

Article

Iterative Learning-Based Path and Speed Profile Optimization for an Unmanned Surface Vehicle

Yang Yang ¹, Quan Li ¹, Junnan Zhang ¹ and Yangmin Xie ^{2,*}

¹ Research Institute of USV Engineering, Shanghai University, Shanghai 200444, China; yangyang_shu@shu.edu.cn (Y.Y.); shu_liquan@shu.edu.cn (Q.L.); zjnshu@shu.edu.cn (J.Z.)

² Shanghai Key Laboratory of Intelligent Manufacturing and Robotics, Shanghai University, Shanghai 200444, China

* Correspondence: xieym@shu.edu.cn; Tel.: +86-21-5633-7148

Received: 4 December 2019; Accepted: 7 January 2020; Published: 13 January 2020



Abstract: Most path-planning algorithms can generate a reasonable path by considering the kinematic characteristics of the vehicles and the obstacles in hydrographic survey activities. However, few studies consider the influence of vehicle dynamics, although excluding system dynamics may considerably damage the measurement accuracy especially when turning at high speed. In this study, an adaptive iterative learning algorithm is proposed to optimize the turning parameters, which accounts for the dynamic characteristics of unmanned surface vehicles (USVs). The resulting optimal turning radius and speed are used to generate the path and speed profiles. The simulation results show that the proposed path-smoothing and speed profile design algorithms can largely increase the path-following performance, which potentially can help to improve the measurement accuracy of various activities.

Keywords: USV; iterative parameter-tuning; path-smoothing; speed profile design

1. Introduction

Unmanned surface vehicles (USVs), also called autonomous surface craft (ASC), are self-driving marine vessels widely used in environmental monitoring and hydrographic survey [1]. Path-following performance of the USVs is one of the main factors influencing the measurement accuracy in hydrographic survey activities [2].

A reliable autonomous navigation system with the capabilities of path-planning, obstacle avoidance, and auto-guidance is essential for an USV to deal with various and dynamic marine situations [3]. The most widely used obstacle avoidance and path-planning algorithms include artificial potential fields (APF) [4], rapidly exploring random trees (RRTs) [5], greedy mechanism-based particle swarm optimization (PSO) [6], and grid map-based path-planning algorithms (e.g., A* algorithm [7], Field D* [8], and Theta* [9]). These algorithms focus on path-planning for obstacle avoidance considering the kinematics of the vehicle and the obstacles.

By using the abovementioned techniques, a segmented, first-order discontinuous line-shape path is obtained, as shown in Figure 1a. The discontinuity does not considerably affect the path-following performance when the USV drives at a relatively low speed but can induce large differences between the planned and achieved paths when the speed is high and the dynamics of the vehicle become non-neglectable. The deviation causes the vehicle to fail the obstacle avoidance task when the vehicle is required to make sharp turns for non-holonomic systems, such as USVs [10,11] and, thus, should be diminished. Typically, there are two solutions. The first solution is to improve the mechanical and/or control system so that the vehicle itself could be sufficiently agile to make accurate and shape turns. The second solution is to modify the path and the speed profiles in Figure 1b so that they are more

suitable for the USV to follow. In Figure 1b, the path is bent to arcs at the sharp corners and better fits the dynamic characteristics of the USV.

In this study, we focus on the work of the latter solution. Specifically, by assuming that the physical system and motion control of the USV is already provided, we study how the path and speed profiles can be optimized so that the overall performance of path-following and obstacle avoidance can be improved. There are usually two approaches to solve such a motion planning problem [12]: (1) trajectory planning with a simultaneous optimization of path and speed profiles; (2) geometric path-planning followed by velocity planning, which is also known as path-velocity decomposition. The latter is easier to implement in real time where path-planning usually considers geometric continuity and velocity planning addresses the kinematic constraints.

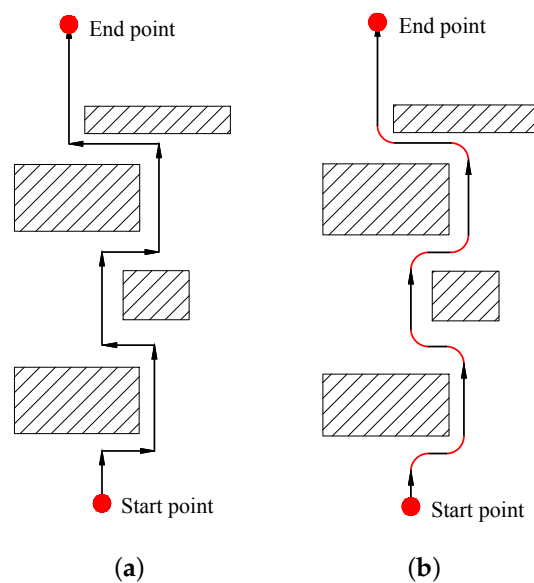


Figure 1. Original path and smooth path profiles: (a) Original path. (b) Smooth path.

To achieve path smoothness, the Dubins curve [13] is a good choice. It is created by concatenating line segments and arcs of circles. The Dubins curve is first-order continuous and thus smoother than the original line-based path. In addition, it is computationally cost efficient, which makes it suitable for real-time path-planning cases. The maximum curvature, or correspondingly, the minimum turning radius of the system, is used as a constraint to generate the optimal Dubins curve [14,15]. To some extent, this approach considers the dynamic properties of the vehicle. However, because the minimum turning radius is closely related to the vehicle speed, the isolation of path and speed profile design may compromise the overall path-following performance. There are other studies that use second-order continuous paths such as cubic spirals [16], Bezier splines [10], clothoidal spirals [17,18], quintic polynomials [19], polar splines [20] and cardioids [21]. These approaches may further smooth the path but also make the computation more complex and time consuming. Therefore, the decision of whether to adopt first-order or second-order curves should be balanced by considering the mission demands on path continuity and real-time ability.

For speed-planning strategies, researchers usually set up a time-optimal problem with constraints from the kinematic properties of the vehicles including maximum speed, longitudinal and lateral acceleration and jerk [22]. The kinematic model of the vehicle is established, and the key design parameters regarding the velocity profile are specified to form a numerical optimization problem [23], the solution of which defines the final speed profile. In some cases, the experimental choice of the speed is used to define a speed profile when the time-optimality is not an essential requirement [24,25].

There are also studies that take into consideration the kinematic feasibility, and model predictive control is introduced to generate a speed profile based on the predicted performance [18].

However, few studies mention the consideration of vehicle dynamics for either path-planning or speed-planning though the exclusion of the dynamics of the system may largely compromise the path tracking performance especially at high speed turning. There are mainly two reasons for this. First, the use of a dynamic model may increase the complexity in the formulation and computation of the optimization problem. Second, in most cases, an accurate dynamical model is not easy to obtain. Therefore, this study realizes a practical and real-time solution for the path and speed profile generation by considering the influence of vehicle dynamics on path-following performance. To achieve this goal, the schematic of the path and speed planner is shown in Figure 2. The offline learning block uses experimental data to obtain an optimal turning radius and speed pair for the vehicle, which provides design constraints for the smooth path-following and speed profile optimization. Thus, the generated path and speed profile are customized for the specific vehicle according to its dynamical properties.

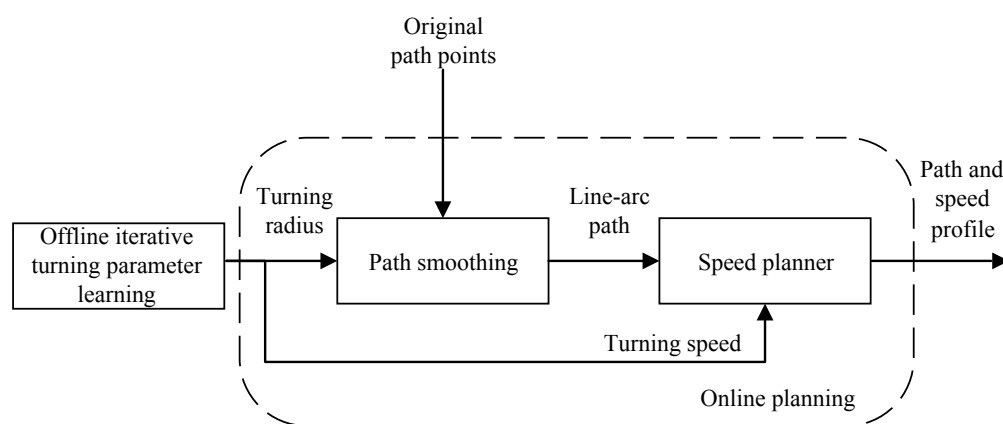


Figure 2. Path and speed-planning scheme

In summary, previous studies usually emphasized the use of methods that generate smooth and optimal path or/and speed profiles according to their geometric or kinematic properties. However, this study provides a practical and experiment-based approach that takes the USV dynamics into consideration by:

1. Determining an optimal arc radius and the corresponding turning speed using an offline iteratively tuning method.
2. Generating a smooth trajectory with a provided arc radius and ensuring small path deviation from the original line segments.
3. Planning optimal speed profile on the line-arc combined trajectory using step response curves of the USV.

The remainder of the paper is structured as follows. Section 2 presents the iterative tuning method and its application in determining an optimal arc radius and corresponding turning speed. Then, the method of path-smoothing with fixed turning radius is introduced in Section 3, followed by an optimal speed-planning strategy in Section 4. Section 5 shows the simulation verification. Finally, Section 6 presents concluding remarks.

2. Iterative Learning Method for Path Parameter-Tuning

2.1. Problem Setup

Iterative learning is broadly used to deal with repetitive tasks by updating control signals or system parameters. It learns the system behavior via repeated tests which is possible to achieve

optimal performance without providing an explicit system model [24]. Iterative learning sometimes is called a data-driven approach [26]. When the control signals are updated iteration by iteration, it is called iterative learning control (ILC) [27]; if the parameters in the systems are updated during the process, it is called iterative parameter-tuning (IPT) [28]. In this paper, IPT is used to obtain a suitable combination of the references of the turning radius and speed for a specific USV to decrease the path-following error with limited sacrifice of turning speed and turning radius. Generally speaking, small turning radius and large turning speed induce large path deviation. With proper selection on the weight factors for the three aforementioned terms, IPT can provide an optimized solution with balanced path-following performance.

The parameter vector used for tuning is $\mathbf{x} = (r_r, v_r)$, where r_r is the reference of the turning radius and v_r is the reference of the forwarding speed at the turning corner. The cost function is defined as

$$J(\mathbf{x}) = \lambda_1(r_r - r_{\min})^2 + \lambda_2(v_r - v_d)^2 + \lambda_3 \frac{1}{n} \sum_{i=1}^n \|\bar{p}_i - p_i\| \quad (1)$$

where r_{\min} is the empirical value that defines the minimum turning radius of USV when it is within its normal speed range; v_d is also provided by the user and specifies the desired speed that the USV will drive at, which is generally chosen as a relatively high speed value to speed up the turning action; p_i is the reference trajectory vector generated by r_r and v_r ; \bar{p}_i is the actual trajectory vector; and λ_1 , λ_2 and λ_3 are the weight factors. The first term in J represents the difference between the reference radius with the minimum turning radius; the second term evaluates the difference between the reference speed with the desired speed; and the last terms show the tracking error under the current setup of r_r and v_r . By choosing different λ_1 , λ_2 and λ_3 , the user can specify the respective emphasis on the three abovementioned aspects.

2.2. Two-Point Step-Size Adaptation (TPA)-IPT Algorithm

The goal of IPT is to obtain the smallest J by iteratively tuning the parameter of r_r and v_r , which can be treated as a constrained optimization problem below. Similar to r_{\min} , r_{\max} , v_{\min} and v_{\max} are also user-specified parameters based on the USV system capabilities.

$$\begin{aligned} \min J(r_r, v_r) \\ r_{\min} < r_r < r_{\max} \\ v_{\min} < v_r < v_{\max} \end{aligned} \quad (2)$$

In this study, the line search method schematic [29] is adopted to identify an optimal J . The updating law of the path parameter is written as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad (3)$$

where α_k is the step length, and \mathbf{p}_k is the search direction. In general, \mathbf{p}_k is a descent direction and frequently has the form of

$$\mathbf{p}_k = -\mathbf{B}_k^{-1} \nabla J_k \quad (4)$$

where \mathbf{B}_k^{-1} is the symmetric and nonsingular matrix, and ∇J_k is the gradient. When $\mathbf{B}_k^{-1} = \mathbf{I}$, where \mathbf{I} is the identity matrix, it is the fastest descent direction. The selection of the step length α_k is highly related to the gradient ∇J_k , and these parameters must satisfy $|1 - \alpha_k \nabla J_k^T| \leq \rho < 1$ to guarantee stability, where ρ is a positive scalar. If ∇J_k is known, the updating gains can be chosen as $\mathbf{p}_k = -\nabla J_k$ and $\alpha_k = \nabla J_k^{-1}$. In reality, J is an unknown nonlinear function of \mathbf{x} . Therefore, an additional algorithm needs to be developed to identify steady descent directions and step length, which is not a trivial task.

Particularly for the tuning task in this study, the dimension of the variable space is as low as 2, which make it possible to estimate a descent direction with a limited number of tests. Taking the k^{th}

iteration as an example, the procedure starts at a given point \mathbf{x} . Four trials provide an estimation of the gradient direction at \mathbf{x} by

$$\mathbf{y}_{kj} = \mathbf{x}_k + \Delta_{kj} \quad (5)$$

where j is the trial index number with $\Delta_{k1} = (+\mu\alpha_k, 0)$, $\Delta_{k2} = (-\mu\alpha_k, 0)$, $\Delta_{k3} = (0, +\alpha_k)$, $\Delta_{k4} = (0, -\alpha_k)$. Where $\mu = \frac{r_{\max} - r_{\min}}{v_{\max} - v_{\min}}$ is the normalized coefficient. α_k represents the step size of the $(k-1)^{th}$ parameter update, and it is the step length of the previous iterations for the purpose of saving parameter-tuning trials. The calculation of an adaptive step length will be further explained later. Based on the test results, the cost function values can be obtained, and the unit gradient vector \mathbf{e}_k is numerically estimated from the previous two iterations by using Equations (6) and (7).

$$\mathbf{D}_k = \sum_{j=1}^4 |J(\mathbf{y}_j) - J(\mathbf{x}_k)| \frac{J(\mathbf{y}_j) - J(\mathbf{x}_k)}{\mathbf{y}_j - \mathbf{x}_k} \quad (6)$$

$$\mathbf{e}_k = \frac{\mathbf{D}_k}{\|\mathbf{D}_k\|} \quad (7)$$

The second step to update \mathbf{x} is to determine the line search step length. Considering the cost of conducting many tests, it is essential for the tuning process to converge as fast as possible to be applicable to real systems. Therefore, a large step size is preferred for the IPT algorithm to speed up the convergence and reduce the needed number of trials. However, convergence occurs only if the step length is sufficiently small [30]. Therefore, a rule that can dynamically adapt the step size among iterations needs to be developed.

Self-adaptive algorithms for the step length are not a new concept in numerical optimization. A typical method is called Two-Point Step-Size Adaptation (TPA), where only two trial points are needed to determine the change of the step length between iterations. This method works well with the general optimization setups such as the steepest descent method [30] and the evolutionary gradient search [31]. Another method combines the adjustment strategy of the step length with the evolution optimization method [32], where the step length is calculated by the weighted distance sum of randomly generated offspring with descending costs. To make the algorithm robust enough, a sufficient number of offspring is required. A cumulative step-size adaptation (CSA) method is further developed to better remove the randomness of evolution strategies [33]. The step length is determined by historical records and is optimized accordingly. However, the authors admit its sensitivity to the sampling procedure and replace it with TPA in their future work [34].

Based on the short review of the abovementioned self-adaptive step length strategies, it is clear that TPA stands out for its simplicity and widely proven robustness. TPA is suitable for the IPT framework because only two trials are required for the step length adaptation in each iteration. Therefore, in this study, we propose to merge the TPA method with the tuning framework to obtain the TPA-IPT algorithm, as shown in Algorithm 1. In addition, the tuned step-size updates the parameters and functions as the off-distance of the next trial points, which is similar to [32].

For each iteration, extra two trials are proceeded by two different step lengths, $\alpha_k \zeta$ and α_k / ζ , where $\zeta \in (0, 1)$ is a user-tuned number. The comparison of the two trials in terms of their cost function values provides a superior choice between the two for the step-size update, as shown in (8).

$$\alpha_{k+1} = \begin{cases} \alpha_k \zeta, & J(\mathbf{x}_k - \alpha_k \zeta \mathbf{e}_k) \leq J(\mathbf{x}_k - \alpha_k / \zeta \mathbf{e}_k) \\ \alpha_k / \zeta, & J(\mathbf{x}_k - \alpha_k \zeta \mathbf{e}_k) \geq J(\mathbf{x}_k - \alpha_k / \zeta \mathbf{e}_k) \end{cases} \quad (8)$$

Thus, the updating rule of the path parameters is obtained as

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_{k+1} \mathbf{e}_k \quad (9)$$

When the cost function value starts to increase as $J(x_{k+1}) \geq J(x_k)$, the variables are close to the minimum point. Then, the abovementioned TPA rule can induce vibrations around the minimum and slow down the convergence. Therefore, the dichotomy is adopted to quickly decrease the step size by imposing $\alpha_{k+1}^* = \frac{1}{2^n} \alpha_k$, where n is the smallest positive integer that satisfies this condition $J(x_{k+1}^*) < J(x_k)$.

The abovementioned optimization process is terminated when $|x_{k+1}^* - x_k| \leq \psi$, where ψ is a user-specified positive value.

Algorithm 1: TPA-IPT.

Given $\lambda_1, \lambda_2, \lambda_3, r_{\min}, v_{\min}, v_d, \zeta, \psi, n \in N_+, k \in N_+$

Initialize x_0, α_0 ,

Calculate cost function $J(x_0)$

while 1 **do**

 Flag=0

 Estimated gradient \mathbf{e}_k

if $J(\mathbf{x}_k - \alpha_k \zeta \mathbf{e}_k) \leq J(\mathbf{x}_k - \alpha_k / \zeta \mathbf{e}_k)$ **then**

$\alpha_{k+1} = \alpha_k \zeta$

else

$\alpha_{k+1} = \alpha_k / \zeta$

end

 Update $x_{k+1}, J(x_{k+1})$

while $J(x_{k+1}) \geq J(x_k)$ **do**

$\alpha_{k+1}^* = \frac{1}{2^n} \alpha_k$

$\mathbf{x}_{k+1}^* = \mathbf{x}_k - \alpha_{k+1}^* \mathbf{e}_k$

 Update $J(x_{k+1}^*)$

$J(x_{k+1}) = J(x_{k+1}^*)$

 Flag=1

$n = n + 1$

end

if Flag=1 and $|x_{k+1}^* - x_k| \leq \psi$ **then**

 return x_{k+1}^*

break

end

$k = k + 1$

end

3. Path-Smoothing Algorithm

The paths generated by the path-planning algorithm usually consist of line sets connected by first-order discontinuous corners. In general, due to their dynamics at high speed, the USV systems cannot follow the path accurately at the corners, and the offset of the real trajectory from the planned one sometimes causes problems such as obstacle collision or undesirable path-following performance. To modify the original non-smooth path to be more suitable for a USV to follow, the optimized tuning radius determined by the IPT algorithm in Section 2 is used for path-smoothing in this section.

The path-smoothing problem is set up as follows. Given the start point w_1 and the target point w_3 , the path-smoothing problem is defined to find a path w such that:

1. w is based on the original line-segment path and satisfies the dynamic characteristics and security requirements such as the curvature (turning radius) limitation.
2. w begins at the starting point w_1 and ends at the target point w_3 .
3. w does not collide with the obstacles.
4. The path deviation De between the smoothed path and the original path should be small.

The Dubins curves are adopted to smooth the path with slight improvement compared to the previously used approaches since they require a low computation cost and are easy to apply. In a previous study, typically one arc or two arcs are used to smooth the corners [14,15,35], as shown in Figure 3. The black solid lines represent the original path, and the red solid lines represent the smoothed path. Break points usually imply nearby obstacles, and the deviation from the original path increases the risk of collisions between obstacles and the smoothed path.

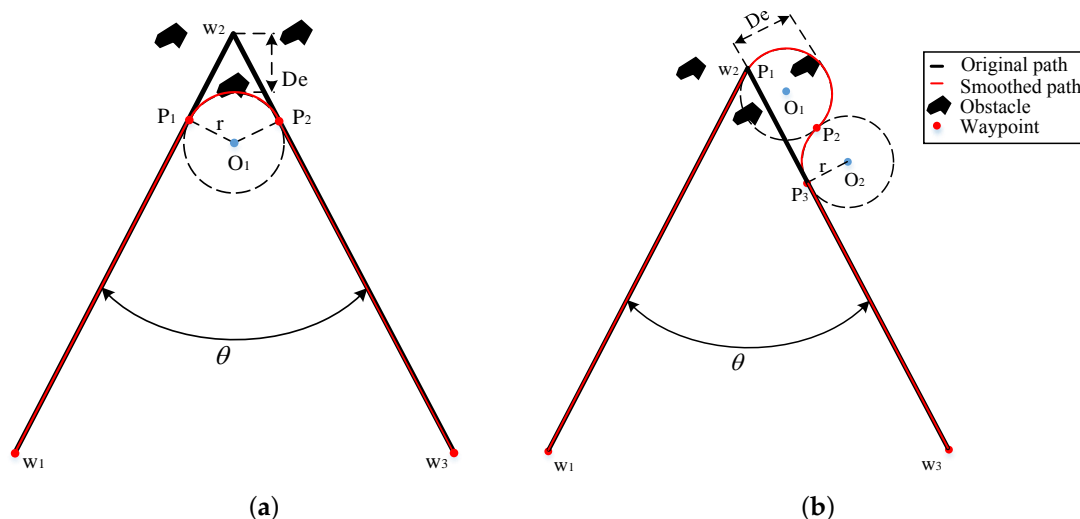


Figure 3. Smoothed paths: (a) Created by using one Dubins curve. (b) Created by using two Dubins curves.

In this study, to decrease the path deviation and alleviate the collision risk, we propose to use three Dubins curves for every break point, which is named the L-3C-L method, as shown in Figure 4. w_1, w_2 and w_3 are the path points on the original path, and w_1, p_1, p_2, p_3, p_4 and w_3 are the path points on the smoothed path. O_2 is in the angle bisector of the path angle.

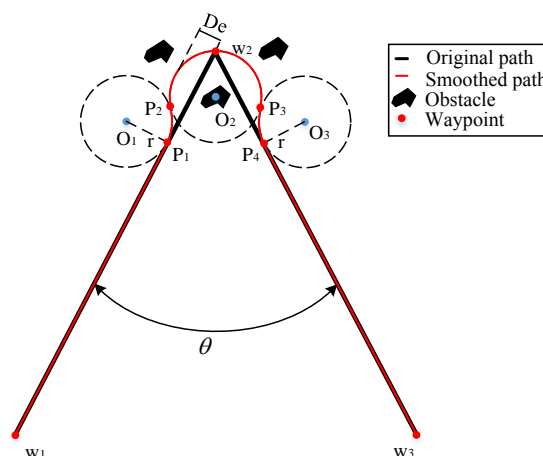


Figure 4. Smoothed path created using the L-3C-L method.

To compare the path deviation using the methods shown in Figure 3 and the L-3C-L method, their theoretical formulas are provided below.

The path deviation [13] of the method in Figure 3a is:

$$De_a = r \left(\frac{1}{\sin \frac{\theta}{2}} - 1 \right) \quad (10)$$

where r is the turning radius obtained by the IPT algorithm, and θ is the corner angle of the original path.

The path deviation [35] of the method in Figure 3b is:

$$De_b = r(1 + \cos \theta) \quad (11)$$

The path deviation of the L-3C-L method is:

$$De_{L3CL} = r \left(1 - \sin \frac{\theta}{2} \right) \quad (12)$$

Figure 5 shows the deviation comparison of the three smoothing methods: one arc, two arcs, and L-3C-L methods, where De is the largest path deviation, and r is the radius of the corner. The path deviation of the L-3C-L algorithm is the smallest of the three smoothing algorithms, even at very sharp corners.

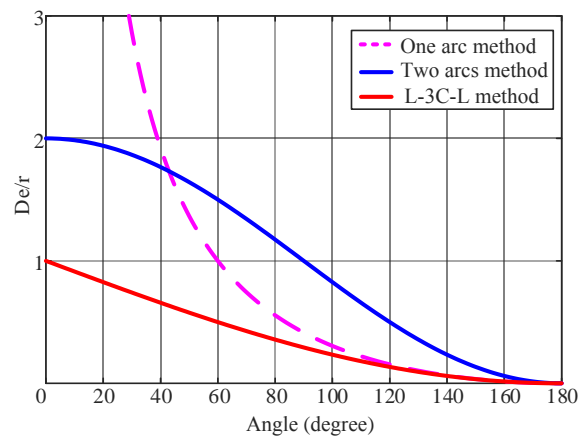


Figure 5. Path deviation comparison of the three smoothing methods.

There is a special case when two corners are very close to each other, as shown in Figure 6. w_1 , w_2 , w_3 and w_4 represent the original path, and w_1 , p_1 , p_2 , p_3 , p_4 , p_5 , p_6 and w_4 represent the smoothed path using our method. p_3p_4 is the common tangent of the circle O_2 and O_3 . In this case, two L-3C-L corners are merged to be two L-2C-L corners. The corresponding two deviations are:

$$\begin{cases} De_1 = r(1 - \sin \frac{\theta_1}{2}) \\ De_2 = r(1 - \sin \frac{\theta_2}{2}) \end{cases} \quad (13)$$

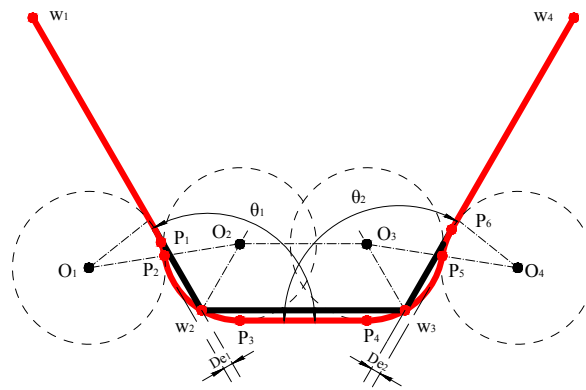


Figure 6. Adaptation to the special case with two close corners.

4. Optimal Speed Profile Design

Using the path-planning algorithm proposed in Section 3, any path can be decomposed into a succession of turns composed of circular arcs and straight lines. This is very useful in finding closed-form optimal speed profiles because both straight-line segments and circle arcs can be associated with constant speeds. The IPT algorithm in Section 2 provides an optimized turning speed, which differs from the desired USV speed when it follows straight lines. Therefore, when a turn is initiated and terminated, the velocity of the system should be regulated so that it can smoothly switch the motion status between the two modes.

Therefore, in this study, a method is proposed to plan the velocity. The work is simplified to plan the process of accelerations and decelerations between the two turns because the turning speed and the line-following speed have both been fixed. Thus, the overall speed profile will consist of three families of curves.

1. Constant speed curves at a minimum value v_r when the curvature profile is a circular arc. v_r is obtained from Section 2.
2. Constant cruising speed at the straight line with a cruising speed of the USV denoted as v_c .
3. Changing speed profile between v_r and v_c when the USV switches between the turning mode and the line-following mode. For these stages, there are two conditions to consider.

Condition 1: If the straight-line segment is sufficiently long, the USV will perform a full acceleration and deceleration at the two ends of the line, as shown in Figure 7a. The corresponding minimum length of the line segment is denoted as d_{\min} .

Condition 2: If the straight-line segment between two turns is not sufficiently long, $d < d_{\min}$, the USV will not be able to accelerate to v_c . Instead, it decelerates to some middle point to make sure that the velocity is v_r at the next turn, as shown in Figure 7b.

To determine whether the straight-line segment between two turns is sufficiently long for a specific USV, a test is needed to estimate the minimum length of the line segment. The path length of the acceleration section d_a and of the deceleration section d_d are recorded when the USV follows the given command. For a USV without a closed-loop velocity control, a full-power acceleration/deceleration test is required. When the USV has a velocity control unit, the command is usually given as an expected velocity as v_c when accelerating and v_r when decelerating, as shown in Figure 8a. The minimum length of the line segment for condition 1 is $d_{\min} = d_a + d_d$, as shown in Figure 8b.

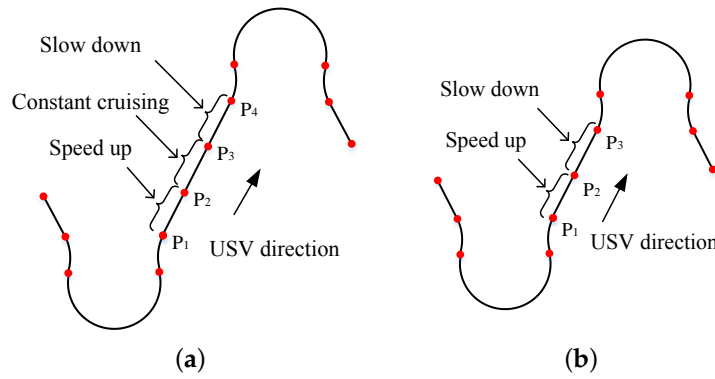


Figure 7. Speed design diagram: (a) The straight-line segment is sufficiently long. (b) The straight-line segment is not sufficiently long.

For the first condition $d \geq d_{\min}$ in Figure 7a, the USV can perform a full acceleration and deceleration at the two ends of the line. The acceleration and deceleration positions in Figure 7a are chosen to make $P_1P_2 = d_a$ and $P_3P_4 = d_d$.

For the second condition $d < d_{\min}$ in Figure 7b, we need further treatment to find the switch point P_2 from acceleration to deceleration. The functions for the acceleration and deceleration curves are denoted as $f_1(t)$ and $f_2(t)$, respectively. Assuming that the USV requires t_{x1} to accelerate and t_{x2} to decelerate so that it has speed v_r at p_3 , the equations in Equation (14) should be satisfied. t_{x1} and t_{x2} can be solved, and the switching point p_2 is chosen to make $P_1P_2 = \int_0^{t_{x1}} f_1(t)dt$.

$$\begin{cases} \int_0^{t_{x1}} f_1(t)dt + \int_{t_d-t_{x2}}^{t_d} f_2(t)dt = d \\ f_1(t_{x1}) = f_2(t_d-t_{x2}) \end{cases} \quad (14)$$

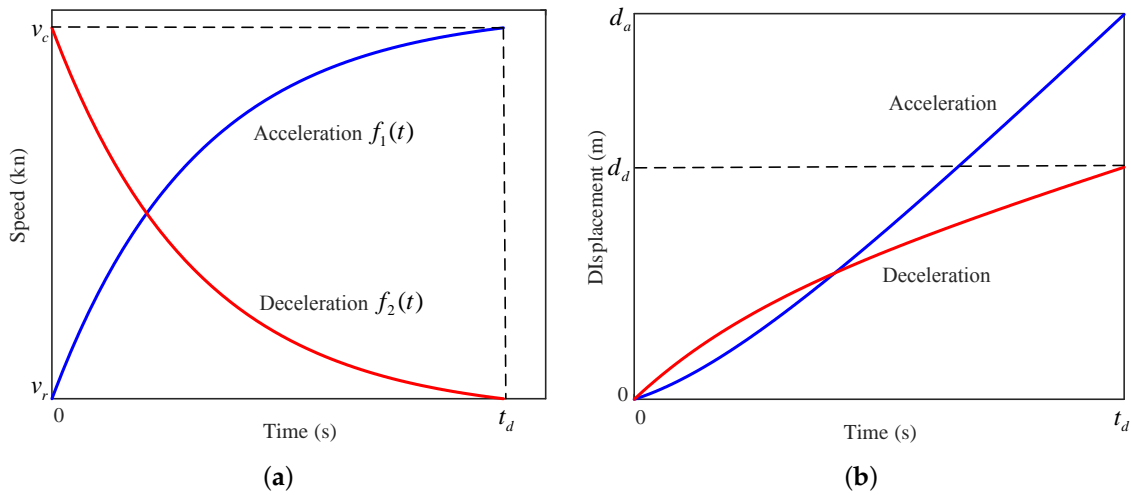


Figure 8. Acceleration and deceleration test: (a) Speed curves. (b) Displacement curves.

5. Simulation Results and Discussion

Since most ships do not have actuators in all degrees of freedom, the reduced-order model is often used. The 3 DOF horizontal plane model (surge, sway, yaw) is widely used in trajectory tracking control systems [36] with the assumption of planar motion. A classic 3 DOF USV model, the accuracy of which is verified in [37], is adopted in the following simulation work. The parameters of the vessel system are shown in Tables 1 and 2. Additional details on the dynamic model are presented in Appendix A, and the readers are referred to [37] for further information.

Table 1. Vessel properties and controller parameters.

Vessel Properties	Value	Controller Parameters	Value
Long	8.5 m	$K_{p,u}$	0.1 1/s
Width	3.0 m	$K_{p,\psi}$	5.0 1/s
Mass	3980 kg	$K_{d,\psi}$	1.0 s
Inertia	19,703 kg/m ²		

Table 2. Vessel Model Parameters.

Parameters	Value	Parameters	Value
X_u	−50	$X_{ u u}$	−135
Y_v	−200	$Y_{ v v}$	−2000
Y_r	0	$N_{ r r}$	0
N_v	0	X_{uuu}	0
N_r	−3224	Y_{vvv}	0
$F_{x,\min}$	−6550 N	N_{rrr}	−3224
$F_{x,\max}$	13,100 N	$X_{u'}$	0
$F_{y,\min}$	−645 N	$Y_{v'}$	0
$F_{y,\max}$	645 N	$Y_{r'}$	0
l_r	4.0 m	$N_{v'}$	0
		$N_{r'}$	0

5.1. IPT Results

According to the literature [37], the maximum speed v_{\max} of the USV is 18 kn. By using empirically determined values, the minimum turning radius, minimum speed and desired turning velocity are set as $r_{\min} = 20$ m, $v_{\min} = 1$ kn and $v_d = v_{\max} = 18$ kn. In the cost function, λ_1 , λ_2 and λ_3 emphasize the turning radius, turning speed and path deviation, respectively. Users can tune the parameters accordingly to specific task demands. The weights of the cost function in this paper are chosen as $(\lambda_1, \lambda_2, \lambda_3) = (1, 3.05, 278)$ to emphasize on decreasing path deviations at turning corners rather than the needs to obtain small turning radius and high turning speed. The initial step size and step-size adjustment coefficient are set to $\alpha_0 = 0.5$, $\xi = 1.1$. The optimization process is terminated when the updated value of the path parameter is smaller than the user-specified positive value $|x_{k+1}^* - x_k| \leq 0.65$.

To verify the convergence of the algorithm, three different groups of the path parameters ($r_0 = 49$ m, $v_0 = 16$ kn), ($r_0 = 39$ m, $v_0 = 12$ kn), and ($r_0 = 28$ m, $v_0 = 10$ kn), are chosen as the initial values for the IPT simulations. To compare the performance of the TPA-IPT method proposed in this study with the fixed step-size version in [28], simulations using both methods are conducted for each group, and the results are shown in Figure 9 and Table 3. Figure 9a,c,e show the convergence of the cost function value and the two path parameters for each initial value set using TPA-IPT, and Figure 9b,d,f are their counterparts for the fixed step IPT. It is observed that both algorithms can achieve a similar minimum point with small variance (Table 3) regardless of how the initial value changes. However, the convergence speed of the TPA-IPT is much faster than that of the fixed step IPT.

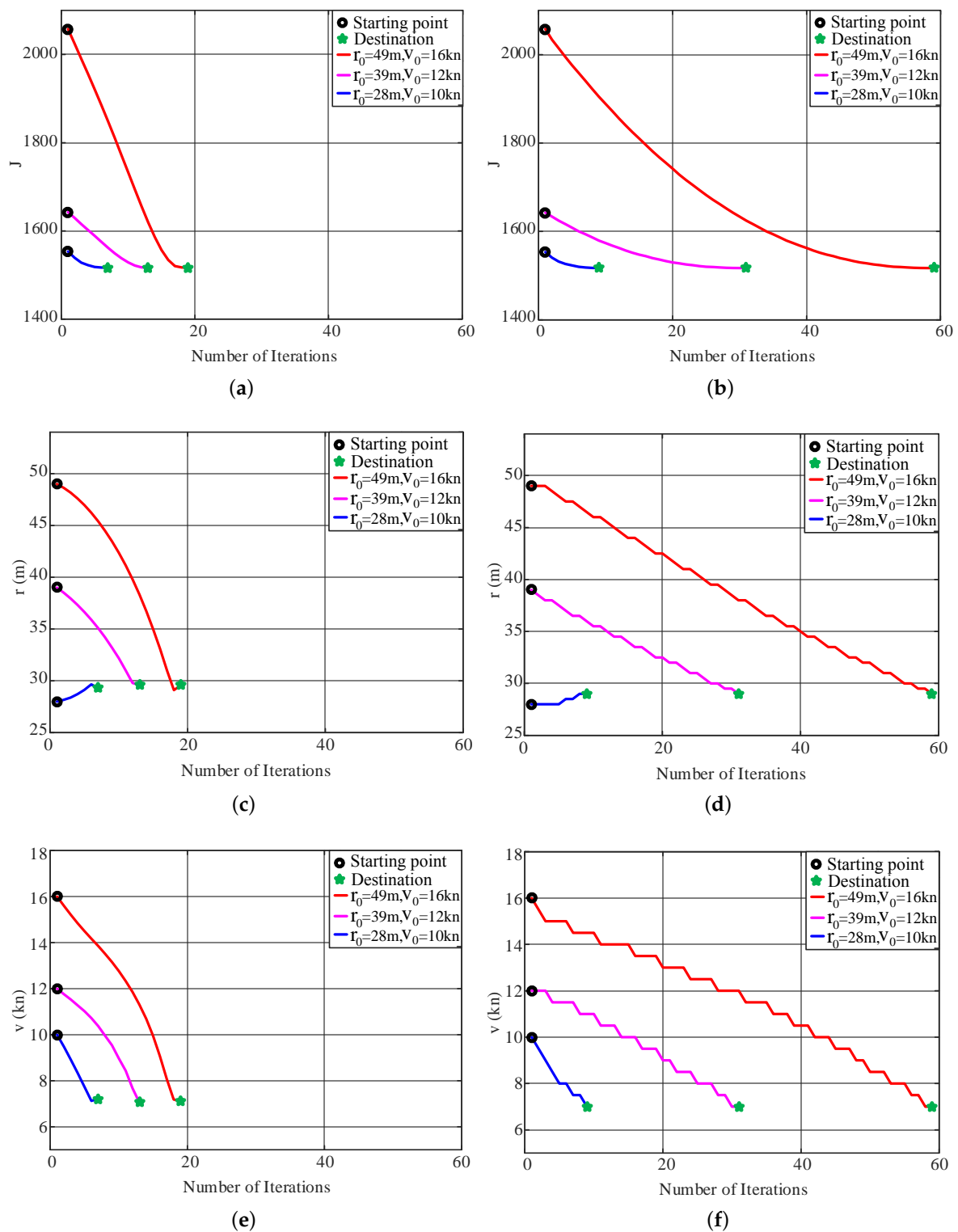


Figure 9. Optimization result graphs: (a,c,e) are the convergence curves of the cost function value and the two path parameters for each initial value set using TPA-IPT; (b,d,f) are their counterparts for the fixed step IPT.

Table 3. Comparison of results with and without an adaptive step length.

	Radius (m)	Speed (kn)	Average Tracking Error (m)	Number of Iterations
Initial value	49	16	4.33	
Normal IPT	29.00	7.00	3.84	59
TPA-IPT	29.58	7.10	3.82	19
Initial value	39	12	4.21	
Normal IPT	29.00	7.00	3.84	31
TPA-IPT	29.60	7.05	3.81	13
Initial value	28	10	4.66	
Normal IPT	29.00	7.00	3.84	9
TPA-IPT	29.31	7.21	3.87	7

To perform a better comparison, the descent processes of the cost function value are drawn in Figure 10 for the first group of the initial value by using both a contour graph and in a 3D cost function surface. The sharp slope of the function indicates that TPA-IPT speeds up and arrives at the minimum point with only 19 iterations, which is approximately one third of the number of iterations needed using fixed step IPT. For each iteration, TPA-IPT needs 2 more trials to adjust the step length. Therefore, the total number of trials needed by TPA-IPT is 114, which is less than half of the trial numbers for the fixed step strategy. This difference helps decrease the working load of the IPT tests when it is applied in real vessel tests.

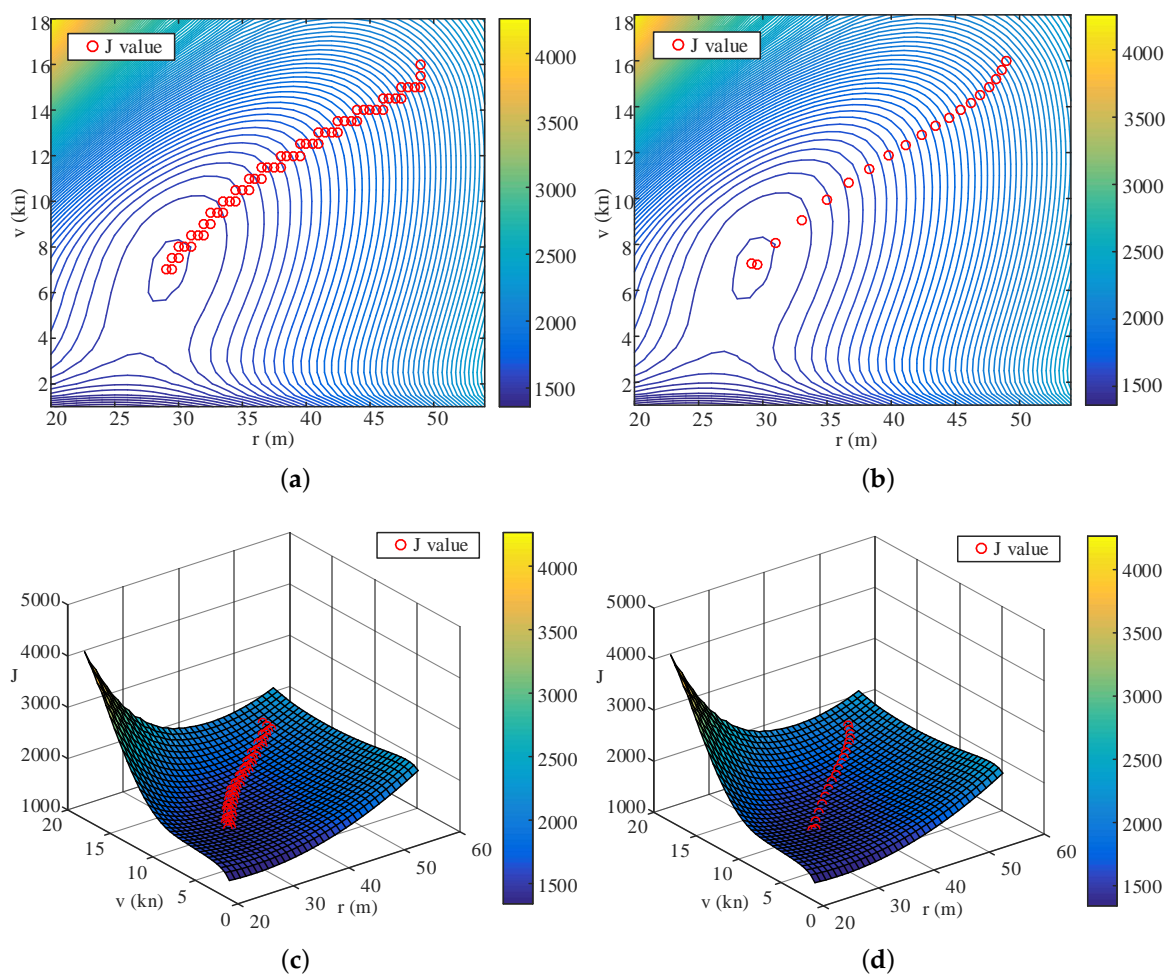


Figure 10. Diagram of descending processes with the first group initial values: (a,c) are the iterative processes of TPA-IPT, and (b,d) are the corresponding results of the normal IPT.

A similar result is obtained for other initial value setups. The variance of the step length using TPA-IPT is shown in Figure 11, and a drop in the step size at the last several iterations is observed. This result is very reasonable considering the need for the fine tuning of the parameters when they are close to the minimum point.

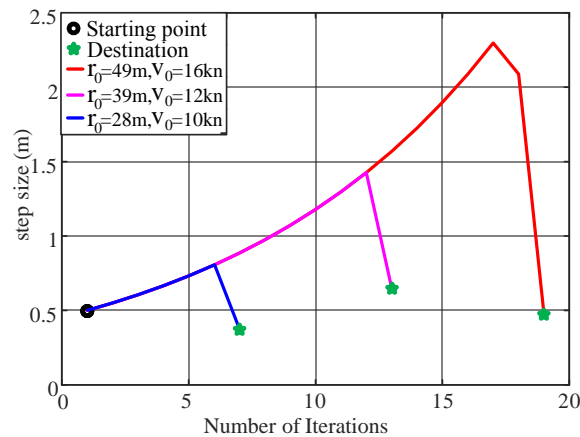


Figure 11. Step-size variations of the optimization process.

5.2. Path-Smoothing and Speed-Planning Simulations

Section 5.1 shows three sets of the initial path parameter values that are tested using the TPA-IPT method; these values provide close to optimal solutions for the turning radius and turning speed. The optimized tuning parameters $r_r = 29.60$ m, $v_r = 7.05$ kn will be used in the following work on the path-smoothing and speed profile-planning. To verify the effect of the path-smoothing and speed-planning algorithm, various turning conditions are designed. Four tested scenes, including a path with a large turning angle, a path with a small turning angle, a path for water sampling, and a path for bathymetric measurement, are designed to verify the performance of the proposed algorithm in this paper. The code was run on a computer with an Intel i5 2.30 GHz processor and 4.00 GB of RAM. To verify the real-time performance of the proposed algorithms, each test scene was run 10 times. As shown in Table 4, the average time used for the path and speed profile calculation in the four test scenes are 0.044 s/km, 0.072 s/km, 0.016 s/km and 0.079 s/km respectively. In addition, the path-following performances are listed in Table 5 with a comparison between the original and optimized path and speed profiles. As expected, the path-following accuracy and the drive safety are considerably improved when the method described in this study is used. The details on the results will be further discussed for individual cases in the following paragraphs.

Table 4. Average runtime of algorithm.

	Large Angle	Small Angle	Water Sampling	Bathymetry
Average runtime (s/km)	0.044	0.072	0.016	0.079

Table 5. Tracking parameters of USV with and without route optimization.

Tracking Parameters	Large Angle	Small Angle	Water Sampling	Bathymetry
Original displacement (m)	2931.05	2728.44	23,265.18	6975.18
Optimized displacement (m)	2881.94	2693.33	23,123.05	6563.18
Original travelling time (s)	512.92	479.54	3811.95	1156.29
Optimized travelling time (s)	580.45	553.85	3987.13	1543.95
Original maximum path error (m)	35.87	19.47	34.96	30.40
Optimized maximum path error (m)	5.34	5.06	5.53	5.78

Figure 12 shows the simulation results for the large-angle turning scene. Black curves are the planned path, blue curves are the high-speed USV trajectories, red curves are the low-speed trajectories, the purple curve is the acceleration trajectory, and the green curve is the deceleration trajectory. The corresponding speed change for Figure 12b,d are shown in Figure 13. When the USV makes a large-angle turn at a speed of 12 kn, due to the limitation of dynamics, a large tracking error will be produced; this error will oscillate when returning to track the straight line, as shown in Figure 12a,c. The maximum path error is 35.87 m, as shown in Table 5. After applying the path-smoothing and speed-planning work proposed in this paper, it was found that the USV slows down near the turning corner and follows the curve at a speed of 7.05 kn. This algorithm agrees with the dynamics of USVs; therefore, the tracking error is small, and the oscillation is considerably reduced, as shown in Figure 12b,d. The maximum path error is 5.34 m, and the error value is insensitive to the path conditions. The maximum path error is reduced by 85%, and the driving of the USV is considerably stabilized with a small increase (13%) in the travelling time.

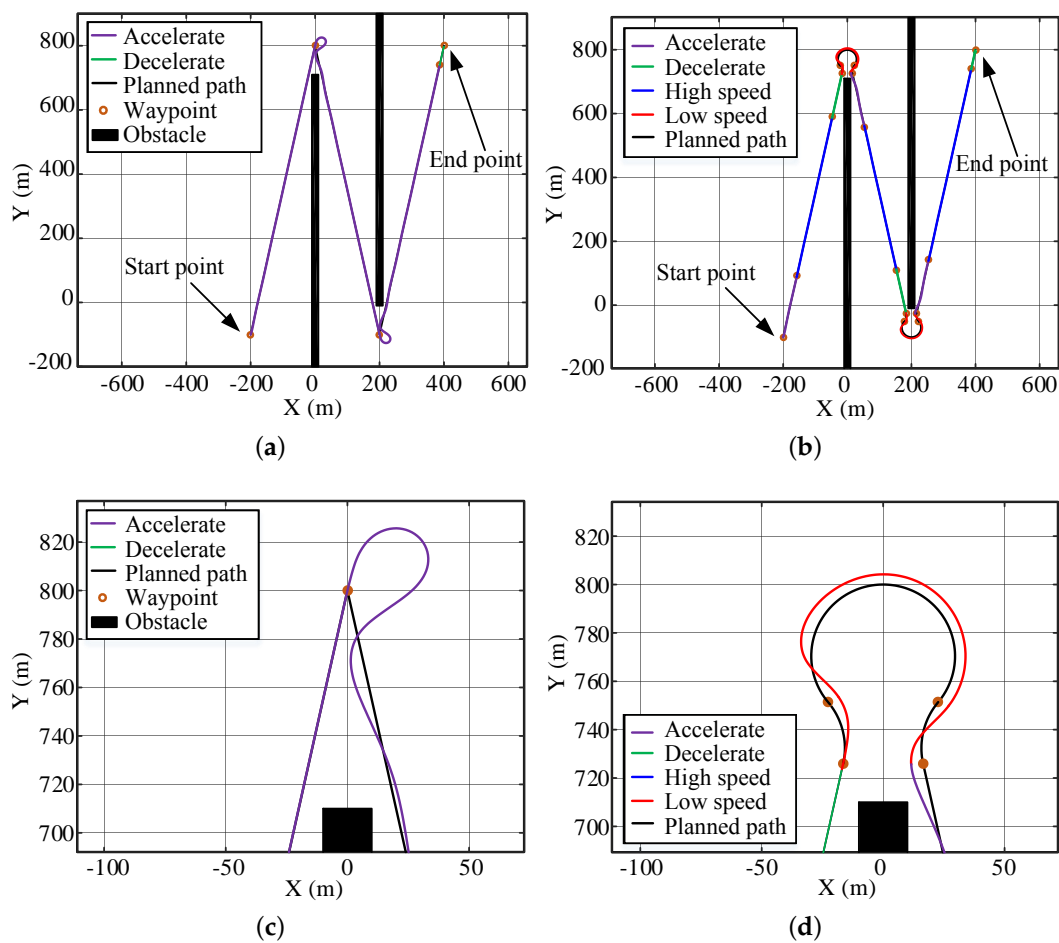


Figure 12. Simulation of large-angle turning tracking scene: (a) Original path-tracing results. (b) Optimized path-tracing results. (c,d) Enlarged partial views of the paths for the turns in (a,b).

For tasks with less demands on small path deviation and turning safety but more on small travelling time, the weight factors in (1) should be readjust to have large λ_2 and small λ_1 . In this case, the resulted turning radius would be small, and the turning speed would be large using the IPT method. However, the authors do not recommend high-speed turning considering the risk of capsizing.

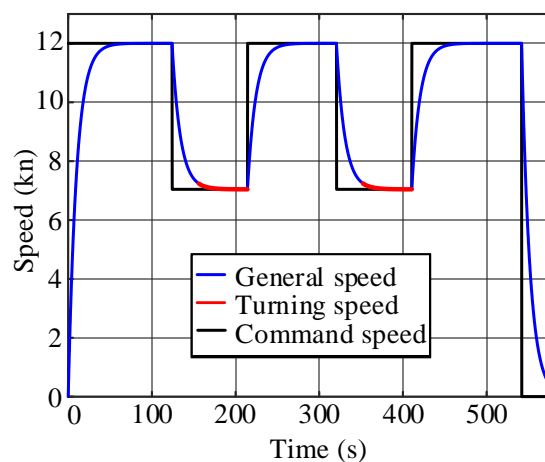


Figure 13. Speed curve of the large-angle turning tracking scene.

When the USV turns at small angles, the path-following performances of the USV without and with using the method described in this paper are compared. The influence of its dynamic constraints will decrease, as shown in Figure 14a. Therefore, the maximum path deviation is originally 19.47 m, which is smaller than that for the large-angle case. By applying the path-smoothing and speed-planning algorithm, the maximum path deviation is decreased to 5.06 m, as shown in Figure 14b. The displacement of the optimized path is 1.3% less than that of the original path, and the time used is 15.5% more than that of the original path. With a slight sacrifice of travelling speed, the method described in this paper shortens the travelling distance and, more importantly, considerably decreases the path-following error.

The decreased path deviation sometimes is vital for driving safety. As shown in Figure 14a, the USV crashed onto an obstacle using the original path and speed profiles when it is turning into a narrow passage. The optimized path and speed design largely decreased the path deviation, and the USV successfully passed through the cluttered area, as shown in Figure 14b.

Various situations for speed profile-planning are also designed to verify the algorithm in Section 4. As shown in Figure 14, the line segment between the triangular obstacle and the rectangular obstacle is too short to achieve full acceleration, which corresponds to the condition 2 in Section 4 for the speed profile-planning. With the acceleration and deceleration tests in Figure 15a, the corresponding displacement curves are calculated and shown in Figure 15b. Thus, $d_a = 172.04$ m, $d_d = 138.62$ m, $d_{\min} = 310.66$ m. The length of the second line segment in Figure 14 $d = 198.97$ m < 310.66 m. Thus, (14) is used to plan the speed profile for the segment. The acceleration and deceleration distances in the straight-line segment can be obtained as $d_1 = 73.5$ m, $d_2 = 125.47$ m. As shown in Figure 16, the USV can reduce the speed to the optimal value before turning at the next corner.

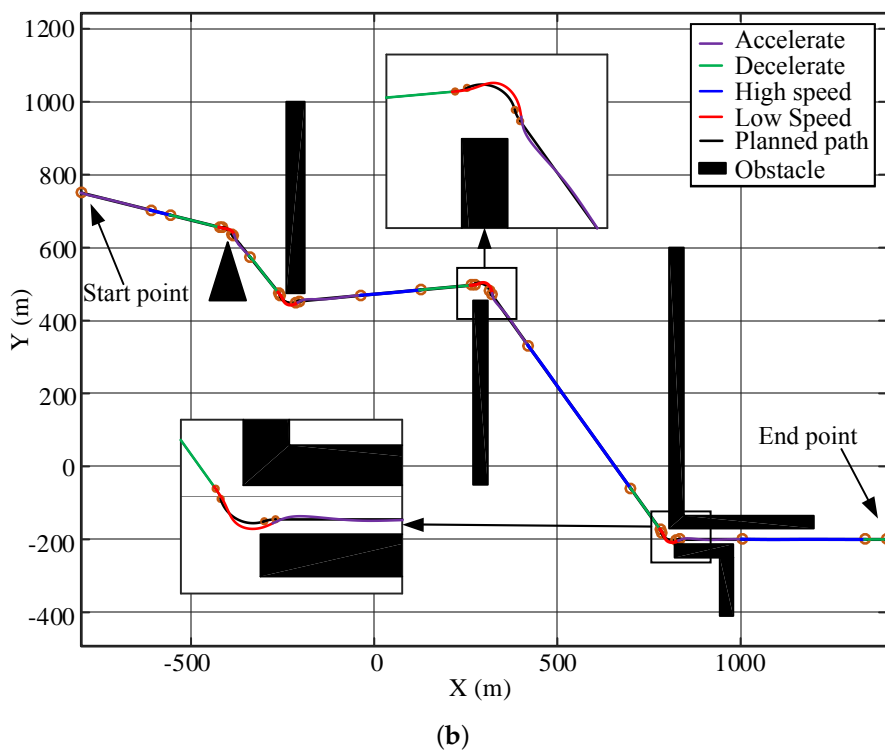
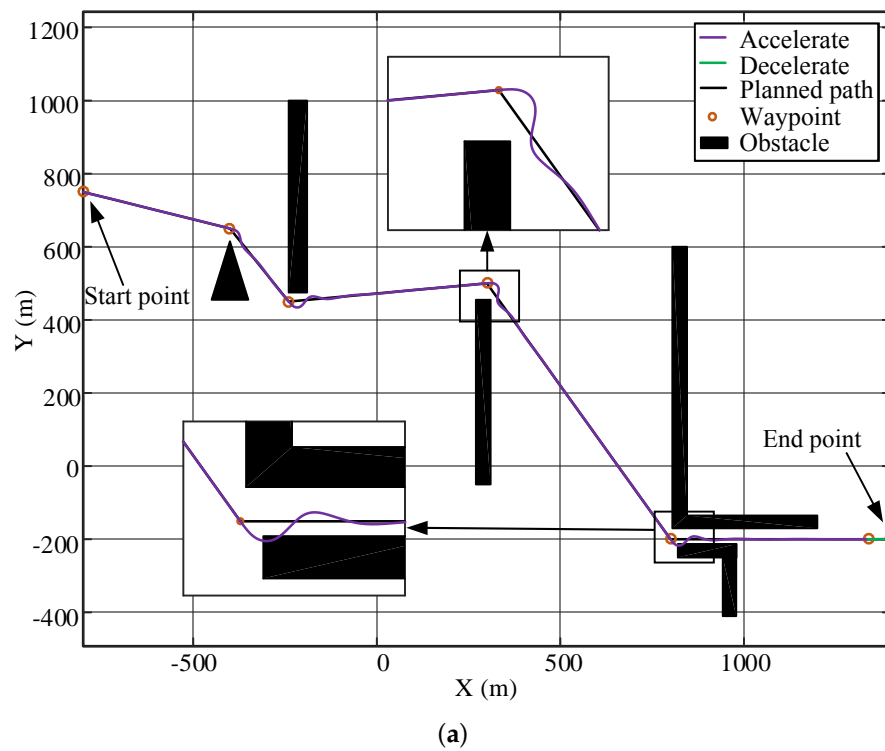


Figure 14. Simulation of the small-angle turning tracking scene: (a) Original path-tracing results. (b) Optimized path-tracing results.

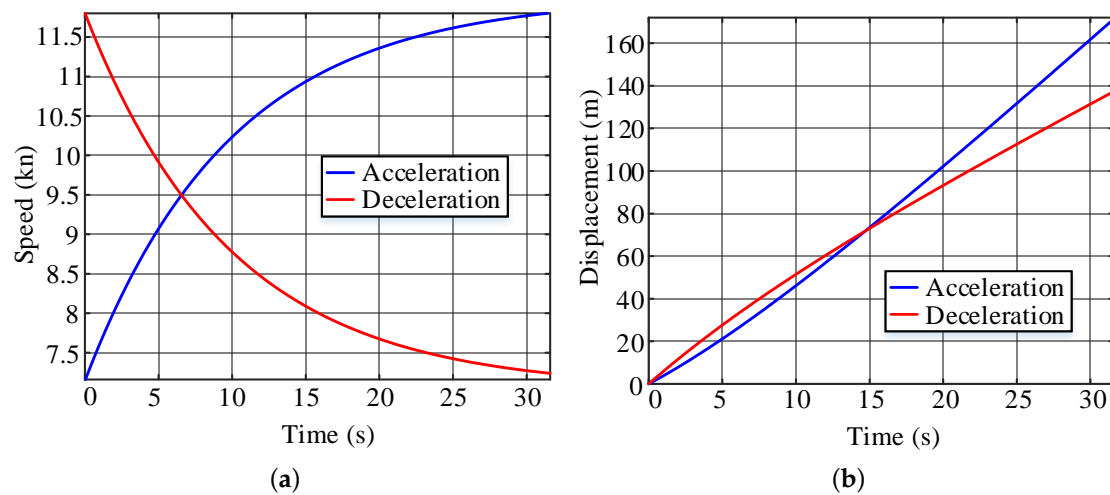


Figure 15. Acceleration and deceleration tests: (a) Speed test curves. (b) Displacement test curves.

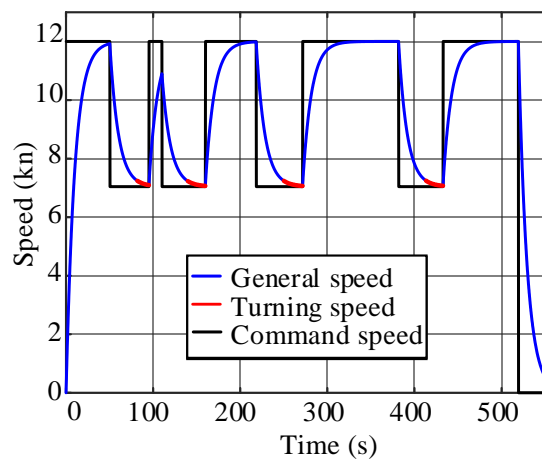


Figure 16. Speed curve of the small-angle turning tracking scene.

A similar result is obtained for the water sampling task around an island. A map of Mount Putuo, which is one of the 1390 islands in Zhoushan Archipelago, China, is drawn in Figure 17. The sampling points scatter around the island and the USV needs to visit these points accurately to retrieve the water samples.

During the mission, the USV encountered two small-angle turns and six large-angle turns. As shown in Figure 17a, the maximum path error at the corners by the original path and speed setup is 34.96 m, so the USV largely deviates from the desired sampling locations. By smoothing the path and planning the speed, the maximum tracking error can be reduced to 5.53 m, as shown in Figure 17b. The motion stability of the USV is considerably improved, and the water samples are retrieved close to the desired locations.

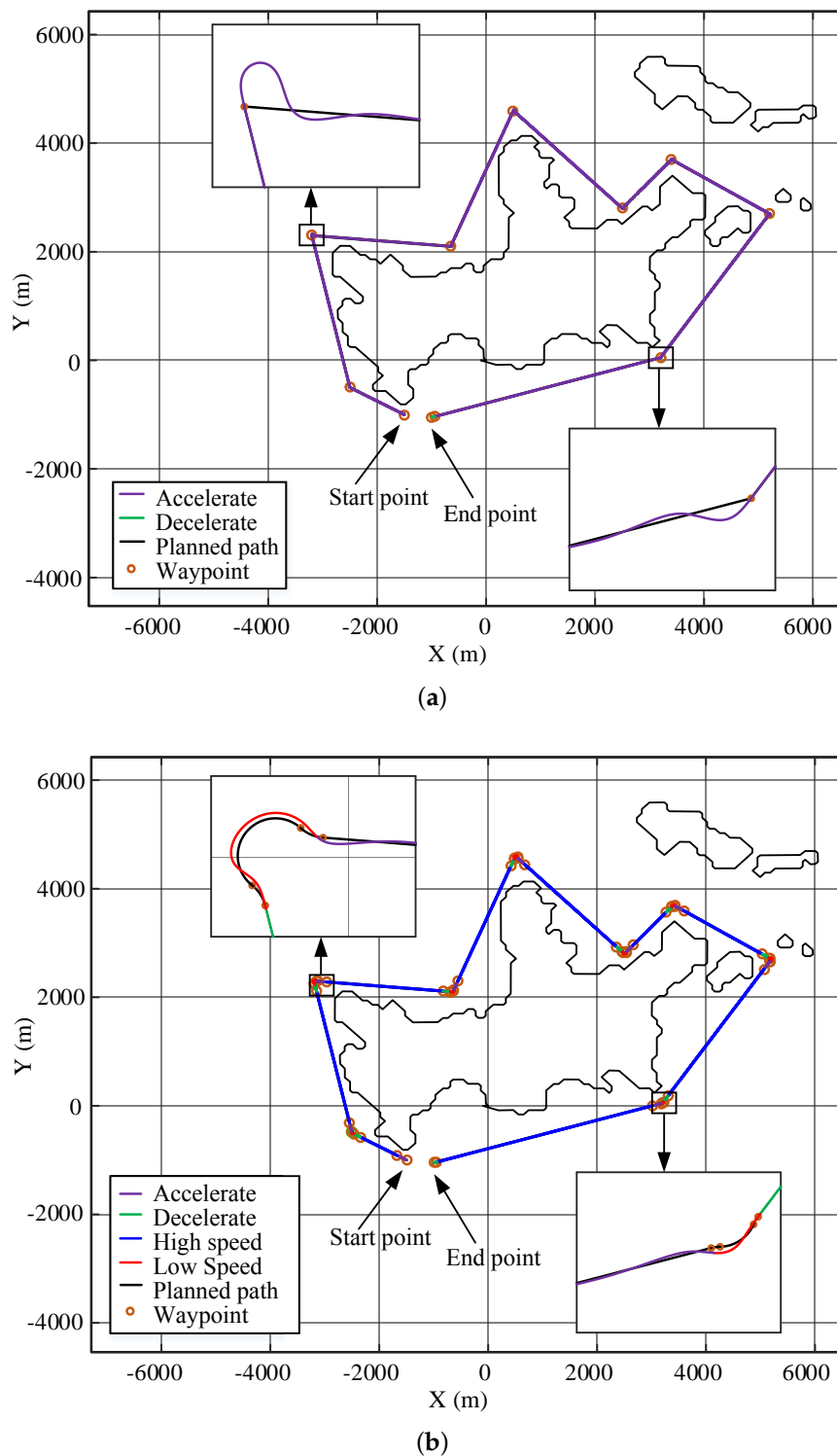


Figure 17. Simulation of water sampling: (a) Original path-tracing results. (b) Optimized path-tracing results.

The bathymetric measurement is another example to show the benefits of the IPT method. As shown in Figure 18, the USV needs to traverse an area to obtain the water depth data following a pre-defined path. This task requires the USV to have higher path-following accuracy and lower cruise speed [2]. The new weights of the cost function and the minimum turning radius in the IPT algorithm are reset as $(\lambda_1, \lambda_2, \lambda_3) = (5, 4, 278)$ and $r_{\min} = 14$ m. The remaining parameters remain unchanged.

The original parameters $r_0 = 39$ m, $v_0 = 12$ kn were optimized by IPT to be $r_r = 16.66$ m, $v_r = 4.96$ kn. According to the line spacing in Reference [38] and the property of the USV model, the line spacing is selected as 40 m.

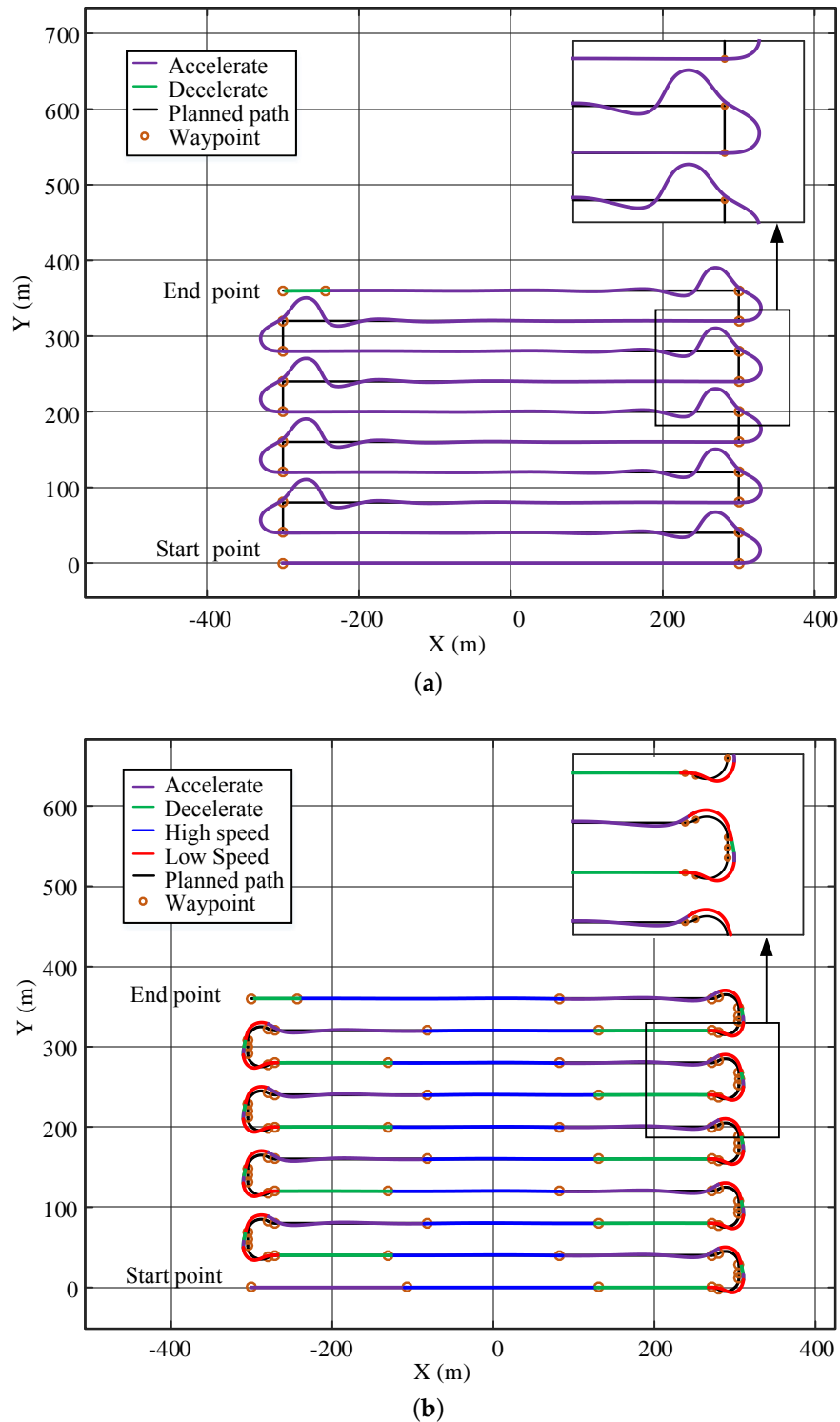


Figure 18. Simulation of bathymetric measurement: (a) Original path-tracing results. (b) Optimized path-tracing results.

As shown in Figure 18, the maximum path deviation generated by the USV using the original cruise speed is 30.40 m, which is larger than the line spacing and causes intersections between the

paths at the corners. This could leave some places uncovered at the corner areas. The path-following performance is greatly improved after path-smoothing and speed-planning, and its maximum path deviation is 5.78 m. In addition, the smooth actions at the corners can also help to reduce the measurement noises caused by highly dynamical turning behaviors of the USVs.

6. Conclusions

This study proposes a practical and real-time solution for the path and speed profile generation by considering the influence of vehicle dynamics on hydrographic survey tasks. The offline learning method uses experimental data to obtain optimal turning radius and speed for the vehicle, which provides design constraints for the path and speed profile optimization. The improved IPT method is proven to be effective in obtaining satisfactory turning parameters within a limited number of trials, which balances the demands of small turning radius, fast turning speed and small path-following error. The resulting radius is used to smooth the sharp corners with the L-3C-L /L-2C-L path-smoothing method, which guarantees relatively small path deviations from the original path. The resulting turning speed is used to plan the speed trajectory and shape smooth transients between the fast-straight-line following mode and the relatively slow turning mode.

The simulation results show that the proposed path-speed profile optimization approach produces smaller path deviations and is more suitable for hydrographic survey activities. The relatively slow turning speed and smoothed path largely improves the path-following performance. Compared to the case using original path and speed profiles, the maximum path deviations decreased by 81.07% on average at a very small cost of time increase by 16.70%. The smoothness of the trajectory and the driving safeness are largely improved as well, and the calculation cost is low, which makes it potentially beneficial for real-time tasks in complex marine environments.

In addition, the algorithm proposed in this paper can be extended to unmanned systems with similar dynamical properties and demands on trajectories. For example, it can be used to optimize the path and speed profiles for an UAV system to have smoothed turning behaviors and more accurate path-following performance.

Author Contributions: Conceptualization, Y.X.; Data curation, Q.L.; Funding acquisition, Y.Y.; Methodology, Y.X.; Project administration, Y.Y.; Software, J.Z.; Writing—original draft, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This study was supported by the National Natural Science Foundation of China, grant no. 61903243, and the Shanghai Rising-Star Program, China, No. 19QA1403500.

Conflicts of Interest: The authors declare no conflict of interest.

Abbreviations

The following abbreviations are used in this manuscript:

USVs	Unmanned surface vehicles
ASC	Autonomous surface craft
APF	Artificial potential fields
RRTs	Rapidly exploring random trees
PSO	Particle Swarm Optimization
ILC	Iterative learning control
IPT	Iterative parameter-tuning
TPA	Two-point step-size adaptation
CSA	Cumulative step-size adaptation
3D	Three dimensions

Appendix A. Theoretical Background

Appendix A.1. Reference Frame

The North-East Down (NED) coordinate system (x_n, y_n, z_n) is defined by the moving tangent plane of the earth's surface. x -axis, y -axis and z -axis point to north, east and downwards respectively, as shown in Figure A1.

The body-fixed (BODY) coordinate system (x_b, y_b, z_b) is a frame fixed to the body of an USV and therefore moves together with the body. The origin and axis are selected according to the natural behavior of USV. In which roll, pitch and yaw is rotation around the x -axis, y -axis and z -axis, respectively. When calculating the position and direction of USV in NED, the origin of the BODY is used to replace the USV, as shown in Figure A1. In this study, the origin of the BODY coordinate system is located at the USV's center of gravity.

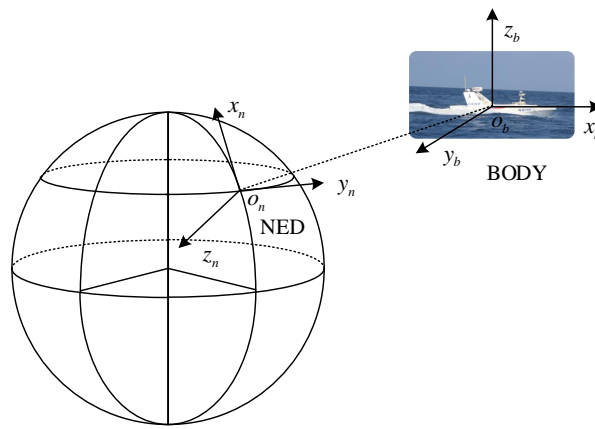


Figure A1. Coordinate frames.

Appendix A.2. Mathematical Model

A commonly used ship three-degree-of-freedom model is based on the assumption that heave, roll and pitch are small. It's a very reasonable assumption for most traditional ships and underwater vehicles. In the 3 DOF model, the effect of heave, roll and pitch is ignored and only the effect of surge, sway and yaw is analyzed. The 3 DOF kinematic equations can be expressed in vector form as:

$$\dot{\eta} = J(\psi)v \quad (A1)$$

$$M\dot{v} + C(v)v + D(v)v = \tau + \tau_{wind} + \tau_{wave} \quad (A2)$$

where

$$\eta = [N, E, \psi]^T \quad (A3)$$

$$v = [u, v, r]^T \quad (A4)$$

$$\tau = [X, Y, N]^T \quad (A5)$$

where η represents position and orientation vector. The position is the distance between NED and BODY under NED coordinate, and the orientation is a Euler angle vector. The vector v includes the linear velocity vector v and angular velocity vector ω under the BODY coordinate. The vector τ contains the forces f and moments m in the BODY coordinate. Matrices J , M , D and C is the transformation, inertia, damping and Coriolis matrix, respectively. The vectors τ_{wind} and τ_{wave} are environmental forces. Table A1 shows the motions and rotations of the 3 DOF that USV can operate, which is represented by the SNAME (1950) notation.

Table A1. The notation of SNAME (1950) for marine vessels.

	Forces and Moments	Linear and Angular Velocities	Positions and Euler Angles
Motions in the x -direction (surge)	X	u	x
Motions in the y -direction (sway)	Y	v	y
Rotation about the z -axis (yaw)	N	r	ψ

The transformation matrix $J(\psi)$ which equals the rotation matrix $R_{z,\psi}$ is:

$$J(\psi) = R_{z,\psi} = \begin{pmatrix} \cos \psi & -\sin \psi & 0 \\ \sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{A6})$$

The inertia matrix and Coriolis-centripetal matrix are obtained by adding the additional mass matrix and the rigid-body matrix. The inertia matrix is calculated as follows:

$$M = M_{RB} + M_A \quad (\text{A7})$$

with

$$M_{RB} = \begin{pmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & I_z \end{pmatrix} \quad M_A = - \begin{pmatrix} X_{\dot{u}} & 0 & 0 \\ 0 & Y_{\dot{v}} & Y_{\dot{r}} \\ 0 & N_{\dot{v}} & N_{\dot{r}} \end{pmatrix} \quad (\text{A8})$$

where I_z is the moment of inertia and m is the mass of the USV. $N_{\dot{v}}$, $N_{\dot{r}}$, $X_{\dot{u}}$, $Y_{\dot{v}}$ and $Y_{\dot{r}}$ are added mass terms.

The Coriolis-centripetal matrix is calculated as:

$$C(v) = C_{RB}(v) + C_A(v) \quad (\text{A9})$$

with

$$C_{RB} = \begin{pmatrix} 0 & 0 & -mv \\ 0 & 0 & mu \\ mv & -mu & 0 \end{pmatrix} \quad (\text{A10})$$

$$C_A = \begin{pmatrix} 0 & 0 & -Y_{\dot{v}}v + \frac{Y_{\dot{r}} + N_{\dot{v}}}{2}r \\ 0 & 0 & X_{\dot{u}}u \\ Y_{\dot{v}}v + \frac{Y_{\dot{r}} + N_{\dot{v}}}{2}r & -X_{\dot{u}}u & 0 \end{pmatrix} \quad (\text{A11})$$

where v and u are the linear velocities and r is the angular velocity, as shown in Table A1.

Adding the linear damping part and the nonlinear damping part to obtain a total hydrodynamic damping matrix:

$$D(v) = D_L + D_{NL}(v) \quad (\text{A12})$$

with

$$D_L = - \begin{pmatrix} X_u & 0 & 0 \\ 0 & Y_v & Y_r \\ 0 & N_v & N_r \end{pmatrix} \quad (\text{A13})$$

$$D_{NL}(v) = - \begin{pmatrix} X_{|u|u}|u| + X_{uuu}u^2 & 0 & 0 \\ 0 & Y_{|v|v}|v| + Y_{vvv}v^2 & 0 \\ 0 & 0 & N_{|r|r}|r| + N_{rrr}r^2 \end{pmatrix} \quad (\text{A14})$$

where $X_{|u|u}$, $Y_{|v|v}$, $N_{|r|r}$, X_{uuu} , Y_{vvv} , N_{rrr} are nonlinear damping terms and X_u , Y_v , Y_r , N_v , N_r are linear damping terms.

Appendix A.3. Actuator Forces and Low-level Controller

The USV has only two actuators: a propeller and a rudder. The control input which are rudder angle and propeller force is less than the number of degrees of freedom. Therefore, the USV is underactuated. The control input vector is :

$$\tau = \begin{pmatrix} X \\ Y \\ N \end{pmatrix} = \begin{pmatrix} F_x \\ F_y \\ l_r F_y \end{pmatrix} \quad (\text{A15})$$

where F_x and F_y are proportional to $n \cos \theta$ and $n \sin \theta$ respectively. where n and θ are the propeller shaft speed and rudder deflection angle, respectively. l_r is the rudder length.

The low-level controller consists of a speed controller and a heading controller, which are used to control the speed and heading, respectively.

Velocity controller:

$$F_x = -X_u u - X_{|u|u} |u| u - X_{uuu} u^3 - mrv + K_{p,u} m(u_d - u) \quad (\text{A16})$$

Heading controller:

$$l_r F_y = K_{p,\psi} I_z (\Gamma(\psi_d - \psi) + K_{d,\psi} (r_d - r)) \quad (\text{A17})$$

where $K_{p,u}$ and $K_{p,\psi}$ are proportional gains. $K_{d,\psi}$ is derivative time.

References

1. Manley, J.E. Unmanned surface vehicles, 15 years of development. In Proceedings of the OCEANS 2008, Quebec City, QC, Canada, 15–18 September 2008; pp. 1–4.
2. Specht, M.; Specht, C.; Lasota, H.; Cywiński, P. Assessment of the Steering Precision of a Hydrographic Unmanned Surface Vessel (USV) along Sounding Profiles Using a Low-Cost Multi-Global Navigation Satellite System (GNSS) Receiver Supported Autopilot. *Sensors* **2019**, *19*, 3939. [\[CrossRef\]](#)
3. Tang, P.; Zhang, R.; Liu, D.; Huang, L.; Liu, G.; Deng, T. Local reactive obstacle avoidance approach for high-speed unmanned surface vehicle. *Ocean Eng.* **2015**, *106*, 128–140. [\[CrossRef\]](#)
4. Naeem, W.; Henrique, S.C.; Hu, L. A reactive colregs-compliant navigation strategy for autonomous maritime navigation. *IFAC-PapersOnLine* **2016**, *49*, 207–213. [\[CrossRef\]](#)
5. Bruce, J.; Veloso, M.M. Real-time randomized path planning for robot navigation. In *Robot Soccer World Cup*; Springer: Berlin/Heidelberg, Germany, 2002; pp. 288–295.
6. Xin, J.; Zhong, J.; Li, S.; Sheng, J.; Cui, Y. Greedy Mechanism Based Particle Swarm Optimization for Path Planning Problem of an Unmanned Surface Vehicle. *Sensors* **2019**, *19*, 4620. [\[CrossRef\]](#) [\[PubMed\]](#)
7. Hart, P.E.; Nilsson, N.J.; Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Trans. Syst. Sci. Cybern.* **1968**, *4*, 100–107. [\[CrossRef\]](#)
8. Ferguson, D.; Stentz, A. Field D*: An interpolation-based path planner and replanner. In *Robotics Research*; Springer: Berlin/Heidelberg, Germany, 2007; pp. 239–253.
9. Daniel, K.; Nash, A.; Koenig, S.; Felner, A. Theta*: Any-angle path planning on grids. *J. Artif. Intell. Res.* **2010**, *39*, 533–579. [\[CrossRef\]](#)
10. Yang, K.; Sukkarieh, S. An analytical continuous-curvature path-smoothing algorithm. *IEEE Trans. Robot.* **2010**, *26*, 561–568. [\[CrossRef\]](#)
11. Yang, K.; Sukkarieh, S. 3D smooth path planning for a UAV in cluttered natural environments. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 794–800.
12. Consolini, L.; Locatelli, M.; Minari, A.; Piazzini, A. A linear-time algorithm for minimum-time velocity planning of autonomous vehicles. In Proceedings of the 2016 24th Mediterranean Conference on Control and Automation (MED), Athens, Greece, 21–24 June 2016; pp. 490–495.
13. Dubins, L.E. On curves of minimal length with a constraint on average curvature, and with prescribed initial and terminal positions and tangents. *Am. J. Math.* **1957**, *79*, 497–516. [\[CrossRef\]](#)

14. Fraichard, T. Smooth trajectory planning for a car in a structured world. In Proceedings of the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA, USA, 9–11 April 1991; pp. 318–323.
15. Yang, G.; Kapila, V. Optimal path planning for unmanned air vehicles with kinematic and tactical constraints. In Proceedings of the 41st IEEE Conference on Decision and Control, Las Vegas, NV, USA, 10–13 December 2002; Volume 2, pp. 1301–1306.
16. Kanayama, Y.J.; Hartman, B.I. Smooth local-path planning for autonomous vehicles. *Int. J. Robot. Res.* **1997**, *16*, 263–284. [\[CrossRef\]](#)
17. Pamali, A.P. Using Clothoidal Spirals to Generate Smooth Tool Paths for High Speed Machining. Master's Thesis, North Carolina State University, Raleigh, NC, USA, 2004.
18. Lima, P.F.; Trincavelli, M.; Mårtensson, J.; Wahlberg, B. Clothoid-based speed profiler and control for autonomous driving. In Proceedings of the 2015 IEEE 18th International Conference on Intelligent Transportation Systems, Las Palmas, Spain, 15–18 September 2015; pp. 2194–2199.
19. Talcahashi, A.; Hongo, T.; Ninomiya, Y.; Sugimoto, G. Local path planning and motion control for agv in positioning. In Proceedings the IEEE/RSJ International Workshop on Intelligent Robots and Systems (IROS '89), The Autonomous Mobile Robots and Its Applications, Tsukuba, Japan, 4–6 September 1989.
20. Nelson, W. Continuous-curvature paths for autonomous vehicles. In Proceedings of the 1989 International Conference on Robotics and Automation, Scottsdale, AZ, USA, 14–19 May 1989; pp. 1260–1264.
21. Vendittelli, M.; Laumond, J.P.; Nissoux, C. Obstacle distance for car-like robots. *IEEE Trans. Robot. Autom.* **1999**, *15*, 678–691. [\[CrossRef\]](#)
22. Villagra, J.; Milanés, V.; Pérez, J.; Godoy, J. Smooth path and speed planning for an automated public transport vehicle. *Robot. Auton. Syst.* **2012**, *60*, 252–265. [\[CrossRef\]](#)
23. Lini, G.; Piazzzi, A.; Consolini, L. Algebraic solution to minimum-time velocity planning. *Int. J. Control. Autom. Syst.* **2013**, *11*, 805–814. [\[CrossRef\]](#)
24. Campbell, S.F. Steering Control of an Autonomous Ground Vehicle with Application to the DARPA Urban Challenge. Ph.D. Thesis, Massachusetts Institute of Technology, Cambridge, MA, USA, 2007.
25. Liu, S. An on-line reference-trajectory generator for smooth motion of impulse-controlled industrial manipulators. In Proceedings of the 7th International Workshop on Advanced Motion Control, Maribor, Slovenia, 3–5 July 2002; pp. 365–370.
26. Yin, S.; Ding, S.X.; Xie, X.; Luo, H. A review on basic data-driven approaches for industrial process monitoring. *IEEE Trans. Ind. Electron.* **2014**, *61*, 6418–6428. [\[CrossRef\]](#)
27. Bristow, D.A.; Tharayil, M.; Alleyne, A.G. A survey of iterative learning control. *IEEE Control Syst. Mag.* **2006**, *26*, 96–114.
28. Xu, J.X.; Huang, D.; Pindi, S. Optimal tuning of PID parameters using iterative learning approach. *SICE J. Control Meas. Syst. Integr.* **2008**, *1*, 143–154. [\[CrossRef\]](#)
29. Nocedal, J.; Wright, S. *Numerical Optimization*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2006; pp. 30–65.
30. Solomon, R.; Van Hemmen, J.L. Accelerating backpropagation through dynamic self-adaptation. *Neural Netw.* **1996**, *9*, 589–601. [\[CrossRef\]](#)
31. Salomon, R. Evolutionary algorithms and gradient search: Similarities and differences. *IEEE Trans. Evol. Comput.* **1998**, *2*, 45–55. [\[CrossRef\]](#)
32. Hansen, N.; Atamna, A.; Auger, A. How to assess step-size adaptation mechanisms in randomised search. In Proceedings of the International Conference on Parallel Problem Solving from Nature, Ljubljana, Slovenia, 13–17 September 2014; Springer: Berlin/Heidelberg, Germany, 2014; pp. 60–69.
33. Hansen, N. The CMA evolution strategy: A tutorial. *arXiv* **2016**, arXiv:1604.00772.
34. Hansen, N. CMA-ES with two-point step-size adaptation. *arXiv* **2008**, arXiv:0805.0231.
35. McKeever, S.D. *Path Planning for an Autonomous Vehicle*; Technical Report; Massachusetts Inst of Tech Cambridge: Cambridge, MA, USA, 2000.
36. Fossen, T.I. *Handbook of Marine Craft Hydrodynamics and Motion Control*; John Wiley & Sons: Hoboken, NJ, USA, 2011.

37. Myre, H. Collision Avoidance for Autonomous Surface Vehicles Using Velocity Obstacle and Set-Based Guidance. Master's Thesis, NTNU, Taipei, Taiwan, 2016.
38. Bouwmeester, E.; Heemink, A. Optimal line spacing in hydrographic survey. *Int. Hydrogr. Rev.* **1993**, *70*. Available online: <https://journals.lib.unb.ca/index.php/ihr/article/download/23240/27015> (accessed on 13 January 2020).



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).