Supplement to the paper "DVRP-MHSI: Dynamic Visualization Research Platform for Multimodal Human-Swarm Interaction"

Pengming Zhu, Zhiwen Zeng, Weijia Yao, Wei Dai, Huimin Lu, and Zongtan Zhou

***Abstract*—This material is the supplement to the paper "DVRP-MHSI: Dynamic Visualization Research Platform for Multimodal Human-Swarm Interaction". Related work on swarm systems is introduced in Section I. Section II gives some important parameters of the screen and explains in detail the design of the robot (including structural design, electrical design, and motion control). The results of demonstration experiments are presented and discussed in Section III.**

I. Related work on swarm systems

In the past decades, a considerable number of swarm robotic platforms have been developed to meet different research needs. This section introduces existing simulators and hardware platforms for swarm research.

A. Swarm simulators

Simulation is one of the most commonly used tools for analyzing and validating swarm robotic systems [1]. Before deploying algorithms on physical robots, testing in simulation is a necessary step because such a practice can reduce costs and potential risks, lowering barriers to entry for swarm research. Besides, simulation is easier to conduct and facilitates algorithms debugging. Moreover, one can accelerate the simulation such that it is faster than physical real-time to observe the long-term behaviors of swarms in advance. However, it is difficult for simulation to model the exact robot hardware system and the physical interaction characteristics of the robots. Inaccurate modeling of adverse factors inherent in real-world environments, such as perceptual noise, communication delays, and collisions, can lead to significant differences between simulations and physical experiments. Main characteristics of commonly used simulation platforms.

There are many kinds of software platforms for simulating robots based on the type of robot and the purpose of the research. TABLE I summarizes some important characteristics of commonly used simulators for robots, which provides a useful reference for choosing appropriate swarm robotics simulators. One simulator is Gazebo, a powerful open-source 3D simulation platform built on ODE and other physics engines, widely used in scientific research and engineering. This simulator provides various common robot models with optional sensors such as LiDAR, IMU, and infrared sensors. It can also simulate physical properties such as gravity, friction, and lighting. Besides, Gazebo is closely integrated with the Robot Operating System (ROS), leveraging the ecosystem and modularity of ROS to improve the development environment and allow for realistic simulations. Thus, researchers can conduct in-depth studies on robots and migrate the testing algorithms to physical robotics platforms easily. In our previous work [2]–[4], we developed a multi-robot simulation system, Simatch[1], based on Gazebo and ROS for RoboCup MSL Soccer Robots. Moreover, Simatch is open source and free, and it has been used for multi-robot research such as encirclement control [5], [6], task allocation [7], human-multi-robot shared control [8], and so on. In addition, simulators such as ARGoS [9], CoppeliaSim [10], USARSim [11], Webots [12], and others have also been used for robotic swarm research.

B. Hardware platforms

Platforms for swarm robots have several fundamental characteristics that distinguish them from other types of platforms, of which the key ones are size and cost, as these two aspects will contribute to the scalability of swarm systems in the real world. The various robot platforms used in swarm robotics research have different characteristics, and these differences ultimately determine which tasks the robots can perform. This subsection presents the most common and relevant platforms used for general swarm experiments. We limit the comparison to small robots with dimensions less than 10 cm, as smaller robots are usually better suited for swarm robotics. Based on the locomotion mechanisms, robots can be classified into vibration-driven and wheel-driven ones.

The Kilobot [14] and Droplet [15] are typical vibration-driven small swarm robotic platforms. They both have a symmetrical kinematic structure with three rigid supporting legs. The difference is that the Kilobot uses only two vibration motors, whereas the Droplet uses three. The Kilobot is one of the most popular research platforms because of its low-cost (parts cost only $14) and open-source features. The robot uses a kinematic principle based on two vibration motors, which reduces cost and size but also limits its maximum speed to 1 cm/s. The platform demonstrates decentralized self-assembly behavior, with 1,024 Kilobot swarms that can be assembled into user-defined shapes, allowing researchers to test on large swarms. Although robots using vibration-driven motion mechanisms are becoming more common and are easily coupled to robots, their poor kinematic capabilities render precise motion execution over long periods challenging.

[1]Available at https://github.com/nubot-nudt/simatch. It is the official designated platform for the middle-size league simulation match of the RoboCup China Open.

TABLE I
MAIN CHARACTERISTICS OF COMMONLY USED SIMULATION PLATFORMS.

Simulator	Physics engines	Languages	Main sensors	Windows	Linux	MAC OS	ROS	Last update	Website(Accessed:2024)
ARGoS [9]	ODE, 3D particle engine, 2D-kinematics engine	C++,Lua	Altitude, Camera, LiDAR, Proximity		✓	✓	✓	2022	https://www.argos-sim.info
CoppeliaSim [10]	ODE, Bullet, Vortex, Newton	C, C++, Python, Java, LUA, Matlab, Octave	Camera, Force sensor, Infrared, LiDAR, Proximity	✓	✓	✓	✓	2023	https://www.coppeliarobotics.com
Gazebo [13]	ODE, Bullet, Simbody, DART	C, C++, Python,Java	Camera, Contact sensor, Force sensor, GPS, IMU, LiDAR, Torque sensor	✓	✓	✓	✓	2023	https://gazebosim.org/home
Isaac Sim	NVIDIA Omniverse	C, C++, Python, Java, Lua, MatLab, Octave	Camera, IMU, Infrared, LiDAR,	✓	✓	✓	✓	2023	https://developer.nvidia.com/isaac-sim
PyBullet	Bullet	C, C++, Python, Java, Lua, MatLab, Octave	Camera, Force sensor, LiDAR, Torque sensor	✓	✓	✓	✓	2024	https://pybullet.org/wordpress
MORSE	Blender engine, Bullet	Python	Camera, GPS, Proximity Sensor, Velocity		✓	✓	✓	2016	https://www.openrobots.org morse/doc/stable/morse.html
MRDS	PhysX engine	VPL, C#, Visual Basic, JavaScript, Iron-Python	Camera, Contact sensors, Encoder, GPS	✓				2012	https://learn.microsoft.com/en-us/previous-versions/microsoft-robotics/bb648760(v=msdn.10)
USARSim [11]	Unreal engine	C, C++, Java	Camera, Contact sensors, Encoder, GPS, Sound sensor	✓	✓	✓	✓	2015	https://sourceforge.net/projects/usarsim
Webots [12]	ODE	C, C++, Python, Java, MATLAB	Accelerometer, Camera, Contact sensors, GPS, LiDAR,	✓	✓	✓	✓	2023	https://cyberbotics.com

Wheel-driven robots are more widely used in swarm robotics research compared to vibration-driven robots. Most current platforms use a two-wheel differential drive model, which has good balance and is easier to control. Typically, two wheels are attached to the motor and distributed symmetrically on the robot, while the front wheel (or the support point) provides 360 degrees of free rotational motion. There are a number of commonly used platforms that use this type of movement, such as HeRo2.0 [16], Mona [17], Colias IV [18], GRITSBot [19], AMiR [20], E-puck [21], Jasmine [22], Alice [23], and so on. Among these, E-puck is one of the most successful commercially available small robots. It has a small footprint of 7 cm, is driven by two stepper motors, and can move at a speed of up to 10 cm/s. Moreover, the robot has a variety of built-in sensors, including microphones, proximity sensors, a camera, and an inertial motion unit, and communicates using Wi-Fi. It can also be extended with other sensing modules, such as an omnidirectional camera. The platform's ability to integrate with a variety of simulators and robotics frameworks, such as Gazebo and ROS, has made it a popular choice for many users. Despite its many advantages, the basic commercial version of the E-puck is expensive (retailing at $975), making it unaffordable for those working on larger swarms. Also, as two-wheeled differential robots, to minimize the robot's footprint, the Millibot [24] and the Zooids [25] employ two micro motors placed non-collinearly, with ball casters to help balance the robot.

In addition, there are robots with special locomotion mechanisms, such as the single-wheeled mobile robot [26] and Morphobot [27]. The Morphobot is driven by four DC coreless motors via rotation, which are mounted evenly on the robot base at a specific angle relative to the ground. By adjusting the speed of the motors, the robot is capable of omnidirectional movement. The single-wheeled mobile robot is fitted with an omnidirectional wheel and six spring suspensions. The omnidirectional wheel is placed at the bottom center of the robot, and the six spring suspensions are evenly distributed on the periphery of the robot to keep it balanced. Two motors drive the omnidirectional wheel, one forward and one rotating in its direction. With this ingenious design, the robot can move in any direction, addressing some of the issues with traditional bulky omnidirectional wheels such as orthogonal, spherical, and McCannon wheels, which are too large for small mobile robots.

II. DETAILS OF THE SCREEN AND THE DESIGNED ROBOT

A. Some important parameters of the screen

The important parameters of the screen are shown in TABLE III.

B. Three-wheeled omnidirectional robot

Given the limited area of the test environment, to support a large-scale multi-robot system carrying out the test simultaneously, the designed robot should meet some requirements, including small size, low cost, easy maintenance/replacement, and individual capabilities of communication, control, planning, and movement.

1) structure and main components: Most of the robots proposed for swarm robotics are based on a differential drive model. In short, the robot changes direction by controlling the relative speed of the wheels on either side of the robot's body, thus eliminating the need for additional steering motors. However, robots of this construction do not allow for flexible omnidirectional motion, which is not conducive to simplifying the robot model. Therefore, we design a small, low-cost

TABLE II
COMPARISON OF POPULAR MULTI-ROBOT PLATFORMS

Robotic platform	Year	Locomotion	Speed(cm/s)	Size(cm)	Cost	Sensors	Communication
Ours	**2024**	**Wheel, symmetric** **3 omni-wheel**	**30**	**9.6**	**$28**	**Encoder**	**Wi-Fi**
Single-wheeled Mobile Robot [26]	2023	Wheel, symmetric 1 omni-wheel + 6 spring suspensions	7.3	10	N/A	Encoder	Wi-Fi
HeRo2.0 [16]	2023	Wheel, symmetric 2 wheel differential drive	25	7.3	$18*	Distance, Encoder, Light, IMU	Wi-Fi
Millibot [24]	2023	Wheel, non-symmetric 2 wheel differential drive	10	5.5	$23	Distance, Microphones, Encoder, Touch	IR/Audio/Wi-Fi/Bluetooth
Morphobot [27]	2023	Pin ratation, symmetric 4 motor-driven 4 pin rotation	20	3.5	$35	3D Accelerometer, Distance, Light	Wi-Fi/Bluetooth
Mona [17]	2019	Wheel, symmetric 2 wheel differential drive	1.2	6.5	$108	Distance, Encoder, Light	IR/Wi-Fi/Bluetooth
Colias IV [18]	2018	Wheel, symmetric 2 wheel differential drive	25	4	$100*	Camera, Distance, IMU, Microphones	Bluetooth
Zooids [25]	2016	Wheel, non-symmetric 2 wheel differential drive	44	2.6	$50*	Light, Touch	RF/Wi-Fi
GRITSBot [19]	2015	Wheel, symmetric 2 wheel differential drive	25	3	$50*	3D Accelerometer, 3D Gyro, Bearing, Distance	RF
Droplets [15]	2014	Vibration, symmetric 3 vibration motors + 3 rigid legs	1	4.4	$100	Bearing, Distance, Light	IR
Kilobot [14]	2013	Vibration, symmetric 2 vibration motors + 3 rigid legs	1	3.3	$14*	Ambient light, Distance	IR
AMiR [20]	2009	Wheel, symmetric 2 wheel differential drive	14	6.6	$72*	Bearing, Distance, Light	IR
E-puck [21]	2009	Wheel, symmetric 2 wheel differential drive	13	7.5	$975	3D Accelerometer, Camera, Distance, Light, Microphones	Wi-Fi/Bluetooth
Jasmine [22]	2009	Wheel, symmetric 2 wheel differential drive	30	3.0	$120*	Bearing, Color, Distance, Light, Touch	IR
Alice [23]	2003	Wheel, symmetric 2 wheel differential drive	4	2	N/A	Camera, Distance, Light	IR/RF

* parts only

TABLE III
SOME IMPORTANT PARAMETERS OF THE SCREEN.

Attribute	Parameter
Display size	55” 16:9 wide
Native resolution	Full HD 1920 × 1080
Display active area	1209 × 680 mm
Picture frame rate	60 Hz
Tracking speed	Up to 200 fps
Positional accuracy	2mm, directional compensation
Interaction methods	Fingers and hands, objects with optical markers, basic shapes, infrared pen. Click, hold and drag.
Hand recognition	Each finger orientation recognized and fingers identified to certain hand.
Object recognition	Supported with 2D fiducial markers. Some basic shapes. Unlimited simultaneous pen and touch input.
Touch latency	10 ms typical. Measured from touch to processed tracking data output.
Stackability	Any number of Cells can be stacked to run a single multitouch application.

robot based on three omnidirectional wheels, which are evenly spaced 120 degrees apart, to achieve omnidirectional motion of the robot[2].

The structure and components of the designed robot are shown in Fig. 1. The entire robot has a radius of 4.8cm and weighs 200g. For ease of assembly and further expansion, the robot chassis are designed to be modular and 3D printable, so they can be easily printed using a 3D printer. Its basic frame and shell are both 3D-printed from polylactic acid (PLA) material. The shell takes on a triangle-like shape to reduce the robot’s footprint. Overall, the robot structure consists of three main parts from top to bottom: the hat module, the PCB board, and the motor chassis.

The hat module primarily loads the battery and LEDs. The robot uses an 11.1V 500mAh Li-Po battery to drive the motors and power the board, and the LEDs serve as indicators for switch status, communication connections, and power levels.

The PCB board contains two layers: the main control on the bottom board and the driver board on top. The mainboard is designed based on the ESP32 microcontroller, which includes a 2.4GHz WiFi module and DC-DC conversion modules.

[2]Differential drive wheeled mobile robots can also be used in our platform. Therefore, this platform, for example, supports heterogeneous multi-robot experiments with both omnidirectional wheeled and differential drive mobile robots.

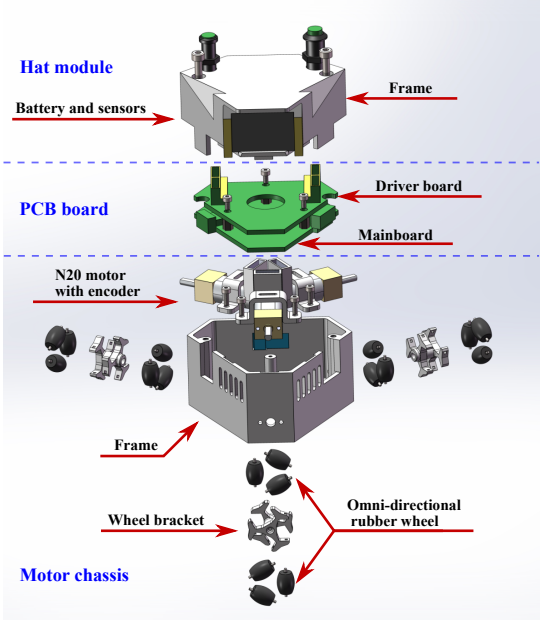


Fig. 1. The structure and components of the robot.

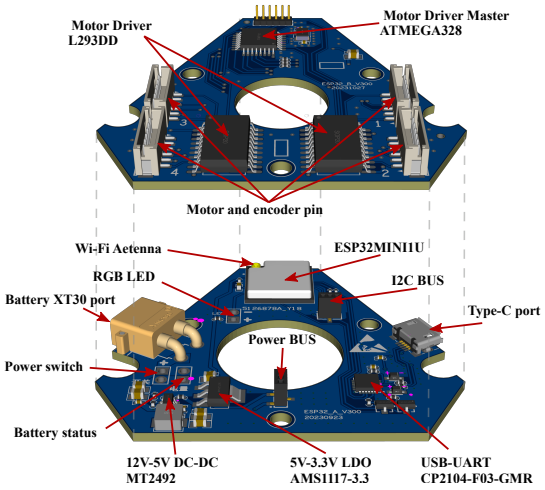


Fig. 2. Overview of PCB board.

The driver board drives the N20 motors and reads encoder information.

The motor chassis supports all motors and the wheel shaft. Three N20 motors with Hall encoders are mounted on the chassis, each distributed in a 120-degree arrangement in the same plane. Each omnidirectional wheel is mounted on a motor shaft and consists of two layers of 3D-printed structural brackets and six identical rubber wheels, with 60 rotational degrees between the brackets. To achieve the omnidirectional motion of the robot, we adopt a double-layer orthogonal omnidirectional wheel design with a 6-roller structure, which enhances the flexibility of the robot's motion. We test rubber wheels with different friction coefficients to ensure enough friction to allow the robot to move smoothly on the screen.

2) Electrical design: In addition to the robot's mechanical design, we also perform its electrical design to ensure the robot meets our requirements. To accommodate the frame's shape and the layout of the motors, we arrange our various components on a triangular PCB board, as illustrated in Fig. 2. A circular cut-out is made in the middle of the board for the motor connection wires to pass through. The upper board is the motor driver board, while the lower board is the main control board. These two layers are directly connected via I2C and power BUS, facilitating communication and power distribution.

A crucial aspect of the robot's electrical design is the selection of an appropriate microcontroller. This component dictates the robot's computing power as well as the number and type of components that can be utilized. After evaluating numerous alternatives, we selected the ESP32MINI1U as the main processing unit. This microcontroller offers impressive processing power (32-bit, 160 MHz) and 4 MB of memory, all at a very affordable price. Additionally, it features an integrated Wi-Fi chip capable of providing a complete TCP/IP stack and high-speed IEEE 802.11 connectivity. Consequently, robots can communicate with each other or with remote computers using robust and scalable protocols. The ESP32MINI1U is not only efficient and easy to program but also widely adopted within developer communities, facilitating the development of custom modules by other developers.

For the driver board, we utilize two L293DD chips to control three N20 motors and to read the feedback from the Hall encoder. Each chip can manage two motors. We employ an ATMEGA328 to receive information from the main controller and distribute it to the two L293DD chips.

Additionally, each component of the robot may require a different supply voltage. To address this, we utilize the MT2492 and AMS1117-3.3 voltage regulators to convert the 11.1V battery voltage to the specific voltages needed by various components.

For firmware implementation, we have chosen the Arduino IDE due to its user-friendly nature and widespread adoption within developer communities. This choice enhances the ease of following and modifying the firmware code. Furthermore, the Arduino IDE offers numerous libraries that facilitate control over microcontroller ports and management of TCP sockets, sensors, and actuators.

3) motion control: As previously stated, the robot operates in a three-wheeled omnidirectional motion mode, propelled by three N20 motors equipped with hall encoders. These encoders accurately measure the rotational speed of each wheel, providing crucial feedback to the controller for precise closed-loop motion control. Additionally, the speed and direction of the motors are regulated through pulse width modulation (PWM) signals.

The relationship between body velocity and wheel velocity is shown in Fig. 3. Given the linear speed v_x, v_y, and angular speed ω, we can determine the required speed for each wheel to achieve the desired motion of the robot. This inverse kinematic model can be represented by the following linear equations:

$$v_1 = v_x + \omega L \tag{1}$$

$$v_2 = -v_x \cos\theta_1 - v_y \sin\theta_1 + \omega L \tag{2}$$

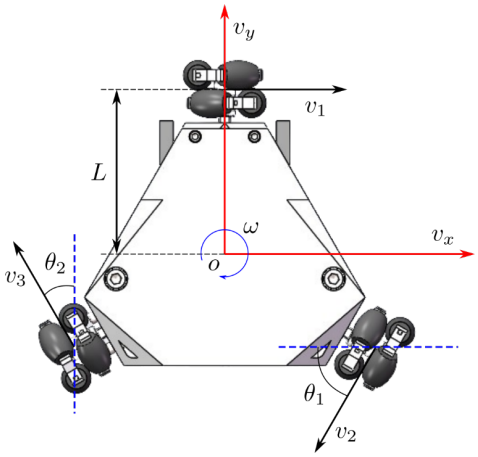


Fig. 3. The relationship between body velocity and wheel velocity.

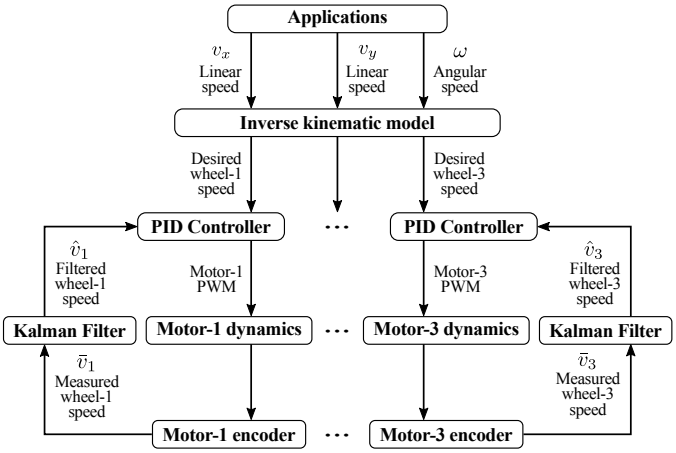


Fig. 4. The velocity control of our three-wheeled omnidirectional robot.

$$v_3 = -v_x \sin\theta_2 + v_y \cos\theta_2 + \omega L, \tag{3}$$

and the corresponding matrix form is

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & L \\ -\cos\theta_1 & -\sin\theta_1 & L \\ -\sin\theta_2 & \cos\theta_2 & L \end{bmatrix} \begin{bmatrix} v_x \\ v_y \\ \omega \end{bmatrix}, \tag{4}$$

where v_1, v_2, v_3 are wheel velocities, θ_1 and θ_2 are the angles in Fig. 3, L is the distance from the wheel to the center of the robot, v_x is the velocity along the x-axis, v_y is the velocity along the y-axis, and ω is the angular velocity.

After calculating the required tangential speeds for each wheel, we use PID control methods to regulate the motors and maintain these speeds. The PID controller corrects errors through proportional action, eliminates steady-state bias via integral action, and predicts future trends using derivative action. The mathematical model of a PID controller is defined as:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt + K_d \frac{de(t)}{dt}, \tag{5}$$

where $e(t)$ is the error between the desired tangential velocity and the current tangential velocity for each wheel, K_p, K_i, K_d are the non-negative coefficients of the proportional, integral, and derivative terms, respectively.

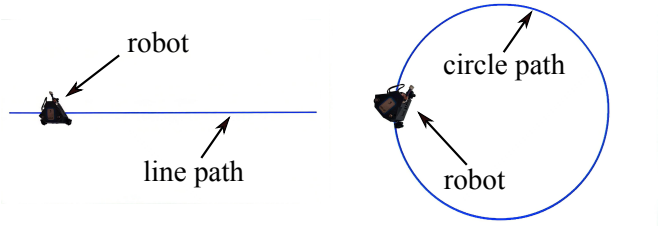
Moreover, a simple Kalman filter is employed to reduce measurement noise and enhance the accuracy of the estimated speeds. The Kalman filter process is:

$$K = \frac{\sigma_e^-}{\sigma_e^- + \sigma_m}, \tag{6}$$

$$\hat{v} = \hat{v}^- + K(\bar{v} - \hat{v}^-), \tag{7}$$

$$\sigma_e = (1 - K)\sigma_e^- + |\hat{v}^- - \hat{v}|q, \tag{8}$$

where K is the Kalman gain, σ_e and σ_m represent the estimation uncertainty and measurement uncertainty respectively, $\hat{v}$ is the filtered speed, $\bar{v}$ is the measured speed, q is the process variance. The superscript $(-)$ indicates the previous values of a variable. The whole velocity control process of the robot is shown in Fig. 4.



(a) The robot follows a line path (b) The robot follows a circle path

Fig. 5. The robot locomotion experiments. (a) following a line path. (b) following a circle path.

III. Validation

In this section, we conduct experiments to demonstrate the locomotion ability of the designed robot.

We perform path-following experiments on the screen to test the mobility of robots. Desired linear and circular trajectories can be displayed on the screen to visualize the robot's movements, as shown in Fig. 5. In addition, the screen records and transmits the motion data via a ROS node for quantitative analysis, which can be replayed in simulation with ROS bag if necessary.

We test the robot's ability to follow a straight line at six different speeds and take ten samples at each speed. The position coordinates of the robot are sampled at a frequency of 30 Hz, and the trajectories at different speeds are shown in Fig. 6, with the desired trajectory in blue. It is worth noting that although intuitively it looks like the motion fluctuates more, it is actually due to the fact that the scale of the y-axis does not equal with x-axis. In fact, when moving at maximum speed, the maximum position deviation is only 2cm.

We define the error as the sum of the difference between the experimental trajectory coordinates and the desired trajectory, and the absolute error as the sum of the absolute value of the difference between the experimental trajectory coordinates and the desired trajectory. Fig. 7(a) shows the average error and the corresponding standard deviation for all speeds, and Fig. 7(b) shows the average absolute error and the corresponding standard deviation. The results show that, regardless of the

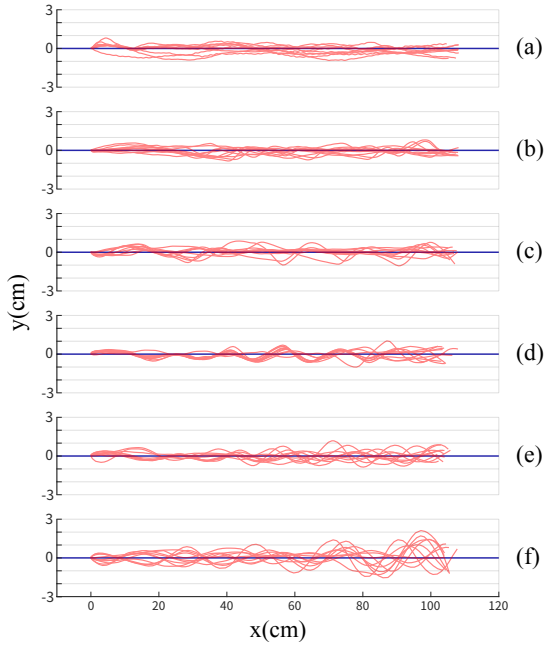


Fig. 6. Linear trajectories at different speeds: (a) 5cm/s. (b) 10cm/s. (c) 15cm/s. (d) 20cm/s. (e) 25cm/s. (f) 30cm/s.

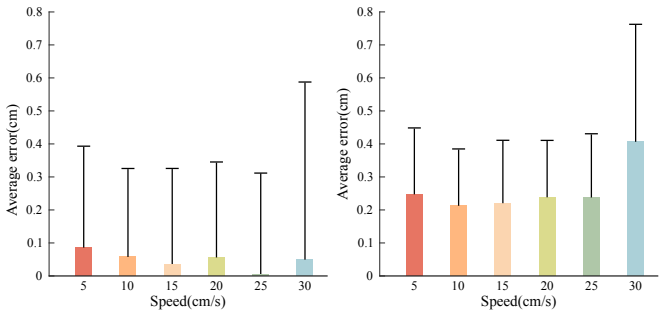


Fig. 7. The statistical analysis for linear motion.

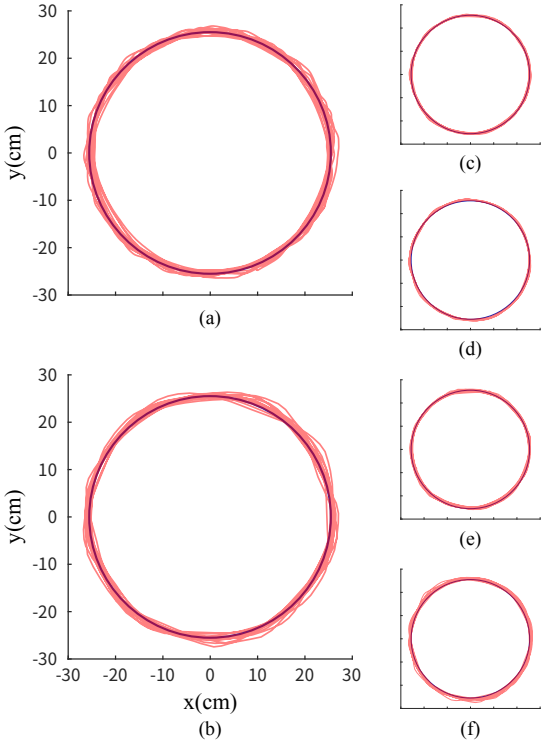


Fig. 8. Circle trajectories at different speeds: (a) 5cm/s. (b) 10cm/s. (c) 15cm/s. (d) 20cm/s. (e) 25cm/s. (f) 30cm/s.

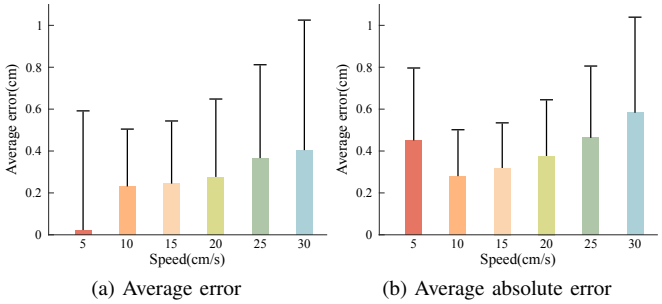


Fig. 9. The statistical analysis for circular motion.

speeds, our self-designed robot is able to follow straight lines well.

At the same speed, we test the ability of the robots to follow circular paths, with each robot moving ten laps clockwise along the circular paths separately. The trajectories at different speeds are shown in Fig. 8, with the desired trajectory in blue. It can be observed intuitively that the fluctuations in the robot's motion along the circumference are larger at speeds of 5cm/s and 30cm/s. The main reason is that 5cm/s is close to the robot's motion speed dead zone, where the control accuracy is not high. In addition, 30cm/s is almost the robot's maximum speed, at which the centrifugal force is large, resulting in fluctuations in the robot's motion.

We define the error as the sum of the difference in distance from the robot's position to the circular path, which is positive when the robot is outside the circle and negative when it is inside the circle, and the absolute error as the sum of the absolute values of the difference in distance from the robot position to the circular path. Fig. 9(a) shows the average error and the corresponding standard deviation for all speeds, and Fig. 9(b) shows the average absolute error and the corresponding standard deviation. The results show that, regardless of the speeds, our self-designed robot is able to follow a circle well. Based on the aforementioned results, we have roughly determined the velocity range for the robot to move with high precision; henceforth, it is advisable to maintain the robot's speed within the interval of 10cm/s to 25 cm/s in future experiments.

REFERENCES

[1] N. Nedjah and L. S. Junior, "Review of methodologies and tasks in swarm robotics towards standardization," *Swarm and Evolutionary Computation*, vol. 50, p. 100565, 2019.

[2] W. Yao, W. Dai, J. Xiao, H. Lu, and Z. Zheng, "A simulation system based on ros and gazebo for robocup middle size league," in *2015 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE, 2015, pp. 54–59.

[3] J. Xiao, D. Xiong, W. Yao, Q. Yu, H. Lu, and Z. Zheng, "Building software system and simulation environment for robocup msl soccer robots based on ros and gazebo," in *Robot Operating System (ROS)*,

A. Koubaa, Ed. Cham: Springer International Publishing, 2017, vol. 707, pp. 597–631.

[4] Z. Zhou, W. Yao, J. Ma, H. Lu, J. Xiao, and Z. Zheng, "Simatch: A simulation system for highly dynamic confrontations between multi-robot systems," in *2018 Chinese Automation Congress (CAC)*. IEEE, 2018, pp. 3934–3939.

[5] W. Yao, Z. Zeng, X. Wang, H. Lu, and Z. Zheng, "Distributed encirclement control with arbitrary spacing for multiple anonymous mobile robots," in *2017 36th Chinese Control Conference (CCC)*. Dalian, China: IEEE, 2017, pp. 8800–8805.

[6] J. Ma, H. Lu, J. Xiao, Z. Zeng, and Z. Zheng, "Multi-robot target encirclement control with collision avoidance via deep reinforcement learning," *Journal of Intelligent & Robotic Systems*, vol. 99, no. 2, pp. 371–386, Aug 2020.

[7] W. Dai, H. Lu, J. Xiao, and Z. Zheng, "Task allocation without communication based on incomplete information game theory for multi-robot systems," *Journal of Intelligent & Robotic Systems*, vol. 94, no. 3, pp. 841–856, Jun 2019.

[8] Y. Liu, W. Dai, H. Lu, Y. Liu, and Z. Zhou, "Brain-computer interface for human-multirobot strategic consensus with a differential world model," *Applied Intelligence*, vol. 51, no. 6, pp. 3645–3663, Jun 2021.

[9] C. Pinciroli, V. Trianni, R. O'Grady, G. Pini, A. Brutschy, M. Brambilla, N. Mathews, E. Ferrante, G. Di Caro, F. Ducatelle, M. Birattari, L. M. Gambardella, and M. Dorigo, "Argos: a modular, parallel, multi-engine simulator for multi-robot systems," *Swarm Intelligence*, vol. 6, no. 4, pp. 271–295, Dec 2012.

[10] E. Rohmer, S. P. N. Singh, and M. Freese, "V-rep: A versatile and scalable robot simulation framework," in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2013, pp. 1321–1326.

[11] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "Usarsim: a robot simulator for research and education," in *Proceedings 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 1400–1405.

[12] O. Michel, "Cyberbotics ltd. webots™: professional mobile robot simulation," *International Journal of Advanced Robotic Systems*, vol. 1, no. 1, p. 5, 2004.

[13] N. Koenig and A. Howard, "Design and use paradigms for gazebo, an open-source multi-robot simulator," in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. 3, 2004, pp. 2149–2154 vol.3.

[14] M. Rubenstein, C. Ahler, N. Hoff, A. Cabrera, and R. Nagpal, "Kilobot: A low cost robot with scalable operations designed for collective behaviors," *Robotics and Autonomous Systems*, vol. 62, no. 7, pp. 966–975, 2014.

[15] J. Klingner, A. Kanakia, N. Farrow, D. Reishus, and N. Correll, "A stick-slip omnidirectional powertrain for low-cost swarm robotics: Mechanism, calibration, and control," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. Chicago, IL, USA: IEEE, 2014, pp. 846–851.

[16] P. Rezeck, H. Azpúrua, M. F. S. Corrêa, and L. Chaimowicz, "Hero 2.0: a low-cost robot for swarm robotics research," *Autonomous Robots*, vol. 47, no. 7, pp. 879–903, 2023.

[17] F. Arvin, J. Espinosa, B. Bird, A. West, S. Watson, and B. Lennox, "Mona: an affordable open-source mobile robot for education and research," *Journal of Intelligent & Robotic Systems*, vol. 94, no. 3-4, pp. 761–775, 2019.

[18] C. Hu, Q. Fu, and S. Yue, "Colias iv: The affordable micro robot platform with bio-inspired vision," in *Towards Autonomous Robotic Systems*. Cham: Springer International Publishing, 2018, pp. 197–208.

[19] D. Pickem, M. Lee, and M. Egerstedt, "The gritsbot in its natural habitat - a multi-robot testbed," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*. Seattle, WA, USA: IEEE, 2015, pp. 4062–4067.

[20] F. Arvin, K. Samsudin, A. R. Ramli *et al.*, "Development of a miniature robot for swarm robotic application," *International Journal of Computer and Electrical Engineering*, vol. 1, no. 4, pp. 436–442, 2009.

[21] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J.-C. Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering," in *Proceedings of the 9th conference on autonomous robot systems and competitions*, vol. 1, no. 1, 2009, pp. 59–65.

[22] S. Kernbach, R. Thenius, O. Kernbach, and T. Schmickl, "Re-embodiment of honeybee aggregation behavior in an artificial micro-robotic system," *Adaptive Behavior*, vol. 17, no. 3, pp. 237–259, 2009.

[23] G. Caprari and R. Siegwart, "Design and control of the mobile micro robot alice," in *Proceedings of the 2nd International Symposium on Autonomous Minirobots for Research and Edutainment, AMiRE 2003: 18-20 February 2003, Brisbane, Australia*. CITI, 2003, pp. 23–32.

[24] P. G. Luan and N. T. Thinh, "Millibot-miniature mobile robot platform for scalable swarm robot research," *International Journal of Mechanical Engineering and Robotics Research*, vol. 12, no. 6, pp. 417–424, 2023.

[25] M. Le Goc, L. H. Kim, A. Parsaei, J.-D. Fekete, P. Dragicevic, and S. Follmer, "Zooids: Building blocks for swarm user interfaces," in *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. Tokyo Japan: ACM, 2016, pp. 97–109.

[26] M. Wang, Y. Su, H. Li, J. Li, J. Liang, and H. Liu, "Aggregating single-wheeled mobile robots for omnidirectional movements," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. Detroit, MI, USA: IEEE, 2023, pp. 314–320.

[27] X. Qin, Y. Yang, M. Pan, L. Cui, and L. Liu, "Morphobot: A platform for morphogenesis in robot swarm," *IEEE Robotics and Automation Letters*, vol. 8, no. 11, pp. 7440–7447, 2023.