Proceedings of 2021 IEEE
International Conference on Mechatronics and Automation
August 8 - 11, Takamatsu, Japan

Trajectory Tracking Control of an Amphibious Spherical Robot Using MPC Approach

Meng Liu[1], Shuxiang Guo[1,2*], Liwei Shi[1*], Xihuan Hou[1,2], He Yin[1], Ao Li[1], Zan Li[1], Debin Xia[1], Mugen Zhou[1]
1 Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, the Ministry of Industry and Information Technology, School of Life Science, Beijing Institute of Technology
No.5, Zhongguancun South Street, Haidian District, Beijing 100081, China
[2] Faculty of Engineering, Kagawa University, 2217-20 Hayashi-cho, Takamatsu, Kagawa, Japan
E-mails: liumeng@bit.edu.cn; guoshuxiang@bit.edu.cn; shiliwei@bit.edu.cn,
* Corresponding author

***Abstract*—Trajectory tracking control is a basic problem in rescue, detection and obstacle avoidance tasks. The trajectory tracking process has its own constraints, and the core of model predictive control is to solve the quadratic programming problem with constraints, so MPC is used to solve the trajectory tracking problem of amphibious spherical robot in this paper. Firstly, based on the 3-DOF dynamic state space equation of the robot, the model is approximately linearized and discretized to facilitate the design of controller. To solve the difficulty of adjusting the weight matrix of MPC, an adaptive parameter adjustment method based on output error is designed in this paper. The linear trajectory and square trajectory are simulated on MATLAB and Gazebo simulation platform respectively. The simulation results verify the applicability and stability of the designed controller.**

Index Terms—Trajectory tracking, Model predictive control, Amphibious spherical robot, Adaptive parameter adjustment.

I. INTRODUCTION

Underwater robot is an indispensable and important equipment in the fields of Marine scientific research, Marine resources investigation, Marine engineering, underwater archaeological search and rescue, and Marine rights and interests protection. In recent years, there are more and more autonomous underwater vehicles(AUVS) [1] Autonomous Surface Vehicles (USVS) [2], and, Remotely operated Vehicles (ROVS) [3] are operational. Various constraints, such as control force and environmental boundary, are needed to be considered when the underwater vehicle moves in the real environment.

A lot of work is devoted to trajectory tracking. When the mathematical model is changed, the language rules of fuzzy control are relatively independent, so it is widely applied to the tracking control of AUV. However, the establishment of fuzzy control rules depends on expert knowledge and experience. PID is a control algorithm widely used in various fields. PID control is a widely used control algorithm, which was used for the trajectory tracking control of underwater robot[4]. The inner closed loop controller of double loop PID [5] can reduce the change of fluid dynamics and reject disturbance quickly, but the result of PID control is over dependent on parameters. PID controller has a good control effect on single input and single output system, but it is difficult to meet the demand for multi-input and multi-output system. To overcome the insensitivity of dynamic parameter uncertainty, sliding mode control [6]-[9] is a common method for underwater vehicle tracking control. It is difficult to slide strictly along the sliding mode to approach the equilibrium point when the state of the trajectory of robot reaches the sliding mode surface. The state traverse back and forth on both sides of the sliding mode to approach the equilibrium point, thus generating chattering. The nonlinear H-infinity controller [10],[11] achieves the desired trajectory tracking by weakening the internal and external disturbances to ensure the internal stability and robustness. However, the controller is complex and has a large amount of computation, so it is very difficult to implement. Compared with optimal control, suboptimal control reduces the amount of calculation, but the solution is not optimal. Neural network control is another control algorithm widely applied to the trajectory tracking control of underwater vehicles. It is difficult to obtain training data under underwater conditions, so it is necessary to calculate the learning burden [12], [13]. Backstepping is often used in combination with other control algorithms.[14]. The traditional backstepping algorithm has the problem of control signal jumping, which is not considered in most work. Backstepping mainly focuses on the design of the controller, but lacks in the problem of controllable or uncontrollable.

Most of the above research only implement trajectory tracking in simulation, and do not pay enough attention to some constraints in real environment. As an ideal constrained control algorithm, model predictive control (MPC) can optimize the objective function online through the input-output predictive model in the limited sampling time in the future.

Inspired by the above motion control schemes and MPC [15], this paper aims to design a trajectory tracking control scheme by using MPC considering the constraint of control increment and design a new weight matrix adjustment method.

The rest of this paper is organized as follows. Section II introduces the modeling of the amphibious spherical robot and the problem formulation. In Section III, the trajectory tracking control methodology is detailed. Simulation and results are presented in Section IV. Finally, the conclusion is summarized in Section V.

II. MODELING OF THE AMPHIBIOUS SPHERICAL ROBOT AND PROBLEM FORMULATION

A. Design of the Amphibious Spherical Robot

A new amphibious spherical robot was designed based on previous researches[16]-[19]. The robot is mainly composed of three parts: battery cabin, upper hemisphere and driving mechanism. The robot structure is shown in Fig.1. Water storage tanks located in the upper hemisphere avoid extra weight. The control board is located in a watertight enclosure. The driving mechanism is composed of long ducted water jet motors and steering gears. This design can improve the adaptability of the

978-1-6654-4098-1/21/$31.00 ©2021 IEEE

robot in the land and underwater environment [20], [21]. There is a removable battery under the capsule which designed to power the robot. The robot uses binocular camera as its main sensing sensor, and the inertial measurement unit (IMU), GPS positioning module and depth sensor as its attitude detection sensor. The communication module is the underwater acoustic communication.

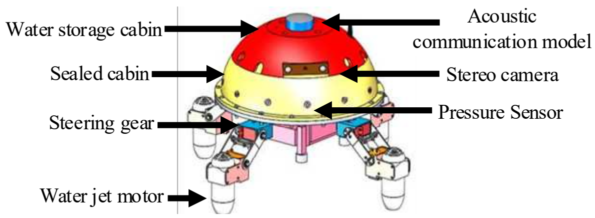


Fig. 1. The prototype of the amphibious spherical robot.

B. Kinematics model and Dynamics model of the ASR

In order to describe the motion state of the robot, we set up two coordinates, which are the global coordinate system (g-frame) and the robot coordinate system (r-frame). The relationship between the two coordinate systems is shown in Fig.2. The motion state $\mathbf{v}$ and pose state $\boldsymbol{\eta}$ are defined as follows:

$$\mathbf{v} = [u, v, w, p, q, r] \quad (1)$$

$$\boldsymbol{\eta} = [x, y, z, \varphi, \psi, \theta] \quad (2)$$

In equation(1), u, v, w are the linear velocities and p, q, r are the angular velocities. In equation(2), x, y, z are positions and φ, ψ, θ are orientations.

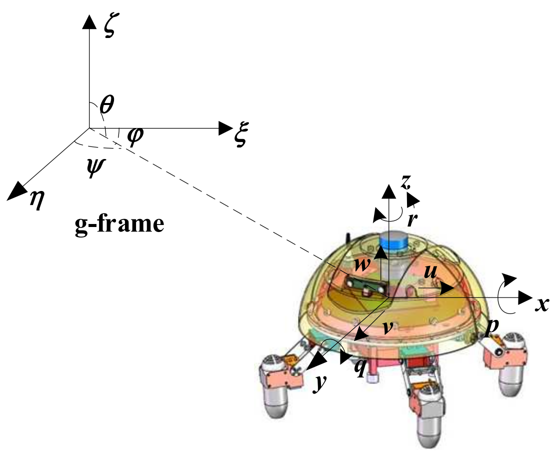


Fig. 2. Global coordinate system and robot coordinate system

This paper mainly considers the underwater trajectory tracking problem in two - dimensional plane. Traditionally, underwater vehicles have had to consider three degrees of freedom: yaw, swing, and swing. However, due to the H-shaped distribution of ASR[23] thrusters and low robot speed, the sway was ignored. The simplified motion equation of robot is:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{R}(\theta)\mathbf{v} \quad (3)$$

where $\boldsymbol{\eta} = [x, y, \theta]^{\mathrm{T}}$ represent the two dimensional position coordinates and orientation in the g-frame; $\mathbf{v} = [u, r]^{\mathrm{T}}$ represent the linear velocity in x direction and Yaw angle velocity in the r-frame. $\boldsymbol{R}(\theta)$ is the rotation matrix calculated by the heading angle θ.

$$\boldsymbol{R}(\theta) = \begin{bmatrix} \cos(\theta) & 0 \\ \sin(\theta) & 0 \\ 0 & 1 \end{bmatrix} \quad (4)$$

According to Newton [24], [25], the dynamic equations are established as follows:

$$\left(\boldsymbol{M}_{RB} + \boldsymbol{M}_{A}\right)\dot{\mathbf{v}} + \boldsymbol{C}(\mathbf{v})\mathbf{v} + \left(\boldsymbol{D}_{l} + \boldsymbol{D}_{q}(\mathbf{v})\right)\mathbf{v} + \boldsymbol{g}(\boldsymbol{\eta}) = \boldsymbol{\tau} \quad (5)$$

where $\boldsymbol{M}_{RB}$ is rigid body mass matrix, $\boldsymbol{M}_{A}$ is additional mass matrix , $\boldsymbol{C}(\mathbf{v})$ is Coriolis and centripetal matrix , $\boldsymbol{D}_{l}$ is hydrodynamic linear damping matrix, $\boldsymbol{D}_{q}(\mathbf{v})$ is hydrodynamic nonlinear damping matrix , $\boldsymbol{g}(\boldsymbol{\eta})$ is restoring force and moment matrices, $\boldsymbol{\tau}$ is robot driving force and moment matrix 。

Coriolis and centripetal matrices $\boldsymbol{C}(\boldsymbol{v})$ can be ignored because of the slow speed of ASR. The center of buoyancy, the center of mass coincides with the geometric center of the robot. The restoring force $\boldsymbol{g}(\boldsymbol{\eta})$ is set to $\mathbf{0}_{6\times1}$. After simplification above, the dynamic equations are as follows:

$$\boldsymbol{M}\dot{\mathbf{v}} + \boldsymbol{D}(\mathbf{v})\mathbf{v} = \boldsymbol{\tau} \quad (6)$$

The vectors in the equation (6) are composed as follows.

$$\begin{gathered} \boldsymbol{M} = diag\left(M_{\dot{u}}, M_{\dot{r}}\right) \\ \boldsymbol{D}(\mathbf{v}) = diag\left(d_{11}, d_{22}\right) + diag\left(X_{u}|u|, N_{r}|r|\right) \\ \mathbf{v} = [u, r]^{T} \\ \boldsymbol{\tau} = \left[F_{u}, F_{r}\right]^{T} = \boldsymbol{Lu} \\ \boldsymbol{u} = \left[u_{1}, u_{2}, u_{3}, u_{4}\right]^{T} \end{gathered} \quad (7)$$

where $\boldsymbol{L}$ is transition matrix of torque and control inputs, $\boldsymbol{u}$ is the control input solved by the controller

C. Problem Formulation

When ASR runs in the real dynamic environment, constraints such as saturation and velocity increment of thrusters need to be taken into account. Here, we address the Problem 1.

Problem 1: Design a control rule for ASR to track a expected trajectory while satisfying the following considerations.

1) Error between expected and real trajectory is as small as possible.

2) Operating limits in the form of state constraints (such as speed limits) and input constraints (thrust saturation) are considered.

III. TRAJECTORY TRACKING CONTROLLER

To solve the Problem 1, this paper adopts controller based on the MPC strategy, as shown in Fig. 3.The continuous reference trajectory is manually input and the trajectory points are obtained after discretization of the continuous trajectory. The MPC controller calculates the control input to enable the robot to track the upper trajectory points.

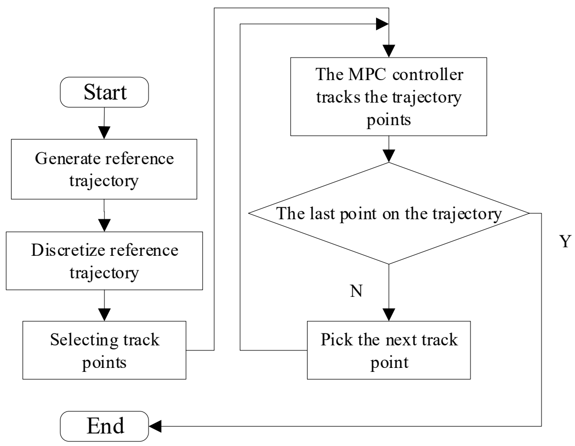


Fig. 3. The trajectory tracking controller based MPC

A. Trajectory tracking controller using MPC

The designed tracking controller needs to achieve the following control effects:(1) The error between the real trajectory of the robot and the expected trajectory is as small as possible. (2) Consider the input constraint (thrust saturation) constraint form. The output of the controller is set to the thrust of the water jet motor. The state space model which combine dynamics and kinematics is adopted as the prediction model of the MPC controller, as shown in equation (8).

$$\begin{aligned}\dot{\mathbf{x}} &= \begin{bmatrix} \mathbf{R}(\boldsymbol{\theta})\mathbf{v} \\ \mathbf{M}^{-1}(\boldsymbol{\tau}-\boldsymbol{D}(\mathbf{v})\mathbf{v}) \end{bmatrix} \\ &= \mathbf{A}\mathbf{x}+\mathbf{B}\tau \\ &= \mathbf{f}(\mathbf{x},\boldsymbol{\tau})\end{aligned} \tag{8}$$

where the control input is defined as $\boldsymbol{\tau}=[F_u,F_r]^T$ and the state vector is defined as $\boldsymbol{x}=[x,y,\theta,u,r]^T$. The discrete form of equation(8) is expressed as following:

$$\begin{aligned}\dot{\boldsymbol{x}}(k+1) &= \mathbf{A}\boldsymbol{x}(k)+\mathbf{B}\boldsymbol{\tau}(k) \\ \boldsymbol{\eta}(k+1) &= \boldsymbol{C}\boldsymbol{x}(k)\end{aligned} \tag{9}$$

where $\boldsymbol{\eta}=[x,y,\theta]^T$. In order to avoid the abrupt change of the force of the water-jet motor, which will affect the continuity of the motor speed, we introduce the control increment. The equation of state is transformed into:

$$\begin{aligned}\boldsymbol{\xi}(k+1) &= \tilde{\boldsymbol{A}}\,\boldsymbol{\xi}(k)+\tilde{\boldsymbol{B}}\,\Delta\boldsymbol{U}(k) \\ \boldsymbol{\eta}(k) &= \tilde{\boldsymbol{C}}\,\boldsymbol{\xi}(k) \\ \boldsymbol{Y}(\boldsymbol{k}) &= \boldsymbol{\Psi}_k\xi(\boldsymbol{k})+\boldsymbol{\Theta}_k\Delta\boldsymbol{U}(\boldsymbol{k}) \\ \xi(\boldsymbol{k}) &= [\boldsymbol{x}(\boldsymbol{k}),\boldsymbol{\tau}(\boldsymbol{k}-1)]^T \\ \boldsymbol{Y}(\boldsymbol{k}) &= [\boldsymbol{\eta}(\boldsymbol{k}),\ldots,\boldsymbol{\eta}(\boldsymbol{k}+\boldsymbol{N}_p)]^T\end{aligned} \tag{10}$$

The solution of trajectory tracking controller based on MPC can be transformed into the following optimization problems:

$$\begin{aligned}J(k) &= \sum_{i=1}^{N_p}\left\|\boldsymbol{\eta}(k+i)-\boldsymbol{\eta}_{ref}(k+i)\right\|_Q^2 \\ &+\sum_{i=1}^{N_c-1}\left\|\Delta\boldsymbol{U}(k+i)\right\|_R^2+\rho\varepsilon^2 \\ \text{s. t. } & \Delta\boldsymbol{U}_{\min}\le\Delta\boldsymbol{U}_t\le\Delta\boldsymbol{U}_{\max} \\ & \boldsymbol{U}_{\min}\le A\Delta\boldsymbol{U}_t+\boldsymbol{U}_t\le\boldsymbol{U}_{\max}\end{aligned} \tag{11}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ is the weight matrices can be adjusted, N_C is the controlling horizon and N_P is the prediction horizon. $\boldsymbol{U}_{\max}$ is the upper limit and $\boldsymbol{U}_{\min}$ is the lower limit of control inputs. $\Delta\boldsymbol{U}_{\max}$ is the upper limit and $\Delta\boldsymbol{U}_{\min}$ is the lower limit of control inputs increments.

The optimization problem of equation (11) above can be transformed into the following standard quadratic optimization problem.

$$\begin{aligned}& J(\boldsymbol{\xi}(t),\boldsymbol{u}(t-1),\Delta\boldsymbol{U}(t)) = \\ & \left[\Delta\boldsymbol{U}(t)^{\mathrm{T}},\varepsilon\right]^{\mathrm{T}}\boldsymbol{H}_t\left[\Delta\boldsymbol{U}(t)^{\mathrm{T}},\varepsilon\right]+\boldsymbol{G}_t\left[\Delta\boldsymbol{U}(t)^{\mathrm{T}},\varepsilon\right]\end{aligned} \tag{12}$$

$$\boldsymbol{H}_t=\begin{bmatrix}\boldsymbol{\Theta}_t^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\Theta}_t+\boldsymbol{R} & 0 \\ \mathbf{0} & \rho\end{bmatrix},\boldsymbol{G}_t=\left[2\boldsymbol{e}_t^{\mathrm{T}}\boldsymbol{Q}\boldsymbol{\Theta}_t,\mathbf{0}\right] \tag{13}$$

where $\boldsymbol{e}^{\mathrm{T}}$ is to predict the tracking error of the robot.

After the minimization problem is solved, the first element is used to the actual control input to control the robot. The trajectory tracking process is shown in Fig. 4.

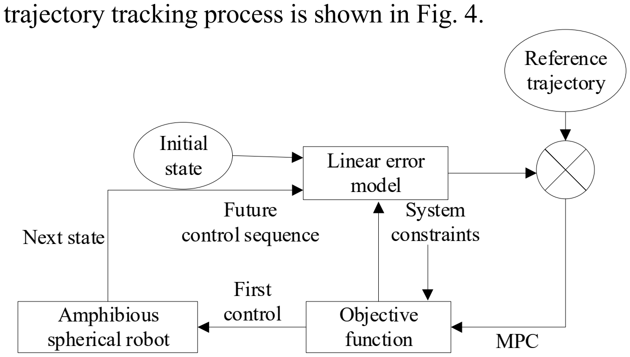


Fig. 4. The trajectory tracking process

B. Trajectory tracking controller using adaptive MPC

There are many adjustable parameters of MPC. For multivariable system, the influence of adjustable parameters on different performance indexes is coupled with each other, which increases the debugging difficulty of MPC control system. This paper designs an adaptive parameter adjustment method, which automatically adjusts the weight matrix Q according to the change of tracking error in MPC.

Assuming the reference trajectory is known, the system output of the reference is

$$\boldsymbol{Y}_{ref}(k)=[\eta_{ref}(k),\ldots,\eta_{ref}(k+N_p)]^T \tag{14}$$

Set the system output error $\boldsymbol{V}(i)$ as:

$$V(i)=\frac{\sum_{j=1}^{N_p}\left[Y(j)-Y_{ref}(j)\right]}{N_p} \quad (15)$$

The weight matrix is designed as follow:

$$Q(i)=KV(i) \quad (16)$$

where K is a parameter that can be adjusted. In order to ensure that the weights corresponding to the calculated parameters do not differ too much, the normalization of $Q(i)$ is carried out. S_i is the amplification coefficient, which can be set according to actual needs.

$$Q_{normal}(i)=\frac{S_i}{1+e^{-Q(i)+2.5}} \quad (17)$$

IV. Simulation And Results

A. Straight trajetory tracking in Matlab

The initial states of the robot is set to $x=[x,y,\theta,u,r]^T$ = $[0,0,0,0,0]^T$ and the sampling time is set to $T=0.05s$. Parameters of MPC controller are set to predictive horizon $N_p=15$ and control horizon $N_c=8$. The control input constraint is [-4, -4]~[4, 4]. The reference trajectories are $y = 3$ and $y=x$. Based on the above simulation parameters, simulation results of tracking are shown in Fig. 5. and Fig. 6.

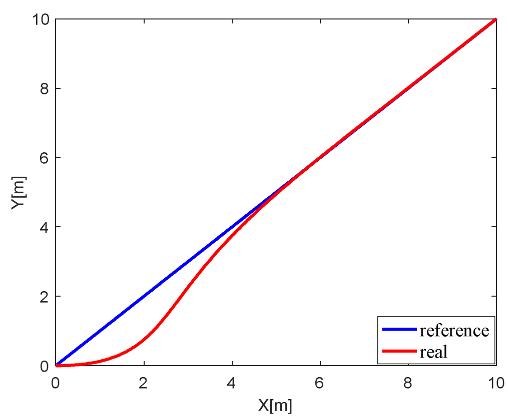


Fig. 5. Line $y = 3$ tracking generated by MPC

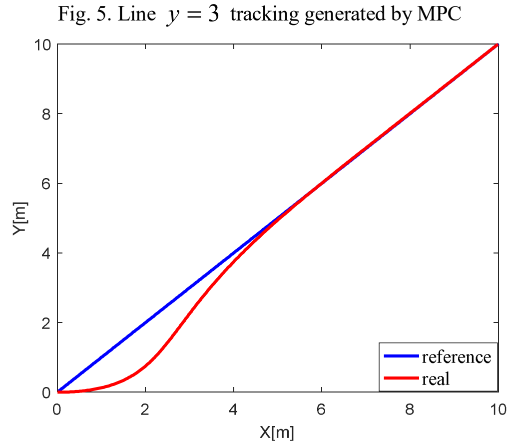


Fig. 6. Line $y = x$ tracking generated by MPC

B. Rectangle trajetory tracking in MATLAB

The simulation environment is the same as the previous section, and the reference trajectory is a broken line composed of $y=3$, $x=15$ and $y=-12$.

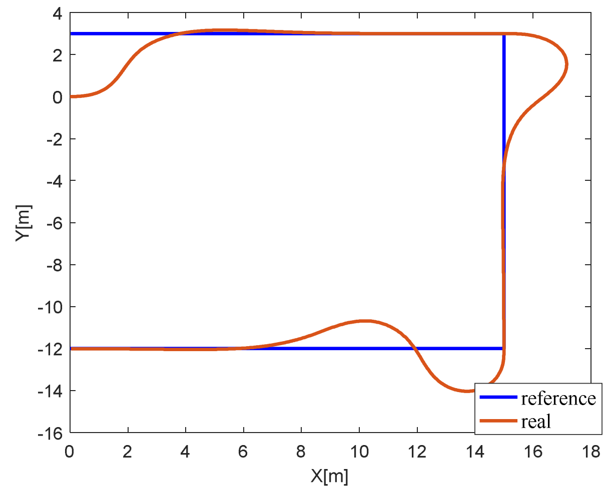


Fig. 7. Square tracking generated by MPC

C. Trajetory tracking in Gazebo

The Gazebo platform is designed to simulate a real underwater environment, so the Gazebo platform is used to verified the validity of the algorithm designed in this paper when tracking straight lines. According to Fig. 8. and Fig. 9., we can see that the robot keeps up with the expected linear trajectory, and the effect of the algorithm meets the requirements.

In Fig. 8., the initial value of the Y direction error is very large, after adjusting the Y direction, controller gradually pay attention to the Y direction error adjustment.

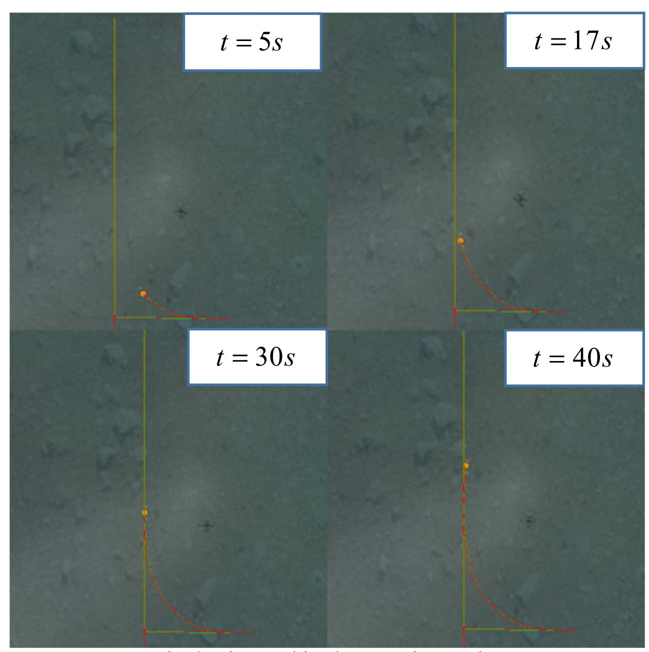
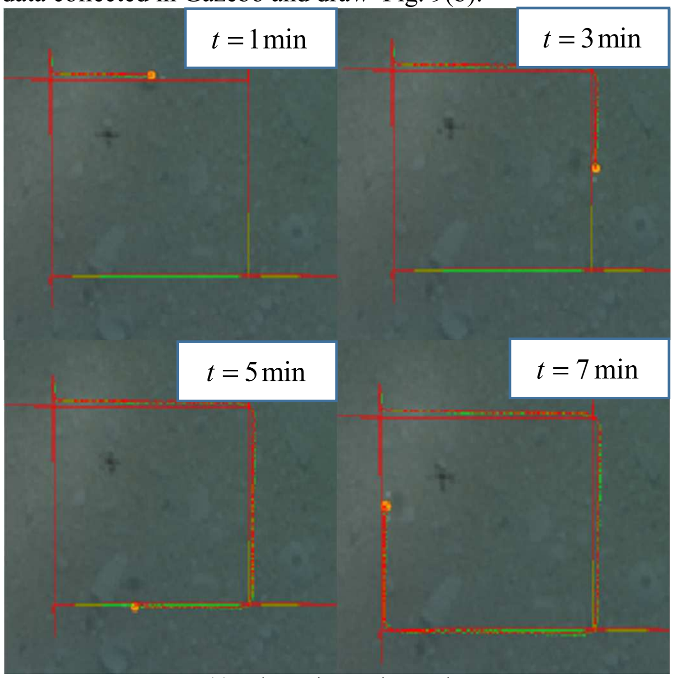
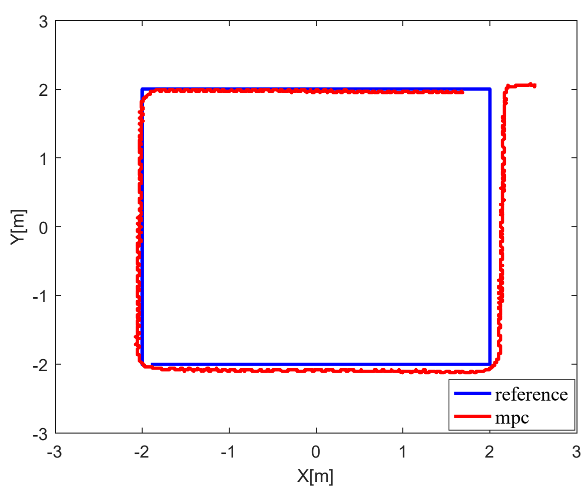


Fig. 8. Line tracking by MPC in Gazebo

In Fig. 9, The expected trajectory is a clockwise rectangle whose vertex is located at (2,2),(-2,2),(-2,-2),(2,-2). The starting point is (2.5,2). Fig. 9(a) shows the actual trajectory of the robot in Gazebo. In order to show the reference track and the actual track more clearly, MATLAB is used to integrate the data collected in Gazebo and draw Fig. 9(b).



(a) Robot trajectory in Gazebo



(b) Trajectory in Gazebo drawn by MATLAB
Fig. 9. Square tracking by MPC in Gazebo

The tracking effects of adaptive MPC and MPC are compared below. The MPC parameter is set to $Q = diag(1,10,1,1,1)$, and the adaptive MPC parameter is set to $S_2 = 10$. The comparison effect is shown in the Fig. 10. It can be seen that the adaptive MPC has better adjustment effect than MPC.

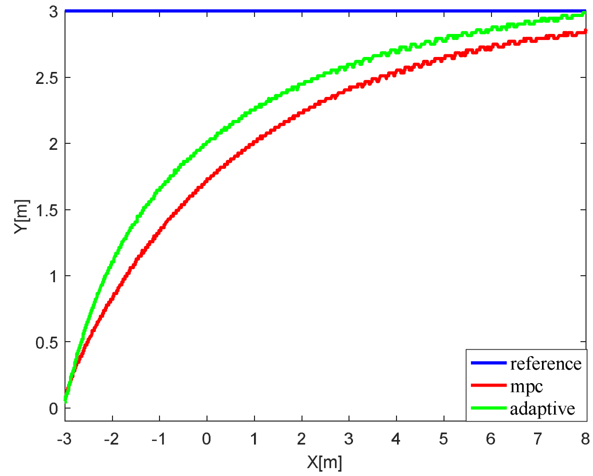


Fig. 10. Line tracking by MPC and adaptive-MPC in Gazebo.

V. Conclusions

To solve the problem of trajectory tracking, A MPC-based controller was designed through the simulation verification of linear trajectory and square curve trajectory on the simulation platform. The ASR is able to track the expected trajectory with deviation as small as possible. The stability and universality of the controller designed in this paper are verified.

Acknowledgment

This work was partly supported by National High Tech. Research and Development Program of China (No.2015AA043202), the National Natural Science Foundation of China (61773064, 61503028), the National Key Research and Development Program of China (No. 2017YFB1304401).

References

[1] S. Heshmati-Alamdari, G. C. Karras, P. Marantos, and K. J. Kyriakopoulos, "A Robust Predictive Control Approach for Underwater Robotic Vehicles," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 6, pp. 2352–2363, October 2019.

[2] W. Wang, T. Shan, P. Leoni, et al. "Robot II: A Novel Autonomous Surface Vessel for Urban Environments," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Jul. 2020, pp. 1740–1747.

[3] Y. Cao, B. Li, Q. Li, et al, "A Nonlinear Model Predictive Controller for Remotely Operated Underwater Vehicles With Disturbance Rejection," *IEEE Access*, vol. 8, pp. 158622–158634, doi: 10.1109/ACCESS.2020. 3020530, 2020.

[4] J. Guo, C. Li, and S. Guo, "A Novel Step Optimal Path Planning Algorithm for the Spherical Mobile Robot Based on Fuzzy Control," *IEEE Access*, vol. 8, pp. 1394-1405, doi: 10.1109/ACCESS.2019. 2962074, 2020.

[5] W. Gan, D. Zhu, W. Xu, et al, "Survey of trajectory tracking control of autonomous underwater vehicles," *Journal of marine science and technology*, vol. 25, no. 6, pp. 722–731, March 2017.

[6] Geranmehr B, Nekoo S R, "Nonlinear suboptimal control of fully coupled non-affine six-DOF autonomous underwater vehicle using the state-dependent Riccati equation," *Ocean Engineering*, pp. 248-257, doi: 10.1016/j.oceaneng.2014.12.032, March 2015.

[7] A, Hossein, V. A. A, and M. D. B, "A Time Delay Controller included terminal sliding mode and fuzzy gain tuning for Underwater Vehicle-Manipulator Systems,"*Ocean Engineering*, pp. 97-107, doi: 10.1016/ j.oceaneng.2015.07.043, 2015.

[8] Joe, H, M. Kim, and S. C. Yu, "Second-order sliding-mode controller for autonomous underwater vehicle in the presence of unknown disturbances," *Nonlinear Dynamics* pp. 183-196, doi: 10.1007/s11071-014-1431-0, 2014.

[9] Elmokadem, T., M. Zribi, and K. Youcef-Toumi, "Terminal sliding mode control for the trajectory tracking of underactuated Autonomous Underwater Vehicles," *Ocean Engineering*, pp. 613-625, doi: 10.1016/j.oceaneng.2016.10.032, 2016.

[10] S. Mahapatra, and B. Subudhi. "Nonlinear H∞ state and output feedback control schemes for an AUV in the dive plane," *Transactions of the Institute of Measurement and Control*, pp. 2024-2038, doi: 10.1177/0142331217695671, 2018.

[11] Geranmehr, B., and S. R. Nekoo, "Nonlinear suboptimal control of fully coupled non-affine six-DOF autonomous underwater vehicle using the state-dependent Riccati equation," *Ocean Engineering,* pp. 248-257, doi: 10.1016/j.oceaneng.2014.12.032, 2015.

[12] Peng, Z., and J. Wang, "Output-Feedback Path-Following Control of Autonomous Underwater Vehicles Based on an Extended State Observer and Projection Neural Networks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 4, pp. 535-544, April 2018.

[13] R. Cui, C. Yang, Y. Li, et al, "Adaptive neural network control of auvs with control input nonlinearities using reinforcement learning," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 6, pp. 1019–1029, June 2017.

[14] X. Liang, X. Qu, L. Wan, et al, "Three-dimensional path following of an underactuated auv based on fuzzy backstepping sliding mode control," *International Journal of Fuzzy Systems*, vol. 20, no. 2, pp. 640–649, February, 2017.

[15] X. Hou, S. Guo, L. Shi, et al, "Improved Model Predictive-Based Underwater Trajectory Tracking Control for the Biomimetic Spherical Robot under Constraints," *Applied Sciences*, vol. 10, no. 22, pp. 8106, doi: 10.3390/app10228106, 2020.

[16] S. Guo, Y. He, L. Shi, et al, "Modeling and experimental evaluation of an improved amphibious robot with compact structure," *Robotics and Computer-Integrated Manufacturing*, pp. 37-52, doi: 10.1016/j.rcim. 2017.11.009, 2018.

[17] L. Zheng, S. Guo, Y. Piao, et al, "Collaboration and task planning of turtle-inspired multiple amphibious spherical robots," *Micromachines*, vol. 11, no. 1, pp. 71–71, January 2020.

[18] S. Gu, S. Guo, and L. Zheng, "A highly stable and efficient spherical underwater robot with hybird propulsion devices," *Autonomous Robots*, pp. 759–771, doi: 10.1007/s10514-019-09895-8, 2020.

[19] H. Xing, S. Guo, L. Shi, X. Hou, and H. Liu, "Design, modeling and experimental evaluation of a legged, multi-vectored water-jet composite driving mechanism for an amphibious spherical robot," *Microsystem Technologies*, vol. 26, no. 5, doi: 10.1007/s00542-019-04536-7, 2019.

[20] H. Xing, S. Guo, L. Shi, et al, "Hybrid locomotion evaluation for a novel amphibious spherical robot," *Applied Sciences*, vol. 8, no. 2, pp. 156, January 2018.

[21] H. Xing, L. Shi, K. Tang, et al, "Robust rgb-d camera and imu fusion-based cooperative and relative close-range localization for multiple turtle-inspired amphibious spherical robots," *Journal of Bionic Engineering*, vol. 16, no. 3, pp. 442–454, May 2019.

[22] H. Xing, S. Guo, L. Shi, et al, "A novel small-scale turtle-inspired amphibious spherical robot," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Nov. 2019, pp. 1702–1707.

[23] X. Hou, S. Guo, L. Shi, et al, "Hydrodynamic analysis-based modeling and experimental verification of a new water-jet thruster for an amphibious spherical robot," *Sensors*, vol. 19, no. 2, pp. 259, January 2019.

[24] Z. Yan, J. Li, G. Zhang, et al, "A real-time reaction obstacle avoidance algorithm for autonomous underwater vehicles in unknown environments," *Sensors*, vol. 18, no. 2, pp. 438–438, February 2018.

[25] C. Shen, B. Buckham, and Y. Shi, " Modified C/GMRES Algorithm for Fast Nonlinear Model Predictive Tracking Control of AUVs," *IEEE Transactions on Control Systems Technology*, vol. 25, no. 5, pp. 1896-1904, November 2016.