Journal of Intelligent & Robotic Systems (2020) 99:891–908
https://doi.org/10.1007/s10846-019-01146-3

Path Planning under Constraints and Path Following Control of Autonomous Underwater Vehicle with Dynamical Uncertainties and Wave Disturbances

Xiaowei Wang [1] · Xuliang Yao [2] · Le Zhang [3]

Received: 17 April 2019 / Accepted: 30 December 2019 / Published online: 4 February 2020
© Springer Nature B.V. 2020

Abstract
The path planning and following control problems of autonomous underwater vehicle (AUV) in three-dimension (3D) are studied in this paper. In order to realize obstacle avoidance and path optimization, a path planning method based on particle swarm optimization (PSO) and cubic spline interpolation is developed. The curvature of the path obtained by this method is continuous, which can not only avoid obstacles but also meet the constraint of AUV's minimum radius of rotation. In the design of kinematics controller, an optimal guidance scheme based on model predictive control (MPC) is proposed, which takes into account the wave disturbances. Adaptive dynamical sliding mode control (ADSMC) technology is used to design dynamic controller, which can effectively overcome the influence of model uncertainties. In order to ensure the stability of the system, the stability condition of MPC is designed, and the stability of the closed-loop system is analyzed by applying cascade system theory. The control strategy proposed in this paper is compared with the line-of-sight (LOS) guidance through simulation experiment. The simulation results demonstrate that the proposed control strategy can not only improve the quality of path following, but also reduce the disturbance of waves, and thus is more conducive to energy saving.

Keywords AUV · Path planning · Path following · MPC · SMC · PSO

1 Introduction

The autonomous underwater vehicle (AUV) is widely used in military, ocean exploitation and scientific research. Path planning and path following are two key techniques for AUV to perform underwater reconnaissance missions. Obstacles often exist in the underwater environment where AUV works. In order to ensure the safety of AUV, it is necessary to plan a reference path to avoid obstacles. Path planning is mainly divided into static and dynamic planning. The static refers to a planning that the optimal global path can be obtained in advance through offline planning when the information of obstacles is known. On the contrary, if online planning is required to obtain the local optimal path in real time, that is dynamic planning. When operated in the open sea, the path of an AUV is usually described in terms of waypoints, which are connected by a series of lines [1, 2]. Its advantages are that the path planning is simple and the computation burden is small. However, the path made up of straight lines is not smooth because the derivative at the waypoint is discontinuous. This problem can be solved by adding an arc between every two adjacent straight paths [3]. But the curvature of the path is still discontinuous, which causes the path following control to be not smooth when the path is switched. The curvature continuous path can be obtained by replacing arc with Bessel curve [4]. However, the path obtained by this method does not pass through the waypoints. To obtain the path with continuous curvature and can pass through all waypoints, an

This work has been supported by the National Natural Science Foundation of China (51279039).

✉ Xiaowei Wang
wangxiaowei@hrbeu.edu.cn

Xuliang Yao
yaoxuliang@hrbeu.edu.cn

Le Zhang
zhangle98538@163.com

[1] College of Mechanical engineering, Jiujiang Vocational and Technical College, Lianxi District, Jiujiang 332007, China

[2] College of Automation, Harbin Engineering University, Nangang District, Harbin 150001, China

[3] The Center for International Exchange and Cooperation, Jiujiang Vocational and Technical College, Lianxi District, Jiujiang 332007, China

effective method is to use spline interpolation for path planning. In [5], when the underwater environment information is not obtained in advance, according to the real-time obstacle information detected by the onboard sensors, real-time obstacle avoidance of AUV is achieved by adding waypoint near the obstacle and using spline interpolation technique to partially modify the original path. In [6], based on the obstacle information detected by the onboard sensors in real time, the feasible underwater area is approximately defined by two polygonal chains. Then, the spline interpolation technique is applied for local path planning, and the parameters of the local path are optimized online. Therefore, the smooth path with the minimum curvature in real time can be obtained. Since the above two methods are dynamic planning, the online computing burden will be increased, and the configuration requirements of AUV onboard sensors are also relatively high. In order to reduce the online computation burden, static planning can be used to obtain the global optimal path when the environment information is known. In [7, 8], spline interpolation is adopted to plan smooth path with continuous curvature. However, both methods assume that the waypoints are known and do not consider the optimization of the path. In order to find the best path, path optimization is also a hot topic. At present, common path optimization methods are A* algorithm [9], potential fields [10] algorithm, and RRT algorithm [4], etc. In addition, intelligent optimization algorithms are also applied to path planning of robots, such as ant colony algorithm [11], genetic algorithm [12] and PSO algorithm [13]. However, each optimization method also has some shortcomings, so the improvement measures for them have been the research focus. In order to reduce the cost, AUV is often designed in underactuated. They often lack actuation in certain degrees of freedom which imposes nonintegrable acceleration constraints. Therefore, they have the constraint of minimum radius of rotation. If the minimum radius of curvature of the path is less than the minimum radius of rotation of AUV, the path cannot be effectively tracked. This problem is not considered in the above literature.

Path following control for ship and AUV is also an active field of research. However, the controller design is not an easy task because AUV's motions and model are coupled, nonlinear, and uncertain. In addition, the environment in which AUV works often has various disturbances, such as waves. A complete path following control system is generally composed of guidance and attitude control. The guidance system belongs to the kinematics controller, which is used to generate the desired attitude signal. In [14, 15], the line-of-sight (LOS) guidance law is applied for path following of ships. In order to improve the performance of traditional LOS guidance law, LOS guidance law with time-varying lookahead distance is presented in [16, 17]. Alternatively, the vector field (VF) guidance law is also a popular guidance method [18, 19]. The dynamic controller has also been extensively studied. The prevailing control methods for marine surface vessels and AUV in path following include proportional-integral-derivative (PID) control [20, 21], feedback linearizing control techniques [22–24], backstepping method [25, 26], Lyapunov direct method [27], gain scheduling control theory [28], adaptive control [29–32], robust control [33–35], sliding mode control (SMC) [36–40], fuzzy logic control [41], and neural network control [42], etc. However, AUV is very susceptible to waves when working near the surface. If the wave interference cannot be dealt with properly, the high frequency component will be added to the control signal and the forward resistance of the AUV will be increased. If the tracking accuracy of the path is blindly pursued, the energy consumption will be greatly increased. Due to the limited energy carried by the AUV, energy saving must be considered in order to increase the endurance. This problem is rarely considered in the above literature. In [43], MPC is proposed to solve the depth control problem of AUV when near the surface with wave disturbances. However, this method only solves the two-dimensional tracking control in the vertical plane. In practical work, for many underwater tracking tasks, the location of AUV only needs to be kept within a certain target region to meet the requirements. In [44], an adaptive region tracking control method is proposed to make the AUV converge to the target region.

Motivated by the above literatures, path planning and path following control for underactuated AUV are studied simultaneously in this paper. Assuming that the underwater environment model is known, cubic spline interpolation technique and an improved PSO algorithm are used for global path planning. The smooth path with continuous curvature can be obtained. The path can avoid the obstacles and meets the constraint of minimum radius of rotation. In order to improve the path following performance in the presence of wave disturbances and model uncertainties, the kinematics and dynamics controllers are designed based on the cascade control strategy. In kinematic controller, an optimal guidance scheme is developed based on MPC, which can effectively improve the quality of path following and reduce the wave disturbances. Dynamic controller, which is based on adaptive dynamical sliding mode control (ADSMC) theory, is developed to overcome the model uncertainties.

The remainder of this article is organized as follows: The path planning problem is studied in section 2. The problem of 3D path following control of underactuated AUV is formulated in section 3. The kinematics and dynamics controller are derived in section 4. Simulation experiments are presented in section 5. Conclusions are presented in section 6.

2 Path Planning

2.1 Cubic Spline Interpolation

Given a finite number of waypoints, a path with continuous curvature can be obtained by cubic spline interpolation. If given $N_s + 1$ waypoints(ξ_k, η_k, ζ_k), $(k = 0, 1, \cdots N_s)$ in the fixed coordinate system, then N_s-segment splines can be generated which can be expressed as the following cubic polynomial by the parameter $\omega \in [0, 1]$,

$$\begin{bmatrix} \overline{\xi}_k(\omega) \\ \overline{\eta}_k(\omega) \\ \overline{\zeta}_k(\omega) \end{bmatrix} = \begin{bmatrix} a_k^\xi & b_k^\xi & c_k^\xi & d_k^\xi \\ a_k^\eta & b_k^\eta & c_k^\eta & d_k^\eta \\ a_k^\zeta & b_k^\zeta & c_k^\zeta & d_k^\zeta \end{bmatrix} \begin{bmatrix} 1 & \omega & \omega^2 & \omega^3 \end{bmatrix}^T, \quad (k = 0, 1, \cdots, N_s - 1). \tag{1}$$

Since the calculation method of the spline in three coordinate directions is the same, taking the ξ-axis direction as an example, the k-segment spline curve can be expressed as $\overline{\xi}_k(\omega) = a_k^\xi + b_k^\xi \omega + c_k^\xi \omega^2 + d_k^\xi \omega^3$. The remaining task is to compute the parameters $a_k^\xi, b_k^\xi, c_k^\xi, d_k^\xi, (k = 0, 1, \cdots, N_s - 1)$. According to the boundary conditions of each segment spline, the following equation can be obtained

$$\begin{cases} \overline{\xi}_k(0) = a_k^\xi, \\ \overline{\xi}_k(1) = a_k^\xi + b_k^\xi + c_k^\xi + d_k^\xi, \\ \overline{\xi}'_k(0) = D_k = b_k^\xi, \\ \overline{\xi}'_k(1) = D_{k+1} = b_k^\xi + 2c_k^\xi + 3d_k^\xi, \end{cases} \tag{2}$$

where $\overline{\xi}'_k(\omega) = d\overline{\xi}_k(\omega)/d\omega$.

Since $\overline{\xi}_k(0) = \xi_k, \overline{\xi}_k(1) = \xi_{k+1}$, we can get

$$\begin{cases} \xi_{k+1} = \xi_k + D_k + c_k^\xi + d_k^\xi, \\ D_{k+1} = D_k + 2c_k^\xi + 3d_k^\xi. \end{cases} \tag{3}$$

Then, the eq. (2) can be rewritten as

$$\begin{cases} a_k^\xi = \xi_k, \\ b_k^\xi = D_k, \\ c_k^\xi = 3(\xi_{k+1} - \xi_k) - 2D_k - D_{k+1}, \\ d_k^\xi = -2(\xi_{k+1} - \xi_k) + D_k + D_{k+1}. \end{cases} \tag{4}$$

Next, we need to calculate the parameterD_k, $(k = 0, 1, \cdots N_s)$ in the eq. (4). Since the second derivative of the spline is continuous, we can get that

$$\overline{\xi}_{k-1}'(1) = 2c_{k-1}^\xi + 6d_{k-1}^\xi = \overline{\xi}_k''(0) = 2c_k^\xi, \quad (k = 1, \cdots, N_s - 1). \tag{5}$$

Put the last two equations in (4) into eq. (5), and then we can get

$$D_{k-1} + 4D_k + D_{k+1} = 3(\xi_{k+1} - \xi_{k-1}), \quad (k = 1, \cdots, N_s - 1). \tag{6}$$

In addition, according to the natural boundary conditions $\overline{\xi}_0''(0) = 0, \overline{\xi}_{N_s-1}''(1) = 0$, we can get that

$$\begin{cases} 2D_0 + D_1 = 3(\xi_1 - \xi_0), \\ D_{N_s-1} + 2D_{N_s} = 3(\xi_{N_s} - \xi_{N_s-1}). \end{cases} \tag{7}$$

Finally, eqs. (6) and (7) can be rewritten as matrix form in the following

$$\begin{bmatrix} 2 & 1 & 0 & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & 0 & 0 \\ 0 & 0 & \ddots & \ddots & \ddots & 0 \\ 0 & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & 0 & 0 & 1 & 2 \end{bmatrix} \begin{bmatrix} D_0 \\ D_1 \\ D_2 \\ \vdots \\ D_{N_s-1} \\ D_{N_s} \end{bmatrix} = \begin{bmatrix} 3(\xi_1 - \xi_0) \\ 3(\xi_2 - \xi_0) \\ 3(\xi_3 - \xi_1) \\ \vdots \\ 3(\xi_{N_s} - \xi_{N_s-2}) \\ 3(\xi_{N_s} - \xi_{N_s-1}) \end{bmatrix} \tag{8}$$

2.2 PSO Algorithm

2.2.1 Traditional PSO Algorithm

PSO algorithm is widely used because it has the advantages of fewer parameters, fast convergence, no decoding required, and so on. It is based on group and moves individuals in the group to good areas according to their fitness to the environment. It regards each individual as a particle flying at a certain speed in the D_{PSO}-dimensional search space. Let's define the number of particles as N_{PSO}, the current speed and position of the particles are $v_i = (v_{i,1}, v_{i,2}, \cdots, v_{i,D_{PSO}})^T$ and $x_i = (x_{i,1}, x_{i,2}, \cdots, x_{i,D_{PSO}})^T$, $(i = 1, \cdots, N_{PSO})$, the optimal position obtained by individual is$p_i = (p_{i,1}, p_{i,2}, \cdots, p_{i,D_{PSO}})^T$, the best position obtained by the group is $p_g = \left(p_{g,1}, p_{g,2}, \cdots, p_{g,D_{PSO}}\right)^T$. In every generation of evolution, the speed and position of the particles are updated as following

$$\begin{cases} v_{i,j}(t+1) = w_i v_{i,j}(t) + b_1 r_1 \left[p_{i,j} - x_{i,j}(t)\right] \\ \qquad\qquad + b_2 r_2 \left[p_{g,j} - x_{i,j}(t)\right], \\ x_{i,j}(t+1) = x_{i,j}(t) + v_{i,j}(t+1), \\ (i = 1, \cdots, N_{PSO}), (j = 1, \cdots, D_{PSO}), \end{cases} \tag{9}$$

where $w_i > 0$ is inertia weight, $b_1 > 0$ is the learning parameter based on individual experience, $b_2 > 0$ is the learning parameter based on group experience, $r_1 \in [0, 1]$ and $r_2 \in [0, 1]$are random values.

2.2.2 Improved PSO Algorithm (IPSO)

Although the traditional PSO algorithm converges quickly, it is easy to fall into local optimum, i.e. premature problem. In view of the shortcomings of the algorithm, researchers have made various improvements. Inertia weight w determines how much the particle inherits its current velocity. A larger inertia weight is conducive to improving the global search ability, while a smaller inertia weight is conducive to improving the local search ability. In order to improve the optimization ability of the algorithm, the inertial weight can be adjusted according to the fitness of particles. However, it is often not enough to improve the premature problem only by adaptive inertia weight method. Therefore, some researchers proposed the ideas of introducing particle swarm concentration parameters, and adjusting selection probability or topological structure, while others combine particle swarm optimization with genetic algorithm or other optimization methods. In this article, in order to balance the global search ability and local search ability of the algorithm, the PSO algorithm is improved as follows:

Half of the particles in front are defined as global search particle, and the other half is defined as local search particle. The global search particle uses the larger inertia weight $w_{\max}$, while the local search particle uses the smaller inertia weight $w_{\min}$.

In the evolution of each generation, half of the global search particle with better fitness will be directly selected as the local search particle, while the remaining half of the local search particle will select the global search particle according to the probability. The probability of each global search particle being selected is calculated as follows

$$P_i^s = \frac{1}{\frac{N_{PSO}}{2}-1}\left(1-\frac{fit(x_i)}{\sum_{i=1}^{N_{PSO}/2} fit(x_i)}\right), \left(i=1,\cdots,\frac{N_{PSO}}{2}\right). \quad (10)$$

In order to increase the diversity of the group and improve the local search ability, in the evolution of each generation, the crossover operation is carried out for every two adjacent local search particles as following

$$\begin{cases} x_i = r_3 x_i + (1-r_3)x_{i+1}, \\ x_{i+1} = r_3 x_{i+1} + (1-r_3)x_i, \end{cases} \left(i=\frac{N_{PSO}}{2}+1, \frac{N_{PSO}}{2}+3, \cdots, N_{PSO}-1\right), \quad (11)$$

where $r_3 \in [0, 1]$ are random values. Because local search particles are all superior individuals in the current population, better individuals may be produced after crossover operation.

In the process of evolution, if the individual optimal value of a global search particle is not improved for successive $\overline{G}_L$ generations, the particle will be reinitialized.

2.3 Path Optimization Based on IPSO Algorithm and Cubic Spline Interpolation (IPSO-SP)

2.3.1 The Encoding of Particle Position and Speed

Particle positions are encoded using the coordinates of the waypoints. Since the two endpoints of the path (ξ_k, η_k, ζ_k), (k = 0, N_s)are known, we only need to determine the remaining $N_s - 1$ waypoints (ξ_k, η_k, ζ_k),(k = 1, $\cdots N_s - 1$). Firstly, the algorithm needs to initialize the waypoints. The initial coordinates of the waypoints are randomly obtained around the $N_s - 1$ average points between the two endpoints as following

$$\begin{cases} \xi_{i,k} = \xi_0 + \frac{\xi_{N_s}-\xi_0}{N_s}(k+r_4), \\ \eta_{i,k} = \eta_0 + \frac{\eta_{N_s}-\eta_0}{N_s}(k+r_4), \\ \zeta_{i,k} = \zeta_0 + \frac{\zeta_{N_s}-\zeta_0}{N_s}(k+r_4), \\ (i=1,\cdots,N_{PSO}), (k=1,\cdots N_s-1), \end{cases} \quad (12)$$

where r_4 is random values varying in the range of [-0.5, 0.5]. The position information of particles can be represented by the following matrix

$$X_{PSO} = [x_1 \quad x_2 \quad \cdots \quad x_{N_{PSO}}]^T = \begin{bmatrix} \xi_{1,1} & \cdots & \xi_{1,N_s-1} & \eta_{1,1} & \cdots & \eta_{1,N_s-1} & \zeta_{1,1} & \cdots & \zeta_{1,N_s-1} \\ \xi_{2,1} & \cdots & \xi_{2,N_s-1} & \eta_{2,1} & \cdots & \eta_{2,N_s-1} & \zeta_{2,1} & \cdots & \zeta_{2,N_s-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \xi_{N_{PSO},1} & \cdots & \xi_{N_{PSO},N_s-1} & \eta_{N_{PSO},1} & \cdots & \eta_{N_{PSO},N_s-1} & \zeta_{N_{PSO},1} & \cdots & \zeta_{N_{PSO},N_s-1} \end{bmatrix}. \quad (13)$$

The number of rows in the matrix (13) is N_{PSO}. The number of columns in the matrix (13) is $D_{PSO} = 3 \times (N_s - 1)$, which represents the dimension of the particle. If the ζ-coordinate is ignored in the matrix (13), then the three-dimensional path planning is simplified to the two-dimensional path planning in horizontal plane. If the η-coordinate is ignored in the matrix (13), then the three-dimensional path planning is simplified to the two-dimensional path planning in vertical plane. The

encoding of particle speed has the same form as that of position. The initial speed of particles is generated randomly within [$-V_{\max}$, $V_{\max}$], where $V_{\max}$ is the maximum speed of particle.

2.3.2 The Selection of Fitness Functions

In order to improve the path optimization ability when there are obstacle avoidance requirements and curvature constraints, the fitness function is selected as following

$$F_{PSO} = L(1 + N_1 + N_2) \tag{14}$$

where

$$L = \sum_{i=0}^{N_s/\Delta\omega-1} \sqrt{\left(\xi_{i+1}^s - \xi_i^s\right)^2 + \left(\eta_{i+1}^s - \eta_i^s\right)^2 + \left(\zeta_{i+1}^s - \zeta_i^s\right)^2},$$

$$N_1 = \sum_{i=0}^{N_s/\Delta\omega} n_i^o, N_2 = \sum_{i=0}^{N_s/\Delta\omega} n_i^c,$$

$$n_i^o = \begin{cases} 1 & , when \quad \zeta_i^s > \zeta_i^e \\ 0 & , when \quad \zeta_i^s \le \zeta_i^e \end{cases}, n_i^c = \begin{cases} 1 & , when \quad c_i^s > c_{\max} \\ 0 & , when \quad c_i^s \le c_{\max} \end{cases}.$$

The $\left(\xi_i^s, \eta_i^s, \zeta_i^s\right)$ represents the coordinates of the sampling point numbered i in fixed coordinates. The sampling interval of the sampling points on the path is $\Delta\omega$. The ζ_i^e represents the depth of water at the sampling point numbered i. The c_i^s represents the curvature of the path at the sampling point numbered i. The $c_{\max}$ represents the maximum allowable curvature of the path. The L in the fitness function represents the length of the path. N_1 is the number of sampling points which fall into obstacles. N_2 is the number of sampling points on the path whose curvature does not meet the constraint conditions.

In the process of algorithm evolution, some illegal solutions are often generated. The traditional method is to discard the illegal solution, but this will reduce the efficiency of the algorithm. In fact, some illegal paths may evolve to eligible paths. The fitness function (14) preserves the path with better adaptive value and lower illegal degree, thus improving the efficiency of the algorithm.

2.3.3 The Steps of the IPSO-SP Algorithm

The path planning process based on IPSO-SP is listed as follows:

Step1: Set the number of waypoints $N_s + 1$, the number of particles N_{PSO}, learning parameters b_1 and b_2, inertia weight $w_{\min}$ and $w_{\max}$, the maximum speed $V_{\max}$, the maximum number of iterations $\overline{G}_g$, premature judgment parameters $\overline{G}_L$.

Step2: Set the number of iterations $G_g = 0$, $G_{L,\,i} = 0$, ($i = 1, \cdots, N_{PSO}/2$). The position and velocity of particles are initialized according to the eqs. (12) and (13).

Step3: The parameters of the splines determined by each particle are calculated according to the eqs. (4) and (8). The fitness of each particle is calculated according to eq. (14). The position information and the fitness of each particle are stored as the individual optimum p_i, ($i = 1, \cdots, N_{PSO}$). The individual with optimal fitness values is stored as the global optimum p_g.

Step4: Half of the global search particles with better fitness are selected as the local search particles, while the remaining half of the local search particles select the global search particle according to the probability based on eq. (10). The local search particles are updated based on eq. (9) with $w = w_{\min}$.

Step5: The parameters of the splines determined by each local search particle are calculated according to the eqs. (4) and (8). The fitness of each local search particle is recalculated according to eq. (14). If the latest individual fitness is better than its individual optimal value, then the individual optimal information p_i, ($i = N_{PSO}/2 + 1, \cdots, N_{PSO}$) is updated. If the fitness of local search particle is better than the global optimum, then the global optimum information p_g is updated.

Step6: The crossover operation is carried out for every two adjacent local search particles according to eq. (11). The parameters of the splines determined by each local search particle are recalculated according to the eqs. (4) and (8). The fitness of each local search particle is recalculated according to eq. (14). If the latest individual fitness is better than its individual optimal value, then the individual optimal information p_i, ($i = N_{PSO}/2 + 1, \cdots, N_{PSO}$) is updated. If the latest individual fitness is better than the global optimum, then the global optimum information p_g is updated.

Step7: If the individual optimum of a global search particle is not improved, then $G_{L,\,i} = G_{L,\,i} + 1$. Otherwise, set $G_{L,\,i} = 0$. If $G_{L,i} = \overline{G}_L$, ($i = 1, \cdots, N_{PSO}/2$), then the global search particle is reinitialized, and set $G_{L,\,i} = 0$.

Step8: The global search particles are updated based on eq. (9) with $w = w_{\max}$. The parameters of the splines determined by each global search particle are recalculated according to the eqs. (4) and (8). The fitness of each global search particle is recalculated according to eq. (14). If the latest individual fitness is better than its individual optimal value, then the individual optimal information p_i, ($i = 1, \cdots, N_{PSO}/2$) is updated. If the latest individual fitness is better than the global optimum, then the global optimum information p_g is updated.

Step9: $G_g = G_g + 1$. If $G_g = \overline{G}_g$ is satisfied, then the iteration is ended and the optimal solution is output. Otherwise, return to step4.

3 Analysis of Path Following Problem

The underactuated AUV studied in this paper is equipped with a propeller to control the surge speed, a pair of horizontal rudder to control the pitch, and a pair of vertical rudder to control the yaw, respectively. The AUV is not equipped with driving force in the lateral and vertical directions.

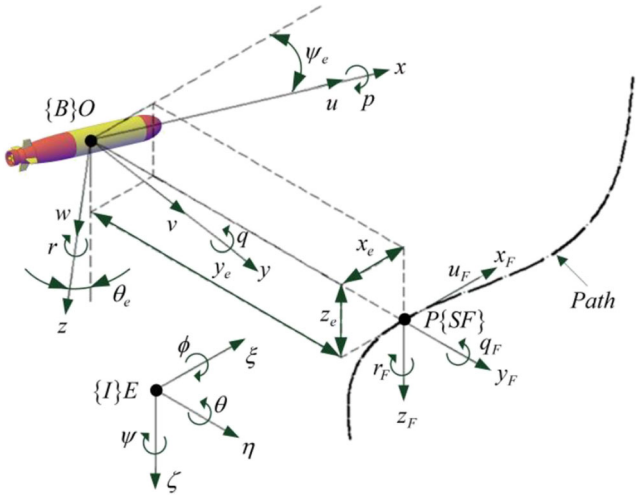

Fig. 1 Path following schematic diagram of AUV

3.1 Kinematics and Dynamics Modeling

As shown in Fig. 1, the equations of motion for AUV in 3D can be built base on fixed coordinate system $\{I\}: E-\xi\eta\zeta$ and moving coordinate system $\{B\}: O-xyz$. The center of buoyancy (*CB*) of AUV is set to the origin of $\{B\}$ coordinate system. Since the roll is very small and needn't to be controlled, the kinematics and dynamics model of AUV can be simplified as the following,

$$\dot{\eta} = J(\eta)v_r + v_w \tag{15}$$

$$M\dot{v}_r + C(v_r)v_r + D(v_r)v_r + g(\eta) = \tau + d \tag{16}$$

where $\eta = [\xi \quad \eta \quad \zeta \quad \theta \quad \psi]^T$, the (ξ, η, ζ) are the coordinates of *CB* defined in $\{I\}$ frame, the (θ, ψ) are the attitude angles of AUV defined in $\{I\}$ frame, θ represents the pitch angle, ψ represents the yaw angle. The $J(\eta)$ represents the rotation transformation from $\{B\}$ frame to $\{I\}$ frame,

$$J(\eta) = \begin{bmatrix} J_1(\eta) & 0_{3\times2} \\ 0_{2\times3} & J_2(\eta) \end{bmatrix}, J_2(\eta) = \begin{bmatrix} 1 & 0 \\ 0 & \frac{1}{cos\theta} \end{bmatrix},$$
$$J_1(\eta) = \begin{bmatrix} \cos\psi\cos\theta & -\sin\psi & \cos\psi\sin\theta \\ \sin\psi\cos\theta & \cos\psi & \sin\psi\sin\theta \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}. \tag{17}$$

The $v_r = [u_r \quad v_r \quad w_r \quad q \quad r]^T$ represents the relative velocity between fluid and AUV which defined in the $\{B\}$ frame, where (u_r, v_r, w_r) are the surge, sway, heave relative velocities, and (q, r) are the pitch, yaw rates. The $v_w = [u_w \quad v_w \quad w_w \quad 0 \quad 0]^T$ represents the velocity of wave which satisfies bounded condition $|v_w| \le \overline{v}_w$. The system matrices $\boldsymbol{M} = \boldsymbol{M}^T$, $\boldsymbol{C}(\boldsymbol{v}) = -\boldsymbol{C}(\boldsymbol{v})^T$, $\boldsymbol{D}(\boldsymbol{v}) > 0$. The restoring moment is defined as $g(\eta) = [0 \quad 0 \quad 0 \quad M_{HS} \quad 0]^T$, the pitch restoring moment $M_{HS} = -z_g G \sin\theta$, where z_g is distance between the center of mass and the *CB*, *G* is the gravity of AUV. The $\tau = [X_\tau \quad 0 \quad 0 \quad \delta_s \quad \delta_r]^T$ represents the control signal, X_τ is the thrust of propeller, δ_r is the vertical rudder angle, and δ_s is the horizontal rudder angle. Vector $\boldsymbol{d}$ describes the model uncertainties. In order to make it convenient for the dynamic controller design, dynamic eq. (16) can be simplified as following,

$$\begin{cases} \dot{u}_r = F_u u_r + F_X X_\tau + d_u, \\ \dot{v}_r = F_v v_r + d_v, \\ \dot{w}_r = F_w w_r + d_w, \\ \dot{q} = F_q q + F_M M_{HS} + F_M k_\delta \delta_s + d_q, \\ \dot{r} = F_r r + F_N k_\delta \delta_r + d_r, \end{cases} \tag{18}$$

where $F_u = \frac{X_u + X_{u|u|}|u_r|}{m - X_u}, F_X = \frac{1}{m - X_u}, F_v = \frac{Y_v + Y_{v|v|}|v_r|}{m - Y_v}, F_w = \frac{Z_w + Z_{w|w|}|w_r|}{m - Z_w}, F_q = \frac{M_q + M_{q|q|}|q|}{I_y - M_q}, F_M = \frac{1}{I_y - M_q}, F_r = \frac{N_r + N_{|r|r}|r|}{I_z - N_r}, F_N = \frac{1}{I_z - N_r}.$

The symbol m represents the mass of the AUV. The I_y, I_z denote the moment of inertia. The $X_{\{\bullet\}}$, $Y_{\{\bullet\}}$, $Z_{\{\bullet\}}$, $M_{\{\bullet\}}$, $N_{\{\bullet\}}$ represent the hydrodynamic parameters. k_δ is the lift coefficient of the rudder. The d_i, $(i = u, v, w, q, r)$ are model uncertainties.

3.2 Error Model of Path Following

As shown in Fig. 1, the virtual target and the tracking errors are defined in the Serret-Frenet frame $\{SF\}: P - x_F y_F z_F$. The control objectives can be expressed as following

$$\begin{cases} \lim\limits_{t\to\infty} u_r(t) = u_{rd}, \lim\limits_{t\to\infty} x_e(t) = 0, \lim\limits_{t\to\infty} y_e(t) = 0, \\ \lim\limits_{t\to\infty} z_e(t) = 0, \lim\limits_{t\to\infty} \theta_e(t) = 0, \lim\limits_{t\to\infty} \psi_e(t) = 0, \end{cases} \tag{19}$$

where u_{rd} is the desired constant relative surge velocity, $(x_e, y_e, z_e, \theta_e, \psi_e)$ are tracking errors defined in the $\{SF\}$ frame. The *P*, which is the origin of $\{SF\}$ frame, is the virtual target. The u_F denotes the surge velocity of *P* along the path. The position and attitude of *P* in $\{I\}$ frame are defined as $P_P = [\xi_P \quad \eta_P \quad \zeta_P \quad \theta_F \quad \psi_F]$, and the position and attitude of *O* in $\{I\}$ frame are defined as $P_o = [\xi_o \quad \eta_o \quad \zeta_o \quad \theta \quad \psi]$. Then, we can calculate the tracking errors as following

$$P_e = [x_e \quad y_e \quad z_e \quad \theta_e \quad \psi_e]^T = R_I^F (P_O - P_P) \tag{20}$$

where R_I^F represents the rotation transformation from $\{I\}$ frame to $\{SF\}$ frame

$$R_I^F = \begin{bmatrix} R_1 & 0_{3\times2} \\ 0_{2\times3} & R_2 \end{bmatrix}, R_2 = \begin{bmatrix} 1 & 0 \\ 0 & \cos\theta_F \end{bmatrix},$$
$$R_1 = \begin{bmatrix} \cos\psi_F\cos\theta_F & \sin\psi_F\cos\theta_F & -\sin\theta_F \\ -\sin\psi_F & \cos\psi_F & 0 \\ \cos\psi_F\sin\theta_F & \sin\psi_F\sin\theta_F & \cos\theta_F \end{bmatrix}. \tag{21}$$

The rotation matrix R_1 and R_2 satisfy the properties $\dot{R}_1 = S_1R_1, \dot{R}_2 = S_2R_2, \chi^T S_1 \chi = 0, \chi \in \Re^3$,

$$S_1 = \begin{bmatrix} 0 & r_F & -q_F \\ -r_F & 0 & -r_F \tan\theta_F \\ q_F & r_F \tan\theta_F & 0 \end{bmatrix},$$

$$S_2 = \begin{bmatrix} 0 & 0 \\ 0 & -q_F \tan\theta_F \end{bmatrix}.$$

Defining $P_{O1} = [\xi_O \quad \eta_O \quad \zeta_O]^T, P_{P1} = [\xi_P \quad \eta_P \quad \zeta_P]^T$, $P_{e1} = [x_e \quad y_e \quad z_e]^T$, then we can get $P_{e1} = R_1(P_{O1} - P_{P1})$, pre-multiply the derivative of P_{e1} by P_{e1}^T, then

$$\begin{aligned} P_{e1}^T \dot{P}_{e1} &= P_{e1}^T \left[\dot{R}_1 (P_{O1} - P_{P1}) + R_1 \left(\dot{P}_{O1} - \dot{P}_{P1}\right)\right] \\ &= P_{e1}^T S_1 P_{e1} + P_{e1}^T R_1 \left(\dot{P}_{O1} - \dot{P}_{P1}\right) \\ &= P_{e1}^T R_1 \left(\dot{P}_{O1} - \dot{P}_{P1}\right). \end{aligned} \tag{22}$$

Eliminate P_{e1}^T on both sides of the eq. (22) we can get the results

$$\dot{P}_{e1} = R_1 \left(\dot{P}_{O1} - \dot{P}_{P1}\right) \tag{23}$$

Using the relations

$$\begin{aligned} R_1 \dot{P}_{P1} &= [u_F \quad 0 \quad 0]^T, \\ R_1 \dot{P}_{O1} &= R_1 \left\{ J_1(\eta) \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} + \begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix} \right\} \\ &= \begin{bmatrix} \cos\psi_e \cos\theta_e & -\sin\psi_e & \cos\psi_e \sin\theta_e \\ \sin\psi_e \cos\theta_e & \cos\psi_e & \sin\psi_e \sin\theta_e \\ -\sin\theta_e & 0 & \cos\theta_e \end{bmatrix} \begin{bmatrix} u_r \\ v_r \\ w_r \end{bmatrix} + \begin{bmatrix} u_{wf} \\ v_{wf} \\ w_{wf} \end{bmatrix} \\ &= \begin{bmatrix} u_r \cos\psi_e \cos\theta_e + d_x \\ u_r \sin\psi_e \cos\theta_e + d_y \\ -u_r \sin\theta_e + d_z \end{bmatrix}, \\ \begin{bmatrix} d_x \\ d_y \\ d_z \end{bmatrix} &= \begin{bmatrix} -v_r \sin\psi_e + w_r \cos\psi_e \sin\theta_e + u_{wf} \\ v_r \cos\psi_e + w_r \sin\psi_e \sin\theta_e + v_{wf} \\ w_r \cos\theta_e + w_{wf} \end{bmatrix}, \begin{bmatrix} u_{wf} \\ v_{wf} \\ w_{wf} \end{bmatrix} = R_1 \begin{bmatrix} u_w \\ v_w \\ w_w \end{bmatrix}. \end{aligned}$$

The (u_{wf}, v_{wf}, w_{wf}) are velocities of wave in $\{SF\}$ frame. Due to the lack of driving force, the v_r and w_r are very small. So the velocities of wave are the dominant part of d_i, $(i = x, y, z)$. It can be assumed that $|d_i| \le \overline{v}_w$. Then, eq. (23) can be rewritten as

$$\dot{P}_{e1} = \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{z}_e \end{bmatrix} = \begin{bmatrix} u_r \cos\psi_e \cos\theta_e - u_F + d_x \\ u_r \sin\psi_e \cos\theta_e + d_y \\ -u_r \sin\theta_e + d_z \end{bmatrix} \tag{24}$$

Defining
$P_{O2} = [\theta \quad \psi]^T, P_{P2} = [\theta_F \quad \psi_F]^T, P_{e2} = [\theta_e \quad \psi_e]^T$, since $P_{e2} = R_2(P_{O2} - P_{P2})$, the derivative of P_{e2} can be computed as

$$\begin{aligned} \dot{P}_{e2} &= \dot{R}_2 (P_{O2} - P_{P2}) + R_2 \left(\dot{P}_{O2} - \dot{P}_{P2}\right) \\ &= S_2 P_{e2} + R_2 \left(\dot{P}_{O2} - \dot{P}_{P2}\right) \\ &= \begin{bmatrix} q - q_F \\ -q_F \psi_e \tan\theta_F + \dfrac{\cos\theta_F}{\cos\theta} r - r_F \end{bmatrix}, \end{aligned} \tag{25}$$

where q_F and r_F are the pitch and yaw rates of $\{SF\}$ frame.

4 Controller Design

4.1 The Design of Virtual Target

The u_F, which is the speed of the virtual target, will be designed to stabilize the along-track error x_e. Firstly, the uncertainty d_x is estimated by the nonlinear observer as follows

$$\begin{cases} \hat{d}_x = d_{xp} + k_{xp} x_e, \\ \dot{d}_{xp} = -k_{xp} d_{xp} - k_{xp} \left(u_r \cos\psi_e \cos\theta_e - u_F + k_{xp} x_e\right), \end{cases} \tag{26}$$

where $k_{xp} > 0$, $\hat{d}_x$ is the estimate value of d_x. Define the estimation error as $\tilde{d}_x = d_x - \hat{d}_x$, then we can get

$$\begin{aligned} \dot{\tilde{d}}_x &= \dot{d}_x - \dot{\hat{d}}_x = \dot{d}_x - \left(\dot{d}_{xp} + k_{xp} \dot{x}_e\right) \\ &= \dot{d}_x - \left(-k_{xp} d_{xp} - k_{xp}^2 x_e + k_{xp} d_x\right) = \dot{d}_x - k_{xp} \tilde{d}_x \\ &\le -k_{xp} \tilde{d}_x + \overline{\dot{d}}_x, \end{aligned} \tag{27}$$

where $\overline{\dot{d}}_x$ is the upper bound of the $\dot{d}_x$, which satisfies the conditions $|\dot{d}_x| \le \overline{\dot{d}}_x$. The Lyapunov function is selected as

$$V_1 = \frac{1}{2} x_e^2 + \frac{1}{2} \tilde{d}_x^2. \tag{28}$$

Designing the control law as follows

$$\begin{cases} u_F = u_r \cos\psi_e \cos\theta_e + k_1 x_e + \hat{d}_x + \varepsilon_1 \mathrm{sgn}(x_e), \\ \varepsilon_1 = k_2 \left(1 - e^{-k_3 |x_e|}\right), \end{cases} \tag{29}$$

where $k_1 > 0, k_2 > 0, k_3 > 0$, the sgn is sign function. In order to reduce the chattering, the switch gains ε_1 is adjusted between 0 to k_2 based on the tracking error. Then, the derivative of V_1 becomes

$$\begin{aligned} \dot{V}_1 &= x_e \dot{x}_e + \tilde{d}_x \dot{\tilde{d}}_x \\ &= x_e \left(u_r \cos\psi_e \cos\theta_e - u_F + \hat{d}_x\right) - k_{xp} \tilde{d}_x^2 + x_e \tilde{d}_x + \tilde{d}_x \dot{d}_x \\ &\le x_e \left(u_r \cos\psi_e \cos\theta_e - u_F + \hat{d}_x\right) - \left(k_{xp} - 1\right) \tilde{d}_x^2 + \frac{1}{2} x_e^2 + \frac{1}{2} \overline{\dot{d}}_x^2 \\ &\le -\left(k_1 - \frac{1}{2}\right) x_e^2 - \left(k_{xp} - 1\right) \tilde{d}_x^2 - \varepsilon_1 |x_e| + \frac{1}{2} \overline{\dot{d}}_x^2 \\ &\le -2 k_{V_1} V_1 + \Delta_{V_1}, \end{aligned}$$

where $k_{V_1} = \min\left[\left(k_1 - \frac{1}{2}\right), \left(k_{xp} - 1\right)\right]$, $\Delta_{V_1} = \frac{1}{2}\overline{d}_x^2$.

The V_1 is uniformly bounded as long as the requirements $k_1 > 0.5$, $k_{xp} > 1$are fulfilled. In fact, the conditions are very conservative. Since the spline is parameterized by ω, the update law of ω can be calculated by the following equation

$$\dot{\omega} = \frac{u_F}{\sqrt{\overline{\xi}'_k(\omega)^2 + \overline{\eta}_k'(\omega)^2 + \overline{\zeta}_k'(\omega)^2}} \tag{30}$$

4.2 The MPC Guidance Law

Traditional LOS guidance, which imitates the actions of a helmsman, has several nice properties. However, it is sensitive to the disturbance of waves. MPC is known as an optimization-based control method. In addition, there have been many studies on the stability analysis of MPC, such as literature [45, 46]. Next, MPC will be applied to design the guidance law, which can make up for the deficiency of traditional LOS guidance. Considering that the response of the dynamic controller to input has a certain delay. The response of the dynamic controller to the desired attitude is approximated by the following equations

$$\dot{\theta}_e = \frac{1}{T_1}(\theta_{ed} - \theta_e), \dot{\psi}_e = \frac{1}{T_2}(\psi_{ed} - \psi_e) \tag{31}$$

where $T_1 > 0$ and $T_2 > 0$ are adjustable time constant, θ_{ed} and ψ_{ed} are the desired approach angles in vertical and horizontal planes.

4.2.1 The Predictive Model

According to the Eqs. (24) (31), the control objective can be equivalent to the stabilization control problem for the following error variables

$$\begin{bmatrix} \dot{y}_e \\ \dot{z}_e \\ \dot{\theta}_e \\ \dot{\psi}_e \end{bmatrix} = \begin{bmatrix} u_r \sin\psi_e \cos\theta_e + d_y \\ -u_r \sin\theta_e + d_z \\ \frac{1}{T_1}(\theta_{ed} - \theta_e) \\ \frac{1}{T_2}(\psi_{ed} - \psi_e) \end{bmatrix} = \begin{bmatrix} u_r k_y \psi_e + d_y \\ -u_r k_z \theta_e + d_z \\ \frac{1}{T_1}(\theta_{ed} - \theta_e) \\ \frac{1}{T_2}(\psi_{ed} - \psi_e) \end{bmatrix} \tag{32}$$

where $k_y = \frac{\sin\psi_e}{\psi_e}\cos\theta_e$, $k_z = \frac{\sin\theta_e}{\theta_e}$. In this paper, k_y is set to $k_y = \cos\theta_e$ when $|\psi_e| \le \frac{\pi}{12}$, k_z is set to $k_z = 1$ when $|\theta_e| \le \frac{\pi}{12}$.

The system (32) can be can be written as

$$\dot{x} = f(x, u) \tag{33}$$

where $x = [y_e \quad z_e \quad \theta_e \quad \psi_e]^T$ is state variables, $u = [\theta_{ed} \quad \psi_{ed}]^T$ is input variables. Obviously, the equilibrium points of system (33) are $x = 0$ and $u = 0$. Then, we can discretize the system as follows

$$\begin{cases} x_{k+1,k} = A_k x_{k,k} + B_k u_{k,k}, \\ y_{k,k} = C_k x_{k,k}, \end{cases} \tag{34}$$

$$A_k = \begin{bmatrix} 1 & 0 & 0 & Tu_{rd}k_y \\ 0 & 1 & -Tu_{rd}k_z & 0 \\ 0 & 0 & 1 - \frac{T}{T_1} & 0 \\ 0 & 0 & 0 & 1 - \frac{T}{T_2} \end{bmatrix}, B_k = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ \frac{T}{T_1} & 0 \\ 0 & \frac{T}{T_2} \end{bmatrix}, C_k = I_4,$$

where T represents the sampling time.

Using the predictive model (34), the predictive value of the state variables are

$$\begin{aligned} x_{k+1,k} &= A_k x_{k,k} + B_k u_{k,k}, \\ x_{k+2,k} &= A_k x_{k+1,k} + B_k u_{k+1,k} \\ &= A_k^2 x_{k,k} + A_k B_k u_{k,k} + B_k u_{k+1,k}, \\ x_{k+3,k} &= A_k x_{k+2,k} + B_k u_{k+2,k} \\ &= A_k^3 x_{k,k} + A_k^2 B_k u_{k,k} + A_k B_k u_{k+1,k} + B_k u_{k+2,k}, \\ &\vdots \\ x_{k+N_p,k} &= A_k^{N_p} x_{k,k} + A_k^{N_p-1} B_k u_{k,k} + \cdots + A_k^{N_p-N_c} B_k u_{k+N_c-1,k}, \end{aligned}$$

where N_c, N_p represent the control horizon and the prediction horizon. Then, the predictive value of the output variables can be calculated as.

$$\begin{aligned} y_{k+1,k} &= C_k A_k x_{k,k} + C_k B_k u_{k,k}, \\ y_{k+2,k} &= C_k A_k^2 x_{k,k} + C_k A_k B_k u_{k,k} + C_k B_k u_{k+1,k}, \\ y_{k+3,k} &= C_k A_k^3 x_{k,k} + C_k A_k^2 B_k u_{k,k} + C_k A_k B_k u_{k+1,k} + C_k B_k u_{k+2,k}, \\ &\vdots \\ y_{k+N_p,k} &= C_k A_k^{N_p} x_{k,k} + C_k A_k^{N_p-1} B_k u_{k,k} + \cdots + C_k A_k^{N_p-N_c} B_k u_{k+N_c-1,k}, \end{aligned}$$

Finally, it is rewritten into the matrix form as

$$Y_{k+1,k} = \Psi x_{k,k} + \Theta U_{k,k} \tag{35}$$

where

$$Y_{k+1,k} = \begin{bmatrix} y_{k+1,k} \\ y_{k+2,k} \\ y_{k+3,k} \\ \vdots \\ y_{k+N_p,k} \end{bmatrix}, U_{k,k} = \begin{bmatrix} u_{k,k} \\ u_{k+1,k} \\ u_{k+2,k} \\ \vdots \\ u_{k+N_c-1,k} \end{bmatrix}, \Psi = \begin{bmatrix} C_k A_k \\ C_k A_k^2 \\ C_k A_k^3 \\ \vdots \\ C_k A_k^{N_p} \end{bmatrix},$$

$$\Theta = \begin{bmatrix} C_k B_k & 0 & 0 & \cdots & 0 \\ C_k A_k B_k & C_k B_k & 0 & \cdots & 0 \\ C_k A_k^2 B_k & C_k A_k B_k & C_k B_k & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ C_k A_k^{N_p-1} B_k & C_k A_k^{N_p-2} B_k & C_k A_k^{N_p-3} B_k & \cdots & C_k A_k^{N_p-N_c} B_k \end{bmatrix}$$

4.2.2 The Control Constraint

Here, the control constraints are considered as

$$u_{\min} \le u_{k+t,k} \le u_{\max}, t = 0, 1, \cdots N_c - 1, \tag{36}$$
$$u_{\min} = \begin{bmatrix} -\frac{\pi}{4} & -\frac{\pi}{2} \end{bmatrix}^T, u_{\max} = \begin{bmatrix} \frac{\pi}{4} & \frac{\pi}{2} \end{bmatrix}^T.$$

The constraints conditions (36) can be translated into linear inequalities as following

$$\begin{bmatrix} M_1 \\ -M_1 \end{bmatrix} U_{k,k} \le \begin{bmatrix} N_{\max} \\ N_{\min} \end{bmatrix},$$
$$M_1 = I_{N_c} \otimes I_2, N_{\max} = 1_{N_c} \otimes u_{\max}, \tag{37}$$
$$N_{\min} = -1_{N_c} \otimes u_{\min}, 1_{N_c}^T = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \end{bmatrix}_{1 \times N_c}.$$

The symbol $\otimes$ represents the kronecker product.

The next step is to design stability condition to ensure the stability of the MPC. Consider the Lyapunov function

$$V_2 = \frac{1}{2}\left(y_e^2 + z_e^2\right) \tag{38}$$

First, let's assume that $u_r = u_{rd}$, $\theta_e = \theta_{ed}$, $\psi_e = \psi_{ed}$ and ignore the uncertain terms, then according to the error eq. (24), the derivative of V_2 is equal to

$$\begin{aligned} \dot{V}_2 &= y_e \dot{y}_e + z_e \dot{z}_e \\ &= y_e u_{rd} \sin\psi_{ed} \cos\theta_{ed} - z_e u_{rd} \sin\theta_{ed} \\ &= k_\psi |y_e| u_{rd} \frac{\sin\psi_{ed}}{\psi_{ed}} \cos\theta_{ed} + k_\theta |z_e| u_{rd} \frac{\sin\theta_{ed}}{\theta_{ed}}, \end{aligned} \tag{39}$$

where $k_\psi = \mathrm{sgn}(y_e)\psi_{ed}$, $k_\theta = -\mathrm{sgn}(z_e)\theta_{ed}$. Obviously, as long as conditions $k_\psi \le 0$ and $k_\theta \le 0$ are satisfied, then $\dot{V}_2 \le 0$. The stability condition can be translated into linear inequalities as following

$$\begin{cases} k_\theta = M_2 U_{k,k} \le 0, \\ k_\psi = M_3 U_{k,k} \le 0, \end{cases} M_2 = -\mathrm{sgn}(z_e) C_\theta, C_\theta = \begin{bmatrix} 1 & 0_{1\times(N_c-1)} \end{bmatrix} \otimes \begin{bmatrix} 1 & 0 \end{bmatrix}, M_3 = \mathrm{sgn}(y_e) C_\psi, C_\psi = \begin{bmatrix} 1 & 0_{1\times(N_c-1)} \end{bmatrix} \otimes \begin{bmatrix} 0 & 1 \end{bmatrix}. \tag{40}$$

The stability condition parameterize by parameter vector$U_{k,k}$, making it convenient to solve the optimization problem.

4.2.3 Optimization with Control Constraint

The cost function at time k can be selected as

$$J_{k,k} = Y_{k+1,k}^T \overline{Q} Y_{k+1,k} + U_{k,k}^T \overline{R} U_{k,k} \tag{41}$$

where $\overline{Q} = I_{N_p} \otimes Q, \overline{R} = I_{N_c} \otimes R, Q = diag\ (Q_{11}, Q_{22}, Q_{33}, Q_{44})$ and $R = diag\ (R_{11}, R_{22})$are positive definite weighting matrices. In order to reduce the ocean wave disturbances, the weight coefficient will be adjusted as following

$$Q_{11} = \begin{cases} Q_{\max} & , |y_e| \ge \Delta_L \\ Q_{\min} & , |y_e| < \Delta_L \end{cases}, Q_{22} = \begin{cases} Q_{\max} & , |z_e| \ge \Delta_L \\ Q_{\min} & , |z_e| < \Delta_L \end{cases} \tag{42}$$

where $Q_{\min} \ge 0$, $Q_{\max} > 0$ are the weight coefficient, $\Delta_L \ge 0$ is the boundary layer distance.

To find the optimal $U_{k,k}$ that will minimize $J_{k,k}$, by inserting eq. (35) into eq. (41), $J_{k,k}$ is expressed as $J_{k,k} = \left[\Psi x_{k,k}\right]^T \overline{Q} \Psi x_{k,k} + U_{k,k}^T E U_{k,k} + 2 U_{k,k}^T F$, where $E = \Theta^T \overline{Q} \Theta + \overline{R}$, $F = \Theta^T \overline{Q} \Psi x_{k,k}$.

Since $\left[\Psi x_{k,k}\right]^T \overline{Q} \Psi x_{k,k}$ is constant, the optimization process of MPC is equivalent to solving the quadratic programming problems

$$\min_{U_{k,k}} J_{k,k} = U_{k,k}^T E U_{k,k} + 2 U_{k,k}^T F, s.t. \begin{bmatrix} M_1 \\ -M_1 \\ M_2 \\ M_3 \end{bmatrix} U_{k,k} \le \begin{bmatrix} N_{\max} \\ N_{\min} \\ 0 \\ 0 \end{bmatrix} \tag{43}$$

By solving eq. (43), we can get the control vector $U_{k,\,k}$. Then, the first element of the control sequence $u_{k,\,k}$ is implemented as the actual control input. When the next sample period arrives, the more recent measurement is taken to form the state vector for calculation of the new sequence of control signal. This procedure is repeated in real time to give the receding horizon control law.

4.3 The Dynamic Controller

4.3.1 The Control of Relative Surge Velocity

Define error variables $\tilde{u}_r = u_r - u_{rd}$, then choosing sliding mode function as follows

$$s_1 = c_1 \tilde{u}_r + \dot{\tilde{u}}_r = c_1 \tilde{u}_r + F_u u_r + F_X X_\tau + d_u \tag{44}$$

where $c_1 > 0$. Since u_{rd} is constant, the system can be considered as autonomous. So we can assume that d_i, ($i = u, v, w, q, r$)are slowly varying. Define $\tilde{d}_u = d_u - \hat{d}_u$, where$\hat{d}_u$ is the estimate values of d_u. Consider the Lyapunov function

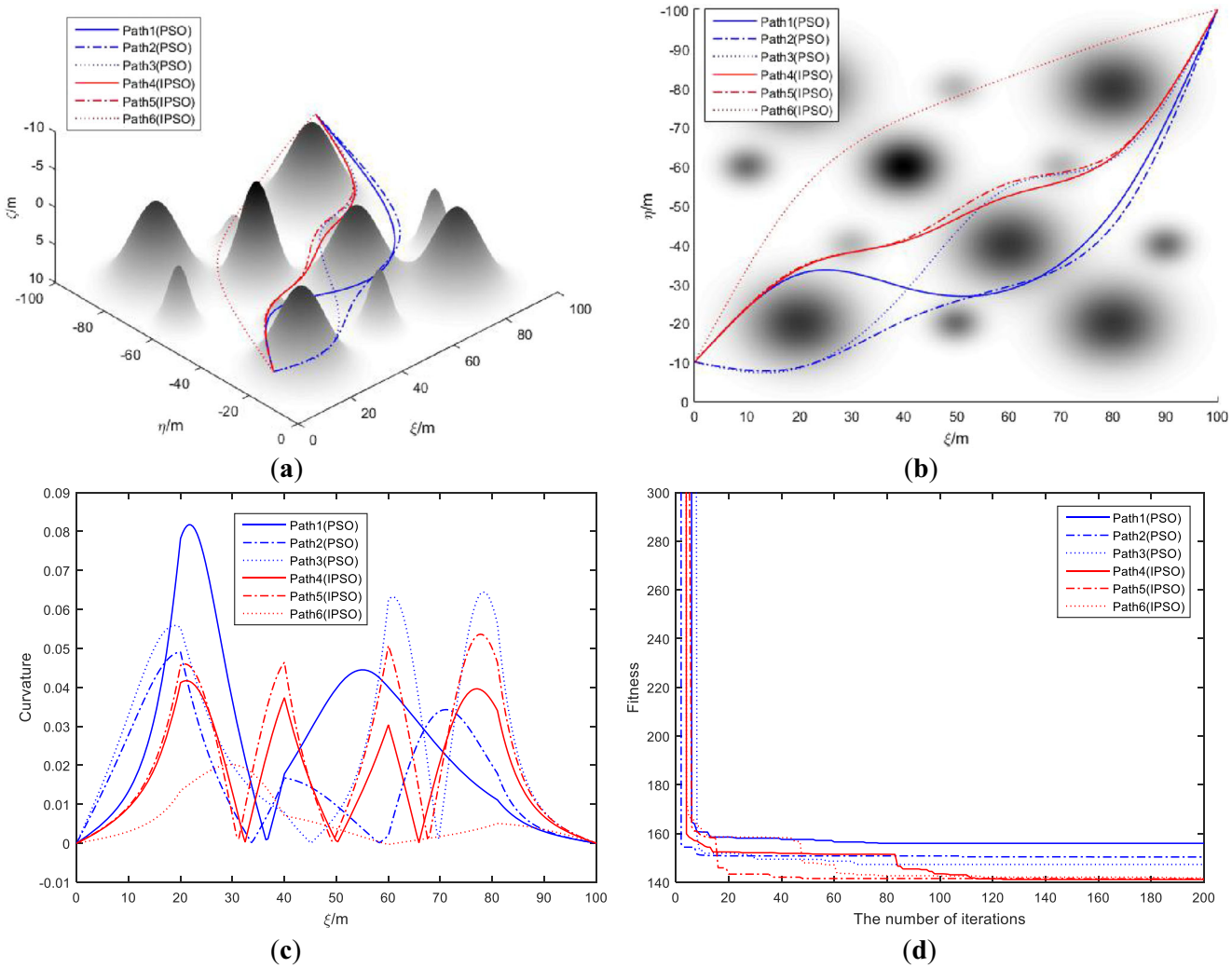

Fig. 2 Path planning results. (**a**) The paths are displayed in three-dimensional space. (**b**) The paths are shown in horizontal plane. (**c**) The curvature of the paths. (**d**) The fitness curves

$$V_2 = \frac{1}{2}\tilde{u}_r^2 + \frac{1}{2k_4}\tilde{d}_u^2 + \frac{1}{2}s_1^2 \tag{45}$$

Designing the control law and adaptive law as

$$\begin{cases} \dot{X}_\tau = \dfrac{1}{F_X}\Big[-c_1\Big(F_u u_r + F_X X_\tau + \hat{d}_u\Big) - F_u \dot{u}_r - \tilde{u}_r \\ \qquad\qquad -k_5 s_1 - \varepsilon_2 \mathrm{sgn}(s_1)\Big], \\ \dot{\hat{d}}_u = k_4 c_1 s_1, \varepsilon_2 = k_2\Big(1 - e^{-k_3|s_1|}\Big), \end{cases} \tag{46}$$

where $k_4 > 0$, $k_5 > 0$.

Differentiating V_2 with respect to time, then.

$$\begin{aligned} \dot{V}_2 &= \tilde{u}_r\dot{\tilde{u}}_r - \frac{1}{k_4}\tilde{d}_u\dot{\hat{d}}_u + s_1\dot{s}_1 \\ &= \tilde{u}_r s_1 - c_1\tilde{u}_r^2 - \frac{1}{k_4}\tilde{d}_u\dot{\hat{d}}_u + s_1\left[c_1(F_u u_r + F_X X_\tau + d_u) + F_u\dot{u}_r + F_X\dot{X}_\tau\right] \\ &= \tilde{u}_r s_1 - c_1\tilde{u}_r^2 + \tilde{d}_u\left(-\frac{1}{k_4}\dot{\hat{d}}_u + c_1 s_1\right) \\ &+ s_1\left[c_1\left(F_u u_r + F_X X_\tau + \hat{d}_u\right) + F_u\dot{u}_r + F_X\dot{X}_\tau\right] = -c_1\tilde{u}_r^2 - k_5 s_1^2 - \varepsilon_2|s_1| \le 0. \end{aligned}$$

The speed of the propeller can be calculated according to the hydrodynamic model as following

$$n_p = \sqrt{\frac{X_\tau}{K_T \rho D_p^4}} \tag{47}$$

where K_T, ρ, D_P denote thrust coefficient, fluid density, diameter of the propeller, respectively.

4.3.2 The Control of Pitch Angle

Define error variables $\tilde{\theta}_e = \theta_e - \theta_{ed}$, consider Lyapunov function as

$$V_3 = \frac{1}{2}\tilde{\theta}_e^2 \tag{48}$$

Differentiating V_3 with respect to time, we can get $\dot{V}_3 = \tilde{\theta}_e\dot{\tilde{\theta}}_e = \tilde{\theta}_e(q - q_F - \dot{\theta}_{ed})$. Designing the desired value of

the pitch rate as $q_d = \dot{\theta}_{ed} + q_F - k_6\tilde{\theta}_e$, then $\dot{V}_3 = -k_6\tilde{\theta}_e^2 + \tilde{\theta}_e\tilde{q}$, where $k_6 > 0$, $\tilde{q} = q - q_d$. Choosing sliding mode function as follows

$$\begin{aligned} s_2 &= c_2\tilde{q} + \dot{\tilde{q}} + \tilde{\theta}_e \\ &= c_2\tilde{q} + F_q q + F_M M_{HS} + F_M k_\delta \delta_s + d_q - \dot{q}_d + \tilde{\theta}_e, \end{aligned} \tag{49}$$

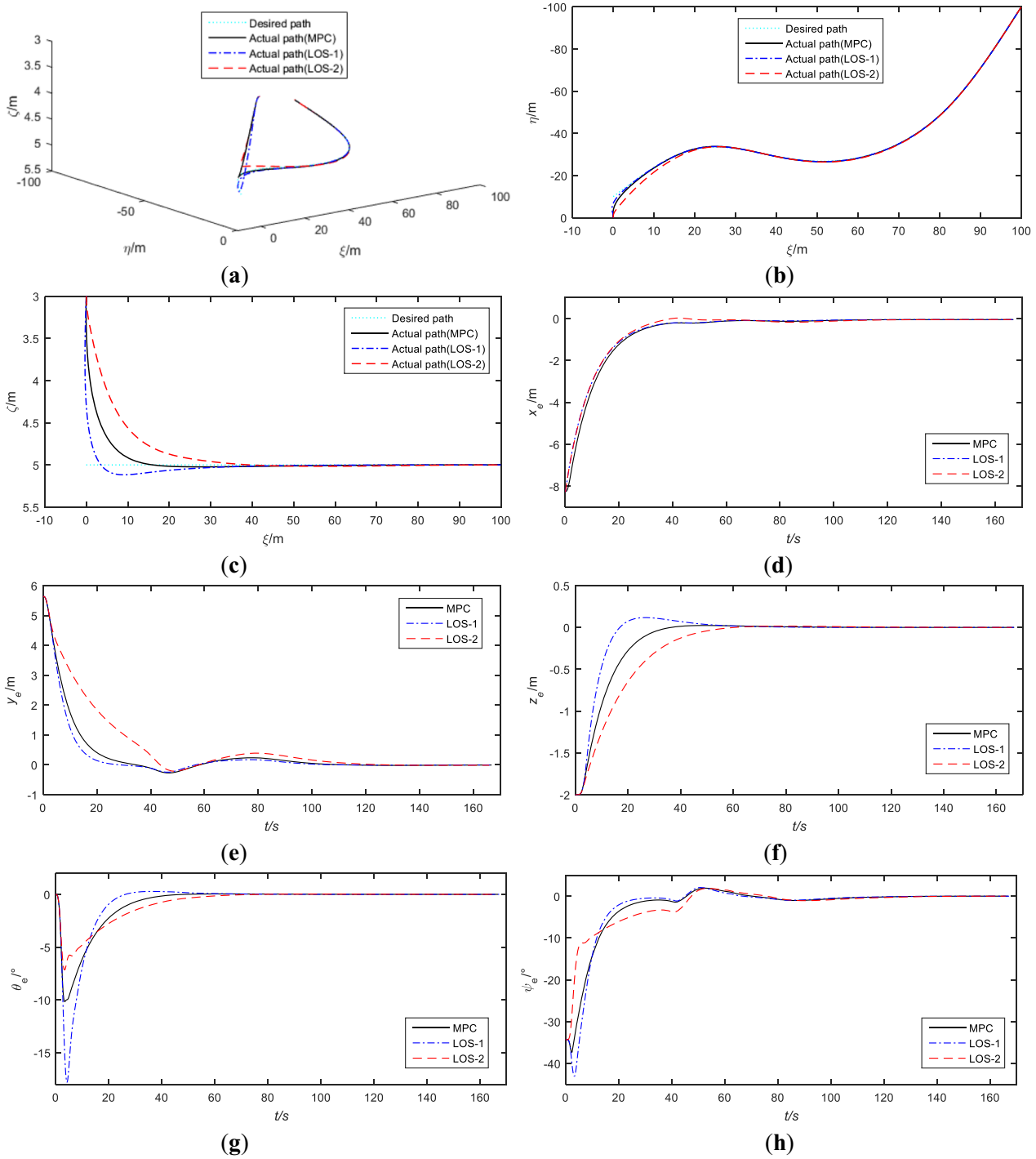

Fig. 3 Simulation results of path following in still water. (**a**) The simulation results are displayed in three-dimensional space. (**b**) The simulation results are displayed in horizontal plane. (**c**) The simulation results are shown in vertical plane. (**d**) The along-track error. (**e**) The lateral error. (**f**) The vertical error. (**g**) The pitch angle error. (**h**) The yaw angle error. (**i**) The desired pitch angle error. (**j**) The desired yaw angle error. (**k**) The stability condition. (**l**) The surge relative velocity of AUV. (**m**) The speed of the virtual target. (**n**) The sway relative velocity of AUV. (o) The heave relative velocity of AUV. (p) The speed of the propeller. (q) The horizontal rudder angle. (r) The vertical rudder angle

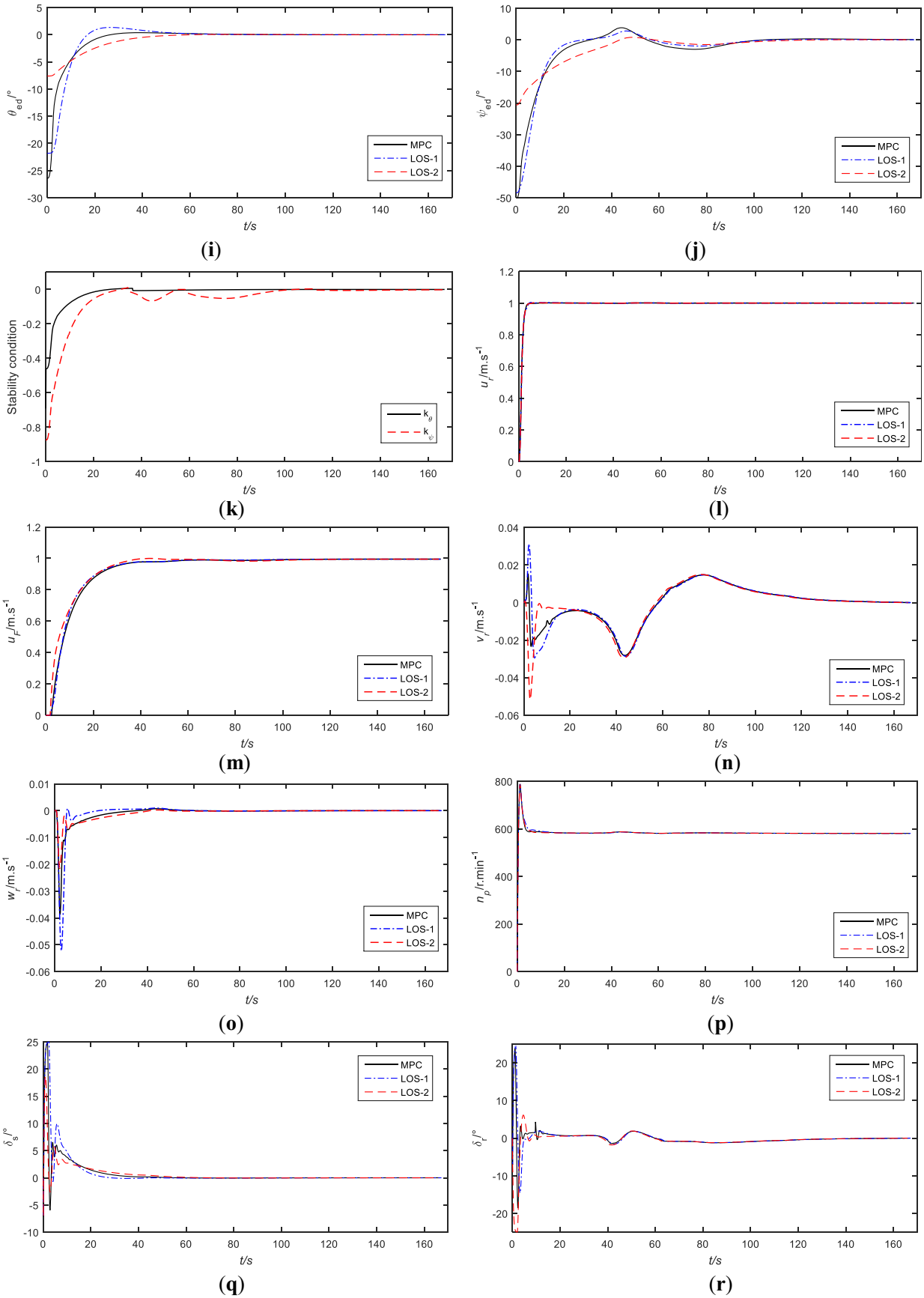

Fig. 3 (continued)

where $c_2 > 0$. Define $\tilde{d}_q = d_q - \hat{d}_q$, where $\hat{d}_q$ is the estimate values of d_q. Consider the Lyapunov function

$$V_4 = V_3 + \frac{1}{2}\tilde{q}^2 + \frac{1}{2k_7}\tilde{d}_q^2 + \frac{1}{2}s_2^2 \tag{50}$$

Designing the control law and adaptive law as

$$\begin{cases} \dot{\delta}_s = \dfrac{1}{F_M k_\delta}\Big[-c_2\Big(F_q q + F_M M_{HS} + F_M k_\delta \delta_s + \hat{d}_q - \dot{q}_d\Big) \\ \quad -F_q\dot{q} - F_M\dot{M}_{HS} + \ddot{q}d - \tilde{\theta}_e - \tilde{q} - k_8 s_2 - \varepsilon_3 \mathrm{sgn}(s_2)\Big], \\ \quad \dot{\hat{d}}_q = k_7 c_2 s_2, \varepsilon_3 = k_2\Big(1 - e^{-k_3|s_2|}\Big), \end{cases} \tag{51}$$

where $k_7 > 0$, $k_8 > 0$, then, the derivative of V_4 becomes

$$\begin{aligned} \dot{V}_4 &= \dot{V}_3 + \tilde{q}\dot{\tilde{q}} - \frac{1}{k_7}\tilde{d}_q\dot{\hat{d}}_q + s_2\dot{s}_2 \\ &= -k_6\tilde{\theta}_e^2 - c_2\tilde{q}^2 + \tilde{q}s_2 + \tilde{d}_q\Big(-\frac{1}{k_7}\dot{\hat{d}}_q + c_2 s_2\Big) \\ &\quad + s_2\Big[c_2\big(F_q q + F_M M_{HS} + F_M k_\delta \delta_s + \hat{d}_q - \dot{q}_d\big) \\ &\quad + F_q\dot{q} + F_M\dot{M}_{HS} + F_M k_\delta \dot{\delta}_s - \ddot{q}_d + \dot{\tilde{\theta}}_e\Big] \\ &= -k_6\tilde{\theta}_e^2 - c_2\tilde{q}^2 - k_8 s_2^2 - \varepsilon_3|s_2| \le 0. \end{aligned}$$

4.3.3 The Control of Yaw Angle

Define error variables $\tilde{\psi}_e = \psi_e - \psi_{ed}$, consider Lyapunov function as

$$V_5 = \frac{1}{2}\tilde{\psi}_e^2 \tag{52}$$

Designing the desired value of the yaw rate as $r_d = \Big(\dot{\psi}_{ed} + q_F\psi_e \tan\theta_F + r_F - k_9\tilde{\psi}_e\Big)\cos\theta/\cos\theta_F$, then $\dot{V}_5 = -k_9\tilde{\psi}_e^2 + k_r\tilde{r}$, where $k_9 > 0$, $k_r = \tilde{\psi}_e\cos\theta_F/\cos\theta$, $\tilde{r} = r - r_d$. Choosing sliding mode function as

$$s_3 = c_3\tilde{r} + \dot{\tilde{r}} + k_r = c_3\tilde{r} + F_r r + F_N k_\delta \delta_r + d_r - \dot{r}_d + k_r \tag{53}$$

where $c_3 > 0$. Define $\tilde{d}_r = d_r - \hat{d}_r$, where $\hat{d}_r$ is the estimate values of d_r. Consider the Lyapunov function

$$V_6 = V_5 + \frac{1}{2}\tilde{r}^2 + \frac{1}{2k_{10}}\tilde{d}_r^2 + \frac{1}{2}s_3^2 \tag{54}$$

Designing the control law and adaptive law as

$$\begin{cases} \dot{\delta}_r = \dfrac{1}{F_N k_\delta}\Big[-c_3\Big(F_r r + F_N k_\delta \delta_r + \hat{d}_r - \dot{r}_d\Big) \\ \quad -F_r\dot{r} + \ddot{r}_d - \dot{k}_r - \dot{\tilde{r}} - k_{11}s_3 - \varepsilon_4 \mathrm{sgn}(s_3)\Big], \\ \quad \dot{\hat{d}}_r = k_{10}c_3 s_3, \varepsilon_4 = k_2\Big(1 - e^{-k_3|s_3|}\Big), \end{cases} \tag{55}$$

where $k_{10} > 0$, $k_{11} > 0$, then, the derivative of V_6 becomes

$$\begin{aligned} \dot{V}_6 &= \dot{V}_5 + \dot{\tilde{r}}\tilde{r} - \frac{1}{k_{10}}\tilde{d}_r\dot{\hat{d}}_r + s_3\dot{s}_3 \\ &= -k_9\tilde{\psi}_e^2 - c_3\tilde{r}^2 + \tilde{r}s_3 + \tilde{d}_r\Big(-\frac{1}{k_{10}}\dot{\hat{d}}_r + c_3 s_3\Big) \\ &\quad + s_3\Big[c_3\Big(F_r r + F_N k_\delta \delta_r + \hat{d}_r - \dot{r}_d\Big) + F_r\dot{r} + F_N k_\delta \dot{\delta}_r - \ddot{r}_d + \dot{k}_r\Big] \\ &= -k_9\tilde{\psi}_e^2 - c_3\tilde{r}^2 - k_{11}s_3^2 - \varepsilon_4|s_3| \le 0. \end{aligned}$$

4.4 Stability Analysis of Closed-Loop Systems

According to control law (29), error eq. (24) can be rewritten as

$$\begin{aligned} \dot{P}_{e1} &= \begin{bmatrix} \dot{x}_e \\ \dot{y}_e \\ \dot{z}_e \end{bmatrix} \\ &= \begin{bmatrix} -k_1 x_e - \varepsilon_1 \mathrm{sgn}(x_e) + \tilde{d}_x \\ \Big(u_{rd} + \tilde{u}_r\Big)\sin\Big(\psi_{ed} + \tilde{\psi}_e\Big)\cos\Big(\theta_{ed} + \tilde{\theta}_e\Big) + d_y \\ -\Big(u_{rd} + \tilde{u}_r\Big)\sin\Big(\theta_{ed} + \tilde{\theta}_e\Big) + d_z \end{bmatrix} \end{aligned} \tag{56}$$

Then, the Eq. (56) can be expanded as following by applying the trigonometric properties,

$$\dot{P}_{e1} = A_e + B_e d_{e1} + d_{e2} \tag{57}$$

$$\begin{aligned} &A_e = \begin{bmatrix} -k_1 x_e - \varepsilon_1 \mathrm{sgn}(x_e) \\ u_{rd}\sin\psi_{ed}\cos\theta_{ed} \\ -u_{rd}\sin\theta_{ed} \end{bmatrix}, B_e = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & B_{22} & B_{23} & B_{24} \\ 0 & B_{32} & B_{33} & 0 \end{bmatrix}, \\ &d_{e1} = \begin{bmatrix} \tilde{d}_x & \tilde{u}_r & \tilde{\theta}_e & \tilde{\psi}_e \end{bmatrix}^T, d_{e2} = [0 \quad d_y \quad d_z]^T, B_{22} = \sin\psi_e\cos\theta_e, \\ &B_{23} = u_{rd}\sin\psi_{ed}\cos\tilde{\psi}_e\left[\cos\theta_{ed}\frac{\Big(\cos\tilde{\theta}_e - 1\Big)}{\tilde{\theta}_e} - \sin\theta_{ed}\frac{\sin\tilde{\theta}_e}{\tilde{\theta}_e}\right], \\ &B_{24} = u_{rd}\sin\psi_{ed}\cos\theta_{ed}\frac{\Big(\cos\tilde{\psi}_e - 1\Big)}{\tilde{\psi}_e} + u_{rd}\cos\psi_{ed}\cos\theta_e\frac{\sin\tilde{\psi}_e}{\tilde{\psi}_e}, \\ &B_{32} = -\sin\theta_e, B_{33} = u_{rd}\sin\theta_{ed}\frac{\Big(1 - \cos\tilde{\theta}_e\Big)}{\tilde{\theta}_e} - u_{rd}\cos\theta_{ed}\frac{\sin\tilde{\theta}_e}{\tilde{\theta}_e}. \end{aligned}$$

The system (57) can be equivalent to the $\dot{P}_{e1} = A_e$ perturbed by the d_{e1} and d_{e2}. Since the stability of $\dot{P}_{e1} = A_e$ is guaranteed by the kinematics controller, moreover, d_{e1} is uniform global asymptotic stability and B_e, d_{e2} are bounded, the P_{e1} is uniformly ultimately bounded (UUB) can be concluded.

5 Simulation Results

5.1 Path Planning Based on IPSO-SP

The path in horizontal plane starts at (ξ_0, η_0, ζ_0)= (0, −10m, 5m)and ends at$(\xi_{N_s}, \eta_{N_s}, \zeta_{N_s})$ = (100, −100m, 5m). The depth is set to 5 m and does not participate in optimization. The underwater environment information is modeled as follows

$$\zeta^e(\xi,\eta) = 10 - \sum_{i=1}^{12} H_o(i) e^{-k_\xi(i)[\xi-\xi_o(i)]^2 - k_\eta(i)[\eta-\eta_o(i)]^2} \tag{58}$$

$$\begin{aligned}
&\xi_o = [20, 50, 80, 10, 40, 70, 30, 60, 90, 20, 50, 80],\\
&\eta_o = -[80, 80, 80, 60, 60, 60, 40, 40, 40, 20, 20, 20],\\
&H_o = [11, 4, 11, 8, 14, 4, 4, 11, 8, 11, 8, 11],\\
&k_\xi = k_\eta = 0.01 \times [1, 5, 1, 5, 2, 5, 5, 1, 5, 1, 5, 1].
\end{aligned}$$

The $\zeta^e(\xi, \eta)$ represents the depth of water at any location. The (ξ_o, η_o) represents the coordinates of the center of the obstacle in the horizontal plane. The $H_o$represents the height of obstacles. The $k_\xi$and $k_\eta$represent the slope of obstacles. Since the minimum turning radius of AUV is 10 m, the maximum allowable curvature of the path is $c_{\max} = 0.1$. The main parameters of the IPSO-SP algorithm are set as $N_s = 5$, $N_{PSO} = 20$, $b_1 = 2$, $b_2 = 2$, $w_{\max} = 0.8$, $w_{\min} = 0.5$, $V_{\max} = 1$,$\overline{G}_g = 200$,$\overline{G}_L = 10$, $\Delta\omega = 0.02$. The results of path planning are shown in Fig. 2. Paths 1–3 are results of planning with the traditional PSO algorithm and paths 4–6 are the results of planning with the improved PSO (IPSO) algorithm. The planning results show that both methods can work out qualified paths under the premise of sufficient iteration times. All paths can avoid obstacles and the maximum curvature meets the constraints. The average fitness values of the two methods are 151.2 and 141.4, respectively. The traditional PSO algorithm has a short convergence time. However, the optimization results are basically not improved in the later stage, which indicating that the algorithm falls into the premature problem. Although the convergence rate of the IPSO algorithm is a little slower, it can effectively improve the premature problem, so that a better path can be obtained.

5.2 Path Following in Still Water

Firstly, path following control simulation experiment is performed when there is no environmental disturbance. The AUV model is the REMUS 100 [47]. The controller designed parameters are set as $k_{xp} = 1.1$, $k_1 = 0.6$, $k_2 = 0.2$, $k_3 = 30$, $k_4 = 0.1$, $k_5 = 0.1$, $k_6 = 0.1$, $k_7 = 0.1$, $k_8 = 0.1$, $k_9 = 0.1$, $k_{10} = 0.1$, $k_{11} = 0.1$, $T = 0.1$, $T_1 = 1$, $T_2 = 1$, $N_c = 3$, $N_p = 6$, $Q_{\max} = 1$, $Q_{\min} = 0.02$, $Q_{33} = 2$, $Q_{44} = 1$, $R_{11} = 1$, $R_{22} = 1$, $\Delta_L = 0$, $c_1 = 2$ $c_2 = 0.3$, $c_3 = 0.3$. Since the first path (path1) has the greatest curvature, it is selected as the desired path, which is more convenient to verify the tracking performance of the controller. The initial states of the AUV are $P_o(0) = [0 \quad 0 \quad 3 \quad 0 \quad -\pi/2]$. The desired forward speed u_{rd} is set to 1 ms^{-1}. Two different guidance laws are applied. The first is the MPC guidance laws. The second one is LOS guidance law and two different lookahead distance 5 m (LOS-1) and 15 m (LOS-2) are adopted in simulation. The simulation results are displayed in Fig. 3. According to Figs. 3a-h, we can see that path following results of both the MPC and LOS guidance law is nearly ideal except in the early stages. Position errors and attitude errors can be stabilized to zero. When the lookahead distance is small, the LOS guidance law is prone to overshoot. On the contrary, the position errors converge slowly. We can also see that position errors converge faster and have no overshoot when MPC guidance law is applied. Figs 3i-k show the desired approach angles and the MPC stability condition. Obviously, they all satisfy the constraints in Eq. (43). Figures 3l-o show the velocities of AUV and virtual target. The surge relative velocity of AUV and the speed of the virtual target converge to the expected value. At the same time, it can be seen clearly that the heave and sway relative velocities are convergent and very small because there are no driving forces in the lateral or vertical directions. Figures 3p-r are the actual control inputs of AUV. Due to the smooth path, the rudder angles and speed of the propeller are very stable.

5.3 Path Following with Wave Disturbances

The AUV is often disturbed by waves when it travels close to the surface of the water. Next, path following control simulation experiment is performed with wave disturbances. The wave height is set as 1 m. Firstly, the path following control is simulated by using LOS guidance law. Then, according to the control results of LOS guidance law, the boundary layer distance is set as $\Delta_L = 0.5$, and the MPC guidance law is applied to carry out the path following control simulation. Other designed parameters of the controller remain unchanged. The simulation experiment results are displayed in Fig. 4. We can see that both controllers can achieve the path following task, but the tracking accuracy is decreased due to the disturbance of waves. Although all tracking errors fluctuate, the lateral and vertical tracking errors converge within the boundary layer. Due to the fluctuation of lateral and vertical tracking errors, both the attitude errors of AUV and desired approach angles fluctuate frequently when LOS guidance law is adopted. In order to track the desired approach angles, the rudder angles also fluctuate frequently, which causes the fluctuation of lateral and vertical relative velocities. But the amplitudes of the lateral and vertical relative velocities are very small. Since waves mainly affect the absolute velocity, but not the relative velocity. Therefore, the velocity of the virtual target has some fluctuations.

However, the surge relative velocity of AUV can converge to the expected value smoothly and the speed of the thrusters is also very stable. When MPC guidance law is adopted, the stability of attitude errors, desired approach angles, rudder angles, and relative velocities are obviously improved. Compared with LOS guidance law, the mean square error of horizontal rudder and vertical rudder angles are reduced by 49% and 46%, and the integral of

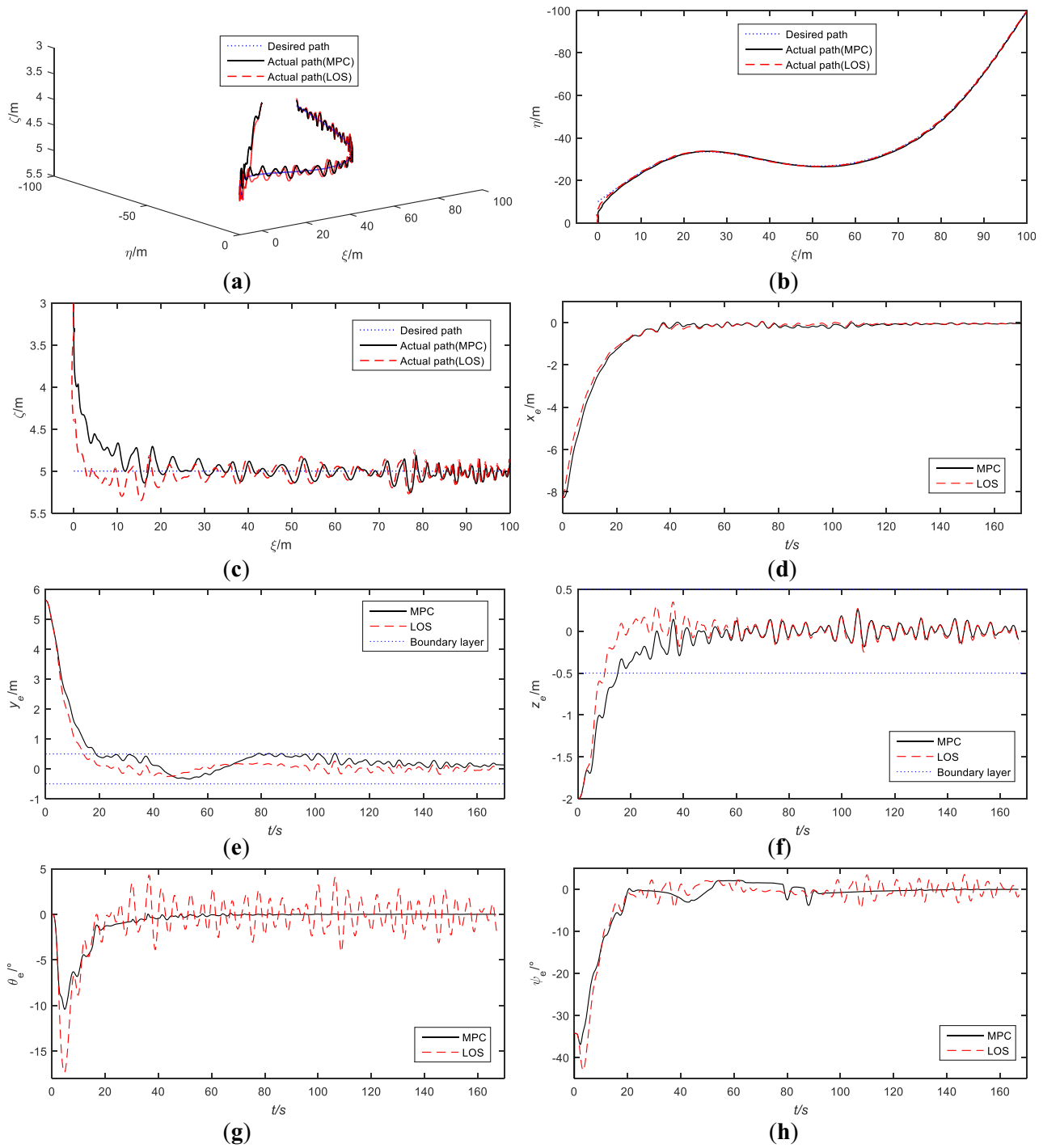

Fig. 4 Simulation results of path following with wave disturbances. (**a**) The simulation results are displayed in three-dimensional space. (**b**) The simulation results are displayed in horizontal plane. (**c**) The simulation results are shown in vertical plane. (**d**) The along-track error. (**e**) The lateral error. (**f**) The vertical error. (**g**) The pitch angle error. (**h**) The yaw angle error. (**i**) The desired pitch angle error. (**j**) The desired yaw angle error. (k) The stability condition. (**l**) The surge relative velocity of AUV. (**m**) The speed of the virtual target. (**n**) The sway relative velocity of AUV. (**o**) The heave relative velocity of AUV. (**p**) The speed of the propeller. (**q**) The horizontal rudder angle. (**r**) The vertical rudder angle

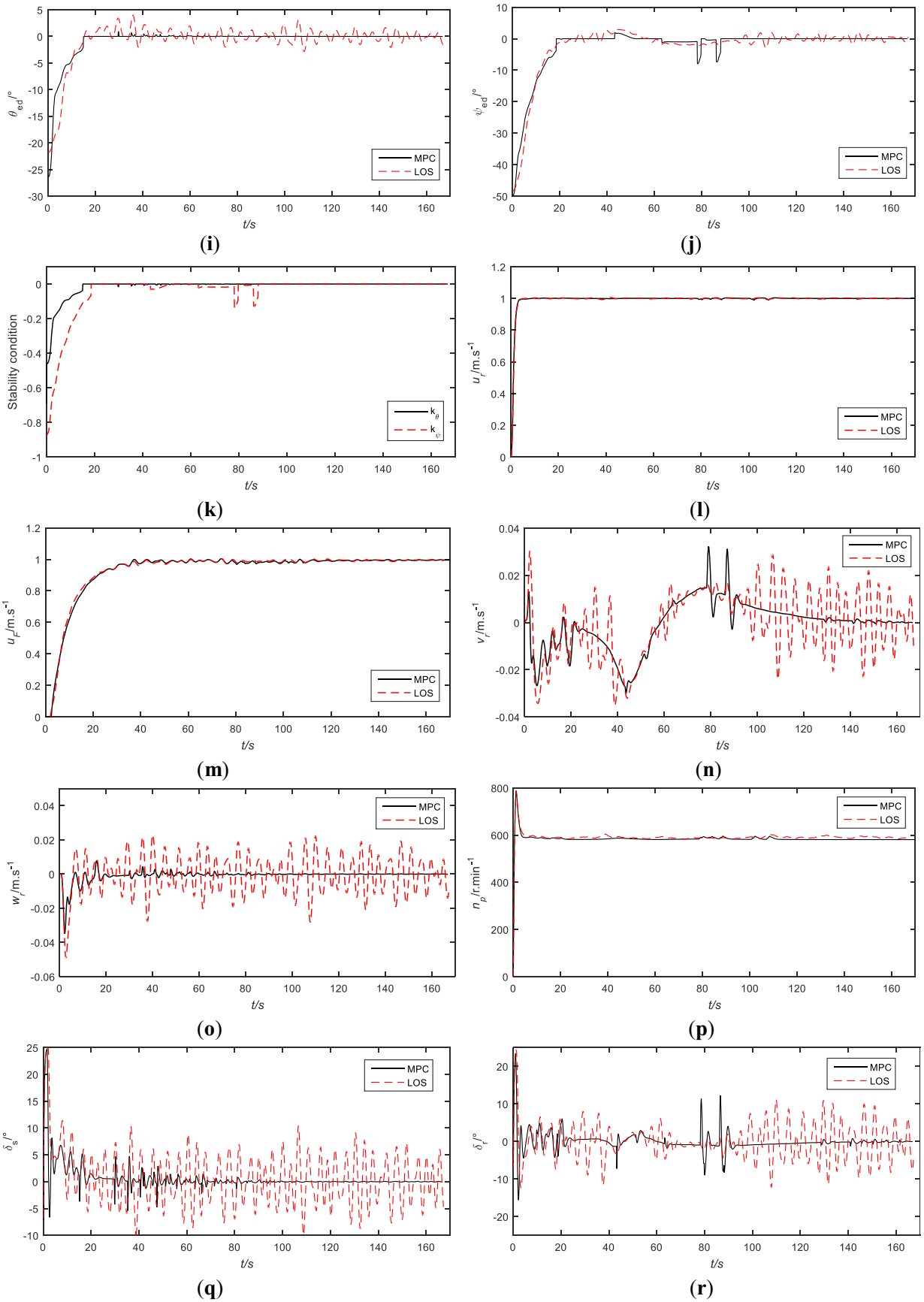

Fig. 4 (continued)

square of horizontal rudder and vertical rudder angles are reduced by 73% and 71%, respectively.

6 Conclusion

The path planning and following control problems of underactuated AUV are studied in this paper. To plan the optimal global path when the underwater environment model is known, the IPSO-SP algorithm is proposed based on PSO algorithm and cubic spline interpolation technology, which can be used to plan smooth paths that meet obstacle avoidance requirements and curvature constraints. The path following controller, which combines MPC and ADSMC, is designed to improve system performance. First, the virtual target and MPC guidance law are developed in the kinematic controller. The virtual target is singularity-free for all regular paths. The MPC guidance law provides the optimal approach angles for the dynamic controller. The weights of lateral and vertical errors in the MPC guidance law are adjusted based on the boundary layer to reduce the wave disturbances. When the tracking errors are within the boundary layer, the weight coefficients Q_{11} and Q_{22} will take the minimum value, so as to reduce the control requirements for position errors and increase the control requirements for attitude errors. In this way, AUV can go forward as far as possible with the wave, avoid blindly overcoming the disturbance of waves, thus making the rudder angles more stable and reducing the energy consumption of the actuator. In order to reduce the computational burden, the linear time-varying MPC is applied. The simulation is implemented in Matlab and the eq. (43) is solved by QP algorithm on a PC (CPU: Intel i5-3230M, 2.6GHz; RAM: 4GB). The average time used to calculate MPC guidance law is within 10 ms. So the controller can meet the real-time requirement of path following. Because the kinematics level is not affected by the model uncertainties, the prediction effect can be guaranteed. Then, the dynamic controller is designed based on ADSMC which can improve the robustness of the system. The dynamic controller is implemented by using relative velocity, thus it is better for energy saving. Furthermore, relative velocity is easier to be measured and relative velocity sensors can provide more accurate results than absolute velocity measurements. This approach can also be extended to path planning and following of other unmanned vehicles.

Funding This work has been supported by the National Natural Science Foundation of China under Grant no. 51279039.

Compliance with Ethical Standards

Conflict of Interest The authors declare no conflict of interest.

References

1. Fredriksen, E., Pettersen, K.Y.: Global κ-exponential way-point maneuvering of ships: theory and experiments. Automatica. **42**(4), 677–687 (2006)
2. Elmokadem, T., Zribi, M., Youcef-Toumi, K.: Control for Dynamic Positioning and Way-Point Tracking of Underactuated Autonomous Underwater Vehicles Using Sliding Mode Control. Journal of Intelligent and Robotic Systems (2018)
3. Ravankar, A., Ravankar, A.A., Kobayashi, Y., Hoshino, Y., Peng, C.C.: Path smoothing techniques in robot navigation: state-of-the-art. Current and Future Challenges. Sensors. **18**(9), 1–30 (2018)
4. Yang, K., Sukkarieh, S.: 3D smooth path planning for a UAV in cluttered natural environments. IEEE/RSJ International Conference on Intelligent Robots and Systems, 794–800 (2008)
5. Filaretov, V., Yukhimets, D.: The method of formation of AUV smooth trajectory in unknown environment. OCEANS'2016, Shanghai, China, 1–8 (2016)
6. Shen, C., Shi, Y., Buckham, B.: Integrated path planning and tracking control of an AUV: a unified receding horizon optimization approach. IEEE/ASME Transactions on Mechatronics. **22**(3), 1163–1173 (2017)
7. Liu, L., Wang, D., Peng, Z.H.: Path following of marine surface vehicles with dynamical uncertainty and time-varying ocean disturbances. Neurocomputing. **173**, 799–808 (2016)
8. Lekkas, A.M., Fossen, T.I.: Integral LOS path following for curved paths based on a monotone cubic Hermite spline Parametrization. IEEE Trans. Control Syst. Technol. **22**(6), 2287–2301 (2014)
9. Chen, S., Liu, C.W., Huang, Z.P., Cai, G.S.: Global path planning for AUV based on sparse a* search algorithm. Torpedo Technology. **20**(4), 271–275 (2012)
10. Hwang, Y., Ahuja, N.: A potential field approach to path planning. IEEE Trans. Robot. Autom. **8**(1), 23–32 (1992)
11. Liu, J.H., Yang, J.G., Liu, H.P., Tian, X.J., Gao, M.: An improved ant colony algorithm for robot path planning. Soft. Comput. **21**(19), 5829–5839 (2017)
12. Satoshi, T., Makoto, T., Toshiharu, H., Katsuji, U.: Path planning for mobile robot using a genetic algorithm. Stochastic Systems Theory and its Applications. **1999**, 243–247 (2018)
13. Yao, X.L., Wang, F., Wang, J.F., Wang, X.W.: Bilevel Optimization-Based Time-Optimal Path Planning for AUVs. Sensors. **18**(12), (2018)
14. Fossen, T. I., Breivik, M., Skjetne, R.: Line-of-sight path following of underactuated marine craft. 6th IFAC Conference on Manoeuvring and Control of Marine Craft (MCMC), Girona, Spain, 244–249 (2003)
15. Lekkas, A. M., Fossen, T. I.: Minimization of cross-track and along-track errors for path tracking of marine underactuated vehicles. European Control Conference (ECC), 3004–3010 (2014)
16. Lekkas, A.M., Fossen, T.I.: A time-varying lookahead distance guidance law for path following. IFAC Proceedings Volumes. **45**(27), 398–403 (2012)
17. Mu, D.D., Wang, G.F., Fan, Y.S., Sun, X.J., Qiu, B.B.: Adaptive LOS path following for a podded propulsion unmanned surface vehicle with uncertainty of model and actuator saturation. APPLIED SCIENCES-BASEL. **7**(12), 1–20 (2017)
18. Caharija, W., Pettersen, K.Y., Calado, P., Braga, J.: A comparison between the ILOS guidance and the vector field guidance. IFAC-Papers OnLine. **48**(16), 89–94 (2015)
19. Xu, H., Soares, C.G.: Vector field path following for surface marine vessel and parameter identification based on LS-SVM. Ocean Eng. **113**, 151–161 (2016)
20. Fossen, T.I., Pettersen, K.Y., Galeazzi, R.: Line-of-sight path following for dubins paths with adaptive sideslip compensation of drift forces. IEEE Trans. Control Syst. Technol. **23**(2), 820–827 (2015)

21. Calvo, O., Rozenfeld, A., Souza, A., Valenciaga, F., Puleston, P.F., Acosta, G.: Experimental results on smooth path tracking with application to pipe surveying on inexpensive AUV. IEEE/RSJ International Conference on Intelligent Robots and Systems, 3647–3653 (2008)
22. Moe, S., Caharija, W., Pettersen, K. Y., Schjolberg, I.: Path following of underactuated marine surface vessels in the presence of unknown ocean currents. American Control Conference (ACC), 3856–3861 (2014)
23. Caharija, W., Pettersen, K. Y., Gravdahl, J. T., Borhaug, E.: Path following of underactuated autonomous underwater vehicles in the presence of ocean currents. IEEE 51st IEEE Conference on Decision and Control (CDC), 528–535 (2012)
24. Caharija, W., Pettersen, K.Y., Sørensen, A.J., Candeloro, M., Gravdahl, J.T.: Relative velocity control and integral line of sight for path following of autonomous surface vessels: merging intuition with theory. Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment. **228**(2), 180–191 (2014)
25. Pettersen, K.Y., Nijmeijer, H.: Underactuated ship tracking control: theory and experiments. Int. J. Control. **74**(14), 1435–1446 (2001)
26. Repoulias, F., Papadopoulos, E.: Planar trajectory planning and tracking control design for underactuated AUVs. Ocean Eng. **34**(11), 1650–1667 (2006)
27. Jiang, Z.P.: Global tracking control of underactuated ships by Lyapunov's direct method. Automatica. **38**(2), 301–309 (2002)
28. Silvestre, C., Pascoal, A., Kaminer, I.: On the design of gain-scheduled trajectory tracking controllers. International Journal of Robust and Nonlinear Control. **12**(9), 797–839 (2002)
29. Fossen, T.I., Lekkas, A.M.: Direct and indirect adaptive integral line-of-sight path-following controllers for marine craft exposed to ocean currents. International Journal of Adaptive Control and Signal Processing. **31**(4), 445–463 (2017)
30. Antonelli, G., Caccavale, F., Chiaverini, S., Fusco, G.: A novel adaptive control law for underwater vehicles. IEEE Trans. Control Syst. Technol. **11**(2), 221–232 (2003)
31. Li, J.H., Lee, P.M.: Design of an adaptive nonlinear controller for depth control of an autonomous underwater vehicle. Ocean Eng. **32**(17–18), 2165–2181 (2005)
32. Li, J.H., Lee, P.M., Jun, B.H., Lim, Y.K.: Point-to-point navigation of underactuated ships. Automatica. **44**(12), 3201–3205 (2008)
33. Do, K.D., Pan, J.: State-and output-feedback robust path- following controllers for underactuated ships using Serret–Frenet frame. Ocean Eng. **31**(5–6), 587–613 (2004)
34. Do, K.D., Pan, J.: Global robust adaptive path following of underactuated ships. Automatica. **42**(10), 1713–1722 (2006)
35. Lapierre, L., Jouvencel, B.: Robust nonlinear path-following control of an AUV. IEEE J. Ocean. Eng. **33**(2), 89–102 (2008)
36. Liao, Y.L., Wan, L., Zhang, J.Y.: Backstepping dynamical sliding mode control method for the path following of the underactuated surface vessel. Procedia Engineering. **15**, 256–263 (2011)
37. Ji, D., Liu, J., Zhao, H., Wang, Y.: Path following of autonomous vehicle in 2D space using multivariable sliding mode control. Journal of Robotics. **2014**, 1–6 (2014)
38. Xu, J., Wang, M., Qiao, L.: Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles. Ocean Eng. **105**, 54–63 (2015)
39. Liu, C., Zou, Z., Yin, J.: Trajectory tracking of underactuated surface vessels based on neural network and hierarchical sliding mode. J. Mar. Sci. Technol. **20**(2), 322–330 (2015)
40. Chen, W., Wei, Y. H., Zeng, J. H., Han, H., Jia, X.: Adaptive terminal sliding mode NDO-Based control of underactuated AUV in vertical plane. Discrete Dynamics in Nature and Society, 1–9 (2016)
41. Ishaque, K., Abdullah, S.S., Ayob, S.M., Salam, Z.: Single input fuzzy logic controller for unmanned underwater vehicle. J. Intell. Robot. Syst. **59**(1), 87–100 (2010)
42. Zhou, J.J., Tang, Z.D., Zhang, H.H., Jiao, J.F.: Spatial path following for AUVs using adaptive neural network controllers. Math. Probl. Eng. **2013**, 1–9 (2013)
43. Yao, X. L., Yang, G. Y., Peng, Y.: Nonlinear Reduced-Order Observer-Based Predictive Control for Diving of an Autonomous Underwater Vehicle. Discrete Dynamics in Nature and Society, 1–15 (2017)
44. Ismail, Z.H., Dunnigan, M.W.: A region boundary-based control scheme for an autonomous underwater vehicle. Ocean Eng. **38**(17–18), 2270–2280 (2011)
45. Mayne, D.Q., Rawlings, J.B., Rao, C.V., Scokaert, P.O.M.: Constrained model predictive control: stability and optimality. Automatica. **36**(6), 789–814 (2000)
46. Falcone, P., Borrelli, F., Tseng, H.E., Asgari, J., Hrovat, D.: Linear time-varying model predictive control and its application to active steering systems: stability analysis and experimental validation. International Journal of Robust and Nonlinear Control. **18**(8), 862–875 (2008)
47. Prestero, T.: Development of a six-degree of freedom simulation model for the REMUS autonomous underwater vehicle. Oceans. **1**, 450–455 (2002)

Xiaowei Wang received the M.S. degree in College of Automation from the Harbin Engineering University, Harbin, Heilongjiang, China, in 2009. Currently, he is a Lecturer in the College of Mechanical engineering, Jiujiang Vocational and Technical College, Jiujiang, Jiangxi, China. He is pursuing the Ph.D. degree in Control science and engineering at Harbin Engineering University, College of Automation, Harbin, Heilongjiang, China. His research interests include advanced control theory and control of AUV.

Xuliang Yao received the Ph.D. degree in College of Automation from the Harbin Engineering University, Harbin, Heilongjiang, China, in 2005. Currently, he is a Professor in the College of Automation, Harbin Engineering University, Harbin, Heilongjiang, China. His research interests include model predictive control, optimization and optimal control, modeling and control of ocean robotics, and nonlinear system control.

Le Zhang received the B.A. degree in School of Humanities from Xidian University, Xi'an, Shanxi, China, in 2006, and M. Eng. degree in Hefei University of Technology, Hefei, Anhui, China, in 2013. Currently, he is a Lecturer in the Center for International Communication and Cooperation, Jiujiang Vocational and Technical College, Jiujiang, Jiangxi, China. His research interests include British and American Literature, and Logistics Engineering.