Robotica (2024), **42**, pp. 1909–1918
doi:10.1017/S0263574724000638

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Autonomous navigation and steering control based on wireless non-wheeled snake robot

Liming Bao, Yongjun Sun and Zongwu Xie

State Key Laboratory of Robotics and System, Harbin Institute of Technology, Harbin, China
Corresponding author: Yongjun Sun; Email: sunyongjun@hit.edu.cn

Received: 27 February 2024; **Revised:** 26 March 2024; **Accepted:** 28 March 2024; **First published online:** 8 May 2024

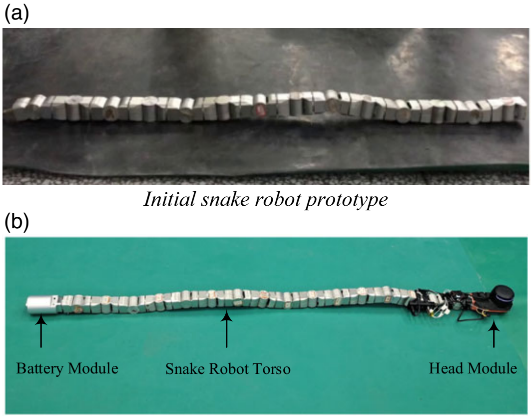
Keywords: snake robot; autonomous navigation; steering control; path following

Abstract
This paper mainly studies an autonomous path-planning and real-time path-tracking optimization method for snake robot. Snake robots can perform search and rescue, exploration, and other tasks in a variety of complex environments. Robots with visual sensors such as LiDAR can avoid obstacles in the environment through autonomous navigation to reach the target point. However, in an unstructured environment, the navigation of snake robot is easily affected by the external environment, causing the robot to deviate from the planned path. In order to solve the problem that snake robots are easily affected by environmental factors in unstructured environments, resulting in poor path-following ability, this paper uses the Los algorithm combined with steering control to plan the robot in real time and control the robot's steering parameters in real time, ensuring that the robot can stably follow the planned path.

1. Introduction
Snake robot has super redundant degrees of freedom, strong flexibility, and can adapt to various complex environments [1]. At present, snake robot is mainly used in disaster area search and rescue [2], Aerospace [3], firefighting [4], pipeline exploration [5, 6], dangerous area investigation [7], battlefield blasting, medical treatment [8], and other aspects. Due to its various advantages, snake robot has high research value.

Since snake robot typically needs to complete tasks in unstructured environments, external sensors are needed to construct a global map, plan paths, and autonomously avoid obstacles based on the map, track paths based on planned path, and complete tasks at the target point. The path planning of snake robot is usually carried out using fast extended random Tree (RRT) algorithm or Artificial Potential Field method (APF). RRT algorithms are often suitable for finding optimal path in complex maps. The APF algorithm is often used for autonomous obstacle avoidance in environments with many obstacles, by which planned path is relatively smooth. The path tracking of snake robot is usually carried out using the line of sight (LOS) algorithm, which has characteristics of high computational efficiency and small error and can efficiently control snake robot to follow the planned path.

Kenglung Hsu uses bidirectional RRT algorithm for path planning, introducing virtual forces to optimize the path, so that snake robot can reach the target point along a smooth path [9]; Zhen Wang established a kinematic model of snake robot arm and obstacles based on an improved RRT algorithm, conducted collision detection, and generated an effective collision-free path [10]. Xu Chen uses the APF algorithm in conjunction with ant colony algorithm to plan the path of snake robot [11]. Weixin Yang uses the LOS algorithm and cubic spline interpolation path-planning method to plan the path of an under-actuated snake robot [12]. Yang Xiu uses an adaptive path-tracking controller and combines the anti-sideslip LOS algorithm for path tracking [13].



Initial snake robot prototype

Improved snake robot prototype

Figure 1. *Snake robot prototype improvements.*

When snake robot adopts a peristaltic gait for navigation, to ensure that the robot can stably follow the path, it is necessary to carry out real-time steering control of the robot to reduce the error between the robot and the path during the navigation process. To solve this problem, this paper improves the prototype of the non-wheeled snake robot [14] on the basis of the RRT algorithm and the LOS algorithm and uses the improved steering control strategy to control the robot in real time to ensure that snake robot can stably follow the path to the target point.

2. Hardware improvement

Facing the autonomous navigation task of snake robot, this paper improves and optimizes the hardware equipment of the snake robot prototype. Based on the original 16-joint snake robot [15], a head module and battery module were added.

In the past, we have developed a non-wheeled snake robot with 16 DOF (degrees of freedom). The robot consists of 16 modular joints. The robot is shown in Fig. 1(a). Snake robot communicates and supplies power to the host computer and power supply through wired means. Snake robot can complete tasks such as peristaltic gait, rolling gait, spiral rolling gait, and head control based on Bezier curve [16], and relevant main work is described in ref. [15]. Snake robot communicates and supplies power to the host computer and power supply through wired means.

In this paper, wired communication and power supply are improved to wireless means. Snake robot consists of 16 modular joints, a head module, and a tail battery module. The robot is shown in Fig. 1(b). The joint modules of snake robot are orthogonally connected and can perform three-dimensional motion in space. The tail battery module is responsible for supplying power to snake robot. The head module includes devices such as a head controller, inertial sensors, and laser rangefinder. The head controller of snake robot is a computing rod responsible for processing sensor information and overall motion planning of snake robot.

2.1. Head module

Snake robot head controller consists of external sensors and internal controllers. External sensors include LiDAR and inertial sensors (IMU).

IMU: The IMU in the head module of snake robot measures the head posture and stably controls the head through the posture compensation control strategy to ensure stable LiDAR mapping.

https://doi.org/10.1017/S0263574724000638 Published online by Cambridge University Press

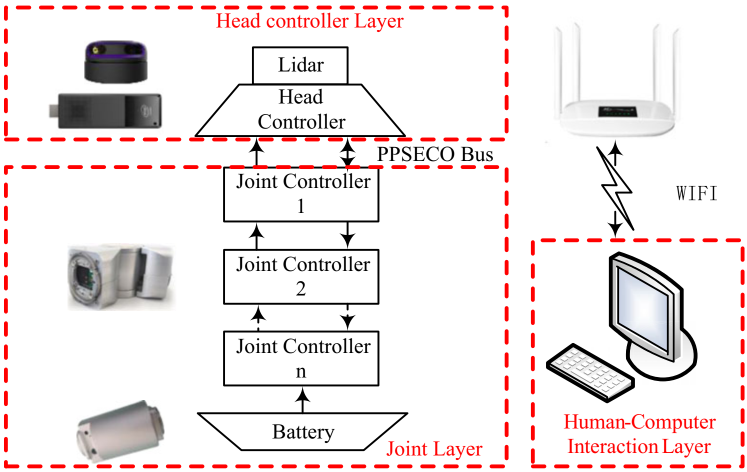


Figure 2. *Snake robot control system block diagram.*

LiDAR: Snake robot adopts 2D planar LiDAR, which has the ability of laser ranging sampling of 16,000 times per second, which can realize 360° all-round ranging scanning within a radius of 25 m and generate spatial plane point cloud data. The measurement blind area is 0.2 m, the angular resolution is 0.225°, and the scanning pitch angle range is ± 1.5°.

Head controller: The control system of snake robot adopts a hierarchical control method, which are human-computer interaction layer, head controller layer, and joint layer. The head controller integrates a computing stick, camera, photosensitive element, LED light, and PCB board. Among them, the computing stick is the main controller, which can complete the collection, processing, and sending of visual information; the gait planning of snake robot is based on the gait equation calculates the joint angle of each joint, and transmits it to the FPGA via USB. After receiving the instruction, the FPGA performs trajectory planning, reads the actual angle of each joint through the PPSECO bus [17], performs PD control [18] or speed damping control, and converts the control parameters into PWM values and sends them to each joint controller, thereby achieving control of each joint motor.

The main controller in the head controller is a computing stick, which is produced by Intel. Its model is STK1AW32SCL. The processor is Atom X5-Z8000. It has 2 g, 32G memory, and 2 USB sockets (USB2.0 and USB3.0), its built-in program can be developed using C++ on VS2017, using Windows system, and has high computing processing capabilities. The computing stick is equipped with a wireless network card, which can receive instructions sent by the host computer through Wi-Fi communication; send instructions to the FPGA; receive data from the FPGA; and then forward the data to the host computer through Wi-Fi. The wireless communication capabilities of snake robot were verified through wireless communication test experiments. The snake robot control system is shown in Fig. 2.

Head controller solves the problem of remote communication between snake robot, ensuring real-time communication and control real time between host computer and robot in the process of autonomous navigation.

2.2. Battery module

To meet autonomous obstacle avoidance task of snake robot, robot needs to be wirelessly powered. Tail battery is added to robot body to power the robot. Robot uses lithium batteries for power supply.

After installing the battery module, the power supply test was carried out, and the results proved that battery module could provide stable power supply to the robot, which could ensure the stable operation of snake robot and solve the problem of wireless power supply of snake robot. The battery drive test is shown in Fig. 3.

https://doi.org/10.1017/S0263574724000638 Published online by Cambridge University Press

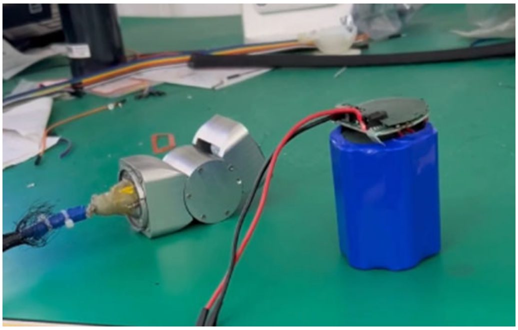


Figure 3. *Snake robot battery module driving test.*

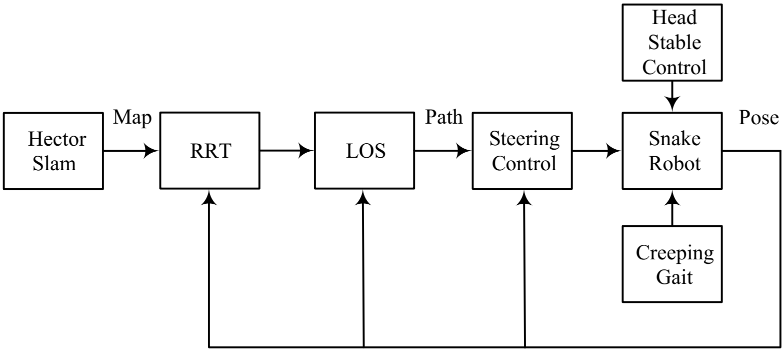


Figure 4. *Autonomous navigation flowchart of snake robot.*

After adding head module and battery module, robot can perform tasks such as stabilizing head and real-time mapping through laser radar, IMU, and other equipment. It can realize real-time motion control of robot through head controller and battery module. Autonomous power supply for robot.

3. Autonomous navigation and steer control

Autonomous navigation of snake robot is carried out by synchronous positioning and map construction through hector slam [19], path planning is carried out by RRT, and path tracking of snake robot is carried out by LOS with steering control [20]. Basic process is shown in Fig. 4.

3.1. Hector slam algorithm

Based on the IMU posture feedback method to stabilize head [21], the head LiDAR is used to map the environment when snake robot is moving. In Fig. 5, the schematic diagram of robot positioning and mapping is depicted.

For Hector slam [22], this project uses planar LiDAR, so planar map construction is required. First, a raster map is constructed based on the center of the LiDAR as the origin, and the scale coefficient of the raster map and the actual world coordinates is set to correspond to the scale of the raster map and the real environment. For raster maps, each raster has an occupied probability value *M(Si)*, ranging from [0,1]. 0 means completely free and not occupied. Finger 1 is fully occupied. Unknown areas are assigned a value of −1.

$$\chi^* = \underset{\chi}{argmin} \sum_{i=1}^{n} [1 - M(S_i(\chi))]^2 \quad (1)$$

https://doi.org/10.1017/S0263574724000638 Published online by Cambridge University Press

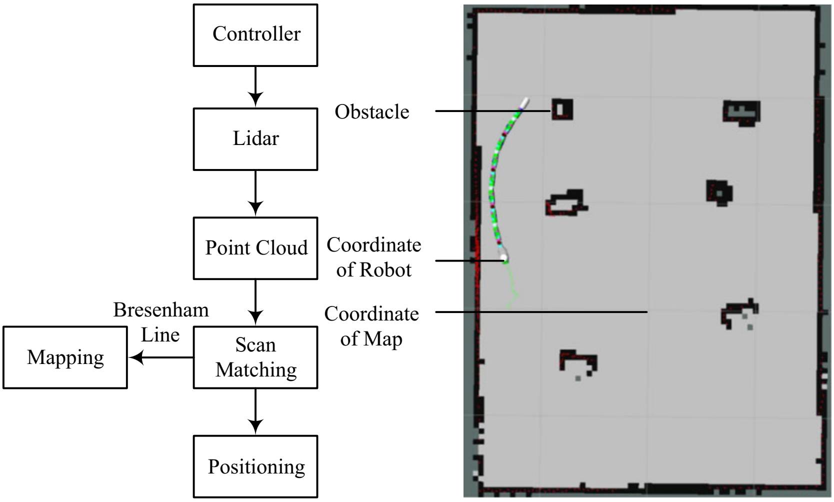


Figure 5. *Robot positioning and map construction.*

where n is the number of points acquired in one LiDAR scan. $\chi = [P_x, P_y, \phi]$, contains the displacement (P_x, P_y) and plane rotation angle ϕ when scan matching occurs. $S_i(\chi)$ is the position of the scanning point in the world coordinate system.

$$S_i(\chi) = \begin{pmatrix} \cos(\psi) & -\sin(\psi) \\ \sin(\psi) & \cos(\psi) \end{pmatrix} \begin{pmatrix} s_{i,x} \\ s_{i,y} \end{pmatrix} + \begin{pmatrix} p_x \\ p_y \end{pmatrix} \tag{2}$$

where $S_i = [S_{i,x}, S_{i,y}]$ is the coordinate of the LiDAR scanning point. Given the initial value $\chi_0 = [P_{x0}, P_{y0}, \phi]$, based on the steepest descent method (1), the scan matching pose, $\chi*$ can be solved, and then, the plane pose of snake robot can be calculated. Once the pose of the robot head is determined, the grid map is updated through the Bresenham's straight-line algorithm [23].

3.2. LOS algorithm tracking path

Based on IMU posture feedback method to stabilize the head, the head LiDAR is used to map the environment when snake robot is moving. LOS algorithm is shown in Fig. 6.

After obtaining the raster map, the RRT algorithm [24] is used to plan the robot's autonomous navigation path. After the planned path is obtained, the LOS algorithm [25, 26] is used for path tracking.

P_s represents the current position of the robot head, P_k, P_{k+1}, P_{k+2} represent the trajectory points of the planned trajectory. Select the radius R_{LOS} to represent the point that intersects with P_k-P_{k+1}. Select the point P_{Los} close to P_{k+1} as the LOS point, the direction the robot to P_{LOS} is the LOS angle, and the LOS angle can be calculated by the following equation:

$$\psi_{LOS} = \arctan\left(\frac{y_{k+1} - y_s}{x_{k+1} - x_s}\right) \tag{3}$$

ψ_{LOS} in formula is the angle of LOS angle. By calculating the angle between the head direction and the LOS angle, the robot steering is controlled.

$$u = \psi_{los} - \psi_{head} \tag{4}$$

The steering angle of the robot is indirectly controlled by controlling u.

When the robot reaches the vicinity of P_{k+1} and the radius is within the R_{AC} range, the current path point is updated as the next path point.

https://doi.org/10.1017/S0263574724000638 Published online by Cambridge University Press

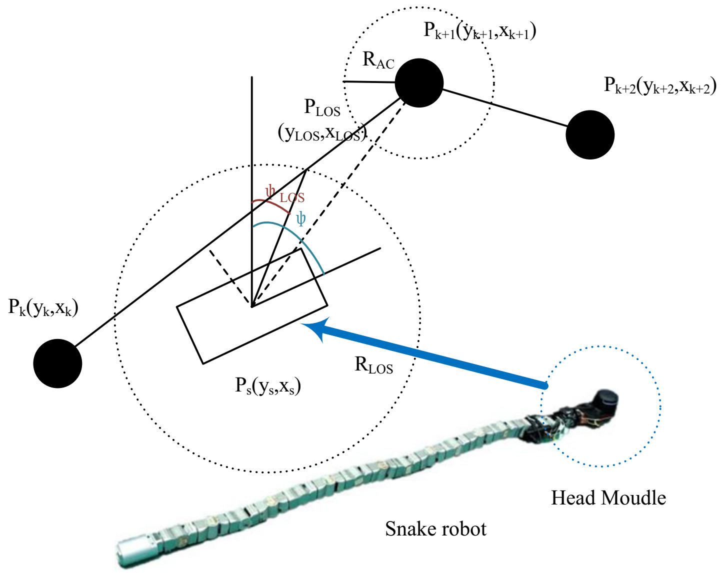


Figure 6. *LOS algorithm diagram.*

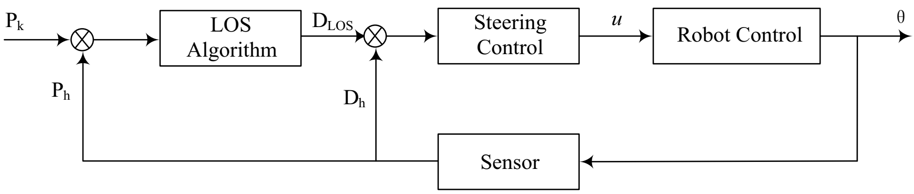


Figure 7. *Block diagram of robot's autonomous navigation control system.*

3.3. Robot steering control

Based on the steering control quantity u, snake robot is steered and controlled [27]. The block diagram of robot's autonomous navigation control system is shown in Fig. 7.

For the creeping gait [28] of snake robot, the gait equation is as follows:

$$\theta = \begin{cases} A\sin(\omega_t t_k + \omega_n n) & n = odd \\ u_k & n = even \end{cases} \tag{5}$$

where θ is the n-th joint angle at the k-th moment. ω_t controls the frequency of creeping gait changing with time, and ω_n is the spatial frequency of creeping gait. u_k can control the steering of snake robot. Based on the LiDAR local coordinate system, we have

$$\varphi = \arctan 2\left(\det(M), v_h \cdot v_t\right) \tag{6}$$

where M=[v_h;v_t], v_h=[-1,0] are the unit direction vectors of the head module axis in the LiDAR local coordinate system. v_t is the unit direction vector from the origin of the radar local coordinate system to the center of the fitting circle. ϕ is the deflection angle of the dynamic target relative to the robot head. Arctan2 is the arctangent function of two parameters. For the steering control amountδu, PD control is used as:

$$\delta u = -k_p\varphi - k_d\dot{\varphi} \tag{7}$$

$$u_{k+1} = u_k + \delta \mathrm{u} - u_{lim} \le u_{k+1} \le u_{lim} \tag{8}$$

https://doi.org/10.1017/S0263574724000638 Published online by Cambridge University Press

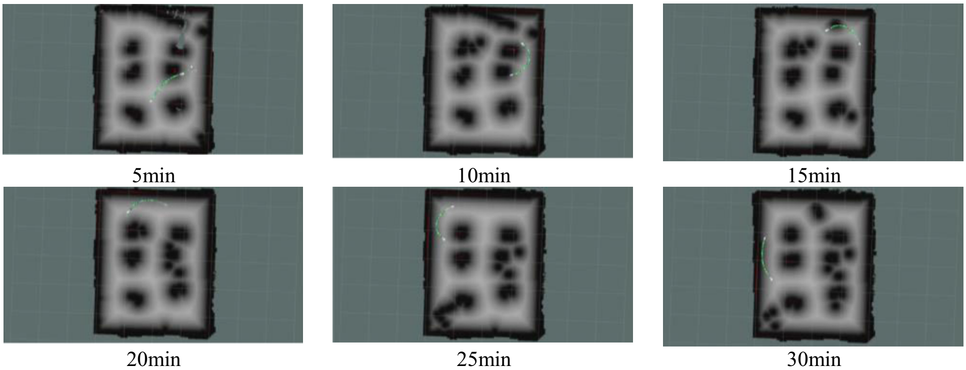


Figure 8. *Snake robot's autonomous navigation simulation.*

where k_p and k_d are pd control parameters, respectively. u_{lim} is the upper limit of the turning control variable u_{k+1}.

Real-time steering control of the robot is performed through steering control parameters, and the robot's path tracking is completed in conjunction with the creeping gait.

The steering joints of snake robot are controlled by the steering parameters of robot.

4. Simulation and experiment

This article uses the GAZEBO simulation environment to conduct autonomous navigation simulation verification of snake robot.

Fig. 8 shows the map and planned path constructed by LiDAR under RVIZ, and it can be seen that snake robot can stably follow the planned path and reach the target point.

In this paper, the wireless snake robot is used to carry out autonomous navigation and path-following experiments on snake robot. Snake robot and experimental scene are shown in Fig. 9. The ground in the scene is made of rubber to increase the friction parameters between the robot and the ground and improve the robot's navigation efficiency. Based on theory in section 3, experiments on autonomous navigation and path following were conducted on snake robot prototype.

Trajectory tracking error under simulation and experiment is shown in Fig. 10. By comparing the planned path in the experiment with the actual tracking path of the robot, it can be seen that snake robot's trajectory tracking error is between 0-6 cm, the average tracking error in experiment is 2.54 cm, which is 3.66 cm in simulation. Snake robot can stably follow the planned path, avoid obstacles autonomously, and finally reach the target point to complete its autonomous navigation.

During the simulation and experimental verification process, it can be found that snake robot completed autonomous navigation in simulation and experiment for about 30 minutes. The average trajectory error in the simulation is 3.66 cm, while the average trajectory error in the experiment is 2.54 cm. There are two reasons for the trajectory error, one is caused by the error of the map data itself during the robot movement and the other is caused by the communication interval between the robot path planning and the robot sensor data.

5. Discussion

From the error data of simulation and experiment, it can be seen that with the help of RRT algorithm for path planning and LOS algorithm for path tracking, the trajectory error of snake robot is small in the process of real-time steering, and the robot can stably follow the established path to the target point, with high efficiency. The creeping gait movement of the snake robot itself may introduce fluctuations

https://doi.org/10.1017/S0263574724000638 Published online by Cambridge University Press

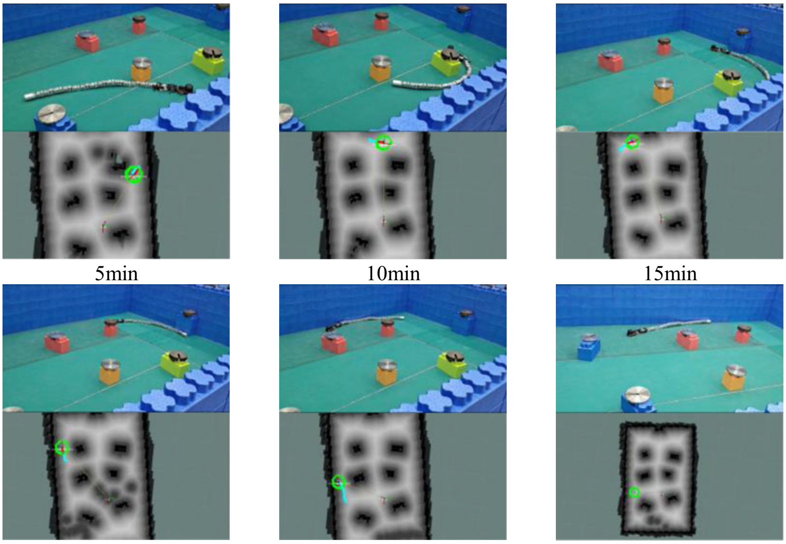


Figure 9. *Snake robot's autonomous navigation experiment.*

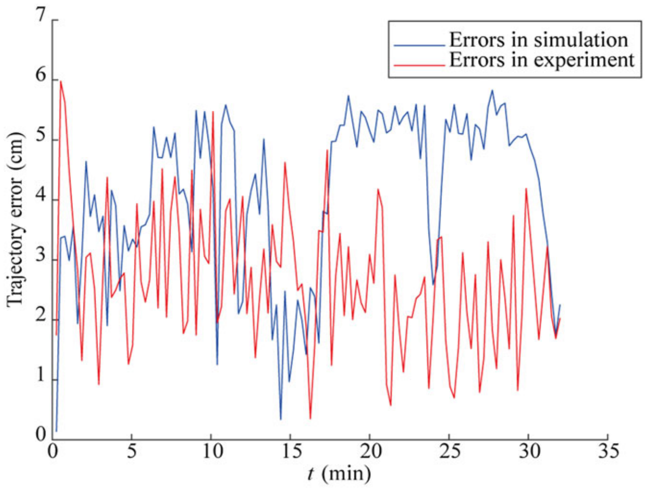


Figure 10. *Trajectory tracking error under simulation and experiment.*

in the LiDAR scanning data, leading to errors in the mapped data and affecting trajectory accuracy. Additionally, there is a time interval between the communication of snake robot and the host computer, resulting in a delay in updating the robot's pose on the host computer. These factors collectively contribute to trajectory errors, which are targeted for improvement in future developments.

6. Conclusion

By improving the prototype of snake robot, the robot can use its own equipment to carry out synchronous positioning and map construction in an unknown environment, and with the help of Wi-Fi, it can communicate with the host computer in real time to complete the real-time control the robot by the host computer, and the battery module ensures the power supply problem under the stable operation the

https://doi.org/10.1017/S0263574724000638 Published online by Cambridge University Press

robot. Based on the improved snake robot, with the help of RRT algorithm and LOS algorithm for path planning and tracking, this paper adopts a real-time steering control strategy based on the peristaltic gait of snake robot to control the robot, to ensure that snake robot can stably follow the path to the target point during autonomous navigation.

Author contributions. Liming Bao, Yongjun Sun, and Zongwu Xie conceived and designed the study and wrote the article.

Financial support. This research was funded by China Postdoctoral Science Foundation funded project (2021T140159).

Competing interests. The authors declare no competing interests exist.

Ethical approval. None.

References

[1] Y. A. Baysal and I. H. Altas, "Adaptive Snake Robot Locomotion in Different Environments," **In:** *2020 International Conference on Control, Automation and Diagnosis (ICCAD)*, Paris, France (2020) pp. 1–6.
[2] W. Yang, G. Wang and Y. Shen, "Perception-Aware Path Finding and Following of Snake Robot in Unknown Environment," **In:** *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA (2020) pp. 5925–5930.
[3] S. Han, S. Chon, J. Y. Kim, J. Seo, D. G. Shin, S. Park, J. T. Kim, J. Kim, M. Jin and J. Cho, "Snake robot gripper module for search and rescue in narrow spaces," *IEEE Robot Auto Lett* **7**(2), 1667–1673 (2022).
[4] P. Liljebäck. *Snake Robots: Modelling, Mechatronics, and Control* (Springer, London, 2013).
[5] S. Toyoshima and F. Matsuno, "A Study on Sinus-Lifting Motion of a Snake Robot with Energetic Efficiency," **In:** *2012 IEEE Inter-national Conference on Robotics and Automation*, Saint Paul, USA (2012) pp. 2673–2678.
[6] E. Prada, Valášek M., Virgala, I., Gmiterko, A., Kelemen, M., Hagara, M. and Lipták, T., "New Approach of Fixation Possibilities Investigation for Snake Robot in the Pipe," **In:** *2015 IEEE International Conference on Mechatronics and Automation (ICMA)*, Beijing, China (2015) pp. 1204–1210.
[7] J. Liu, Y. Wang, M. Li and R. Deng, "Joint Linkage and Motion System Design of Pipeline Detecting Snake Robot," **In:** *35th Chinese Control and Decision Conference (CCDC)*, Yichang, China (2023) pp. 1545–1550. doi: 10.1109/CCDC58219.2023.10327219
[8] C. Wang, V. R. Puranam, S. Misra and V. K. Venkiteswaran, "A snake-inspired multi-segmented magnetic soft robot to-wards medical applications," *IEEE Robot Auto Lett* **7**(2), 5795–5802 (2022).
[9] K. L. Hsu, "Obstacle avoidance path scheme of snake robot based on bidirectional fast expanding random tree algorithm," *J King Saud Univ - Sci* **34**(4), 101975 (2022).
[10] Z. Wang, J. Chang, B. Li, C. Wang and C. Liu, "Application of Improved Rapidly-exploring Random Trees (RRT) algorithm for Obstacle Avoidance of Snake-like Manipulator," **In:** *2020 IEEE International Conference on Mechatronics and Automation (ICMA)*, Beijing, China (2020) pp. 490–495. doi: 10.1109/ICMA49215.2020.9233573
[11] X. Chen and Y. Jiang, "Three Dimensional Path Planning of Snake-Arm Robot Based on Improved Ant Colony Algorithm," **In:** *12th International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, Baishan, China (2022) pp. 888–892. doi: 10.1109/CYBER55403.2022.9907588
[12] W. Yang, G. Wang, H. Shao and Y. Shen, "Spline Based Curve Path Following of Underactuated Snake Robots," **In:** *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada (2019) pp. 5352–5358. doi: 10.1109/ICRA.2019.8793531
[13] Y. Xiu, D. Li, M. Zhang, R. Law and E. Q. Wu, "Anti-sideslip Line of Sight Method-based Path Tracking Control for a Multi-joint Snake Robot," **In:** *2022 IEEE International Conference on Networking, Sensing and Control (ICNSC)*, Shanghai, China (2022) pp. 1–6. doi: 10.1109/ICNSC55942.2022.10004143
[14] T. Takanashi, M. Nakajima, T. Takemori and M. Tanaka, "Obstacle-aided locomotion of a snake robot using piecewise helixes," *IEEE Robot Auto Lett* **7**(4), 10542–10549 (2022). doi: 10.1109/LRA.2022.3194689.
[15] F. Ni, Y. Li, Y. Zhou, L. Zhao and H. Liu, "Design of a Hierarchical Control System for Tetherless Snake Robot," **In:** *2019 IEEE International Conference on Mechatronics and Automation (ICMA)*, Tianjin, China (2019) pp. 1254–1259.
[16] Y. Zhou, Y. Zhang, F. Ni and H. Liu, "Head-raising method of snake robots based on the Bézier curve," *Robotica* **39**(3), 503–523 (2021). doi: 10.1017/S0263574720000533.
[17] P. He, M. H. Jin, L. Yang, R. Wei, Y. W. Liu, H. G. Cai, H. Liu, N. Seitz, J. Butterfass and G. Hirzinger, "High Performance DSP/FPGA Controller for Implementation of HIT/DLR Dexterous Robot Hand," **In:** *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04*, New Orleans, USA (2004) pp. 3397–3402. doi: 10.1109/ROBOT.2004.1308779.
[18] P. Tomei, "A simple PD controller for robots with elastic joints," *Ieee Trans Automat Contr* **36**(10), 1208–1213 (1991).

[19] X. Wei, C. Yang, L. Kong and P. Sun, "Improved Hector-SLAM Algorithm Based On Data Fusion of LiDAR and IMU for a Wheeled Robot Working in Machining Workshop," **In:** *2022 China Automation Congress (CAC)*, Xiamen, China (2022) pp.2126–2131. doi: 10.1109/CAC57257.2022.10055160.

[20] Y. Xiu, D. Li, H. Deng, S. Jiang and E. Q. Wu, "Path-following based on fuzzy line-of-sight guidance for a bionic snake robot with unknowns," *IEEE/ASME Trans Mechatr* **28**(6), 3167–3179 (2023). doi: 10.1109/TMECH.2023.3254817.

[21] L. Bao, Y. Sun, Q. Wang and Z. Xie, "Study on head stabilization control strategy of non-wheeled snake robot based on inertial sensor," *Appl Sci* **13**(7), 4477 (2023). doi: 10.3390/app13074477.

[22] S. Nagla, "2D Hector SLAM of Indoor Mobile Robot using 2D Lidar," **In:** *International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, Chennai, India (2020) pp. 1–4. doi: 10.1109/ICPECTS49113.2020.9336995

[23] M. K. Kaleem, D. Verma and M. J. Idrisi, "Generalization of Line Drawing Algorithm – An Effective Approach to Minimize the Error in the Existing Bresenham's Line Drawing Algorithm," **In:** *2021 International Conference on Emerging Smart Computing and Informatics (ESCI)*, Pune, India (2021) pp. 516–521. doi: 10.1109/ESCI50559.2021.9396940

[24] S. I. Ullah, T. Mahmood and Anayatullah, "Autonomous Navigation and Mapping of Snake Robot for Urban Search and Rescue," **In:** *2023 International Conference on Robotics and Automation in Industry (ICRAI)*, Peshawar, Pakistan (2023) pp. 1–8. doi: 10.1109/ICRAI57502.2023.10089544

[25] A. Singh, C. G. Anshul and H. Choset, "Modelling and Path Planning of Snake Robot in Cluttered Environment," **In:** *2018 International Conference on Reconfigurable Mechanisms and Robots (ReMAR)*, Delft, Netherlands (2018) pp. 1–6. doi: 10.1109/REMAR.2018.8449833

[26] D. Li, Y. Zhang, P. Li, R. Law, Z. Xiang, X. Xu, L. Zhu and E. Q. Wu, "Position errors and interference prediction-based trajectory tracking for snake robots," *IEEE/CAA J Automatica Sinica* **10**(9), 1810–1821 (2023). doi: 10.1109/JAS.2023.123612.

[27] Z. Cao, D. Zhang and M. C. Zhou, "Direction control and adaptive path following of 3-D snake-like robot motion," *IEEE Trans Cybernet* **52**(10), 10980–10987 (2022). doi: 10.1109/TCYB.2021.3055519.

[28] N. T. Shugen Ma, B. Li and K. Inoue, "Analysis of Creeping Locomotion of a Snake Robot on a Slope," **In:** *2003 IEEE International Conference on Robotics and Automation (Cat*, Taipei, Taiwan (2003) pp. 2073–2078. doi:10.1109/ROBOT.2003.1241899

Cite this article: L. Bao, Y. Sun and Z. Xie (2024). "Autonomous navigation and steering control based on wireless non-wheeled snake robot", Robotica **42**, 1909–1918. https://doi.org/10.1017/S0263574724000638