

Dynamics-Constrained Global-Local Hybrid Path Planning of an Autonomous Surface Vehicle

Ning Wang , Senior Member, IEEE, and Hongwei Xu 

Abstract—In this paper, under unforeseen circumstances, a dynamics-constrained global-local (DGL) hybrid path planning scheme incorporating global path planning and local hierarchical architecture is created for an autonomous surface vehicle (ASV) with constrained dynamics. By encapsulating ASV safety area into Theta*-like heuristics, global path planning algorithm is developed to optimally generate sparse waypoints which are sufficiently clear to constraints. To deal with dynamically unforeseen environments, a local hierarchy is established by fuzzy decision-making (FDM) and fine dynamic window (FDW) layers, which are responsible for large- and close-range collision avoidance, respectively, by governing surge and yaw velocity guidance signals. With the aid of the FDW, constrained dynamics pertaining to the ASV, i.e., actuatable surge/yaw velocities and accelerations, are elaboratively embedded into local path planning, which in turn governs trackable collision-avoidance local path. By inserting virtual waypoints onto the globally optimal path, a seamless interface between global and local path-planning mechanism is devised, and thereby contributing to the entire DGL hybrid path planning scheme. Simulations and comparisons in various real-world geographies demonstrate the effectiveness and superiority of the proposed DGL hybrid path planning scheme.

Index Terms—Hybrid path planning, dynamics-constrained path planning, fuzzy decision-making, fine dynamic window, autonomous surface vehicle.

I. INTRODUCTION

DUCE to the increasing utilization of autonomous surface vehicles (ASVs) in both civil and military communities, booming academic and technological advances pertaining to the ASV have emerged in latest works [1]–[12]. To be specific, in order to autonomously execute various missions and/or tasks in human-inaccessible situations, a high-autonomy path planning scheme is urgently desirable and has attracted persistent

Manuscript received March 16, 2019; revised February 15, 2020; accepted April 27, 2020. Date of publication April 29, 2020; date of current version July 16, 2020. This work was supported in part by the National Natural Science Foundation of P. R. China under Grants 51009017 and 51379002, in part by the Liaoning Revitalization Talents Program under Grant XLYC1807013, in part by the Stable Supporting Fund of Science and Technology on Underwater Vehicle Laboratory under Grant SXJQR2018WDKT03, and in part by the Fundamental Research Funds for the Central Universities under Grant 3132019344. The review of this paper was coordinated by Dr. V. Cerone. (Corresponding author: Ning Wang)

Ning Wang is with the College of Shipbuilding Engineering, Harbin Engineering University, Harbin 150001, China, and is also with the School of Marine Electrical Engineering, Dalian Maritime University, Dalian 116026, China (e-mail: n.wang@ieee.org).

Hongwei Xu is with the School of Marine Engineering, Dalian Maritime University, Dalian 116026, China.

Digital Object Identifier 10.1109/TVT.2020.2991220

attention. According to the availability of environment information, path planning methodologies can be classified into two categories i.e., deliberative and reactive architectures [13]. The former is known as off-line or global path planning, whereby the ASV is guided to the destination in a mapped environment with stationary obstacles including islands, shallow water, beacon, *etc* [14]–[15]. The latter is regarded as on-line or local path modification/planning, which requires real-time reactions to perform rapid responses to uncertainties and/or unknown obstacles in critical situations [16]–[18].

It is well recognized that the strategy of discretizing the terrain into rectangular-grid cells has provided path-planning solutions in robotics and video games [19]. Actually, the grid map-based method was derived by incorporating heuristic information into graph searching [20]. The seminal A* algorithm [21] was devised to perform heuristic search depending on known environment information. It should be pointed out that the A* may find rather long and/or non-realistic paths which require tedious heading changes since the path searching is constrained by grid edges and/or vertices. Alternatively, high-resolution grids are required to optimize more nodes within the resulting path. However, the guidance techniques, e.g., the light-of-sight (LOS) [22], hyperbolic-tangent LOS [23], error-constrained LOS [24], surge-heading guidance [25], and bridge-trajectory guidance [26] *etc.*, actually navigate the ASV along the desired path which is expected to be supported by sparsely successive waypoints [26]. In order to relax sharp constraints on grids, the A* with post-smoothed (A* PS) [27] was proposed to further smooth the path, and thereby making the resulting path more flexible by reducing the number of waypoints. As a benchmark improvement on A* algorithm, the Theta* method [28]–[29] was developed by incorporating LOS check which in turn makes the path propagate flexibly. Furthermore, simplifications on the LOS check contributed to the Lazy-Theta* [30], whereby a balance between path length and runtime is essentially required.

In order to update and/or replan an existing path according to newly-observing environment information, the dynamic A*, i.e., D* algorithm [31], was developed, whereby the re-calculation of the previously entire path can be avoided. Furthermore, the lifelong planning A* (LPA*) [32], the focussed D* and D* Lite algorithms [33] have also been proposed to make further modification and/or optimization on the path. By virtue of interpolation, the Field D* approach [34] was devised by extending D* and D* Lite algorithms, whereby unnecessary turnings would probably be reduced. In order to expedite the

optimization process of A*-based approaches, Accelerated A* [35] and Block A* [36] algorithms were developed by using adaptive sampling and blocks of grid cells, respectively. Subsequently, the A* on visibility graphs (VG) [37] was proposed to find a shortest path in known terrain with polygonal obstacles. However, the VG has to search each edge and suffers from the inefficiency that the computation burden is much heavier than that of the A* algorithm in the process of interleave. In recent years, a series of evolutionary algorithms (EA), i.e., genetic algorithm (GA) [38]–[40], ant colony optimisation (ACO) [41]–[42], and particle swarm optimization (PSO) [43], *etc.*, have been incorporated into global path planning. Note that the foregoing EA-based optimization methods are actually heuristic, and thereby probably resulting in unrealistic paths to be executed in practice.

Comparing to global case, local path planning which can accommodate unforeseen obstacles plays a fundamental role in safely maneuvering the ASV. Note that the artificial potential field (APF) method [44]–[45] features mathematic simplicity with low computational burden, but might lead to local minimum which renders the ASV stagnant or trapped. By virtue of a transition derived from global convergence, by measuring the distance to the goal, the tangent-bug algorithm [46]–[48] was proposed to avoid local minimum. By using an occupancy grid representation for obstacles information, the vector field histogram (VFH) method [49] was developed for real-time local obstacle avoidance, whereby the vehicle motion direction and velocity are calculated by occupancy grids. In [50], the enhanced vector field histogram (VFH+) method was proposed by taking the vehicle scale into consideration. For local path planning, the fast marching method (FMM) [51] has also been deployed since it can generate an effective path within sufficient time. Recently, by mimicking behaviors of fireflies, as a specific kind of swarm-intelligence-based approaches, the firefly algorithm (FFA) [52] has been employed to carry out local collision avoidance within a local path between successive waypoints. However, aforementioned local methods cannot address complex constraints on velocities and/or accelerations of the vehicle, and thereby resulting in a probably infeasible path. In other words, the paths generated by foregoing approaches would require unaccessible forces/torques pertaining to an ASV, of which both propulsive force and steering torque are limited. Intuitively, for instance, a sharp turn is actually infeasible for an ASV since steering maneuver would also lead to significant sideslip which deviates from the planned path. In this context, motion dynamics of the ASV should be sufficiently taken into account in local collision avoidance.

Inspired by foregoing observations, taking ASV dynamics and constraints into consideration in local collision avoidance, it becomes extremely challenging to find a shortest global path with trackable waypoints, and thereby establishing global-local hybrid approaches to handle unforeseen environments.

In this paper, in order to address the aforementioned challenging problem, a dynamics-constrained global-local (DGL) hybrid path planning scheme is proposed for an ASV. The global path planning algorithm can generate rather sparse waypoints

TABLE I
NOTATION SYSTEM

Notation	Description
$\mathbf{p} = [x, y]^T$	planar position coordinate of the ASV
$\mathbf{p}_0, \mathbf{p}_f$	start and target points
\mathbf{p}_{obs}	obstacle position
ψ, u, v, r	heading, surge, sway and yaw of the ASV
\underline{u}, \bar{u}	lower and upper bounds of surge velocity
\underline{r}, \bar{r}	lower and upper bounds of yaw velocity
$\underline{\varrho}_u, \bar{\varrho}_u$	lower and upper bounds of surge acceleration
$\underline{\varrho}_r, \bar{\varrho}_r$	lower and upper bounds of yaw acceleration
\mathcal{D}_{SA}	safety area of the ASV
$\mathcal{D}, \mathcal{D}_s, \mathcal{D}_d, \mathcal{D}_f$	map domain, stationary occupancies, dynamic unforeseen obstacles, and feasible space
\mathcal{D}_{SA}	safety area of the ASV
$\mathcal{D}_{\text{FDM}}, \mathcal{D}_{\text{FDW}}$	active regions of the FDM and FDW modules
$\mathcal{A}_{\text{FDM}}, \mathcal{A}_{\text{FDW}}, \mathcal{A}_l$	FDM, FDW and local actions
$\mathcal{A}_a, \mathcal{A}_r, \mathcal{A}_p, \mathcal{A}_c$	admissibility, reachability, performability, and compounded constraint sets
$\mathcal{M}_o, \mathcal{M}_b, \mathcal{M}_b^s$	original, binary and SA-meshed maps
$f_s(\cdot), f_d(\cdot)$	occupancy functions of stationary and dynamic obstacles
$f(\cdot), g(\cdot), g_s(\cdot)$	gray-value, discriminant, and square-grid discriminant functions
S_i	SA-fitted square grid of waypoint \mathbf{p}_i
$\text{prt}(\mathbf{p}_i), \text{chd}(\mathbf{p}_i)$	parent and child nodes of point \mathbf{p}_i
\mathcal{P}	path consisting of successive waypoints
\mathcal{P}_g	global path consisting of successive waypoints
$\ \cdot\ $	2-norm operation
$\mathcal{S}_{\text{act}}, \mathcal{S}_{\text{ina}}, \mathcal{S}_{\text{tree}}$	sets of active, inactive and spanning-tree waypoints
$\text{Str}(\mathbf{p}_n)$	a triple-structure
R_w, R_o	$\text{Str}(\mathbf{p}_n) := \{\mathbf{p}_n, \text{prt}(\mathbf{p}_n), \xi(\mathbf{p}_n, \text{prt}(\mathbf{p}_n))\}$
$\mu_{\text{LF}}, \mu_{\text{F}}, \mu_{\text{RF}}$	fuzzy variables of φ_w and φ_o
	membership functions of fuzzy sets LF, F, RF

which contribute to a globally optimal path. In lieu of geometric/kinematic path, the local collision-avoidance is realized by surge and yaw velocity guidance signals which are derived from a hierarchical architecture using fuzzy decision-making (FDM) and fine dynamic window (FDW) modules. Furthermore, an interface of the proposed DGL hybrid path planning is formed to encapsulate global and local path planning. As a consequence, the proposed DGL hybrid path planning scheme is able to govern a shortest global path with trackable waypoints and autonomously deal with uncertainties and/or unknown obstacles in harsh situations.

The rest of this paper is organized as follows. Notations frequently used in this paper are collected in Table I. In Section II, the ASV path planning problem associated with collision avoidance is mathematically formulated. The DGL hybrid path planning scheme is developed in Section III. In Section IV, simulation results and discussions are presented. Conclusions are drawn in Section V.

II. PROBLEM FORMULATION

In general, as shown in Fig. 1, under complex circumstances including both stationary (e.g., reefs, and islands, *etc.*) and dynamic obstacles (e.g., ships, and rigs, *etc.*), a real-time path

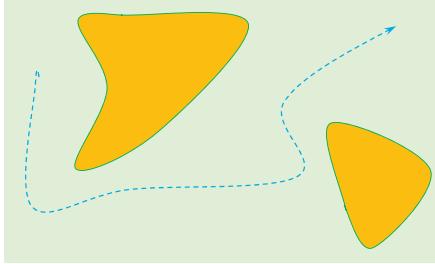


Fig. 1. Illustration of hybrid path planning for an ASV.

for the ASV is expected to be planned by a hybrid approach which would incorporate both global and local planners.

The ASV kinematics within 2-D horizontal plane are modeled as follows [22]:

$$\begin{aligned}\dot{x} &= u \cos \psi - v \sin \psi \\ \dot{y} &= u \sin \psi + v \cos \psi \\ \dot{\psi} &= r\end{aligned}\quad (1)$$

where (x, y, ψ) are the position and orientation of the ASV in the earth-fixed frame, and (u, v, r) are the surge, sway, and yaw velocities in the body-fixed frame.

Moreover, for an underactuated ASV, sway velocity v is passively bounded with respect to surge and yaw dynamics [53]–[55], and can be roughly formulated as follows [22, pp. 133–135]:

$$\dot{v} = -k_v v - k_{ur} u r + \delta_v(t) \quad (2)$$

where $k_v > 0$ and $k_{ur} > 0$ are appropriate constants, and δ_v is bounded nonlinearity satisfying $|\delta_v| < d_v$.

Proposition 1: Sway velocity v is strictly bounded by

$$|v| \leq \frac{k_{ur} d_u d_r + d_v}{k_v} \quad (3)$$

if $|u| \leq d_u$ and $|r| \leq d_r$.

Proof: Consider a Lyapunov candidate $V = \frac{1}{2}v^2$. Differentiating V along (2) and using Young's inequality yields

$$\dot{V} \leq -\frac{1}{2}k_v v^2 + \frac{1}{2k_v} (k_{ur} d_u d_r + d_v)^2 \quad (4)$$

Using [56, Theorem 4.18, pp. 172], we immediately have (3) holds. This concludes the proof. \blacksquare

In this context, it is only desirable to constrain surge and yaw within dynamic path planning, such that desired velocities are admissible for an ASV suffering from constraints on magnitudes and frequencies of control inputs, i.e.,

$$\underline{u} \leq u \leq \bar{u}, \quad \underline{r} \leq r \leq \bar{r}, \quad \underline{\varrho}_u \leq \dot{u} \leq \bar{\varrho}_u, \quad \underline{\varrho}_r \leq \dot{r} \leq \bar{\varrho}_r \quad (5)$$

where $\underline{x}, \bar{x}, \underline{\varrho}_x, \bar{\varrho}_x > 0, x \in \{u, r\}$ are constrained bounds.

As shown in Fig. 2, a safety area (SA) [57], [58] of the ASV is expected to be strictly inviolable during path planning, and is deployed to accommodate uncertainties, positioning tolerances and/or obstacle avoidance. The SA is determined by

$$\mathcal{D}_{SA}(t) = \{\mathbf{p}_s \mid \|\mathbf{p}_s - \mathbf{p}(t)\| \leq d_s\} \quad (6)$$

where $\mathbf{p}_s = [x_s, y_s]^T$, $\mathbf{p} = [x, y]^T$, and d_s is the SA radius.

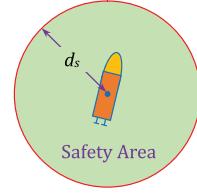


Fig. 2. Safety area of an ASV.

Consider *a priori* knowledge on stationary occupancies \mathcal{D}_s which cannot be passed through, together with unforeseen dynamic obstacles $\mathcal{D}_d(t)$, i.e.,

$$\mathcal{D}_s = \{\mathbf{p}_s \mid f_s(\mathbf{p}_s) \leq 0\} \quad (7)$$

$$\mathcal{D}_d(t) = \{\mathbf{p}_s(t) \mid f_d(\mathbf{p}_s(t)) \leq 0\} \quad (8)$$

where $f_s(\cdot)$ and $f_d(\cdot)$ are appropriate functions characterizing occupancies and/or points within the horizontal plane.

Our objective is to develop a hybrid path planning scheme for the ASV under foregoing complex environments such that an optimal path \mathcal{P} from the initial point $\mathbf{p}_0 = [x_0, y_0]^T$ to the ending point $\mathbf{p}_f = [x_f, y_f]^T$, in terms of length, safety, feasibility of constrained dynamics, etc., simultaneously, can be generated in real time within the problem domain $\mathcal{D} := \mathcal{D}_{SA} \cup \mathcal{D}_s \cup \mathcal{D}_d \cup \mathcal{D}_f$ where \mathcal{D}_f is regarded as feasible space for searching paths.

Remark 1: By virtue of Proposition 1, the upper bound of sway velocity v can be derived from bounded surge and yaw dynamics, and further rationalizes surge-yaw guidance within local path planning.

Remark 2: In this paper, in lieu of geometric kinematics in local path, the dynamic collision-avoidance is expected to be executed by surge and yaw velocity guidance signals which will be derived later from a hierarchical architecture consisting of fuzzy decision-making (FDM) and fine dynamic window (FDW) modules.

Remark 3: It should be noted that our concern still focuses on path planning algorithms, rather than controller synthesis. In this context, surge and yaw velocity guidance signals for local planning are governed by addressing ASV dynamics and feasible constraints on both velocities and accelerations. Certainly, guidance signals for surge and yaw velocities can be tracked by further synthesizing tracking controllers [59]. From the viewpoint of guidance-control architecture, it is also reasonable that only surge and yaw velocities are guided while the sway velocity is left to be passively driven by surge and yaw dynamics, since only surge and yaw dynamics can be actuated by controllers while sway dynamics are unactuated directly. As a consequence, within the proposed global-local path-planning scheme, the “dynamics-constrained” functionality is created by guiding surge and yaw velocities which are dynamically constrained by ASV dynamics and constraints.

III. HYBRID PATH PLANNING SCHEME

In this section, a dynamics-constrained global-local (DGL) hybrid path planning scheme is created by addressing both off-line and online path planning modules, simultaneously, and

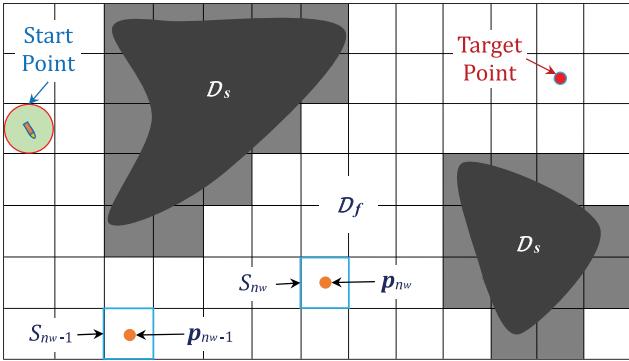


Fig. 3. A binary map with SA-fitted square-grids.

thereby contributing to the entire hybrid scheme which can tackle both *a priori* and unforeseen dynamic environments.

A. Global Path Planning

1) *Stationary Mapping*: In order to clearly partition feasible and infeasible domains, i.e., \mathcal{D}_f and \mathcal{D}_s , within the whole map domain \mathcal{D} , the binarization method is usually deployed to mesh the original map $\mathcal{M}_o(\mathbf{p}_s, f(\mathbf{p}_s))$. To be specific, a binary map $\mathcal{M}_b(\mathbf{p}_s, g(\mathbf{p}_s))$ can be characterized by a discriminant function as follows:

$$g(\mathbf{p}_s) = \begin{cases} 0, & f(\mathbf{p}_s) \leq \vartheta \\ 255, & f(\mathbf{p}_s) > \vartheta \end{cases} \quad (9)$$

where $\vartheta > 0$ is a threshold, and f is gray-value function.

In this context, we have

$$\mathcal{D}_f = \{\mathbf{p}_s \mid g(\mathbf{p}_s) = 0\}, \quad \mathcal{D}_s = \{\mathbf{p}_s \mid g(\mathbf{p}_s) = 255\} \quad (10)$$

Within the feasible space \mathcal{D}_f characterized by binary map \mathcal{M}_b , global path can be optimized by searching a series of waypoints. To this end, as shown in Fig. 3, using SA-fitted square grids $S_i(\mathbf{p})$ with width $2d_s$ and centering potential waypoint $\mathbf{p}_i = [x_i, y_i]^T$, i.e., $S_i(\mathbf{p}) = \{\mathbf{p} = [x, y]^T \mid |x - x_i| \leq d_s, |y - y_i| \leq d_s\}$, the binary map \mathcal{M}_b is converted to be an SA-meshed binary map $\mathcal{M}_b^s(S_i, g_s(S_i))$ with square-grid discriminant function g_s governed by

$$g_s(S_i) = \max_{\mathbf{p} \in S_i} \{g(\mathbf{p})\}. \quad (11)$$

2) *Path Planning*: For an ASV in the SA-meshed binary map \mathcal{M}_b^s , the general objective of global path planning is to minimize the path length L_p given by

$$L_p = \sum_{i=1}^{N_w} \|\mathbf{p}_i - \mathbf{p}_{i-1}\| \quad (12)$$

where N_w is the total number of planned waypoints, and \mathbf{p}_0 and $\mathbf{p}_{N_w} = \mathbf{p}_f$ are starting and target points, respectively.

Alternatively, inspired by Theta* algorithm [28], the objective can be incrementally evaluated by minimizing the cost function as follows:

$$\xi(\mathbf{p}_{n_c}, \text{prt}(\mathbf{p}_{n_c})) = \xi_1(\mathbf{p}_{n_c}, \text{prt}(\mathbf{p}_{n_c})) + \xi_2(\mathbf{p}_{n_c}) \quad (13)$$

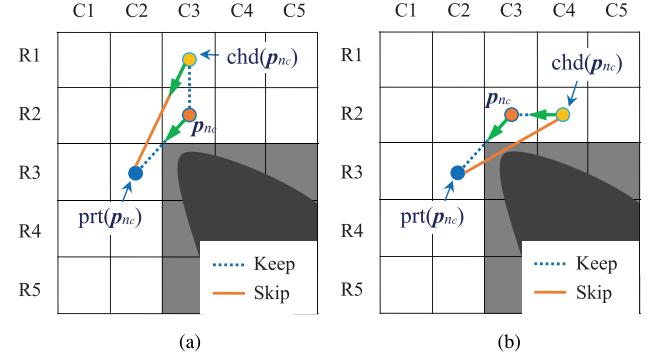


Fig. 4. The LOS strategy in global path planning. (a) LOS appears. (b) LOS does not appear.

where \mathbf{p}_{n_c} is the current candidate waypoint, $\text{prt}(\mathbf{p}_{n_c})$ is the parent waypoint of \mathbf{p}_{n_c} (conversely, $\mathbf{p}_{n_c} \in \text{chd}(\text{prt}(\mathbf{p}_{n_c}))$), i.e., \mathbf{p}_{n_c} is one child node of $\text{prt}(\mathbf{p}_{n_c})$; clearly, $\text{prt}(\mathbf{p}_0) = \mathbf{p}_0$, $\xi_1(\mathbf{p}_{n_c}, \text{prt}(\mathbf{p}_{n_c}))$ is the length of partial-path $\mathcal{P}_p := \{\mathbf{p}_0, \dots, \text{prt}^i(\mathbf{p}_{n_c}), \dots, \text{prt}(\mathbf{p}_{n_c}), \mathbf{p}_{n_c}\}$ with $\text{prt}^{i+1}(\mathbf{p}_{n_c}) = \text{prt}(\text{prt}^i(\mathbf{p}_{n_c}))$, $\mathbf{p}_0 = \text{prt}^{n_p}(\mathbf{p}_{n_c})$ and $n_p = |\mathcal{P}_p| - 1$, and $\xi_2(\mathbf{p}_{n_c})$ is the Euclidean distance between the current node \mathbf{p}_{n_c} and the target point \mathbf{p}_f , i.e.,

$$\begin{aligned} \xi_1(\mathbf{p}_{n_c}, \text{prt}(\mathbf{p}_{n_c})) \\ = \xi_1(\text{prt}(\mathbf{p}_{n_c}), \text{prt}^2(\mathbf{p}_{n_c})) + \|\mathbf{p}_{n_c} - \text{prt}(\mathbf{p}_{n_c})\| \end{aligned} \quad (14)$$

$$\xi_2(\mathbf{p}_{n_c}) = \|\mathbf{p}_{n_c} - \mathbf{p}_f\| \quad (15)$$

with $\xi_1(\mathbf{p}_0, \text{prt}(\mathbf{p}_0)) = 0$.

To this end, three sets, i.e., \mathcal{S}_{act} , \mathcal{S}_{ina} and $\mathcal{S}_{\text{tree}}$, are defined to collect active, inactive and spanning-tree waypoints, respectively, whereby \mathcal{S}_{act} stores associated waypoints which have not been expanded so far, \mathcal{S}_{ina} seals expanded waypoints which become inactive for ever, and $\mathcal{S}_{\text{tree}}$ collects waypoints which have been spanned for constructing the optimal path. Within \mathcal{S}_{act} and $\mathcal{S}_{\text{tree}}$, triple-structures $\text{Str}(\mathbf{p}_{n_i}) := \{\mathbf{p}_{n_i}, \text{prt}(\mathbf{p}_{n_i}), \xi(\mathbf{p}_{n_i}, \text{prt}(\mathbf{p}_{n_i}))\}$ are deployed as individual elements, such that waypoints \mathbf{p}_{n_i} can be sorted by $\xi(\mathbf{p}_{n_i}, \text{prt}(\mathbf{p}_{n_i}))$ in \mathcal{S}_{act} , and can be successively indexed by $\text{prt}(\mathbf{p}_{n_i})$ in $\mathcal{S}_{\text{tree}}$, respectively.

To recursively minimize the cost ξ , the optimal candidate \mathbf{p}_{n_c} in current stage is selected from the active set \mathcal{S}_{act} , i.e.,

$$\mathbf{p}_{n_c} = \arg \min_{\text{Str}(\mathbf{p}_{n_i}) \in \mathcal{S}_{\text{act}}} \{\xi(\mathbf{p}_{n_i}, \text{prt}(\mathbf{p}_{n_i}))\} \quad (16)$$

and is pushed into the set $\mathcal{S}_{\text{tree}}$, i.e., $\mathcal{S}_{\text{tree}} \leftarrow \mathcal{S}_{\text{tree}} \cup \text{Str}(\mathbf{p}_{n_c})$. Meanwhile, the point \mathbf{p}_{n_c} becomes inactive for ever, i.e., $\mathcal{S}_{\text{ina}} \leftarrow \mathcal{S}_{\text{ina}} \cup \mathbf{p}_{n_c}$ and $\mathcal{S}_{\text{act}} \leftarrow \mathcal{S}_{\text{act}} \setminus \text{Str}(\mathbf{p}_{n_c})$.

Expanding from \mathbf{p}_{n_c} , all child waypoints which have not been expanded, i.e., $\mathbf{p}_{n'_c} \in \text{chd}(\mathbf{p}_{n_c})$ and $\mathbf{p}_{n'_c} \notin \mathcal{S}_{\text{ina}}$, are firstly checked whether there exists a light-of-sight (LOS) from its grandparent $\text{prt}(\mathbf{p}_{n_c})$, i.e., $\text{LOS}(\text{prt}(\mathbf{p}_{n_c}), \mathbf{p}_{n'_c})$ shown in Fig. 4, where $\text{LOS}(\cdot, \cdot)$ is a discriminant function with output 1 or 0 for an LOS appears or not. If $\text{LOS}(\text{prt}(\mathbf{p}_{n_c}), \mathbf{p}_{n'_c}) = 1$, the parent is changed to be its grandparent, i.e., $\text{prt}(\mathbf{p}_{n'_c}) \leftarrow \text{prt}(\mathbf{p}_{n_c})$. Otherwise, the parent is updated by $\text{prt}(\mathbf{p}_{n'_c}) \leftarrow \mathbf{p}_{n_c}$. Accordingly, both costs $\xi_1(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c}))$ and $\xi(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c}))$

are updated. In addition, the triple-structure $Str(\mathbf{p}_{n'_c}) = \{\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c}), \xi(\mathbf{p}_i, \text{prt}(\mathbf{p}_{n'_c}))\}$ is refreshed in/into \mathcal{S}_{act} .

Eventually, if the target point \mathbf{p}_f is reached, i.e., $\mathbf{p}_{n_c} = \mathbf{p}_f$, the optimal global-path $\mathcal{P}_g := \{\mathbf{p}_0, \dots, \mathbf{p}_{n_w}, \dots, \mathbf{p}_f\}$ is extracted from $\mathcal{S}_{\text{tree}}$, i.e., $\mathcal{P}_g = \text{path}(\mathcal{S}_{\text{tree}})$ with $\text{path}(\mathcal{S}_{\text{tree}}) = \{\mathbf{p}_{n_w} = \text{prt}^{N_w - n_w}(\mathbf{p}_f) | Str(\mathbf{p}_{n_w}) \in \mathcal{S}_{\text{tree}}, N_w = |\mathcal{S}_{\text{tree}}|\}$.

To sum up, the entire mechanism for searching global path \mathcal{P}_g is summarized in Algorithm 1.

Remark 4: Different from the Theta* algorithm [28] achieving any-angle paths via grid vertexes, in addition to the LOS-based Skip Path trimming, the proposed global path planning scheme in this paper exclusively works for the ASV in an SA-meshed binary map \mathcal{M}_b^s , such that both safety and feasibility of the global path can be ensured by locating the ASV at the center of the SA-grids, of which the radius d_s has sufficiently accommodated the ASV maneuverability. Moreover, triple-structures $Str(\mathbf{p}_{n_i})$ and tree set $\mathcal{S}_{\text{tree}}$ are devised to facilitate sorting and indexing of waypoints, and thereby avoiding repeated computation on partial cost ξ_1 .

Remark 5: Using a proof by contradiction, the proposed global path planning, i.e., Algorithm 1, can be proved to generate the shortest path if $\mathcal{P}_g \neq \emptyset$. Assume that there exists a much shorter path \mathcal{P}'_g , i.e., $\xi(\mathcal{P}'_g) < \xi(\mathcal{P}_g)$. This assumption definitely implies that there must exist waypoints $\mathbf{p}'_i \in \mathcal{P}'_g$ and $\mathbf{p}_j \in \mathcal{P}_g$ satisfying $\xi_1(\mathbf{p}_j, \text{prt}(\mathbf{p}_j)) \geq \xi_1(\mathbf{p}'_i, \text{prt}(\mathbf{p}'_i))$. From lines 7, 18, 19, 23, and 24 in Algorithm 1, we can always find $\mathbf{p}_k := \text{prt}^{j-k}(\mathbf{p}_j) \in \mathcal{P}_g, 0 \leq k < j$ such that $\xi_1(\mathbf{p}_k, \text{prt}(\mathbf{p}_k)) < \xi_1(\mathbf{p}'_i, \text{prt}(\mathbf{p}'_i))$ and $\xi(\mathbf{p}_k, \text{prt}(\mathbf{p}_k)) < \xi(\mathbf{p}'_i, \text{prt}(\mathbf{p}'_i))$. Hence, we have $\mathbf{p}_k = \text{prt}^q(\mathbf{p}'_i)$ for an appropriate integer $q \geq 0$. It implies that $\mathcal{P}'_g \cap \mathcal{P}_g \supseteq \{\mathbf{p}_0, \dots, \mathbf{p}_k\}$. Recursively, from Algorithm 1, we have $\mathcal{P}'_g \cap \mathcal{P}_g \supseteq \mathcal{P}_g$, i.e., $\xi(\mathcal{P}'_g) \geq \xi(\mathcal{P}_g)$, and thereby leading to the contradiction.

B. Local Path Planning

1) *Hierarchical Architecture:* In order to facilitate high autonomy within path planning of an ASV, unforeseen stationary and dynamic obstacles are expected to be avoided efficiently by adjusting local paths. To this end, a hierarchical architecture of local path planning is developed by combining with fuzzy decision-making (FDM) and fine dynamic window (FDW) [60] approaches. To be specific, the FDM and FDW are responsible for large- and close-range reactions to unforeseen obstacles, respectively, i.e.,

$$\mathcal{A}_l = \begin{cases} \mathcal{A}_{\text{FDM}}(\mathbf{p}, \mathbf{p}_{\text{obs}}), & \mathbf{p}_{\text{obs}} \in \mathcal{D}_{\text{FDM}} \\ \mathcal{A}_{\text{FDW}}(\mathbf{p}, \mathbf{p}_{\text{obs}}), & \mathbf{p}_{\text{obs}} \in \mathcal{D}_{\text{FDW}} \end{cases} \quad (17)$$

where $\mathcal{A}_l := (u^*, r^*)$ are local actions for collision avoidance, \mathcal{A}_{FDM} and \mathcal{A}_{FDW} are FDM and FDW actions determined later, respectively, $\mathbf{p}_{\text{obs}} = [x_{\text{obs}}, y_{\text{obs}}]^T$ is the obstacle position, \mathcal{D}_{FDM} and \mathcal{D}_{FDW} are active regions of the FDM and FDW modules, respectively, and are defined by

$$\mathcal{D}_{\text{FDM}} = \left\{ \mathbf{p}_{\text{obs}} \mid L_{\text{FDW}} \leq \|\mathbf{p}_{\text{obs}} - \mathbf{p}\| - d_s - d_{so} \leq L_{\text{FDM}} \right\} \quad (18)$$

$$\mathcal{D}_{\text{FDW}} = \left\{ \mathbf{p}_{\text{obs}} \mid 0 < \|\mathbf{p}_{\text{obs}} - \mathbf{p}\| - d_s - d_{so} < L_{\text{FDW}} \right\} \quad (19)$$

TABLE II
FUZZY RULE BASE OF FDM MODULE

r_{FDM}	$R_o(\varphi_o)$		
	LF	F	RF
	$R_w(\varphi_w)$	N	N
LF	Z	P	Z
RF	P	P	P

Algorithm 1: Global Path Planning

```

Input:  $\mathbf{p}_0, \mathbf{p}_f, \mathcal{M}_b^s$ 
Output:  $\mathcal{P}_g = \{\mathbf{p}_0, \dots, \mathbf{p}_{n_w}, \dots, \mathbf{p}_f\}$ 
1  $\mathcal{S}_{\text{act}} = \emptyset, \mathcal{S}_{\text{ina}} = \emptyset, \mathcal{S}_{\text{tree}} = \emptyset;$ 
2  $\text{prt}(\mathbf{p}_0) \leftarrow \mathbf{p}_0;$ 
3  $\xi_1(\mathbf{p}_0, \text{prt}(\mathbf{p}_0)) = 0, \xi(\mathbf{p}_0, \text{prt}(\mathbf{p}_0)) = \xi_1(\mathbf{p}_0, \text{prt}(\mathbf{p}_0)) + \xi_2(\mathbf{p}_0);$ 
4  $Str(\mathbf{p}_0) = \{\mathbf{p}_0, \text{prt}(\mathbf{p}_0), \xi(\mathbf{p}_0, \text{prt}(\mathbf{p}_0))\};$ 
5  $\mathcal{S}_{\text{act}} \leftarrow \mathcal{S}_{\text{act}} \cup Str(\mathbf{p}_0);$ 
6 while  $\mathcal{S}_{\text{act}} \neq \emptyset$  do
7    $\mathbf{p}_{n_c} = \arg \min_{Str(\mathbf{p}_{n_i}) \in \mathcal{S}_{\text{act}}} \{\xi(\mathbf{p}_{n_i}, \text{prt}(\mathbf{p}_{n_i}))\};$ 
8    $\mathcal{S}_{\text{tree}} \leftarrow \mathcal{S}_{\text{tree}} \cup Str(\mathbf{p}_{n_c});$ 
9   if  $\mathbf{p}_{n_c} = \mathbf{p}_f$  then
10    | return  $\mathcal{P}_g = \text{path}(\mathcal{S}_{\text{tree}});$ 
11    $\mathcal{S}_{\text{ina}} \leftarrow \mathcal{S}_{\text{ina}} \cup \mathbf{p}_{n_c};$ 
12    $\mathcal{S}_{\text{act}} \leftarrow \mathcal{S}_{\text{act}} \setminus Str(\mathbf{p}_{n_c});$ 
13   for  $\mathbf{p}_{n'_c} \in \text{chd}(\mathbf{p}_{n_c})$  do
14    | if  $\mathbf{p}_{n'_c} \notin \mathcal{S}_{\text{ina}}$  then
15     | | if  $\text{LOS}(\text{prt}(\mathbf{p}_{n_c}), \mathbf{p}_{n'_c})$  then
16     | | | /* Skip Path */
17     | | |  $\text{prt}(\mathbf{p}_{n'_c}) \leftarrow \text{prt}(\mathbf{p}_{n_c});$ 
18     | | | if  $\xi_1(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n_c})) < \xi_1(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c}))$  then
19     | | | |  $\xi_1(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c})) = \xi_1(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n_c}));$ 
20    | | else
21     | | | /* Keep Path */
22     | | |  $\text{prt}(\mathbf{p}_{n'_c}) \leftarrow \mathbf{p}_{n'_c};$ 
23     | | | if  $\xi_1(\mathbf{p}_{n'_c}, \mathbf{p}_{n_c}) < \xi_1(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c}))$  then
24     | | | |  $\xi_1(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c})) = \xi_1(\mathbf{p}_{n'_c}, \mathbf{p}_{n_c});$ 
25   | |  $\xi(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c})) = \xi_1(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c})) + \xi_2(\mathbf{p}_{n'_c});$ 
26   | if  $Str(\mathbf{p}_{n'_c}) \in \mathcal{S}_{\text{act}}$  then
27   | |  $Str(\mathbf{p}_{n'_c}) = \{\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c}), \xi(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c}))\};$ 
28   | else
29   | |  $Str(\mathbf{p}_{n'_c}) = \{\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c}), \xi(\mathbf{p}_{n'_c}, \text{prt}(\mathbf{p}_{n'_c}))\};$ 
30   | |  $\mathcal{S}_{\text{act}} \leftarrow \mathcal{S}_{\text{act}} \cup Str(\mathbf{p}_{n'_c});$ 
31 return  $\mathcal{P}_g = \emptyset;$ 

```

with velocity-dependent obstacle margin expansion $d_{so} = \eta \cdot u$ where $\eta > 0$, and thresholds $0 < L_{\text{FDW}} < L_{\text{FDM}} < \infty$.

2) *FDM Module:* This functional layer is expected to roughly execute large-range avoidance. To this end, a fuzzy inference system is established by deploying relative orientations of successive waypoints and obstacles to the ASV, respectively.

To be specific, the fuzzy rule base is built in Table I, whereby input fuzzy sets “LF”, “F”, and “RF” (i.e., left forward, forward, and right forward, respectively) of input fuzzy variables R_w and R_o are defined by the following membership functions:

$$\mu_{\text{LF},*}(\varphi_{*}) = \frac{1}{1 + \exp(\gamma_{\text{L}}(\varphi_{*} - 0.5\varphi_{\text{LF}}))} \quad (20)$$

$$\mu_{F,*}(\varphi_*) = \exp\left(-\frac{(\varphi_* - \varphi_F)^2}{2\sigma^2}\right) \quad (21)$$

$$\mu_{RF,*}(\varphi_*) = \frac{1}{1 + \exp(-\gamma_R(\varphi_* - 0.5\varphi_{RF}))} \quad (22)$$

where $\star \in \{w, o\}$, φ_w and φ_o are relative orientations of waypoints and obstacles, respectively, with respect to the ASV, $\varphi_{LF} < 0$, $\varphi_F = 0$ and $\varphi_{RF} > 0$ are centers of corresponding fuzzy sets, $\sigma > 0$ is the width, and $\gamma_L = -10/\varphi_{LF}$ and $\gamma_R = -10/\varphi_{RF}$. In addition, output fuzzy sets “N”, “Z”, and “P” (i.e., negative, zero, and positive, respectively) are singleton numbers trivially defined by $w_N = \lambda \underline{r}$, $w_Z = 0$, and $w_P = \lambda \bar{r}$ with $0 < \lambda \leq 1$.

In this context, using fuzzy rule base in Table I, the FDM module can be realized by fuzzy inference system as follows:

Rule₁: IF φ_w is LF and φ_o is LF, THEN $r_{FDM} = w_N$

⋮

Rule₉: IF φ_w is RF and φ_o is RF, THEN $r_{FDM} = w_P$ (23)

Accordingly, using product-sum fuzzy inference mechanism yields the FDM action as follows:

$$\mathcal{A}_{FDM} = \left\{ (u, r_{FDM}) \mid r_{FDM} = \mathbf{w}^T \boldsymbol{\phi}(\varphi_w, \varphi_o) \right\} \quad (24)$$

where $\mathbf{w} = [\mathbf{w}_1^T, \mathbf{w}_2^T, \mathbf{w}_3^T]^T$ with $\mathbf{w}_1 = [w_N, w_N, w_N]^T$, $\mathbf{w}_2 = [w_Z, w_P, w_Z]^T$ and $\mathbf{w}_3 = [w_P, w_P, w_P]^T$, and $\boldsymbol{\phi} = [\boldsymbol{\phi}_{LF}^T, \boldsymbol{\phi}_F^T, \boldsymbol{\phi}_{RF}^T]^T$ with

$$\boldsymbol{\phi}_*(\varphi_w, \varphi_o) = \frac{\mu_{*,w}(\varphi_w)}{\Delta} \begin{bmatrix} \mu_{LF,o}(\varphi_o) \\ \mu_{F,o}(\varphi_o) \\ \mu_{RF,o}(\varphi_o) \end{bmatrix} \quad (25)$$

$$\Delta = \sum_{*\in\{LF,F,RF\}} \mu_{*,w}(\varphi_w) \mu_{*,o}(\varphi_o) \quad (26)$$

for $* \in \{LF, F, RF\}$.

3) *FDW Module*: In this internal layer, close-range obstacle avoidance is feasibly conducted by elaboratively addressing constraints on ASV dynamics, since dynamic local planning has to be strictly accessible in a close-quarters situation. Especially for an underactuated ASV, only surge u and yaw r dynamics are actively driven. In this context, a velocity pair (u, r) should be timely feasible for control forces and torques, and can be assumed as constants within a small time-interval, and thereby contributing to arc-paths in local planning.

To this end, three-fold constraints, i.e., admissibility, reachability, and performability, are addressed in the sequel.

Within the FDW module, admissible velocities are required to ensure that the ASV can urgently stop for emergent avoidance. An admissibility set \mathcal{A}_a is characterized by

$$\mathcal{A}_a = \left\{ (u, r) \mid |u| \leq \sqrt{2\varrho_u^* \|\mathbf{p} - \mathbf{p}_{obs}\|}, |r| \leq \sqrt{\pi \varrho_r^*} \right\} \quad (27)$$

where $\varrho_u^* = \min\{-\underline{\varrho}_u, \bar{\varrho}_u\}$ and $\varrho_r^* = \min\{-\underline{\varrho}_r, \bar{\varrho}_r\}$.

The reachability set \mathcal{A}_r is defined to shape a dynamic window such that the planned velocities can be reached feasibly within the next time-interval. It implies that surge and yaw

accelerations, i.e., \dot{u} and \dot{r} , are required to be constrained by $\underline{\varrho}_*, \bar{\varrho}_* > 0, * \in \{u, r\}$ in (5), respectively. On the basis of current velocities (u_c, r_c) of the ASV, the reachability set \mathcal{A}_r is governed by

$$\mathcal{A}_r = \left\{ (u, r) \mid u \in [u_c + T\underline{\varrho}_u, u_c + T\bar{\varrho}_u], r \in [r_c + T\underline{\varrho}_r, r_c + T\bar{\varrho}_r] \right\} \quad (28)$$

where $T > 0$ is the time period.

In addition, due to inherent limitations on actuator dynamics, the performability set \mathcal{A}_p is utilized to saturate the planned velocities as follows:

$$\mathcal{A}_p = \left\{ (u, r) \mid u \in [\underline{u}, \bar{u}], r \in [\underline{r}, \bar{r}] \right\} \quad (29)$$

In this context, combining with \mathcal{A}_a (27), \mathcal{A}_r (28), and \mathcal{A}_p (29), the compounded constraint set \mathcal{A}_c can be obtained

$$\mathcal{A}_c = \mathcal{A}_a \cap \mathcal{A}_r \cap \mathcal{A}_p \quad (30)$$

Subject to constraints by \mathcal{A}_c in (30), optimal velocities are acquired by maximizing the cost function given by

$$\mathcal{F}_c(u, r) = k_1 \mathcal{H}(u, r) + k_2 \mathcal{L}(u, r) + k_3 \mathcal{V}(u, r) \quad (31)$$

where $k_1, k_2, k_3 > 0$ are user-defined weights, and functions \mathcal{H} , \mathcal{L} and \mathcal{V} evaluate the predicted path in terms of heading to the successive waypoint or target, the distance to the adjacent obstacle that intersects with the arc-path, and the speed of approaching to the successive waypoint or target, respectively, and are defined as follows:

$$\mathcal{H} = 180 - |\varphi_w(\mathbf{p}(u, r)) - \varphi_o(\mathbf{p}(u, r))| \quad (32)$$

$$\mathcal{L} = \|\mathbf{p}(u, r) - \mathbf{p}_{obs}\| - d_s - d_{so} \quad (33)$$

$$\mathcal{V} = -\|\mathbf{p}(u, r) - \mathbf{p}_w\| \quad (34)$$

The FDW action set \mathcal{A}_{FDW} can be eventually derived

$$\mathcal{A}_{FDW} = \left\{ (u_{FDW}, r_{FDW}) = \arg \max_{(u, r) \in \mathcal{A}_c} \mathcal{F}_c(u, r) \right\} \quad (35)$$

In summary, the entire local path planning is established in Algorithm 2.

Remark 6: Comparing to the FDW method [60] focusing on close collision-avoidance of a mobile robot in a limited space, the FDM and FDW modules within the proposed hierarchy of local path planning are cohered to collaboratively execute collision avoidance in both large- and close-range situations.

Remark 7: Within the proposed local path planning (i.e., Algorithm 2), collision risks are divided into 2 hierarchies, i.e., large and close ranges, and are sufficiently accommodated by the FDM and FDW modules, respectively. Moreover, an SA pertaining to the ASV is also imposed to measure safe distances. In this context, the safety of locally planned path can always be ensured by selecting appropriate parameters, and will be demonstrated by simulation results in Section IV.

Remark 8: From (31)–(35), we can observe that the FDW-based velocity guidance is directly derived from the resultant oriental and geometric discrepancies, i.e., (32)–(34), which have

Algorithm 2: Local Path Planning

Input: $\mathbf{p}_{\text{obs}}, u, r, \mathcal{P}_g, \underline{u}, \bar{u}, \underline{\varrho}_u, \bar{\varrho}_u, \underline{r}, \bar{r}, \underline{\varrho}_r, \bar{\varrho}_r, t, T, \mathbf{w}, k_1, k_2, k_3$
Output: \mathcal{A}_l

- 1 Calculate φ_w and φ_o ;
- 2 if $\mathbf{p}_{\text{obs}} \in \mathcal{D}_{\text{FDM}}$ then
- 3 /* FDM */
- 4 Calculate $\mu_{*,*,*} \in \{\text{LF, F, RF}\}, * \in \{w, o\}$;
- 5 Calculate $\phi(\varphi_w, \varphi_o)$;
- 6 $r_{\text{FDM}} = \mathbf{w}^T \phi(\varphi_w, \varphi_o)$;
- 7 $\mathcal{A}_l \leftarrow (u, r_{\text{FDM}})$;
- 8 if $\mathbf{p}_{\text{obs}} \in \mathcal{D}_{\text{FDW}}$ then
- 9 /* FDW */
- 10 Update $\mathcal{A}_a, \mathcal{A}_r$, and \mathcal{A}_p ;
- 11 $\mathcal{A}_c = \mathcal{A}_a \cap \mathcal{A}_r \cap \mathcal{A}_p$;
- 12 $\mathcal{H} = 180 - |\varphi_w - \varphi_o|$;
- 13 $\mathcal{L} = \|\mathbf{p} - \mathbf{p}_{\text{obs}}\| - d_s - d_{so}$;
- 14 $\mathcal{V} = -\|\mathbf{p} - \mathbf{p}_w\|$;
- 15 $\mathcal{F}_c = k_1 \mathcal{H} + k_2 \mathcal{L} + k_3 \mathcal{V}$;
- 16 $(u_{\text{FDW}}, r_{\text{FDW}}) = \arg \max_{(u, r) \in \mathcal{A}_c} \mathcal{F}_c(u, r)$;
- 17 $\mathcal{A}_l \leftarrow (u_{\text{FDW}}, r_{\text{FDW}})$;

Algorithm 3: DGL Hybrid Path Planning

Input: $\mathbf{p}_0, \mathbf{p}_f, \mathbf{p}_{\text{obs}}, \mathcal{M}_b^s, \underline{u}, \bar{u}, \underline{\varrho}_u, \bar{\varrho}_u, \underline{r}, \bar{r}, \underline{\varrho}_r, \bar{\varrho}_r, T, \epsilon$
Output: \mathcal{P}_{DGL}

- 1 /* Global Path Planning */
- 2 $\mathcal{P}_g \leftarrow \text{Algorithm 1}$;
- 3 $\mathcal{P}_{\text{DGL}} \leftarrow \mathcal{P}_g$;
- 4 $t = 0$;
- 5 while $\|\mathbf{p} - \mathbf{p}_f\| > \epsilon$ do
- 6 $t \leftarrow t + T$;
- 7 /* Local Path Planning */
- 8 $\mathcal{A}_l \leftarrow \text{Algorithm 2}$;
- 9 $\mathcal{P}_{\text{DGL}} \leftarrow \mathcal{P}_{\text{DGL}} \cup \mathcal{A}_l$;
- 10 if $t > T_{\text{max}}$ then
- 11 return "The target \mathbf{p}_f can not be reached";
- 12 return \mathcal{P}_{DGL} , and "The target \mathbf{p}_f is reached";

accommodated ASV dynamics driven by surge and yaw velocities, while the sway velocity is regarded as an intermediate state since it is passively actuated by surge and yaw dynamics. To be more intensive, within the local path planning, surge-yaw velocities and the cost function \mathcal{F}_c in (31) are taken as inputs and output/objective, respectively, while the position, orientation and sway velocity of the ASV are all intermediate states. In this context, the existence of sway velocity has been sufficiently accommodated by geometric penalty (33) and reward (34), and would definitely not arise path-planning deviation.

C. DGL Hybrid Path Planning Scheme

As summarized in Algorithm 3, combining with global and local path planning algorithms contributes to the entire DGL hybrid path planning scheme, whereby *a priori* information is sufficiently explored within off-line path planning while online modification executes dynamic avoidance for unknown and/or unforeseen obstacles. It should be noted that a threshold $\epsilon > 0$ of target-reaching tolerance is deployed in case of computation truncation and/or sensor noises. Moreover, the time spending on

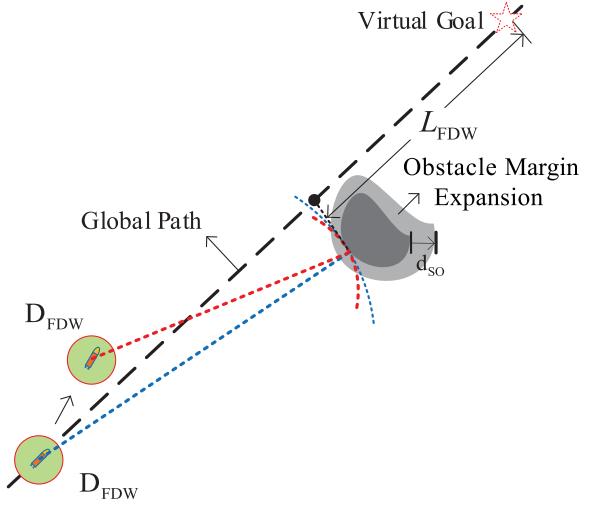


Fig. 5. The coherence of DGL hybrid path planning scheme.

the whole DGL hybrid path planning procedure is also reasonably limited by $T_{\text{max}} < \infty$ such that endless loop can be avoided. Furthermore, in order to smoothly incorporate local planning into global scheme, as shown in Fig. 5, a virtual waypoint is inserted onto the global path and is L_{FDW} ahead of the obstacle. It implies that the virtual waypoint is the return point to global path.

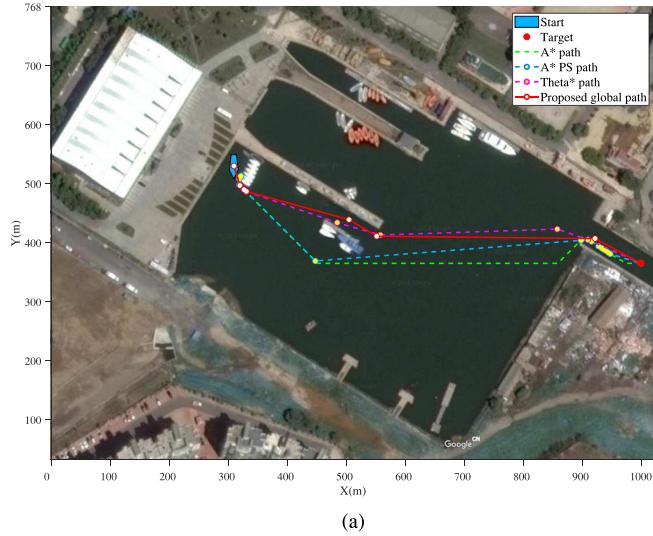
Remark 9: It should be noted that, in Algorithms 2 and 3, the compounded constraint set \mathcal{A}_c governed by (30) is intersected by admissibility set \mathcal{A}_a , reachability set \mathcal{A}_r and performability set \mathcal{A}_p , respectively. Moreover, the reachability set \mathcal{A}_r in (28) is essentially constrained by surge and yaw accelerations.

Remark 10: In combination with Remarks 2, 3, 8 and 9, we can intensively conclude that ASV dynamics and constraints can be sufficiently accommodated by the proposed local path-planning hierarchy within the DGL hybrid path planning scheme. In this context, from the viewpoint of hybrid path planning philosophy, the "dynamics-constrained" feature is reasonably understandable within the proposed DGL scheme.

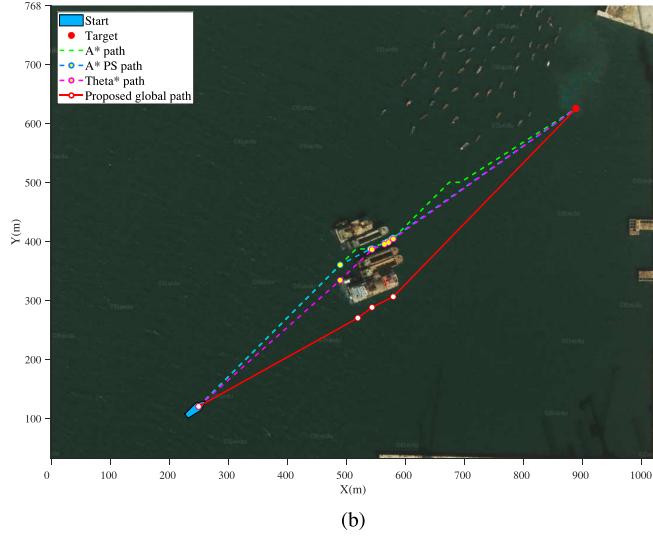
Remark 11: Comparing to the Theta* method [28], SA-grids, triple-structures and a tree set are deployed in Algorithm 1, such that repeated computations on partial costs are saved while the required storage would increase. In Algorithm 2 whereby the FDM and FDW modules are cohered, the complexity is identical to fuzzy logic and the FDW algorithm [60].

Remark 12: In summary, salient contributions pertaining to the DGL scheme can be unfolded as follows:

- A DGL hybrid path planning framework consisting of globally sparse waypoints generation and locally dynamic collision-avoidance navigation is established.
- By devising SA-fitted square grid and LOS-like pruning technique, globally optimal waypoints are generated with sufficient sparsity, and can further flexibly support potential modifications in local planning.
- The local path-planning hierarchy is created by integrating the FDM and FDW functionality layers, which are in charge of large and close ranges, respectively, such that



(a)



(b)

Fig. 6. Comparisons of the proposed global path planning to A*, A* PS and Theta* algorithms in Linghai Harbor and Dayao Bay, respectively. (a) Linghai Harbor. (b) Dayao Bay.

both surge and yaw velocities can be optimally guided by accommodating ASV dynamics and constraints.

IV. SIMULATION STUDIES

In order to demonstrate effectiveness and superiority of the proposed DGL hybrid path planning scheme, global, local, and hybrid path planning scenarios are conducted comprehensively in various real-world geographies, i.e., Linghai Harbor and Dayao Bay, Dalian, China, which are shown in Fig. 6. All simulations are performed in a PC with Intel i7 3.4 GHz and 8GB RAM, running by the MATLAB R2016a. Constraints on the ASV dynamics are listed in Table II.

A. Global Path Planning

In this subsection, comparisons of the proposed global path planning algorithm to the A*, A* PS and Theta* methods are shown in Fig. 6 and Table III, from which it can be observed

TABLE III
CONSTRAINTS ON THE ASV DYNAMICS

u	-1.0 m/s	\bar{u}	1.2 m/s
r	-0.15 rad/s	\bar{r}	0.2 rad/s
$\underline{\varrho}_u$	-0.15 m/s^2	$\bar{\varrho}_u$	0.2 m/s^2
$\underline{\varrho}_r$	-0.1 rad/s^2	$\bar{\varrho}_r$	0.1 rad/s^2
d_u	1.2	d_r	0.2
d_v	1		

TABLE IV
COMPARISONS OF GLOBAL PATH PLANNING

Map	Alg.	Waypoint	Init. Path (m)	Time Cost (s)
Linghai	A*	714	808.27	11.23
	A* PS	18	782.32	12.02
	Theta*	9	741.89	19.32
	DGL	8	743.20	20.90
Dayao	A*	446	833.87	8.07
	A* PS	10	820.32	9.23
	Theta*	7	817.69	16.58
	DGL	5	823.93	16.93

that the A* algorithm obtains a global path using numerous waypoints, while the A* PS method significantly reduces the number of waypoints and further shortens the path length. It should be noted that the Theta* algorithm generates the shortest path via even less waypoints. However, as shown in Fig. 7, the Theta* algorithm executes radically, and makes the waypoints much closer to the obstacles. Comparing to the A*, A* PS, and Theta* algorithms, the proposed global path planning approach can find a nearly shortest path supported by pretty sparse waypoints.

More practically, within the proposed global path planning algorithm, the SA expansion of the ASV is devised in addition that the obstacles are expanded, such that the resultant path preserves uniformly safe distances to the obstacles. Using the the A* PS and Theta* algorithms which cannot accommodate safety areas, as shown in Fig. 7, the distances to closet obstacles show that the waypoints are rather close to obstacles, and thereby rendering the path practically infeasible. Fortunately, in the proposed global path planning algorithm, using the SA expansion makes the distances to closet obstacles no less than 3.7m, and thereby guaranteeing globally safe navigation.

To be more interesting, as shown in Figs. IV and 7(b), for a narrow waterway which might arise collision risk, the proposed global path planning algorithm can avoid passing through the risky strait. However, the A*, A* PS and Theta* methods uniformly tend to pursue short paths without considering the security of maneuvering. Clearly, the proposed global path planning algorithm is able to sufficiently balance the length-shortness and risk-avoidance.

In addition, time consumption is also a crucial concern on path planning and waypoints optimization. From comprehensive comparisons in Table III, we can observe that pursuing more

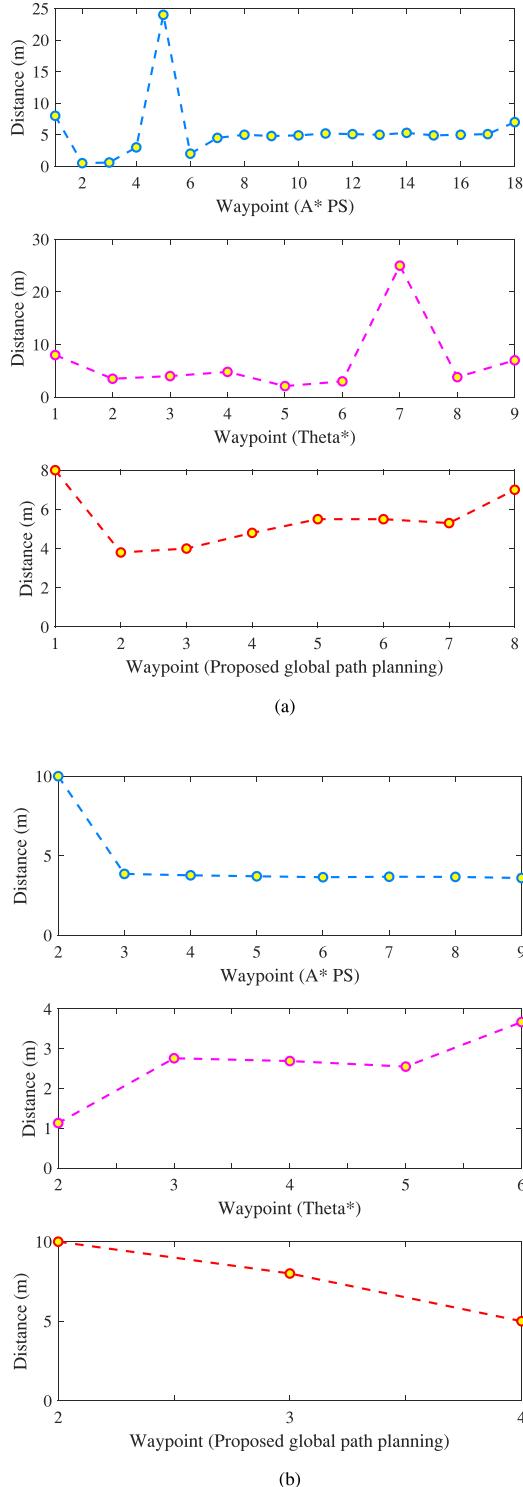


Fig. 7. Comparisons of distances to closet obstacles. (a) Linghai Harbor (b) Dayao Bay.

sparse waypoints, safer margins and shorter path length, simultaneously, the proposed global path planning scheme needs to speed 2%–8% more time than the Theta* algorithm, and takes nearly double time of the A* method. It should be clarified that the cost of 20 seconds is absolutely acceptable for an initial global path planning.

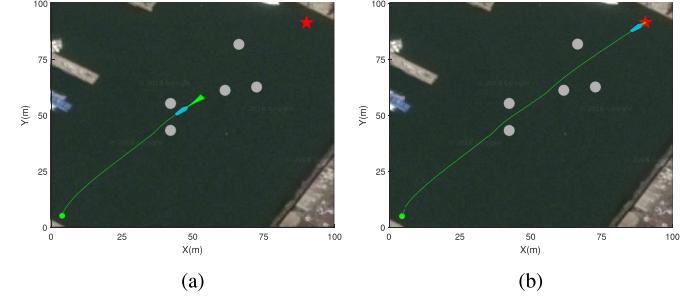


Fig. 8. Local path planning for static obstacles ($k_1 = 0.6$, $k_2 = 0.1$, $k_3 = 0.3$). (a) Avoiding obstacles. (b) Reaching the goal.

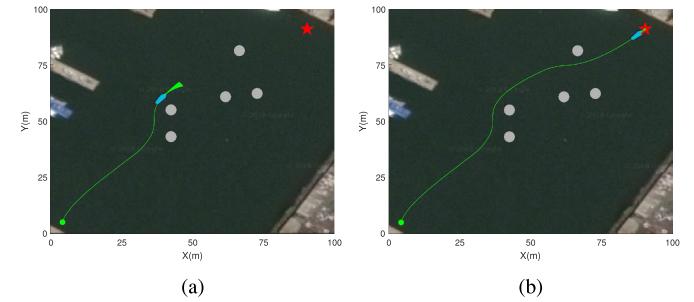


Fig. 9. Local path planning for static obstacles ($k_1 = 0.1$, $k_2 = 0.6$, $k_3 = 0.3$). (a) Avoiding obstacles. (b) Reaching the goal.

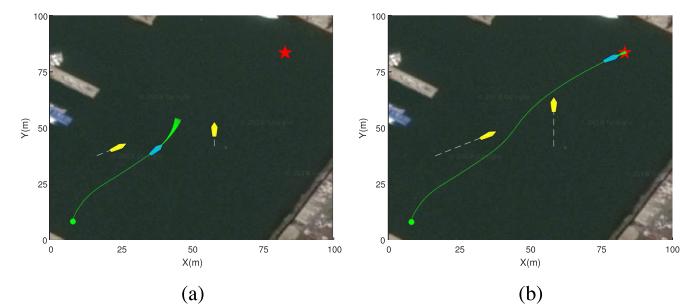


Fig. 10. Local planning for dynamic obstacles ($k_1 = 0.3$, $k_2 = 0.4$, $k_3 = 0.3$). (a) Avoiding obstacles. (b) Reaching the goal.

B. Local Path Planning

In this subsection, the proposed local path planning scheme using hierachial architecture (i.e., FDW and FDM) is evaluated by avoiding unknown static and dynamic obstacles. Using different weights k_1 , k_2 , and k_3 , simulation results on various scenarios are shown in Figs. 8–11. The results of avoiding unknown static obstacles are presented in Figs. 8 and 9, where static obstacles are marked by grey circles, while start and goal points are indicated by green point and red star, respectively. From Fig. 8, we can see that a local path avoiding static obstacles can be found, such that the ASV can radically sail through the gap between obstacles towards the goal with the aid of a higher heading weight, i.e., $k_1 = 0.6$. Alternatively, in Fig. 9, using smaller k_1 and larger k_2 results in a much more conservative local path which pursues much safer collision-avoidance. In this context, it can be derived

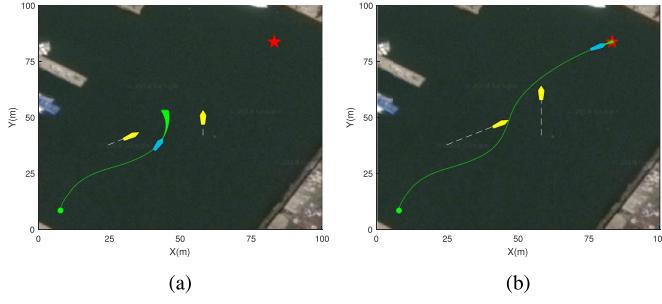


Fig. 11. Local planning for dynamic obstacles ($k_1 = 0.4, k_2 = 0.3, k_3 = 0.3$). (a) Avoiding obstacles. (b) Reaching the goal.

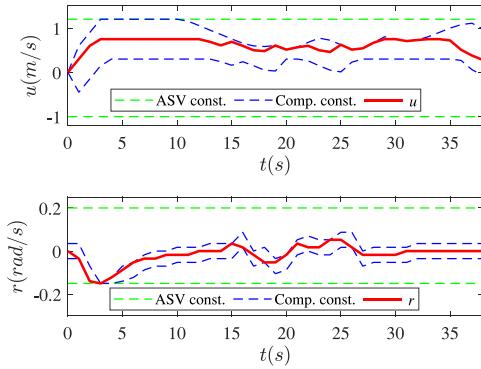


Fig. 12. Velocity guidance for static obstacles ($k_1 = 0.6, k_2 = 0.1, k_3 = 0.3$).

from Figs. 8–9 that the FDW and FDM modules can achieve effective collision-avoidance in static situations.

In order to further demonstrate the effectiveness in more complex environments, whereby moving obstacles are taken into account. As shown in Figs. 10–11, using different parameters, the resultant local path can sufficiently avoid moving vessels by virtue of slight or significant steering, respectively, which depends on weight selection in the FDW module. To be specific, in both static and dynamic environments, the alignment of the ASV with the goal is mainly determined by k_1 , while the distance between the ASV and the obstacle is critically affected by k_2 .

Accordingly, local collision-avoidance actions for surge u and yaw r are shown in Figs. 12–15, from which we can clearly see that all guidance signals of surge and yaw can sufficiently satisfy compounded constraints which accommodate the admissibility, reachability and performability. Meanwhile, as shown in Fig. 16, surge and yaw accelerations, i.e., \dot{u} and \dot{r} , are restricted within prescribed constraints. In addition, sway velocities v driven by surge u and yaw r are shown in Fig. 17, from which it can be seen that sway dynamics of the ASV are actually constrained by the estimated bounds.

Furthermore, considering constraints on ASV dynamics, in order to demonstrate the superiority of the FDW and FDM modules, comparisons to the fast marching method (FMM) are conducted in Fig. 18. Since the FMM method cannot accommodate complex constraints on the ASV, the path generated by the FMM, as shown in Fig. 18, would inevitably result in

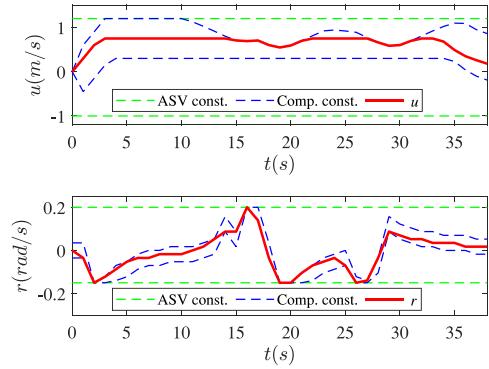


Fig. 13. Velocity guidance for static obstacles ($k_1 = 0.1, k_2 = 0.6, k_3 = 0.3$).

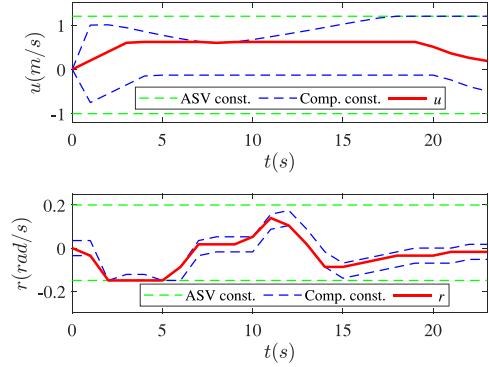


Fig. 14. Velocity guidance for dynamic obstacles ($k_1 = 0.3, k_2 = 0.4, k_3 = 0.3$).

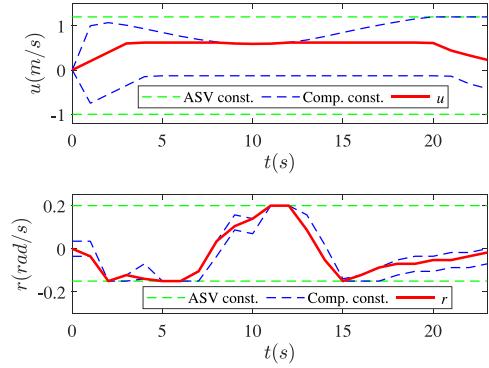


Fig. 15. Velocity guidance for dynamic obstacles ($k_1 = 0.4, k_2 = 0.3, k_3 = 0.3$).

sharp turns in front of obstacles, and thereby making the local path practically infeasible and/or hazardous. Promisingly, the FDW and FDM modules can collaboratively generate a much smoother path, as shown in Fig. 18, which effectively avoids unreasonable maneuvers pertaining to the path passing through obstacles, and thereby ensuring the local-path safety.

C. Hybrid Path Planning

The efficiency and superiority of the proposed DGL hybrid path planning scheme are eventually evaluated by considering both static and dynamic obstacles in real-world geographies of Linghai Harbor and Dayao Bay. Constraints on ASV motion

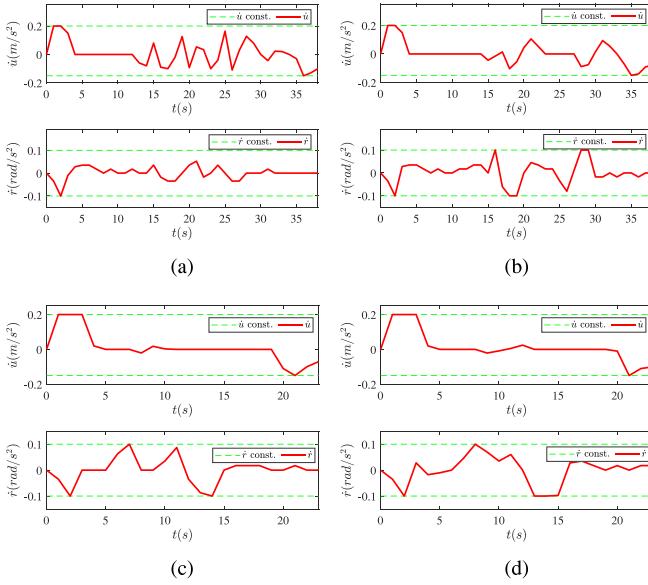


Fig. 16. Surge and yaw accelerations during local collision-avoidance. (a) Static obstacles (0.6, 0.1, 0.3). (b) Static obstacles (0.1, 0.6, 0.3). (c) Dynamic obstacles (0.3, 0.4, 0.3). (d) Dynamic obstacles (0.4, 0.3, 0.3).

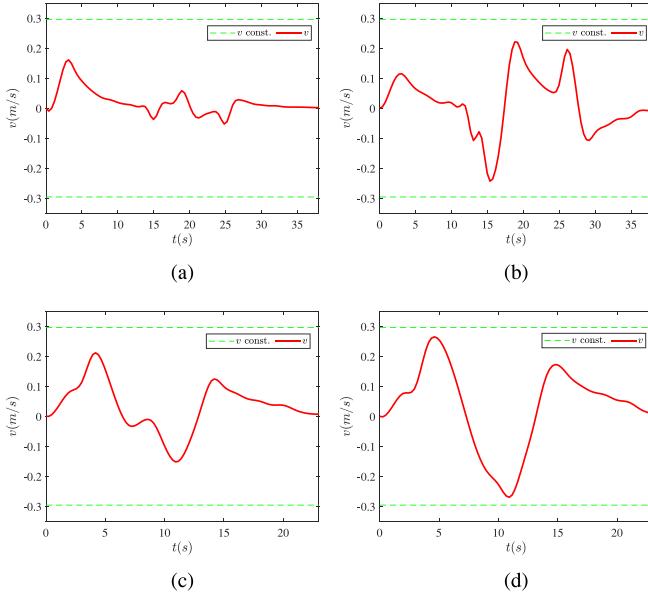


Fig. 17. Sway velocities during local collision-avoidance. (a) Static obstacles (0.6, 0.1, 0.3). (b) Static obstacles (0.1, 0.6, 0.3). (c) Dynamic obstacles (0.3, 0.4, 0.3). (d) Dynamic obstacles (0.4, 0.3, 0.3).

dynamics, as listed in Table II, are also addressed. To ensure much safer local collision-avoidance, the weights k_1 , k_2 and k_3 are set to be 0.1, 0.6, and 0.3, respectively. In the process of guiding the ASV towards successive waypoints and the target, the LOS guidance is deployed for path following, while the local hierarchical architecture including the FDM and FDW modules is executed to repeatedly plan locally dynamic path which avoids collision risk.

As shown in Fig. 19(a), within the DGL hybrid path, the local planning mechanism using the FDM and FDW modules

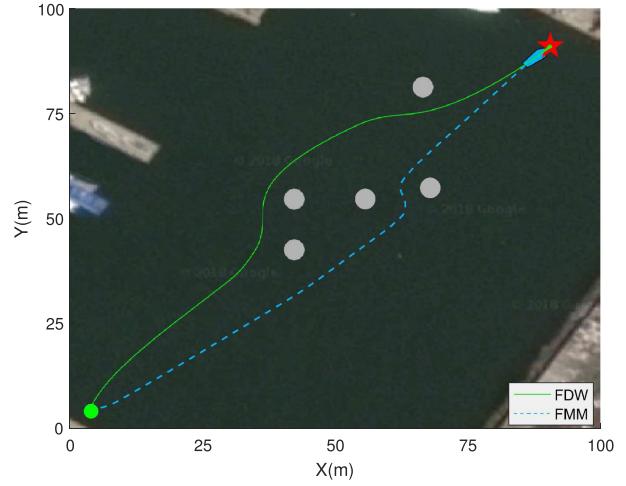


Fig. 18. Comparisons of the local path planning to the FMM.

renders the ASV reasonably sail around the stern of a moving vessel. Moreover, once obstacles enter into the FDW layer, a virtual waypoint (marked by a bright-blue star) is inserted onto the global path, such that the ASV can tactfully return to the backbone global-path. For a fair comparison, the Theta* combining with the FDW, i.e., Theta*-FDW, is deployed in Fig. 19(b), whereby all settings including start, target, static and dynamic obstacles are identical to those for the DGL. It is clear that the Theta*-FDW tends to execute offensive path-planning in both global and local sense, such that the ASV is forced to aggressively cross through the bow of a moving vessel. In essence, by virtue of SA extension, the DGL global path features sufficient security with respect to *a priori* geographical information. Moreover, locally dynamic modifications are driven by two-layer awareness, i.e., the FDM and FDW for large and close ranges, respectively. In this context, comparing to the Theta*-FDW, the DGL hybrid scheme performs significant preferability which can make a reasonable balance between length-optimality and path-security.

Within the Dayao Bay map, the superiority of the proposed DGL hybrid scheme is further illustrated in much more complicated scenarios including multiple static and dynamic obstacles which are unforeseen. Simulation results are shown in Figs. 20 and 21. For multi-static obstacles, as shown in Fig. 20, local actions of the DGL scheme are taken earlier than the Theta*-FDW, such that the local path keeps sufficiently large clearance to obstacles and smoothly returns to the global path, thereby avoiding close-quarters situations which might result from the Theta*-FDW. Encountering with a crossing vessel, with the aid of FDM module, the DGL scheme initially tries to make a significant steering such that the ASV safety area can be kept clean, while the Theta*-FDW can only work within close range for unforeseen obstacles and is apt to critically grab the front of moving obstacles especially in the middle of local path. Considering an even harsh situation whereby static-dynamic mixed obstacles are involved, as shown in Fig. 21, pertaining to the global path, the DGL scheme can locally tackle multiple static and dynamic obstacles, simultaneously, such that sufficient

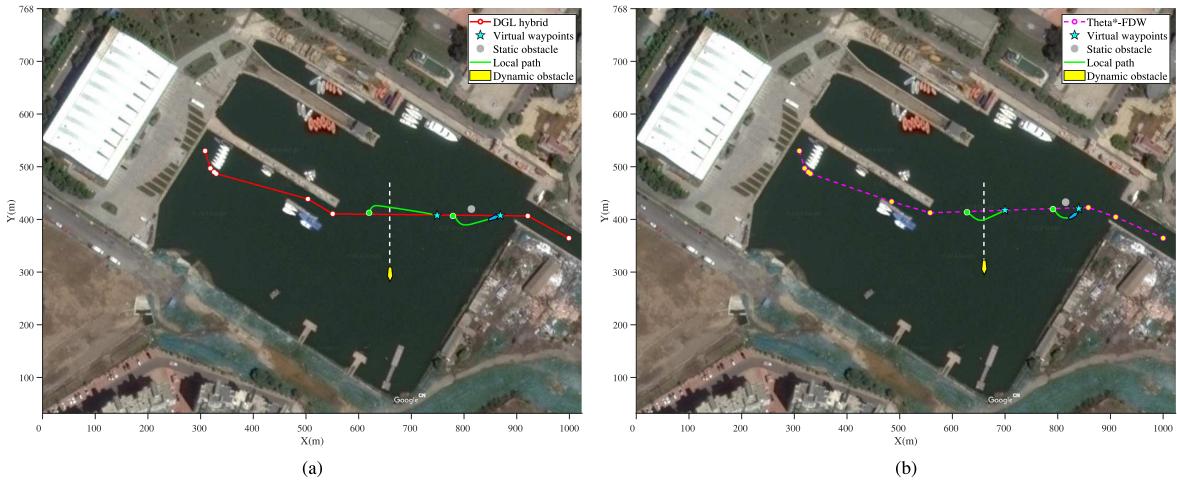


Fig. 19. Comparison of the DGL hybrid scheme to the Theta*-FDW (single-static and single-dynamic obstacles, Linghai Harbor, Dalian China). (a) The DGL hybrid path planning. (b) The Theta*-FDW path planning.

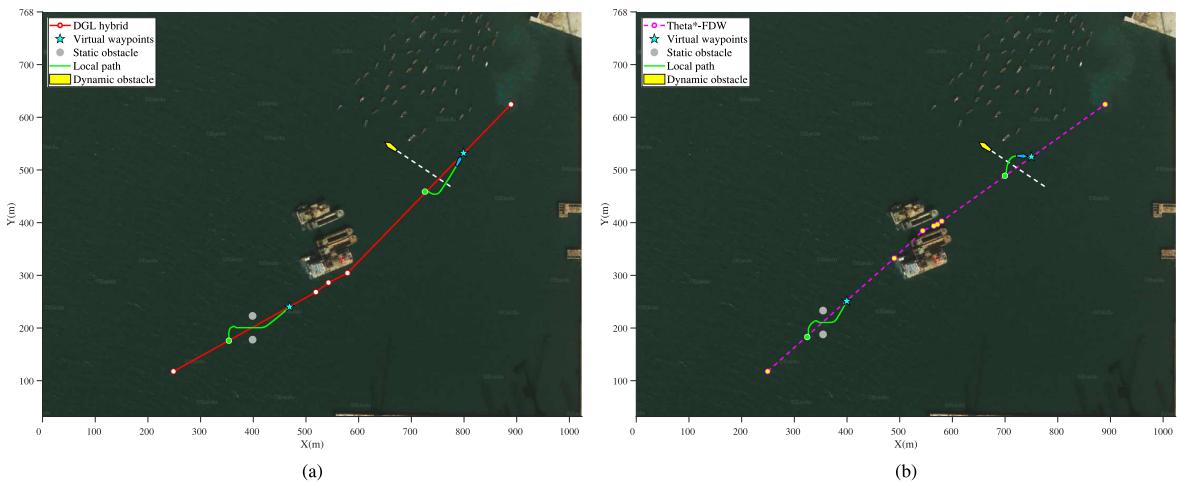


Fig. 20. Comparison of the DGL hybrid scheme to the Theta*-FDW (multi-static and single-dynamic obstacles, Dayao Bay, Dalian China). (a) The DGL hybrid path planning. (b) The Theta*-FDW path planning.

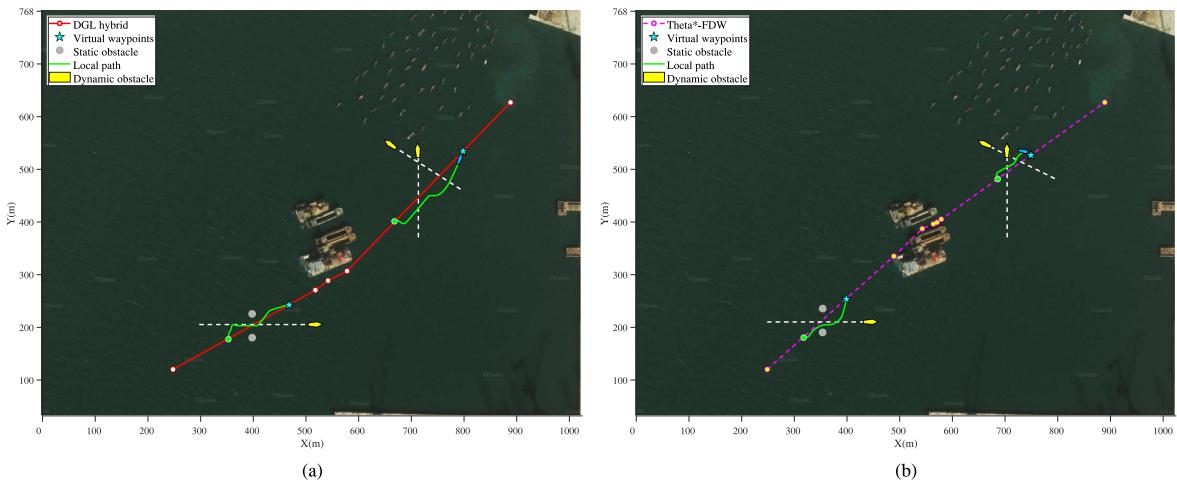


Fig. 21. Comparison of the DGL hybrid scheme to the Theta*-FDW (multiple static-dynamic obstacles, Dayao Bay, Dalian China). (a) The DGL hybrid path planning. (b) The Theta*-FDW path planning.

TABLE V
COMPARISONS OF GLOBAL-LOCAL PLANNING

Map	Obs. [†]	Alg.	Tot. Path (m)	Loc. Path (m)	Sta. Saf. Mar. (m)	Dyn. Saf. Mar. (m)	Onl. Cost [‡] (ms)
Linghai	SS-SD	DGL	762.01	237.32	5.79	12.77	127.62
		Theta*-FDW	764.62	143.94	2.51	8.65	121.73
Dayao	MS-SD	DGL	857.50	269.26	5.13	6.43	139.11
		Theta*-FDW	851.53	197.23	1.22	4.54	132.86
	MS-MD	DGL	862.21	358.36	5.13	5.69	155.47
		Theta*-FDW	847.97	217.25	1.22	2.89	146.79

[†]SS-SD: single-static & single-dynamic, MS-SD: multiple-static & single-dynamic, MS-MD: multiple-static & multiple-dynamic. [‡]Average time cost of each online-detection run within the local path planning.

safety margin of local path can be uniformly ensured. However, the Theta*-FDW local path leans too close to static obstacles due to avoidance of the moving vessel. Confronting multiple vessels on the voyage, the DGL scheme is able to execute clear collision-avoidance by effective steering, while the Theta*-FDW has to hesitate the surge and leads to an indeterminate local-path.

Furthermore, in order to make intensively quantitative comparisons between the DGL and Theta*-FDW schemes, concerns on actually total path length (Tot. Path), local path length (Loc. Path), static safety margin (Sta. Saf. Mar.), dynamic safety margin (Dyn. Saf. Mar.), and online-detection time cost (Onl. Cost) for each local decision run are summarized in Table IV. It can be observed that the DGL and Theta*-FDW approaches generate almost the same Tot. Path, while the Loc. Path is significantly diverse. Note that the Theta*-FDW scheme persistently search the shortest path in both global and local sense, while safety margins in static and dynamic environments, i.e., Sta. and Dyn. Saf. Mar., become rather congested. Preferably, the proposed DGL scheme can uniformly retain flexible safety areas in terms of both Sta. and Dyn. Saf. Mar., which are 2.3–4.2 and 1.5–2.0 times those of the Theta*-FDW, respectively, and thereby reasonably preserving 1.36–1.65 times Loc. Path of the Theta*-FDW. In terms of time cost for online decision, i.e., Onl. Cost in Table IV, the DGL scheme performs nearly same to the Theta*-FDW, i.e., 0.12s–0.16s for each detection run, and is absolutely acceptable for real-time operations.

Totally, from Figs. 19–21 and Table IV, the DGL hybrid path planning scheme enables the ASV retaining flexible safety margins to avoid both static and dynamic obstacles which are unpredictable. In this context, the proposed DGL hybrid path planning scheme combining with global path planning and local hierarchy can efficiently carry out off-line path planning and online modifications, simultaneously.

V. CONCLUSION

In this paper, in the presence of ASV dynamics, constraints, and unforeseen environments, a DGL hybrid path planning scheme combining with global path planning algorithm and local hierarchical architecture has been established. The proposed global path planning algorithm has generated an optimally safe path which can not only significantly reduce the number of waypoints but also shorten the path length. A local

hierarchical architecture consisting of FDM and FDW layers, which dynamically govern surge and yaw velocity guidance signals, has been built to interact with unknown environments during online collision-avoidance navigation. With the aid of virtual waypoints, global and local path planning algorithms have been seamlessly cohered into the entire DGL hybrid path planning scheme. Simulation studies and comprehensive comparisons in real-world geographies have been conducted to demonstrate the effectiveness and superiority of the proposed DGL hybrid path planning scheme.

ACKNOWLEDGMENT

The authors would like to thank the Editor-in-Chief, the Associate Editor, and anonymous reviewers for their invaluable suggestions and comments. The authors thank Mr. Yuncheng Gao for his supportive efforts on simulation debugging.

REFERENCES

- [1] S. L. Dai, S. He, H. Lin, and C. Wang, "Platoon formation control with prescribed performance guarantees for USVs," *IEEE Trans. Ind. Electron.*, vol. 65, no. 5, pp. 4237–4246, May 2018.
- [2] N. Wang, H. R. Karimi, H. Li, and S.-F. Su, "Accurate trajectory tracking of disturbed surface vehicles: A finite-time control approach," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 3, pp. 1064–1074, Mar. 2019.
- [3] N. Wang and Z. Deng, "Finite-time fault estimator based fault-tolerance control for a surface vehicle with input saturations," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 1172–1181, Feb. 2020.
- [4] H. Qin, H. Chen, Y. Sun, and L. Chen, "Distributed finite-time fault-tolerant containment control for multiple ocean bottom flying node systems with error constraints," *Ocean Eng.*, vol. 189, 2019, doi: [10.1016/j.oceaneng.2019.106341](https://doi.org/10.1016/j.oceaneng.2019.106341).
- [5] H. Qin, H. Chen, and Y. Sun, "Distributed finite-time fault-tolerant containment control for multiple ocean bottom flying nodes," *J. Franklin Inst.*, to be published, doi: [10.1016/j.jfranklin.2019.05.034](https://doi.org/10.1016/j.jfranklin.2019.05.034).
- [6] H. Qin, X. Yu, Z. Zhu, and Z. Deng, "An expectation-maximization based single-beacon underwater navigation method with unknown ESV," *Neurocomputing*, vol. 378, pp. 295–303, Feb. 2020. doi: [10.1016/j.neucom.2019.10.066](https://doi.org/10.1016/j.neucom.2019.10.066).
- [7] F. Liu, S. Gao, H. Han, Z. Tian, and P. Liu, "Interference reduction of high-energy noise for modal parameter identification of offshore wind turbines based on iterative signal extraction," *Ocean Eng.*, vol. 183, pp. 372–383, Jul. 2019.
- [8] F. Liu, Z. Tian, B. Wang, P. Liu, X. Cheng, and Z. Li, "A new residue-based dynamic analysis method for offshore structures with non-zero initial conditions," *Ocean Eng.*, vol. 162, pp. 138–149, Aug. 2018.
- [9] C. Yu, X. Xiang, L. Lapierre, and Q. Zhang, "Robust magnetic tracking of subsea cable by AUV in the presence of sensor noise and ocean currents," *IEEE J. Ocean. Eng.*, vol. 43, no. 2, pp. 311–322, Apr. 2018.
- [10] S. L. Dai, M. Wang, and C. Wang, "Neural learning control of marine surface vessels with guaranteed transient tracking performance," *IEEE Trans. Ind. Electron.*, vol. 63, no. 3, pp. 1717–1727, Mar. 2016.

- [11] N. Wang, G. Xie, X. Pan, and S.-F. Su, "Full-state regulation control of asymmetric underactuated surface vehicles," *IEEE Trans. Ind. Electron.*, vol. 66, no. 11, pp. 8741–8750, Nov. 2019.
- [12] N. Wang, S.-F. Su, X. Pan, X. Yu, and G. Xie, "Yaw-guided trajectory tracking control of an asymmetric underactuated surface vehicle," *IEEE Trans. Ind. Informat.*, vol. 15, no. 6, pp. 3502–3513, Jun. 2019.
- [13] S. Campbell, W. Naeem, and G. W. Irwin, "A review on improving the autonomy of unmanned surface vehicles through intelligent collision avoidance maneuvers," *Annu. Rev. Control*, vol. 36, no. 2, pp. 267–283, Dec. 2012.
- [14] H. Kim, D. Kim, J. U. Shin, H. Kim, and H. Myung, "Angular rate-constrained path planning algorithm for unmanned surface vehicles," *Ocean Eng.*, vol. 84, no. 4, pp. 37–44, Jul. 2014.
- [15] H. Kim, D. Kim, H. Kim, J. U. Shin, and H. Myung, "An extended any-angle path planning algorithm for maintaining formation of multi-agent jellyfish elimination robot system," *Int. J. Control Autom. Syst.*, vol. 14, no. 2, pp. 598–607, Apr. 2016.
- [16] N. Wang and S.-F. Su, "Finite-time unknown observer based interactive trajectory tracking control of asymmetric underactuated surface vehicles," *IEEE Trans. Control Syst. Technol.*, to be published, doi: 10.1109/TCST.2019.2955657.
- [17] Y. Cheng and W. Zhang, "Concise deep reinforcement learning obstacle avoidance for underactuated unmanned marine vessels," *Neurocomputing*, vol. 272, pp. 63–73, Jan. 2018.
- [18] A. Lazarowska, "A new deterministic approach in a decision support system for ship's trajectory planning," *Expert Syst. Appl.*, vol. 71, pp. 469–478, Apr. 2017.
- [19] M. DeLoura, *Game Programming Gems*, Charles River Media, Boston, MA, 2000.
- [20] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numer. Math.*, vol. 1, no. 1, pp. 269–271, Dec. 1959.
- [21] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, Jul. 1968.
- [22] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, Wiley, New York, 2011.
- [23] N. Wang and C. K. Ahn, "Hyperbolic-tangent LOS guidance-based finite-time path following of underactuated marine vehicles," *IEEE Trans. Industrial Electron.*, vol. 67, no. 10, pp. 8566–8575, Oct. 2020.
- [24] Z. Zheng, L. Sun, and L. Xie, "Error-constrained LOS path following of a surface vessel with actuator saturation and faults," *IEEE Trans. Syst. Man Cybern.*, vol. 48, no. 10, pp. 1794–1805, Oct. 2018.
- [25] N. Wang, Z. Sun, Y. Jiao, and G. Han, "Surge-heading guidance based finite-time path-following of underactuated marine vehicles," *IEEE Trans. Veh. Technol.*, vol. 68, no. 9, pp. 8523–8532, Sep. 2019.
- [26] N. Wang and H. R. Karimi, "Successive waypoints tracking of an underactuated surface vehicle," *IEEE Trans. Ind. Informat.*, vol. 16, no. 2, pp. 898–908, Feb. 2020.
- [27] A. Stentz, "Path relaxation: Path planning for a mobile robot," in *Proc. IEEE OCEANS Conf.*, 1984, pp. 576–581.
- [28] K. Daniel, A. Nash, S. Koenig, and A. Felner, "Theta*: Any-angle path planning on grids," *J. Artif. Intell. Res.*, vol. 39, no. 1, pp. 533–579, Oct. 2010.
- [29] A. Nash, K. Daniel, and S. Koenig, "Theta*: Any-angle path planning on grids," in *Proc. AAAI Conf. Artif. Intell.*, 2007, pp. 1177–1183.
- [30] A. Nash, S. Koenig, and T. Craig, "Lazy Theta*: Any-angle path planning and path length analysis in 3D," in *Proc. AAAI Conf. Artif. Intell.*, 2010, pp. 147–154.
- [31] A. Stentz, "The focussed D* algorithm for real-time replanning," in *Proc. 14th Int. Joint Conf. Artif. Intell.*, 1995, pp. 1652–1659.
- [32] M. Likhachev and S. Koenig, "A generalized framework for lifelong planning A* search," in *Proc. 5th Int. Conf. Automat. Plan. Schedul.*, 2005, pp. 99–108.
- [33] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 354–363, Jun. 2005.
- [34] D. Ferguson and A. Stentz, "Using interpolation to improve path planning: The Field D* algorithm," *J. Field Robot.*, vol. 23, no. 2, pp. 79–101, Mar. 2006.
- [35] D. Sislak, P. Volf and M. Pechoucek, "Accelerated A* path planning," in *Proc. AAMAS Conf. Auton. Agent. Multiagent Syst.*, 2009, pp. 1133–1134.
- [36] P. Yap, N. Burch, R. C. Holte and J. Schaeffer, "Block A*: Database-driven search with applications in any-angle path-planning," in *Proc. AAAI Conf. Artif. Intell.*, 2011, pp. 120–125.
- [37] T. L. Perez and M. A. Wesley, "An algorithm for planning collision-free paths among polyhedral obstacles," *Communicat. ACM*, vol. 22, no. 10, pp. 560–570, Oct. 1979.
- [38] V. Kanakakis, and N. Tsourveloudis, "Evolutionary path planning and navigation of autonomous underwater vehicles," in *Proc. Mediterranean Conf. Control Autom.*, 2007, pp. 1–6.
- [39] J. Li, G. Deng, C. Luo, Q. Lin, Q. Yan, and Z. Ming, "A hybrid path planning method in unmanned air/ground vehicle (UAV/UGV) cooperative systems," *IEEE Trans. Veh. Technol.*, vol. 65, no. 12, pp. 9585–9596, Dec. 2016.
- [40] A. Alvarez, A. Caiati, and R. Onken, "Evolutionary path planning for autonomous underwater vehicles in a variable ocean," *IEEE J. Ocean. Eng.*, vol. 29, no. 2, pp. 418–429, Apr. 2004.
- [41] V. P. Vinay and R. Sridharan, "Development and analysis of heuristic algorithms for a two-stage supply chain allocation problem with a fixed transportation cost," *Int. J. Serv. Operat. Manag.*, vol. 12, no. 2, pp. 244–268, 2012.
- [42] X. Chen, Y. Kong, X. Fang, and Q. Wu, "A fast two-stage ACO algorithm for robotic path planning," *Neural Comput. Appl.*, vol. 22, no. 2, pp. 313–319, Feb. 2013.
- [43] V. Roberge, M. Tarbouchi, and G. Labont, "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning," *IEEE Trans. Ind. Informat.*, vol. 9, no. 1, pp. 132–141, Feb. 2013.
- [44] O. Khatib, "Real-time obstacle avoidance for manipulators and mobile robots," *Int. J. Robot. Res.*, vol. 5, no. 1, pp. 90–98, 1986.
- [45] Z. Pan, J. Li, K. Hu, and H. Zhu, "Intelligent vehicle path planning based on improved artificial potential field method," *Appli. Mech. Mater.*, vol. 742, pp. 349–354, Mar. 2015.
- [46] I. Kamon, E. Rivlin, and E. Rimon, "A new range-sensor based globally convergent navigation algorithm for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2002, pp. 429–435.
- [47] J. Ng and T. Unl, "Performance comparison of bug navigation algorithms," *J. Intell. Robot. Syst.*, vol. 50, no. 1, pp. 73–84, Sep. 2007.
- [48] I. Kamon, E. Rimon, and E. Rivlin, "TangentBug: A range-sensor-based navigation algorithm," *Int. J. Robot. Res.*, vol. 17, no. 9, pp. 934–953, Sep. 1998.
- [49] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Automat.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.
- [50] I. Ulrich and J. Borenstein, "VFH+: Reliable obstacle avoidance for fast mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, 1998, pp. 1572–1577.
- [51] J. A. Sethian, "A fast marching level set method for monotonically advancing fronts," *Proc. Nati. Acad. Sci. USA*, vol. 93, no. 4, pp. 1591–1595, Feb. 1996.
- [52] S. M. Zadeh, D. M. W. Powers, and K. Sammut, "An autonomous reactive architecture for efficient AUV mission time management in realistic dynamic ocean environment," *Robot. Autonom. Syst.*, vol. 87, pp. 81–103, Jan. 2017.
- [53] N. Wang and X. Pan, "Path-following of autonomous underactuated ships: A translation-rotation cascade control approach," *IEEE/ASME Trans. Mechatron.*, vol. 24, no. 6, pp. 2583–2593, Dec. 2019.
- [54] N. Wang, Z. Sun, J. Yin, Z. Zou, and S.-F. Su, "Fuzzy unknown observer-based robust adaptive path following control of underactuated surface vehicles subject to multiple unknowns," *Ocean Eng.*, vol. 176, pp. 57–64, Mar. 2019.
- [55] Y. Weng, N. Wang, and C. G. Soares, "Data-driven sideslip observer-based adaptive sliding-mode path-following control of underactuated marine vessels," *Ocean Eng.*, vol. 197, pp. 106910, Feb. 2020.
- [56] H. K. Khalil, *Nonlinear Systems*, 3rd ed., Prentice-Hall Press, Upper Saddle River, NJ, USA, 2002.
- [57] N. Wang, "A novel analytical framework for dynamic quaternion ship domains," *J. Navigation*, vol. 66, no. 2, pp. 265–281, Mar. 2013.
- [58] N. Wang, X. Jin, and M. J. Er, "A multilayer path planner for a USV under complex marine environments," *Ocean Eng.*, vol. 184, pp. 1–10, Jul. 2019.
- [59] N. Wang, C. Qian, J.-C. Sun, and Y.-C. Liu, "Adaptive robust finite-time trajectory tracking control of fully actuated marine surface vehicles," *IEEE Trans. Control Syst. Technol.*, vol. 24, no. 4, pp. 1454–1462, Apr. 2016.
- [60] D. Fox, W. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," *IEEE Robot. Automat. Mag.*, vol. 4, no. 1, pp. 23–33, Mar. 1997.



Ning Wang (Senior Member, IEEE) received the B.Eng. degree in marine engineering and the Ph.D. degree in control theory and engineering from the Dalian Maritime University, Dalian, China, in 2004 and 2009, respectively. From September 2008 to September 2009, he was financially supported by China Scholarship Council to work as a Joint-Training Ph.D. Student at the Nanyang Technological University (NTU), Singapore. In view of his significant research at NTU, he received the Excellent Government-funded Scholars and Students Award in 2009. From August 2014 to August 2015, he worked as a Visiting Scholar at the University of Texas at San Antonio. His research interests include self-learning modeling and control, unmanned (marine) vehicles, machine learning and autonomous systems. Dr. Wang received the Nomination Award of Liaoning Province Excellent Doctoral Dissertation, and also won the State Oceanic Administration Outstanding Young Scientists in Marine Science and Technology, the China Ocean Engineering Science and Technology Award (First Prize), the Liaoning Province Award for Technological Invention (First Prize), the Liaoning Province Award for Natural Science (Second Prize), the Liaoning Youth Science and Technology Award (Top10 Talents), the Liaoning BaiQianWan Talents (First Level), the Liaoning Excellent Talents (First Level), the Science and Technology Talents the Ministry of Transport of the P. R. China, the Youth Science and Technology Award of China Institute of Navigation, and the Dalian Leading Talents. He has authored 3 books, and more than 100 SCI-indexed journal papers. He currently serves as members of IEEE TC on Industrial Informatics, the American Society of Mechanical Engineers (ASME), the Society of Naval Architects and Marine Engineers (SNAME), the Chinese Association of Automation (CAA), and the China's Society of Naval Architecture and Marine Engineering (CSNAME). He has been Leading Guest Editors of the *Neurocomputing*, the *International Journal of Fuzzy Systems*, the *International Journal of Advanced Robotics System*, and the *Advances in Mechanical Engineering*. He currently serves as Associate Editors of the *International Journal of Fuzzy Systems*, the *Electronics (MDPI)*, the *Journal of Electrical Engineering & Technology*, and the *Cyber-Physical Systems (Taylor-Francis)*.



Hongwei Xu received the B.E. degree in marine engineering from the Guangdong Ocean University, Zhanjiang, China, in 2017. He is currently working toward the master's degree with Dalian Maritime University, Dalian, China. His research interests include underactuated marine vehicles, path planning and deep reinforcement learning.