THE JOURNAL OF NAVIGATION (2019), **72**, 321–341.
doi:10.1017/S0373463318000668

Trajectory Tracking and Re-planning with Model Predictive Control of Autonomous Underwater Vehicles

Zhen Hu[1], Daqi Zhu[2], Caicha Cui[2] and Bing Sun[2]

[1](*China Ship Scientific Research Center, Wuxi, China*)
[2](*Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Haigang Avenue 1550, Shanghai, 201306, China*)
(E-mail: zdq367@aliyun.com)

The trajectory tracking of Autonomous Underwater Vehicles (AUV) is an important research topic. However, in the traditional research into AUV trajectory tracking control, the AUV often follows human-set trajectories without obstacles, and trajectory planning and tracking are separated. Focusing on this separation, a trajectory re-planning controller based on Model Predictive Control (MPC) is designed and added into the trajectory tracking controller to form a new control system. Firstly, an obstacle avoidance function is set up for the design of an MPC trajectory re-planning controller, so that the re-planned trajectory produced by the re-planning controller can avoid obstacles. Then, the tracking controller in the MPC receives the re-planned trajectory and obtains the optimal tracking control law after calculating the object function with a Sequential Quadratic Programming (SQP) optimisation algorithm. Lastly, in a backstepping algorithm, the speed jump can be sharp while the MPC tracking controller can solve the speed jump problem. Simulation results of different obstacles and trajectories demonstrate the efficiency of the proposed MPC trajectory re-planning tracking control algorithm for AUVs.

KEYWORDS

1. Autonomous Underwater Vehicles. 2. Trajectory Re-planning. 3. Trajectory Tracking. 4. Obstacle Avoidance. 5. Model Predictive Control.

Submitted: 9 February 2018. Accepted: 12 August 2018. First published online: 21 September 2018.

1. INTRODUCTION. The trajectory tracking control of Autonomous Underwater Vehicles (AUVs) infers that an AUV will follow a predefined reference trajectory under the designed control law. It starts from a given initial state and achieves global uniform asymptotic stability on the premise of meeting the requirements of position error (Fossen, 2002). The underwater environment is complex and changeable, and the study of trajectory tracking control and AUV obstacle avoidance has been a research hotspot.

From earlier studies, it is known that sliding mode control can be used in the tracking control system (Barth et al., 2016; Jia et al., 2012). Sliding mode control is essentially a special class of nonlinear control, it is insensitive to parameter changes and does not need to accurately model vehicle dynamics. For example, in Wu et al. (2017), a novel adaptive

terminal sliding mode controller with a strictly lower convex function is proposed for the longitudinal dynamics of a Hypersonic Flight Vehicle (HFV), and the controller exhibits an excellent robustness and disturbance rejection performance. Work has been done on tracking control for AUVs (Bessa et al., 2008; Bagheri and Moghaddam, 2009; Serdar et al., 2008), but the 'chattering' problem still exists. Frequent buffeting will cause significant loss of electric power and excessive wear of actuator mechanisms, which will have a great influence on the precision equipment of an AUV. The neural network algorithm has been studied for tracking control of robots (Ye, 2014; Hoseini et al., 2008). Pan et al. (2013) proposed a single-layer neural network structure for tracking control. In Miao et al. (2013), an adaptive neural network controller was reported, and a Radial Basis Function (RBF) neural network was used to calculate the tracking error and achieved stable tracking. Neural network control does not need a precise dynamics model, and it has strong adaptability and learning ability. It is suitable for uncertain or highly nonlinear control objects like an AUV. However, because of the complex and changeable underwater conditions, the training sample is difficult to collect and the learning process lags behind.

Backstepping control is widely used in the field of trajectory tracking control for ground mobile robots (Tsai et al., 2004; Yang and Luo, 2004; Luo et al., 2008). In recent years, backstepping control has been further applied to trajectory tracking control of underwater vehicles. For example, in Sun et al. (2014), the backstepping method was integrated with a bio-inspired model and a control approach was designed for three-dimensional tracking control of Unmanned Underwater Vehicles (UUV). Xiang et al. (2015) designed a nonlinear controller based on Lyapunov theory and the backstepping technique for an under-actuated and fully-actuated AUV, and smooth control transition between fully/under-actuated configurations was enabled using this control system, but for the traditional backstepping algorithm, it will have a speed jump problem if the tracking error is large or a tracking inflection point exists. As the required thrust in AUV trajectory tracking increases with the increase of the velocity hopping variable, once the speed jump is too large, tracking and planning accuracy will be affected, and the AUV will not be able to maintain the desired trajectory due to the constraint of limited thrust. Thus, a sharp speed jump should be avoided in the real motion of an AUV.

The above reported works do not address the internal constraints, including actuator saturation, velocity increment, and acceleration constraints of the AUV. To include these constraints in the controller designs, Model Predictive Control (MPC) is an ideal tool, because it can handle constraints through optimisation procedures. Some work on the tracking of AUV and ground robots using MPC has been done in recent years (Li et al., 2016; Bahadorian et al., 2012). MPC can conduct online optimisation of an objective function through an input–output predictive model over a finite future horizon of sample times. In Shen et al. (2016), an objective function based on the current controlled variables can be optimally computed at each step by a Sequential Quadratic Programming (SQP) algorithm. In the next horizon sampling interval, the optimisation is re-conducted with updated corresponding variables. Then, a sequence of control inputs can be obtained at each step. MPC consists of control and planning.

In addition, most of the research into trajectory tracking of AUVs does not take into account obstacles, and tracks a human-set trajectory to test the tracking efficiency. Trajectory tracking and planning are separated. In the real underwater environment, an AUV needs to track an online planned trajectory to avoid obstacles (Zhu et al., 2018). Some research on this problem for unmanned robots has been done in recent years (Kim et al.,

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

2002; Cui et al., 2012). However, the study of tracking a planned trajectory with obstacle avoidance for an AUV is rare.

Building from previous research, this paper focuses on the problem of speed jump and separation of tracking and planning. Inspired by the guidance and control thought in Xiang et al. (2018), a trajectory re-planning controller and a trajectory tracking controller based on a Model Predictive Control (MPC) algorithm are presented in this paper. Using the AUV states, the global reference trajectory and the obstacle information from sensors, the MPC trajectory re-planning controller plans a local reference trajectory (re-planned trajectory) and provides it to the MPC trajectory tracking controller for guiding the AUV to avoid obstacles. Because of the initial error, an MPC tracking controller is designed so that it can make sure the AUV tracks along the re-planned trajectory. The tracking controller deals with the re-planned trajectory and states information and calculates an optimal control law after solving an objective function with AUV system constraints by a SQP optimisation method. Finally, under the two-layer control frame synthesising the planning and tracking, the AUV overcomes the initial state error and executes the re-planned trajectory to avoid obstacles until it reaches the target position. Compared to backstepping control in different trajectories with different types of obstacles, it can be seen from simulation results on MATLAB that backstepping control has a sharp speed jump problem, while because the MPC can deal with models of constraints, the MPC controller can solve the speed jump problem and stably track the re-planned trajectory to avoid obstacles by taking into account the input velocity constraints, and the tracking error converges to zero.

The main contribution of this paper is two-fold: First, a MPC trajectory re-planning tracking control algorithm is proposed to solve the AUV trajectory tracking control problem. With the proposed MPC, the performance of the tracking control can be significantly improved. Besides which, the speed jump problem can be simultaneously addressed within the controller design. Secondly, a MPC re-planning controller is designed and added to the tracking controller, and the AUV can avoid obstacles in real-time by following a re-planned trajectory when tracking under the tracking controller.

In this paper, the kinematic model of an AUV is described in Section 2. Then in Section 3, the algorithm of tracking and re-planning based on MPC is proposed. Simulation results are given in Section 4 and discussed. The work is summarised in Section 5.

2. THE MULTI-AUV SYSTEM PROBLEM.

An underwater vehicle has six degrees of freedom of motion in space. The kinematics problem is described by setting up a coordinate system.

2.1. *The coordinate system.*

In order to make the motion analysis of the vehicle clearer and simpler, this paper uses two coordinate systems to describe the kinematics problems of an underwater vehicle, an inertial coordinate system and a vehicle coordinate system. Figure 1 depicts the coordinate systems (Sun et al., 2016). Inertial coordinate system $\mathrm{E}-\xi\eta\zeta$ is also called the Earth coordinate system or fixed coordinate system, and its origin is a certain point on Earth. Vehicle coordinate system O–*xyz* is fixed on the AUV and moves with the AUV.

The vector $\boldsymbol{\eta} = [\,x\ y\ z\ \phi\ \theta\ \psi\,]^T$ is used to describe the position and attitude of the AUV in inertial coordinates, and the vector $\boldsymbol{\nu} = [\,u\ v\ w\ p\ q\ r\,]^T$ describes the linear and angular velocity of the AUV in the vehicle coordinates. The symbols x, y, z represent the position of the AUV respectively in inertial coordinates; ϕ, θ, ψ represent the attitude of the AUV;

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

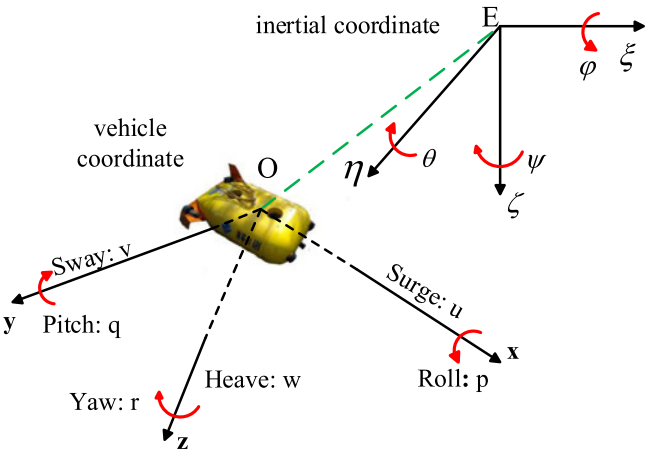

Figure 1. The coordinate system.

roll angle, pitch angle and yaw angle in inertial coordinates. The symbols u, v, w represent the three components of the AUV's linear velocity vector in vehicle coordinates and p, q, r represent the three components of the AUV's angular velocity vector in vehicle coordinates.

2.2. *Simplified kinematic model.* The separate description of the coordinate systems can reduce the difficulty of understanding to a great extent. The method of combining two coordinate systems has a coordinate transformation relationship when the kinematic model is established, and the kinematics equation is obtained as follows:

$$\dot{\boldsymbol{\eta}} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{v} \tag{1}$$

The matrix $\boldsymbol{J}(\boldsymbol{\eta})$ is the posture transformation matrix between inertial coordinates and the vehicle coordinates. We call it the Jacobi transform matrix.

$$\boldsymbol{J}(\boldsymbol{\eta}) = \begin{bmatrix} \boldsymbol{J}_1 & 0_{3\times3} \\ 0_{3\times3} & \boldsymbol{J}_2 \end{bmatrix} \tag{2}$$

$$\boldsymbol{J}_1(\boldsymbol{\eta}) = \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\varphi - \sin\psi\cos\varphi & \cos\psi\sin\theta\sin\varphi + \sin\psi\sin\varphi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\varphi + \cos\psi\cos\varphi & \sin\psi\sin\theta\cos\varphi - \cos\psi\sin\varphi \\ -\sin\theta & \cos\theta\sin\varphi & \cos\theta\cos\varphi \end{bmatrix}$$

$$\boldsymbol{J}_2(\boldsymbol{\eta}) = \begin{bmatrix} 1 & \tan\theta\sin\varphi & \cos\varphi\tan\theta \\ 0 & \cos\varphi & -\sin\varphi \\ 0 & \sin\varphi/\cos\theta & \cos\varphi/\cos\theta \end{bmatrix}$$

Since the trajectory tracking we research in this paper is Two-Dimensional (2D), the rolling, pitching and heaving can be overlooked. Usually, we only consider the trajectory tracking control problem under the three most common degrees of freedom: surge, sway and yaw. We can use $z = \theta = \varphi = 0$ and $w = p = q = 0$ to simplify Equation (1). The mutual relationship between $\boldsymbol{\eta} = [\,x\ y\ \psi\,]^T$ and $\boldsymbol{v} = [\,u\ v\ r\,]^T$ is as follows:

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = \boldsymbol{J}(\boldsymbol{\eta})\boldsymbol{v} = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ r \end{bmatrix} \tag{3}$$

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

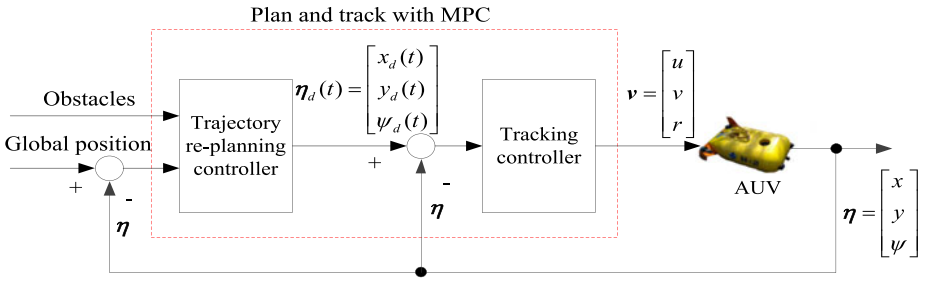

Figure 2. Trajectory tracking control system combined with planning layer.

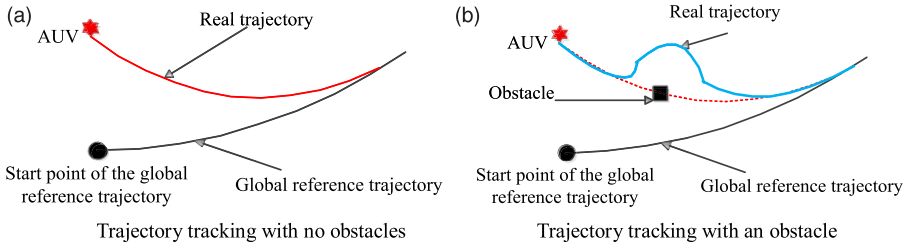

Figure 3. Sketch map of re-planning and tracking.

When the velocity is obtained, the state of the AUV can be obtained by solving differential equations.

2.3. *Description of the trajectory tracking of an AUV.* Here we consider the trajectory tracking problem. This means that we want to find a control law so that the AUV can track the re-planned reference path. The re-planned reference position of the AUV can be described as $\boldsymbol{\eta}_d(t) = [\, x_d(t)\ y_d(t)\ \psi_d(t) \,]$, and $\boldsymbol{\eta}(t) = [\, x(t)\ y(t)\ \psi(t) \,]$ is the actual position. The so-called kinematic trajectory tracking control of an AUV is through the control of actual surge speed $u(t)$, sway speed $v(t)$, and yaw speed $r(t)$, so that the AUV can follow the desired trajectory $\boldsymbol{\eta}_d$, which has been given, and ultimately making the error between the actual trajectory $\boldsymbol{\eta}$ and desired trajectory gradually converge to zero. That is:

$$\lim_{t\to\infty} e_\eta(t) = \lim_{t\to\infty} (\boldsymbol{\eta}_d(t) - \boldsymbol{\eta}(t)) = 0 \tag{4}$$

When time t tends to be infinite, the posture error of the AUV approaches to zero.

3. TRACKING AND RE-PLANNING USING MPC. The trajectory re-planning controller is added to the tracking controller and forms a new control system, as shown in Figure 2. If there is no obstacle in the way, the AUV will track the global reference trajectory under the instructions of the tracking controller, shown as Figure 3(a). When the AUV encounters an obstacle, the trajectory re-planning controller generates an optimised trajectory $\boldsymbol{\eta}_d(t)$ with obstacle avoidance as shown in Figure 3(b), and then the optimised trajectory is sent to the tracking controller which navigates the AUV to avoid the obstacle.

The inputs of the MPC re-planning controller include the goal point, the deviation of global reference position, the AUV's real state $\boldsymbol{\eta}$ and the obstacle information. The re-planned trajectory in the prediction horizon with obstacle avoidance can be obtained after

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

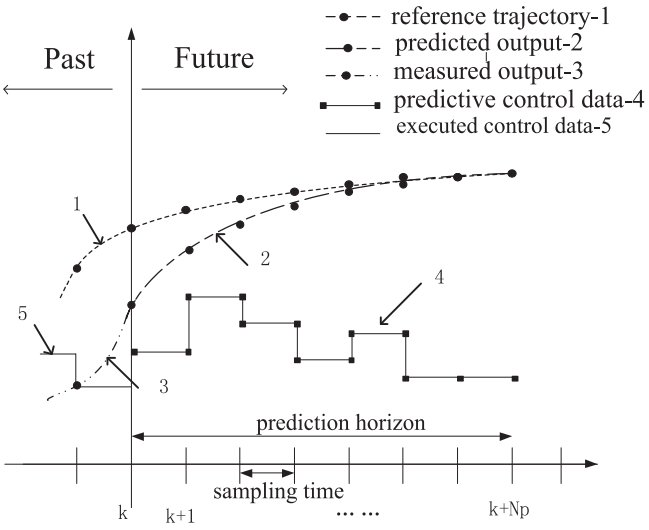

Figure 4. Model predictive control algorithm.

solving a cost function with obstacles and speed constraints by using the MPC algorithm and SQP optimisation method. The input of the MPC tracking controller is the deviation of the re-planned trajectory (in a prediction horizon) and the AUV's real state $\boldsymbol{\eta}$. The tracking controller based on MPC calculates a tracking control sequence for tracking the re-planned trajectory by optimising the object function with SQP. Notice that the system constraints, especially the speed constraints, are added in the object function. Finally, the tracking controller outputs the control signal $\boldsymbol{v}$ (the first item in the optimised control sequence) to the AUV, and under the control signal, the AUV will return to track the global reference trajectory after avoiding the obstacles. More detailed design information of the controllers is discussed in Sections 3.2 and 3.3.

3.1. *The MPC algorithm.* Model predictive control consists of three basic principles, predictive model, receding horizon optimisation and feedback correction. Based on these principles, the theory of MPC can be expressed in Figure 4. Np represents the time length of the predicted output, that is, the prediction horizon. In Figure 4, curve 1 represents a desired trajectory that always exists. At every time step k, the future system output is predicted for a pre-determined Np steps ahead of the controller, as shown in curve 2. The predicted outputs are a function of measurements of current and past inputs and outputs in conjunction with a future control input. By solving the optimisation problem by minimising a cost function subject to constraints on the system, a control sequence can be obtained in the control horizon $[k, k + N_c]$, as shown in curve 4. The first item of the control sequence is the actual control signal working on the controlled object. When the next moment $k + 1$ comes, the above process is repeated to achieve continuous control of the controlled object.

The stages of implementing the MPC algorithm on the tracking control process are shown in Figure 5. The MPC controller is mainly composed of a predictive model, system constraints and an objective function. The predictive model describes the tracking control system and it is also the basis for building the control algorithm. System constraints include physical constraints of the AUV, control constraints, obstacles and so on.

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

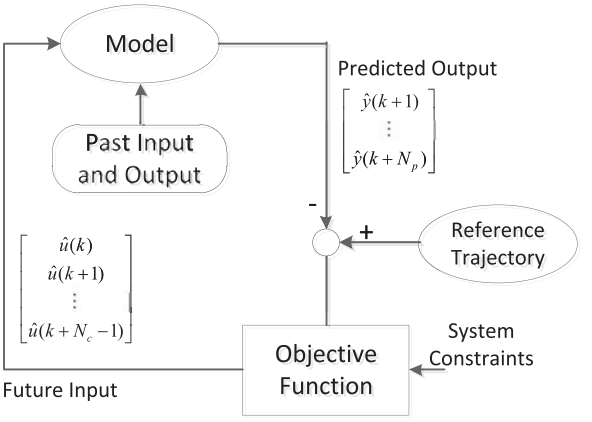

Figure 5. Stages of MPC algorithm.

The optimisation method is used to solve the objective function and then the optimised control law is obtained.

3.2. *Trajectory tracking controller with MPC.* The MPC trajectory tracking controller needs to meet two conditions: (1) The deviation between the actual trajectory and the desired trajectory should be as small as possible. Here the reference trajectory is the trajectory after re-planning. (2) The control data obtained should satisfy the restriction of the AUV speed.

The AUV kinematic model in the horizontal plane has been given as Equation (3). The input value of the AUV system is a speed vector $\boldsymbol{\nu} = [\,u\ v\ r\,]^T$, and the states vector is $\xi = [\,x\ y\ \psi\,]^T$. For a nonlinear system, the discrete form of its general form is:

$$\begin{aligned} &\xi(t+1) = f\,(\xi(t), \boldsymbol{\nu}(t)) \\ &\xi(t) \in \boldsymbol{\chi}, \boldsymbol{\nu}(t) \in \boldsymbol{\Gamma} \end{aligned} \tag{5}$$

where $f\,()$ is the state transfer function, and $f\,(0, 0) = 0$, that is, the origin of the state space and input is an equilibrium point. $\boldsymbol{\chi}$ denotes the state constraints, and $\boldsymbol{\Gamma}$ is the speed constraints.

The control objective is to steer the state of the system Equation (5) to the origin $\xi_e = 0, v_e = 0$. For arbitrary time horizon $N \in Z^+$, consider the following cost function $\boldsymbol{J}_N()$:

$$\boldsymbol{J}_N(\xi(t), \boldsymbol{V}(t)) = \sum_{k=t}^{t+N-1} l(\xi(k), \boldsymbol{\nu}(k)) + P(\xi(t+N)) \tag{6}$$

where $\boldsymbol{V}(t) = [\,\boldsymbol{\nu}(t)\ \ldots\ \boldsymbol{\nu}(t+N-1)\,]^T$ is the input control sequence over the time horizon and $\xi(k)$ for $k = t, \ldots, t+N$ is the real states trajectory after applying the control data $\boldsymbol{V}(t)$ to the AUV, starting from the initial state $\xi(t)$. In Equation (6), the first part $l()$ is the stage cost, and the second part $P()$ is the terminal cost. We assume that Q_0 and Q penalise the terminal trajectory deviation and the running trajectory deviation respectively, and R penalises the

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

large input value.

$$P(\xi(t+N)) = \xi(t+N)^T Q_0 \xi(t+N) \tag{7}$$

$$\begin{aligned} l(\xi(k), \boldsymbol{v}(k)) &= \tilde{\xi}(k)^T Q \tilde{\xi}(k) + \boldsymbol{v}(k)^T R \boldsymbol{v}(k) \\ \tilde{\xi}(k) &= \xi(k) - \xi_{ref,\, plan}(k) \end{aligned} \tag{8}$$

At each sampling time t we solve the following finite time optimisation problem:

$$\min_{V_t, \xi_{t+1}, \ldots, \xi_{t+N}} \quad \boldsymbol{J}_N(\xi_t, \boldsymbol{V}_t) \tag{9a}$$

$$s.t. \quad \xi_{k+t,t} = f(\xi_{k,t}, \boldsymbol{v}_{k,t}), k = t, \ldots, N-1 \tag{9b}$$

$$\xi_{k,t} \in \boldsymbol{\chi} \quad k = t+1, \ldots, t+N-1 \tag{9c}$$

$$\boldsymbol{v}_{k,t} \in \boldsymbol{\Gamma} \quad k = t, \ldots, t+N-1 \tag{9d}$$

$$\xi_{t,t} = \xi(t) \tag{9e}$$

$$\xi_{N,t} \in \boldsymbol{\chi}_{fin} \tag{9f}$$

where Equation (9f) is the final state constraint and $\boldsymbol{\chi}_{fin}$ is a polytope.

Optimal control sequence $\boldsymbol{V}_t^* = [\, \boldsymbol{v}_{t,t}^* \cdots \boldsymbol{v}_{t+N-1,t}^* \,]$ can be obtained after solving Equations (9) at time t by the SQP optimisation method. Denote by $\xi_{k,t}^*$ for $k = t, \ldots, t+N$ the optimal state trajectory obtained by applying the optimal input sequence $\boldsymbol{V}_t^*$ to Equation (9b). According to the basic principles of model predictive control, only the first element of the sequence is applied to the plant:

$$\boldsymbol{v}(\xi(t)) = \boldsymbol{v}_{t,t}^* \tag{10}$$

At the next sampling time, the optimisation problem Equations (9) are solved over a shifted horizon. For more information, see Falcone (2007).

Remark: In the formulation of the constrained finite-time optimal control Equation (9), we distinguish between the state $\xi(t)$ and input control $\boldsymbol{V}(t)$ of the system Equation (5) and the variables $\xi_{k,t}$ and $\boldsymbol{v}_{k,t}$ of the optimisation Equation (9).

3.3. *Trajectory re-planning controller with MPC.* According to the actual situation, in order to improve the accuracy and reduce the error, the trajectory re-planning and tracking controller based on MPC needs to meet certain conditions: (1) The deviation between the re-planning trajectory and the given global reference trajectory information should be as small as possible. (2) The planned trajectory must satisfy the kinematic constraints of the AUV. (3) The planned trajectory can successfully avoid obstacles and meet the safety requirements (Xiang et al., 2017).

3.3.1. *Select the reference point.* Trajectory re-planning requires the calculation of the deviation between the reference trajectory and the predicted trajectory: $\boldsymbol{\eta} - \boldsymbol{\eta}_{ref}$. The method developed by Yoon et al. (2009) is used for basic obstacle avoidance. In order to avoid the phenomenon that the reference point has been re-selected as a new reference point, it is necessary to add the target point information in the selection process of the reference trajectory point.

In the inertial coordinate system, two lines from the AUV parallel to the X and Y axes intersect with the global reference trajectory, and these lines produce two intersection points on the reference trajectory, shown in Figure 6. Point η_k represents the AUV's position at

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

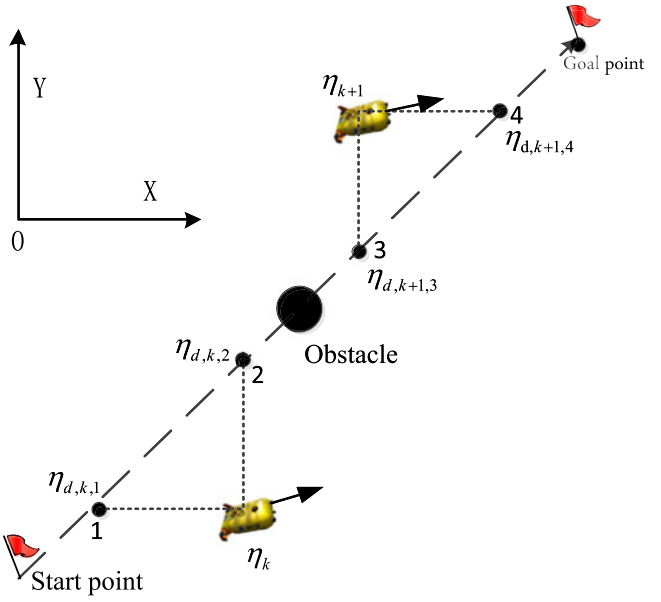

Figure 6. Method for trajectory error calculation.

k time; two intersection points are found on the reference line corresponding to this point, $\eta_{d,k,1}$ and $\eta_{d,k,2}$. Calculate the distance between the goal point and the two intersection points respectively and select the intersection point that is closer to the goal point as the real reference starting point. So, in this case, $\eta_{d,k,2}$ is chosen to be the reference point. When the next time $k+1$ comes, the AUV comes to point η_{k+1}, $\eta_{d,k+1,4}$ will be the real desired point due to its proximity to the goal point. Thus, the reference trajectory criterion has considered the approach to the goal point rather than just following the trajectory in any direction (Abbas et al., 2017).

3.3.2. *Obstacle avoidance function.* The basic idea of a penalty function is to adjust the function value according to the deviation range between the obstacle point and the object point, and the smaller the range, the greater the function value. Obstacle avoidance is incorporated into the controller as a constraint, by including a point-wise repulsive potential function $\boldsymbol{J}_{obs,i}$ into the cost:

$$\boldsymbol{J}_{obs,i} = \frac{S_{obs} v_i}{(x_i - x_0)^2 + (y_i - y_0)^2 + \zeta} \tag{11}$$

where S_{obs} is the weight coefficient, $v_i = v_x^2 + v_y^2$, (x_i, y_i) is the current location of the AUV and (x_0, y_0) is the location of the obstacle to be avoided. ζ is a small positive number that prevents the denominator from going to zero. In order to make the trajectory planning feasible, it is necessary to expand the obstacle according to the size of the AUV. This paper uses the radius of the inner tangent circle and the outer circle of the moving centre of the AUV as the dimension for the expansion of the obstacle. When the size of the obstacle is large, it is necessary to segment the obstacles according to a certain resolution so as to prevent the AUV from passing through the obstacle.

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

The goal of trajectory re-planning is to minimise the deviation from the global reference trajectory when avoiding obstacles. R and Q are positive-definite matrices, $\boldsymbol{\eta}_{plan}$ and $\boldsymbol{\eta}_{ref}$ denote the re-planned trajectory and the global reference trajectory respectively and $\boldsymbol{U}$ is the control input of the re-planning controller. The MPC re-planning controller is designed using the following cost function:

$$\min_{V_t} \sum_{i=1}^{N_p} \left\| \boldsymbol{\eta}_{plan}(t+i|t) - \boldsymbol{\eta}_{ref}(t+i|t) \right\|_Q^2 + \|\boldsymbol{U}_i\|_R^2 + \sum_{j=0}^{N_{obs}} \boldsymbol{J}_{obs,i,j} \tag{12}$$

$$s.t.\ \boldsymbol{U}_{\min} \le \boldsymbol{U}_t \le \boldsymbol{U}_{\max}$$

The cost function Equation (12) has been generalised to include as many obstacles as desired. Any obstacle shape can be generalised by this simple point obstacle method $\boldsymbol{J}_{obs,i}$ if the nearest point on that obstacle is determinable. For example, if a linear object is to be avoided, an appropriate method can be used to find the nearest point on the obstacle's surface and then we treat this point as $\boldsymbol{J}_{obs,i}$. The case with circular obstacles is similar.

According to the AUV kinematic model given by Equation (3), we generalise differential Equation (3) as $\dot{\eta} = f(t, \eta)$. By making the function discrete and using the one-step Euler method, the iterative equation can be obtained:

$$\eta_{n+1} = \eta_n + hf(t_n, \eta_n) \tag{13}$$

where $f()$ denotes the transfer function, h is the iterative step size, and η_{n+1} is the predicted output. In this part, the optimised control input in control horizon $\boldsymbol{U}_t^* = [\boldsymbol{u}_{t+i}^* \cdots \boldsymbol{u}_{t+N_c-1}^*]$ can be obtained after solving the optimisation problem by the SQP method. The last control input, $\boldsymbol{u}_{t+N_c-1}^*$, is held constant until the end of the prediction horizon N_p. Only the first optimal control input calculated in the control sequence is implemented on the plant. Then the process starts over. The objective is to get the predicted output $\boldsymbol{\eta}_{plan}(t+i)$ to avoid the obstacles and reduce the planning error of the predicted trajectory $\boldsymbol{\eta}_{plan}(t+i)$ (re-planned trajectory) and global reference trajectory $\boldsymbol{\eta}_{ref}(t+i)$ by utilising the proposed inputs $\boldsymbol{u}_{t+i}^*$ during the prediction horizon N_p.

The re-planned trajectory $\boldsymbol{\eta}_{plan}(t+i)$ will be sent to the tracking controller (designed in Section 3.2) and the relationship between the trajectory re-planning controller and the AUV tracking control is: the trajectory re-planning controller guides the AUV to avoid obstacles during the tracking control process. After avoiding obstacles, the AUV can revert to following the global reference trajectory, until it reaches the goal point.

4. SIMULATION RESULTS.

In order to verify the effectiveness of trajectory tracking and re-planning with the MPC algorithm, simulation results on MATLAB are outlined in this Section. The simulation conditions consist of different kinds of trajectories and obstacles. We compare the algorithm's performance with the backstepping tracking control method, observe and record the results.

4.1. *Circular re-planning and tracking.*

The original circle trajectory is defined as follows:

$$\begin{cases} x_d = 10 * \sin(0{\cdot}5 * t) \\ y_d = -10 * \cos(0{\cdot}5 * t) \\ \psi_d = 0{\cdot}5 * t \end{cases}$$

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

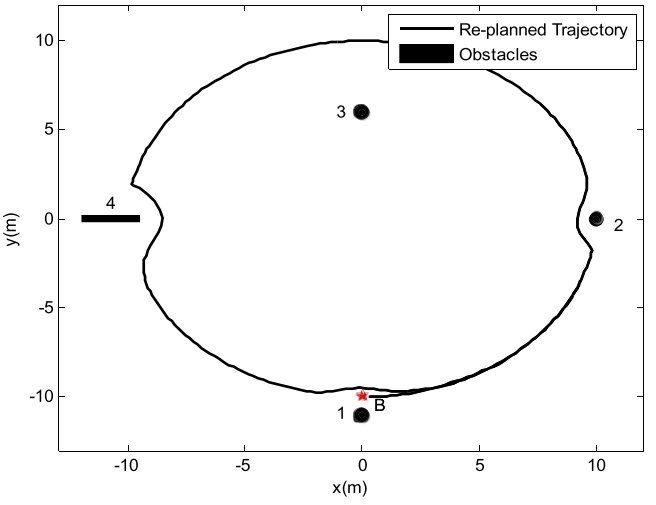

Figure 7. Re-planned trajectory.

We put linear and point obstacles into this circle path, and set the simulation condition. For normal situations, the actual initial position of the AUV is not on the reference trajectory. Here the initial reference position is $\boldsymbol{\eta}_d = [\,x_d\ y_d\ \psi_d\,]^T = [\,0\ -10\ 0\,]^T$, and the actual initial state of the AUV is set as $\boldsymbol{\eta} = [\,0\ -5\ \pi/4\,]^T$. There are 300 sampling points on the reference trajectory, and the sampling time $T = 0.05s$, prediction horizon $N_p = 9$ and control horizon $N_c = 2$. The linear speed limit is $-8\,m/s \leq u_{linear} \leq 8\,m/s$ and yaw speed limit is $-1\,rad/s \leq r \leq 1\,rad/s$. The weighting coefficient is $\boldsymbol{S}_{obs} = 200$. The backstepping control parameter is $k_1 = k_2 = 12$.

Based on the situations above, the MPC trajectory re-planning controller produces a new reference trajectory to navigate the AUV to avoid obstacles when tracking the trajectory. This new reference trajectory is called the re-planned trajectory in this paper, which is shown in Figure 7. Obstacle 1 is close to the original trajectory, obstacles 2 and 4 are on it, and obstacle 3 is far away from it. It is clear in Figure 7 that the re-planned trajectory successfully avoids the obstacles and maintains a safe distance. So, the MPC re-planning controller is working. The tracking controller makes the AUV follow the re-planned trajectory, and the tracking results of backstepping and MPC control are shown in Figure 8. The AUV resumes tracking the circle reference trajectory after successfully avoiding the obstacles.

The red dotted line in Figure 8 represents backstepping tracking results, and the blue dotted line represents the MPC tracking results, while the black solid line represents the reference trajectory (re-planned trajectory). As can be seen in Figure 8, when the AUV reaches the re-planned trajectory and begins to track steadily, the three lines almost coincide and the trajectory tracking errors are close to zero. This means the tracking controllers based on both the backstepping and MPC methods ultimately allow the AUV to overcome the initial error and track the predetermined path successfully. However, the backstepping control result in Figure 8 is obtained without considering the AUV control constraints (for example, thruster and power); here it assumes that control can be realised unconditionally, and it is a simulation result, while in the actual use of backstepping tracking, the control

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

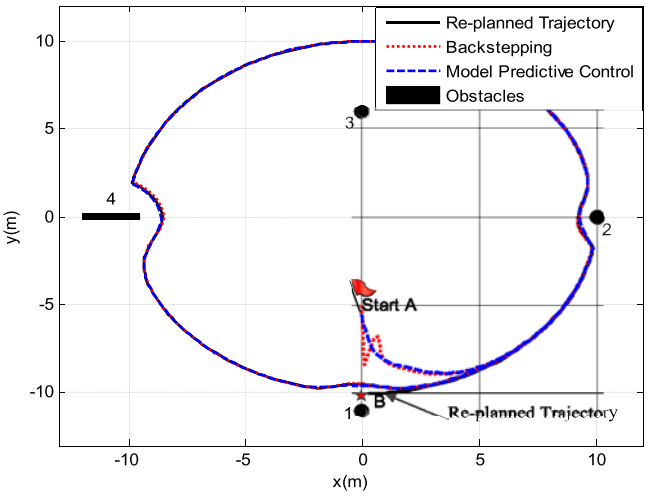

Figure 8. Re-planned trajectory tracking control.

Table 1. Normalised velocity comparison of circular tracking.

	u		v		r	
Controller	C	D	C	D	C	D
Backstepping	−5·80	2·99	−6·18	−2·28	2·95	−0·98
MPC	−0·9	−0·65	−0·88	−0·88	0·13	0·13

signal given by the simulation result here is often beyond the permissible limit. So, in reality the AUV cannot track the re-planned trajectory as calculated under the backstepping control, while the result of the MPC method is obtained with due consideration of the AUV system constraints, and has a higher practical value. More analysis can be seen below.

In order to make a comparative analysis of the simulation results more intuitive and accurate, a normalised method is adopted to process the boundary conditions of the velocities. Here the maximum velocity of the AUV is 8 m/s, and the minimum is −8 m/s. With the normalisation of AUV velocities, the acceptable normalised value for an AUV is limited to the range [−1, 1]. The normalised velocity curve and speed jump of these two controllers are shown in Figures 9 and 10 in which it is obvious that the two lines almost coincide and remain stable to the end, which means a successful tracking. However, at the beginning, the normalised velocity values of backstepping are much bigger than those of MPC. For example, the difference between these two methods is obvious at time 0·05 s and 0·1 s, corresponding to points C and D in Figure 9, and the time interval between them is 0·05 s (a sampling step).

Table 1 shows the normalised velocity at points C and D. The MPC controller takes into account the speed constraints, and the speed change is limited, so that in this small time interval, all normalised velocities of the MPC controller are within the limits [−1, 1]. What is more, the value changes slowly and the sign of the value keeps steady. However, for the backstepping tracking controller, from C to D, the linear velocity u changes from a negative

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

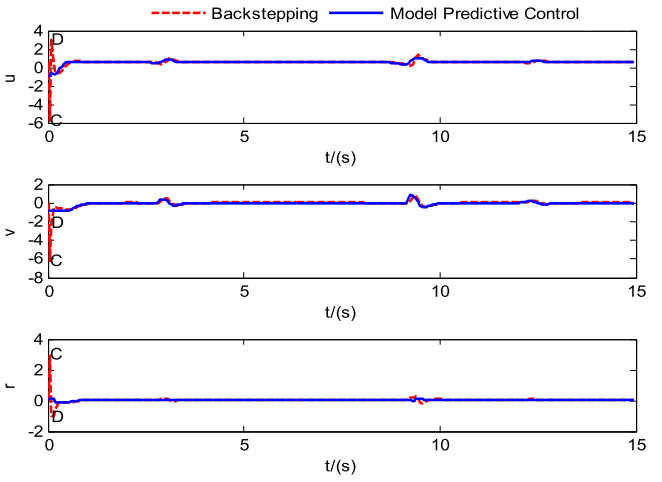

Figure 9. Comparison of normalised velocity.

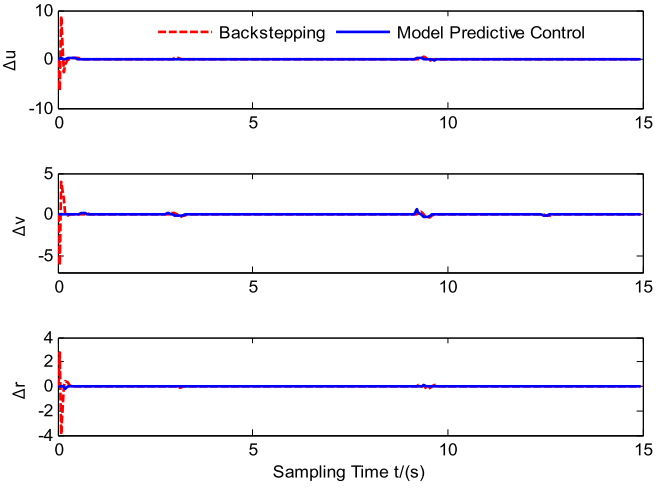

Figure 10. Comparison of normalised speed jump.

value −5·80 to a positive value 2·99, v changes from −6·18 to −2·28, and r changes from 2·95 to −0·98. They all exceed the limit [−1, 1] and their sign changes over a sampling step. Due to the hardware constraints of a real AUV, a large value change is impossible to realise. The sign of the value represents the directions of motion, and in real conditions, sign changes over a short period will easily damage the thrusters of an AUV.

Table 2 shows the maximum normalised speed jumps of the AUV at the beginning and at the obstacle's position. It can be seen in the table that at the beginning, for the backstepping algorithm, the maximum linear speed jump Δu_c and Δv_c are $8{\cdot}78 > 1$ and $-6{\cdot}18 < -1$, and the yawing speed jump Δr_c is $-3{\cdot}91 < -1$, and all exceed the range [−1, 1]. The maximum speed jumps Δu_c, Δv_c and Δr_c of the MPC algorithm are −0·9, −0·65 and −0·88 at the beginning, and clearly all of them are within the limit range [−1, 1]. Also,

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

Table 2. Maximum normalised speed jump of circular tracking.

	Beginning			Obs2			Obs4		
Controller	Δu_c	Δv_c	Δr_c	Δu_c	Δv_c	Δr_c	Δu_c	Δv_c	Δr_c
Backstepping	8·78	−6·18	−3·91	0.19	−0·25	−0·07	0·42	−0·42	−0·21
MPC	0·35	0·18	−0·25	0·11	−0·19	−0·01	0·23	0·40	0·13

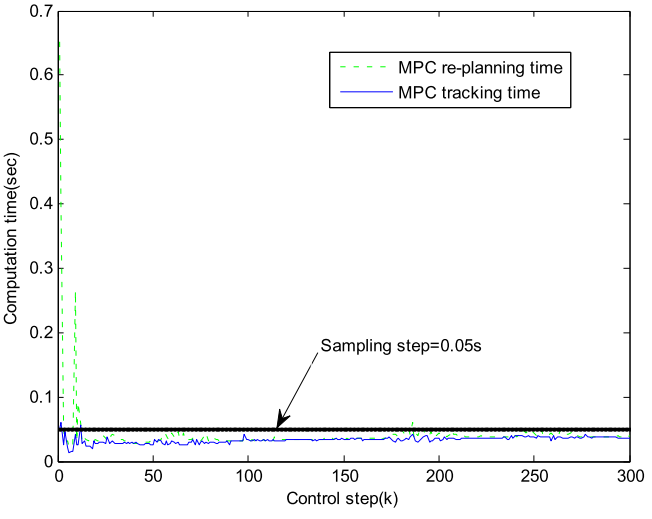

Figure 11. Computation time per controller step.

the values in Table 2 show that the speed jump of the MPC around the obstacles is smaller than the backstepping control and are within the constraint [−1, 1], meaning that trajectory tracking and obstacle avoidance are more achievable using the MPC algorithm.

Figure 11 shows the calculation times per controller step in circle trajectory tracking, and the green broken line represents the computation time of the MPC re-planning controller (minimising the deviation from the global reference trajectory when avoiding obstacles), and the blue line shows the calculation time of the MPC trajectory tracking controller. From the results it can be seen that if there is an obstacle in range or the vehicle is away from the desired trajectory, the computation time per controller step increases, and most of the controller processing is completed within one controller time step, which tells us that the proposed method is suitable for online application. It is worth mentioning here that these time measurements are highly variable, depending on factors such as the processor speed and programming environment, and are presented here for reference purposes only.

An AUV has hardware constraints and driving force limits. The condition of sharp speed jump (outside the range of [−1, 1]) and sign changes over a short period cannot be realised in actual use. So, in reality, the backstepping method cannot track the re-planned trajectory as per the simulation results. The model predictive control method has no sharp speed jump or fast sign changes because it operates within the constraints, so the MPC control process is easier to achieve in real life.

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

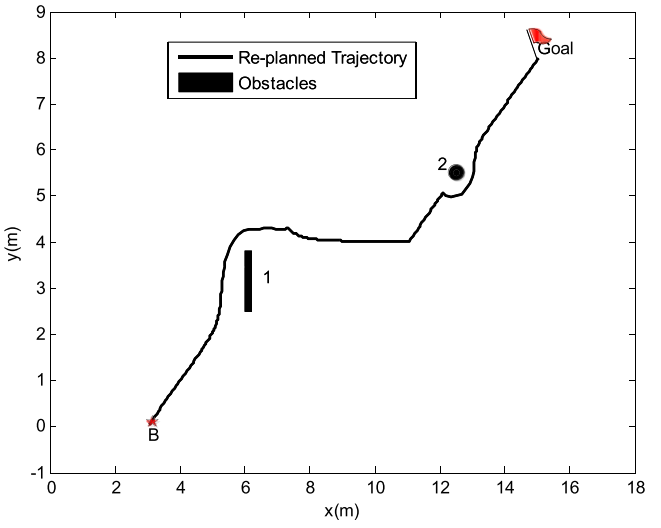

Figure 12. Re-planned trajectory.

4.2. *Polyline re-planning and tracking.* In order to verify the universality of the re-planning and tracking algorithm, a polyline test is made. The original polyline trajectory is defined as follows:

$$\text{When } 0 \le t \le 5\,s,\ \begin{cases} x_d(t) = 0{\cdot}8 * t + 3 \\ y_d(t) = 0{\cdot}8 * t \\ \psi_d(t) = \pi/4 \end{cases},$$

$$\text{When } 5\,s \le t \le 10\,s,\ \begin{cases} x_d(t) = 0{\cdot}8 * (t-1) + 3{\cdot}8 \\ y_d(t) = 4 \\ \psi_d(t) = 0 \end{cases},$$

$$\text{When } 10\,s \le t \le 15\,s,\ \begin{cases} x_d(t) = 0{\cdot}8 * (t-2) + 4{\cdot}6 \\ y_d(t) = 0{\cdot}8 * (t-2) - 2{\cdot}4 \\ \psi_d(t) = \pi/4 \end{cases}.$$

The sampling time T is 0·05s, prediction horizon $N_p = 8$, control horizon $N_c = 2$. The linear speed and yaw speed limit are $-3\,m/s \le u_{linear} \le 3\,m/s$ and $-1\,rad/s \le r \le 1\,rad/s$. There are 300 sampling points on the reference trajectory, and the backstepping control parameters are $k_1 = k_2 = 12$. The weighting coefficient is $\boldsymbol{S}_{obs} = 10$. Point and linear obstacles are put on the original polyline path. The MPC re-planning controller will then produce a re-planned trajectory to avoid the obstacles. This re-planned trajectory is shown in Figure 12. 'B' in Figure 12 is the initial reference position $\boldsymbol{\eta}_d = [\,x_d\ y_d\ \psi_d\,]^T = [\,3\ 0\ \pi/4\,]^T$ and the goal point is $\boldsymbol{\eta}_{dg} = [\,15\ 8\ \pi/4\,]^T$. It is clear in Figure 12 that the re-planned trajectory can successfully avoid the linear and point obstacles. The actual initial state of the AUV is $\boldsymbol{\eta} = [\,1\ 1\ \pi/4\,]^T$, and is not on the reference trajectory. The AUV tracks the re-planned trajectory to reach the target instead of the original circle trajectory so as to avoid the obstacles.

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

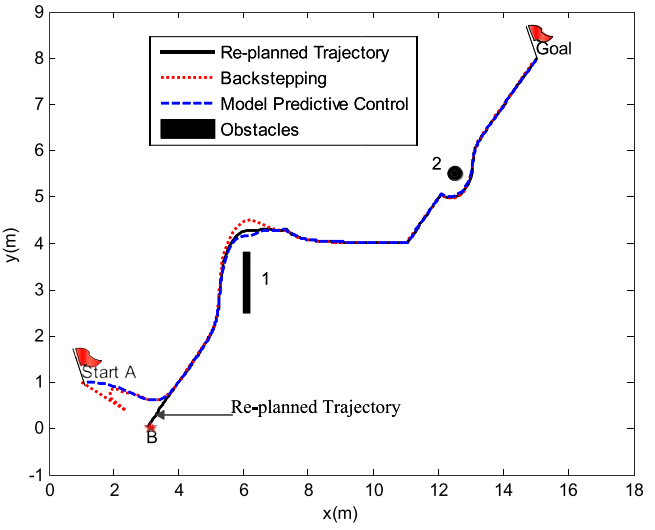

Figure 13. Re-planned trajectory tracking control.

The re-planned trajectory tracking results of backstepping and MPC tracking controllers are shown in Figure 13.

The red dotted line represents backstepping tracking results, and the blue dotted line represents the MPC tracking results, while the black solid line represents the reference trajectory (re-planned trajectory). It can be seen from Figure 13 that these two methods overcome the same initial trajectory error. After the AUV reaches the desired trajectory and begins to track stably, the three lines almost coincide. This means the tracking controllers based on both backstepping and MPC methods successfully make the AUV track the trajectory as desired.

As introduced before, Figure 13 is a simulation result and the control is thought to be realised unconditionally. Due to the real-world AUV constraints, the control data produced by the backstepping method is difficult to realise. Thus, in reality, the backstepping controller cannot track a re-planned trajectory such as the result shown in Figure 13.

In this part, the maximum velocity of the AUV is 3 m/s, and the minimum is −3 m/s. Only a normalised velocity value in the range of [−1, 1] is acceptable for the AUV. The normalised velocity curve and speed jump of these two controllers are shown in Figures 14 and 15, and it is obvious that the two normalised velocity lines almost coincide as time progresses. However, at the beginning, in Figure 14, the velocity values of the backstepping algorithm changes quickly. Table 3 shows the normalised velocity of points C and D, corresponding to times 0·05 s and 0·1 s, and the time interval between them is 0·05 s (a sampling step).

The MPC controller considers the speed constraints of the AUV, so at points C and D in Table 3, it can be seen that all normalised velocities of the MPC controller are in the range [−1, 1] and the sign of the velocity stays steady over this small time step. However from C to D, the linear velocity u of the backstepping algorithm changes from a positive value 3·65 to a negative value −2·61, v changes from −9·32 to 2·33 and yawing speed r changes from −3·86 to 2·20. It is clear that u, v and r are outside the range [−1, 1]. Also, in a

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

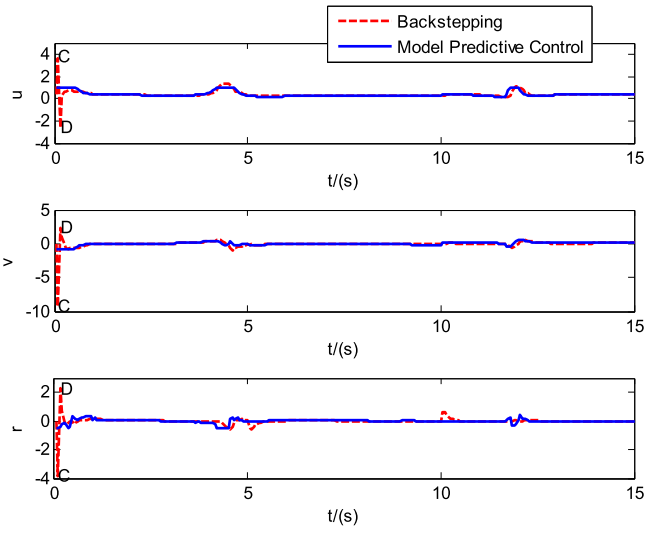

Figure 14. Comparison of normalised velocity.

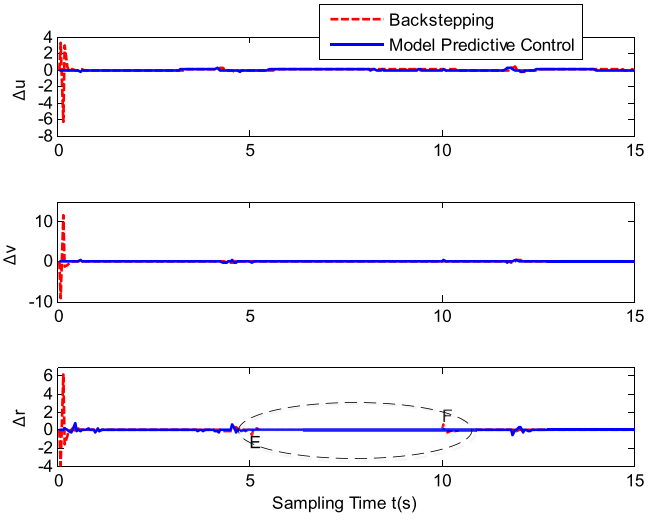

Figure 15. Comparison of normalised speed jump.

sampling step, the sign changes too quickly and the speed jump is sharp. In real conditions, as mentioned before, because of the constraints of the AUV, a big speed jump is difficult to achieve and the fast sign change should be avoided.

Since the polyline is not a continuous trajectory, it is not differentiable at the turning point. The main problem of trajectory tracking is the speed jump caused by the initial position and the error of the turning point position. Table 4 shows the maximum normalised speed jump of the AUV at the beginning, the obstacle positions and the turning point. At the beginning, for the backstepping algorithm, the linear speed jump Δu_c and Δv_c are $-6{\cdot}26 < -1$ and $11{\cdot}64 > 1$, the yawing speed jump is $6{\cdot}60 > 1$, and they all exceed

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

Table 3. The normalised velocity comparison of polyline tracking.

	u		v		r	
Controller	C	D	C	D	C	D
Backstepping	3·65	−2·61	−9·32	2·33	−3·86	2·20
MPC	1·00	1·00	−0·83	−0·83	−0·50	−0·45

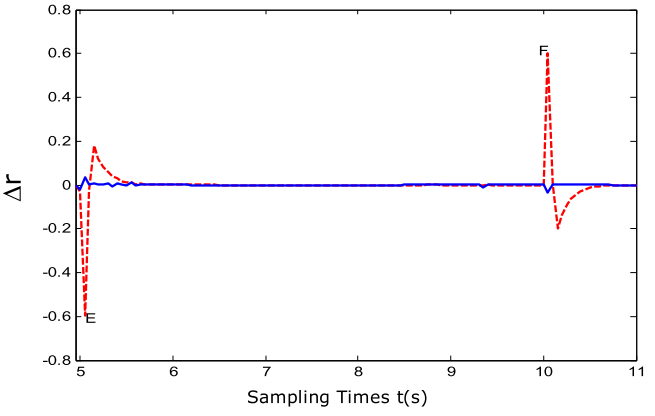

Figure 16. Enlarged view of angular speed jump.

Table 4. The maximum normalised speed jump of polyline tracking.

										Turning point	
	Beginning			Obs1			Obs2			Δr_c	
Controller	Δu_c	Δv_c	Δr_c	Δu_c	Δv_c	Δr_c	Δu_c	Δv_c	Δr_c	E	F
Backstepping	−6·25	11·64	6·06	0·26	0·29	−0·60	0·35	0·33	0·60	−0·60	0·60
MPC	0·16	0·20	0·79	0·20	0·45	0·60	0·29	−0·33	−0·52	0·04	0·034

the limit [−1, 1]. The normalised value of the linear and yawing velocities of the MPC algorithm at the beginning are small and all inside the limit [−1, 1]. At the turning point, the main speed jump is caused by the angular velocity and the linear speed jump is not obvious. So here, the angular speed jump is the focus. Points E and F in Figure 15 represent the angular speed jump of the backstepping method (with enlarged view in Figure 16). The normalised value at points E and F are −0·6 and 0·6, which are shown in Table 4. At the same place, the angular speed jumps of the MPC controller are 0·04 and −0·032. The speed jump of backstepping control is almost 15 times bigger than the MPC control method. When the AUV encounters the obstacles, the maximum speed jump is in the acceptable range [−1, 1], which means that with this method trajectory tracking and avoidance motion can be realised.

Figure 17 shows the calculation times per controller step in polyline trajectory tracking. Similar to the computation time analysis of circle tracking mentioned above, if there is an

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

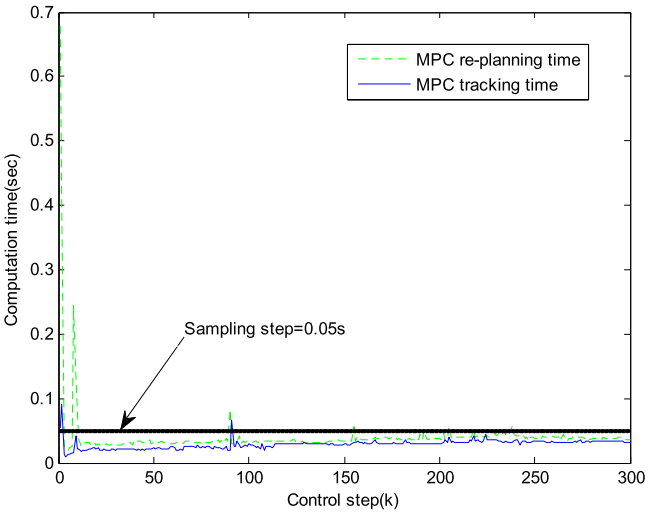

Figure 17. Computation time per controller step.

obstacle in range or the vehicle is away from the desired trajectory, the computation time per controller step increases, and most of the controller processing is completed within one controller time step, showing that the proposed method is suitable for online application. Notice that computation time is presented here for reference purposes only, because these time measurements are highly variable depending on factors such as the processor speed and programming environment.

As introduced in Section 4.1, sharp speed jumps and fast sign changes will not work in reality. So, the backstepping method cannot track a re-planned trajectory such as in our simulation results. The MPC algorithm, because of its own constraints, is more achievable in real life.

From the two simulation results above, it can be seen that the MPC re-planning method can successfully plan a trajectory to avoid the obstacles. Model predictive control can accomplish a certain function in meeting the constraint conditions, and the trajectory tracking based on MPC can help the AUV solve the speed jump problem effectively compared to the trajectory tracking of the backstepping algorithm.

5. CONCLUSIONS. Based on the model predictive control algorithm, the trajectory re-planning and tracking problem of an autonomous underwater vehicle has been studied. In this paper, on the one hand, the reference trajectory is a re-planned one based on an MPC algorithm with obstacle avoidance, and it is not a simple human-set trajectory without considering obstacles. On the other hand, through the simulation results, it can be seen that both the backstepping controller and the MPC tracking controller can make an AUV track along the re-planned trajectory to avoid obstacles. In comparison with the backstepping algorithm, the MPC algorithm is more feasible in solving the speed jump problem in re-planned trajectory tracking control of an AUV, because of its own constraint ability. However, in order to make it easier to understand and calculate, at present only a kinematic control problem is studied in this paper. The control problem of a dynamics model will

https://doi.org/10.1017/S0373463318000668 Published online by Cambridge University Press

be studied in the following work. The study of kinematics will lay a foundation for the expansion of dynamics.

ACKNOWLEDGEMENT

This project is supported by the National Natural Science Foundation of China (U1706224, 51575336, 91748117) and the Creative Activity Plan for Science and Technology Commission of Shanghai (16550720200, 18JC1413000).

REFERENCES

Abbas, M. A., Milman, R. and Eklund, J. M. (2017). Obstacle avoidance in real time with nonlinear model predictive control of autonomous vehicles. *Canadian Journal of Electrical and Computer Engineering*, **40**, 12–22.

Bagheri, A. and Moghaddam, J. J. (2009). Simulation and tracking control based on neural-network strategy and sliding-mode control for under-water remotely operated vehicle. *Neurocomputing*, **72**, 1934–1950.

Bahadorian, M., Savkovic, B., Eaton, R. and Hesketh, T. (2012). Robust model predictive control for automated trajectory tracking of an unmanned ground vehicle. *American Control Conference*, Montreal, Canada, 4251–4256.

Barth, A., Reichhartinger, M., Wulff, K., Horn, M. and Reger, J. (2016). Certainty equivalence adaptation combined with super-twisting sliding-mode control. *International Journal of Control*, **89**(9), 1767–1776.

Bessa, W.M., Dutra, M.S. and Kreuzer, E. (2008). Depth control of remotely operated underwater vehicles using an adaptive fuzzy sliding mode controller. *Robotics and Autonomous Systems*, **56**, 670–677.

Cui, M., Sun, D., Liu, W., Zhao, M. and Liao, X. (2012). Adaptive tracking and obstacle avoidance control for mobile robots with unknown sliding. *International Journal of Advanced Robotic Systems*, **9**, 1–14.

Falcone, P. (2007). *Nonlinear model predictive control for autonomous vehicles.* Benevento: University of Sannio.

Fossen, T. I. (2002). *Marine control system: guidance, navigation, and control of ships, rigs and underwater vehicles*. Trondheim: Marine Cybernetics.

Hoseini, S. M., Farrokhi, M. and Koshkouei, A. J. (2008). Robust adaptive control of uncertain non-linear systems using neural networks. *International Journal of Control*, **81**, 1319–1330.

Jia, H., Zhang, L., Cheng, X., Bian, X., Yan, Z. and Zhou, J. (2012). Three dimensional path following control for underatuated UUV based on nonlinear iterative sliding mode. *Acta Automatica Sinica*, **38**, 308–314.

Kim, H. J., Shim, D. H. and Sastry, S. (2002). Nonlinear model predictive tracking control for rotorcraft-based unmanned serial vehicles. *Proceedings of American Control Conference*, Anchorage, USA, 3576–3581.

Li, Z., Deng, J., Lu, R., Xu, Y., Bai, J. and Su, C. (2016). Trajectory-tracking control of mobile robot systems incorporating neural-dynamic optimized model predictive approach. *IEEE Transactions on Systems, Man, and Cybernetics: System*, **46**, 740–949.

Luo, C. and Yang, S. X. (2008) A bio-inspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environment. *IEEE Transactions on Neural Networks*, **19**, 1279–1298.

Miao, B., Li, T. and Luo, W. (2013). A DSC and MLP based robust adaptive NN tracking control for underwater vehicle. *Neurocomputing*, **111**, 184–189.

Pan, C. Z., Lai, X. Z., Yang, S. X. and Wu, M. (2013). An efficient neural network approach to tracking control of an autonomous surface vehicle with unknown dynamics. *Expert Systems with Applications*, **40**, 1629–1635.

Serdar, S., Bradley, J. B. and Ron, P. P. (2008). A chattering-free sliding-mode controller for underwater vehicles with fault-tolerant infinity-norm thrust allocation. *Ocean Engineering*, **35**, 1647–1659.

Shen, C., Shi, Y. and Buckham, B. (2016). Nonlinear model predictive control for trajectory tracking of an AUV: A distributed implementation. *Conference on Decision and Control*, Las Vegas, NV, USA, 5998–6003.

Sun, B., Zhu, D. and Yang S.X. (2014). A bio-inspired filtered backstepping tracking control of 7000 m manned submarine vehicle. *IEEE Transactions on Industrial Electronics*, **61**, 3682–3693.

Sun, B., Zhu, D. and Yang, S.X. (2016). A novel tracking controller for autonomous underwater vehicles with thruster fault accommodation. *The Journal of Navigation*, **69**, 593–612.

Tsai, P. S., Wang, L. S. and Chang, F. R. (2004). Systematic backstepping design for b-spline trajectory tracking control of the mobile robot in hierarchical model. *IEEE International Conference on Networking, Sensing and Control, Taipei*, IEEE, 713–718.

Wu, Y. J., Zuo, J. X. and Sun, L. H. (2017). Adaptive terminal sliding mode control for hypersonic flight vehicles with strictly lower convex function based nonlinear disturbance observer. *ISA transactions*, **71**, 215–226.

Xiang, X, Lapierre, L. and Jouvencel, B. (2015). Smooth transition of AUV motion control: from fully-actuated to under-actuated configuration. *Robotics and Autonomous Systems*, **67**, 14–22.

Xiang, X., Yu, C. and Zhang. Q. (2017). On intelligent risk analysis and critical decision of underwater robotic vehicle. *Ocean Engineering*, **140**, 453–465, Aug. 2017.

Xiang, X., Yu, C., Lapierre, L., Zhang, J. and Zhang, Q. (2018). Survey on fuzzy-logic-based guidance and control of marine surface vehicles and underwater vehicles. *International Journal of Fuzzy Systems*, **20**, 572–586.

Yang, S. X. and Luo, C. (2004). A neural network approach to complete coverage path planning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, **34**, 718–724.

Ye, J. (2014). Tracking control of a non-holonomic wheeled mobile robot using improved compound cosine function neural networks. *International Journal of Control*, **88**, 364–373.

Yoon, Y., Shin, J., Kim, H. J., Park, Y. and Sastry, S. (2009). Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. *Control Engineering Practice*, **17**, 741–750.

Zhu, Z., Cao, X., Sun, B. and Luo, C. (2018) Biologically Inspired Self-Organizing Map Applied to Task Assignment and Path planning of an AUV System. *IEEE Transactions on Cognitive and Developmental Systems*, **10**, 304–313.