

PID-Based Path Following for Turning Circle Manoeuvres of an Unmanned Surface Vehicle in Gazebo

SINGH, Yogang and OBAJINMI, Oluwadurotimi

Available from Sheffield Hallam University Research Archive (SHURA) at:

<https://shura.shu.ac.uk/36275/>

This document is the Accepted Version [AM]

Citation:

SINGH, Yogang and OBAJINMI, Oluwadurotimi (2026). PID-Based Path Following for Turning Circle Manoeuvres of an Unmanned Surface Vehicle in Gazebo. In: DOROFTEI, Ioan, BAUDOIN, Yvan, TAQVI, Zafar and KELLER FUCHTER, Simone, (eds.) Measurements and Control in Robotics. Proceedings of the 26th International Symposium on Measurements and Control in Robotics (ISMCR 2025). Mechanisms and Machine Science (199). Cham, Springer, 121-134. [Book Section]

Copyright and re-use policy

See <http://shura.shu.ac.uk/information.html>

PID-Based Path Following for Turning Circle Manoeuvres of an Unmanned Surface Vehicle in Gazebo

Oluwadurotimi Obajinmi¹ and Yogang Singh²

^{1,2}School of Engineering and Built Environment,
Sheffield Hallam University, Sheffield S1 1WB, England, United Kingdom
OluwadurotimiObajinmi@gmail.com and y.Singh@shu.ac.uk

Abstract.

This paper presents a simulation study of path following for the *Heron* Unmanned Surface Vehicle (USV) using a Line-of-Sight (LOS) guidance approach. The tests were performed on both straight-line and circular paths to evaluate the USV's ability to converge to and track the desired trajectory. Simulations were conducted in the Robotic Operating System (ROS)–*Gazebo* environment, where the LOS algorithm was combined with a Proportional–Integral–Derivative (PID) controller to regulate heading and reduce cross-track error. Controller gains were tuned using the *Ziegler–Nichols* method and refined through manual adjustments, enabling stable and accurate turning performance. The results demonstrate that the PID-based LOS guidance method provides effective and reliable path following for underactuated USVs and other marine vehicles.

Keywords: Line-of-Sight, PID, Path-following, Turning circle, Unmanned Surface Vehicle, ROS, *Gazebo*.

1 Introduction and Literature Review

Autonomous marine vehicles, such as USVs and Autonomous Underwater Vehicles (AUVs), have become essential tools for applications ranging from oceanographic research and environmental monitoring to maritime security [1], [2]. A key requirement for these platforms is the ability to maintain accurate and stable course-keeping during missions, even under the influence of external disturbances such as wind, waves, and currents. In manned vessels, this responsibility lies with the helmsman; in USVs, it must be achieved through reliable guidance and control algorithms.

Path following is the task of ensuring a marine vehicle converges to and tracks a predefined path as accurately as possible until the mission is complete. This differs from trajectory tracking, where the vehicle must follow a time-parameterized reference with strict timing requirements, a concept more common in mobile robotics and land vehicles [7], [8]. For underactuated marine systems, path following remains an active area of research due to the challenges introduced by their non-holonomic constraints and limited actuation [8].

Several guidance strategies have been developed to address these challenges, with LOS guidance emerging as one of the most widely used methods. LOS guidance relies on geometric properties of the reference path and generates a desired heading for the vessel to converge smoothly to the path [3]. Researchers have applied LOS guidance in both straight-line and circular path-following scenarios, and extensions such as integral LOS have been proposed to improve robustness against ocean currents [11], [13]. In addition, [13] recently proposed enhancements to LOS to improve autonomous navigation for surface vessels in dynamic marine environments.

Accurate heading control is critical for effective LOS-based path following. Various controllers have been studied to achieve this, with PID controller being the most common due to its simplicity, effectiveness, and ease of implementation [5], [15]. Enhanced versions of PID have also been developed, such as S-PID [1] and fuzzy-adaptive PID [9], to further improve course-keeping performance. More advanced control methods, including Model Predictive Control (MPC), Linear Quadratic Regulation (LQR), sliding mode control, and backstepping, have also been integrated with LOS guidance for improved robustness [10]–[12].

Turning circle manoeuvres are a standard test of marine vessel manoeuvrability, providing insight into a vessel's ability to change course safely and efficiently [4]. For underactuated USVs such as the *Heron* platform, path-following during

turning manoeuvres is particularly challenging due to sideslip effects. To address this, [3] introduced the use of a sideslip angle in LOS guidance to correct heading and improve course convergence.

In this study, the *Heron* USV is simulated within the ROS–Gazebo environment to evaluate LOS guidance combined with a PID controller for path-following tasks, with emphasis on turning circle manoeuvres. The PID controller is used to regulate heading error after the LOS guidance law determines the desired course, with the resulting control output applied as angular velocity to the thrusters. In addition to validating the LOS–PID framework against the approach described in [3], this work assesses robustness under simulated disturbances and provides a baseline comparison with a simpler guidance method, thereby addressing both performance and reliability in autonomous marine navigation.

2 Aims and Objectives

The aim of this experimental simulation is to validate the Line-of-Sight (LOS) guidance method presented in [3] by performing a path-following turning circle manoeuvre on the *Heron* USV within the ROS–Gazebo environment, using a PID-based controller. The objectives are to design the simulation environment for the *Heron* USV in line with the conditions specified in [3], implement a turning circle path using the LOS guidance law combined with a PID controller for heading regulation, and compare the obtained results with those reported in [3]. In addition, the study evaluates the robustness of the approach under simulated environmental disturbances and provides a baseline comparison against a simpler guidance method to demonstrate performance improvements and reliability.

3 Contribution of the Study

The study builds on the work of [3] by implementing their geometric path-following algorithm with a PID controller instead of the original backstepping controller within a ROS–*Gazebo* environment, resulting in faster convergence and reduced settling times for USVs in simulation. Additionally, this study extends the 2D simulation of [3] into a high-fidelity 3D *Gazebo* environment using a USV simulation package, providing a more realistic evaluation of path-following performance.

4 LOS Guidance Algorithm

The LOS guidance algorithm for geometric path following is derived mathematically, with its foundations established by [3]. Further developments and detailed explanations are provided in [16]. The *Heron* USV, being a surface vessel, operates in a two-dimensional plane defined by $(\mathbf{X}_{\text{ref}}, \mathbf{Y}_{\text{ref}})$. As illustrated in Figure 1, the blue point

denotes the current position of the *Heron* USV, while the black point represents a reference position on the geometric path toward which the vessel is required to converge.

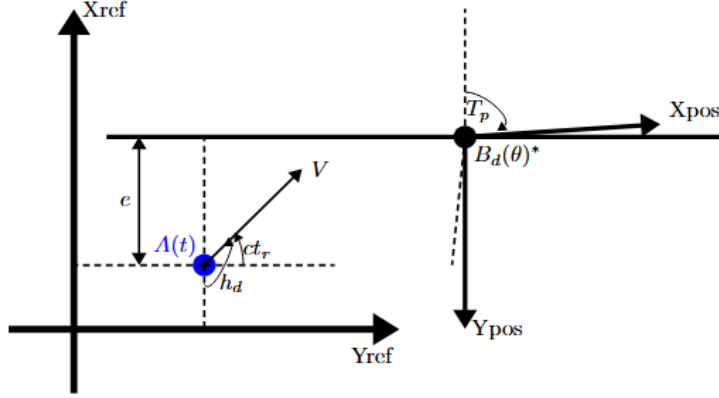


Fig. 1. Line-of-Sight Guidance Algorithm Illustration

The USV is assumed to be located at the position $A(x,y)$ in a two-dimensional space, where x and y represent the coordinates of point A . The USV moves with a velocity vector V , which has magnitude $|V|$ representing the net speed with a direction χ , which corresponds to the course angle or heading of the USV. The velocity magnitude and course angle of the USV can be expressed as

$$|V| = \sqrt{u^2 + v^2} \quad (1)$$

$$\chi = \tan^{-1} \left(\frac{v}{u} \right) \quad (2)$$

Where u denotes the surge velocity (velocity component in the x -direction) and v denotes the sway velocity (velocity component in the y -direction).

On the two-dimensional plane representing the water surface, the reference path can be parameterized by a scalar variable $\theta \in \mathbb{R}$. Any point along this path is expressed as

$$\mathbf{B}_d(\theta) \in \mathbb{R}^2.$$

For a given USV position, there exists a unique point on the path that minimizes the Euclidean distance between the vessel's current location and all points on the path. This point is denoted as θ^* , and can be defined mathematically as

$$\|\mathbf{A} - \mathbf{B}_d(\theta^*)\|_2 \triangleq \|\mathbf{A} - \mathbf{B}_d(\theta)\|_2 \quad (3)$$

At the point $\mathbf{B}_d(\theta^*)$, on the path, a local reference frame—referred to as the **path-parallel frame (Pos)**—is established. This frame is local because it is defined relative to the specific position on the parameterized path. The frame is rotated by an angle with

respect to the global (origin) frame, corresponding to the **heading angle** at that position, which is denoted as

$$P_t(\theta^*) = \tan^{-1} \left(\frac{y'_d(\theta^*)}{x'_d(\theta^*)} \right) \quad (4)$$

$x'_d(\theta^*) = \frac{dx_d}{d\theta}(\theta)$ represents how the USV's position coordinates (x, y) changes with respect to the path parameter. The x -axis of the **Pos** is aligned with the tangential vector to the path at point $\mathbf{B}_d(\theta^*)$.

In the **Pos** frame, the error vector between the USV's current position $\mathbf{A}$ and closest position on the path $\mathbf{B}_d(\theta^*)$ is represented as

$$\boldsymbol{\varepsilon} = \mathbf{R}_p^T(P_t)(\mathbf{A} - \mathbf{B}_d(\theta^*)) \quad (5)$$

Where P_t is the angle of rotation or the path tangential angle, and $\mathbf{R}_p(P_t)$ is the rotational matrix from the reference frame to the **Pos** frame, given by

$$\mathbf{R}_p(P_t) = \begin{bmatrix} \cos P_t & -\sin P_t \\ \sin P_t & \cos P_t \end{bmatrix} \quad (6)$$

As described in [16], the error vector consists of two components: the along-track error s and the cross-track error e . The **cross-track error** represents the shortest distance from the USV to the path. The LOS guidance method, as formulated in [3], aims to reduce the cross-track error to zero, ensuring that the USV accurately follows the path. The rate of change of the cross-track error can be expressed as.

$$\dot{e} = U \sin(\chi - P_t) \quad (7)$$

where, $U = |\mathbf{V}|$ and χ , is the heading angle of the USV

Ideally a desired course angle (h_d) can be assigned to the course angle of the velocity vector thereby making $\chi = h_d$. Hence equation (7) becomes.

$$\dot{e} = U \sin(h_d - P_t) \quad (8)$$

This equation is the virtual input for stabilizing cross-track error (e). It is the relative angle between the desired course and the path-tangential course, which can be represented as

$$ct_r = h_d - P_t \quad (9)$$

This is dependent on the cross-track error such that $ct_r = ct_r(e)$.

$$ct_r(e) = \tan^{-1} \left(\frac{-e}{\delta} \right) \quad (10)$$

This is commonly referred to as the LOS angle, where δ is called the look ahead distance, and it is used to guide the marine vehicle as it converges to the path.

Making the desired course angle the subject of formulae from equation (9).

$$h_d(\theta^*, e) = P_t(\theta^*) + ct_r(e) \quad (11)$$

This equation represents the core formulation of the LOS guidance method. Using this approach, the USV—as well as other marine vessels—can converge to a geometric path on the two-dimensional water surface, thereby achieving accurate path-following.

4.1 LOS design for a straight-line path

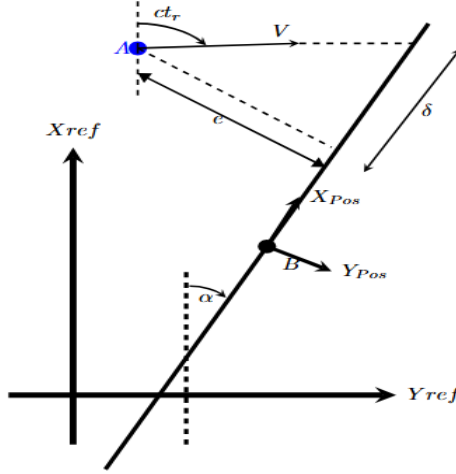


Fig. 2. LOS Guidance Algorithm for an angled straight path.

Figure 2 represents a path with an angle of rotation α . The USV located at position A is represented by a blue point, while position B denotes an arbitrary point on the path that the USV must converge to, shown as the black point. The angle of rotation of the straight line is defined relative to the origin of the 2D environment X_{ref} and Y_{ref} . Therefore, the cross-track error vector for this path is expressed as

$$\mathbf{e} = \mathbf{R}_p^T(\alpha)(\mathbf{A} - \mathbf{B}) \quad (12)$$

The desired heading for the LOS guidance along a straight path with a given rotation angle is obtained by substituting path rotation angle with P_t . This yields the resulting expression as:

$$h_d(e) = \alpha + ct_r(e) \quad (13)$$

This guidance formula is used to compute the desired heading of a marine vessel as it follows a geometric straight-line path, based on the specified path tangential angle.

4.2 LOS design for a circular path

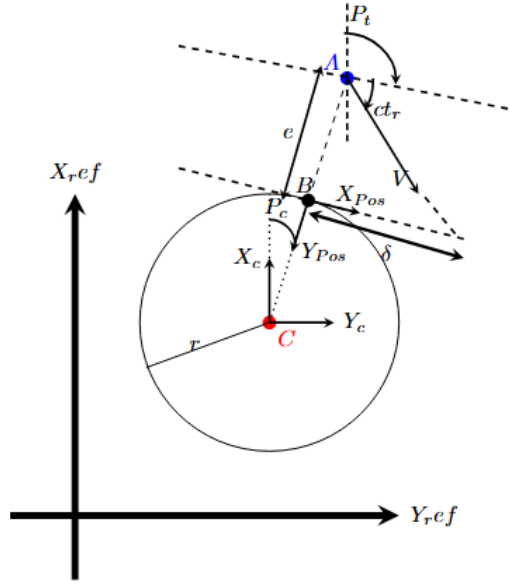


Fig. 3. Circular path illustration for the LOS algorithm

For the circular Illustration the blue point is the position of the USV denoted as **A**, the black point is a position on the circle **B**, the red point is the center of the circle denoted as **C**.

For a circular path, the heading angle of a point relative to the origin frame and vessel P_t changes with time therefore it can be denoted as

$$P_t(t) = P_c(t) \pm \frac{\pi}{2} \quad (14)$$

Where the positive sign correlates with a clockwise circular motion and the negative sign an anticlockwise motion.

$P_c(t)$ represents the path tangential angle of the USV with respect to the center of the circle (x_c, y_c) and the USV position $(x(t), y(t))$ and it is represented as

$$P_c(t) = \tan^{-1} \left(\frac{y(t) - y_c}{x(t) - x_c} \right) \quad (15)$$

The cross-track error is computed by first calculating the Euclidean distance between the USV and the center of the circular path. The circle's radius is then subtracted from

this distance, yielding the deviation of the USV from the circumference. This value indicates how far the USV is from—or how close it is to—the desired circular path.

$$e(t) = r - \sqrt{(x(t) - x_c)^2 + (y(t) - y_c)^2} \quad (16)$$

The LOS guidance for a circular path is derived by substituting the path tangential angle and the cross-track error angle into the standard LOS guidance formula. This can then be expressed as follows:

$$h_d = \tan^{-1} \left(\frac{y(t) - y_c}{x(t) - x_c} \right) \pm \frac{\pi}{2} + \tan^{-1} \left(\frac{-e}{\delta} \right) \quad (17)$$

4.3 Slip angle

During motion, USVs tend to drift away from their desired course due to their limited control capabilities. To compensate for this, the slip angle of the USV must be calculated and incorporated into the LOS guidance, ensuring accurate guidance, navigation, and control. The slip angle is defined as the angle between the direction of motion of the USV and the direction in which it is pointing. It is related to the course angle and heading of the vessel and can be expressed as:

$$\chi = \psi + \beta \quad (18)$$

Where χ is the course angle (h_d) from the LOS guidance, ψ is the desired heading, and the sideslip angle is β .

The sideslip angle or drift angle is the arc tangent of the sway velocity and the surge velocity of the USV and is given as

$$\beta = \tan^{-1} \left(\frac{v}{u} \right) \quad (19)$$

The desired heading to control the USV for accurate convergence is expressed as:

$$\psi = h_d - \beta \quad (20)$$

5 Control System Design

The control system employed in this study is a PID controller, commonly used to minimize the error between desired and actual outputs in both linear and nonlinear systems. The input–output relationship of the system is illustrated in the block diagram shown in Figure 4.

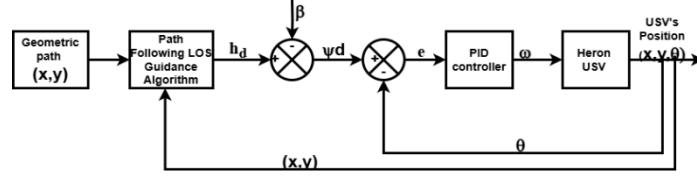


Fig. 4. System Block Diagram.

The PID controller minimizes the **heading error**, defined as the difference between the desired heading of the USV—calculated from the LOS guidance method and the slip angle—and the actual orientation of the USV, represented by the yaw angle θ , which corresponds to rotation about the z -axis. The controller computes a weighted sum of this error using the PID gains K_p , K_i and K_d . The resulting output is the angular velocity, which is applied to the *Heron* USV thrusters, thereby controlling both its orientation and position along the desired path.

$$e(t) = \psi - \theta \quad (21)$$

$$\omega(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt} \quad (22)$$

Where ψ is the desired heading, θ is, the USV's yaw orientation, $e(t)$ denotes the error input to the PID controller, and $\omega(t)$ is the controller's output, corresponding to the angular velocity.

For the *Heron* USV experiments, the PID gains were tuned through a combination of the *Ziegler–Nichols* method and simulation-based fine-tuning in ROS-*Gazebo* to ensure stability and fast convergence. The selected values of the gains for straight line path following experiment were $K_p = 1.95$, $K_i = 0.06$ and $K_d = 0.5$ while for the Turning Circle experiment were $K_p = 0.9$, $K_i = 0.02$ and $K_d = 0.5$.

6 Methodology and Results

This study was conducted in a 3D simulation environment using *Gazebo Classic* (*Gazebo* 7), which runs on the ROS Kinetic distribution. ROS is a collection of libraries, tools, and frameworks that enable software development for robotic systems. ROS Kinetic, the tenth ROS release, is compatible with Ubuntu 16.04 [17].

For this study, *Ubuntu 16.04* was installed as a guest operating system on a VMware Workstation virtual machine. The host system was a *Windows 11 Acer Aspire 5 laptop with Intel Iris Xe graphics, an 11th-generation Core i7 processor running at 2.5 GHz, and 20 GB of RAM*. The virtual machine was configured to run the Unix-based OS to carry out the simulation experiments.

The USV used in this study is a *Heron*, developed by *Clearpath Robotics*. It is designed for ROS compatibility, specifically with the ROS Kinetic distribution [18]. The necessary packages and dependencies were obtained from the *Heron GitHub* repository [19] to ensure proper functionality within ROS and *Gazebo*. The *Heron* USV is a twin hull catamaran-style vessel, measuring 1.35 m in length and 0.98 m in width as shown in Figure 5 [20]. During the simulation, results were recorded in a ROSbag file, which was later imported into MATLAB for further analysis and plotting

Table 1. *Heron* USV Parameters and Dimension (Source:[18])

<i>Heron</i> USV Parameter	Dimensions
Length	1.35m
Width	0.98m
Height	0.32m
Mass	28kg



Fig. 5. *Heron* USV on water. (Source:[18])

6.1 Benchmarking experiment for straight line path following

To validate the research of [3], the study begins with a benchmarking approach using a straight-line path with cross-track error serving as the primary metric for assessing path-following performance. The path was created following the original experimental setup, with a rotation angle $\alpha = \frac{\pi}{4}$ degree relative to the origin of the earth fixed frame. The USV was initially positioned at $x=10$ m and $y=0$ m, with a yaw rate of 1.3 rad/s. The look-ahead distance was set to 5 times the vessel's length (6.25 m for the *Heron* USV). Results from [3] were digitized and plotted in MATLAB alongside the ROS simulation

results, as shown in Figure 6. [3] evaluated the *Cybership II* for its ability to follow a linear path from the origin while maintaining alignment with the desired trajectory.

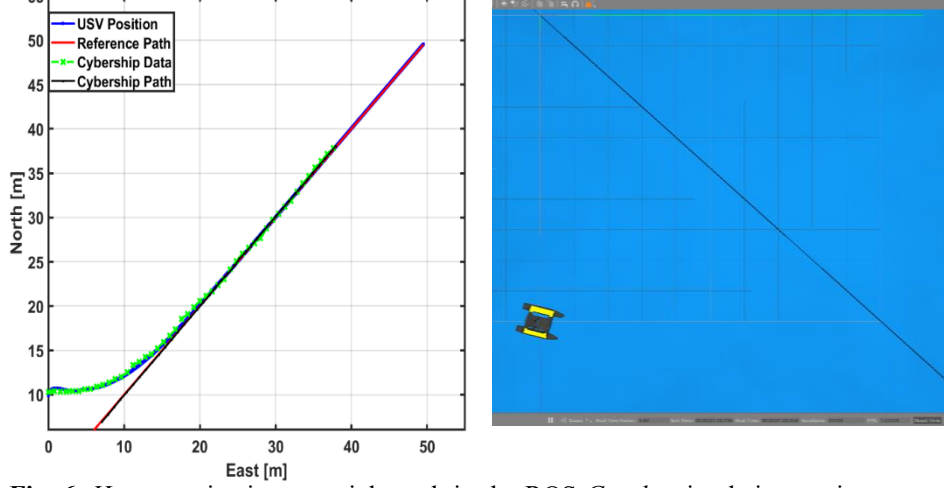


Fig. 6. *Heron* navigating a straight path in the ROS-Gazebo simulation environment (left) with comparison of the results for a straight-line trajectory with *Cybership II* from [3] (right)

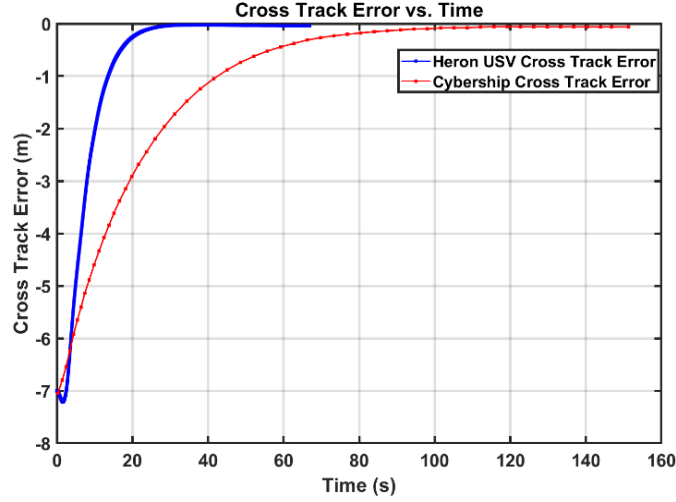


Fig. 7. Cross-track error response of the *Heron* USV compared with the *Cybership II*, showing successful convergence to the desired path

Figure 7 presents the cross-track error response of the *Heron* USV during path convergence, alongside that of the *Cybership II* for comparison. Both responses exhibit an exponential rise from negative values before stabilizing at zero, confirming successful convergence to the desired path. Notably, the *Heron* USV achieved convergence more quickly than the *Cybership II*, attributed to the difference in control strategies: the *Heron* employed a PID controller, while the *Cybership II* used a backstepping controller. This distinction in controller design significantly influenced the rate of convergence

to the straight-line path. This demonstrates the superior performance of the PID controller in minimizing lateral deviation for straight-line path-following tasks.

6.2 Turning Circle experiment using LOS algorithm

The approach was further extended for geometric convergence using cross track error as metric for autonomous Turning Circle manoeuvre. The paths were parameterized in ROS using a set of Pose-stamped message types, with the circle parameters derived from the equation of a circle. The turning circle path was defined with an advance length of 5.40 meters, a circular radius of 2.7 meters, and a tactical diameter of 5.40 meters, following the International Maritime Organisation (IMO) standard for marine vessels, which recommends that the advance distance should not exceed five times the vessel's length as shown in Figure 7. For the *Heron* USV, with a length of 1.35 meters, the advance distance and tactical diameter were set to four times the vessel's length for this experiment. Figure 8 illustrates the turning circle path-following performance of the *Heron* USV. The left panel presents a MATLAB-generated plot of the USV trajectory (red) overlaid on the planned circular waypoint path (blue). The start and end positions are clearly marked, highlighting the vessel's ability to accurately follow the intended path with minimal deviation. The right panel shows a corresponding ROS-Gazebo simulation, where the *Heron* USV navigates the same circular waypoint path in a realistic simulated environment. Together, these visualizations demonstrate the effectiveness of the path-following controller in maintaining trajectory tracking both in planning and in real-time execution.

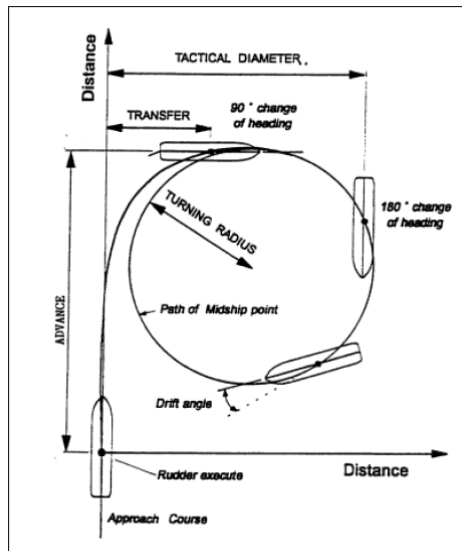


Fig. 7. Turning Circle definition based on IMO requirements

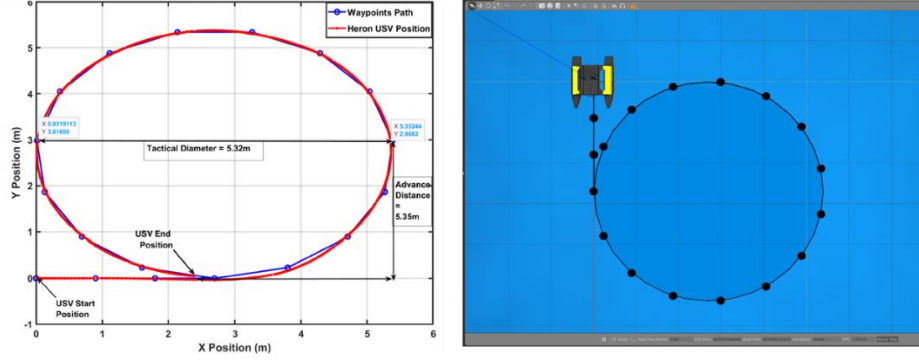


Fig. 8. Turning Circle path-following performance of the *Heron* USV. Left: MATLAB plot showing the USV trajectory (red) against the planned waypoints path (blue) with start and end positions indicated. Right: ROS-Gazebo simulation view of the USV following the circular waypoint path.

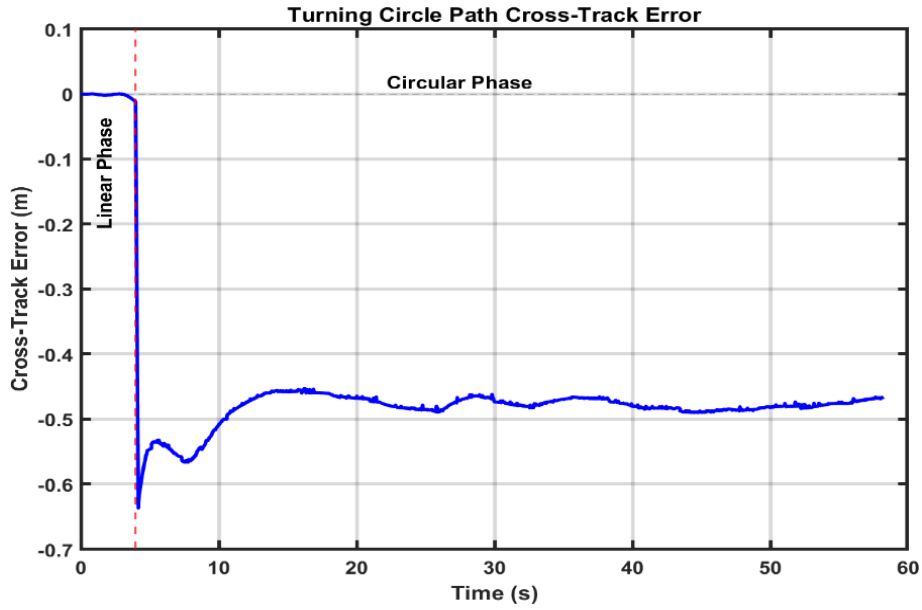


Fig. 9. Cross track error of the *Heron* USV for the Turning Circle experiment

Figure 9 shows the cross-track error of the *Heron* USV during the turning circle test. The trajectory is divided into two phases: the linear phase and the circular phase. During the linear phase, corresponding to the advance motion before initiating the circular turn, the cross-track error remains approximately zero, indicating accurate straight-line tracking. As the USV enters the circular phase, the cross-track error exhibits oscillations that gradually settle, reflecting the dynamic response of the vessel while following the circular path. These results validate the performance of the LOS guidance law and the theoretical framework presented by [3]. A detailed step-by-step procedure for

reproducing the experimental simulation is provided on the project *GitHub* page [21], with supporting video results available at [22, 23].

7 Conclusion and Future Work

This study has demonstrated that LOS guidance method, when combined with a PID controller, effectively ensures convergence and accurate path following for underactuated marine vehicles, specifically USVs. Implementing the PID controller in a simulated ROS-*Gazebo* environment resulted in faster convergence and reduced settling times, as observed in the *Heron* USV (25s) and *Cybership II* (105s). The approach was further validated using a more complex Turning Circle experiment, where the LOS method performed reliably. This study establishes a foundation for further research and can be extended beyond simulation to real-world deployments. Future studies should integrate GPS and IMU sensors to provide odometry data, enabling the USV to autonomously generate and follow course maps. From a control perspective, the algorithm may be further improved by incorporating advanced nonlinear control strategies such as LQR and MPC, among others, to enhance robustness and performance under varying environmental conditions.

References

- [1] R. Miao, Z. Dong, L. Wan, and J. Zeng, “Heading control system design for a micro-USV based on an adaptive expert S-PID algorithm,” *Polish Maritime Research*, vol. 25, no. 2, pp. 6–13, 2018, doi: 10.2478/pomr-2018-0049.
- [2] R. Zhang, Y. Liu, and E. Anderlini, “Robust trajectory tracking control for unmanned surface vessels under motion constraints and environmental disturbances,” *Proc. Inst. Mech. Eng. Part M: J. Engineering for the Maritime Environment*, vol. 236, pp. 394–411, 2021, doi: 10.1177/14750902211039663.
- [3] M. Breivik and T. I. Fossen, “Path following of straight lines and circles for marine surface vessels,” *IFAC Proc.*, vol. 37, pp. 65–70, 2004, doi: 10.1016/S1474-6670(17)31709-3.
- [4] H. Liu, Y. Yang, and S. Yin, “Study on the turning characteristics and influencing factors of the unmanned sailboat,” *J. Marine Science and Engineering*, vol. 12, no. 8, p. 1374, 2024, doi: 10.3390/jmse12081374.
- [5] Z. Świder, L. Trybus, and A. Stec, “Consistent design of PID controllers for an autopilot,” *Polish Maritime Research*, vol. 30, no. 1, pp. 78–85, 2023, doi: 10.2478/pomr-2023-0008.
- [6] A. M. Lekkas and T. I. Fossen, “A time-varying lookahead distance guidance law for path following,” in *Proc. 9th IFAC Conf. Maneuvering and Control of Marine Craft*, vol. 45, no. 27, pp. 398–403, 2014, doi: 10.3182/20120919-3-IT-2046.00068.

- [7] A. M. Lekkas and T. I. Fossen, *Line-of-Sight Guidance for Path Following of Marine Vehicles: Advances in Marine Robotics*. Saarbrücken, Germany: Lambert Academic Publishing, 2013.
- [8] M. Bibuli, M. Caccia, L. Lapierre, and G. Bruzzone, “Guidance of unmanned surface vehicles: Experiments in vehicle following,” *IEEE Robotics & Automation Magazine*, vol. 19, no. 3, pp. 92–102, 2012, doi: 10.1109/MRA.2011.2181784.
- [9] Y. Li, Y.-Q. Jiang, L.-F. Wang, J. Cao, and G.-C. Zhang, “Intelligent PID guidance control for AUV path tracking,” *J. Central South University*, vol. 22, no. 9, pp. 3440–3449, 2015, doi: 10.1007/s11771-015-2884-0.
- [10] I. A. Jimoh and H. Yue, “Path following model predictive control of a coupled autonomous underwater vehicle,” in *Proc. 15th IFAC Conf. Control Applications in Marine Systems, Robotics and Vehicles (CAMS)*, vol. 58, no. 20, pp. 183–188, 2024, doi: 10.1016/j.ifacol.2024.10.052.
- [11] E. Borhaug, A. Pavlov, and K. Y. Pettersen, “Integral LOS control for path following of underactuated marine surface vessels in the presence of constant ocean currents,” in *Proc. IEEE Conf. Decision and Control (CDC)*, Cancun, Mexico, 2008, pp. 4981–4991, doi: 10.1109/CDC.2008.4739352.
- [12] S. Moe, W. Caharija, K. Y. Pettersen, and I. Schjølberg, “Path following of underactuated marine surface vessels in the presence of unknown ocean currents,” in *Proc. 10th IFAC Symp. Nonlinear Control Systems (NOLCOS)*, vol. 49, no. 18, pp. 589–593, 2016, doi: 10.1016/j.ifacol.2016.10.229.
- [13] Ø. K. Kjerstad and E. M. Coates, “Enhancing line-of-sight guidance to improve path following for marine surface vessels,” in *Proc. 15th IFAC Conf. Control Applications in Marine Systems, Robotics and Vehicles (CAMS)*, vol. 58, no. 20, pp. 1–8, Blacksburg, USA, 2024, doi: 10.1016/j.ifacol.2024.10.024.
- [14] T. I. Fossen, M. Breivik, and R. Skjetne, “Line-of-sight path following of underactuated marine craft,” in *Proc. 6th IFAC Conf. Maneuvering and Control of Marine Craft (MCMC)*, vol. 36, no. 21, pp. 211–216, Girona, Spain, 2003, doi: 10.1016/S1474-6670(17)37809-6.
- [15] X. Wang, H. Yi, J. Xu, C. Xu, and L. Song, “PID controller based on improved DDPG for trajectory tracking control of USV,” *J. Marine Science and Engineering*, vol. 12, no. 10, pp. 1771–1780, 2024, doi: 10.3390/jmse12101771.
- [16] T. I. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*, 1st ed. Chichester, U.K.: Wiley, 2011.
- [17] ROS.org, “Ubuntu install of ROS Kinetic,” Accessed: Sept. 15, 2025. [Online]. Available: <https://wiki.ros.org/kinetic/Installation/Ubuntu>
- [18] C. Bogden, “Heron USV gets a new simulator,” *Clearpath Robotics Blog*, 2019. [Online]. Available: <https://clearpathrobotics.com/blog/2019/01/heron-usv-gets-a-new-simulator/>

- [19] T. Baltovski, “Heron packages,” *GitHub Repository*, 2022. [Online]. Available: <https://github.com/heron>
- [20] Clearpath Robotics, “Heron datasheet,” [Online]. Available: https://pdf.directindustry.com/pdf/clearpath-robotics/heron/177123-660678-_2.html
- [21] O. Obajinmi, “Heron USV LOS path following,” *GitHub Repository*, 2025. [Online]. Available: https://github.com/duro-timi/Heron_USV_LOS_Path_Following
- [22] O. Obajinmi, “Heron USV performing turning circle test,” *Vimeo Video*, 2025. [Online]. Available: <https://vimeo.com/1064622580/c0c6a4aa51?share=copy>
- [23] O. Obajinmi, “Heron USV converging to a straight line with a path angle of 45 degrees using LOS method,” *Vimeo Video*, 2025. [Online]. Available: <https://vimeo.com/1064617628/86605f4d8d?share=copy>