Indian Journal of Geo Marine Sciences
Vol. 48 (07), July 2019, pp. 1104-1112

Underwater dual manipulators-Part II: Kinematics analysis and numerical simulation

Xianbo Xiang[1,2], Shuaiqi Gan[1], Yuhao Liu[1], Mingjie Bi[1], & Qin Zhang[3*]

[1]School of Naval Architecture and Ocean Engineering, Huazhong University of Science and Technology, Wuhan, China

[2]Shenzhen Huazhong University of Science and Technology Research Institute, Shenzhen 518057, China

[3]State Key Lab of Digital Manufacturing, Equipment and Technology, Huazhong University of Science and Technology, Wuhan 430074, China

*[E-mail: qin.zhang@hust.edu.cn]

This paper introduces dual-arm underwater manipulators mounted on an autonomous underwater vehicle (AUV), which can accomplish the underwater handling task. Firstly, the mechanical structure of the dual-arm system is briefly introduced, wherein each 4-DOF manipulator has an additional grasping function. In addition, the kinematics model of the manipulator is derived by using the improved D-H method. Secondly, the working space of the underwater dual-arm system is analyzed, which is obtained by using Monte Carlo method. The cubic polynomial interpolation and the five polynomial interpolation trajectory planning methods are compared in the joint space. Finally, with the help of the Robotics Toolbox software, the numerical test is conducted to verify the functions of the underwater dual-arm manipulator system.

[Keywords: Underwater manipulator; Dual-arm; Trajectory planning; Monte Carlo method**]**

Introduction

Underwater environment is highly complex, dynamic and uncertain[1,2,3,4,5], yet the exploration and exploitation of underwater resources can be achieved by resorting to intelligent underwater robotic technology rather than human operators. Underwater vehicle and manipulator can be applied to underwater missions. For instance, in the marine exploration, it can extract mineral in the depths of several kilometers, capture the biological resources of the seabed to learn the diversity of seabed organisms, and test the submarine environment. In terms of underwater equipment maintenance, due to low visibility, strong pressure, limited working space and low flexibility in deep water operations, the underwater manipulator is able to carry out these tasks in the constrained underwater environment. Research work on modeling, control and simulations of intelligent control systems, including advanced ship[6], manipulator[7], aerial robot[8,9] and marine robot[10,11,12,13,14], is helpful to enhance the intelligent operational capability of the underwater robotic system.

Robotic manipulator has attracted attentions over the past few decades[15,16]. At present, there are various kinds of industrial manipulators on the land, while the underwater ones are mainly single-armed. With the increasing variety and difficulty of underwater operations, single-armed underwater manipulator is hard to meet engineering needs, such as heavy objects salvage, and target capture. The complexity of dual-arm manipulator and even multi-armed manipulator presents more challenges.

Chang et al.[17] used Lie group to build a dynamic model of underwater vehicle manipulator system (UVMS). The AMDUS (Advanced Manipulation for Deep Underwater Sampling) program of the European Union has developed an underwater hand operating system, whose joints are driven by motors[18]. The dual-manipulator system is designed with high motion precision but small snatch force. The 'HAIMA' ROV is equipped with a hydraulic dual-manipulator operating system with five joints[19].

The object of this research is aimed at the underwater dual-arm manipulator. In this paper, the D-H coordinate system is established based on the software AutoCAD. And the dual-arm manipulator model is built based on the Robotics Toolbox in Matlab. To solve the kinematics equation of the system, the Monte Carlo method is used to obtain the reachable workspace of the dual-armed manipulator by random sampling. In addition, the comparative trajectory planning is performed, based on the cubic polynomial interpolation and the fifth-order polynomial interpolation. Finally, the point-to-point

trajectory planning is achieved for the motion of the arms through the Robotics Toolbox.

Materials and Methods

Manipulator design

The developed 4-DOF dual-arm manipulator possesses an additional grasping function. The manipulator, designed according to the principle of zero buoyancy, consists of shoulder, upper arm, lower arm, wrist joints, and a claw-shaped end effector. The 3-D model of the manipulator is shown in Figure1. The length of the manipulator is 1.28 m and the mass is 12.5 kg. The technical parameters of the joints are listed in Table 1.

Manipulator model

In this paper, each arm of the dual-arm manipulator system is 4-DOF and each joint is a rotating pair, so it is a type of 2×4R manipulator. The manipulator modeling is actually a mapping problem from joint space to Cartesian coordinate system. Therefore, the coordinate system of each link is established according to the lower joint method in the D-H parameter method, as shown in Figure 2.

$O(x_0, y_0, z_0)$ is the base coordinate system of the two arms. For the convenience of modeling, the earth-fixed coordinate system $O(x_{00}, y_{00}, z_{00})$ is established, wherein O_{00} is the origin of the earth-fixed coordinate system, which is equivalent to an additional link. In addition, a joint is added to indicate the claw function. Both joints cannot be rotated, so there are 6 joints on each side.

The model of the manipulator is simulated based on Robotics Toolbox. The Link and SerialLink classes in the toolbox are used to build a dual-arm manipulator model. The specific call format is as follows:

$$L = \text{Link}([theta\ d\ a\ alpha\ sigma], \text{convention})$$

where $theta$ is the rotation angle θ; d is the offset; a is the link length; $alpha$ is the torsion angle α; $sigma$ may take 0 and 1 with default 0:0 indicates that the rotation is deputy, 1 indicates the mobile pair; $convention$ usually takes $standard$ and $modified$, the former represents the standard D-H model, and the latter represents the improved D-H model[20, 21, 22]. In this paper, the improved D-H model is used.

$$\text{robot} = \text{Serial Link}(\text{L}, 'name', 'manipulator')$$

It means that an object named $robot$ is created based on the D-H parameter table, named *manipulator*. The Serial Link function is used to connect the various links to the corresponding joints.

As shown in Figure 3, the dual-arm manipulator model is drawn in Matlab. At the beginning, the *teach (robot)* function is used to draw the dual-arm manipulator model, as shown in Figure 3(a). This function can not only display the model image, but also teach manually. There is a GUI command box on the left side of the figure, and the upper and lower command boxes control the left and right arms respectively. (x, y, z) represent the position of the origin of the

Table 1 — Parameters of the manipulator

Joint	Length (m)	Diameter (m)	Thickness (m)	Rotation range (°)
Shoulder	0	0.12	/	-130~0
Upper-arm	0.352	0.11	0.01	-55 ~55
Lower-arm	0.22	0.10	0.01	-55 ~55
Wrist	0.3	0.09	0.01	-45 ~45

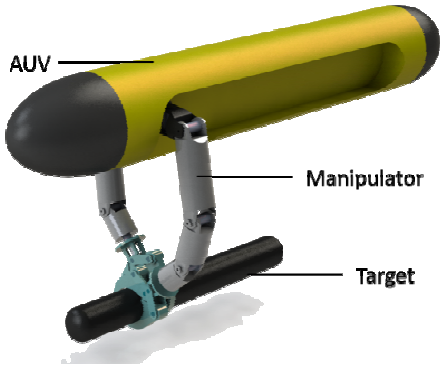


Fig. 1-3 —D model of the dual-arm underwater manipulator system

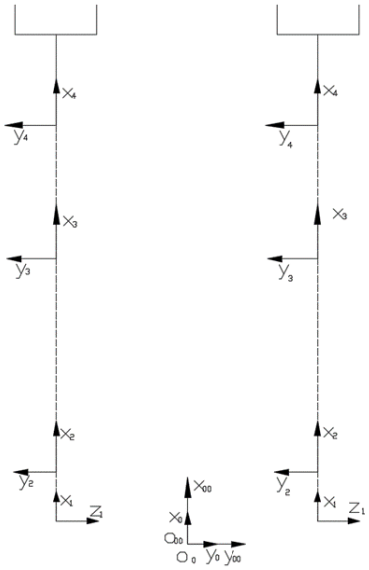


Fig. 2 — D-H coordinate system

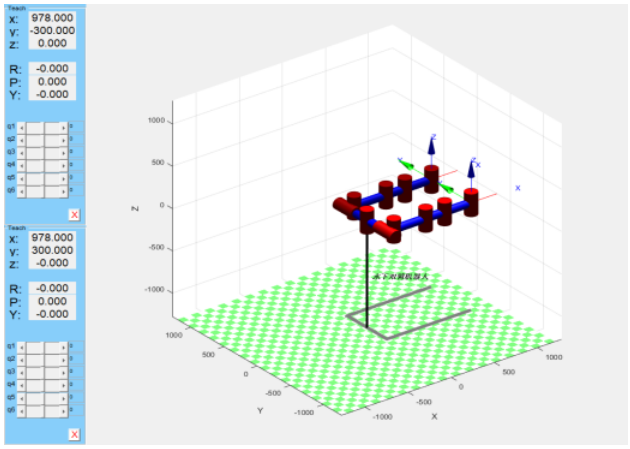


Fig. 3 — (a) Initial state of dual-arm manipulator

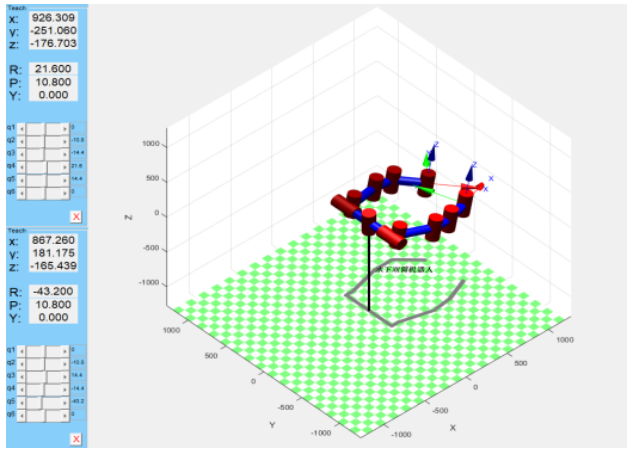


Fig. 3 — (b) Arbitrary state of dual-arm manipulator

coordinate system of the claw link. (R, P, Y) represent the state of the claw link, where *R, P* and *Y* represent the yaw, pitch and roll angles respectively, and $q_1 \sim q_6$ represent the angle θ of each joint. By sliding the scroll bar of each joint angle in the command box, the size of the joint value can be changed. Also, the joint value can be directly input to the box on the right side of the scroll bar. The two driving modes are shown in Figure 3(b), reflecting the change of the orientation of the arm and position of the claw.

Kinematic equations

The kinematic equation of the system is established based on the D-H method by multiplying the transformation matrices between the respective links sequentially. The result is the transformation matrix of the manipulator, which is also the final homogeneous transformation matrix.

According to the D-H parameters, each transformation matrix can be listed. For the left arm, there are five transformation matrices:

${}^0_0T$、${}^0_1T$、${}^1_2T$、${}^2_3T$、${}^3_4T$、${}^4_5T$:

$$ {}^0_1T = \begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0 \\ s\theta_1 & c\theta_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1) $$

$$ {}^1_2T = \begin{bmatrix} c\theta_2 & -s\theta_2 & 0 & 80 \\ 0 & 0 & 1 & 0 \\ -s\theta_2 & -c\theta_2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2) $$

$$ {}^2_3T = \begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & 370 \\ s\theta_3 & c\theta_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3) $$

$$ {}^3_4T = \begin{bmatrix} c\theta_4 & -s\theta_4 & 0 & 220 \\ s\theta_4 & c\theta_4 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4) $$

$$ {}^4_5T = \begin{bmatrix} 1 & 0 & 0 & 308 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (5) $$

Then, these matrices are multiplied to get the final transformation matrix:

$$ {}^0_5T = {}^0_1T\,{}^1_2T\,{}^2_3T\,{}^3_4T\,{}^4_5T $$

$$ = \begin{bmatrix} c_1c_{234} & -c_1s_{234} & -s_1 & c_1(80 + 370c_2 + 308c_{234} + 220c_{23}) \\ s_1c_{234} & -s_1s_{234} & c & s_1(80 + 370c_2 + 308c_{234} + 220c_{23}) \\ -s_{234} & -c_{234} & 0 & -370s_2 - 220s_{23} - 308s_{234} \\ 0 & 0 & 0 & 1 \end{bmatrix} $$

$$ = \begin{bmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (6) $$

where ${}^0_5T$ indicates the pose state of the claw coordinate system relative to the earth-fixed coordinate system; c represents *cos*; s represents *sin*; and c_1, c_{23}, c_{234} represents $cos\,\theta_1$, $cos(\theta_2 + \theta_3)$, $cos(\theta_2 + \theta_3 + \theta 4)$, respectively. By substituting the joint angles into equation (6), the kinematic model of the left arm can be obtained. Similarly, the kinematic model of the right arm can be obtained.

Workspace

Under the premise of the inverse kinematics of the manipulator, the area of the end point is called the

workspace of the robot. There are two types of workspace:

(1) Dexterous workspace is a collection of points that an end effector can reach in any direction or arbitrary orientation.

(2) Reachable workspace is a collection of points that an end-effector can reach in at least one direction or one orientation.

In general, reachable workspace of the manipulator is obtained by Monte Carlo method[23,24], which is based on probability statistics theory. The workspace of the manipulator is certain without randomness, but for the convenience of calculation, a random probability statistical model is constructed artificially to simulate the workspace of the manipulator. The basic principle is that each joint angle of the manipulator has a certain range. And some joint angles are determined by random numbers within the range of each joint angles. The set of random accessible points at the end of the manipulator can be used to form the manipulator's workspace.

A. *Calculation method:* Using the method of Monte Carlo to get the robot workspace mainly includes the following steps in Matlab:

1) The coordinate of the end effector, $[Px, Py, Pz\,]\hat{}T$ can be related to the joint angle, which is calculated via the kinematic model. The coordinate is given as:

$$\begin{cases} \mathrm{p}_x = C_1(80 + 370C_2 + 308C_{234} + 220C_{23}) \\ \mathrm{p}_y = S_1(80 + 370C_2 + 308C_{234} + 220C_{23}) \\ \mathrm{p}_z = -370S_2 - 220S_{23} - 308S_{234} \end{cases} \tag{7}$$

The vector expression of the end effector relative to the base coordinate system can be obtained by equation (7).

2) Generate N random numbers between 0 and 1 by using Rand () function. Then, N random joint angle values of each joint can be obtained by the equation,

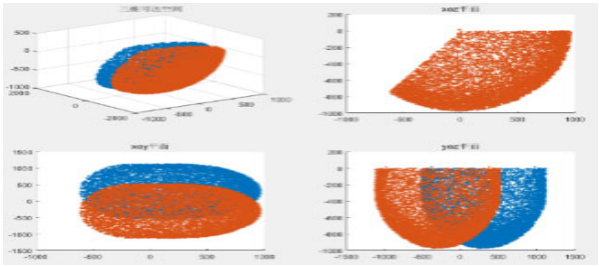
$$q_i = q_{\min i} + (q_{\max i} - q_{\min i})\mathrm{Rand}\ (1, \mathrm{N}) \tag{8}$$

where q_i is the i-th joint angle; and $q_{\min i}$ and $q_{\max i}$ are the minimum and maximum angles respectively.

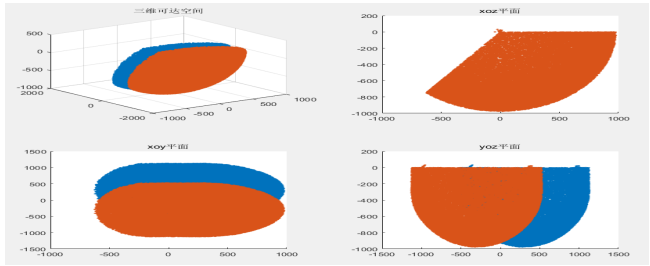
3) Substitute a series of random joint angles obtained by equation (8) into the equation (7) to get random spatial points $[Px, Py, Pz\,]\hat{}T$. Then, generate a series of point cloud images to obtain the workspace of the manipulator.

B. *Simulation analysis of workspace:* In this paper, 10000, 50000 and 100,000 random values are selected, and according to the range of activity of each joint angle given in Table 1. We use the equation (8) to get N random joints. The cloud points grip of workspace is drawn through the three functions of scatter3 (x, y, z), scatter (x, z), scatter (x, y) and scatter (y, z) in Matlab.

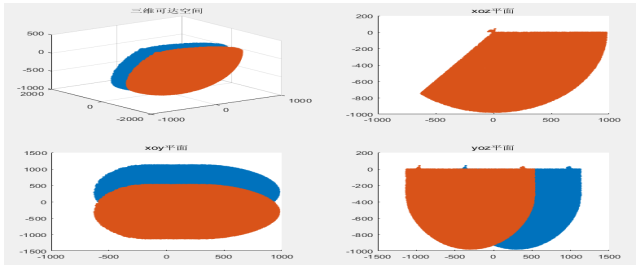
The workspace cloud diagram of the dual arm manipulator is shown in Figure 4. It consists of two workspaces, where the left and right arms are blue and red, respectively. The 3-D cloud diagram of the dual manipulator is shown in Figure 4, while the rest



(a) N=10000



(b) N=50000



(c) N=100000

Fig. 4 —The workspace cloud diagram

are the 2-D cloud diagram representing the projection of the workspace cloud diagram on the three planes x-o-z, x-o-y, and y-o-z.

As the number of random points increase, the workspace become denser and tnhe edge contours are more clearly visible. However, the distribution of random point is not uniform. Although the number of random points is large, some points are still sparse at the boundary. The end of the arm must be able to reach a given point, which is caused by the nonlinear mapping from the joint space to the workspace. In addition, a decrease in the accuracy of extracting workspace boundaries can be caused by this non-linear mapping, which may lead to large deviation.

Path planning

The trajectory can be deemed as the path determined by the displacement, velocity and acceleration of the manipulator at any time in the course of motion[25,25,27]. Motion planning includes path planning and trajectory planning. The path planning refers to the location of the intermediate point in the path from the start point to the end point. The trajectory planning defines the way to reach the point, which is related to time. When the robot is working, the point-to-point (PTP) motion simply describes its starting state and target state, which is mainly used for gripping. For other operating environments, such as curved surface machining and arc welding, not only a starting point, but also some of intermediate points are required. These points are called through points and these motions are called continuous-path or contour motions.

When realizing the trajectory planning of joint space, the motion of each joint is related to time, so it is necessary to use the first and second order time derivative of the motion path to realize the plan of the opponent's claw trajectory. According to the inverse solution of the kinematic equation, the angles of each joint in each path are obtained, then a smooth polynomial function to each joint can be fitted by interpolation, so that it reaches the endpoint from the starting point along the middle point. In this way, all joints simultaneously reach the middle and end points. It is worth noting that each section of the path experiences the same time, but the joint function between each path is not interfered with.

A. *Three-time polynomial interpolation:* There are four unknown coefficients when applied cubic interpolation method:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \tag{9}$$

To solve the unknown constants, four constraints need to be determined at least, two of which correspond to the starting point of the joint angle.

$$\begin{cases} \theta(0) = \theta_0 \\ \theta(t_f) = \theta_f \end{cases} \tag{10}$$

Furthermore, to satisfy the continuity of the joint velocity, there is a constraint on the joint velocity of the start and end points.

$$\begin{cases} \dot{\theta}(0) = 0 \\ \dot{\theta}(t_f) = 0 \end{cases} \tag{11}$$

where θ_0 and $\dot{\theta_0}$ represent the joint angle and angular velocity of the start point at t=0; θ_f and $\dot{\theta_f}$ represent the joint angle and angular velocity of the end point at t=t_f. The speed and acceleration function of time can be obtained by the first-order and second-order derivative from Equation (11):

$$\begin{cases} \dot{\theta}(t) = a_1 + 2a_2 t + 3a_3 t^2 \\ \ddot{\theta}(t) = 2a_2 + 6a_3 t \end{cases} \tag{12}$$

Equation (13) can be obtained by substituting the equations (9) and (12) into the four constraint conditions:

$$\begin{cases} \theta_0 = a_0 \\ \theta_f = a_0 + a_1 t_f + a_2 {t_f}^2 + a_3 {t_f}^3 \\ \dot{\theta_0} = a_1 \\ \dot{\theta_f} = a_1 + 2a_2 t_f + 3a_3 {t_f}^2 \end{cases} \tag{13}$$

Four coefficients are available for solving equations:

$$\begin{cases} a_0 = \theta_0 \\ a_1 = \dot{\theta_0} \\ a_2 = \frac{3}{t_f^2}(\theta_f - \theta_0) - \frac{2}{t_f}\dot{\theta}_0 - \frac{1}{t_f}\dot{\theta}_f \\ a_3 = -\frac{2}{t_f^3}(\theta_f - \theta_0) + \frac{1}{t_f^2}(\dot{\theta_0} - \dot{\theta_f}) \end{cases} \tag{14}$$

The four joints of the left arm can be studied by cubic spline interpolation. The initial joint angular value of q1 is [0 0 0 0] and the final joint angular value of q4 is [-90 10 -30 -35]. As shown in Table 2, two middle point positions 2 and 3 are inserted between the start position 1 and end position 4 as the path points. The start and end position speed are v= [0 0 0 0], and the instantaneous velocity of the intermediate point is taken as the average of the angular velocity of the adjacent two short paths. The

time of reaching position 2, 3, 4 are 10 s, 20 s, 30 s, so there is a time vector t= [0 10 20 30].

In Figure 5, the position and velocity are continuous, the acceleration is discontinuous and has mutation, which has great influence on the motion of the manipulator. This coincides with the cubic polynomial interpolation, as the second-order constraints are not considered in the cubic polynomial interpolation.

Table 2(a) — Joint position

Joint No. I	Position 1	Position 2	Position 3	Position 4
1	0	-10	-75	-90
2	0	25	20	10
3	0	5	-10	-30
4	0	5	-15	-35

Table 2(b) — Angular velocity

Joint No. I	Velocity 1	Velocity 2	Velocity 3	Velocity 4
1	0	-0.123	-0.131	0
2	0	0.033	-0.025	0
3	0	-0.016	-0.057	0
4	0	-0.025	-0.065	0

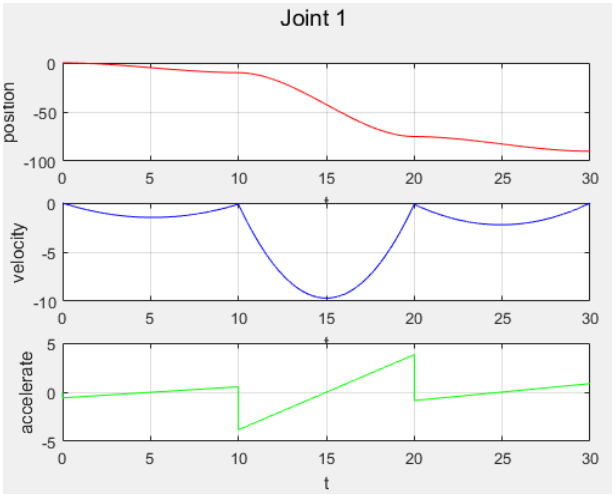


Fig. 5(a) — Joint position, velocity and acceleration

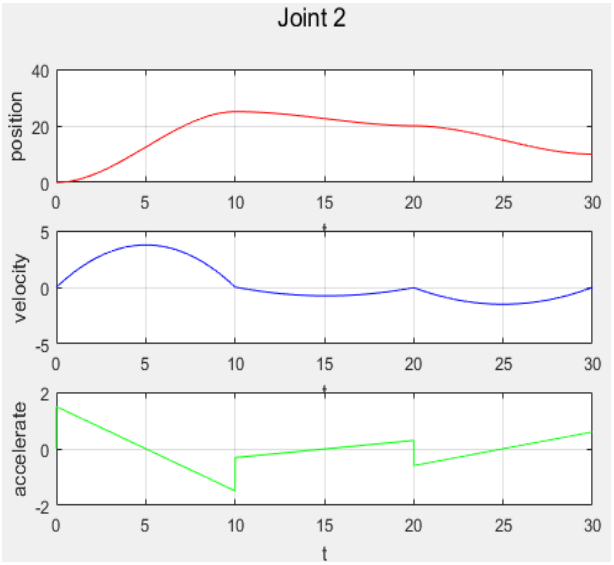


Fig. 5(b) — Joint position, velocity and acceleration

B. *Five-time polynomial interpolation:* More constraints are needed if the trajectory performance is highly required, so there must be more high-order polynomial interpolation calculation. If a path not only requires the position and speed of the start and end points, but also requires the acceleration, at least five-order polynomial is required to interpolate the trajectory. There is:

$$\theta(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 + a_4 t^4 + a_5 t^5 \quad (15)$$

And six constraints need to be met:

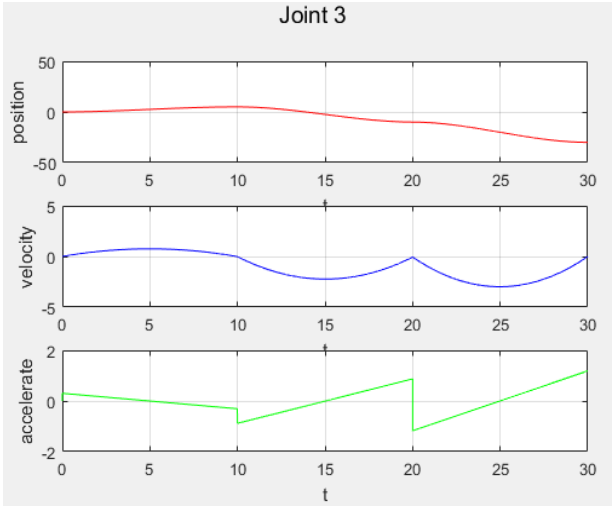


Fig. 5(c) — Joint position, velocity and acceleration

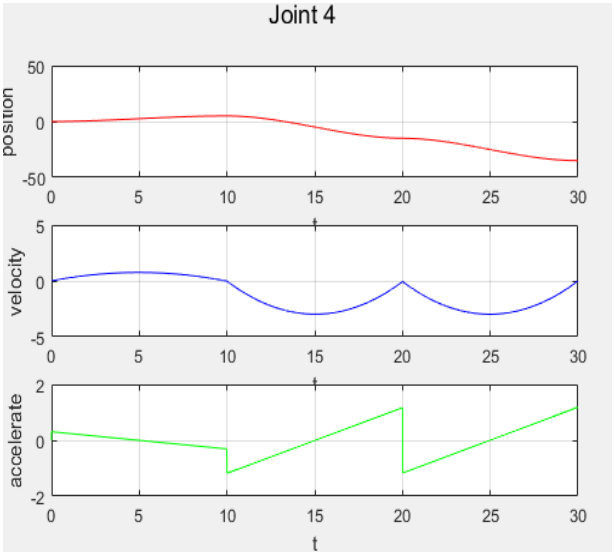


Fig. 5(d) — Joint position, velocity and acceleration

$$\begin{cases} \theta_0 = a_0, \ \theta_f = a_0 + a_1 t_f + a_2 t_f^{\ 2} \\ \quad + a_3 t_f^{\ 3} + a_4 t_f^{\ 4} + a_5 t_f^{\ 5} \\ \dot{\theta}_0 = a_1, \ \dot{\theta}_f = a_1 + 2a_2 t_f \\ \quad + 3a_3 t_f^{\ 2} + 4a_4 t_f^{3} + 5a_5 t_f^{4} \\ \ddot{\theta}_0 = 2a_2, \ddot{\theta}_f = 2a_2 + 6a_3 t_f \\ \quad + 12a_4 t_f^{2} + 20a_5 t_f^{3} \end{cases} \tag{16}$$

There is:

$$\begin{cases} a_0 = \theta_0 \\ a_1 = \dot{\theta}_0 \\ a_2 = \frac{\ddot{\theta}_2}{2} \\ a_3 = \frac{20\theta_f - 20\theta_0 - (8\dot{\theta}_f + 12\dot{\theta}_0)t_f - (3\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^3} \\ a_4 = \frac{30\theta_0 - 30\theta_f + (14\dot{\theta}_f + 16\dot{\theta}_0)t_f + (3\ddot{\theta}_0 - 2\ddot{\theta}_f)t_f^2}{2t_f^4} \\ a_5 = \frac{12\theta_f - 12\theta_0 - (6\dot{\theta}_f + 6\dot{\theta}_0)t_f - (\ddot{\theta}_0 - \ddot{\theta}_f)t_f^2}{2t_f^5} \end{cases} \tag{17}$$

By considering the constraint on the acceleration to the path, the acceleration of the start and the end point are 0. In addition, the instantaneous acceleration of the middle point is the average value of the angular acceleration of the two neighboring paths. The acceleration of each joint is shown in Table 3:

As shown in Figure 6, the joint angular position, velocity and acceleration are continuous.

C. *Comparison:* Based on the previous simulations, the cubic polynomials can ensure that the position and velocity of the joint are continuous. When applied to the manipulator, it may cause a certain impact on the joint motors. On the contrary, due to the second-order constraint in the derivation of the five polynomials, it ensures that the position, velocity, acceleration of each joint angles is continuous, so that the joint motors can be operated smoothly. However, it should be noted that, it is not absolutely accurate that higher-order polynomials are better than lower-order polynomials. In some cases, higher-order interpolation will appear in

Table 3 — The acceleration value of each position

Joint No. I	Acceleration 1	Acceleration 2	Acceleration 3	Acceleration 4
1	0	-0.0377	0.0402	0
2	0	0.01005	0.00755	0
3	0	0.01005	0.0176	0
4	0	0.01255	0.0201	0

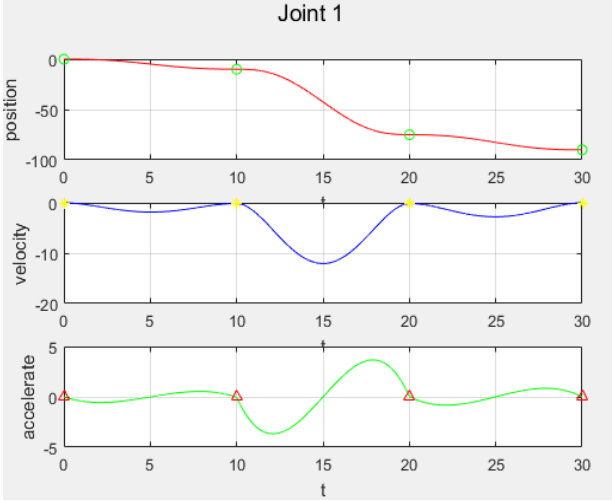


Fig. 6(a) — Joint position, velocity and acceleration

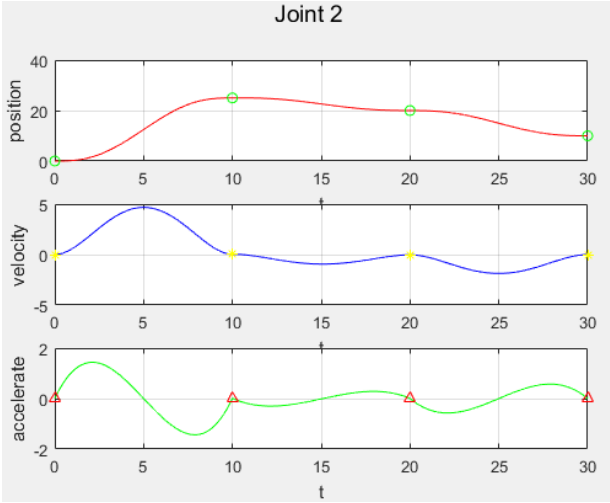


Fig. 6(b) — Joint position, velocity and acceleration

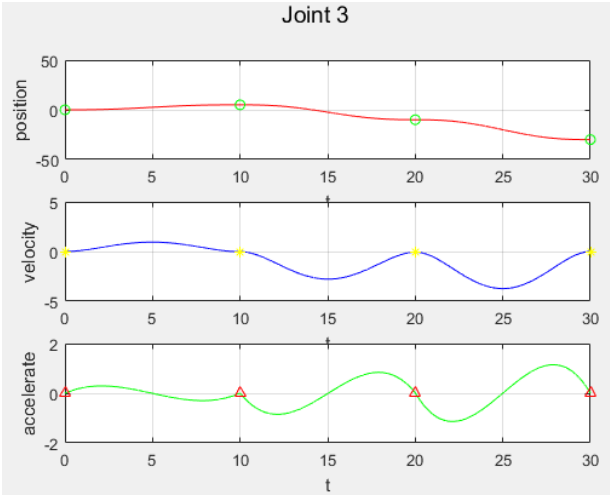


Fig. 6(c) — Joint position, velocity and acceleration

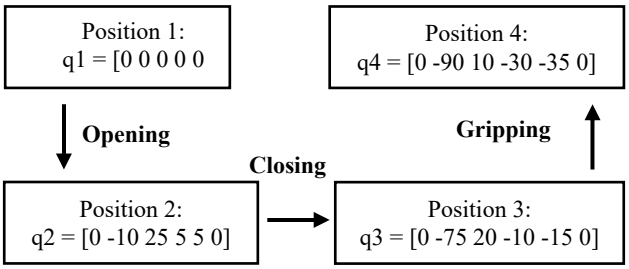


Fig. 7 — Motion steps of dual-arm manipulator

the long lattice phenomenon, that is, the insertion point will appear large fluctuations.

Results and Discussion

The gripping simulation of the dual-arm manipulator is analyzed by using the Robotics Toolbox in Matlab and the following functions of the Toolbox are introduced[28]:

Jtraj function: Trajectory interpolation operation function in joint space returns a series of joint trajectory Q values from joint coordinates q0 to q1. Its invocation format is:

$$[\mathrm{q\ qd\ qdd}] = \mathrm{jtraj}(\mathrm{q0}, \mathrm{q1}, \mathrm{N}) \tag{18}$$

$$[\mathrm{q\ qd\ qdd}] = \mathrm{jtraj}(\mathrm{q0}, \mathrm{q1}, \mathrm{t}) \tag{19}$$

where q is the joint space trajectory (joint angle) sequence; qd and qdd are combined velocity and acceleration sequences respectively; q0 is the initial joint state; q1 is the joint angular state of the terminating point; n is the path number; and t is the time.

Ctraj function: Cartesian space trajectory interpolation operation function returns a series of joint track TC from the attitude T0 to the attitude T1. Its path is generally linear motion. Its invocation format is:

$$\mathrm{TC} = \mathrm{ctraj}(\mathrm{T0}, \mathrm{T1}, \mathrm{m}) \tag{20}$$

$$\mathrm{TC} = \mathrm{ctraj}\ (\mathrm{T0}, \mathrm{T1}, \mathrm{r}) \tag{21}$$

where TC is the Cartesian trajectory; T0 is the initial end pose state; T1 is the position/orientation of the end point, m is the number of points; and r is the given vector length.

According to the model of the dual-arm manipulator, the trajectory planning of dual-manipulator is carried out to simulate the motion process of the actual gripping target. There are three steps: First open arms, and then slowly downward move to the target, and finally grip target. The path

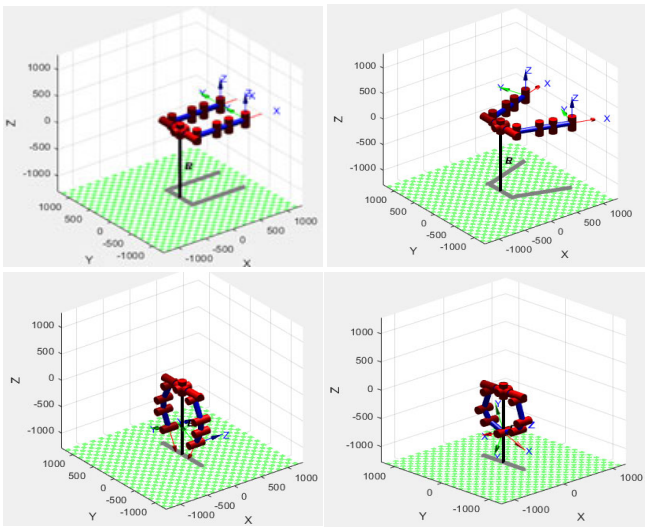


Fig. 8 — Motion process of dual-arm manipulator

planning of the PTP is realized in the joint space, where the motion process can be divided into the following steps:

In Figure 8, the process from (a) to (b) is opening, the process from (b) to (c) is closing, and the process from (c) to (d) is gripping.

Conclusion

In this paper, the kinematic modeling and simulation of an underwater dual-arm manipulator system is presented, which has 4-DOF and five operational functions. The forward kinematic equation of the manipulator is derived based on the improved D-H method. The workspace of the manipulator is analyzed, which is obtained by Monte Carlo method. The trajectory planning in joint space is presented and the similarities and differences between cubic polynomial interpolation and five polynomial interpolation are compared. Based on the Matlab Robotics Toolbox, the numerical test is conducted to verify the functions of the underwater dual-arm manipulator system.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 51579111, in part by the Shenzhen Science and Technology Plan Project under Grant JCYJ20170 4I311305468, in part by the National Key Research and Development Program of China (Grant 2017YFB 1302302) and in part by the Research Fund from Science and Technology on Underwater Vehicle Technology under Grant SXJQR2017KFJJ06.

References

1 Arshad, M.R., Recent advancement in sensor technology for underwater applications, *Indian J. Mar. Sci.*, 38(2009) 267-273.

2 Xiong, C.K., Chen, D.F., Lu, D., Zeng, Z. & Lian, L., Path planning of multiple autonomous marine vehicles for adaptive sampling using Voronoi-based ant colony optimization, *Robot Auton Syst.*, 115(2019) 90-103.

3 Zhang, Q., Zhang, J.L, Chemori, A. &.Xiang, X.B., Virtual submerged floating operational system for robotic manipulation, *Complexity*, (2018) 1-18. DOI: 10.1155/2018/9528313

4 Kang, H.-S., Tang, C.H., Quen, L.K. & Yu, S.X, Parametric resonance avoidance of offshore crane cable in subsea lowering operation through A* heuristic planner, *Indian J. Geo Mar. Sci. 46 (2017) 2422–2433.*

5 Zeng, Z., Sammut, K., Lian, L., Lammas, A. & He, F.P., Rendezvous path planning for multiple autonomous marine vehicles, *IEEE J. Oceanic Eng.*, 43(2018) 640–664

6 Xiang, X.B., Yu, C.Y., Xu, H. and Zhu, Stuart X. Optimization of heterogeneous container loading problem with adaptive genetic algorithm. *Complexity*, 2018: 1–12. DOI: 10.1155/2018/2024184

7 He, W., Ouyang, Y.C. and Hong, J., Vibration control of a flexible robotic manipulator in the presence of input deadzone, *IEEE Transactions on Industrial Informatics*, 13(2017) 48–59.

8 Yu, C., Xiang, X.B., Wilson, P.A., & Zhang, Q., Guidance-error-based robust fuzzy adaptive control for bottom Following of a flight-style AUV with saturated actuator dynamics, *IEEE T Cybernetics*, (2019), DOI: 10.1109/TCYB.2018.2890582

9 Xiang, X.B., Liu, C., Su, H.S. & Zhang, Q., On decentralized adaptive full-order sliding mode control of multiple UAVs, *ISA Trans.*, 71(2017) 196–205.

10 Ye, Z., Hou, P. and Chen, Z., 2D maneuverable robotic fish propelled by multiple ionic polymer-metal composite artificial fins. *Int. Journal of Intelligent Robotics and Applications*, 2(2017) 195–208.

11 Yu, C.Y., Xiang, X.B., Lapierre, L. & Zhang. Q., Robust magnetic tracking of subsea cable by AUV in the presence of sensor noise and ocean currents, *IEEE J. Oceanic Eng.*, 43(2018) 311–322.

12 Phamduy, P., Vazquez, M., Kim, C., Mwaffo, V., Rizzo, A. & Porfiri, M., Design and characterization of a miniature free-swimming robotic fish based on multi-material 3D printing. *Int. Journal of Intelligent Robotics and Applications*, 2(2017) 209-223.

13 Chu, Z.Z., Xiang, X.B., Zhu, D.Q., Luo, C.M., Xie, D.. Adaptive fuzzy sliding mode diving control for autonomous underwater vehicle with input constraint. *Int. J. Fuzzy Syst.*, 20(2018) 1460-1469

14 Yu, C.Y., Xiang, X.B., Zhang, Q. & Xu, G.H., Adaptive fuzzy trajectory tracking control of an under-actuated autonomous underwater vehicle subject to actuator saturation, *Int. J. Fuzzy Syst.*, 20(2018) 269–279.

15 Dwivedy S K, Eberhard P. Dynamic analysis of flexible manipulators, a literature review[J]. *Mechanism & Machine Theory*, 41(2006) 749-777.

16 Piltan F, Sulaiman N, Rashidi M, et al. Design and Implementation of Sliding Mode Algorithm: Applied to Robot Manipulator-A Review. *International Journal of Robotics & Automation*, 2(2011) 265-282.

17 Chang, Z.Y., Chen, B.C. & Yuan, P., Dynamic Modeling of Underwater Vehicle-manipulator System Based on Lie Group, paper presented at the *Seventh International Conference on Design and Manufacturing Science*, (Guangzhou, China), pp.176-179, 2006.

18 Antonelli, G., Chiaverini S., Fuzzy redundancy resolution and motion coordination for underwater vehicle-manipulator systems. *IEEE Trans. on Fuzzy Systems*. 11(2003) 109-120

19 Lian, L., Ma X. & Tao Jun, Research course of 'HAIMA'-4500 ROV system. *Naval Architecture and Ocean Engineering*, 1 (2015) 9-12.

20 Krainin M, Henry P, Ren X, et al. Manipulator and object tracking for in-hand 3D object modeling. *International Journal of Robotics Research*, 30 (2016) 1311-1327.

21 Trujillo J L A, Serrezuela R R, Azhmyakov V, et al. Kinematic model of the scorbot 4PC manipulator implemented in Matlab's Guide. *Contemporary Engineering Sciences*, 11 (2018) 183-199.

22 Pineda-Rico Z. Dynamic Model of a 7-DOF Whole Arm Manipulator and Validation from Experimental Data[C]// International Conference on Informatics in Control, Automation and Robotics. 2015:217-222.

23 Herrero S, Mannheim T, Prause I, et al. Enhancing the useful workspace of a reconfigurable parallel manipulator by grasp point optimization. *Robotics & Computer Integrated Manufacturing*, 31(2015) 51-60

24 Jianxun F U, Feng G. Optimal Design of a 3-Leg 6-DOF Parallel Manipulator for a Specific Workspace[J]. *Chinese Journal of Mechanical Engineering*, 2016, 29(4):659-668.

25 Zou, P.H., Research on Underwater Manipulator ADAMS, *Robot Technology*, 4(2017) 52-53.

26 Zhang Q, Zhao M Y. Minimum time path planning of robotic manipulator in drilling/spot welding tasks[J]. *Journal of Computational Design & Engineering*, 3(2016) 132-139.

27 Yue, P., Zhang, Q.F., An, X.W., Qu, F.J., Structure Design of 7F Underwater Electric Manipulator and Feasibility Analysis, *Machinery Design & Manufacture*, 4 (2014) 114-117.

28 Pan, D.Z., Wang, Q.M., Song, R.H., Yao, X., Gu, Y.H., Kinematics analysis and simulation on Underwater Manipulator, *Shipbuilding of China*,1 (2009) 122-127.