Journal of Marine Science and Technology

Volume 25 | Issue 6 Article 7

CASCADED UUV TRAJECTORY TRACKING CONTROL BASED ON MODEL PREDICTIVE AND SLIDING MODE CONTROL

Bing Sun
Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Haigang Avenue 1550, Shanghai, 201306, China.

Wenyang Gan
Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Haigang Avenue 1550, Shanghai, 201306, China.

Man Mei
Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Haigang Avenue 1550, Shanghai, 201306, China.

Daqi Zhu
Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Haigang Avenue 1550, Shanghai, 201306, China., zdq367@aliyun.com

Simon X Yang
Advanced Robotics and Intelligent Systems Laboratory, University of Guelph, Guelph, ON. N1G2W1, Canada.

CASCADED UUV TRAJECTORY TRACKING CONTROL BASED ON MODEL PREDICTIVE AND SLIDING MODE CONTROL

Acknowledgements

This project was supported by the National Natural Science Foundation of China (61503239, U170620065), the National Key Project of the Research and Development Program (2017YFC0306302), and the Creative Activity Plan for the Science and Technology Commission of Shanghai (15550722400). This manuscript was edited by Wallace Academic Editing.

Journal of Marine Science and Technology, Vol. 25, No. 6, pp. 671-679 (2017)
DOI: 10.6119/JMST-017-1226-07

CASCADED UUV TRAJECTORY TRACKING CONTROL BASED ON MODEL PREDICTIVE AND SLIDING MODE CONTROL

Bing Sun[1], Wenyang Gan[1], Man Mei[1], Daqi Zhu[1], and Simon X Yang[2]

Key words: unmanned underwater vehicle, trajectory tracking, model predictive control, sliding mode control.

ABSTRACT

This paper describes a cascaded dynamic trajectory tracking control approach based on model predictive control (MPC) and sliding mode control for unmanned underwater vehicles (UUVs). The proposed method combines kinematic and dynamic controllers to achieve robust control. First, MPC is employed to realize tracking control kinematically. The MPC algorithm is proposed as a solution to the speed jump problem. Considering the computational burden for nonlinear UUV dynamic models, the MPC method is used in conjunction with sliding mode control to achieve dynamic tracking control, which solves problems of modeling uncertainty and external disturbances. Experimental results showed that the proposed method could solve the speed jump and thruster saturation problems more effectively than other methods could.

I. INTRODUCTION

Unmanned underwater vehicles (UUVs) are increasing in use in fields of marine science, search and rescue, submarine pipeline tracking, submarine cable maintenance, and national defense (Li, 2015; Xiang, 2017). Trajectory tracking control is a critical aspect of UUV research. Trajectory tracking control refers to the UUV's ability to track a reference trajectory in an inertial coordinate system. From an initial state given by the designed control laws, it achieves global uniform asymptotic stability through position error tracking. Because of strong coupling, the nonlinear characteristics of underwater vehicles, and the complex underwater environment, UUV trajectory tracking control is a challenging area of research. The prevailing methods of UUV trajectory tracking control include proportional-integral-derivative (PID) control (Wang, 2009; De Paula, 2012), sliding mode control (SMC; Wallace, 2008; Khadija, 2012; Xu, 2015), backstepping control, fuzzy logic (Wai, 2007; Sun, 2017), and neural network control (Luo, 2008; Bagheri, 2010; Zhu, 2017).

This paper proposes a model predictive control (MPC) method and focuses on solving the speed jump problem encountered in UUV tracking. MPC is a model-based closed-loop control optimization control strategy, and the core of the algorithm is a dynamic model that can predict future outputs with repeated online optimization calculations and implement corrections based on feedback regarding model error.

MPC is effective and robust. The proposed tracking control algorithm can overcome the uncertainties of the UUV model (nonlinear and parallel resistance) and can readily manage various constraints placed on process control variables and manipulated variables. Speed constraints are specified in the control design to solve speed jump issues.

Some papers have addressed the speed jump problem of UUV tracking control. The most successful of these has been bioinspired algorithms (Fossen, 2011; Pan, 2015; Cao, 2016). Bioinspired neurodynamic models can reduce the speed jump problem caused by tracking error because of their bounded smooth characteristics. However, these models only have a smoothing effect, and are unable to keep the speed within the constraints as the MPC algorithm does.

Although the MPC method alleviates the tracking control problem, the issues of computation burden and complexity must be accounted for, especially when for the nonlinear UUV dynamic model. To solve this problem, we propose a cascaded control design, including kinematic dynamic control. The MPC algorithm is used to generate a bounded velocity control signal. In the dynamic part, the sliding mode method is selected for dynamic tracking control because of its insensitivity to parameter variations and ability to reject disturbances.

In section 2, the background formulation of UUV tracking control is given. In section 3, a cascaded MPC-SMC dynamic controller is presented. The results of simulation experiments for various situations are presented in section 4. Finally, our conclusions are stated in section 5.

Paper submitted 06/14/17; revised 08/22/17; accepted 12/13/17. Author for correspondence: Daqi Zhu (e-mail: zdq367@aliyun.com).
[1] Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University, Haigang Avenue 1550, Shanghai, 201306, China.
[2] Advanced Robotics and Intelligent Systems Laboratory, University of Guelph, Guelph, ON. N1G2W1, Canada.

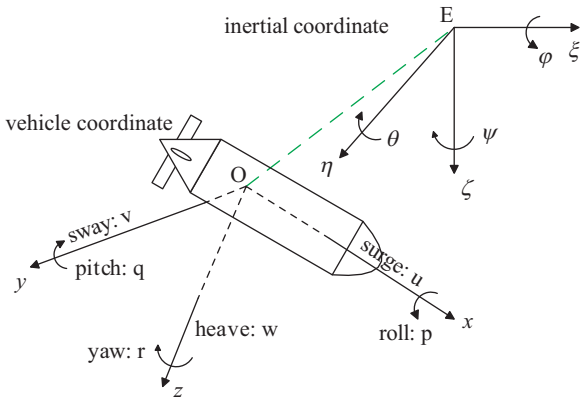

Fig. 1. Coordinate system.

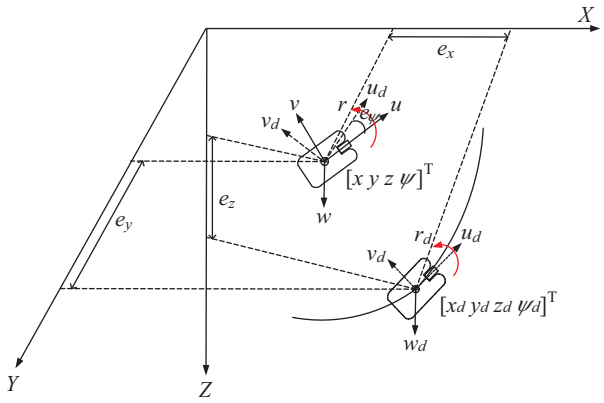

Fig. 2. Three-dimensional tracking control formulation.

II. BACKGROUND FORMULATION OF UUV TRACKING CONTROL

To study the kinematic and dynamic characteristics of UUVs, a UUV coordinate system is established first. This consists of two parts: an inertial coordinate system and a vehicle coordinate system. The inertial coordinate system (also called the earth coordinate system) $E-\xi\eta\zeta$ has its origin at a certain point on the earth. The vehicle coordinate system (also called the motion coordinate system) $O-xyz$ is fixed on the UUV and moves with it as shown in Fig. 1.

The posture vector of a UUV is given by $\boldsymbol{\eta}=[x\ y\ z\ \phi\ \theta\ \psi]^T$ in the inertial coordinate system, and the velocity vector is given by $\boldsymbol{\nu}=[u\ v\ w\ p\ q\ r]^T$ in the vehicle coordinate system. The transformation between the posture vector and velocity vector is represented by $\boldsymbol{J}(\boldsymbol{\eta})$, which is called the Jacobi transform matrix. The kinematic equation with six degrees of freedom (DOF) (Fossen, 2011) is

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

The dynamic equation with six DOF in the body-fixed frame can be presented as

$$\boldsymbol{M}\dot{\boldsymbol{\nu}}+\boldsymbol{C}(\boldsymbol{\nu})\boldsymbol{\nu}+\boldsymbol{D}(\boldsymbol{\nu})\boldsymbol{\nu}+\boldsymbol{g}(\boldsymbol{\eta})=\boldsymbol{\tau} \tag{2}$$

Under normal circumstances, all six DOF of motion are not fully considered. The rolling and pitching motions are often overlooked (i.e., $p=q=0$); therefore, the tracking control problem is commonly solved with four DOF: surge, sway, heave, and yaw (u, v, w, r). Their relationships are as follows:

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

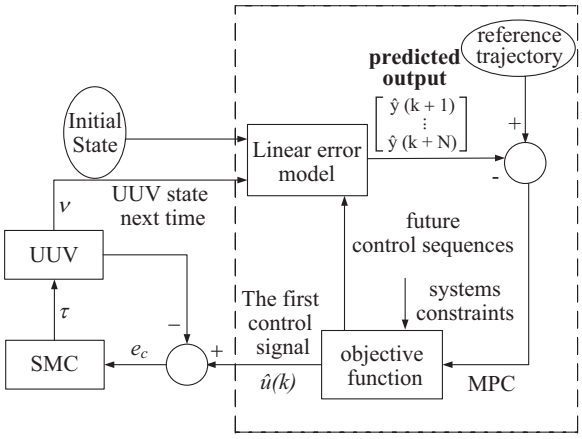

Fig. 3. Proposed cascaded MPC-SMC dynamic controller.

The simplified dynamic model in this paper is given as follows:

$$\begin{aligned}
&(m-X_{\dot{u}})\dot{u}+X_u u+X_{uu}u|u|=\tau_X\\
&(m-Y_{\dot{v}})\dot{v}+Y_v v+Y_{vv}v|v|=\tau_Y\\
&(m-Z_{\dot{w}})\dot{w}+Z_w w+Z_{ww}w|w|=\tau_Z\\
&(I_z-N_{\dot{r}})\dot{r}+N_r r+N_{rr}r|r|=\tau_N
\end{aligned} \tag{4}$$

Trajectory tracking control of UUVs involves controlling the actual speed so that the UUV can track the desired trajectory, and ultimately, the error between the actual trajectory and the desired trajectory converges to zero. A detailed model of tracking control is shown in Fig. 2.

III CASCADED MPC-SMC DYNAMIC CONTROLLER

The basic control architecture of the tracking control system is illustrated in Fig. 3. The cascaded control strategy consists of two parts: (1) an outer loop kinematic controller with MPC; (2) an inner loop dynamic controller with SMC. If the kinematic

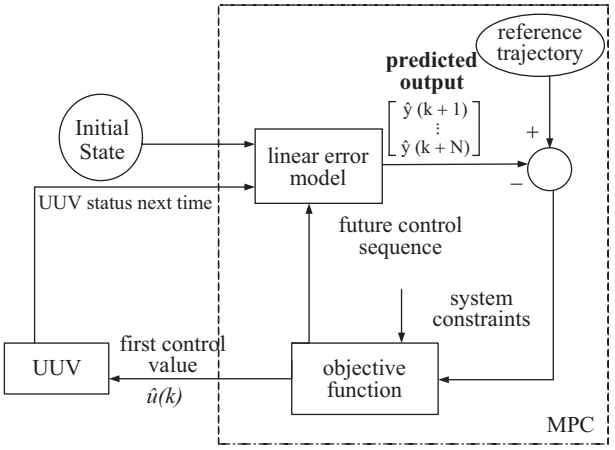

Fig. 4. MPC algorithm flowchart.

model is considered, the tracking control design is usually considered complete. However, kinematic control is assumed to have "perfect velocity tracking," an assumption that is unrealistic in practice. Therefore, dynamic control is also implemented. The forces and moments driving a given reference vehicle are then derived from the velocity control input. To address parametric uncertainties and unforeseen environmental disturbances, an adaptive SMC technique is proposed.

1. Kinematic MPC Controller

This section presents the kinematic controller based on MPC. MPC generally has three implementation components: a prediction model, rolling optimization, and feedback correction (Su, 2013). The MPC algorithm used in the tracking control process is shown in Fig. 4. The design of the objective function considers both the efficiency and stability of tracking. Yu (2015) proved the convergence of the MPC algorithm for nonlinear systems.

1) UUV Error Model Building

As the UUV kinematic model illustrates, the system can be understood as a control system with input ν and state η where its general form is

$$\dot{\boldsymbol{\eta}} = f(\boldsymbol{\eta}, \boldsymbol{\nu}) \tag{5}$$

For a given reference trajectory, each point satisfies the preceding kinematic equation. With the subscript d representing the reference, the general form is

$$\dot{\boldsymbol{\eta}}_d = f\left(\boldsymbol{\eta}_d, \boldsymbol{\nu}_d\right) \tag{6}$$

where

$$\boldsymbol{\eta}_d = \begin{bmatrix} x_d & y_d & z_d & \psi_d \end{bmatrix}^T \text{ and } \boldsymbol{\nu}_d = \begin{bmatrix} u_d & v_d & w_d & r_d \end{bmatrix}^T.$$

Using Taylor series expansion and ignoring the higher order terms in formula (5) at the reference trajectory point, formula (5) can be written as

$$\begin{aligned} \dot{\boldsymbol{\eta}} = f(\boldsymbol{\eta}_d, \boldsymbol{\nu}_d) &+ \left.\frac{\partial f(\boldsymbol{\eta}, \boldsymbol{\nu})}{\partial \boldsymbol{\eta}}\right|_{\substack{\boldsymbol{\eta}=\boldsymbol{\eta}_d \\ \boldsymbol{\nu}=\boldsymbol{\nu}_d}} (\boldsymbol{\eta} - \boldsymbol{\eta}_d) \\ &+ \left.\frac{\partial f(\boldsymbol{\eta}, \boldsymbol{\nu})}{\partial \boldsymbol{\nu}}\right|_{\substack{\boldsymbol{\eta}=\boldsymbol{\eta}_d \\ \boldsymbol{\nu}=\boldsymbol{\nu}_d}} (\boldsymbol{\nu} - \boldsymbol{\nu}_d) \end{aligned} \tag{7}$$

The UUV error model can be obtained by subtracting formula (6) from formula (7):

$$\begin{aligned} \dot{\tilde{\boldsymbol{\eta}}} &= \begin{bmatrix} \dot{x} - \dot{x}_d \\ \dot{y} - \dot{y}_d \\ \dot{z} - \dot{z}_d \\ \dot{\psi} - \dot{\psi}_d \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & -u\sin\psi - v\sin\psi \\ 0 & 0 & 0 & u\cos\psi - v\cos\psi \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} x - x_d \\ y - y_d \\ z - z_d \\ \psi - \psi_d \end{bmatrix} \\ &+ \begin{bmatrix} \cos\psi & -\sin\psi & 0 & 0 \\ \sin\psi & \cos\psi & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u - u_d \\ v - v_d \\ w - w_d \\ r - r_d \end{bmatrix} \end{aligned} \tag{8}$$

Formula (8) is the UUV linear error model. To apply this model to MPC controller design, formula (8) must be discretized. It can then be derived as

$$\tilde{\boldsymbol{\eta}}(k+1) = \boldsymbol{A}_{k,t}\tilde{\boldsymbol{\eta}}(k) + \boldsymbol{B}_{k,t}\tilde{\boldsymbol{\nu}}(k) \tag{9}$$

where

$$\boldsymbol{A}_{k,t} = \begin{bmatrix} 1 & 0 & 0 & (-u\sin\psi - v\sin\psi)T \\ 0 & 1 & 0 & (u\cos\psi - v\cos\psi)T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$\boldsymbol{B}_{k,t} = \begin{bmatrix} T\cos\psi & -T\sin\psi & 0 & 0 \\ T\sin\psi & T\cos\psi & 0 & 0 \\ 0 & 0 & T & 0 \\ 0 & 0 & 0 & T \end{bmatrix},$$

and T is sampling time.

2) Optimization Problems of MPC Based on Quadratic Programming

As stated previously, MPC is a model-based control algorithm. A prediction model provides *a priori* knowledge to predict control optimization and to determine which control input sequence

should be adopted. Output changes of the controlled object can be consistent with the intended target in the future. The whole algorithm can be understood as the solution of an optimization problem that entails finding the optimal control law by minimizing the objective function in the control domain.

System state deviation and control optimization must be added so that the objective function is able to ensure that the UUV can track the desired trajectory efficiently. Kuhne (2004) used the following form of the objective function when designing a tracking controller:

$$\begin{aligned} J(k) = &\sum_{j=1}^{N} \tilde{\boldsymbol{\eta}}^T (k+j|k) \boldsymbol{Q} \tilde{\boldsymbol{\eta}}(k+j) \\ &+ \tilde{\boldsymbol{\nu}}^T (k+j-1) \boldsymbol{R} \tilde{\boldsymbol{\nu}}(k+j-1) \end{aligned} \tag{10}$$

where $\boldsymbol{Q}$ and $\boldsymbol{R}$ are weight matrices.

This can be converted to a quadratic programming problem, which is a typical optimization problem. In formula (10), controlling values can be seen as the state values in the objective function. The structure is simple and easy to implement. However, it is impossible to restrain the control increment accurately, which means the speed jump phenomenon cannot be avoided. This affects the continuity of controlling values. Therefore, the control increment is adopted as the state of the objective function and takes the following form:

$$\begin{aligned} J(k) = &\sum_{i=1}^{N_p} \left\| \boldsymbol{\eta}(k+i|t) - \boldsymbol{\eta}_{ref}(k+i|t) \right\|_{\boldsymbol{Q}}^2 \\ &+ \sum_{i=1}^{N_c-1} \left\| \Delta \boldsymbol{\nu}(k+i|t) \right\|_{\boldsymbol{R}}^2 \end{aligned} \tag{11}$$

where N_p is the predictive domain, and N_c is the control domain. N_p reflects the system's ability to follow the reference trajectory, and N_c reflects the smooth change requirements of the controlling values. This expression is designed so that the system tracks the desired trajectory smoothly and quickly. Furthermore, in the actual control system, the system constraints must be satisfied as follows:

Control constraint:

$$\begin{aligned} &\boldsymbol{\nu}_{\min}(t+k) \le \boldsymbol{\nu}(t+k) \le \boldsymbol{\nu}_{\max}(t+k), \\ &k = 0, 1, \ldots N_c - 1 \end{aligned} \tag{12}$$

Control increment constraint:

$$\begin{aligned} &\Delta \boldsymbol{\nu}_{\min}(t+k) \le \Delta \boldsymbol{\nu}(t+k) \le \Delta \boldsymbol{\nu}_{\max}(t+k), \\ &k = 0, 1, \ldots N_c - 1 \end{aligned} \tag{13}$$

In the objective function, it is necessary to calculate the system output for some time in the future. The following change is made to formula (9):

$$\boldsymbol{\xi}(k|t) = \begin{bmatrix} \tilde{\boldsymbol{\eta}}(k|t) \\ \tilde{\boldsymbol{\nu}}(k-1|t) \end{bmatrix} \tag{14}$$

Then, a new state space can be expressed as

$$\boldsymbol{\xi}(k+1|t) = \tilde{\boldsymbol{A}}_{k,t} \boldsymbol{\xi}(k|t) + \tilde{\boldsymbol{B}}_{k,t} \Delta \boldsymbol{V}(k|t) \tag{15}$$

$$\boldsymbol{\eta}(k|t) = \tilde{\boldsymbol{C}}_{k,t} \boldsymbol{\xi}(k|t) \tag{16}$$

where $\tilde{\boldsymbol{A}}_{k,t} = \begin{bmatrix} \boldsymbol{A}_{k,t} & \boldsymbol{B}_{k,t} \\ \boldsymbol{0}_{m \times n} & \boldsymbol{I}_m \end{bmatrix}$, $\tilde{\boldsymbol{B}}_{k,t} = \begin{bmatrix} \boldsymbol{B}_{k,t} \\ \boldsymbol{I}_m \end{bmatrix}$, n is the state dimension, and m is the control dimension.

After derivation, the predicted system output can be obtained as follows:

$$\boldsymbol{Y}(t) = \Psi_t \boldsymbol{\xi}(t|t) + \Theta_t \Delta \boldsymbol{V}(t) \tag{17}$$

For more detailed information, refer to Kuhne (2004).

After inserting formula (17) into formula (11), the full expression of the objective function can be written as

$$J(\boldsymbol{\xi}(t), \boldsymbol{\nu}(t-1), \Delta \boldsymbol{V}(t)) = \Delta \boldsymbol{V}(t) \boldsymbol{H}_t \Delta \boldsymbol{V}(t)^T + \boldsymbol{G}_t \Delta \boldsymbol{V}(t)^T \tag{18}$$

where

$$\boldsymbol{H}_t = \begin{bmatrix} \Theta_t^T \boldsymbol{Q} \Theta_t + \boldsymbol{R} & \boldsymbol{0} \end{bmatrix}, \text{ and } \boldsymbol{G}_t = \begin{bmatrix} 2\boldsymbol{E}(t)^T \boldsymbol{Q} \Theta_t & \boldsymbol{0} \end{bmatrix}.$$

Therefore, solving optimization problems under the constraints of MPC in every step is equivalent to solving quadratic programming problems as follows:

$$\min_{\Delta \boldsymbol{V}(t)} \Delta \boldsymbol{V}(t) \boldsymbol{H}_t \Delta \boldsymbol{V}(t)^T + \boldsymbol{G}_t \Delta \boldsymbol{V}(t)^T \tag{19}$$

$$\text{s.t. } \boldsymbol{V}_{\min} \le \boldsymbol{\nu}(t-1) + \sum_{i=t}^{k} \Delta \boldsymbol{V}(i) \le \boldsymbol{V}_{\max}, \; k = t, \ldots, t+\boldsymbol{H}_c - 1$$

$$\Delta \boldsymbol{V}_{\min} \le \Delta \boldsymbol{V}(k) \le \Delta \boldsymbol{V}_{\max}, \; k = t, \ldots, t+\boldsymbol{H}_c - 1$$

After solving Eq. (19) for each control cycle, a series of control input increments can be obtained in the control domain: $\Delta \boldsymbol{V}_t^* = \begin{bmatrix} \Delta \boldsymbol{\nu}_t^* & \Delta \boldsymbol{\nu}_{t+1}^* & \cdots & \Delta \boldsymbol{\nu}_{t+N_c-1}^* \end{bmatrix}^T$. According to the basic principles of MPC, the first element of the control sequence is selected as the actual control input increment that acts on the system, given by $\boldsymbol{\nu}(t) = \boldsymbol{\nu}(t-1) + \Delta \boldsymbol{\nu}_t^*$. The system uses this controlling value until the next time. At the next time, the system predicts the output with the next time domain, again according to the state information, and obtains a new sequence control

increment through optimization. This procedure repeats until the system completes the control process.

In this paper, the speed control constraints are set according to the actual position of the UUV as follows:

2. Adaptive Sliding Mode Dynamic Controller

For an actual tracking control process $\boldsymbol{e}(t)=\boldsymbol{\eta}_d(t)-\boldsymbol{\eta}(t)\to 0$, the dynamic model must be applied to calculate the actual speed control and torque values to track the desired speed values $\boldsymbol{\nu}\to\boldsymbol{\nu}_c$. Because of modeling uncertainty and external disturbance factors, it is difficult to ensure stable speed tracking in the initial phase. Therefore, a robust sliding mode dynamic controller was designed for speed control. The control signal generated by speed error acts on the UUV to regulate the surge, sway, heave, and yaw in order to achieve stable speed tracking. The control system is illustrated in detail in Fig. 3.

After the velocity controller generates the auxiliary UUV velocity, a sliding mode controller is used to generate the control forces and moments $\boldsymbol{\tau}$. The auxiliary velocity error is defined as

$$\boldsymbol{e}_c=\boldsymbol{\nu}_c-\boldsymbol{\nu} \tag{20}$$

Then, the sliding manifold is defined as

$$\boldsymbol{s}=\dot{\boldsymbol{e}}_c+2\Lambda\boldsymbol{e}_c+\Lambda^2\int\boldsymbol{e}_c \tag{21}$$

Considering that the UUV dynamic model is not fully known, the UUV dynamic equation can be substituted with an estimated dynamics term and an unknown dynamics term:

$$\boldsymbol{\tau}=\hat{\boldsymbol{\tau}}+\tilde{\boldsymbol{\tau}} \tag{22}$$

Derive formula (21) and let it equal zero. Then, the equivalent control law can be expressed as

$$\boldsymbol{\tau}_{eq}=\hat{\boldsymbol{M}}(\dot{\boldsymbol{\nu}}_c+\frac{\ddot{\boldsymbol{e}}_c}{2\Lambda}+\frac{\Lambda}{2}\boldsymbol{e}_c)+\hat{\boldsymbol{C}}\boldsymbol{q}+\hat{\boldsymbol{D}}\boldsymbol{q}+\hat{\boldsymbol{g}} \tag{23}$$

To eliminate the chattering problem caused by the discontinuous term, an adaptive term is added in the control law to replace the switching term based on previous work (Sun, 2014):

$$\boldsymbol{\tau}_{ad}=\tilde{\boldsymbol{\tau}}_{est}+(K+\frac{\hat{\boldsymbol{C}}}{2\Lambda})\boldsymbol{s} \tag{24}$$

where $\tilde{\boldsymbol{\tau}}_{est}$ is an adaptive term that estimates the lumped uncertainty vector $\tilde{\boldsymbol{\tau}}$ defined in formula (22). The estimation of the lumped uncertainty vector is proposed as follows:

$$\dot{\tilde{\boldsymbol{\tau}}}_{est}=\Gamma\boldsymbol{s} \tag{25}$$

When the discontinuous term is replaced by an adaptive term, the problem becomes a robust control problem. This adaptive term relates the error metric to the dynamic uncertainties, and acts on the controller such that the estimated dynamics more closely reflect the actual dynamics.

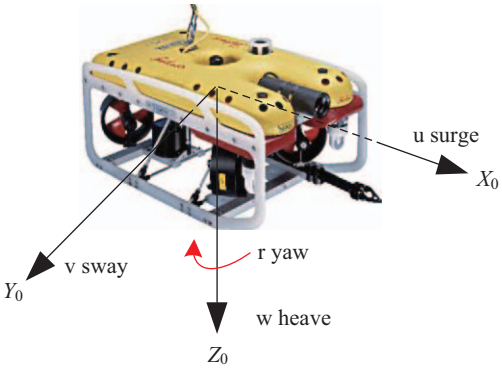

Fig. 5. Structure of FALCON UUV.

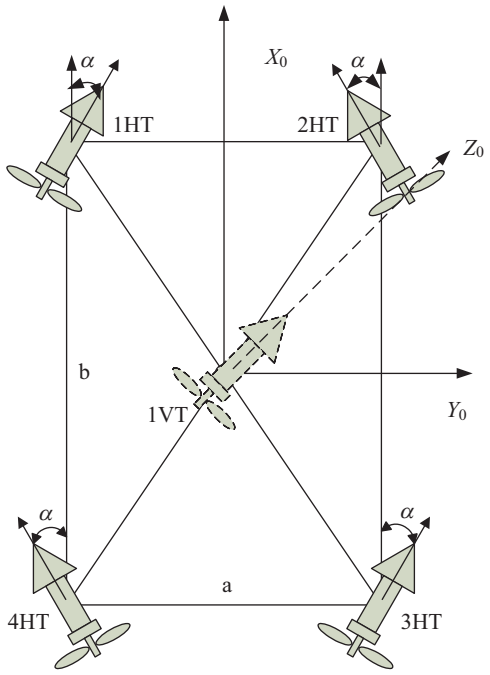

Fig. 6. Thruster distribution.

The total control law can be defined as follows:

$$\boldsymbol{\tau}=\boldsymbol{\tau}_{eq}+\boldsymbol{\tau}_{ad}=\boldsymbol{\tau}_{eq}+\tilde{\boldsymbol{\tau}}_{est}+(K+\frac{\hat{\boldsymbol{C}}}{2\Lambda})\boldsymbol{s} \tag{26}$$

IV. SIMULATION RESULTS AND ANALYSIS

To test the proposed algorithm, simulation study was conducted on a specific UUV model (Seaeye FALCON UUV, from the Laboratory of Underwater Vehicles and Intelligent Systems, Shanghai Maritime University). The FALCON UUV has four horizontal thrusters, denoted as $iHT(i=1, 2, 3, 4)$ and one vertical thruster $1VT$ (Fig. 6). The thruster configuration of the FALCON enables direct control of four DOF: surge, sway, heave, and yaw (Fig. 5). The four thrusters are arranged symmetrically, each with the same performance. A comparison study with the

Table 1. Velocity limits.

Velocity	V_{max}	V_{min}	Velocity Change	ΔV_{max}	ΔV_{min}
u (m/s)	3	-3	Δu (m/s)	0.5	-0.5
v (m/s)	2	-2	Δv (m/s)	0.25	-0.25
w (m/s)	2	-2	Δw (m/s)	0.25	-0.25
r (rad/s)	2	-2	Δr (rad/s)	0.25	-0.25

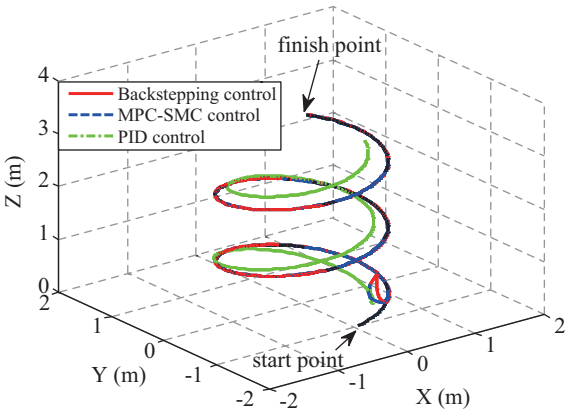

Fig. 7. Spiral line tracking results.

PID and backstepping methods was conducted to study the performance of the proposed control method.

1. Three-Dimensional Spiral Line Tracking Control

The three-dimensional spiral line trajectory was defined as follows: $x_d(t) = \sin(0.5t)$, $y_d(t) = -\cos(0.5t)$, and $z_d(t) = 0.1t$, $\psi_d(t) = 0.5t$. The initial UUV state was (1, 0, 0, 0) while the desired initial state was given as (0, -1, 0, 0). The control parameters for the PID method were set as $k_P = 100$, $k_I = 5$, and $k_D = 1$, and the parameters for the backstepping method were given as $k_1 = k_2 = k_3 = 12$. Time varied from 0 to 30 s.

Fig. 7 shows the results of spiral line trajectory tracking for the proposed MPC-SMC method, the PID method, and the backstepping method. Fig. 8 illustrates the tracking error results of the three controllers. The blue dashed line represents MPC, the red solid line represents backstepping control, and the green dash-dot line represents PID control. PID control could not achieve a satisfactory tracking result even after 30 s, as tracking error existed with oscillation. Therefore PID was disregarded, leaving only a comparison between the proposed MPC-SMC and backstepping methods. The simulation results show that the tracking performance of both the backstepping control algorithm and MPC algorithm was nearly ideal, with errors converging to zero.

A speed curve comparison for spiral line trajectory tracking control is shown in Fig. 9; a comparison of the speed change curve is shown in Fig. 10. Because the spiral line trajectory was continuous and differentiable at every point, the main complication of trajectory tracking was the speed jump problem caused

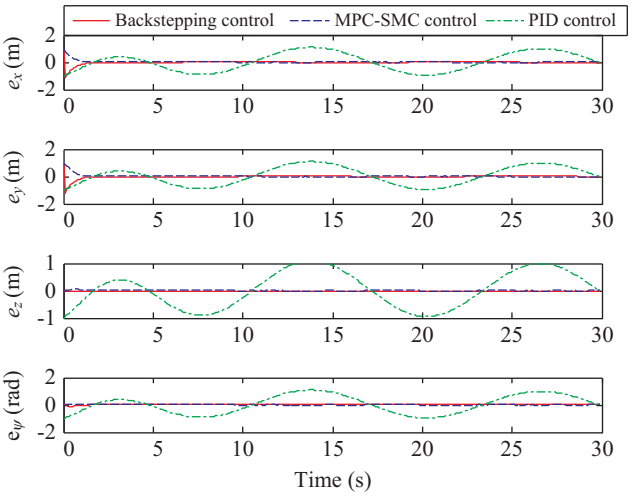

Fig. 8. Tracking errors.

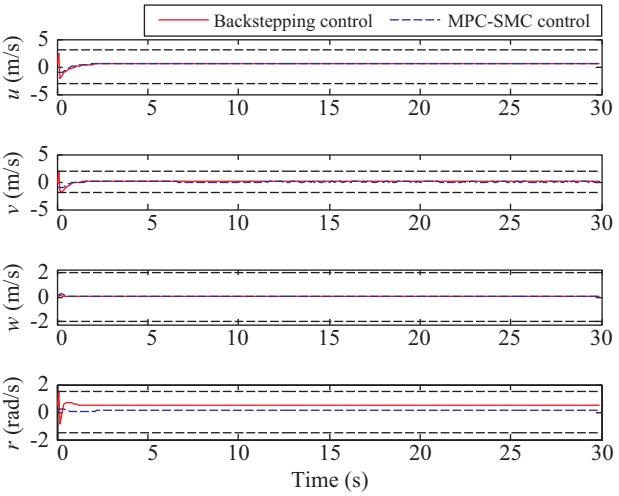

Fig. 9. Output speed values.

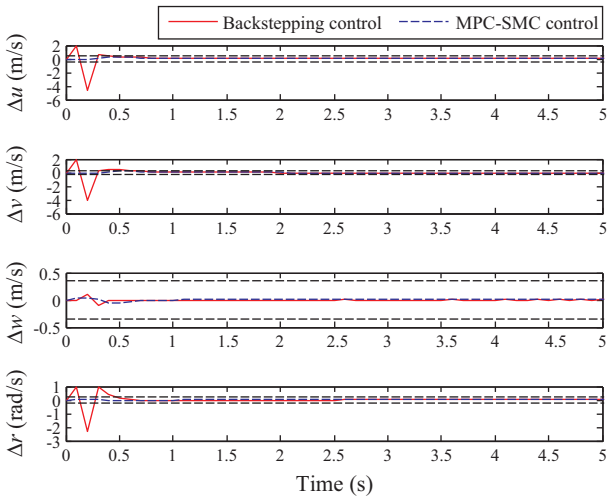

Fig. 10. Speed change values (first 5 s).

Table 2. Maximum thrust values.

Method	$\overline{T_1}$	$\overline{T_2}$	$\overline{T_3}$	$\overline{T_4}$	$\overline{T_5}$
Backstepping	-3.921	0.599	-2.367	-0.954	0.069
MPC-SMC method	-0.809	-0.267	-0.970	0.202	0.041

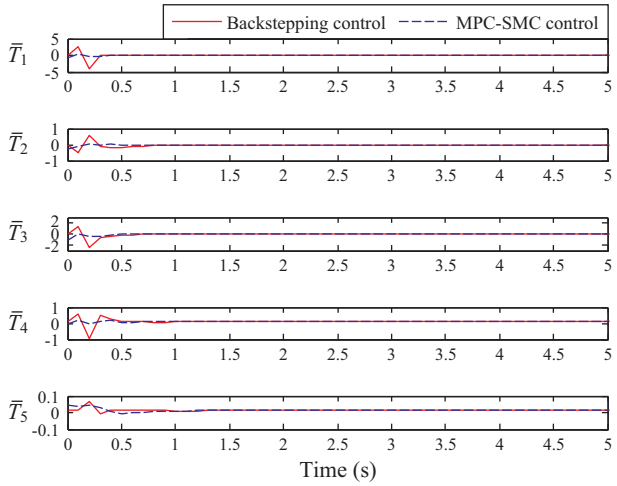

Fig. 11. Normalized thrust values (first 5 s).

by initial position error. After 5 s, both methods achieved stable tracking; therefore, the intercept simulation curve was analyzed only for the first 5 s (Fig. 10).

Table 2 lists the normalized thruster control variables in the initial state. The speed change values for the two methods were noticeably different at the initial time, despite being within a reasonable range. In the initial state, sharp speed jump phenomena occurred in the backstepping method due to position error. For example, the maximum surge speed reached nearly 5 m/s, which is faster than the vehicle's maximum speed. Therefore, the backstepping method would be impossible to implement in practice. Fig. 11 and Table 2 show that multiple maximum thrust values of the backstepping control algorithm are more than 1, while all maximum thrust values are within the permitted scope for the MPC algorithm since it was restricted by its own constraints.

2. Polyline Tracking

Next, polyline tracking control was conducted to evaluate the control performance. Here, the polyline trajectory equation is a piecewise function that is defined as follows:
when

$$0 \le t \le 10,\ \begin{cases} x_d(t) = 0.3 * t + 3 \\ y_d(t) = 0.3 * t \\ z_d(t) = 0.3 * t \\ \psi_d(t) = \pi/4 \end{cases};$$

when

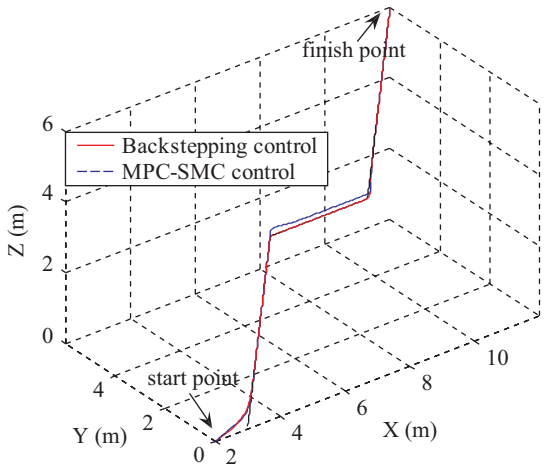

Fig. 12. Polyline tracking result.

$$10 \le t \le 20,\ \begin{cases} x_d(t) = 0.3 * (t - 10) + 6 \\ y_d(t) = 3 \\ z_d(t) = 3 \\ \psi_d(t) = 0 \end{cases};$$

when

$$20 \le t \le 30,\ \begin{cases} x_d(t) = 0.3 * (t - 20) + 9 \\ y_d(t) = 0.3 * (t - 20) + 3 \\ z_d(t) = 0.3 * (t - 20) + 3 \\ \psi_d(t) = \pi/4 \end{cases}$$

Under normal conditions, the UUV initially is not exactly on the desired trajectory. The initial desired state was set as (3, 0, 0, π/4), while the real initial state was (2, 0, 0, π/4). The sampling time was 0.1 s, and time varied from 0 to 30 s. The control parameters for backstepping control were set as $k_1 = k_2 = k_3 = 12$.

The results of the two tracking controllers for three-dimensional polyline tracking are shown in Fig. 12; the tracking error curves are illustrated in Fig. 13. Figs. 12 and 13 demonstrate that both the backstepping and MPC-SMC methods followed the desired trajectory closely. For the reason described previously in section 1, the UUV control constraint is not considered in the backstepping control method. In the simulation study, the control law calculated using the backstepping method was assumed to be satisfied, while in practice it could not be satisfied. The given control signal was beyond the physical limit of the UUV, and the tracking performance could not match the calculated results. This is reflected on the graph in the responses of the velocity curve and the normalized thrust forces.

Because of a large initial posture error between the desired

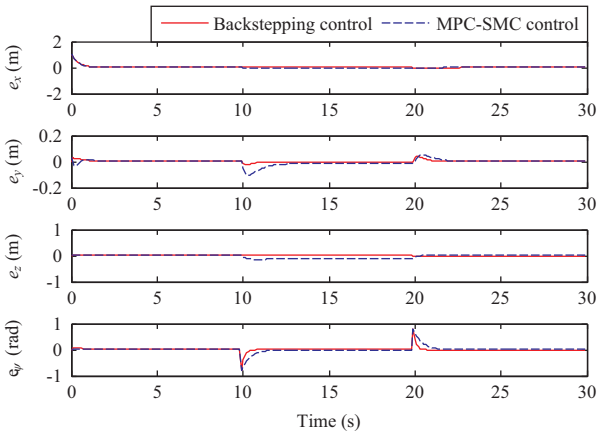

Fig. 13. Tracking errors.

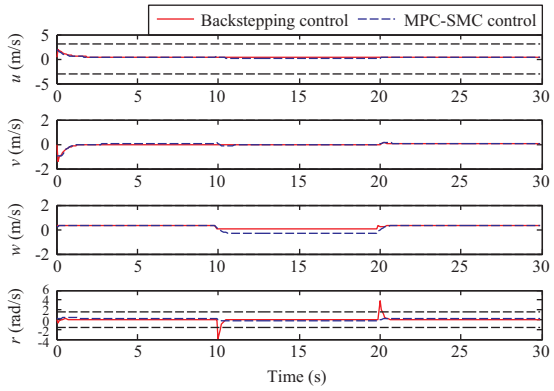

Fig. 14. Output speed values.

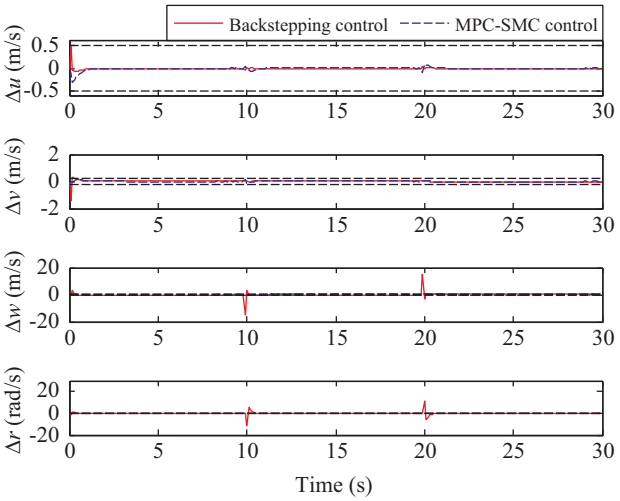

Fig. 15. Speed change values.

state and the actual state, large speed and speed change values were produced, as shown in Figs. 14-16. For example, the yaw velocity at 10 s and 20 s reached 4 rad/s, and the heave speed change value even exceeded 20 m/s , which was beyond the preset range. To generate such high speed values, the corresponding thrusters generated large thrust forces, which are shown in Fig. 17, and the normalized maximum thrust values are listed in Table 3.

Table 3. Maximum thrust values.

method	$\overline{T_1}$	$\overline{T_2}$	$\overline{T_3}$	$\overline{T_4}$	$\overline{T_5}$
Backstepping	-1.774	1.868	1.868	1.774	0.715
MPC-SMC	0.647	0.885	0.371	0.968	0.664

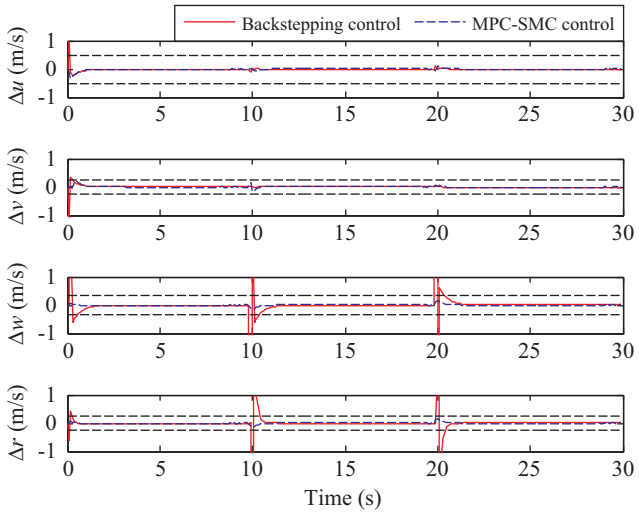

Fig. 16. Enlarged view of speed change values.

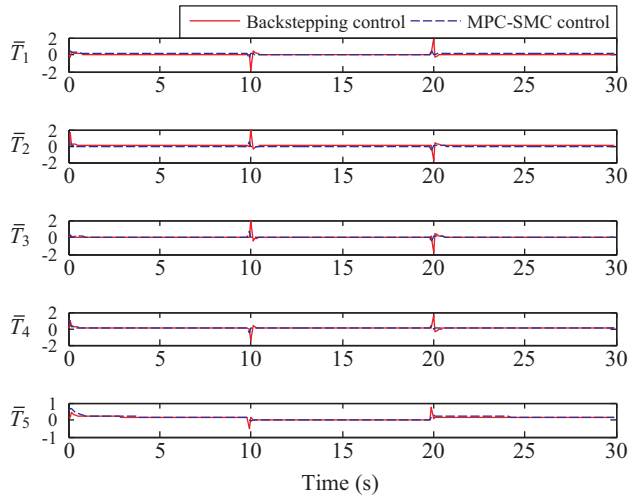

Fig. 17. Normalized thrust values.

As shown in Fig. 17 and Table 3, the first four thruster values of the backstepping control algorithm were substantially beyond the maximum thrust of the propeller. Although the trajectory tracking control results of the backstepping method are promising, obtaining these results is not possible in practice.

Because of the MPC algorithm's constraint conditions, the speed, speed change, and normalized thrust values were all con-

trolled within the maximum range that can be achieved by underwater vehicles while still meeting the thrust saturation limit. At the first turning point, sudden change of the yaw angle generated a speed jump in the yaw velocity *r*. The angular velocity of the backstepping control method jumped directly from 0 to 4 rad/s, which required the thrusters to produce large moments to achieve the desired angular velocity. Fig. 17 shows that the normalized thrust value of thruster 1 exceeded 1. By contrast, angular velocity was lower than the corresponding baseline value with the MPC algorithm, and the speed jump problem caused by state changes was effectively suppressed, with all thrust values less than 1.

V. CONCLUSION

To solve the speed jump problem encountered in UUV tracking control, a cascaded algorithm with a combination of MPC and SMC is proposed. The simulation results showed that the MPC algorithm not only realized the expected trajectory but also solved the speed jump problem while maintaining thruster forces within the thrust saturation limit. Therefore, the MPC-SMC method is a feasible and effective method of UUV trajectory tracking control.

ACKNOWLEDGEMENTS

This project was supported by the National Natural Science Foundation of China (61503239, U170620065), the National Key Project of the Research and Development Program (2017YFC0306302), and the Creative Activity Plan for the Science and Technology Commission of Shanghai (15550722400). This manuscript was edited by Wallace Academic Editing.

REFERENCES

Bagheri, A., T. Karimi and N. Amanifard (2010). Tracking performance control of a cable communicated underwater vehicle using adaptive neural network controllers. Applied Soft Computing 10, 908-918.

Cao, X., D. Zhu and S. X. Yang (2016). Multi-AUV target search based on bioinspired neurodynamics model in 3-D underwater environments. IEEE Transactions on Neural Networks and Learning Systems 27, 2364-2374.

De Paula, C. F. (2012). An improved analytical PID controller design for non-monotonic phase LTI systems. IEEE Transactions on Control Systems and Technology 20, 1328-1333.

Fossen, T. I. (2011). Handbook of Marine Craft Hydrodynamics and Motion Control. John Wiley & Sons, New Jersey.

Khadija, D., L. Majdam and N. A. Said (2012). Discrete second order sliding mode control for nonlinear multivariable systems. Electrotechnical Conference (MELECON), 2012 16th IEEE Mediterranean, Yasmine Hammamet, Tunisia, 387-390.

Kuhne, F. and W. F. Lages (2004). Model predictive control of a mobile robot using linearization. Proceedings of Mechatronics and Robotics, 525-530.

Li, S., X. Wang and L. Zhang (2015). Finite-time output feedback tracking control for autonomous underwater vehicles. IEEE Journal of Oceanic Engineering 40, 727-751.

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

Pan, C. Z., X. Z. Lai and S. X. Yang (2015). A biologically inspired approach to tracking control of underactuated surface vessels subject to unknown dynamics. Expert System with Applications 42, 2153-2161.

Pan, C., X. Lai and S. X. Yang (2015). A bioinspired neural dynamics-based approach to tracking control of autonomous surface vehicles subject to unknown ocean currents. Neural Computing and Applications 26, 1929-1938.

Su, Y., K. Tan and H. Tong (2013) Computation delay compensation for real time implementation of robust model predictive control. Journal of Process Control 23, 1342-1349.

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

Sun, B., D. Zhu and S. X. Yang (2017). An optimized fuzzy control algorithm for three-dimensional AUV path planning. International Journal of Fuzzy Systems, 1-14.

Wai, R. J. (2007). Fuzzy Sliding-mode control using adaptive tuning technique. IEEE Transactions on Industrial Electronics 54, 586-594.

Wallace, M. B., S. D. Max and K. Edwin (2008). Depth control of remotely operated UUV using an adaptive fuzzy sliding mode controller. Robotics and Autonomous Systems 56, 670-677.

Wang, H. F., S. Zhu and S. G. Liu (2009). Adaptive PID control of robot manipulators with H tracking performance. IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Singapore, 1515-1520.

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

Xu, J., M. Wang and L. Qian (2015). Dynamical sliding mode control for the trajectory tracking of underactuated unmanned underwater vehicles. Ocean Engineering 105, 54-63.

Yu, S. Y., X. Li and H. Chen (2015). Nonlinear model predictive control for path following problems. International Journal of Robot and Nonlinear Control 25, 1168-1182.

Zhu, D., X. Cao and B. Sun (2017). Biologically inspired self-organizing map applied to task assignment and path planning of an auv system. IEEE Transactions on Cognitive and Developmental Systems.