

Research Article

Underwater Environment SDAP Method Using Multi Single-Beam Sonars

Zheping Yan,¹ Dongnan Chi,¹ Shuping Hou,² and Yalin Zheng³

¹ College of Automation, Harbin Engineering University, Harbin, Heilongjiang 150001, China

² College of Mechanical and Electrical Engineering, Harbin Engineering University, Harbin, Heilongjiang 150001, China

³ Department of Eye and Vision Science, University of Liverpool, L69 3GA Liverpool, UK

Correspondence should be addressed to Zheping Yan; yanzheping2013@163.com

Received 22 August 2013; Accepted 30 August 2013

Academic Editor: Zhiguang Feng

Copyright © 2013 Zheping Yan et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

A new autopilot system for unmanned underwater vehicle (UUV) using multi-single-beam sonars is proposed for environmental exploration. The proposed autopilot system is known as simultaneous detection and patrolling (SDAP), which addresses two fundamental challenges: autonomous guidance and control. Autonomous guidance, autonomous path planning, and target tracking are based on the desired reference path which is reconstructed from the sonar data collected from the environmental contour with the predefined safety distance. The reference path is first estimated by using a support vector clustering inertia method and then refined by Bézier curves in order to satisfy the inertia property of the UUV. Differential geometry feedback linearization method is used to guide the vehicle entering into the predefined path while finite predictive stable inversion control algorithm is employed for autonomous target approaching. The experimental results from sea trials have demonstrated that the proposed system can provide satisfactory performance implying its great potential for future underwater exploration tasks.

1. Introduction

Underwater exploration often encounters environment that is difficult or even impossible for humans to access due to their physical constraints such as deep depth, narrow spaces, and severe working conditions. Unmanned underwater vehicle has a number of advantages for exploring underwater environments, such as autonomous control ability and self-sufficient energy supply. Autopilot of UUV often relies on the information or characteristics (e.g., geometrical information) of the surrounding environment, reflected by data collected from sensors such as sonars.

When in operation, sonar sends out an acoustic beam and the returned (usually the fastest) beam from the environment is collected to determine the distance and location of the environment. This means that it can detect the point on the contour that has the shortest distance from the sonar. Therefore, sonar data can be used to plan the desired path of the UUV and control it by changing thruster forces and rudder angles of the UUV to approach the target. An important issue for designing UUV control systems is the strength of

the signal observed from sonar. It is weak primarily due to the random effect caused by complicated marine disturbance; other interferences between received beams can be due to delay and scattering effect.

In this paper a new autopilot system, known as simultaneous detection and patrolling (SDAP), is proposed to address this challenge. Autonomous guidance and control are implemented synchronously where the reconstructed environment contour is used as the guidance path for UUV navigation. For the environment contour reconstruction, the major focus of research is on simultaneous localization and mapping (SLAM) [1], where navigation is the key issue to be addressed. With the advances in control theory, UUV control systems have rapidly evolved from classic control theory to modern control models, including PID [2, 3], backstepping [4–7], fuzzy theory [8, 9], neural network [10–13], sliding model [14, 15], model prediction control [16, 17], and feedback linearization [18–23]. In particular, Zou has proposed an optimal inversion-based output tracking approach for the guidance of a vertical takeoff and landing (VTOL) aircraft problem [21]. Song has further improved this approach

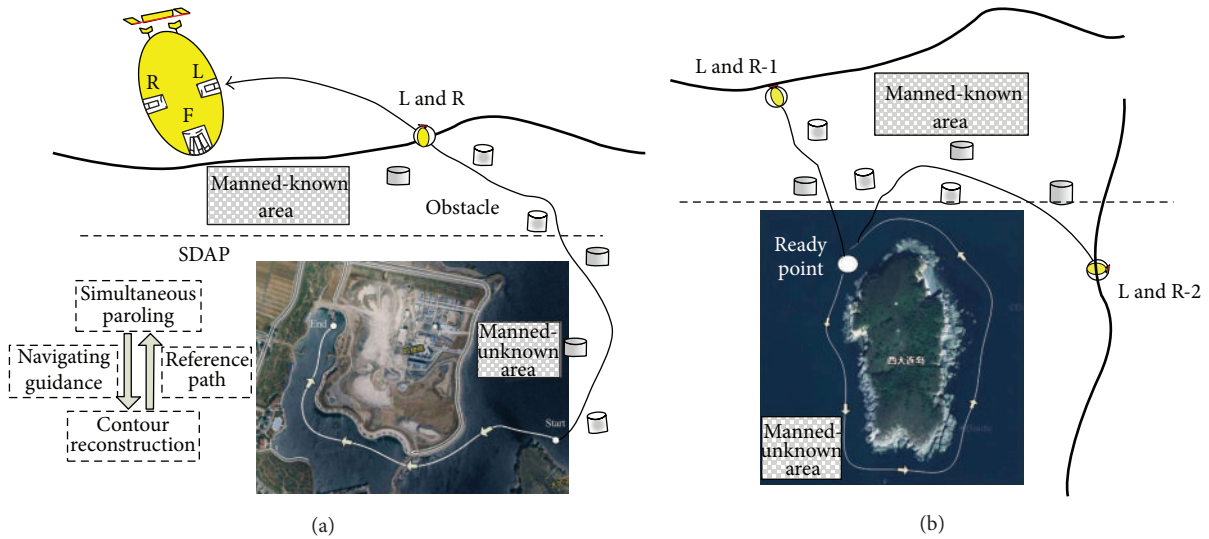


FIGURE 1: Illustration of the proposed SDAP system for different environments.

with better convergence property such that a second order convergence can be achieved even with aggressive trajectories and strong nonlinearities [20]. Given that stable inversion technique has shown excellent performance for achieving stable control inputs, it is chosen to be implemented in the controllers for accuracy, efficiency, and cost effectiveness.

In this paper autopilot of UUV for both closed and open environments is considered, as shown in Figure 1. In the closed port (Figure 1(a)), there is only one entrance and the UUV has to be able to navigate to the only ready point from any launch and recovery position (L&R). For the open island (Figure 1(b)), there are theoretically infinite numbers of possible entrance points around the island. With consideration of the disturbances such as current direction, an appropriate ready point has to be chosen to guide the UUV to enter into the manned-unknown area. In the manned-known areas, accurate predefined path is applied for the UUV to follow in order to reduce the cost and risks. The main issue addressed in the paper is the navigation using autonomous guidance and control is implemented in manned-unknown areas or area inaccessible for humans where the contour has to be reconstructed in real time. Although in reality, the environment is semiclosed where there is more than one entrance, it is not discussed explicitly here as it can be addressed by applying the strategy for closed environment in an iterative manner.

The main novelties of this paper are as follows.

- (1) To address the weak sonar data, a wavelet transform is applied to preprocess the original data so as to eliminate possible outliers in the original sonar data. To reduce the information loss in the preprocessing, the wavelet coefficient values of the current time are estimated using those coefficients estimated in the past times and the original sonar data. All the estimated wavelet coefficients are then regarded as the data resource for the contour reconstruction processes.

- (2) A support vector clustering (SVC) inertia algorithm is proposed to cluster the data into different classes so as to determine the property of the original sonar data and obtain the boundaries of the classes (also known as contour). The resulting contour after this step is composed of successively connected lines.
- (3) To satisfy the inertia property of the UUV path, the initial contour is smoothed using different order Bézier curves which are automatically determined by the local properties of the structural environments.
- (4) With the information of the smoothed contour as the reference path and a predefined safety distance, an improved inversion algorithm, finite predictive stable inversion is proposed to control vehicle navigation.

The remainder of this paper is organized as follows: in Section 2, weak observable sonar data is preprocessed using the wavelet transform and represented as a collection of wavelet coefficients. In Section 3, the wavelet coefficients together with the data confidence limit and DVL information are used to estimate the contour of the environment structure by a support vector clustering inertia method. Addressing the inertia requirement of the vehicle, in Section 4 different order Bézier curves are introduced to smooth the initial contour and automatic decision strategy is made to generate the reference path according to the local properties of the structural environment. In Section 5 details on the controller design methods are presented, and employed for predictive trading during detection mission. The validation experiment design and results are presented in Section 6. Section 7 concludes this paper with some insightful discussions.

2. Preprocessing of Sonar Data Using Wavelet Transform

2.1. Collection of Sonar Data. In this paper five single-beam sonars are configured on the vehicle in order to automatically

TABLE I: Metrics to categorize obstacles.

E_t	$E_{\Delta t}$	L_t change	Class
1	1	No	TO
1	1	Yes	GLTO
1	0	No contour	Noise or SLTO

detect the underwater environment: three sonars in the front to detect local environment characteristics and two on the left and the right side for contour reconstruction. The three front sonars are deployed in a way where the middle one is located on the axis of the vehicle surge direction and the other two are installed on the left and right of the middle one with an angle of 7.5 degree, respectively.

Data collected from sonars often include useful data describing obstacles and noise from random outliers. To address the “false alarm” problem in vehicle navigation, those outliers have to be detected and eliminated as much as possible. Outliers can be divided into noise patches and objects (e.g., fish swarm) that have less or no threats to the vehicle. Therefore, three different types of objects are defined as follows.

- (a) *Threaten Obstacle (TO)*. They are obstacles existing in the environment that can be detected by sonar and have threats to the vehicle, including wreck, reef, and iceberg.
- (b) *Low Threaten Obstacle (LTO)*. They are obstacles existing in the environment that can be detected by sonar but have low level of threat to the vehicle, such as suspensions, and fish swarms. LTO can be further fractionized to single LTO (SLTO) distributing as single isolated objects and group LTO (GLTO) with unfixed contour.
- (c) *Noise*. Data is collected from sonar that denotes non-obstacles.

With the above definitions, data describing TO is considered to be useful while LTO and noise are regarded as outliers in this paper. Let E_t and $E_{\Delta t}$ be data distribution at instants t and $(t + \Delta t)$, respectively, and the values of 1 and 0 describe data existence and nonexistence, respectively; L_t express the contour of data class at time t , and the sonar data can be classified as Table I.

2.2. Preprocessing of Weak Observable Data Using Wavelet Transform Modulus Maxima. Outliers mixed in the dataset can be described as singularities by estimating local Lipschitz exponent using wavelet transforms. Defined either at a certain time instant or in an interval, Lipschitz exponent can be calculated by numerical methods. A wavelet transform modulus maximum (WTMM) is introduced to preprocess the sonar data.

Let $x(t)$ and $\psi(t)$ be the sonar data and a certain function (introduced next) at instant t , respectively; the wavelet transform can be defined as $Wx(b, a) = (1/\sqrt{a}) \int_{-\infty}^{\infty} x(t) \cdot \overline{\psi((t - b)/a)} dt$, where $\overline{}$ is complex conjugation and a and b describe scale element and shift coefficient, respectively.

Mathematically the local regularity indicated by Lipschitz exponent is the precondition for data reconstruction using wavelet transform. Owing to the relationship between the WTMM and Lipschitz exponent, the pattern of change in WTMM at different scales is of great importance for the preprocessing of weak data. At a certain scale, if the maximum modulus exists at some time point, search along the scale decrease direction within the cone of influence will find a singular point or a peak point close to the zero scale, which can be determined by the Lipschitz exponent.

2.2.1. Estimation of Wavelet Coefficients and Compensation of Lost Data. In order to remedy the eliminated sonar data and to guarantee the continuity of the reconstructed contour, the wavelet coefficient estimation method proposed by Liu and Mao [24] is used. More specifically, the original sonar data in previous five instants and the wavelet coefficients in the previous six instants are used for the estimation as follows:

$$\begin{aligned}
Wx(kT, f) = & \sqrt{fT} \{ \delta_1 x[(k-1)T, f] \\
& + \delta_2 x[(k-2)T, f] \\
& + \delta_3 x[(k-3)T, f] \\
& + \delta_4 x[(k-4)T, f] \\
& + \delta_5 x[(k-5)T, f] \\
& - \lambda_1 Wx[(k-1)T, f] \\
& - \lambda_2 Wx[(k-2)T, f] \\
& - \lambda_3 Wx[(k-3)T, f] \\
& - \lambda_4 Wx[(k-4)T, f] \\
& - \lambda_5 Wx[(k-5)T, f] \\
& - \lambda_6 Wx[(k-6)T, f] \}, \tag{1}
\end{aligned}$$

where

$$\begin{aligned}
\varepsilon &= e^{-fT(\sigma - iw_0)}, \\
\lambda_1 &= -6\varepsilon, \\
\delta_1 &= \left[\frac{(\sigma f T)^3}{3} \right. \\
&\quad \left. - \frac{(\sigma f T)^4}{4} + \frac{(\sigma f T)^5}{5} \right] \varepsilon, \\
\delta_2 &= \left[\frac{2(\sigma f T)^3}{3} - \frac{5(\sigma f T)^4}{3} \right. \\
&\quad \left. + \frac{26(\sigma f T)^5}{15} \right] \varepsilon^2, \\
\lambda_2 &= 15\varepsilon^2, \\
\lambda_3 &= -20\varepsilon^3,
\end{aligned}$$

$$\begin{aligned}
\delta_3 &= \left[\frac{22 \cdot (\sigma f T)^5}{5} - 2(\sigma f T)^3 \right] \varepsilon^3, \\
\delta_4 &= \left[\frac{2 \cdot (\sigma f T)^3}{3} + \frac{5 \cdot (\sigma f T)^4}{3} \right. \\
&\quad \left. + \frac{26 \cdot (\sigma f T)^5}{15} \right] \varepsilon^4, \\
\lambda_4 &= 15\varepsilon^4, \\
\lambda_5 &= -6\varepsilon^5, \\
\lambda_6 &= \varepsilon^6, \\
\delta_5 &= \left[\frac{(\sigma f T)^3}{3} + \frac{(\sigma f T)^4}{6} \right. \\
&\quad \left. + \frac{(\sigma f T)^5}{15} \right] \varepsilon^5.
\end{aligned} \tag{2}$$

2.2.2. Confidence Limit of a Single-Beam Sonar Data. Confidence limit is introduced to assess the degree of match between the wavelet coefficients and the model calculated using the data. Hypothesis test is used to estimate this confidence limit.

Assume that M is the calculated model built using Hidden Markov Model [24]; the null hypothesis H_0 and alternative hypothesis H_1 are as the follows.

H_0 : WT is valid data and equals the data calculated using the model. This means the original sonar data are the true signal from threatening obstacles.

H_1 : WT is an outlier and its corresponding original sonar data are from either low threatening obstacles or noise.

The tracking evaluation function is introduced to estimate the confidence limit,

$$Ev(WT, H_0, H_1) = \frac{P\{WT | H_0\}}{P\{WT | H_1\}}, \tag{3}$$

where $P\{WT | \cdot\}$ is the probability to obtain the same model data with WT in corresponding hypothesis.

If T_{\min} is defined as the acceptable minimum threshold, the ratio between the evaluation function value and T_{\min} is used to determine the class property of the data; see Table 2.

3. Initial Contour Reconstruction

The preprocess sonar data by using the wavelet transform can locally amplify abnormal signals or outliers. This observation helps detect and eliminate potential outliers in order to reconstruct the contour of environmental structure. Support vector clustering (SVC) inertial algorithm was used to achieve the initial contour reconstruction.

TABLE 2: Determination of class types.

Ratio	Class type
≥ 1	Outliers
< 1	Valid data

TABLE 3: Criterion of clustering.

Class	Condition	Conclusion
Nonbounded support vector	$\vartheta_i = 0, 0 < \alpha_i < P$	Class contour
Bounded support vector	$\vartheta_i > 0, \alpha_i = P$	Outliers
Data class	$\alpha_i = 0$	Class data

3.1. Structural Environment Construction Using SVC Algorithm. The main idea of SVC is to project sonar data $\{x_1, x_2, \dots, x_N\}$ into high-dimension hypersphere space with a minimum radius using nonlinear mapping $H = \{\Phi(x_i) | 1 \leq i \leq N\}$. For the mapping data in hypersphere, it holds that $\|\Phi(x_i) - a\|^2 \leq R^2$ where a and R denote the center and radius of the sphere. The objective function is described as [25],

$$\max \left\{ \sum_{i=1}^N K(x_i, x_i) \alpha_i - \sum_{i=1, j=1}^N \alpha_i \alpha_j K(x_i, x_j) \right\}. \tag{4}$$

In high dimension space, the distance from x to the sphere center a , $R^2 = \|\Phi(x) - a\|^2$ is adapted as

$$R^2 = K(x, x) - 2 \sum_{i=1}^N \alpha_i K(x, x_i) + \sum_{i=1, j=1}^N \alpha_i \alpha_j K(x_i, x_j), \tag{5}$$

where hypersphere radius $R = \sum_i R(\alpha_i^*)/N$, a support vector α_i^* , and the number of support vector N . Class bound of data-set will be collected

$$\{x | R(x) = R\}. \tag{6}$$

Outliers are defined as

$$\{x | R(x) > R\}. \tag{7}$$

After the clustering, the sonar data are separated into three classes according to the position with respect to the class bound, shown in Table 3. ϑ and P are the relaxation factor and penalty coefficient to balance the performance.

Adjacency matrix $A = (A_{ij})_{N \times N}$ is included to determine type of classes with the distance as its elements:

$$A_{ij} = \begin{cases} 1 & \forall t \in \text{seg}(x_i, x_j), R(t) \leq R \\ 0 & \text{other,} \end{cases} \tag{8}$$

where $\text{seg}(x_i, x_j)$ expresses the line segment between x_i and x_j . If the line is either inside or outside the hypersphere, x_i and x_j are then attributed to the same class. A three-step implementation is as follows.

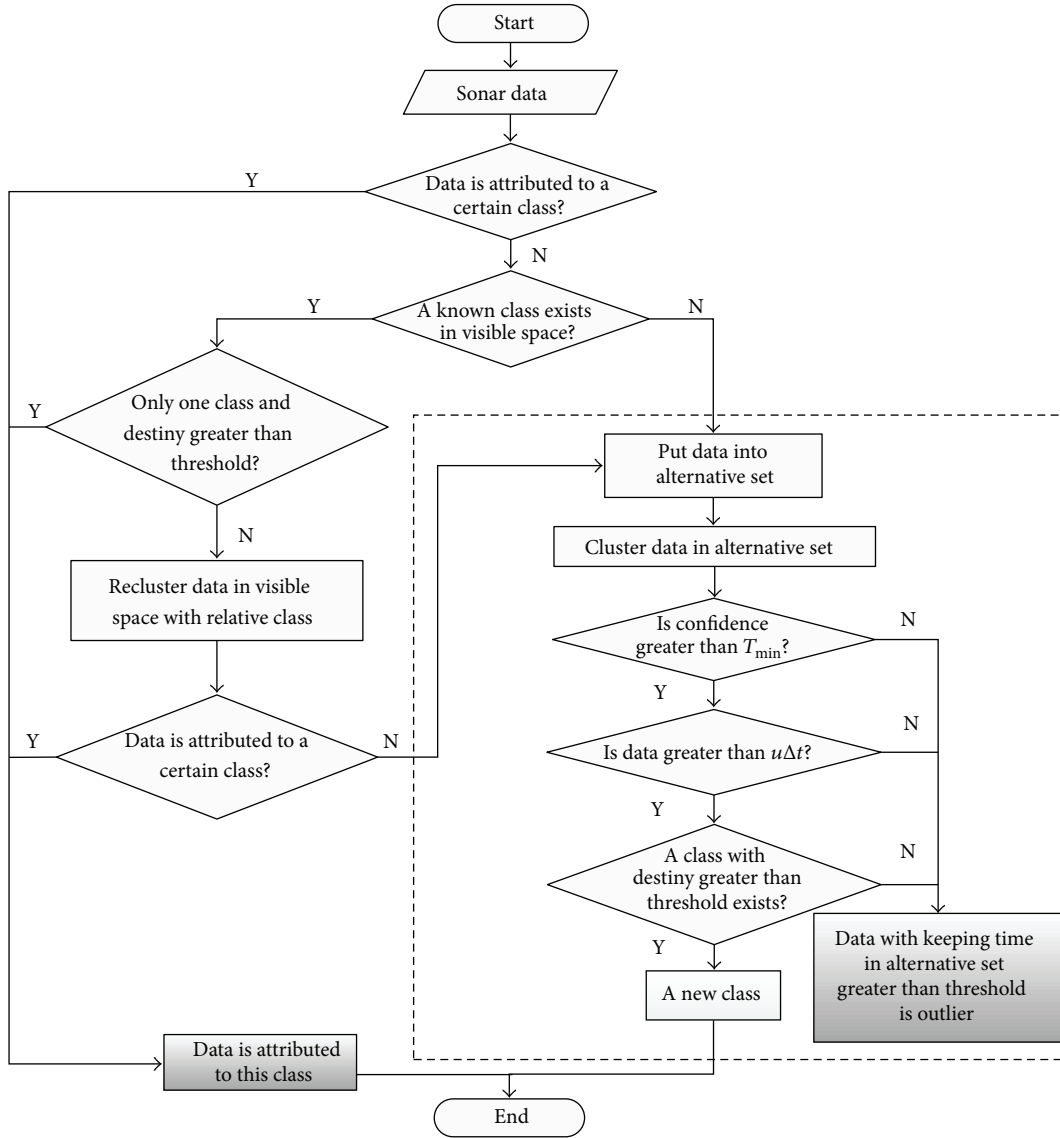


FIGURE 2: Flow chart of outlier inertia detection for preprocessed data.

Step 1. k -average interpolation is applied on the line between x_i and x_j .

Step 2. Compute $x = \|x_i - x_j\| \cdot l/k$, and let Δx be regarded as the determinant distance and l is the number of lines inside or on the hypersphere.

Step 3. If $x \geq \Delta x$, x_i and x_j are attributed to the same class; otherwise they are from different classes and will have to be further differentiated in the next subsection.

3.2. Inertia Algorithm for Improving Construction. Inertia algorithm is a common action delay method proposed to avoid unnecessary actions of the UUV in order to escape from potentially threatening obstacles [25]. This idea is adopted here to determine the classes of those data that has not been successfully classified. Outliers determined by the SVC algorithm are regarded as candidate outliers. Inertia method is used to determine the class label of those data that have

not been explicitly classified. After this step the sonar data are separated into three classes: data class, class bound data, and outliers by using inertia algorithm. The flowchart of inertia algorithm is illustrated in Figure 2.

Let k and ρ_T , respectively, denote the number of classes and a preset minimum constant threshold. $\forall i \geq 1$, if it is satisfied that

$$\exists \bigcup_k \{x_k \mid k > \rho_T\} \subseteq C_i. \quad (9)$$

C_i is regarded as a *known class*. An alternative set is introduced to temporarily place preoutliers, denoted as *AIS*.

The conditions for classifying the sonar data into the alternative set can be described as follows.

- (1) There is not any known class.

At the beginning of path planning, there is not enough sonar data to be clustered and to clearly indicate any obstacles.

The data x is placed into alternative set AIS . If there is more sonar data in the visible range of x and the density increases to ρ_T , it means that the dataset constitutes the first class C_1 . If the density is still lower than ρ_T with time evolution of Δt , x will be determined as LTO or noise and will not be used to reconstruct the contour of the environment.

- (2) If any C_i exists with $i \geq 1$, $i \in \mathbf{N}^+$, the class may include class data, class bound data, and/or outliers and thus cannot be directly determined. Let X be an arbitrary dataset and described as

$$X = \left\{ \bigcup_{j=1}^m x_j \mid m < \rho_T, x_j \notin C_{\text{any}} \right\}, \quad (10)$$

where C_{any} denotes any known class. It implies that once the destiny of sonar data is less than ρ_T , it will be placed into AIS for further assessment.

Assume that N_c is the number of known classes, and Class set C is defined as

$$C = \bigcup_{i \in M} C_i = \{x : \exists i \in N_c, \text{ with } x \in C_i\}. \quad (11)$$

Outlier set is symbol by O , including the outliers and the data with density less than the preset threshold ρ_T .

Criterion 1. For any sonar data x , if it is satisfied that $x \in L_i$, $1 \leq i \leq M$, where L_i is the bound of known class C_i , we will have $x \in C_i$.

For arbitrary data x , Criterion 1 is utilized to identify whether it belongs to any known class. If it just falls into bound L_i , x is assigned to C_i directly and $x \in C_i$; otherwise, go to the next criterion. In order to determine the properties of the data, visible space is defined as follows.

Visible space is an artificial sphere space centered at x with a radius of r . It is used to estimate distribution density near x . In the visible space, if the density of sonar data is larger than a predefined threshold, we can determine that it is a valid class.

Criterion 2. $\forall x, x \notin C_i$, x does not belong to any known class. Assume that $V = \{x \mid |x - x_a| < r\}$, with x_a as a point in the visible range of x . If the number of data satisfying the condition $x_a \in C_i$ is not less than ρ_T , then C_i is a known class in x 's visible range.

Criterion 3. If the number of data in C_i is larger than ρ_T , and C_i is the unique known class of x , it can be ascertained that $x \in C_i$ directly.

To determine the property of any x that has not been successfully classified by using Criterion 1, Criterion 3 is important to ensure whether it can be assigned to a unique known class. If it fails, reclustering has to be performed.

It is observed that the objects for reclustering C_{Re} only contain the classes including the data point and those in its visible range, and therefore the effectiveness of algorithm can be increased:

$$C_{\text{Re}} = \{x \mid x \notin C_i \cap |x - x_a| < r\} \cup \{x_a \mid x_a \in C_i\}. \quad (12)$$

Sonar data x with unknown class in its visible range will be located in the alternative set AIS . Similarly, those data that can still not be classified by reclustering are also placed into AIS for further analysis.

Criterion 4. $\forall 1 \leq i \leq M, \exists x_i \in AIS$, SVC algorithm is applied to cluster and class C_r is obtained. If the density ρ_R of C_r is larger than ρ_T , it is considered that $C_r \subseteq C$. The total number of M increases by 1. Otherwise, data x (or dataset X) exist which satisfies the following condition:

$$\{x \mid t > t_T\} \subseteq O \quad (13)$$

$$\{X \mid \text{Total number of } X < \rho_T, t > t_T\} \subseteq O,$$

where t shows the time related to the data x (or set X).

In environmental structure detection, data gradually accumulates. Through preprocessing using wavelet transform and clustering by SVC inertia algorithm, an initial contour can be reconstructed with data class bound as reference. The initial contour consists of several successively connected lines.

4. Smoothing Initial Contour Based on Local Environment Characters

4.1. Extraction of Local Environmental Characteristics. Based on the assumption that the data collected from the 5 sonars are accurate, three different local environmental characteristics can be determined, and they are described as follows.

4.1.1. Line Path. Figure 3 shows two scenarios where the vehicle is located on the right (Figure 3(a)) and left side (Figure 3(b)) of a local linear environment, respectively. l_1, l_2 , and l_3 denote the data collected from the left, middle, and right sonars in the front of the vehicle, respectively, while l_l and l_r are the data collected from the left and right side sonars.

If the left (resp., right) side sonar data is valid and the right (resp., left) one shows maximum effective distance, then the vehicle is on a line path and on the right (resp., left) side of the contour. Remark: energy carried with an UUV is often limited. When there is no obstacle on one side of the vehicle, the sonar on that side can be turned off in order to conserve energy and extend the working time for the mission.

4.1.2. Narrow Path. If the distance between the environment contours on both sides of the path is not wide enough for the vehicle to turn safely, the vehicle is in a narrow path. Figure 4 illustrates this scenario where the distances l_{12} and l_{23} are calculated by using the data from the three front sonars. If the sonar data from the left and right side are effective and $l_{12} + l_{23} < 2d_c$ where d_c is the predefined minimum turning radius, the vehicle is in a narrow turning path.

Proof. The triangle inequality theorem states that any one side of a triangle is always shorter than the sum of the other two sides. For the triangle with l_{12}, l_{23} , and l_{13} as its sides, it holds that

$$l_{12} + l_{23} > l_{13}. \quad (14)$$

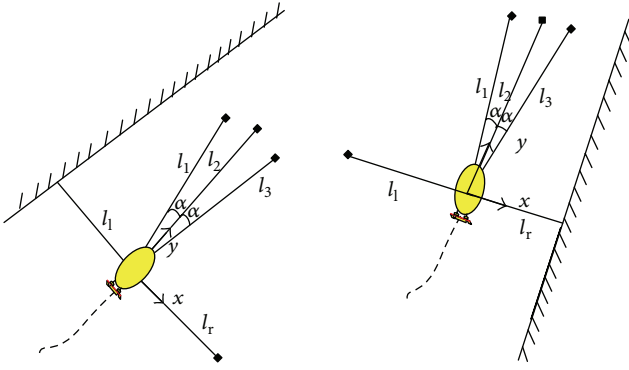


FIGURE 3: Diagram illustrating line path.

Let l_0 be the line connecting the two sides of the contours; it also holds that

$$l_{13} \geq l_0. \quad (15)$$

Then we have

$$l_0 \leq l_{13} < l_{12} + l_{23}. \quad (16)$$

If $l_{12} + l_{23} < 2d_c$ is satisfied, it holds that $l_0 < 2d_c$. That is, the line is smaller than the turning radius, and the vehicle cannot steer out of this narrow turning environment by normal turning motion. Theorem holds. \square

4.1.3. Regular Turning Path. The determination of regular turning path is similar to the narrow turning path. If the left side sonar data is effective, the vehicle in regular turning path can be separated into two situations according to the data from the 3 sonars at the front; see Figure 5. The left front sonar beam is located on the previous path, with the middle and right ones being located on the rear path in Figure 5(a). The difference between (b) and (a) is the left and middle sonar beam located on the previous path, with the right sonar beam being on the rear path. The stage of UUV turning is reflected. If the sonar data on one side is effective and the other side is ineffective or has the maximum value (this is different to narrow turning path), it can be determined that the vehicle is in a regular turning path.

4.2. Smoothness of the Local Environmental Contour. The initial contour reconstructed by using the SVC inertia algorithm comprises lines that are successively connected. Due to the inertia property of the underwater vehicle, this cannot be used as the reference path for tracking. In this paper, Bézier curve is introduced to smooth the initial contour in order to extract a reference path that can be used for navigation.

Given a set of control points $P_0, P_1, \dots, P_n, \forall t \in [0, 1]$, n th-order Bézier curve can be defined as $B(t) = \sum_{i=0}^n P_i b_{i,n}(t)$, with Bernstein Polynomial $b_{i,n}(t) = \binom{n}{i} t^i (1-t)^{n-i}$, $i = 0, 1, \dots, n$. Different order Bézier curves are chosen to smooth the initial contour. To determine the curve orders, control points are located first. Considering navigation requirement and the contour characteristics, it is intrinsic that the starting

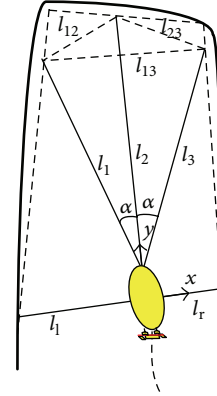


FIGURE 4: Diagram illustrating narrow turn path.

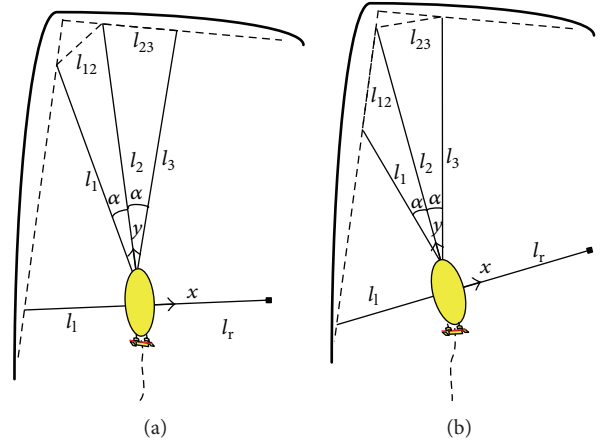


FIGURE 5: Diagram illustrating normal turn paths.

point, end point, and intersection point between turning paths should be the control points. Moreover, the width of the local environment is another significant issue to be considered for the order of Bézier curve.

Based on Section 4.1, the local environmental characteristics are separated into line and turning paths, and then the orders of different characteristics are classified. It has been proved that 2nd-order Bézier curve is sufficient for turning path restriction [26], and as such Bézier curves used here must satisfy C^2 continuity condition.

C^k continuity condition: Bézier curves $P(t)$ and $Q(t)$ in C^k continuous at t_0 are,

$$P(t_0) = Q(t_0)$$

$$\dot{P}(t_0) = \dot{Q}(t_0)$$

$$\vdots$$

$$P^{(k)}(t_0) = Q^{(k)}(t_0).$$

(17)

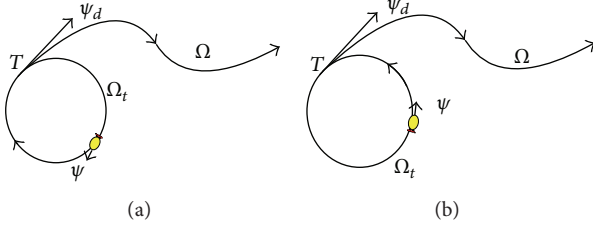


FIGURE 6: Rolling path with different orientations to find initial points on reference path.

Based on the definition of C^k continuity condition, smoothness conditions for n segments can be shown as

$$\begin{aligned} {}^{i-1}P_{n_{i-1}} &= {}^iP_0 \\ n_{i-1} \left({}^{i-1}P_{n_{i-1}} - {}^{i-1}P_{n_{i-1}-1} \right) &= n_i \left({}^iP_1 - {}^iP_0 \right) \\ n_{i-1} (n_{i-1} - 1) \left({}^{i-1}P_{n_{i-1}} - 2 \cdot {}^{i-1}P_{n_{i-1}-1} + {}^{i-1}P_{n_{i-1}-2} \right) & \\ &= n_i (n_i - 1) \left({}^iP_2 - 2 \cdot {}^iP_1 + {}^iP_0 \right), \end{aligned} \quad (18)$$

where $i = 2, 3, \dots, 2N - 3$, and n_i is the number of control points for the i th Bézier subcurves. The first and last segments are special cases and their orders are discussed first and followed by those ones in the middle.

4.2.1. First and Last Segments. For the first and last segments, ${}^1B(t)$ and ${}^{2N-3}B(t)$, with the known control points $W_1 = {}^1P_0$ and $W_N = {}^{2N-3}P_4$. To find the orders of these curves, the following must hold.

Condition 1. The heading ψ_0 and ψ_f at control points W_1 and W_N must be guaranteed as follows:

$$\begin{aligned} {}^1P_1 &= W_1 + c_0 [\cos \psi_0 \quad \sin \psi_0]^T, \quad c_0 \in \mathbf{R}^+, \\ {}^{2N-3}P_3 &= W_N - c_f [\cos \psi_f \quad \sin \psi_f]^T, \quad c_f \in \mathbf{R}^+, \end{aligned} \quad (19)$$

where c_0 and c_f are constant parameters. To satisfy Condition 1, 1P_0 and 1P_1 of the first segment, as well as ${}^{2N-3}P_3$ and ${}^{2N-3}P_4$ of the final segment must be known.

Condition 2. C^2 continuity condition with contiguous subcurves must be guaranteed. Therefore, the points ${}^{i-1}P_{n_{i-1}}$, ${}^{i-1}P_{n_{i-1}-1}$, and ${}^{i-1}P_{n_{i-1}-2}$ should be known in (18).

Given all the above conditions, the total number of the control points required to be known is 5 for the first and last segment, respectively. This implies that 4th-order Bézier curves are needed for both of them.

4.2.2. Middle Segments. To guarantee C^2 continuity condition, for each middle segment it has 6 control points in total: three control points, respectively, from the previous segment (including iP_2 , iP_1 and iP_0) and the segment after (including ${}^{i-1}P_{n_{i-1}}$, ${}^{i-1}P_{n_{i-1}-1}$, and ${}^{i-1}P_{n_{i-1}-2}$) are known (see (30)). Therefore, the order of Bézier subcurves is 5 for the middle segments,

TABLE 4: Orders of Bézier curve.

Local environment character	Order of Bézier curve
Turning	2nd-order
Start and final segments	4th-order
Middle segments	5th-order

${}^iB(t)$, $i = 3, 5, \dots, 2N - 5$. Table 4 summarizes the order selection for different local environment characters.

5. Path Tracking Control

Two types of feedback linearization methods are used for path tracking control: differential geometry feedback linearization and stable inversion, which can be used for exactly automatic target approaching in known region and contour reconstruction in unknown region, two stages of the navigating process. This will be detailed in this section.

The path tracking control model for a UUV can be described with state vectors as follows:

$$\begin{aligned} \dot{x}(t) &= f(x) + g(x) \mu(t), \\ y(t) &= [y_1 \ y_2 \ y_3]^T = h(x(t)), \end{aligned} \quad (20)$$

where $x(t) = [x \ y \ \psi \ u \ v \ r]^T \in \mathbf{R}^6$ denotes states, $\mu(t) = [\tau_u \ \tau_r]^T \in \mathbf{R}^2$ denotes input variable matrix, mapping $f(\cdot) : \mathbf{R}^6 \rightarrow \mathbf{R}^6$, $g(\cdot) : \mathbf{R}^6 \rightarrow \mathbf{R}^{6 \times 2}$, and $h(\cdot) : \mathbf{R}^6 \rightarrow \mathbf{R}^3$ is smooth enough. f and g are nonlinear items and input coefficient items given in the following:

$$\begin{aligned} f &= \begin{bmatrix} u \cos \psi - v \sin \psi + \tau_{du} \\ u \sin \psi + v \cos \psi + \tau_{dv} \\ r + \tau_{dr} \\ p_1 vr + p_2 u \\ p_4 ur + p_5 v \\ p_7 uv + p_8 r \end{bmatrix}, \\ g &= \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ p_3 & 0 \\ 0 & 0 \\ 0 & p_9 \end{bmatrix}, \end{aligned} \quad (21)$$

where $p_1 = m_{22}/m_{11}$, $p_2 = (-X_u - X_{u|u}|u|)/m_{11}$, $p_3 = 1/m_{11}$, $p_4 = m_{11}/m_{22}$, $p_5 = (-Y_v - Y_{v|v}|v|)/m_{22}$, $p_6 = 1/m_{22}$, $p_7 = (m_{11} - m_{22})/m_{33}$, $p_8 = (-N_r - N_{r|r}|r|)/m_{33}$, $p_9 = 1/m_{33}$, $m_{11} = m - X_{\dot{u}}$, $m_{22} = m - Y_{\dot{v}}$, and $m_{33} = I_z - N_{\dot{r}}$. m is the mass of the UUV; $X(\cdot)$, $Y(\cdot)$, and $N(\cdot)$ are the derivatives of the hydrodynamic coefficients related to the added mass; I_z denotes the initial moment of the UUV under body coordinate system, τ_u and τ_r are the input force, and moment τ_{du} , τ_{dv} , and τ_{dr} are the disturbance at those corresponding directions.

5.1. Autonomous Arriving Based on Differential Geometry Feedback Linearization. The UUV is diving underwater with arbitrary states (including heading and position) at any point

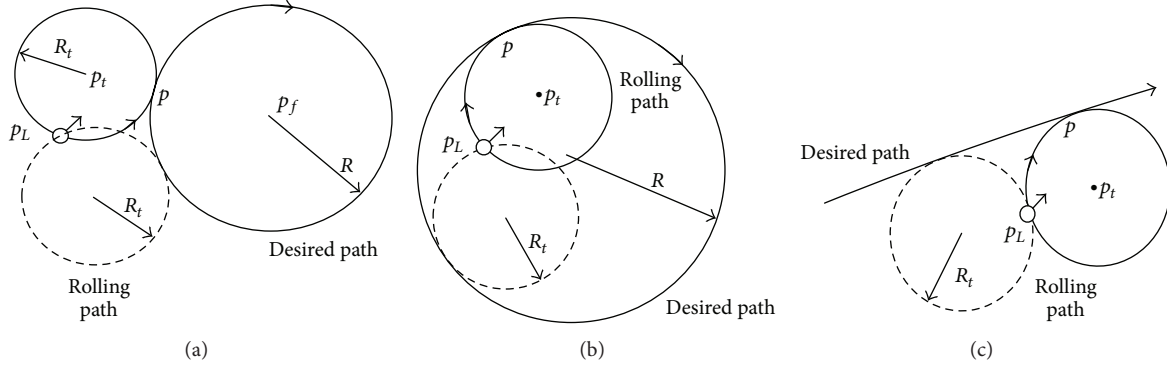


FIGURE 7: Rolling path generation for three different situations. (a) UUV is outside circle path. (b) UUV is outside circle path. (c) UUV is outside line path.

and has to be able to navigate towards a preset target near the structural environment. This is known as *Autonomous Arriving*. A differential geometry feedback linearization control algorithm with rolling path guidance is proposed to achieve accurate tracking. Figure 6 illustrates how to find the initial points: circle arc is used to represent rolling path according to the initial heading and orientation of the vehicle, Ω and Ω_t represent the desired path and rolling path, respectively, ψ and ψ_d are the current heading direction and desired heading direction, respectively, and T denotes the tangent point of Ω and Ω_t , which is the initial point on Ω for the vehicle. Note, the UUVs in Figure 6 have the same initial positions but different headings (upward and downward in Figures 6(a) and 6(b), resp.). Given the desired orientation and the inertia of the vehicle, the vehicle is unable to turn at T in the scenario as shown in Figure 6(b). Further analysis is required and detailed as follows.

Let model output be $\eta = [\eta_1 \ \eta_2]^T$, η_1 is the distance of the chosen path, and $\eta_2 = \psi$.

It has been proved that any nonlinear path can be reconstructed by circular arcs and lines [27]. Therefore, with circle and line as example, rolling paths are generated as the desired path for the guidance of the UUV.

- (1) Circle Path. Assume that $p_f(x_f, y_f)$ is the center of a circle path and the radius of the circle path R_f is a constant; then the output can be given as

$$\eta_1 = \sqrt{(x - x_f)^2 + (y - y_f)^2} - R, \quad (22)$$

where $p_L(x, y)$ is the actual initial position of the vehicle. From the equation above, $|\eta_1|$ is the minimum distance between p_L and the circle path.

According to the position between the desired path and the current position of the vehicle, radius of a rolling path can be derived as follows:

- (a) when the initial position of the UUV is outside of the circular path (see Figure 7(a)) and ω is a constant

$$R_t = \begin{cases} \eta_1 + \omega R & \eta_1 > \frac{R}{2} \\ R & \text{others,} \end{cases} \quad (23)$$

- (b) when the initial position of the UUV is inside of the circular path (see Figure 7(b))

$$R_t = \begin{cases} \frac{\eta_1 + 2R}{2} - \omega R & p_L \neq p_f \\ \frac{R}{2} & \text{others.} \end{cases} \quad (24)$$

- (2) Line Path. With line path $Ax + By + C = 0$, the output η_1 is given as follows,

$$\eta_1 = \frac{(Ax + By + C)}{\sqrt{A^2 + B^2}}, \quad (25)$$

where $A^2 + B^2 \neq 0$. In this case $R_t = \omega\eta_1 + R_0$ and R_0 is a constant.

Assumption 1. UUV tracking system satisfies $m_{11} < m_{22}$ and the surge and yaw velocity are nonobservable.

Figure 7 illustrates how the rolling path is generated in three different cases as discussed above. The solid and dashed circles are tangent to the rolling circular path across the vehicle position p_L . According to the consistency between the heading direction of the vehicle and the desired path, the solid arc $\widehat{p_L P}$ will be chosen as the rolling path while the orientation of the vehicle becomes close to the direction of the desired path. Once a rolling path is generated as above, a feedback linearization controller can be designed for automatic arriving.

Theorem 1. Assume (i) the mass, added mass, and damping coefficients are diagonal matrixes; (ii) Assumption 1 is satisfied. For the nonlinear tracking model (20), according to the relative position between the vehicle and the originally desired path, if a circular arc rolling path with radius R_t is chosen in real time with suitable parameters (k_{11}, k_{12}) and (k_{21}, k_{22}) , the controller can be designed.

Proof. The lines and circular arcs can be combined to form any nonlinear path; therefore, the proof is established from the following two aspects.

(1) Desired path is a circle path.

Obtaining a direct relationship between the output η and the input vector,

$$\begin{aligned} \dot{\eta}_1 &= \frac{1}{\sqrt{(x-x_f)^2 + (y-y_f)^2}} \\ &\cdot [(x-x_f)(\dot{x}-\dot{x}_f) + (y-y_f)(\dot{y}-\dot{y}_f)] \\ \dot{\eta}_2 &= \dot{\psi} \\ \ddot{\eta}_1 &= -0.5e_k^3 \\ &\times [(x-x_f)(\dot{x}-\dot{x}_f) + (y-y_f)(\dot{y}-\dot{y}_f)]^2 \\ &+ e_k [(\dot{x}-\dot{x}_f)^2 + (x-x_f)(\ddot{x}-\ddot{x}_f)] \\ &+ e_k [(\dot{y}-\dot{y}_f)^2 + (y-y_f)(\ddot{y}-\ddot{y}_f)] \\ \ddot{\eta}_2 &= \dot{r}, \end{aligned} \quad (26)$$

where

$$e_k = \frac{1}{\sqrt{(x-x_f)^2 + (y-y_f)^2}}. \quad (27)$$

Define state function $h = [h_1 \ h_2]^T$, and

$$u_1 = p_1 vr + p_2 u + p_3 \tau_{du}$$

$$v_1 = p_4 ur + p_5 v + p_6 \tau_{dv}$$

$$\begin{aligned} h_1 &= -0.5e_k^3 \\ &\times [(\dot{x}-\dot{x}_f)(x-x_f) + (\dot{y}-\dot{y}_f)(y-y_f)]^2 \\ &+ (x-x_f)(u_1 \cos \psi - ur \sin \psi \\ &\quad - v_1 \sin \psi - vr \cos \psi - \ddot{x}_f) \\ &+ e_k \cdot \{(\dot{x}-\dot{x}_f)^2 + (\dot{y}-\dot{y}_f)^2 \\ &\quad + (y-y_f)(u_1 \sin \psi + ur \cos \psi \\ &\quad + v_1 \cos \psi - vr \sin \psi - \ddot{y}_f)\} \\ h_2 &= p_7 uv + p_8 r + p_9 \tau_{dr} \\ \ddot{\eta} &= h + \begin{bmatrix} e_k \cdot (p_3 \tau_u ((x-x_f) \cos \psi + (y-y_f) \sin \psi)) \\ p_9 \tau_r \end{bmatrix}. \end{aligned} \quad (28)$$

Choose a new input $\mu = [v_1 \ v_2]^T$ and neglect nonlinear portion; let $e = \sqrt{(x-x_f)^2 + (y-y_f)^2} - R$ be the tracking error, and the input vector can be derived as follows:

$$\begin{aligned} v_1 &= -k_{11}e_1 - k_{12}\dot{e}_1 = -k_{11} \\ &\times \left(\sqrt{(x-x_f)^2 + (y-y_f)^2} - R \right) \\ &- k_{12} \left[\frac{1}{\sqrt{(x-x_f)^2 + (y-y_f)^2}} \right] \\ &\cdot [(x-x_f)(\dot{x}-\dot{x}_f) + (y-y_f)(\dot{y}-\dot{y}_f)] \end{aligned} \quad (29)$$

$$\begin{aligned} v_2 &= \ddot{\eta}_{2d} - k_{21}e_2 - k_{22}\dot{e}_2 \\ &= \ddot{\psi}_d - k_{21}(\psi - \psi_d) - k_{22}(\dot{\psi} - \dot{\psi}_d), \end{aligned}$$

where k_{11} , k_{12} , k_{21} , and k_{22} are constants, $\eta_{2d} = \psi_d$. The input vector can be shown as

$$\begin{aligned} \tau_u &= \frac{(v_1 - h_1)}{p_3 e_k [(x-x_f) \cos \psi + (y-y_f) \sin \psi]} \\ \tau_r &= \frac{(v_2 - h_2)}{p_9}. \end{aligned} \quad (30)$$

(2) Desired path is line path.

The proving process is similar to the above:

$$\begin{aligned} \dot{\eta}_1 &= \frac{(A\dot{x} + B\dot{y} + C)}{\sqrt{A^2 + B^2}} \\ \dot{\eta}_2 &= \dot{\psi} \\ \ddot{\eta}_1 &= \frac{\dot{u}(A \cos \psi + B \sin \psi) + \dot{v}(B \cos \psi - A \sin \psi)}{\sqrt{A^2 + B^2}} \\ &+ \frac{A(-ur \sin \psi - vr \cos \psi)}{\sqrt{A^2 + B^2}} \\ &+ \frac{B(ur \cos \psi - vr \sin \psi) + C}{\sqrt{A^2 + B^2}} \\ \ddot{\eta}_2 &= \dot{r} \\ h_1 &= t_k [A(-ur \sin \psi - vr \cos \psi) \\ &\quad + B(ur \cos \psi - vr \sin \psi) + C \\ &\quad + u_1(A \cos \psi + B \sin \psi) \\ &\quad + v_1(B \cos \psi - A \sin \psi)] \\ h_2 &= p_7 uv + p_8 r + p_9 \tau_{dr} \\ \ddot{\eta} &= h + \begin{bmatrix} t_k p_3 (A \cos \psi + B \sin \psi) \tau_u \\ p_9 \tau_r \end{bmatrix} \end{aligned} \quad (31)$$

with $t_k = 1/\sqrt{A^2 + B^2}$.

Let $e_1 = t_k(Ax + By + C)$ be the tracking error and $e_2 = \eta_{2d} - \eta_2$ the heading error; the new inputs are described as

$$\begin{aligned} \nu_1 &= -k_{11}e_1 - k_{21}\dot{e}_1 \\ &= -k_{11}t_k(Ax + By + C) \\ &\quad - k_{21}t_k(A\dot{x} + B\dot{y}) \\ \nu_2 &= \dot{\eta}_{d2} - k_{12}e_2 - k_{22}\dot{e}_2 \\ &= \ddot{\psi}_d - k_{12}(\psi - \psi_d) - k_{22}(\dot{\psi} - \dot{\psi}_d). \end{aligned} \quad (32)$$

The control output can be shown as

$$\begin{aligned} \tau_u &= \frac{(\nu_1 - h_1)}{[p_3 t_k (A \cos \psi + B \sin \psi)]} \\ \tau_r &= \frac{(\nu_2 - h_2)}{p_9} \end{aligned} \quad (33)$$

□

5.2. Contour Reconstruction Based on Finite Predictive Stable Inversion. For $\forall \varepsilon > 0$, a predictive time instant T_p can be found to obtain stable input $\mu_d(t_c)$ using future output $y_d(t)$, $t \in [t_c, t_c + T_p]$. It satisfies

$$\|\mu_d(t_c) - \mu_p(t_c)\| \leq \varepsilon, \quad (34)$$

where $\mu_p(t_c)$ is the desired input. The inner dynamic condition is given as

$$L \triangleq \begin{bmatrix} L_s(t_c) \\ L_u(t_f) \end{bmatrix} \triangleq \begin{bmatrix} \sigma_s(t_c) \\ \sigma_u(t_f) \end{bmatrix}, \quad t_f = t_c + T_p. \quad (35)$$

A finite predictive path is regarded as known variables in the time window $[t_c, t_c + T_p]$. The desired output in this time window is utilized to describe the stable and unstable part of the inner dynamics. Picard iteration method [10] is introduced to solve the bounded solution of both parts.

Given the nonlinear tracking model (20), the linearity item of inner dynamics $\sigma(t)$ is extracted to obtain a bounded solution.

Let $A_\sigma = \partial s(\sigma, Y_d) / \partial \sigma|_{Y_d=0, \sigma=0}$, and the linearity part $\dot{\sigma}$ can be resolved from $\sigma(t)$ as $\dot{\sigma} = A_\sigma \sigma$. The inner dynamics can be described as

$$\begin{aligned} \dot{\sigma}(t) &= A_\sigma \sigma(t) + [s(\sigma(t), Y_d(t)) - A_\sigma \sigma(t)] \\ &\triangleq A_\sigma \sigma(t) + \Psi(\sigma(t), Y_d(t)). \end{aligned} \quad (36)$$

A_σ can be further separated into a stable part A_s and an unstable part A_u , which, respectively, denote the characteristic values in the left and right planes of the imaginary axis of the complex plane. Thus, the inner dynamics equations can be rewritten as

$$\begin{aligned} \dot{\sigma}_s &= A_s \sigma_s + I_s \Psi_s([\sigma_s \ \sigma_u]^T, Y_d) \\ &\triangleq A_s \sigma_s + \Psi_s(\sigma_s, \sigma_u, Y_d) \\ \dot{\sigma}_u &= A_u \sigma_u + I_u \Psi_u([\sigma_u \ \sigma_s]^T, Y_d) \\ &\triangleq A_u \sigma_u + \Psi_u(\sigma_s, \sigma_u, Y_d). \end{aligned} \quad (37)$$

With Picard iteration method, the bounded solution for both parts of the inner dynamic can be derived as

$$\begin{aligned} \dot{\sigma}_{s,k}(t) &= A_s \sigma_{s,k}(t) \\ &\quad + \Psi_s(\sigma_{s,k-1}(t), \sigma_{u,k-1}(t), Y_d(t)) \\ \dot{\sigma}_{u,k}(t) &= A_u \sigma_{u,k}(t) \\ &\quad + \Psi_u(\sigma_{s,k-1}(t), \sigma_{u,k-1}(t), Y_d(t)). \end{aligned} \quad (38)$$

For any $t \in [t_c, t_f]$, the bounded solution can be described as follows:

(1) Initial solution is shown with $k = 0$,

$$\sigma_0(t) = \begin{bmatrix} e^{A_s(t-t_c)} L_s(t_c) \\ e^{-A_u(t_f-t)} L_u(t_f) \end{bmatrix}. \quad (39)$$

(2) With $k \geq 1$,

$$\begin{aligned} \sigma_k(t) &= \begin{bmatrix} \sigma_{s,k}(t) \\ \sigma_{u,k}(t) \end{bmatrix} \\ &= \begin{bmatrix} e^{A_s(t-t_c)} L_s(t_c) + \int_{t_c}^t e^{A_s(t-\tau)} \Psi_s(\sigma_{s,k-1}(\tau), \sigma_{u,k-1}(\tau), Y_d(\tau)) d\tau \\ e^{-A_u(t_f-t)} L_u(t_f) - \int_t^{t_f} e^{-A_u(\tau-t)} \Psi_u(\sigma_{s,k-1}(\tau), \sigma_{u,k-1}(\tau), Y_d(\tau)) d\tau \end{bmatrix} \\ &\triangleq S_M(\sigma_{k-1}(\cdot), Y_d(\cdot), L)(t). \end{aligned} \quad (40)$$

It is clear that the integration operation for the stable and unstable inner dynamics are from current time instant t_c to t forward and backward, respectively.

Assumption 2. Inner dynamics character $\Psi(\cdot, \cdot)$ is a nonlinear item, and it is satisfied that local Lipschitz condition at origin with any constants (K_1, K_2) and any small positive constant σ_0 hold that, for any $t \in \mathbf{R}$, a bounded function exists

$$\begin{aligned} & \left\| \Psi(\sigma(t), Y(t)) - \Psi(\bar{\sigma}(t), \bar{Y}(t)) \right\|_{\infty} \\ & \leq K_1 \|\sigma(t) - \bar{\sigma}(t)\|_{\infty} + K_2 \|Y(t) - \bar{Y}(t)\|_{\infty}, \end{aligned} \quad (41)$$

where

$$\begin{aligned} & \|Y(\cdot)\|_{\infty} < \sigma_0, \\ & \|\bar{Y}(\cdot)\|_{\infty} < \sigma_0, \\ & \|\bar{\sigma}(\cdot)\|_{\infty} < \sigma_0, \\ & \|\sigma(\cdot)\|_{\infty} < \sigma_0. \end{aligned} \quad (42)$$

Remark. Nonlinearity for inner dynamics will decrease with the decrease of Lipschitz constants K_1 and K_2 . If $\Psi(\cdot, \cdot)$ is continuous and differentiable, the locally Lipschitz condition is satisfied.

Assumption 3. A positive constant K exists to hold that

$$\begin{aligned} & \|\Phi_s(t, t_0)\|_{\infty} \leq K e^{-\alpha(t-t_0)} \quad \forall t_0 \leq t \leq t_f \\ & \|\Phi_u(t_f, t)\|_{\infty} \leq K e^{-\beta(t_f-t)} \quad \forall t_0 \leq t \leq t_f, \end{aligned} \quad (43)$$

where α and β are minimum character values of A_s and A_u , respectively,

$$\begin{aligned} & \alpha < \inf_i |\operatorname{Re}(\lambda_i(A_s))| \\ & \beta < \inf_i |\operatorname{Re}(\lambda_i(A_u))|, \end{aligned} \quad (44)$$

where “inf” denotes the lower bound, “Re” means the real part, and $\lambda_i(A_s)$ and $\lambda_i(A_u)$ are characteristic values of A_s and A_u , respectively.

Remark. The large α and β are, the further is the distance from dynamics poles to the imaginary axis implying stronger hyperbolic properties.

Theorem 2. *If Assumptions 1 and 2 are satisfied and there exist positive constants K_y and K_{σ} , it holds that*

$$\begin{aligned} & \left\| \mu_d(\sigma(t), Y(t)) - \mu_d(\bar{\sigma}(t), \bar{Y}(t)) \right\|_{\infty} \\ & \leq K_y \|Y(t) - \bar{Y}(t)\|_{\infty} \\ & \quad + K_{\sigma} \|\sigma(t) - \bar{\sigma}(t)\|_{\infty}. \end{aligned} \quad (45)$$

If Lipschitz constants K_1 and K_2 in (41) satisfy $(K_1 + K_2)/2 < 1$, there is only one fixed point $\sigma^(\cdot)$ to hold that*

$\sigma^*(t) = s[\sigma^*(\cdot), Y_d(\cdot)](t)$ for any $\forall t \in [t_c, t_c + T_p]$. The error between the desired input $\mu_{e,d}(t_c)$ and the input $\mu_{d,m}(t_c)$ derived from finite predictive inversion can be quantitated as

$$\begin{aligned} e_{d,m}(t_c) & \triangleq \|\mu_{e,d}(t_c) - \mu_{d,m}(t_c)\|_{\infty} \\ & \leq K_{\sigma} K K_{\alpha\beta} \left[\frac{2KK_{\alpha\beta}}{K_2} + \frac{e^{-\hat{\beta}T_p}}{1 - \delta_{\beta}} \right] \|Y_d(\cdot)\|_{\infty}, \end{aligned} \quad (46)$$

where

$$K_{\alpha\beta} = \frac{K_{\alpha\beta,2}}{(1 - K_{\alpha\beta,1})} \quad (47)$$

$$K_{\alpha\beta,1} = KK_1 \max\{(1/\alpha), (1/\beta)\}$$

$$K_{\alpha\beta,2} = KK_2 \max\{(1/\alpha), (1/\beta)\}.$$

The proof is not included here due to space limit. The reader is referred to the original paper for details [21]. According to Theorem 2, the parameter T_p can be optimized and thus the time window.

6. Evaluations and Results

6.1. Evaluation Criteria. Evaluation criteria are set out for assessing the contour accuracy and predictive control performance towards detection mission using the UUV under weak observable conditions, respectively.

6.1.1. Evaluation Criterion for Contour Accuracy. Errors between the reconstructed contour and the environment model are computed in order to evaluate the accuracy of the contour reconstruction (only the contours estimated using the SVC inertia algorithm are evaluated here). Considering the characteristics of the environment, “accumulate error” and “overall error” are proposed. Figure 8 illustrates how these errors are computed where the solid lines express the actual contours of the environment model while the dashed lines indicate the contour reconstructed by using the SVC inertia algorithm.

(1) *Accumulative Error Δd_a .* The error between the contour of the environment model and the reconstructed one can accumulate over time, and the accumulative error is used to evaluate this trend.

As shown in Figure 8, d_1, d_3 , and d_5 are the deflections between the environment contours and the reconstructed smooth path at the starting point of each segment. d_2, d_4 , and d_6 are the deflections at the end point of each segment. Assume that d_s is the safe distance, and the errors of each segment $\Delta d_{a1}, \Delta d_{a2}$, and Δd_{a3} are described as follows:

$$\begin{aligned} \Delta d_{a1} & = (d_2 - d_s) - (d_1 - d_s) = d_2 - d_1 \\ \Delta d_{a2} & = (d_4 - d_s) - (d_3 - d_s) = d_4 - d_3 \\ \Delta d_{a3} & = (d_6 - d_s) - (d_5 - d_s) = d_6 - d_5. \end{aligned} \quad (48)$$

The accumulative error can be defined as follows:

$$\Delta d_a = \sum_{i=1}^N \Delta d_{ai} / N \quad i = 1, 2, 3. \quad (49)$$

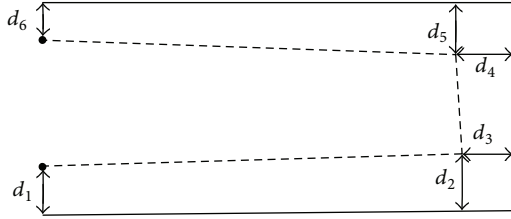


FIGURE 8: Illustration of the accumulate error and overall error.

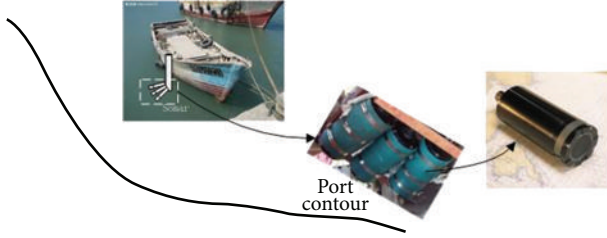


FIGURE 9: Deployment of the sonars.

(2) *Overall Error* Δd_w . The entire environment contour is reconstructed and the overall error Δd_w at the end point is defined as

$$\Delta d_w = d_6 - d_5. \quad (50)$$

Remark. Both Δd_a and Δd_w have signed values. A positive value means the deflection is larger than the safe distance while a negative value means the deflection is smaller than the safe distance implying a threat to the vehicle navigation.

6.1.2. Evaluation Criteria for Autonomous Tracking Control.

To estimate the performance of autonomous tracking control, the error is defined as $e_k = y(k) - \hat{y}(k)$ where $y(k)$ is the navigating path of the vehicle and $\hat{y}(k)$ the smoothed contour from reconstruction. More specifically, the error at the k instant is the minimum distance from the vehicle position to the contour, that is, the tangent point on the contour. Three measures are proposed to analyze the errors.

(1) *Zero Mean Value.* Define $\bar{e} = (1/n) \sum_{k=1}^n e_k$ as the estimated value of error serial $\{e_k\}$. To guarantee the zero mean value, for any $n \in \mathbf{Z}^+$, it is satisfied that $\bar{e} \rightarrow 0$. From the definition of error, it will be affected by the estimated mean values of the output \hat{y} ; therefore, reestimation of the mean value $\bar{e} = \bar{e}/\hat{y} < e_T$ is needed. e_T is preset on the basis of accuracy requirement.

(2) *Validity.* It is defined as the degree of deflection between the error serial $\{e_k\}$ and the zero mean value. Standard deviation $\sigma = \sqrt{(1/(n-1)) \sum_{k=1}^n (e_k - \bar{e})^2}$ is introduced to estimate the data validity.

(3) *Independence.* The error serial describes the error between the real navigating path and the smoothed contour after subtracting the safe distance. Error variables are regarded as random variables and thus the independence describes

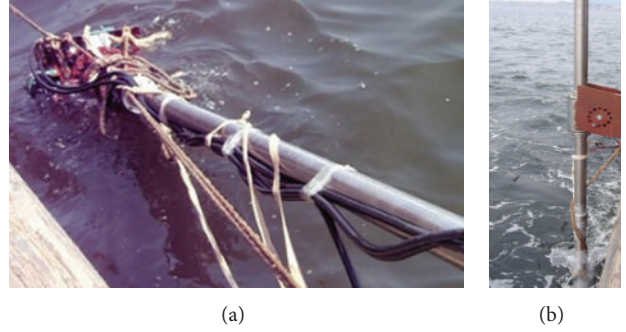


FIGURE 10: Sonar data collection in sea trial.



FIGURE 11: Sea trial in islands environment.

the randomness. Autocorrelation coefficient r_p is used to describe the independence as follows:

$$r_p = \frac{1}{(n-p-1)} \sum_{k=1}^{n-p} (\Delta e_k \Delta e_{p+k}), \quad (51)$$

where $\Delta e_k = e_k - \bar{e}_k$. In this paper the autocorrelation coefficient is rewritten as $\rho_p = r_p/r_0$, and the relation between $\rho_p - k$ will be plotted for demonstration. If autocorrelation coefficients are in the Independence Confidence Limit Interval $(-\sigma, \sigma)$, the error serial is not correlated and vice versa.

6.2. *Validation Using Sea Trial Data.* The performance of the proposed model is verified with the data collected from a sea trial operated at Xiaoping Island in Dalian, China, in August 2009. We choose sonar data collected from two different environments, respectively, a port and two islands. The single-beam sonars were assembled on one side of a fishing boat to simulate UUV sonars. To guarantee the number of sonar data satisfying the clustering requirement, 3 single beam sonars were fixed in the boat side; see Figure 9. The sonars used in the sea trial are manufactured by Kongsberg, Norway. The boat was steered along the contour with acoustic beam vertically acting on environment. The sonars were deployed below the bottom of the boat to avoid acoustic beam to project onto the boat. The boat sails at a velocity of 2 m/s.

Sonar data are collected during the boat navigating along islands horizontally in real time. To verify the effectiveness of the proposed control algorithm in disturbance situation, synthetic disturbance noise is added: $t_{du} = 40 \times (1 + \text{rand}(\cdot))$,

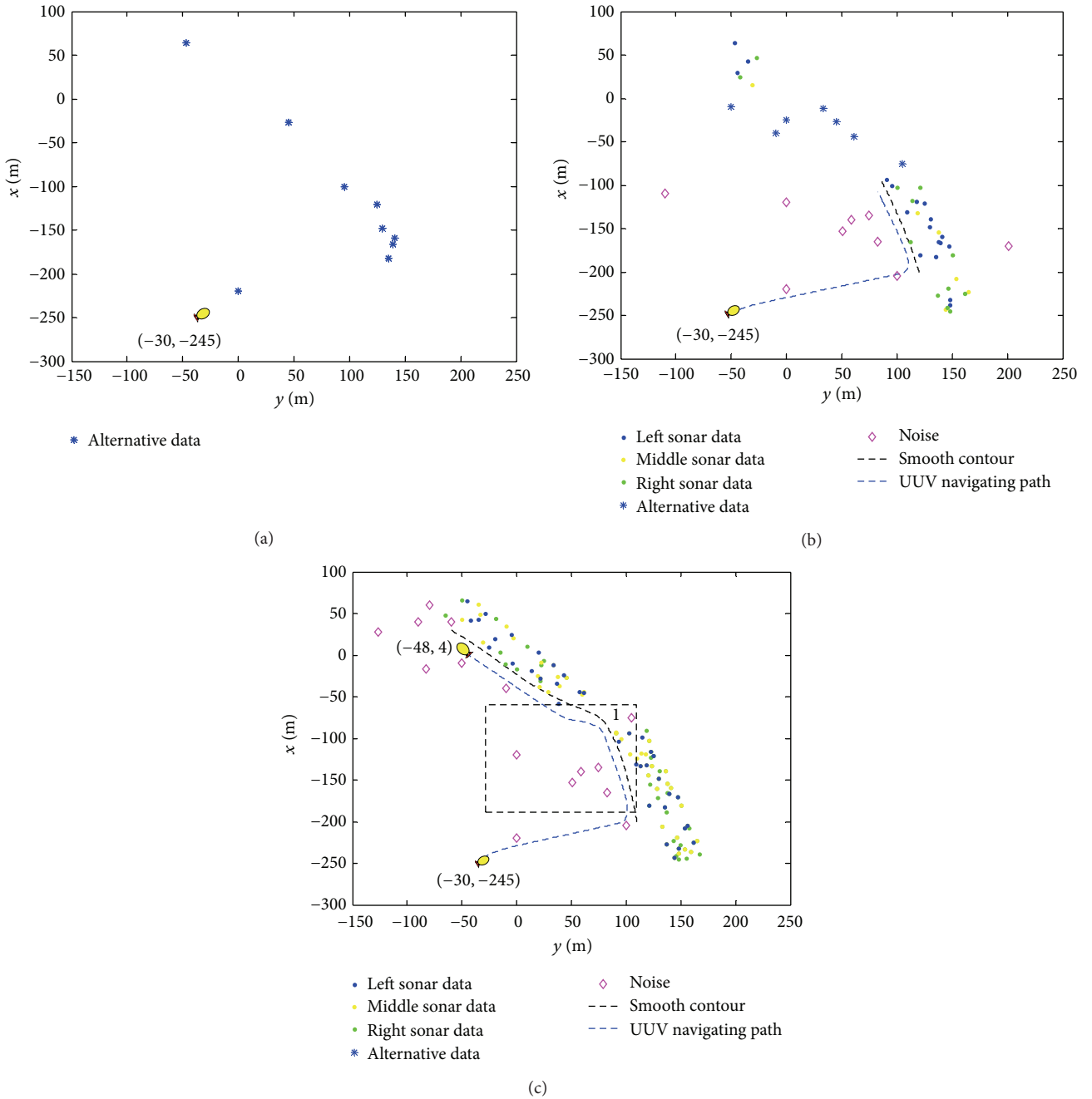


FIGURE 12: Islands contour construction and patrolling.

$\tau_{dv} = 35 \times (1 + \text{rand}(\cdot))$, $\tau_{dr} = 40 \times (1 + \text{rand}(\cdot))$, with $\text{rand}(\cdot)$ denoting zero-mean Gaussian random noise. From Figure 10 the sonar assembly in the sea can be seen, with diving in the water as Figure 10(a) and navigating as Figure 10(b).

6.2.1. Path Following Validation Using Data from Islands. In the first experiment, the boat navigates between two small islands in order to move close to them (see Figure 11). We record the data from initial point A (-30, -245) with heading of 60 degree. The effective distance of sonars is 200 m.

Figure 12 shows the process of data clustering. At the beginning, the number of data is too small to construct accurate data class; therefore, data are placed into alternative set; see Figure 12(a). With the number of sonar data increasing, data with the same property, including data in alternative set, are clustered in corresponding class contours with support vectors distributed on. Sonar data determined by the rules in Section 3 are eliminated as outliers (see Figures 12(b) and 12(c)). The contours close to UUV, determined by normal line and distance, connected with lines successively, are regarded as preliminary contour, waiting for postprocessing.

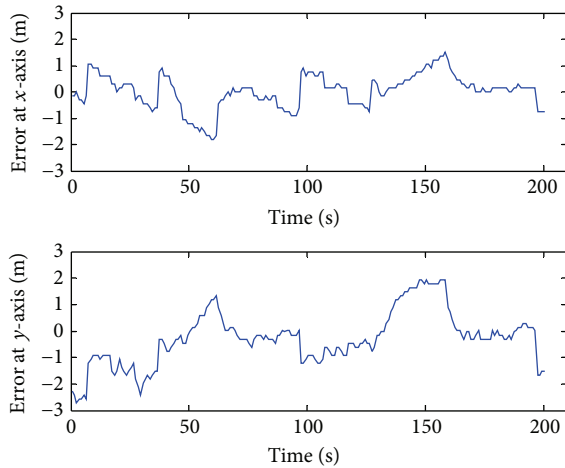


FIGURE 13: Patrolling errors at x - and y -axis in islands contour.

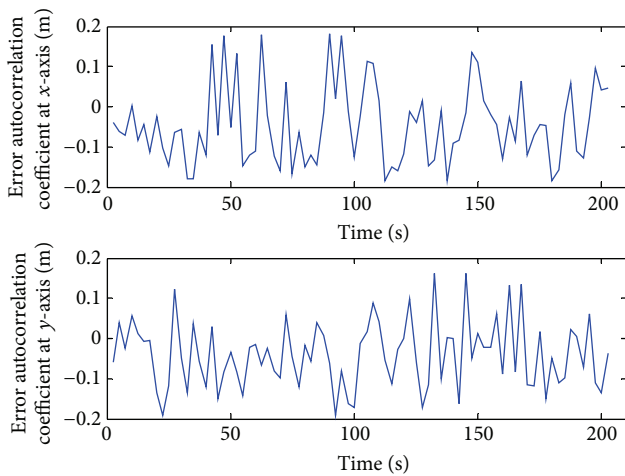


FIGURE 14: Error autocorrelation coefficients of error serials.

On the basis of initial contour (see solid green line in Figures 12(b) and 12(c)), the smoothing method described in Section 4.2 is applied and a reasonable desired reference path (dash black curves) for vehicle navigation can be achieved. When the vehicle is close to the island on the right less than 200 m, sonar data returned is collected to construct the contour in real time which is regarded as the guidance for the vehicle to navigate, keeping a fixed distance (set to 10 m in this paper) as the safe distance. The optimal time window parameter is $T_p \geq 3$ s. Considering the control period 0.5 s, $T_p = 3$ s satisfies the requirement (5 original sonar data and 6 wavelet coefficients) to estimate wavelet coefficients at any time. According to Section 6.1, the error series denoting the deflection between the actual navigation path and the reconstructed contour is computed; see Figure 13.

The mean (standard deviation) value along x - and y -axis is 0.15 (0.37) m and 0.98 (0.30) m, respectively. According to evaluation criterion for contour accuracy in Section 6.1.1, the accumulated error for island contour is 1.45 m, and overall

error is 1.74 m. The negative deflection values are much smaller than the safe distance; this means that the UUV can safely complete the detection mission as designed. The maximum absolute value of the autocorrelation coefficients at x -axis and y -axis is 0.14 and 0.16, respectively, while the respective standard deviations are 0.37 and 0.30 (see Figure 14). This confirms that the error series is independent without influence from the designed controller. This means that the designed controller using finite predictive stable inversion algorithm can control the vehicle to follow a path predicted in real time.

6.2.2. Path Tracking Validation Using Data from Port. With the same deployment and trial methods as above, sonar data was collected in a port as Figure 15 with its environment pictures at the right. The initial position of the boat was $(-1605, -688)$. Owing to the port area is wide and safe distance is set to 30 m. Figure 16 demonstrates the autonomous tracking of the boat in the port. From Figure 16, it can be seen that the smoothed path illustrated by black solid is reconstructed using data from 3 sonars and guides the vehicle to approach the target. The noise in the data is detected and eliminated to guarantee the accuracy of the contour.

Figure 17 illustrates the error of the smoothed path reconstructed and the actual path without consideration of the safe distance. Figure 18 shows the autocorrelation coefficients of the tracking error series to assess whether the errors are from the control method or stochastic disturbance. The maximum absolute value of autocorrelation coefficients at x -axis and y -axis are 0.20 and 0.20, within the range of standard deviations, respectively, $(-0.26, 0.26)$ and $(-0.38, 0.38)$ (see Figure 17). Therefore, it is true that error series are caused stochastically. Moreover, the accumulated error for island contour tracking is 1.50 m, and overall error is 1.20 m, respectively.

7. Conclusions

A new autopilot system, known as SDAP, is proposed for the exploration of underwater environments using UUV equipped with multiple single-beam sonars. The main issue studied is the control problem in detection process which is separated into two stages according to different requirements of the mission, accurate tracking and autonomous tracking. A rolling path generation method is present to guide the vehicle to follow a preset path accurately. For autonomous tracking stage, wavelet transform is introduced to preprocess weak observable data and the wavelet coefficients obtained are used to reconstruct the contour of the environment using the SVC inertia algorithm. To satisfy the inertia property of the UUV which requires a smooth reference path, different order Bézier curves are included to fit the initial contours for the desired reference path by considering a fixed safety distance. Finite predictive stable inversion method is applied to control the vehicle in order to follow the predictive path in real time. Data collected from a sea trial is used to validate the proposed technique, and the results have demonstrated that the algorithms are able to control vehicle navigating along the desired paths that are either preset or predicted automatically.



FIGURE 15: Sea trial in port environment.

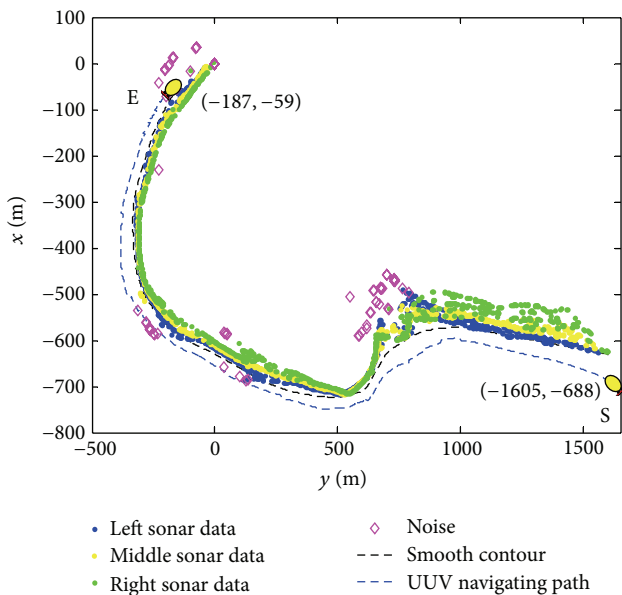


FIGURE 16: Port contour construction and tracking.

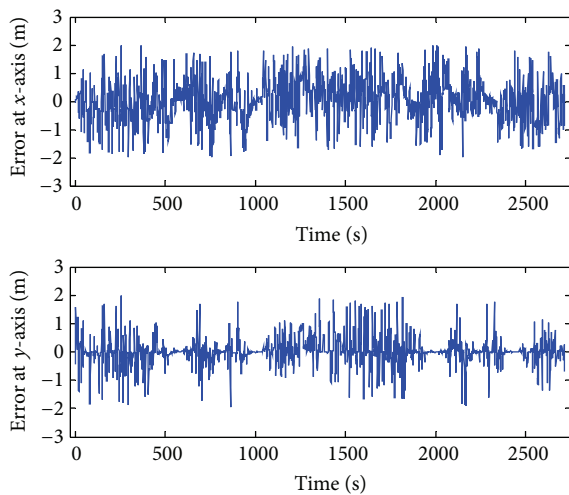


FIGURE 17: Path error serials at x - and y -axis for patrolling port contour.

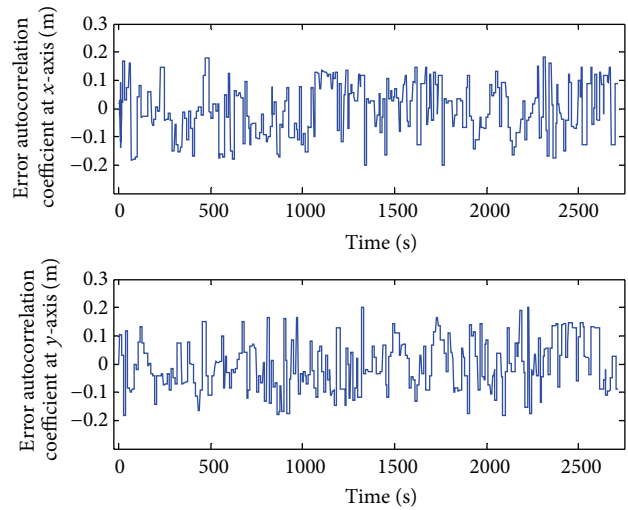


FIGURE 18: Error autocorrelation coefficients of error serials.

It has laid a solid foundation for using UUV to perform SDAP mission.

During environment detection, the accuracy of environment information obtained is vital to guide UUV steering safely. With the insight, the navigation error will affect the environment outline constructed. In this paper, it is assumed that the navigation error is not considered in the UUV steering. In the future study, it is necessary to include navigation error into the SDAP issue for completeness. Otherwise, the further verification should be implemented through inserting the algorithms into UUV and executing mission in the real environment underwater.

Acknowledgments

This work is partially supported by the Natural Science Foundation of China (51179038), the Program of New Century Excellent Talents in University (NCET-10-0053), and Fundamental Research Funds for the Central Universities (HEUCF041323).

References

- [1] A. Chatterjee and F. Matsuno, "A Geese PSO tuned fuzzy supervisor for EKF based solutions of simultaneous localization and mapping (SLAM) problems in mobile robots," *Expert Systems with Applications*, vol. 37, no. 8, pp. 5542–5548, 2010.
- [2] O. Calvo, A. Rozenfeld, A. Souza, F. Valenciaga, P. F. Puleston, and G. Acosta, "Experimental results on smooth path tracking with application to pipe surveying on inexpensive AUV," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '09)*, pp. 3647–3653, September 2008.
- [3] Y. Gan, L. Wan, W. Li et al., "Research on path following of underwater vehicle without rudder and fin," *The Ocean Engineering*, vol. 25, no. 1, pp. 70–75, 2007.
- [4] P. Encarnação and A. Pascoal, "3D path following for autonomous underwater vehicle," in *Proceedings of the 39th IEEE Conference on Decision and Control*, pp. 2977–2982, December 2000.
- [5] L. Lapierre and D. Soetanto, "Nonlinear path-following control of an AUV," *The Ocean Engineering*, vol. 34, no. 11-12, pp. 1734–1744, 2007.
- [6] L. Lapierre, D. Soetanto, and A. Pascoal, "Nonlinear path following with applications to the control of autonomous underwater vehicles," in *Proceedings of the 42nd IEEE Conference on Decision and Control*, pp. 1256–1261, December 2003.
- [7] X. Xiang, *Research on Path following and coordinated control for second-order nonholonomic AUVs [doctor dissertation]*, Huazhong University of Science and Technology, Wuhan, China, 2010.
- [8] H. Ren, L. Li, and X. Bian, "Research on fuzzy heading guidance based track keeping method for AUV," *Applied Science and Technology*, vol. 31, no. 9, pp. 43–45, 2004.
- [9] J. Zhao, *AUV Fuzzy Neural Network Hybrid Learning Algorithm Control*, Harbin Engineering University, Harbin, China, 2007.
- [10] L. Ćirić, A. Rafiq, and N. Cakić, "On Picard iterations for strongly accretive and strongly pseudo-contractive Lipschitz mappings," *Nonlinear Analysis: Theory, Methods and Applications*, vol. 70, no. 12, pp. 4332–4337, 2009.
- [11] K. D. Do, "Global robust and adaptive output feedback dynamic positioning of surface ships," *Journal of Marine Science and Application*, vol. 10, no. 3, pp. 325–332, 2011.
- [12] R. Amin, A. A. Khayyat, and K. G. Osgouie, "Neural networks modeling of autonomous underwater vehicle," in *Proceedings of the IEEE/ASME International Conference on Mechatronic and Embedded Systems and Applications (MESA '10)*, pp. 14–19, July 2010.
- [13] I. Astrov and A. Pedai, "Multirate depth control of an AUV by neural network predictive controller for enhanced situational awareness," in *Proceedings of the 5th International Symposium on Computational Intelligence and Intelligent Informatics (ISCIII '11)*, pp. 47–52, Floriana, Malta, September 2011.
- [14] K. Maček, R. Philippsen, and R. Siegwart, "Path following for autonomous vehicle navigation with inherent safety and dynamics margin," in *Proceedings of the IEEE Intelligent Vehicles Symposium*, vol. 6, pp. 108–113, June 2008.
- [15] Z. Wang, X. Peng, and D. Wang, "Path following robust output control of underactuated ship under Serret-Frenet frame," *Ship Engineering*, vol. 32, no. 3, pp. 24–32, 2010.
- [16] M.-J. Zhang, P. Gao, and J.-A. Xu, "Neural-network-based generalized predictive control for autonomous underwater vehicles," *Robot*, vol. 30, no. 1, pp. 91–96, 2008.
- [17] S.-R. Oh and J. Sun, "Path following of underactuated marine surface vessels using line-of-sight based model predictive control," *The Ocean Engineering*, vol. 37, no. 2-3, pp. 289–295, 2010.
- [18] Y. Jiang, *Neural Network Based Feedback Linearization Control of an Unmanned Aerial Vehicle*, Ottawa-Carleton Institute for Mechanical and Aerospace Engineering University, Ottawa, Canada, 2005.
- [19] Y. Han, Y. Sun, and H. Mo, "Design of nonlinear robust controller for underwater high-speed vehicle," *Journal of Wuhan University of Technology*, vol. 34, no. 2, pp. 221–232, 2010.
- [20] J. SONG, *Research on control of supercavitating vehicle based on feedback linearization [master dissertation]*, Harbin Harbin Institute of Technology, 2011.
- [21] Q. Zou, *Preview-based systems-inversion for output-tracking: theory and application [doctor dissertation]*, University of Washington, Washington, DC, USA, 2003.
- [22] Q. Zou and S. Devasia, "Precision preview-based stable-inversion for nonlinear nonminimum-phase systems: the VTOL example," *Automatica*, vol. 43, no. 1, pp. 117–127, 2007.
- [23] Z. Jianhong and Z. Ping, "Precise decoupling tracking of airspeed and altitude for UAV based on causal solution of stable inversion," *Chinese Journal of Aeronautics*, vol. 22, no. 3, pp. 307–315, 2009.
- [24] F. Liu and Z.-Z. Mao, "Outlier detection for control process data based on wavelet-HMM methods," *Control and Decision*, vol. 26, no. 8, pp. 1187–1196, 2011.
- [25] Z. Yan, D. Chi, Z. Zhao et al., "Outlier inertia detection of UUV obstacles based on support vector clustering," *Journal of Harbin Engineering University*, vol. 33, no. 11, pp. 1377–1383, 2012.
- [26] J.-W. Choi, R. E. Curry, and G. H. Elkaim, "Curvature-continuous trajectory generation with corridor constraint for autonomous ground vehicles," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 7166–7171, Hilton Atlanta Hotel, Atlanta, Ga, USA, December 2010.
- [27] J. A. Reeds and L. A. Shepp, "Optimal paths for a car that goes both forwards and backwards," *Pacific Journal of Mathematics*, vol. 145, no. 2, pp. 367–393, 1990.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

