left(\vec{\mho}_\chi^i, \vec{\mho}_{\mathscr{y}}^i\right)$ are particle position and velocity components at iteration t , respectively; c_1 and c_2 are acceleration coefficients; $\langle \chi^{Pbst}, \mathscr{y}^{Pbst} \rangle$ and $\langle \chi^{Gbst}, \mathscr{y}^{Gbst} \rangle$ are the personal and global best positions, respectively. h_1, $h_2 \in [0, 1]$ are two independent random numbers; and $\mathcal{W}$ is the inertia weight and balances the algorithm between the local and global search. In each epoch, the current state value of the particle is compared with $\langle \chi^{Pbst}, \mathscr{y}^{Pbst} \rangle$ and $\langle \chi^{Gbst}, \mathscr{y}^{Gbst} \rangle$. The process of updating the particles' position and velocity is illustrated in Figure 6.

Each particle in the swarm corresponds to a *Spline* control point of $\vec{\vartheta}^i : (\vec{\vartheta}_x^i, \vec{\vartheta}_y^i)$ along the potential path $\mathscr{p}$. As the algorithm iterates, every particle moves toward its local best according to the outcome of the particle's individual and swarm's search. The mechanism of PSO algorithm for solving the path planning problem is illustrated through Algorithm-1. Figure 7 summarizes the concept of path planning procedure of multiple USVs in this study.



$\begin{bmatrix}\chi^{Gbst}\\ \mathscr{y}^{Gbst}\end{bmatrix}$
$\begin{bmatrix}\chi^{Pbst}\\ \mathscr{y}^{Pbst}\end{bmatrix}$
$\begin{bmatrix}\chi_i(t+1)\\ \mathscr{y}_i(t+1)\end{bmatrix}$
$\vec{\mho}^i(t+1)$
$c_2 h_2 \begin{bmatrix}\chi_i^{Gbst}(t) - \chi_i(t)\\ \mathscr{y}_i^{Gbst}(t) - \mathscr{y}_i(t)\end{bmatrix}$
$c_1 h_1 \times \begin{bmatrix}\chi_i^{Pbst}(t) - \chi_i(t)\\ \mathscr{y}_i^{Pbst}(t) - \mathscr{y}_i(t)\end{bmatrix}$
$\begin{bmatrix}\chi_i(t)\\ \mathscr{y}_i(t)\end{bmatrix}$
$\mathcal{W} \times \vec{\mho}^i(t)$
$\vec{\mho}^i(t)$

Fig.6 Mechanism of updating the particles' position and velocity.

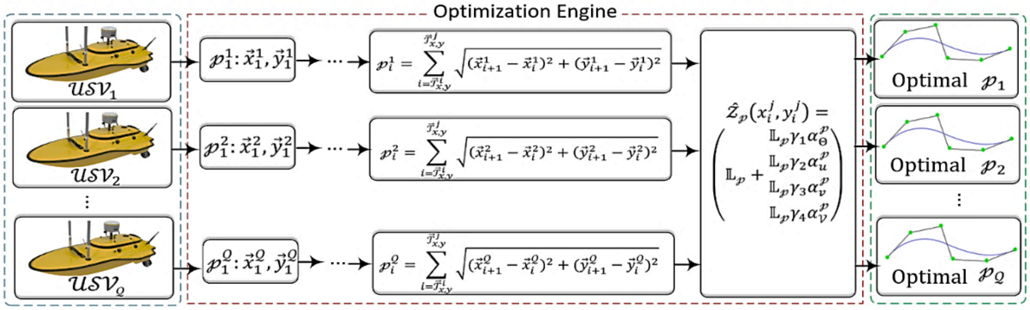

Optimization Engine
$\mathcal{USV}_1$
$\mathcal{USV}_2$
$\mathcal{USV}_Q$
$p_1^1 : \vec{x}_1^1, \vec{y}_1^1$
$p_1^2 : \vec{x}_1^2, \vec{y}_1^2$
$p_1^Q : \vec{x}_1^Q, \vec{y}_1^Q$
$p_i^1 = \sum_{i=\mathcal{T}_{x,y}^i}^{\mathcal{T}_{x,y}^j} \sqrt{(\vec{x}_{i+1}^1 - \vec{x}_i^1)^2 + (\vec{y}_{i+1}^1 - \vec{y}_i^1)^2}$
$p_i^2 = \sum_{i=\mathcal{T}_{x,y}^i}^{\mathcal{T}_{x,y}^j} \sqrt{(\vec{x}_{i+1}^2 - \vec{x}_i^2)^2 + (\vec{y}_{i+1}^2 - \vec{y}_i^2)^2}$
$p_i^Q = \sum_{i=\mathcal{T}_{x,y}^i}^{\mathcal{T}_{x,y}^j} \sqrt{(\vec{x}_{i+1}^Q - \vec{x}_i^Q)^2 + (\vec{y}_{i+1}^Q - \vec{y}_i^Q)^2}$
$\hat{\mathcal{Z}}_p(x_i^j, y_i^j) = \left(\mathbb{L}_p + \begin{matrix}\mathbb{L}_p\gamma_1\alpha_\Theta^p \\ \mathbb{L}_p\gamma_2\alpha_u^p \\ \mathbb{L}_p\gamma_3\alpha_v^p \\ \mathbb{L}_p\gamma_4\alpha_v^p\end{matrix}\right)$
Optimal p_1
Optimal p_2
Optimal p_Q

Fig.7 Concept of USVs path planning structure.

Algorithm (1) – Pseudocode of PSO algorithm for multi-USV path planning

Input:

Population size (p_{max}), Maximum iteration (t_{max}), Maximum available time ($\mathcal{T}_{\text{end}}^j$) for vehicle $\mathcal{USV}_j$, Number of USVs ($\mathcal{Q}$), Number of sampling tasks in a route ($\mathcal{T}_n$), Number of control-points to generate Spline (m)

```
//Initialization
    define $tasks: \langle \mathcal{T}_{i,xyz} \rangle; i \in \{1, \dots, n\}$        //Define "n" tasks in a 2D environment using (4)
    $i = \{1,2, \dots, p_{max}\}$        //Define population index
    $\mathcal{USV}_j,\ j = \{1,2, \dots, \mathcal{Q}\}$        //Define vehicles' index
    $t = \{1,2, \dots, t_{max}\}$        //Define iteration index
// PSO main loop
    For $t = 1$ to $t_{max}$        //for a determined number of iterations
        For $i = 1$ to $p_{max}$        //for each population member
            $ind(i) = \text{cluster}(tasks, \mathcal{Q})$        // cluster the tasks into "$\mathcal{Q}$" groups
            For $j \in \mathcal{USV}_\mathcal{Q}$        // for each vehicle like $\mathcal{USV}_j$
                Get $(ind(i,j))$        // get the cluster assigned to $\mathcal{USV}_j$
                $(\chi_i^j, y_i^j) \coloneqq \langle \vartheta_x^{i,j}, \vartheta_y^{i,j} \rangle$        //assign Spline control-points $\vartheta^i: (\vartheta_x^i, \vartheta_y^i)$ with particle position $(\chi_i, y_i)$
                $\langle \chi^{Pbst}(1), y^{Pbst}(1) \rangle = (\chi_i^j, y_i^j)$at $t = 1$        // initialize particle's personal and global best positions with the
                $\langle \chi^{Gbst}(1), y^{Gbst}(1) \rangle = (\chi_i^j, y_i^j)$at $t = 1$        current position of the particle at first iteration $t = 1$
                $\vec{\mho}_\chi^{i,j} = \mathcal{L}_{\vartheta_x}^{i,j} + Rand_x^{i,j} \left( \mathcal{U}_{\vartheta_x}^{i,j} - \mathcal{L}_{\vartheta_x}^{i,j} \right)$        // initialize each particle with random velocity $\vec{\mho}_{\chi y}^{i,j}$ in the range of
                $\vec{\mho}_y^{i,j} = \mathcal{L}_{\vartheta_y}^{i,j} + Rand_y^{i,j} \left( \mathcal{U}_{\vartheta_y}^{i,j} - \mathcal{L}_{\vartheta_y}^{i,j} \right)$        $\mathfrak{B}_\vartheta^i = [\mathcal{L}_\vartheta^i, \mathcal{U}_\vartheta^i]$
            end For
            $\mathcal{T}(i,j) = ind(i,j)$        // initialize population of the task clusters for each $\mathcal{USV}_j$
            $\hat{\mathcal{Z}}_p(\chi_i^j(t), y_i^j(t))$        // evaluate each particle according to the given cost function
            Update $\langle \chi_i^{Pbst}(t), y_i^{Pbst}(t) \rangle$        // Update the particles personal best position at iteration $t$
            Update $\langle \chi_i^{Gbst}(t), y_i^{Gbst}(t) \rangle$        // Update the particles global best position at iteration $t$
             if $\hat{\mathcal{Z}}_p(\chi_i(t), y_i(t)) \le \hat{\mathcal{Z}}_p(\chi_i^{Pbst}(t-1), y_i^{Pbst}(t-1))$
                $\begin{bmatrix} \chi_i^{Pbst}(t) \\ y_i^{Pbst}(t) \end{bmatrix} = \begin{bmatrix} \chi_i(t) \\ y_i(t) \end{bmatrix}$
            else
                $\begin{bmatrix} \chi_i^{Pbst}(t) \\ y_i^{Pbst}(t) \end{bmatrix} = \begin{bmatrix} \chi_i^{Pbst}(t-1) \\ y_i^{Pbst}(t-1) \end{bmatrix}$
            end (if)
            $\begin{bmatrix} \chi_i^{Gbst}(t) \\ y_i^{Gbst}(t) \end{bmatrix} = \underset{1 \le i}{\text{argmin}}\, \hat{\mathcal{Z}}_p(\chi_i^{Pbst}(t), y_i^{Pbst}(t))$        // Update the state of the particle in the swarm
            $\begin{cases} \vec{\mho}_\chi^i(t+1) = \mathcal{W}\vec{\mho}_\chi^i(t+1) + c_1 h_1[\chi_i^{Pbst}(t) - \chi_i(t)] + c_2 h_2[\chi_i^{Gbst}(t) - \chi_i(t)] \\ \vec{\mho}_y^i(t+1) = \mathcal{W}\vec{\mho}_y^i(t+1) + c_1 h_1[y_i^{Pbst}(t) - y_i(t)] + c_2 h_2[y_i^{Gbst}(t) - y_i(t)] \end{cases}$
            $\begin{cases} \chi_i(t+1) = \chi_i(t) + \vec{\mho}_\chi^i(t+1) \\ y_i(t+1) = y_i(t) + \vec{\mho}_y^i(t+1) \end{cases}$
            $\hat{\mathcal{Z}}_p(\chi_i(t+1), y_i(t+1))$
            $\begin{bmatrix} \chi_i^{Gbst}(t) \\ y_i^{Gbst}(t) \end{bmatrix}$        // transfer the best particles to next generation
        end (For)
    end (For)
Output $\langle \chi^{Gbst}, y^{Gbst} \rangle$
```

Output: The optimum path p for each $\mathcal{USV}$

4.2 Search Space Decomposition

To achieve faster convergence of the optimization routine, the initial local spline sets are generated around the local area between every two consecutive tasks, instead of the whole map area. Therefore, the entire operation field is decomposed to smaller areas and bounded to pairs of consecutive tasks with a certain expansion interval. This represents by coloured squares for every set of local splines as the restricted searching area shown in Figure 8.

As indicated in Figure 8, the square areas are considered as the initial search space for local spline sets. It should be highlighted that the determined areas are only considered to create the initial population, and local spline points can go beyond their local space, if the algorithm's update vector decides to move them out in next generations. The local space limits of a typical pair of task points $\mathcal{T}_{x,y}^{i}$ and $\mathcal{T}_{x,y}^{i+1}$ are defined by (29):

$$\begin{aligned} x_{min} &= \min\left(\mathcal{T}_x^i, \mathcal{T}_x^{i+1}\right) - b \\ x_{max} &= \max\left(\mathcal{T}_x^i, \mathcal{T}_x^{i+1}\right) + b \\ y_{min} &= \min\left(\mathcal{T}_y^i, \mathcal{T}_y^{i+1}\right) - b \\ y_{max} &= \max\left(\mathcal{T}_y^i, \mathcal{T}_y^{i+1}\right) + b \end{aligned} \qquad (29)$$

where x_{min}, x_{max}, y_{min} and y_{max} are lower and upper limits of horizontal and vertical components of 2D span enclosed between two tasks of $\mathcal{T}_{x,y}^{i}$ and $\mathcal{T}_{x,y}^{i+1}$. The size of the square is expandable by adjusting the constant expansion bound of b. Therefore, the size of the local search space is obtained by (30):

$$\begin{aligned} \ell_h &= \left|\mathcal{T}_x^{i+1} - \mathcal{T}_x^i\right| + 2b \\ \ell_v &= \left|\mathcal{T}_y^{i+1} - \mathcal{T}_y^i\right| + 2b \end{aligned} \qquad (30)$$

where ℓ_h and ℓ_v are the length and width of the confined of horizontal and vertical axis, respectively.

5 Simulation Results and Discussion

The multi-USVs path planner in this research uses the probabilistic map of the operation field (Banda Sea, see Figures.1, 2), mission time, and battery capacity of each vehicle to visit and collect the data of sampling spots safely and without interruption. The USVs, furthermore, use their navigation aids such as GPS, radar, sonar, lidar, camera, and their situational awareness modules for better perception of the environment and the assigned task spots. The navigation system can also contribute to percept and detect any moving obstacle such as ships, floating rocks, and whales and the proposed information are fed into the path planning module for generating collision-free paths. The share and exchange of information among the vehicles are possible via the USVs' mounted wireless communication system. The following initialization and setting were used for the proposed path planning system: all computations of this research were performed on a desktop PC with an Intel i7 3.20 GHz quad-core processor in MATLAB®2019a. The PSO algorithm is configured with 80 particles and the maximum iterations of 250. The expansion-contraction coefficients are fixed on c_1=2.2 and c_2=2.6. In addition, the random coefficients of h_1 and h_2 are set at 0.6 and 0.4, respectively. Finally, the inertia weight is configured with $\mathcal{W} = 1.8$. In the following sections, the performance of the multi-USVs motion planner is investigated under different operating conditions.

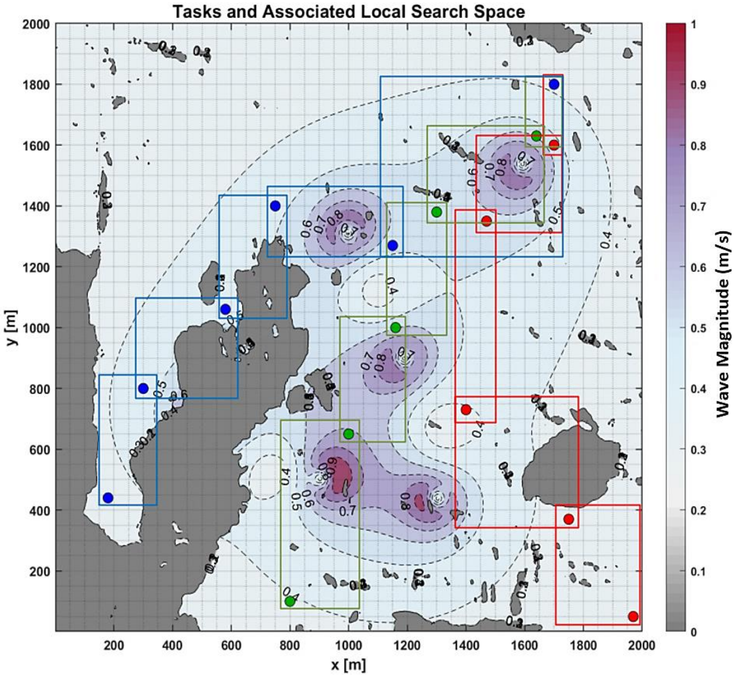


Fig.8 Local search spaces assigned to every two consecutive tasks in the contour

5.1 Conflict- Free Path Planning

Figure 9 shows the behavior of the USVs' path planner in the modelled cluttered operating field where the darkest areas on the map are non-traversable areas, and purple spots correspond to the deepest areas encapsulating ocean turbulences. The map used in this study comprises a grid of 2000×2000 with the resolution of one square meter. The ocean wave components are computed from a random distribution of 3 to 6 Lamb vortices in the grid within a 2D spatial domain according to the size of the captured map. In the given map, the color bar represents the magnitude of ocean wave and its intensity increases in the purple areas. The coastal areas are forbidden zones for operation and get the value of 0 in the probability distribution matrix, while the values over the 0.8 in this matrix corresponds to the strong water turbulence that also should be avoided.

As shown in in Figure 9 (a), the path planner can accurately recognize the coastal areas and avoid colliding forbidden edges. It is notable from Figure 9 (a) that all the generated paths accurately avoid the higher intensity ocean waves and are robust to the growing complexity of the terrain as all three paths accurately guide the vehicles through the task spots and avoid crossing forbidden margins.

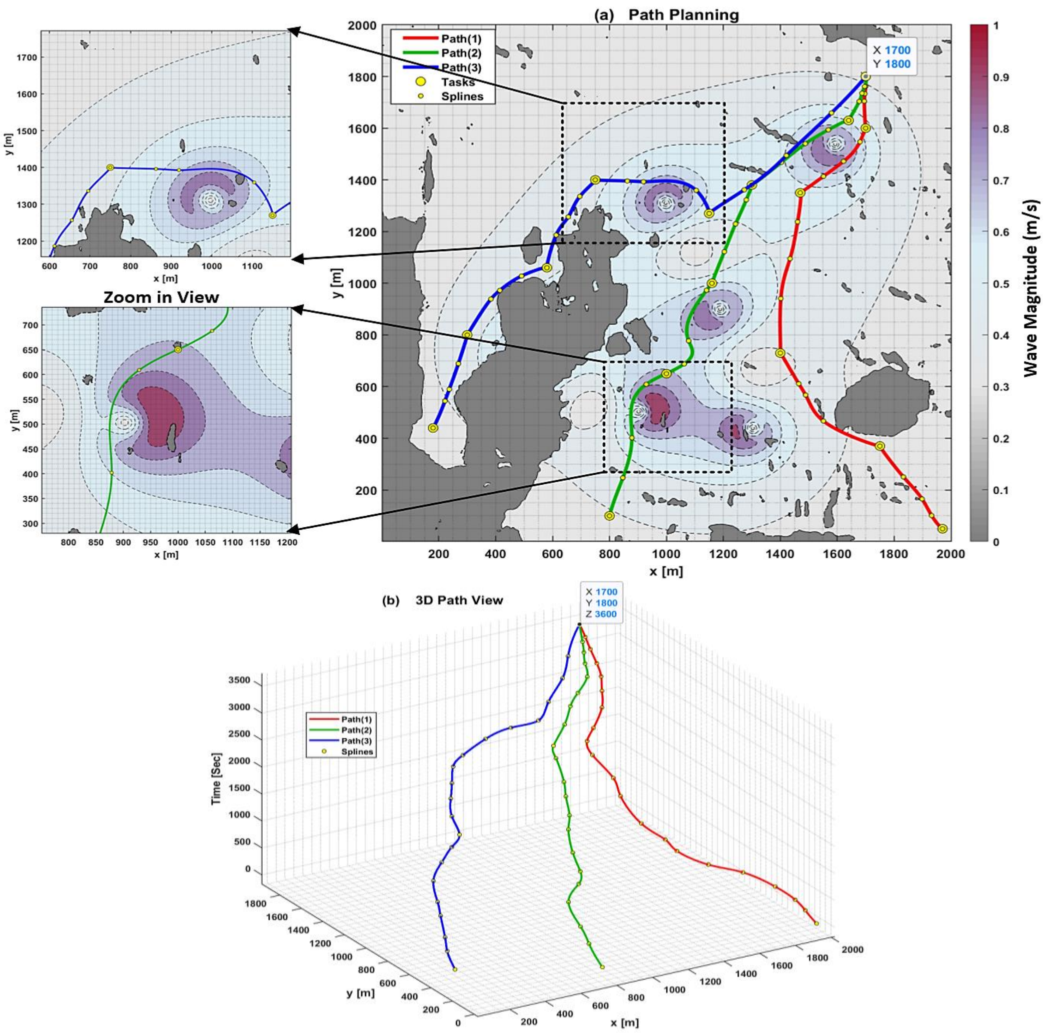


Fig.9 Performance of the multi-USVs path planning with respect to the safety margins; (a) 2D plane; (b) 3D view of the generated path curves.

For better demonstration of non-conflict motions of three USVs, the 3-dimensional view of the paths is presented in Figure 9 (b), where the vertical axis corresponds to the mission time. In Figure 9 (b), the time variable is considered as the third dimension to make sure that the vehicles do not catch a same position at a same sample time. This is pivotal to avoid internal collision or mission overlap among the vehicles. Figure 9(b) clearly shows that there is no collision or task conflict between the vehicles in parallel mission execution.

5.2 Controllable USVs' Maneuver in Water Sampling Mission

In this subsection, the performance of the proposed uninterrupted path planning system is elaborated. To this end, first a piecewise path planning system developed in [48] is utilized to highlight the difficulties of uninterrupted water sampling task accomplishment along the path profile. In the piecewise method, the path curves between every two consecutive spots are simultaneously generated for each vehicle, and this process repeatedly continues for every pair of sampling spot along the mission until the vehicles meet the rendezvous point. In the piecewise method, the vehicles fully stop for $t^{\mathcal{T}}$ seconds when they reach the sampling spots (in this scenario, each sampling spot ($\mathcal{T}^i$) requires $t^{\mathcal{T}^i} = [60,70]$ sec to be completed). Once the sampling task is completed, the vehicles accelerate from zero velocity and abruptly change the heading angle toward the next sampling spot. This behavior implies a large jump and consequently singular arch on the velocity, heading, and heading rate profiles and applies serious difficulties for the optimization engine to solve the problem numerically [49-52]. In addition, the controller cannot track such a spiky trajectory and hence the divergence in the USVs maneuverability occurs.

To accommodate the issue of the piecewise path planning system, the proposed B-Spline data frame (introduced in Section 3.3) is utilized to provide uninterrupted path planning as well as smooth and tractable USV's state profiles. Figure 10 shows the differences between performance of the piecewise and uninterrupted path planning in contour map. Applying the developed B-Spline data frame gives the path extra smoothness; the bended curve for completing the sampling task that is shown in Figure 10 (b).

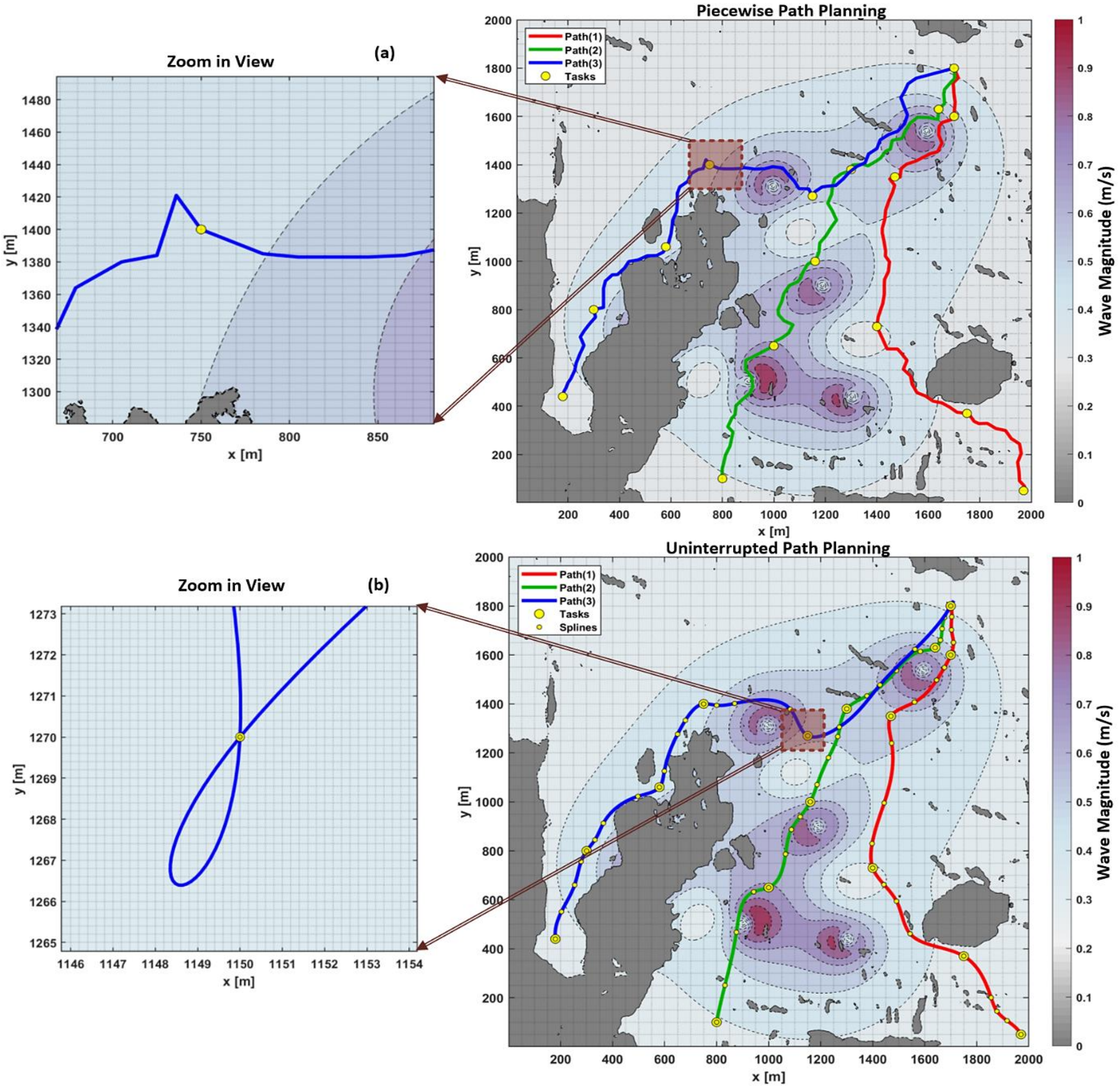


Fig.10 Comparison of piecewise and uninterrupted path planning in the contour map: (a) piecewise path planning; (b) the uninterrupted path planning system.

The magnified area in Figure 10 (a) and (b) reveals how the proposed mechanism for uninterrupted path planning can smoothly turn around the sampling area and adjust its trajectory toward the next sampling spot without taking any sharp turn. In the pricewise path planner, the path between every two consecutive tasks is generated by PSO using l (length of spline sets) spline points, and applying the same cost function given in continuous mode.

As depicted in zoom-in view of Figure 10 (a), the pricewise path planner takes a sharp turn at the junction of two paths to adjust its trajectory toward the next sampling spot. In contrast, the magnified path curves around the task points in Figure 10 (b) reveals that the vehicles have taken a circular shape around the location of the tasks. In other words, the path passes over the tasks and circles the nearby area very slowly with a very small radius, while the time for one perfect circle is equal to $t^{\mathcal{T}^j}$. This provides the time for the USVs to take sample from the marked spots. The rationale behind this process is that two consecutive spline points are located in each sampling spot, and the time needed for interpolating these two spline points is determined as $t^{\mathcal{T}^j}$. Therefore, passing over a sampling spot and returning to the same point takes $t^{\mathcal{T}^j}$ seconds to travel.

The difference between performances of these two strategies is even more evident in the velocity profiles of the USVs obtained via piecewise and uninterrupted path planning systems as indicated in Figure 11 (a) and (b), respectively. As can be seen in the Figure 11(a), the vehicles' resultant velocity takes zero value for $t^{\mathcal{T}} = 60$ seconds in each task position, meaning that the vehicles stop moving in the sampling spots. The sharp turns in the path curvatures impose impulsive patterns in the velocity profile. This artifact can cause singularity and difficulty in tracking the generated manoeuvre. In contrast, it is inferable from Figure 11 (b) that the velocity profiles in the uninterrupted path planning is smooth, bounded and tractable for USVs' controllers. As a result, the path and velocity profiles of the USVs can be controlled much more efficiently with minimum level of complexity.

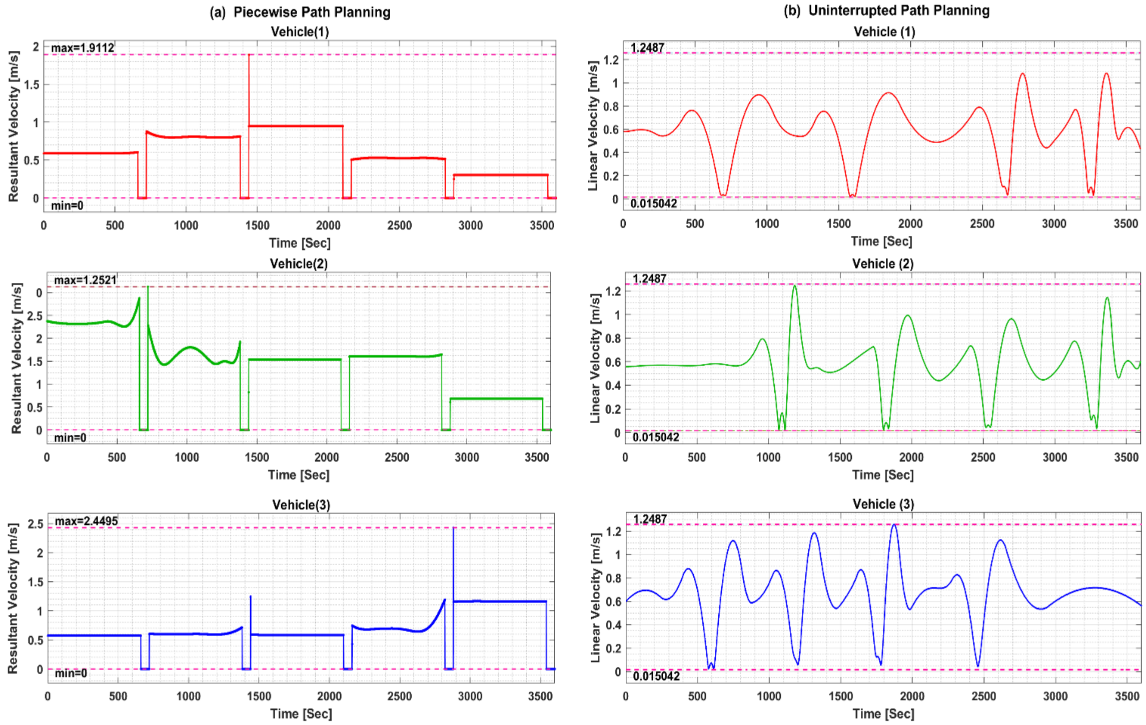


Fig.11 Velocity profile of three USVs resulted from: (a) piecewise path planning; (b) uninterrupted path planning.

Figure 12 elaborates the quality of path planning in the case that USV stops to perform the task, and accomplish the task using the uninterrupted mechanism. As shown in Figure 12 (a), the USV's full stop in the sampling spot can result in a sharp angle in the turning point of the path between the USV's heading and the next stop when the vehicle deviates to move toward next sampling spot. Moreover, a path with sharp angles causes sharp jumps at the control inputs and velocity overshoot, which should be avoided. The mechanism designed in this study enabling the USVs to follow an orbital path and avoiding sharp turns, while giving the vehicle enough time and maneuvering space to collect sampling data. In the uninterrupted path planning method (Figure 12 (b)), the vehicle is given sufficient time and space for taking sample. The vehicle simultaneously and smoothly adjusts the heading direction toward the next destination, while the vehicle is circling around the area and completes the sampling task.

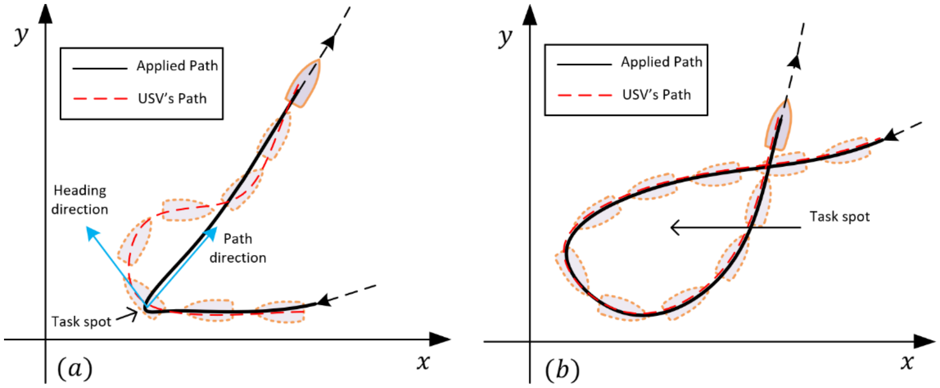


Fig.12 Comparison of piecewise and uninterrupted path planning incorporating the heading angle: (a) path quality piecewise mechanism; (b) path quality using uninterrupted path planning method.

Remark 2- The heading angle profile generated by the path planning system should incorporate the physical motion characteristics of the USV. In other words, sharp changes and non-smooth transitions in the heading/velocity motion profiles can cause the path infeasible. This is due to the fact that there exist physical constraints over the USV's actuators and thus the controller can just deliver a limited thrust and control torque. The consequence of this can increase the possibility of capsizing risk, course misleading, and object collision during the mission.

5.3 Comparative Study

To further evaluate the performance of the uninterrupted path planning system, the RRT algorithm is employed as a benchmark model [53]. The RRT is a waypoint-based algorithm that randomly generates space-filling trees and is particularly designed to explore multidimensional non-convex search spaces. In this method, the tree is formed incrementally and is inherently biased to expand towards unexplored areas of the problem.

Unlike the spline-based methods which provide continuous path curves for vehicles, the path provided by the RRT comprises a set of discrete points that a vehicle chooses to follow a subset of these points in a way to address the defined objective function. Figure 13 shows the pictorial concept of using RRT path planning for the USVs in this study.

Figure 14 illustrates the performance of RRT and the proposed uninterrupted path planning systems for USVs' water sampling mission. In this simulation, both methods use the same setting and configuration of the environment/vehicles.

As shown in Figure 14 (a), the path provided by RRT is fragmented and reveals sharp fluctuations to avoid colliding obstacles. It is also apparent from the given zoom-in view in Figure 14 (a), that to deal with obstacle avoidance, the RRT loses its accuracy in avoiding water turbulence which means the algorithm is not able to satisfy all existing vehicular and environmental constraints concurrently. In contrast, Figure 14 (b) shows excellent flexibility of the uninterrupted method in avoiding severe turbulences and obstacles' barriers. Considering the path deformations in Figure 14 (b), it is evident that refined paths perfectly adapt to the ocean wave force, by preventing the high-density regions (the dark red spots in the contour map) where the water force magnitude is severe.

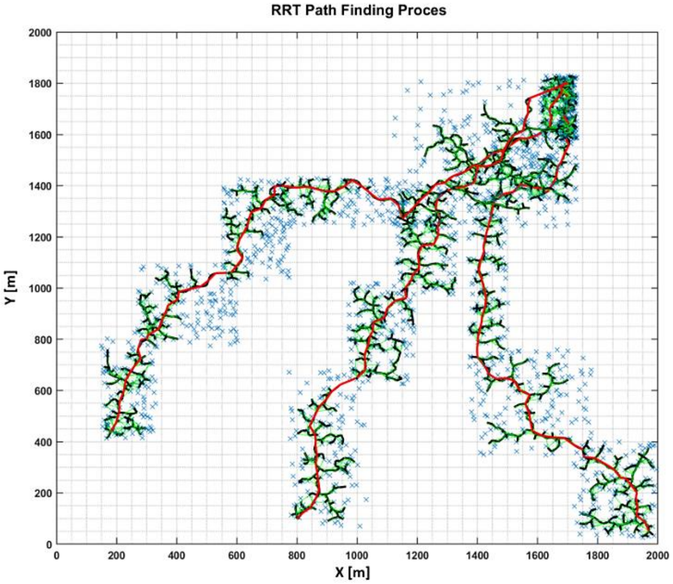


Fig.13 Mechanism of RRT path finding for three vehicles.

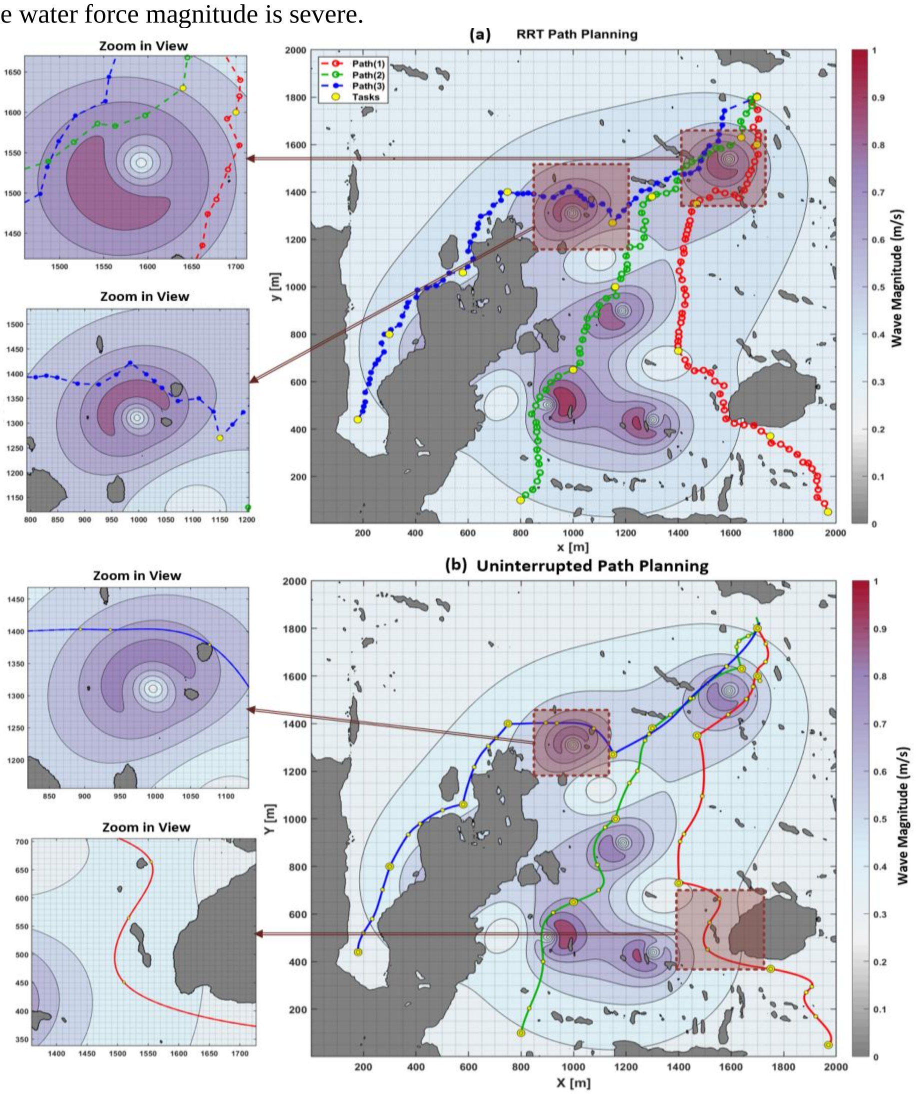


Fig.14 Performance of the path planning systems for navigating the multi-USVs with respect to the safety margins and environmental disturbances; (a) RRT algorithm (b) the uninterrupted method.

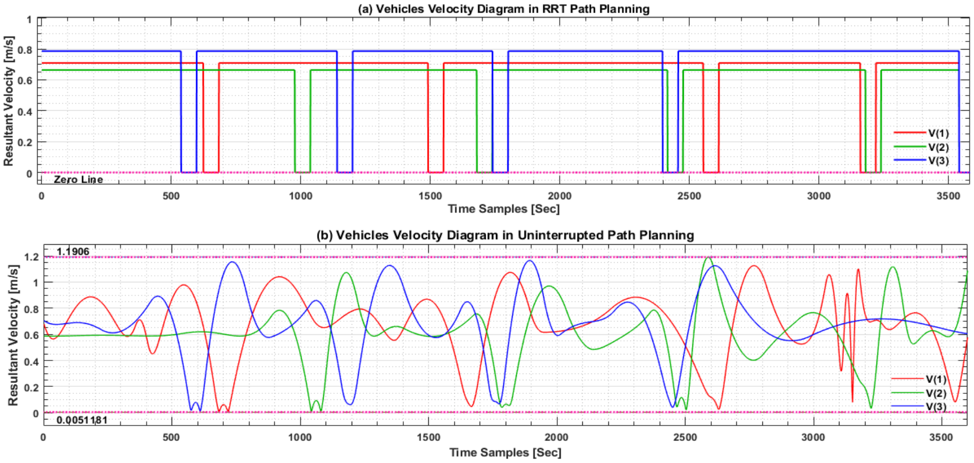
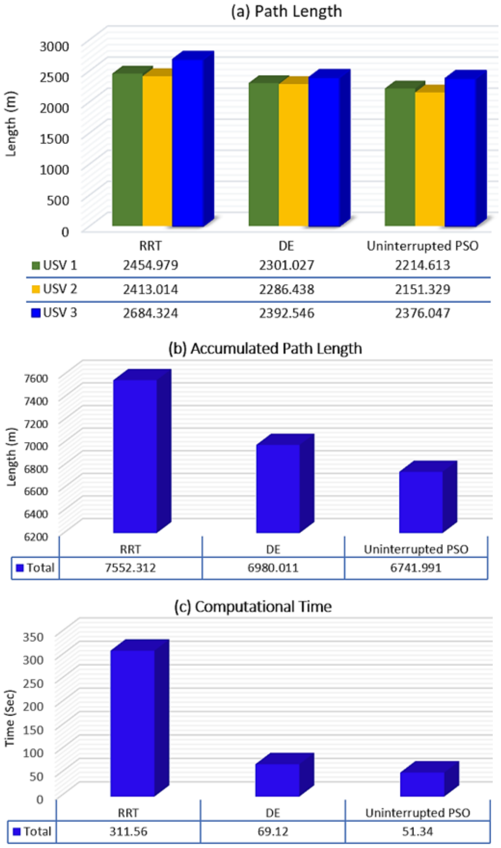


Fig.15 Velocity profiles of three USVs resulted from: (a) RRT algorithm; (b) uninterrupted path planning.

Figure 15 illustrates the difference between velocity profiles generated by the RRT and uninterrupted path planning methods. As shown in Figure 15 (a), the velocity profiles of the vehicles are perfectly flat, but since the vehicles should reach the rendezvous point at a same time, the amplitudes are different for each of vehicles depending on the length of the paths. Due to the RRT's discrete nature, the velocity of the vehicles experiences a sudden rise after a full stop in sampling spots. However, in practice, the sudden up and down in velocity profile is difficult to be tracked by the velocity controller and usually results in instability of the proposed solutions and remarkable trajectory error. By contrast, in continuous path planning, the generated trajectories are smooth, without any sharp turn or sudden fluctuation, which makes the generated paths easier to be tracked by vehicles' controller. It is notable from Figure 14 (b) and Figure 15 (b) that placement of local splines, and their corresponding time-samples, are established in the way that the velocity profiles hover within a reasonable range and never reach zero unless the vehicles meet rendezvous point.

On top of the qualitative performance comparison above, the preposed path planning system is numerically compared with the RRT as well as differential evolution (DE)-based path planning [46, 54] method. Figure 16 numerically compares the performance of the RRT, DE, and the proposed uninterrupted path planning based on several performance indices such as path length, and computational time. Figure 16 (a)- (c) compare the traveled path of each USV, total traveled path, and computational time required for path generation of the proposed uninterrupted method with the DE and RRT benchmark methods. As shown in Figure 16 (a) and (b), each USV travels about 11% less in terms of path length when they use the uninterrupted method as compared with the benchmark RRT solution. Comparing to RRT, the DE-based path planning method shows better performance and provide closer solutions to the proposed uninterrupted solution. However, the travelled path length achieved by DE is still about 4% more than the proposed uninterrupted method.



	RRT	DE	Uninterrupted PSO
🟩 USV 1	2454.979	2301.027	2214.613
🟨 USV 2	2413.014	2286.438	2151.329
🟦 USV 3	2684.324	2392.546	2376.047

	RRT	DE	Uninterrupted PSO
🟦 Total	7552.312	6980.011	6741.991

	RRT	DE	Uninterrupted PSO
🟦 Total	311.56	69.12	51.34

Fig.16 Numerical performance evaluation of the RRT, DE and PSO-based uninterrupted path planning methods.

In addition, the computational efficiency of the proposed uninterrupted method is about 6 times better than the RRT counterpart. The rationale behind this is that the RRT is a deterministic sampling-based motion planning algorithm and explores all the possible pathways in the search space to find the best point that minimizes the distance to the rendezvous station. This search mechanism results in longer computation time and greater path length compared to results provided by PSO-based uninterrupted algorithm which does have a non-deterministic nature. Although the computational efficiency of the DE-based path planning is far better than the RRT and closer to the proposed uninterrupted method (due to its non-deterministic nature), the difference of 25% confirms the computational superiority of the proposed uninterrupted method against the DE method as well.

5.4 Computational Time Complexity Analysis

Analysis of the algorithm complexity is of importance as for any optimization technique, the algorithm convergence rate to the feasible solutions will become slower with an increase in the number of decision variables and system size. In particular, for realistic missions such as what investigated in this study, the analysis of the path planning system complexity can provide an insight to what level the solutions achieved are compatible with computational and resource limitations of the real-time performance.

In order to measure the computational performance and stability of the proposed path planning system, the time computational complexity is investigated w.r.t the variations in the number of sampling spots; that is the distribution of sampling spots (network complexity) and size increase incrementally from 16 to 100. This is further investigated for cases that number of USVs in the team changes from 3 to 5 vehicles. The results of this investigation are summarized in Fig.17.

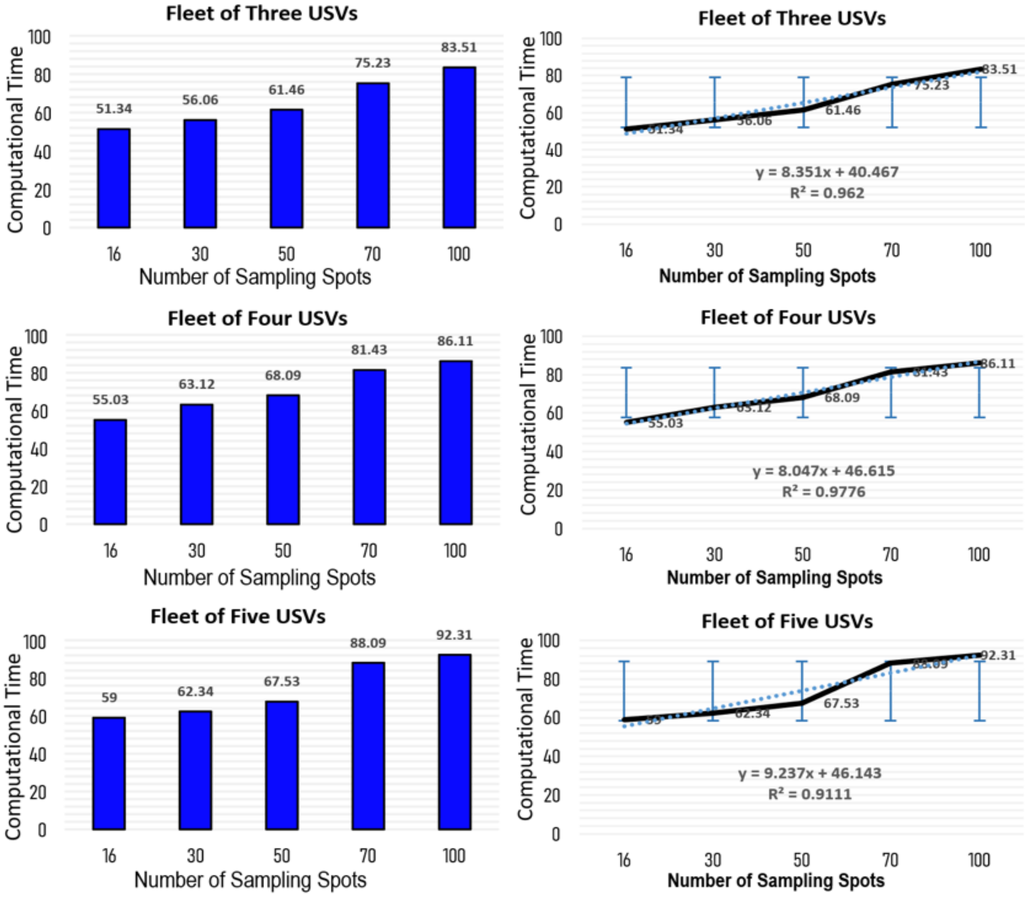


Fig.17 System performance and computational complexity of the proposed path planning system against the variations of number of sampling spots.

As shown in Fig.17, the algorithm accurately manages the complexity growth as the computational time increases linearly with the rise in the number of sampling spots. It is noted that exploiting more vehicles in the fleet has minimal impact on computation time (settled in the range of seconds for all experiments). This efficiency allows the system to integrate more vehicles on demand to handle complex missions where this change does not impact the computation load of planning as the entire CPU time remains within the bound of a suitable real-time solution. This endorses the stability of system performance and computational effectiveness as the computation load is not exponentially depends on both size and complexity of the operation environment, whereas this is recognized as a challenging issue in many of the outlined studies.

6 Conclusion

In this paper, a novel uninterrupted path planning method that facilitates mission of a team of USVs in ocean data sampling was developed and its performance was investigated through extensive simulation studies. The proposed path planning system received the information of sampling spots in form of a series of ordered tasks, provided by the HFC algorithm, and then used a new B-spline data frame, PSO engine, and decomposed optimization search space to generate shortest, constraint-aware, and collision-free path curves for USVs to perform the mission in the complex modeled maritime environment. The proposed path planning system enabled USVs to conduct the sampling task without interruption and prevent frequent stop/start switching of the velocity profile that was to encircle the task area smoothly while simultaneously correcting the heading angle toward the next task spot and preventing sharp changes in the vehicle's heading angle. As a result, the issues of singular arch on the USVs' states profiles were accommodated and more feasible and tractable states' profiles were provided for the vehicles' controller that enhanced the system stability and performance. The extensive simulation studies including comparative performance assessment of the proposed path planner against the benchmark piecewise, RTT, and DE planners demonstrated the effectiveness of and fidelity of the proposed method. Future extension of this study includes an integration of the uninterrupted planner system with an advanced control system such as fast and non-singular terminal sliding mode [55-57] and conducting field trials for evaluation of the planner in practice.

REFERENCES

[1] A. Cadena, "Development of a low-cost, unmanned surface vehicle for military applications," in *Ocean Sensing and Monitoring IV*, 2012, vol. 8372: International Society for Optics and Photonics, pp. 83721D.

[2] H. Cao, Z. Guo, S. Wang, H. Cheng, and C. Zhan, "Intelligent wide-area water quality monitoring and analysis system exploiting unmanned surface vehicles and ensemble learning," *Water,* vol. 12, no. 3, pp. 681, 2020.

[3] Y. Liu and R. Bucknall, "Efficient multi-task allocation and path planning for unmanned surface vehicle in support of ocean operations," *Neurocomputing,* vol. 275, pp. 1550-1566, 2018.

[4] F. Zhang, O. Ennasr, E. Litchman, and X. Tan, "Autonomous sampling of water columns using gliding robotic fish: Algorithms and harmful-algae-sampling experiments," *IEEE Systems Journal,* vol. 10, no. 3, pp. 1271-1281, 2015.

[5] A. Bennett *et al.*, "Autonomous vehicles for remote sample collection: Enabling marine research," in *OCEANS 2015-Genova*, 2015: IEEE, pp. 1-8.

[6] A. Vasilijević, Đ. Nađ, F. Mandić, N. Mišković, and Z. Vukić, "Coordinated navigation of surface and underwater marine robotic vehicles for ocean sampling and environmental monitoring," *IEEE/ASME Transactions on Mechatronics,* vol. 22, no. 3, pp. 1174-1184, 2017.

[7] F. P. Chavez, J. Sevadjian, C. Wahl, J. Friederich, and G. E. Friederich, "Measurements of pCO2 and pH from an autonomous surface vehicle in a coastal upwelling system," *Deep Sea Research Part II: Topical Studies in Oceanography,* vol. 151, pp. 137-146, 2018.

[8] J.Bae, S.Luo, S.Kannan, Y.Singh, B.Lee, R.Voyles, M.Malaga, E.Zenteno, L.Aguilar,and B.Min., "Development of an unmanned surface vehicle for remote sediment sampling with a van veen grab sampler," in *OCEANS 2019 MTS/IEEE SEATTLE*, 2019: IEEE, pp. 1-7.

[9] A. Atyabi, S. MahmoudZadeh, and S. Nefti-Meziani, "Current advancements on autonomous mission planning and management systems: An AUV and UAV perspective," *Annual Reviews in Control,* vol. 46, pp. 196-215, 2018.

[10] D. V. Lyridis, "An improved ant colony optimization algorithm for unmanned surface vehicle local path planning with multi-modality constraints," *Ocean Engineering,* vol. 241, pp. 109890, 2021.

[11] R. Song, Y. Liu, and R. Bucknall, "A multi-layered fast marching method for unmanned surface vehicle path planning in a time-variant maritime environment," *Ocean Engineering,* vol. 129, pp. 301-317, 2017.

[12] N. Wang, X. Jin, and M. J. Er, "A multilayer path planner for a USV under complex marine environments," *Ocean Engineering,* vol. 184, pp. 1-10, 2019.

[13] Y. Singh, S. Sharma, R. Sutton, D. Hatton, and A. Khan, "Feasibility study of a constrained Dijkstra approach for optimal path planning of an unmanned surface vehicle in a dynamic maritime environment," in *2018 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, 2018: IEEE, pp. 117-122.

[14] R. Song, Y. Liu, and R. Bucknall, "Smoothed A* algorithm for practical unmanned surface vehicle path planning," *Applied Ocean Research*, vol. 83, pp. 9-20, 2019.

[15] H. Sang, Y. You, X. Sun, Y. Zhou, and F. Liu, "The hybrid path planning algorithm based on improved A* and artificial potential field for unmanned surface vehicle formations," *Ocean Engineering,* vol. 223, pp. 108709, 2021.

[16] J. Song, C. Hao, and J. Su, "Path planning for unmanned surface vehicle based on predictive artificial potential field," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, pp. 1729881420918461, 2020.

[17] Y. Liu and R. Bucknall, "The angle guidance path planning algorithms for unmanned surface vehicle formations by using the fast marching method," *Applied Ocean Research,* vol. 59, pp. 327-344, 2016.

[18] H. Niu, Z. Ji, A. Savvaris, and A. Tsourdos, "Energy efficient path planning for Unmanned Surface Vehicle in spatially-temporally variant environment," *Ocean Engineering,* vol. 196, pp. 106766, 2020.

[19] C. Zhou *et al.*, "Motion planning for an unmanned surface vehicle based on topological position maps," *Ocean Engineering,* vol. 198, pp. 106798, 2020.

[20] H. Guo, Z. Mao, W. Ding, and P. Liu, "Optimal search path planning for unmanned surface vehicle based on an improved genetic algorithm," *Computers & Electrical Engineering,* vol. 79, pp. 106467, 2019.

[21] X. Guo, M. Ji, Z. Zhao, D. Wen, and W. Zhang, "Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm," *Ocean Engineering,* vol. 216, pp. 107693, 2020.

[22] J. Xin, J. Zhong, F. Yang, Y. Cui, and J. Sheng, "An improved genetic algorithm for path-planning of unmanned surface vehicle," *Sensors,* vol. 19, no. 11, pp. 2640, 2019.

[23] J. Zhong, B. Li, S. Li, F. Yang, P. Li, and Y. Cui, "Particle swarm optimization with orientation angle-based grouping for practical unmanned surface vehicle path planning," *Applied Ocean Research,* vol. 111, pp. 102658, 2021.

[24] G. Zhang, Y. Deng, W. Zhang, and C. Huang, "Novel DVS guidance and path-following control for underactuated ships in presence of multiple static and moving obstacles," *Ocean Engineering,* vol. 170, pp. 100-110, 2018.

[25] G. Zhang, C. Zhang, X. Zhang, and Y. Deng, "ESO-based path following control for underactuated vehicles with the safety prediction obstacle avoidance mechanism," *Ocean Engineering,* vol. 188, pp. 106259, 2019.

[26] Y. Ma, M. Hu, and X. Yan, "Multi-objective path planning for unmanned surface vehicle with currents effects," *ISA transactions,* vol. 75, pp. 137-156, 2018.

[27] H. Kim, T. Lee, H. Chung, N. Son, and H. Myung, "Any-angle path planning with limit-cycle circle set for marine surface vehicle," in *2012 IEEE International Conference on Robotics and Automation*, 2012: IEEE, pp. 2275-2280.

[28] H. Kim, D. Kim, J.-U. Shin, H. Kim, and H. Myung, "Angular rate-constrained path planning algorithm for unmanned surface vehicles," *Ocean Engineering,* vol. 84, pp. 37-44, 2014.

[29] S. Campbell, W. Naeem, and G. W. Irwin, "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance manoeuvres," *Annual Reviews in Control,* vol. 36, no. 2, pp. 267-283, 2012.

[30] G. Tan, J. Zhuang, J. Zou, and L. Wan, "Coordination control for multiple unmanned surface vehicles using hybrid behavior-based method," *Ocean Engineering,* vol. 232, pp. 109147, 2021.

[31] X. Zhou, P. Wu, H. Zhang, W. Guo, and Y. Liu, "Learn to navigate: cooperative path planning for unmanned surface vehicles using deep reinforcement learning," *IEEE Access,* vol. 7, pp. 165262-165278, 2019.

[32] Y. Singh, M. Bibuli, E. Zereik, S. Sharma, A. Khan, and R. Sutton, "A novel double layered hybrid multi-robot framework for guidance and navigation of unmanned surface vehicles in a practical maritime environment," *Journal of Marine science and Engineering,* vol. 8, no. 9, pp. 624, 2020.

[33] L. Li, D. Wu, Y. Huang, and Z.-M. Yuan, "A path planning strategy unified with a COLREGS collision avoidance function based on deep reinforcement learning and artificial potential field," *Applied Ocean Research,* vol. 113, p. 102759, 2021.

[34] J. Woo and N. Kim, "Collision avoidance for an unmanned surface vehicle using deep reinforcement learning," *Ocean Engineering,* vol. 199, pp. 107001, 2020.

[35] M. Hinostroza, H. Xu, and C. G. Soares, "Cooperative operation of autonomous surface vehicles for maintaining formation in complex marine environment," *Ocean Engineering,* vol. 183, pp. 132-154, 2019.

[36] Y. Liu, R. Song, R. Bucknall, and X. Zhang, "Intelligent multi-task allocation and planning for multiple unmanned surface vehicles (USVs) using self-organising maps and fast marching method," *Information Sciences,* vol. 496, pp. 180-197, 2019.

[37] X. Liang, X. Qu, N. Wang, Y. Li, and R. Zhang, "Swarm control with collision avoidance for multiple underactuated surface vehicles," *Ocean Engineering,* vol. 191, pp. 106516, 2019.

[38] M. Bibuli, Y. Singh, S. Sharma, R. Sutton, D. Hatton, and A. Khan, "A two layered optimal approach towards cooperative motion planning of unmanned surface vehicles in a constrained maritime environment," *IFAC-PapersOnLine,* vol. 51, no. 29, pp. 378-383, 2018.

[39] A. Abbasi, S. MahmoudZadeh, and A.M. Yazdani, "A Cooperative Dynamic Task Assignment Framework for COTSBot AUVs," *IEEE Transactions on Automation Science and Engineering,*vol.19, no,2, pp. 1163-1179, 2022.

[40] S. MahmoudZadeh, D. Powers, and K. Sammut, "An autonomous reactive architecture for efficient AUV mission time management in realistic dynamic ocean environment," *Robotics and Autonomous Systems,* vol. 87, pp. 81-103, 2017.

[41] S. MahmoudZadeh, D. M. Powers, K. Sammut, A. M. Yazdani, and A. Atyabi, "Hybrid motion planning task allocation model for AUV's safe maneuvering in a realistic ocean environment," *Journal of Intelligent & Robotic Systems,* vol. 94, no. 1, pp. 265-282, 2019.

[42] S. MahmoudZadeh, D. W. Powers, A. M. Yazdani, K. Sammut, and A. Atyabi, "Efficient AUV path planning in time-variant underwater environment using differential evolution algorithm," *Journal of Marine Science and Application,* vol. 17, no. 4, pp. 585-591, 2018.

[43] S. MahmoudZadeh, "Efficient deployment and mission timing of autonomous underwater vehicles in large-scale operations," in *International Conference on Advanced Data Mining and Applications*, 2019: Springer, pp. 792-804.

[44] T. I. Fossen, "Marine control systems–guidance. navigation, and control of ships, rigs and underwater vehicles," *Marine Cybernetics, Trondheim, Norway, Org. Number NO 985 195 005 MVA, www. marinecybernetics. com, ISBN: 82 92356 00 2,* 2002.

[45] Z. Liu, Y. Zhang, X. Yu, and C. Yuan, "Unmanned surface vehicles: An overview of developments and challenges," *Annual Reviews in Control,* vol. 41, pp. 71-93, 2016.

[46] S. MahmoudZadeh, A. M. Yazdani, K. Sammut, and D. M. Powers, "Online path planning for AUV rendezvous in dynamic cluttered undersea environment using evolutionary algorithms," *Applied Soft Computing,* vol. 70, pp. 929-945, 2018.

[47] S. M. Zadeh, D. M. Powers, and R. B. Zadeh, "Autonomy and Unmanned Vehicles Augmented Reactive Mission-Motion Planning Architecture for Autonomous Vehicles," *CoRR,* 2020.

[48] S. Mahmoudzadeh, D. M. Powers, and A. Atyabi, "UUV's hierarchical DE-based motion planning in a semi dynamic underwater wireless sensor network," *IEEE Transactions on cybernetics,* vol. 49, no. 8, pp. 2992-3005, 2018.

[49] A.M. Yazdani, K. Sammut, O. Yakimenko, A. Lammas, S. MahmoudZadeh, and Y. Tang, "Time and energy efficient trajectory generator for autonomous underwater vehicle docking operations," in *OCEANS 2016 MTS/IEEE Monterey*, 2016: IEEE, pp. 1-7.

[50] A. M. Yazdani, K. Sammut, O. Yakimenko, and A. Lammas, "A survey of underwater docking guidance systems," *Robotics and Autonomous Systems,* vol. 124, pp. 103382, 2020.

[51] A. M. Yazdani, K. Sammut, O. A. Yakimenko, and A. Lammas, "Feasibility analysis of using the hp-adaptive Radau pseudospectral method for minimum-effort collision-free docking operations of AUV," *Robotics and Autonomous Systems,* vol. 133, pp. 103641, 2020.

[52] A. M. Yazdani, K. Sammut, O. A. Yakimenko, A. Lammas, Y. Tang, and S. M. Zadeh, "IDVD-based trajectory generator for autonomous underwater docking operations," *Robotics and Autonomous Systems,* vol. 92, pp. 12-29, 2017.

[53] A. Kaur and M. S. Prasad, "Path planning of multiple unmanned aerial vehicles based on RRT algorithm," in *Advances in Interdisciplinary Engineering*: Springer, 2019, pp. 725-732.

[54] S. MahmoudZadeh, D. M. Powers, K. Sammut, A. Atyabi, and A.M. Yazdani, "A hierarchal planning framework for AUV mission management in a spatiotemporal varying ocean," *Computers & Electrical Engineering,* vol. 67, pp. 741-760, 2018.

[55] Y. Hu, H. Wang, A.M. Yazdani, and Z. Man, "Adaptive full order sliding mode control for electronic throttle valve system with fixed time convergence using extreme learning machine," *Neural Computing and Applications,* pp. 1-13, 2021.

[56] M. Ye, H. Wang, A.M. Yazdani, S. He, Z. Ping, and W. Xu, "Discrete-time integral terminal sliding mode-based speed tracking control for a robotic fish," *Nonlinear Dynamics,* pp. 1-12, 2021.

[57] J. Zhang, H. Wang, M. Ma, M. Yu, A.M. Yazdani, and L. Chen, "Active front steering-based electronic stability control for steer-by-wire vehicles via terminal sliding mode and extreme learning machine," *IEEE Transactions on Vehicular Technology,* vol. 69, no. 12, pp. 14713-14726, 2020.