International Journal of Intelligent Robotics and Applications (2024) 8:1–13
https://doi.org/10.1007/s41315-023-00301-2

REGULAR PAPER

Image-based visual servoing control of remotely operated vehicle for underwater pipeline inspection

Xiongfeng Yi[1] · Zheng Chen[1]

Received: 18 April 2023 / Accepted: 18 September 2023 / Published online: 10 October 2023

Abstract

This paper presents the development of an image-based visual servoing control system for underwater vehicles, designed to track underwater pipelines using an onboard monocular camera. We propose a novel line-pickup criterion to ensure continuous, smooth tracking when the pipeline changes its direction. An angle projection method is developed to capture the current heading angle relative to the target line and the target angle in the camera view using a 3D camera projection model. A dynamic model of a remotely operated vehicle (ROV) is developed for tracking control design. A nonlinear controller based on nonlinear feedback linearization method is designed for the ROV to track the target angle. A Lyapunov-based stability analysis is provided to prove the local stability of the nonlinear control. The visual servoing control with embedded image process is simulated in an OpenGL environment and implemented on a BlueROV in real-time experiments. The comparisons with different conditions reveal the effect of control parameters on the tracking performance. The experiments have been conducted to test the performance of the model-based nonlinear controller and the PD controller. From the experimental results, the coordination among control signals and state values has validated the effectiveness of this method even when the ROV is affected by the disturbance.

1 Introduction

The submarine pipelines are crucial equipment in offshore petroleum transportation. Leaks in these pipelines have caused massive losses and pollution. Yet, inspecting for pipeline deformation and flaws presents a considerable challenge, even for professional divers due to high pressure and potential dangers from unknown environments (Duran et al. 2007). As an alternative approach, using remotely operated vehicle (ROV) for pipeline inspection has proven more efficient and safer. While ROVs have seen widespread use for deep-sea exploration, their manual control based on onboard camera video necessitates extensive training for professional operators. Moreover, automatic control of ROVs presents its own difficulties, such as the nonlinear dynamics with high degrees of freedom and the lack of accurate positioning systems for navigation.

There are several systematic approaches for modeling and control of underwater vehicles, including notable works by Sutulo and Soares (2006) and Fossen (2011). They provided functional formulas and research directions for researchers, paving the foundation for innovations such as those developed by Wang et al., which include a modeling and motion control system for a mini autonomous underwater vehicle (AUV) (Wang et al. 2009), and an elastic band-based algorithm for motion planning in various environments (Wang et al. 2020). However, these earlier works have their limitations. For instance, these methods often require precise information about the system's dynamics and, in some cases, the exact layout of the environment. This information is rarely available in real-world scenarios, especially in unpredictable underwater conditions.

There are various designs and control system architectures for autonomous underwater vehicles. In the summation of Yuh's research (Yuh 2000), most underwater vehicles installed inertial sensors to provide the acceleration information. The drifting error in dynamic estimation is reduced by measuring the magnetic compass with a Doppler Velocity Log (DVL). Since the Global Position System (GPS) is not well performed for underwater positioning, underwater vehicles typically use sonar for environment scanning. Cameras are another frequently-used sensors to provide position and direction. On-board cameras are also frequently used to

✉ Zheng Chen
zchen43@central.uh.edu

[1] Mechanical Engineering, University of Houston, 4726 Calhoun Rd, Room N207, Houston, TX 77204, USA

provide positional and directional information (Tian et al. 2006). However, Inertial sensors are susceptible to drift over time, and the performance of DVLs can be severely degraded in environments with sparse acoustic scatterings.

Despite these challenges, underwater vehicles with onboard cameras have been successful in various pipeline tracking missions using different visual servoing algorithms. Sfahani et al. designed a sliding mode control for an autonomous underwater pipeline tracking task (Sfahani et al. 2017). Shi et al. proposed an underwater pipeline tracking system for amphibious spherical robots, where the vehicle's feedback information is purely based on a monocular camera (Shi et al. 2017). Later, Sheng et al. discussed a new AUV pipeline recognition and localization based on the linear feature extraction in terms of the traditional Hough algorithm (Sheng et al. 2018). Even though these works have advanced the field, they too have limitations. For instance, sliding mode control requires an accurate model of the system dynamics and suffers from chattering, a phenomenon that can lead to wear and tear in real-world systems. While image-based feedback has shown promise, it is still challenging to deal with image distortion due to the refraction and absorption of light in water.

With a low-cost expansion of ROV, Aripin et al. developed their underwater remotely operated crawler (ROC). Instead of extensive power cost on depth control, their ROC can inspect the pipeline by crawling on the seafloor (Aripin et al. 2017). Although this idea saves the energy cost for a long time underwater operation, some parameters in the model and control algorithm have to be identified carefully in the seabed environment before the vehicle is deployed to the real pipeline tracking mission. For example, the pipeline might be buried under the seabed. Bharti et al. developed a semi-heuristic approach to track buried underwater pipelines using Fluxgate Magnetometers (Bharti et al. 2020). Besides the pipeline's bury problem, there are many environmental uncertainties while the vehicle is tracking the path. Almdari et al. addressed a robust trajectory tracking algorithm for under-actuated underwater vehicles in different uncertain environments (Heshmati-Alamdari et al. 2020). Zheng et al. mentioned the fixed-time sliding mode tracking control for AUV (Zheng et al. 2021). In the research (Shojaei 2018), the underwater vehicle tracking control was applied without velocity sensors. Shi et al. developed an distributed circumnavigation control for AUV based only on local information (Shi et al. 2021). However, these control methods require complex control algorithms which may not be suitable for real-time implementation due to their computational requirements. Additionally, these methods typically require knowledge of the vehicle's velocities, which are challenging to measure directly underwater. Some researchers have attempted to combine adaptive neural networks with sliding mode control to manipulate AUVs without a thrust model (Chu et al. 2017). However, these advanced control methods still have challenges to overcome, including the need for extensive training data and high computational power for real-time operation, making them difficult to implement in remote, resource-limited scenarios.

In light of the above challenges, this paper proposes a robust visual servoing control for underwater vehicles to follow underwater pipelines using only a single-lens camera. A path planning method transforms the three variables path following problem into a trajectory tracking problem. in which the ROV follows a desired path by tracking the reference angle. In the camera view, the direction of pipelines in the image is computed by Hough transformation and then a line pickup criteria is developed to choose which pipeline needs to be tracked. With the camera projection relationship, the vehicle heading direction related to the pipeline and the target angle are the inputs for the controller. The required torque is then given from the controller output while the linear force is constant and along the heading direction. The displacement and rotation of the vehicle are updated numerically. A short sampling time is chosen for solving the state-space ODE model in real-time, and the controller delay time interval is large to simulate the transmission delay and computing cost in the simulation. A significant delay introduces more errors because of no feedback information before the next control update. However, it is compatible with more sophisticated algorithms and lighter processors.

A nonlinear control with feedback linearization is then developed to minimize the tracking error between the targeted heading angle and the actual heading angle. A Lyapunov-based stability analysis is provided to prove the local stability of the control system. Both simulation and experimental results have validated the visual servoing control on a BlueROV2 with and without disturbances. The major contributions of this paper are the following. (1) The angle-based path planning method transfers a 2D path following problem into a 1D trajectory tracking problem, which expedites the process in lane detection and tracking without knowing the depth; (2) the line pickup criteria rules out the noise and chooses the proper target direction from multiple pipeline segments; (3) the model-based nonlinear visual servoing controller tracks the reference angle, and the whole process is with the disturbance; (4) the Lyapunov-based local stability analysis defines the range of the control parameters.

The rest of this paper is organized as follows. Section 2 describe the method of pipeline tracking and overall system description. The ROV model is described in Sect. 3. Section 4 introduces the image process including camera angle projection rules and line pickup criteria. Section 5 includes the controller design with the stability analysis and safety angles. Control system simulation in a virtual 3D environment with different parameters and experimental results with a BlueROV2 are presented and discussed in Sect. 6.

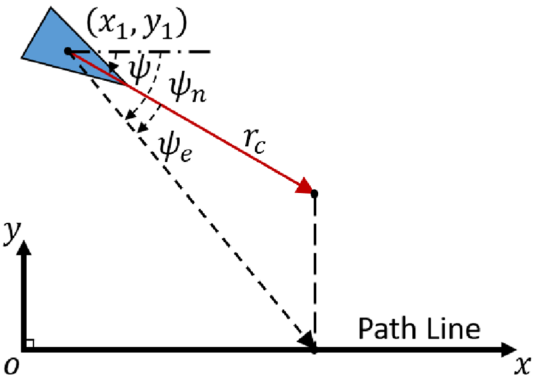

Fig. 1 The path tracking algorithm description

Conclusion and future work directions are discussed in Sect. 7.

2 Method and system diagram

The goal of this research is to develop a visual servoing control for an underwater vehicle to track an underwater pipeline. The vehicle has a detect vector $\vec{r_c}$ with a length of r_c along its moving direction as the red arrow in Fig. 1, where the 2D earth fixed position of the vehicle is (x_1, y_1), the current heading angle is ψ, the targeted heading direction is ψ_n (black dashed arrow), and the control error ψ_e (Reynolds 1999), where

$$\psi_e = \psi_n - \psi. \tag{1}$$

The steering control objective is to make the angle ψ_e go to zero. When ψ_e is zero and y_1 is non-zero, the vehicle moves towards the path line. After the vector r_c passes the path line, since y_1 is non-zero, the targeted angle ψ_n will become smaller thus ψ_e will become negative. Then the control will correct that negative angle to reduce its heading angle ψ to zero. As soon as both ψ_e and y_1 become zero, the vehicle will be kept to move along the path line. Simply controlling ψ to zero cannot guarantee that y_1 approaches zero. The value of r_c determines how smooth the vehicle approaches the path line.

The overall control system diagram is shown in Fig. 2. In each control iteration, an on-board monocular camera sends one image to an image process system which runs in a control computer outside the vehicle. In the image process system, a well-tuned Hough algorithm detects all the line characters, such as the angle and distance respective to the vehicle, in the image; an angle-projection method transfers all the detected lines from the image coordinates to the world coordinates; a line detection criteria derives the current angle ψ and the target angle ψ_n, as shown in Fig. 1, from all the detected lines in world coordinates. The control error ψ_e, is then sent to the controller. At each sampling time, the controller (designed in the control laptop) outputs the PWM signal which is sent back to the ROV motor to steer the vehicle. After that a new control iteration starts. The details inside the line detection block are shown in the lower figure of Fig. 2.

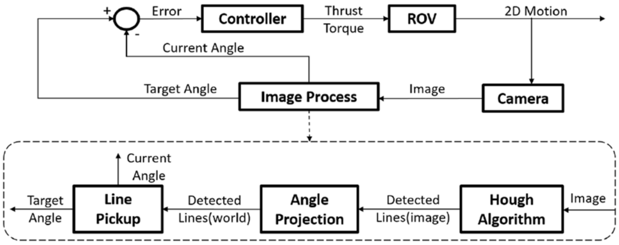

Fig. 2 Overall system diagram. The dash line block shows the detail of the image process, where the input is the image from the camera, and the outputs are the current heading angle and the target angle

3 Dynamic model of ROV

A general underwater vehicle dynamic model in the body-fixed coordinates (Fossen 2011) is:

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

where

$$\begin{aligned} \mathbf{M} &= \text{inertia matrix (including added mass)}, \\ \mathbf{C}(\boldsymbol{\nu}) &= \text{matrix of Coriolis and centripetal terms} \\ &\qquad \text{(including added mass)}, \\ \mathbf{D}(\boldsymbol{\nu}) &= \text{damping matrix}, \\ \mathbf{g}(\boldsymbol{\nu}) &= \text{gravity forces and moments}, \\ \boldsymbol{\tau} &= \text{control inputs}. \end{aligned} \tag{3}$$

In this research, the following assumptions are made before the modeling work:

- The ROV is moving at a slow speed (less than 1 m/s) during the pipeline inspection, thus the lift forces are negligible;
- The ROV is assumed to have port-starboard symmetry and fore-aft symmetry, and the center of gravity (CG) is located in the symmetry planes, thus the roll and pitch motions are neglected;
- The environment water is an ideal fluid;
- The force from the tether attached to the ROV is neglected;

Under the aforementioned assumptions, this 3D pipeline tracking problem becomes a 2D path following the problem. In Eq. 2, the CG is the origin point of the coordinates, and the matrices are given as follows:

$$\mathbf{M} = \begin{bmatrix} m - X_{\dot{u}} & 0 & 0 \\ 0 & m - Y_{\dot{v}} & 0 \\ 0 & 0 & I_z - N_{\dot{r}} \end{bmatrix}, \tag{4}$$

$$\mathbf{C}(\boldsymbol{v}) = \begin{bmatrix} 0 & 0 & Y_{\dot{v}}v - mv \\ 0 & 0 & mu - X_{\dot{u}}u \\ mv - Y_{\dot{v}}v & X_{\dot{u}}u - mu & 0 \end{bmatrix}, \tag{5}$$

$$\mathbf{D}(\boldsymbol{v}) = -\begin{bmatrix} X_u + X_{u|u|}|u| & 0 & 0 \\ 0 & Y_v + Y_{v|v|}|v| & 0 \\ 0 & 0 & N_r + N_{r|r|}|r| \end{bmatrix}, \tag{6}$$

where the values of all the added masses and drag coefficients are negative, and,

$$\boldsymbol{v} = [u, v, r]^T. \tag{7}$$

$\boldsymbol{v}$ is the vector of velocities in the body-fixed coordinates. The states in the earth fixed coordinates can be written as:

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_{\psi}(\boldsymbol{\eta})\boldsymbol{v}, \tag{8}$$

where $\boldsymbol{\eta}$ is the vector of the positions in the x and y direction and the heading angle,

$$\boldsymbol{\eta} = [x, y, \psi]. \tag{9}$$

In the 3 degree-of-freedom (DOF) coordinates, the transformation $\mathbf{J}_{\psi}$ is:

$$\mathbf{J}_{\psi} \overset{3DOF}{=} \boldsymbol{R}(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{10}$$

In Eqn. 2, the control inputs are the total forces in the x and y directions and the torque in the z direction, which can be written as:

$$\boldsymbol{\tau} = [F_x, F_y, T_{\psi}]^T. \tag{11}$$

As shown in Fig. 3, the forces and the torques from thrusters $1-6$, which generate the thrusts $F_1 - F_6$. The thrusters 5, 6 provide the vertical thrust, while the thrusters $1-4$ control the motion in $x-y$ plane. The parameters are defined in the cross-sectional view of the ROV in Fig. 3. The buoyancy control along the z direction is important for the ROV control. However, in this research, we assume that the vertical motion has been well controlled, and the depth is maintained.

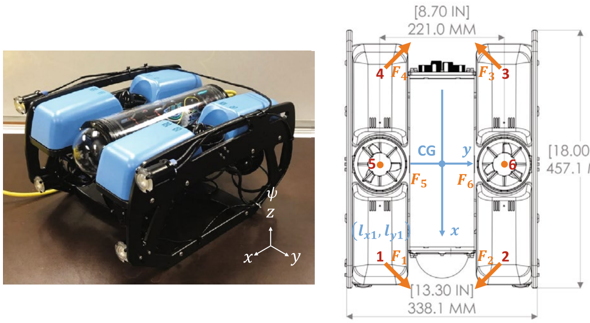

Fig. 3 The BlueROV and schematics of it (top view)

From Fig. 3, the force and the torque from each thruster are:

$$\boldsymbol{\tau}_i = \begin{bmatrix} F_{xi} \\ F_{yi} \\ T_{\psi i} \end{bmatrix} = \begin{bmatrix} F_i \cos\gamma_i \\ F_i \sin\gamma_i \\ F_{yi}l_{xi} - F_{xi}l_{yi} \end{bmatrix}, \tag{12}$$

where F_i, F_{xi}, F_{yi} and $T_{\psi i}$ are the thrust components in x, y direction and the torque for the thruster i $(i = 1, 2, 3, 4)$, respectively. γ_i is the angle of the thrust force, and (l_{xi}, l_{yi}) is the position of the thruster in the body fixed coordinates. An example of (l_{x1}, l_{y1}) is shown in Fig. 3. The total force acting on the ROV is:

$$\boldsymbol{\tau} = \sum_{i=1}^{4} \boldsymbol{\tau}_i, \tag{13}$$

Since a monocular on-board camera is not able to calculate the displacement information without an external reference, our strategy is to set F_x to its maximum F_{xmax}, set F_y to zero, and control the heading angle so that $\psi \rightarrow \psi_n$ in Fig. 1. Besides Eqs. 12 and 13, there is an additional restriction, which is:

$$F_i \in [-F_{max}, F_{max}], \tag{14}$$

where F_{max} is the maximum output force for each thruster. With the assumption of the symmetric distribution of the thrusters $1-4$, according to Eqs. 12, 13, 14, the relationship between F_x and T_{ψ} is:

$$\frac{F_{xmax}}{\cos\gamma_1} + \frac{T_{\psi}}{l_{x1}\cos\gamma_1 - l_{y1}\sin\gamma_1} \in [-4F_{max}, 4F_{max}]. \tag{15}$$

Since F_{xmax} is positive, the restriction of the torque is

$$|T_{\psi}| \le T_{max} = (4F_{max} - \frac{F_{xmax}}{\cos\gamma_1})(l_{x1}\cos\gamma_1 - l_{y1}\sin\gamma_1). \tag{16}$$

Equation 16 separates the forward thrust force and the yaw torque to the different regions. Thus, during the path following control in the next section, one can have $F_x = F_{xmax}$ and $|T| \leq T_{max}$. One can assume that the control of torque does not affect the forward thrust.

4 Image processing

4.1 Angle projection

In a 3D environment, a monocular camera provides the angle information in the $x' - y'$ plane (Body fixed coordinates), where the x' axis is along the vehicle's moving direction (along r_c in Fig. 1). Under the assumption that the distortion is neglected, the relationship between the heading and the pipeline is revealed in Fig. 4 where the camera detects the edge of the pipeline. Figure 4(A) shows the view of the ROV from the on-board camera, (B) shows the view from the right-hand-side of the camera, and Fig. 4(C) shows the view observed from above. The notations with subscripts 'm' and 'w' represent the points in the image and world coordinates. In Fig. 4(A), the blue and red dash-dot line is the vanishing line of the camera view and the direct line of the vehicle, respectively.

For a well-calibrated vehicle camera, the vanishing line is a horizontal line passing through the vanishing point 'O_m', and the direct line is in the middle of the image. They are perpendicular to each other. The camera has a tilting angle of'α' from the horizontal direction about the y' axis. If $\alpha = 0$, the vanishing line will be in the middle of the image, where the middle line (black dash-dot line) is located. The intersection of the direction line and the middle line is the origin point of the image, which is also the focal point of the camera. In the later simulation, α is given directly in the program. The tilting angle of the camera is controlled by a gimbal mounted in the ROV.

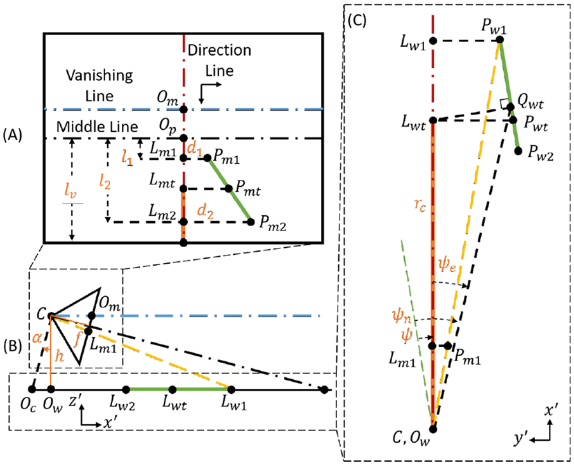

Fig. 4 Description of angle projection relationship from the world to image coordinates. **A** Is the camera image view, **B** is the view from the right-hand-side of the camera, and **C** is observed from above

In Fig. 4A, P_{m1} and P_{m2} have their projections on the direction line as $L_{m.}$. In Fig. 4B, the triangle represents the camera, where the vertical line represents the sensor plane; the black bottom line is the horizontal earth plane. CO_c is perpendicular to the optical axis of the camera, and O_c represents the ground. The projection of $P_{w.}$ in the direction line in the real-world coordinates is $L_{w.}$. The yellow dash line is the light passing through P_{m1} and P_{w1}. In Fig. 4C, the red dash line is the direction line; the thick red line is r_c, the green dash line is parallel to the target line (green line). The angles ψ, ψ_n, and ψ_e defined in Fig. 1 are also shown in Fig. 4C.

For easy implementation in following derivations and programs, in the '$column - row$' coordinates of Fig. 4A, O_p, O_w are the origin points for the image and real-world, respectively. Define $P_{m1}(d_1, l_1)$, $P_{m2}(d_2, l_2)$, the length of CO_w as h, CO_w as f, where h is the vertical distance of the vehicle from the earth plane, f is the focal length of camera in pixels since image distance is approximately equal to focal length in the case of long object distance (Gåsvik 2003). The coordinates of points on the pipeline in the real-world are $P_{w1}(P_{w1x'}, P_{w1y'})$, $P_{w2}(P_{w2x'}, P_{w2y'})$. Since two points have equal states, an example of P_{w1} illustrates the relationship in Fig. 4: $O_cL_{w1} // CO_m$, and the following equation related to the coordinates of P_{w1} holds:

$$\frac{\frac{f}{\cos\alpha}}{f\tan\alpha + l_1} = \frac{P_{w1x'} + h\tan\alpha}{\frac{h}{\cos\alpha}}, \tag{17}$$

$$\frac{d_1}{-P_{w1y'}} = \frac{\frac{f}{\cos\alpha}}{P_{w1x'} + h\tan\alpha}. \tag{18}$$

Thus, the coordinates of the points are:

$$P_{w1x'} = \frac{hf}{\cos^2\alpha(l_1 + f\tan\alpha)} - h\tan\alpha, \tag{19}$$

$$P_{w1y'} = -\frac{d_1 h}{\cos\alpha(l_1 + f\tan\alpha)}, \tag{20}$$

$$P_{w2x'} = \frac{hf}{\cos^2\alpha(l_2 + f\tan\alpha)} - h\tan\alpha, \tag{21}$$

$$P_{w2y'} = -\frac{d_2 h}{\cos\alpha(l_2 + f\tan\alpha)}. \tag{22}$$

Since $P_{w1}P_{w2}$ is the reference path line, the heading angle ψ in Fig. 1 is:

$$\psi = -\arctan\frac{P_{w1y'} - P_{w2y'}}{P_{w1x'} - P_{w2x'}}, \tag{23}$$

where $\psi \in (-\frac{\pi}{2}, \frac{\pi}{2}]$.

In Fig. 4A, L_{mt} is in the direction line (red dash-dot line) and in the middle between the middle line (black dash-dot line) and the bottom edge. The point P_{mt} is in the line $P_{w1}P_{w2}$, and $P_{mt}L_{mt} // P_{m1}L_{m1}$. The projections of L_{mt} and P_{mt} in (C) are L_{wt} and P_{mt}, respectively. The vector O_wL_{wt} is the detected vector r_c in (C) (thicker red solid line). Q_{wt} is on $P_{w1}P_{w2}$, $L_{wt}Q_{wt}$ is normal to $P_{w1}P_{w2}$. The target angle ψ_n is $\angle L_{wt}O_wQ_{wt}$. The positions of P_{wt} and Q_{wt} are computed as:

$$P_{wtx'} = \frac{2hf}{(l_v + 2f\tan\alpha)\cos^2\alpha} - h\tan\alpha, \tag{24}$$

$$P_{wty'} = \frac{-P_{wtx'}(\frac{l_v}{2}(d_1 - d_2) + l_1d_2 - l_2d_1)}{\frac{f}{\cos\alpha}(l_1 - l_2)}, \tag{25}$$

$$Q_{wtx'} = \sin\psi\cos\psi P_{wty'} + P_{wtx'}, \tag{26}$$

$$Q_{wty'} = \cos^2\psi P_{wty'}. \tag{27}$$

where l_v is the distance from the middle to the bottom edge of Fig. 4(A), and all the variables are yellow letters in Fig. 4. The target direction angle for the vehicle is:

$$\begin{aligned}\psi_n &= \psi + \psi_e,\\ &= \psi + \arctan\frac{Q_{wty'}}{Q_{wtx'}},\\ &= \psi + \arctan\frac{\cos^2\psi\frac{l_v(d_1-d_2)+2l_1d_2-2l_2d_1}{(l_1-l_2)}}{\sin 2\psi\frac{l_v(d_1-d_2)+2l_1d_2-2l_2d_1}{2(l_1-l_2)} - \frac{2f}{\cos\alpha}}.\end{aligned} \tag{28}$$

In both Eq. 28, 'h' is canceled in numerator and denominator, thus the height of the ROV is not necessary for the calculation of the target direction angle.

4.2 Line pickup

The Hough algorithm detects possible lines through the image segmentation method with a binary mask (Duda and Hart 1972). The result contains the lines of the pipeline's edge and noise. An example is shown in Fig. 5A, where the orange lines are the detection results from the Hough algorithm. In the world coordinates, all the detected lines from the same pipeline are parallel to each other, and their differences in ρ from the Hough transformation are small, while the noise lines have no relations between each other. Figure 5B shows the distribution of the lines in the $\psi - \rho$ plane, where the clusters with most points represent the target lines for the tracking control. Figure 5C is the 2D histogram of the distribution in (B), where the brighter color represents more points in the block. The two blocks with the most histogram value are chosen regions. The average of lines within that region is the target line to be tracked by the ROV.

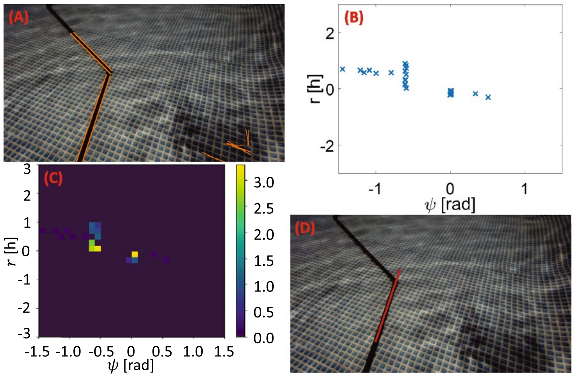

Fig. 5 **A** Is the detection result where orange lines are computed from the Hough algorithm; **B** is the distribution of lines in the world coordinates, where the x axis is the angle, and the y axis is the distance from origin point to the lines; **C** is the histogram of (**B**); **D** shows the target line

In Fig. 5, there are two candidates from the histogram plot. The rule to decide the current tracking line has to be satisfied with three requirements: First, the current line must be closer to the camera; second, if the current line is too short in the image, the target line switches; Finally, after the line switches, the target does not switch back to the previous line for any instance. In order to pick up the target line in the view, for the ith line in the current iteration, each detected line segment has a weight $w^{[i]}$, which is given in Eq. 29. ψ_{prev} is the heading angle in the previous iteration. The candidate with the largest weighted histogram value is the target line to be tracked. In each iteration, since the time interval is short, the angle change for the ROV is very small. The angle difference is small between the detected lines of two neighbor iteration unless the length of the current line is too short in the image. Figure 5D presents the final result pipeline detection.

$$w^{[i]} = \sqrt{(d_1^{[i]} - d_2^{[i]})^2 + (l_1^{[i]} - l_2^{[i]})^2}\frac{\pi}{|\psi^{[i]} - \psi_{prev}| + \frac{\pi}{c_s}}. \tag{29}$$

5 Control design and stability analysis

This section includes the controller design, stability analysis, and angle restriction. With the designed model-based controller, there are two steps for the system to reach a steady-state: (1) Prove the convergence of the angle, where $\psi \rightarrow \psi_n$; (2)

Validate that the whole system is stable with the condition from (1).

For the first step, the controller is defined to guarantee the stability in angle. According to Fig. 1 in the earth fixed coordinates, the reference angle has the following equation

$$\tan \psi_n = -\frac{y}{r_c \cos \psi}. \tag{30}$$

According to Eq. 2, under the assumption that the added mass has $Y_{\dot{v}} = X_{\dot{u}}$, the control in ψ domain is:

$$(I_z - N_{\dot{r}})\dot{r} = T_\psi + N_r r + N_{r|r|}|r|r \tag{31}$$

The control input signal is T_ψ, which is designed to achieve a stable system. To analyze the stability, a Lyapunov function V_1 is selected as:

$$V_1 = \frac{c(\psi - \psi_n)^2 + (I_z - N_{\dot{r}})(r - \dot{\psi}_n)^2}{2}, \tag{32}$$

where, c is a positive gain value. The time derivative of the Lyapunov function is:

$$\begin{aligned} \dot{V}_1 &= c(\psi - \psi_n)(r - \dot{\psi}_n) + (I_z - N_{\dot{r}})(r - \dot{\psi}_n)(\dot{r} - \ddot{\psi}_n) \\ &= c(\psi - \psi_n)(r - \dot{\psi}_n) \\ &\quad + (r - \dot{\psi}_n)(T_\psi + N_r r + N_{r|r|}r|r| - \ddot{\psi}_n(I_z - N_{\dot{r}})). \end{aligned} \tag{33}$$

The designed model-based controller is:

$$T_\psi = -d(r - \dot{\psi}_n) - c(\psi - \psi_n) - N_r r - N_{r|r|}|r|r + \ddot{\psi}_n(I_z - N_{\dot{r}}), \tag{34}$$

where d is a positive gain value. If one substitutes Eqs. 34 into 33, the time derivative of the Lyapunov function is:

$$\dot{V}_1 = -d(r - \dot{\psi}_n)^2. \tag{35}$$

Thus, the system is stable according to the Lyapunov Lasalle theorem (LaSalle 1960) with the controller in Eq. 34.

After the first step, where the condition that the steady-state of the ROV system in ψ domain has been achieved, where $\psi \rightarrow \psi_n$, the ROV reaches the steady state in $x - y$ plane and converges to the target path. The second Lyapunov function is:

$$V_2 = \frac{a\boldsymbol{v}_m^T \mathbf{M} \boldsymbol{v}_m + by^2 + c(\psi - \psi_n)^2 + (I_z - N_{\dot{r}})(r - \dot{\psi}_n)^2}{2}, \tag{36}$$

where $\boldsymbol{v}_m = \boldsymbol{v} - [u_{max}, 0, \psi_n]$, u_{max} is the maximum value of u. a and b are positive gain values. According to Eq. 5, $\boldsymbol{v}_m^T \mathbf{C} \boldsymbol{v}_m \equiv 0$. The time derivative of the second Lyapunov function is:

$$\begin{aligned} \dot{V}_2 &= a((F_{xmax} + X_u u + X_{u|u|}|u|u)(u - u_{max}) + Y_v v^2 \\ &\quad + Y_{v|v|}|v|v^2) + b(y(u \sin \psi + v \cos \psi)) + \dot{V}_1. \end{aligned} \tag{37}$$

Since $\psi \rightarrow \psi_n$, based on Eq. 30, there is:

$$\begin{aligned} & b(y(u \sin \psi + v \cos \psi)) = by(-\frac{y}{r_c}u + \frac{\sqrt{r_c^2 - y^2}}{r_c}v), \\ & = \frac{-b}{ur_c}(yu - \frac{\sqrt{r_c^2 - y^2}v}{2})^2 + \frac{b(r_c^2 - y^2)v^2}{4ur_c}. \end{aligned} \tag{38}$$

Equation 37 is then written as

$$\begin{aligned} \dot{V}_2 &= a(F_{xmax} + X_u u + X_{u|u|}|u|u)(u - u_{max}) \\ &\quad + a(Y_v v^2 + Y_{v|v|}|v|v^2) - \frac{b}{ur_c}(yu - \frac{\sqrt{r_c^2 - y^2}v}{2})^2 \\ &\quad + \frac{b(r_c^2 - y^2)v^2}{4ur_c} + \dot{V}_1, \end{aligned} \tag{39}$$

In Eq. 39, one can have the following inequalities:

$$\dot{V}_1 < 0,$$

$$Y_{v|v|}|v|v^2 < 0,$$

since the thrust is only in the positive direction of 'u',

$$(F_{xmax} + X_u u + X_{u|u|}|u|u)(u - u_{max}) < 0,$$

$$-\frac{b}{ur_c}(yu - \frac{\sqrt{r_c^2 - y^2}v}{2})^2 < 0,$$

and

$$v^2(aY_v + \frac{b(r_c^2 - y^2)}{4ur_c}) < 0,$$

by setting the gain $a > \frac{-b(r_c^2 - y^2)}{4ur_c Y_v}$ and $Y_v < 0$. Thus

$$\dot{V}_2 < 0.$$

According to the Lyapunov Lasalle theorem, the ROV system is stable with the designed model-based controller while following the specific path.

In application, if the target pipeline is outside the camera view, there is no feedback information, and the control is terminated, even if the vehicle is on a stable trajectory. Moreover, if $\psi \notin (-\frac{\pi}{2}, \frac{\pi}{2})$ due to the overshoot situation, the target direction is then opposite. Thus, the target angle ψ_n is assigned according to the calculation value of ψ_n and ψ from Eqs. 23 and 28 with the following restriction

$$\psi_n := \begin{cases} -\psi_s & \psi \in [-\pi, -\frac{\pi}{2}], \\ -\psi_s & \psi_n \in (-\pi, -\psi_s], \psi \in (-\frac{\pi}{2}, \frac{\pi}{2}), \\ \psi_n & \psi_n \in (-\psi_s, \psi_s), \psi \in (-\frac{\pi}{2}, \frac{\pi}{2}), \\ \psi_s & \psi_n \in [\psi_s, \pi], \psi \in (-\frac{\pi}{2}, \frac{\pi}{2}), \\ \psi_s & \psi \in [\frac{\pi}{2}, \pi), \end{cases} \tag{40}$$

where ψ_s is the safety angle. If there is no safety angle, the overshoot in ψ domain will result in a sign change of the tracking direction. The tracking control is terminated due to the system divergence or the loss of reference.

In Eq. 40, if the safety angle ψ_s is too small, the converge rate will be slow when the vehicle is far away from the target line. However, it takes the risk of overshooting when ψ_s is close to $\frac{\pi}{2}$. Rather than setting a value to compromise between continuity and convergence, a better strategy is to set two-stage safety angles ψ_{s1} and ψ_{s2}. The switch occurs when the distance between the vehicle and the target line r_d is larger than r_c. According to Fig. 4 and equations in Sect. 4.1, r_d and r_c are:

$$r_d = \sqrt{Q_{wty'}^2 + Q_{wtx'}^2} \, | \sin \psi_n |, \tag{41}$$

$$r_c = P_{wty'}. \tag{42}$$

The switching condition for the safety angle ψ_s is:

$$\psi_s := \begin{cases} \psi_{s1} & r_d \le r_c, \\ \psi_{s2} & r_d > r_c, \end{cases} \tag{43}$$

where ψ_{s1} is the first stage safety angle and $\psi_{s1} < \psi_{s2}$.

6 Simulation and experiment

This section describes the simulation and experimental results. In the simulation part, the results show how different parameters affect tracking performance. The picked solution from the simulation is then applied to the experiment.

6.1 Simulation setup

The simulation is set-up in the first-person perspective, and a 3D environment is built up with OpenGL and Pygame. Figure 6 shows four on-board camera views of the OpenGL 3D environment with the edges detected by the Hough algorithm and the line pickup criteria, where the red dash line, blue dash line, solid orange line, and solid green line are the direction line, vanishing line, detect vector, and target line, respectively (Table 1). The position of the vehicle is listed in Fig. 6 for the screenshots (1), (2), (3), and (4). All parameters for the ROV (BlueROV2) in simulations are all verified by the experiments reported in Pedersen et al. (2019); Wu (2018):

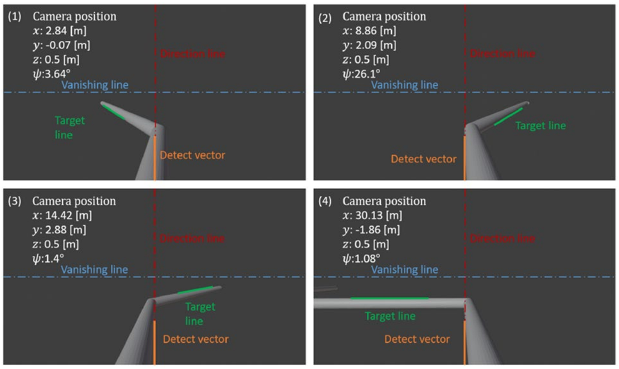

Fig. 6 The 3D simulation environment

6.2 Simulation results

In this paper, the control variable method is used to test each parameter's function to the ROV tracking performance. According to the control variable method, only one variable changes for each simulation. Figure 7 includes the cases with different initial positions and heading angles but a designed path. Nine initial conditions (ICs) are shown in Table 2 (ICs in the legends of Fig. 7). For the same path, the tracking results with different safety angles, detect vector lengths, switch coefficients, simulation control time delays (time interval for each iteration) are shown in Fig. 7, where the detect vector's amplification ratio is based on its actual length in the world coordinates (the default vector is $O_w L_{wt}$ in Fig. 4C).

The first sub-figure in Fig. 7 shows the tracking results with the nine different initial positions and heading angles.

Table 1 Simulation parameters

m	I_z	F_{max}
11.17 kg	0.24 kgm^2	40N
$X_{\dot{u}}$	$Y_{\dot{v}}$	$N_{\dot{r}}$
−5.5 kg	−5.5 kg	−0.12 kgm^2/rad
X_u	Y_v	N_r
−4.03 Ns/m	−6.22 Ns/m	−0.07 Ns/rad
$X_{u\|u\|}$	$Y_{v\|v\|}$	$N_{r\|r\|}$
−18.18 Ns^2/m^2	−21.66 Ns^2/m^2	−1.55 Ns^2/rad^2

Table 2 Initial conditions

Name	x[m]	y[m]	z[m]	ψ [rad]
IC0	0.0	1.0	0.5	– 0.45
IC1	0.0	1.0	0.5	– 0.7
IC2	0.0	2.0	0.5	– 0.45
IC3	0.0	2.0	0.5	– 0.7
IC4	0.0	– 1.0	0.5	0.35
IC5	0.0	– 1.0	0.5	0.6
IC6	0.0	– 2.0	0.5	0.35
IC7	0.0	– 2.0	0.5	0.6
IC8	0.0	0.0	0.5	0.0

The differences among the cases are reduced significantly in the second pipeline segment. As long as other parameters have proper values and the pipeline is in the camera view initially, the initial conditions do not affect the tracking performance too much. When it is close to the right angle, the critical factor is the selection of safety angles based on Eq. 40. In Fig. 7, the first stage safety angle does not change the tracking result with a small angle change. When the tracking angle changes $\frac{\pi}{2}$, and the distance between the vehicle and the line is short, the first safety angle only changes the target angle. When the only changing parameter is the detect vector amplification ratio, the tracking is terminated at the right angle area when it is 0.3 times the default length $(O_w L_{wt})$ because the pipeline is outside of the view. Though the ratio increases to 0.5, the fluctuation is high at the right angle area. In the simulation with different switch coefficients, c_s does not affect the performance when it is no less than 4. However, when it is 2 or even smaller, $\frac{\pi}{c_s}$ is large, and the influence of the angle difference between two neighbor sample times is negligible. According to the target line switch criteria in Eq. 29, a new line is picked up when the current target line is short enough in the image view. If the length of the line has the most impact ratio to decide the target line, the Hough algorithm will introduce uncertainty to the tracking performance. The second stage safety angle only takes effect when the vehicle is outside of the first stage area. The plots in Fig. 7 have differences only when the ROV is near the right angle lines. In Fig. 7, when the time delay increases from 0.05 s to 0.1 s, the deviation of the trajectory is not negligible. When the delay is 0.2 s and 0.5 s, the tracking is terminated before the first turn. Based on the comparison and discussion of the simulation results, the test parameters and their optimal values are summarized in Table 3,

The performance of the vehicle model under different kinds of disturbances is shown in Fig. 8. In Fig. 8, the ROV follows the x-axis. There is a disturbance in $x \in [0.5, 3.5]$. The red, blue, and black lines represent the random, sine, and constant disturbances, respectively, where the dash line is the disturbance (right y-axis) and the solid line is the trajectory under the disturbance (left y-axis). In the simulation, the trajectory is a sine wave under the sine disturbance. The ROV has a steady-state error while moving under the constant error. For the random error, the deviation of the trajectory is also random. According to the mathematical model and the controller design, the ROV tracking method in this paper cannot fully reject the disturbance but with a reasonable amount of steady-state error. The goal of this

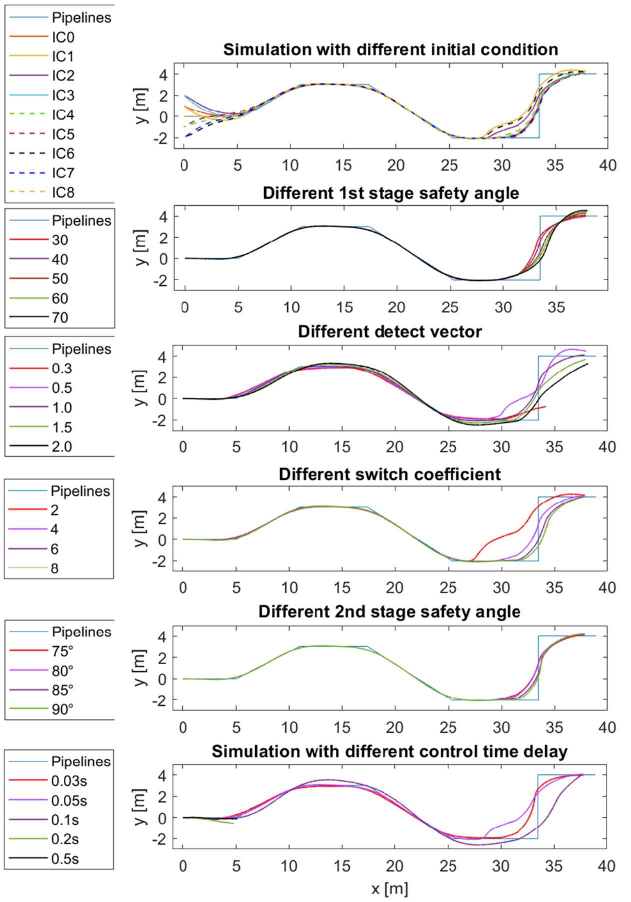

Fig. 7 Tracking results with different initial condition and controller parameters and control time delay

Table 3 Tested values and the picked values for all parameters

Parameters	Picked	Tested
Initial position	(0, 0)	***IC*** 0–8
Initial heading direction	0	***IC*** 0–8
1st safety angle ψ_{s1} [^{o}]	40	30, 50, 60, 70
Detect vector amplify ratio	1	0.3, 0.5, 1.5, 2
Switch coefficient c_s	4	2, 6, 8
2nd safety angle ψ_{s2} [degree]	75	80, 85, 90
Control time delay [s]	0.05	0.03, 0.1, 0.2, 0.5

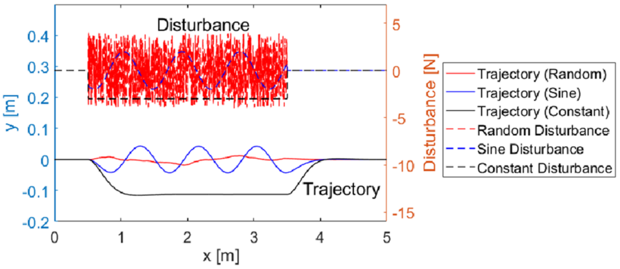

Fig. 8 The simulation of ROV model under different disturbance in the range $x \in [0.5, 3.5]$, where the dash lines are disturbance, and the solid lines are trajectories

method is to control the ROV following the pipeline, thus the pipeline line must be inside the camera view of the ROV so that any damage can be detected. In the disturbance zone, the ROV deviates from its following path and the end of its detect vector (r_c) is on the following path. Under that circumstance, the control error $\psi_e \to 0$, but the heading angle (ψ is not zero). The forward thrust in the y-axis cancels the disturbance, and the force in the x-axis drives the ROV to move along its following path. The deviation in the y-axis and the heading angle together make sure that the pipeline is inside the camera view. This behavior will be verified through an experiment in the next subsection.

6.3 Experimental setup

In the experiment, the testing vehicle was BlueROV2 from Blue Robotics Inc. The camera in the front of the ROV had 80° horizontal field of view and $+/-90°$ tilt angle range. In Fig. 9, the pool's size was $10\ \mathrm{m} \times 5\ \mathrm{m} \times 1.2\ \mathrm{m}$, and the water level was around 1 m. The black PVC pipelines were laid down at the bottom of the pool, where the two bending angles were 45° and 90°, respectively. The jet was located at the top right place of the pool.

The control of the ROV was based on the ArduSub project, which is currently a fully-featured and open source solution for ROVs. The communication between the Ardusub and the vehicle was through a protocol, called MAVlink. In the experiments, the protocol was implemented using python. The thrusters and the ROV motion were controlled by a pulse width modulation (PWM) value. The MAVlink pre-defined PWM output range was [1100, 1900], where 1500 was the mean value, 1900 was the positive maximum, and 1100 was the negative maximum. The PWM control function was integrated to the keyboard for manual control, a mode switch to automatic pipeline tracking, and emergency stop. The time interval for each control loop was set to 0.1 s, which was enough for image processing, controller calculation, and vehicle communication.

In the automatic control process, first, with the jet off, the performances between the model-based nonlinear controller in Eq. 34 and a PD controller were compared. The control signal was

$$T_{PD} = -K_D(r - \dot{\psi}_n) - K_P(\psi - \psi_n).$$

The value of gains in this experiment were $K_D = d = 7$, $K_P = c = 1$. The PWM was mapped from the control torque as $\frac{180T}{\pi}$, where T is T_{PD} or T_ψ. Then, the jet was turned on and the tracking algorithm was tested under the disturbance from the jet flow.

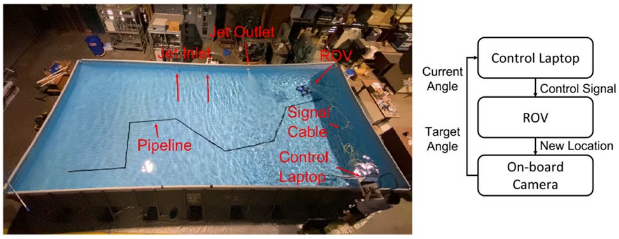

Fig. 9 The experimental setup, where the left is the above view, and the right side is the block diagram

6.4 Experimental results

At the beginning of the experiment, the ROV moved to the initial position of the pipeline, which was the right-upper corner of the pool. After the ROV switched to the "automatic mode", it started to track the pipeline automatically.

The four figures in Fig. 10 are the screenshots of the ROV on-board camera, which are also the original image for the Hough line detection algorithm, where the time, ψ, ρ are at the left upper corner, h from Fig. 4 is unknown. The green line is the detected result for each image, which is also regarded as the target line. According to Sect. 4.1, the target line is the x-axis in the world coordinates. The right-upper corner is the current detected ψ and ρ from the Hough transformation domain in the world coordinates.

The states of ROV are shown in Fig. 11. In the above figure, the angle relative to the current target line ψ, the target angle ψ_n, control error ψ_e, and the ρ from the Hough algorithm in the world coordinate are blue, red, yellow and black lines, respectively. The control input (PWM value) is the blue line below. The black line has a boolean value, where '1' means the tracking target switches to the next pipeline segment. The solid line is the controller from Eq. 34, and the dash line is the result of the PD controller, respectively. Figure 12 shows the states with the model-based nonlinear under an unknown disturbance from a jet. The jet's location is shown in Fig. 9.

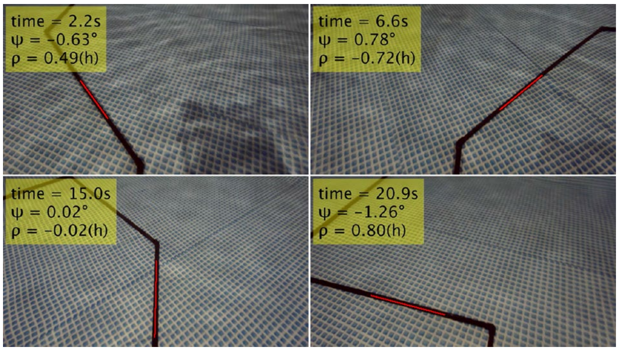

Fig. 10 The snapshots from the on-board camera, where the black lines are detected pipelines and the red line is the target line

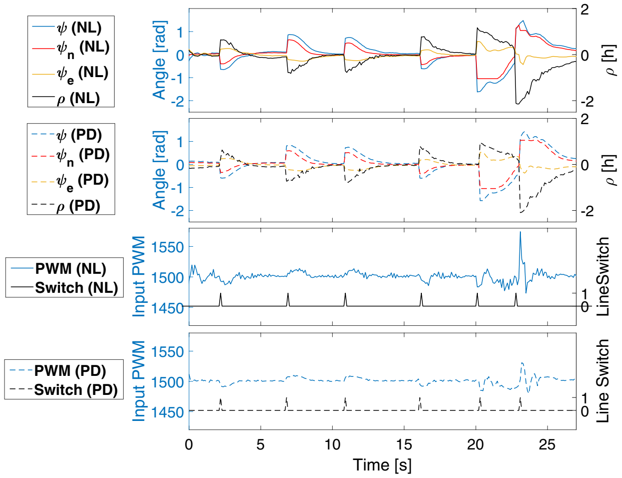

Fig. 11 The states of ROV in the experiment with the model-based nonlinear (NL) and PD controller

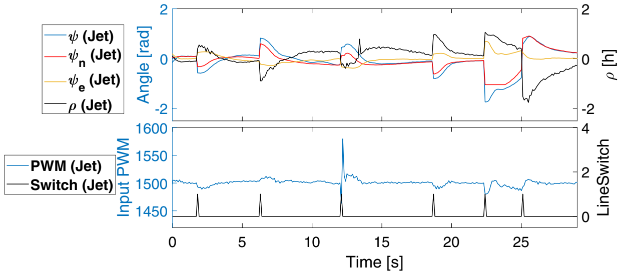

Fig. 12 The states of ROV in the experiment with model-based nonlinear under an unknown disturbance from a jet

6.5 Discussion

To design such a tracking control, one way is to set the controller (e.g., PID control) to regulate y and ψ to their desired values, respectively. However, due to the model differences, it is hard to make sure that the tracking errors for two variables converge to zero simultaneously. The control gain has to be well-tuned. The method in this paper introduces a new variable ψ_n. y and ψ reach to their steady states at the same time if $\psi_n \to 0$.

The experimental results were collected from many repeated tests. Since the initial positions for those tests were not the same based on manual control, it was meaningless to do an average on the tracking states even with the same parameters. However, the trend for each test is similar to Fig. 11. For each pipeline segment, the ROV converged to the target line and remained at the steady state. In the case that the initial angle error between the vehicle and the desired path is very large, one should indicate an initial direction to track. As long as the pipeline is inside the camera view, the tracking can be accomplished even the pipeline is far away from the vehicle. When the target line switched, ψ, ψ_n, ψ_e, and ρ increased sharply and then gradually returned to the steady states. The input PWM value also changed with the new line segment. Its amplitude was related to the angle difference between the previous and current segments. The angle changes were ±45° and ±90°, as shown in Fig. 9. For each line switch, the sudden change of the ψ was regarded as an impulse disturbance in the ROV's heading angle while the ROV was tracking a straight line. Thus, as long as the target line was still in the camera view, the ROV was able to recover from the impulse change in the angle domain.

Compared with the previous experiment (jet off), the jet affected the ROV at the time between 10 s and 15 s in the experiment with the disturbance. At the third line switch place, the change of the PWM value was large at first and then gradually decreases. This is because the target angle was changed from 0° to −45°, and the jet started to push the ROV at the same time. The controller generated a large control signal to maneuver the ROV against the disturbance from the jet and the angle change.

The biggest difference of the experiments compared to the simulation was caused by the environmental noise and errors inside the control loop. In this experiment, the camera lost a few frames due to the signal transmission overwhelming. Under this situation, the feedback information was the prediction based on the previous detected line and Kalman filter (Yi and Chen 2019). During the tracking process, the noise came from the shadow of the ROV itself and water waves on the surface. The brightness of the camera view changed as the relative position between the ROV and the lights varies. Since Hough transformation was based on the color, the brightness change introduced extra error to the line detection result. The histogram had ruled out these errors. In the image process step, the errors came from the pixels allocation and image blurriness due to the sudden shake of the ROV. The control input was not smooth for both controllers because of this error. However, the ROV had an input dead zone, where the propeller did not move when the absolute PWM value was less than 10. Thus the fluctuation of PWM value in Fig. 11 was acceptable, and the ROV accomplished the tracking mission.

7 Conclusions and future work

In this paper, we've developed an image-based visual servoing control for ROVs to follow underwater pipelines, using a monocular camera. We've converted the 2D path-following problem into a 1D trajectory-tracking task using an angle-based planning approach. Our system utilizes the Hough transform to detect lines from the camera feed, and our unique angle projection rules and line pickup criteria to calculate angular states. We designed and analyzed a controller to track these angles. Through OpenGL simulations and real-world tests, we refined parameters for optimal performance, and validated our model-based controller against a well-tuned PD controller.

Future work will focus on enhancing our system's adaptability and performance in complex real-world scenarios. We plan to incorporate more advanced image detection techniques with deep learning algorithms, trained on extensive datasets, to bolster our pipeline detection capabilities. Moreover, more rigorous testing in diverse, challenging underwater environments will be conducted to refine the robustness and reliability of our algorithms. Our ultimate aim is to deliver a robust, adaptable automated underwater pipeline inspection system.

Supplementary Information The online version of this article (https://doi.org/10.1007/s41315-023-00301-2) contains supplementary material, which is available to authorized users.

Acknowledgements This research is supported by Texas Commission on Environmental Quality through Subsea Systems Institute Award #582-15-57593. This project was paid for [in part] with federal funding from the Department of the Treasury through the State of Texas under the Resources and Ecosystems Sustainability, Tourist Opportunities, and Revived Economies of the Gulf Coast States Act of 2012 (RESTORE Act). The content, statements, findings, opinions, conclusions, and recommendations are those of the author(s) and do not necessarily reflect the views of the State of Texas or the Treasury.

Data availability The data published in this paper is publically accessible. Please contact the corresponding author: Dr. Zheng Chen at zchen43@central.uh.edu for the data access.

Declarations

Conflict of interest There is no conflict of interest involved in this article.

References

Aripin, M.K., Aras, M.S.M., Sulaiman, M., Zainal, M.I.M., Harun, M.H., Zambri, M.K.M.: Low cost expansion of unmanned underwater remotely operated crawler (roc) for pipeline inspection. In 2017 IEEE 7th International Conference on Underwater System Technology: Theory and Applications (USYS), pages 1–5. IEEE (2017)

Bharti, V., Lane, D., Wang, S.: A semi-heuristic approach for tracking buried subsea pipelines using fluxgate magnetometers. In: 2020 IEEE 16th International Conference on Automation Science and Engineering (CASE), pages 469–475. IEEE (2020)

Chu, Z., Zhu, D., Luo, C.: Adaptive neural sliding mode trajectory tracking control for autonomous underwater vehicle without thrust model. In: 2017 13th IEEE Conference on Automation Science and Engineering (CASE), pages 1639–1644. IEEE (2017)

Duda, R.O., Hart, P.E.: Use of the hough transformation to detect lines and curves in pictures. Commun. ACM **15**(1), 11–15 (1972)

Duran, O., Althoefer, K., Seneviratne, L.D.: Automated pipe defect detection and categorization using camera/laser-based profiler and artificial neural network. IEEE Trans. Autom. Sci. Eng. **4**(1), 118–126 (2007). https://doi.org/10.1109/TASE.2006.873225

Fossen, T.I.: Handbook of Marine Craft Hydrodynamics and Motion Control. Wiley, Amsterdam (2011)

Gåsvik, K.J.: Optical Metrology. Wiley, Amsterdam (2003)

Heshmati-Alamdari, S., Nikou, A., Dimarogonas, D.V.: Robust trajectory tracking control for underactuated autonomous underwater vehicles in uncertain environments. IEEE Transactions on Automation Science and Engineering (2020)

LaSalle, J.: Some extensions of liapunov's second method. IRE Transactions on Circuit Theory **7**(4), 520–527 (1960)

Pedersen, S., Liniger, J., Sørensen, F.F., Schmidt, K., von Benzon, M., Klemmensen, S.S.: Stabilization of a rov in three-dimensional space using an underwater acoustic positioning system. IFAC-PapersOnLine **52**(17), 117–122 (2019)

Reynolds, C.W.: Steering behaviors for autonomous characters. In Game Developers Conference, volume 1999, pages 763–782. Citeseer (1999)

Sfahani, Z.F., Vali, A., Behnamgol, V.: Pure pursuit guidance and sliding mode control of an autonomous underwater vehicle for pipeline tracking. In 2017 5th International Conference on Control, Instrumentation, and Automation (ICCIA), pages 279–283. Ieee (2017)

Sheng, M., Tang, S., Wan, L., Wang, G., Qin, H., Liu, Y.: A new algorithm for auv pipeline recognition and location. In OCEANS 2018 MTS/IEEE Charleston, pages 1–5. IEEE (2018)

Shi, L., Chen, Z., Guo, S., Guo, P., He, Y., Pan, S., Xing, H., Su, S., Tang, K.: An underwater pipeline tracking system for amphibious spherical robots. In 2017 IEEE International Conference on Mechatronics and Automation (ICMA), pages 1390–1395. IEEE (2017)

Shi, L., Zheng, R., Liu, M., Zhang, S.: Distributed circumnavigation control of autonomous underwater vehicles based on local information. Syst. Control Lett. **148**, 104873 (2021)

Shojaei, K.: Three-dimensional tracking control of autonomous underwater vehicles with limited torque and without velocity sensors. Robotica **36**(3), 374–394 (2018)

Sutulo, S., Soares, C.G.: A unified nonlinear mathematical model for simulating ship manoeuvring and seakeeping in regular waves. In Proc. Int. Conf. Mar. Simul, Ship Manoeuv (2006)

Tian, M., Liu, F., Hu, Z.: Single camera 3d lane detection and tracking based on EKF for urban intelligent vehicle. In 2006 IEEE International Conference on Vehicular Electronics and Safety, pages 413–418. IEEE (2006)

Wang, B., Su, Y., Wan, L., Li, Y.: Modeling and motion control system research of a mini underwater vehicle. In 2009 International Conference on Mechatronics and Automation, pages 4463–4467. IEEE (2009)

Wang, J., Meng, M.Q.-H., Khatib, O.: Eb-rrt: Optimal motion planning for mobile robots. IEEE Trans. Autom. Sci. Eng. **17**(4), 2063–2073 (2020). https://doi.org/10.1109/TASE.2020.2987397

Wu, C.-J.: 6-DoF Modelling and Control of a Remotely Operated Vehicle. PhD thesis, Flinders University, College of Science and Engineering. (2018)

Yi, X., Chen, Z.: A robust visual tracking method for unmanned mobile systems. J. Dyn. Syst. Measure. Control 141(7) (2019)

Yuh, J.: Design and control of autonomous underwater robots: A survey. Auton. Robot. **8**(1), 7–24 (2000)

Zheng, J., Song, L., Liu, L., Yu, W., Wang, Y., Chen, C.: Fixed-time sliding mode tracking control for autonomous underwater vehicles. Applied Ocean Research **117**, 102928 (2021)

Dr. Xiongfeng Yi was born in Wuhan, China in 1993. He received his MS degree in Fluid Mechanics engineering from the Brown University, Providence, RI and BS degree in Mechanical Engineering from the Dalian University of Technology in China. He received his PhD degree in Mechanical Engineering from the University of Houston in Dec 2021. His doctoral research was focused on modeling, localization, navigation, and visual servoing control of Remotely Operated Vehicle (ROV) and 4-DOF robotic arm. After graduation, Dr. Yi joined ArcBest Corp as a robotics engineer in 2022.

Zheng Chen received the B.E. degree in electrical engineering and the M.E. degree in control science and engineering from Zhejiang University, Hangzhou, China, in 1999 and 2002, respectively, and the Ph.D. degree in electrical engineering from Michigan State University, East Lansing, MI, USA, in 2009. He was a Research Associate with the University of Virginia, Charlottesville, VA, USA, from 2009 to 2012, a Research and Development Engineer specializing in control systems with Baker Hughes from 2012 to 2013, and an Assistant Professor of electrical engineering and computer science with Wichita State University, Wichita, KS, USA, from 2013 to 2017. He is currently an Associate Professor of mechanical engineering and the director of Bio-inspired Robotics and Controls Lab (BRCL) with the University of Houston, Houston, Texas. His current research interests include smart material sensors and actuators and bio-inspired underwater robots.