



Design, Implementation and Control of an Amphibious Spherical Robot

Liwei Shi¹ · Zhongyin Zhang¹ · Zhengyu Li¹ · Shuxiang Guo^{1,2} · Shaowu Pan¹ · Pengxiao Bao¹ · Lijie Duan¹

Received: 29 December 2021 / Revised: 30 May 2022 / Accepted: 3 June 2022 / Published online: 14 July 2022
© Jilin University 2022

Abstract

We proposed and implemented a leg-vector water-jet actuated spherical robot and an underwater adaptive motion control system so that the proposed robot could perform exploration tasks in complex environments. Our aim was to improve the kinematic performance of spherical robots. We developed mechanical and dynamic models so that we could analyze the motions of the robot on land and in water. The robot was equipped with an Inertial Measurement Unit (IMU) that provided inclination and motion information. We designed three types of walking gait for the robot, with different stabilities and speeds. Furthermore, we proposed an online adjustment mechanism to adjust the gaits so that the robot could climb up slopes in a stable manner. As the system function changed continuously as the robot moved underwater, we implemented an online motion recognition system with a forgetting factor least squares algorithm. We proposed a generalized prediction control algorithm to achieve robust underwater motion control. To ensure real-time performance and reduce power consumption, the robot motion control system was implemented on a Zynq-7000 System-on-Chip (SoC). Our experimental results show that the robot's motion remains stable at different speeds in a variety of amphibious environments, which meets the requirements for applications in a range of terrains.

Keywords Bionic amphibious spherical robot · Inertial measurement unit · Quadruped gaits · Forgetting factor least squares algorithm · Generalized prediction control

1 Introduction

Amphibious robots have advantages with respect to mobility and environmental adaptability. These factors have resulted in them becoming essential tools for scientific, industrial and, military applications in littoral zones. Recently, a variety of amphibious robots were designed with a focus on applications such as disaster rescue, resource exploration, and reconnaissance [1–4]. The designs of these robots are based on the principles of bionics and compound drive mechanisms [5–10]. Cui et al. [11] proposed

a centipede-inspired amphibious robot named AmBot for monitoring the Swan-Canning River. The multi-leg actuation of a centipede was morphed into tracks and each leg-link was simplified to a track piece consisting of a base and a polystyrene-foam block. Vogel et al. [12] proposed a quadrupedal amphibious robot named RoboTerp. RoboTerp switches gaits to match terrains so that it can move on land and in water with the same legs. This approach mitigated the issues caused by significant turbulence in the robot's surroundings. Zhong et al. [13] proposed an amphibious robot with flexible flipper legs. The flipper legs are composed of seven segments. The robot can adapt to various environments by adjusting the stiffness of each leg component.

Amphibious spherical robots are a special kind of amphibious robots that are shaped like balls. Conventional amphibious robots are driven by racks and wheels; it is easier, however, to control the inclination of amphibious spherical robots. They also have advantages in terms of maneuverability. Thus, amphibious spherical robots have become a popular research topic in the field of autonomous mobile robots. Rotundus AB [14] designed an amphibious

✉ Liwei Shi
shiliwei@bit.edu.cn

¹ Key Laboratory of Convergence Medical Engineering System and Healthcare Technology, The Ministry of Industry and Information Technology, School of Life Science, Beijing Institute of Technology, Beijing 100081, China

² Faculty of Engineering, Kagawa University, 2217-20 Hayashicho, Takamatsu, Kagawa 761-0396, Japan

spherical robot named Groundbot. Groundbot was proposed as an alternative design for a Mars rover. Groundbot is driven by a pendulum and is capable of navigating rough outdoor terrains at speeds of up to 3 m/s and climbing slopes of up to 15°–18°. Jia et al. [15] proposed a novel concept for amphibious spherical robots consisting of a spherical hull and two mechanical arms. The mechanical arms swing to roll the ball on land and provide a thrust vector in underwater environments. Our research team proposed an amphibious spherical robot [16–19]. The robot has a deformable mechanical structure that enables it to walk on land with legs, swim in water using vectored thrusters that provide a zero turning radius, and navigate littoral regions with an improved overstepping ability.

Due to the particularities of amphibious environments, designing practical control systems for amphibious robots remain a thorny problem. In terrestrial environments, the drive systems of amphibious robots should make online adjustments to the motion-control strategy to adapt to different terrains such as sand, mud, and rocks. In underwater environments, the amphibious robot should be able to adjust the speed and direction of motion, pick up and release loads, and respond to variations in the water flow. This results in a dynamic model that varies continuously. Hence, amphibious robots should have control systems that can adapt to different environments and are robust to environmental conditions that can interfere with their motion. There have been some studies on the control of robots that are either amphibious or spherical, but little research has focused specifically on amphibious spherical robots.

Yang et al. [20] designed a motion control system for a frog-inspired amphibious robot named FroBot. They analyzed the dynamic model of the FroBot and achieved smooth velocity control using a fuzzy Proportional Integral Derivative (PID) algorithm. Yue et al. [21] designed a control system for a spherical rolling robot and tried to solve the problem of underactuated vibrations during longitudinal movement. They proposed an adaptive hierarchical sliding mode control approach that incorporated data from a sensor-based state observer. Karavaev et al. [22] designed a system to control spherical rolling robots by propelling them with an internal omnivheel platform. There are various control algorithms for robots. Common ones are Central Pattern Generators (CPG), PID control, fuzzy control, neural network control, adaptive control, and sliding mode control. The control algorithm of existing works analyzed the dynamic and kinematic models to specify transitions from one steady-state motion to another. However, the subjects of all of these studies were either amphibious robots that were not spherical, or amphibious spherical robots that move by rolling. Thus, we cannot use any of the proposed control strategies with our amphibious spherical robot because it

is driven by a compound drive mechanism with both legs and water-jets.

In this study, we designed and implemented an improved amphibious spherical robot with the aim of conducting exploration tasks in complex amphibious environments. We developed a mechanical model and a dynamic model to predict the motions of the robot on land and in water, respectively. We used these models as the basis for an adaptive motion control system for the robot's compound drive system. The robot was equipped with an IMU to detect the inclination and motion information that served as feedback signals to the control system. We designed three types of walking gait for land-based motion. These provided different stabilities and motion speeds so that the robot could move through varied terrains. Furthermore, we improved the robot's stability as it climbs up slopes by adjusting the gaits using an online adjustment mechanism. In the case of underwater motion, where the system function varies continuously, we implemented the online system with a forgetting factor least squares algorithm. Robust underwater motion control was achieved by a generalized prediction control algorithm. Two digital coded optical fiber liquid level sensors are used to identify the water land transition environment. Two liquid level sensors are arranged slightly below the water line at the front end and the rear end of the robot shell. When the robot enters the water from land, the front of the robot enters the water first, and the liquid level sensor in the front outputs high level. Only when the rear of the robot also enters the water, that is, only when both sensors output high levels at the same time, the controller considers that the robot completely enters the water. At this time, the control system controls the propulsion mechanism to make switching action. Similarly, in the process of the robot moving from water to land, if the front of the robot has landed, the liquid level sensor in the front outputs a low level, and the controller thinks that the robot will land, so as to switch the propulsion mechanism to land propulsion mode. When the liquid level sensors at the back end output low level, and the control system considers that the robot has completely landed. We ensured the real-time performance of the robot motion control system and reduced its power consumption by implementing it on a Zynq-7000 SoC. We conducted experiments to evaluate the motion of the robot at different speeds in varied amphibious environments. The motion of the robot through 4 Degrees-of-Freedom (DoF) was stable and its top speeds were 11.9 cm/s on land and 14.7 cm/s in water. Moreover, the stability and mobility of the robot meet the requirements for practical applications in various terrains.

The main contributions of the work are as follows. We designed and built an improved version of our previously reported amphibious spherical robot with the aim of making the robot capable of completing motions in varied

amphibious environments. The contributions contain the configuration and synthesis of driving mechanism, the bio-inspired online gait adjustment mechanism for climbing slopes, the practical control improvements of an autonomous amphibious spherical robot as the embedded implementation of the microprocessor, and an online system recognition with a forgetting factor least squares algorithm.

The rest of this paper is organized as follows. We specify the mechanical and electronic design of our improved amphibious spherical robot in Sect. 2. The design and implementation of the adaptive quadruped gait for robotic motions on land are described in Sect. 3. The design of the generalized prediction control system for robotic motion in water is explained in Sect. 4. In Sect. 5, we report the results of our evaluation and experimental investigation. We draw conclusions and outline related future research in Sect. 6.

2 Amphibious Spherical Robots

2.1 Overview of Amphibious Spherical Robots

Our research team proposed a turtle-inspired amphibious spherical robot for precise and stealthy applications in narrow amphibious spaces. This proposal was followed up by studies into the robot's kinematic performance [23],

improved driving mechanism [24], robotic vision system [25] and mechanical reliability [26]. In this study, we designed and implemented an improved version of the amphibious spherical robot with more stable performance.

The improved amphibious spherical robot consists of an enclosed hemisphere hull (350 mm in diameter) and two openable quarter-sphere shells (360 mm in diameter), as shown in Fig. 1a, b. We installed four legs or compound driving units in a symmetrical configuration on the robot's lower hemisphere. Each compound driving unit was equipped with two servo motors and a water-jet motor. The motion of the robot legs is actuated with "RC servomotor", whose control input is a Pulse-Width Modulation (PWM) signal that actually encodes the desired rotor angle and is then controlled by an internal control board. The electronic system was installed inside the waterproof upper hemisphere, where it is protected from environmental damage.

In land mode, the two openable shells fold up and the robot turns into a quadruped that walks with four legs driven by servo motors and is capable of traversing rugged terrains. Wheeled locomotion is faster on flat ground, and it requires less power than legged locomotion. Hence, we designed the passive omni-directional wheel mechanism shown in Fig. 1f to assist the movement of the robot on land. Four omni-directional wheels were installed on a lifting platform that ascends and descends on two lead screws driven by servo

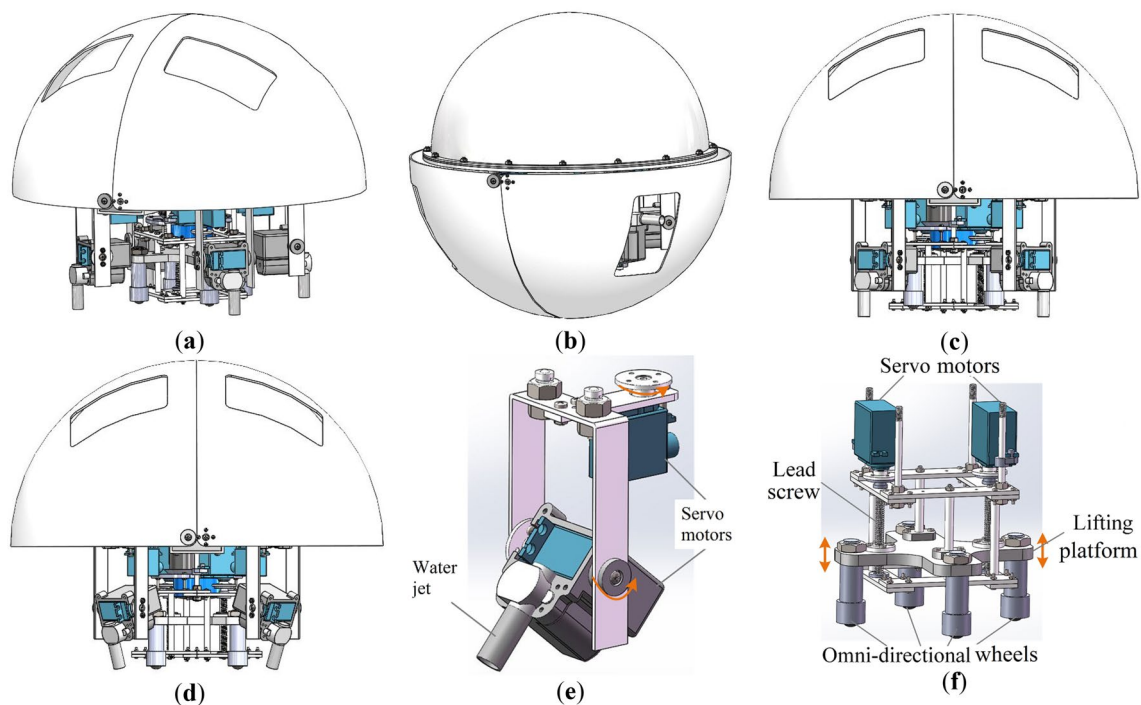


Fig. 1 Mechanical structure of the amphibious spherical robot. **a** Axonometric drawing of the robot in land mode; **b** Axonometric drawing of the robot in underwater mode; **c** Front view of the robot in

the walking state on land; **d** Front view of the robot in the roller-skating state on land; **e** Axonometric drawing of a leg; and **f** Axonometric drawing of the passive omni-directional wheel mechanism

motors. The lifting platform is lifted off the ground when the robot is in the walking state, as shown in Fig. 1c. The lifting platform descends to the ground as the robot transforms from the walking state to the roller-skating state. In this state, the four legs act as poles to move the robot forward, as shown in Fig. 1d.

In underwater mode, the two openable shells close and the robot becomes a ball-shaped underwater vehicle. The counterweight design puts the center of gravity on the axis of the ball so that the robot is suspended in water. The water-jet motors, which form the shanks of the robot, generate vectored thrusts in specialized directions. The magnitudes and directions of the vectored thrusts can be adjusted to enable the robot to move through four DoFs.

Figure 2 shows the diagram of the electronic system used in the amphibious spherical robot. The robot is powered by a set of Li-Po batteries with a total capacity of 24,000 mAh. The electronic system is restricted by the narrow load space and limited power resources. We fabricated the system around a Xilinx XC7Z020 SoC, which combines a dual-core ARM processor with Field Programmable Gate Arrays (FPGA). The ARM processor runs the Xilinx Linux operating system and provides a universal software platform for robotic applications. Functions involving real-time robotic applications, such as motor control and data pre-processing, were deployed on the FPGA section as digital circuits. We connected the SoC-based robot controller to a Raspberry Pi 3 by a network cable. The Raspberry Pi 3 serves as a co-processor for robotic vision applications. In our proposed robot

system, the depth meter and Global Positioning System (GPS) module are adopted to detect the position information and assist in real-time inertial navigation error correction, as shown in Fig. 2. The Micro-Electro-Mechanical System (MEMS) inertial sensing module consisted of a gyroscope (ITG-3200), an accelerometer (ADXL345), and an electronic compass (HMC5883L), with built-in Kalman filter and quaternion solution algorithm. It can output sensing data such as heading angle, angular velocity, and acceleration through I^2C interface. When the MEMS inertial sensing module is only used for attitude estimation, the complementary filtering method based on acceleration and angular velocity output can be used to give accurate pitch angle and roll angle.

2.2 Challenges for Amphibious Spherical Robot Motion Control

The mobility and adaptability of the amphibious spherical robot benefit from the compound drive mechanism and deformable mechanical structure. However, we have not yet overcome the challenge of designing a practical control system for this robot. A robust robotic platform is required before we can realize intelligent functions such as visual servoing, automatic navigation, etc.

The symmetrical structure and relatively high center of gravity of the robot mean that it turns over easily when moving through rugged terrestrial terrains. Meanwhile the interface between the robot's feet and the ground is very limited because the shanks are hollow jet tubes. This can

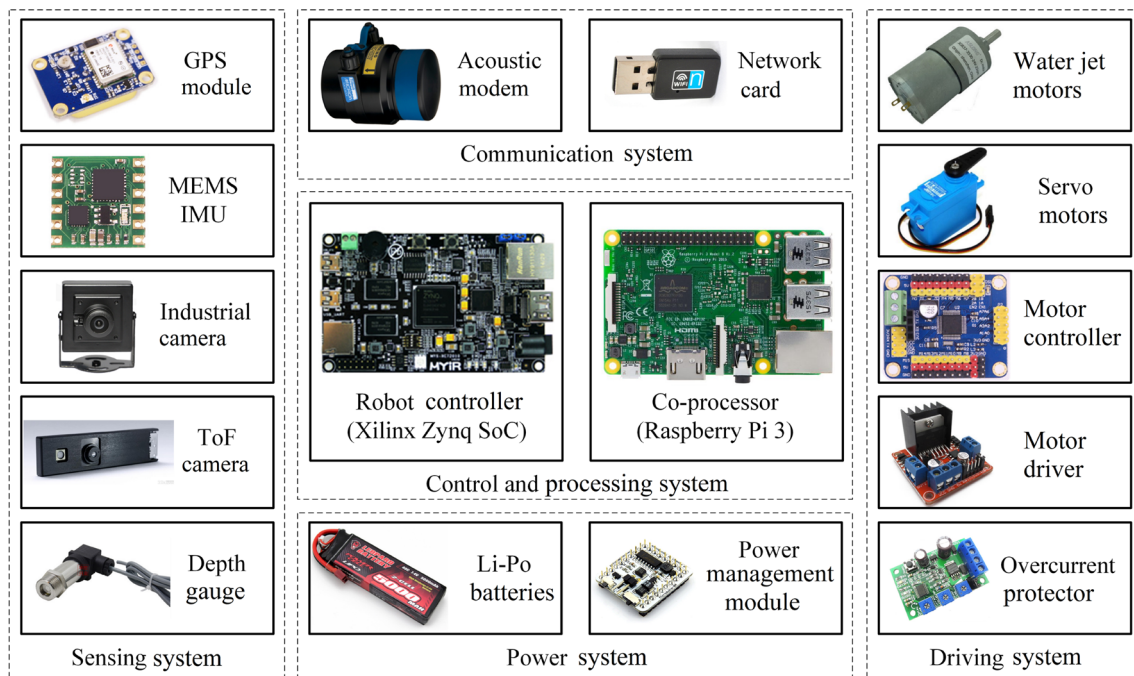


Fig. 2 Diagram of the electronic system used in the amphibious spherical robot

cause the robot to slip and vibrate violently in the direction of the slope as it tries to climb. Moreover, the robot still moves slowly. Faster motion may be essential in emergency circumstances. Thus, the kinetic stability, mobility and environmental adaptability should be taken into account when designing gait systems for amphibious spherical robots. A simple gait system was designed and evaluated with a prototype amphibious spherical robot in an earlier study [27]. The robot was only capable of walking on flat ground because we only implemented a simple, static gait. This robot has the potential to work on a variety of terrains, such as sand, mud, rocky ground, and slopes. Hence, we should design and implement a range of gaits with automatic adjustment mechanisms to improve their stability and viability.

The spherical shape of the robot makes it easier to control in underwater environments, and confers advantages in terms of stabilizing its inclination. However, this shape increases the fluid resistance during motion and the effects of the water flows. Consequently, if we vary the motion parameters of the robot, or it encounters variations in the water flow, its dynamic model will change. Additionally, some specific applications may require the robot to pick up or release loads. This would also have an effect on the dynamic model. Thus, the control system for underwater environments must be adaptive and robust to factors that could potentially interfere with the motion of the robot. The fuzzy PID algorithm [28] and neural network PID algorithm [29] were adopted to control the underwater motion of the prototype amphibious spherical robot. These advanced control algorithms passed the simulation experiments but did not perform as well in actual experiments. We decided that an adaptive control algorithm designed for gradually varied-coefficient systems would be more practical and better suited to the characteristics of the amphibious spherical robot.

3 Adaptive Quadruped Gaits for Motion on Land

3.1 Simplified Mechanical Model and Walk Gait of the Amphibious Spherical Robot

Figure 3 shows the motion of the robot's legs during the walking process. Servo motors rotate the upper and lower joints around the vertical and horizontal axes, respectively. Each step of the robot's walk is composed of the following four operations on the legs: first, the vertical servo motor rotates anticlockwise to lift the shank of the leg; second, the horizontal motor rotates anticlockwise to stretch the leg; third, the vertical servo motor rotates clockwise and the leg turns into the stance phase of the step; finally, the horizontal motor rotates clockwise and the body of the robot is pushed forward by the force of friction with the ground.

As shown in Fig. 4, we developed a simplified mechanical model of the robot so that we could analyze the walking motion quantitatively. We used three-dimensional coordinates with the origin at the center of the sphere. The black solid lines and the blue dashed lines indicate the range of motion of the legs through the horizontal and vertical planes. We determined that $r = 22\sqrt{2}$ mm, $l_1 = 12$ mm, $l_2 = 45$ mm, $l_3 = 50$ mm, $l_4 = 35$ mm and $l_5 = 65$ mm by mounting dimensions or part dimensions of a leg. α_{LF} and β_{LF} are, respectively, the angles that the left foreleg rotates through in the horizontal and vertical planes. The position of Leg_{LF} can be calculated from coordinates A_1 and A , which are defined by Eqs. (1) and (2):

$$\begin{cases} x_A = r + (l_2 + l_4 \cos \beta_{LF} + l_5 \sin \beta_{LF}) \cos \alpha_{LF} \\ y_A = r + (l_2 + l_4 \cos \beta_{LF} + l_5 \sin \beta_{LF}) \sin \alpha_{LF} \\ z_A = -l_1 - l_3 + l_4 \sin \beta_{LF} - l_5 \cos \beta_{LF}, \end{cases} \quad (1)$$

Fig. 3 Principles of the amphibious spherical robot's walk. **a** Top and front views of the leg lifting; **b** Top and front views of the leg stretching; **c** Top and front views of the leg hitting the ground; and **d** Top and front views of the leg pushing the ground

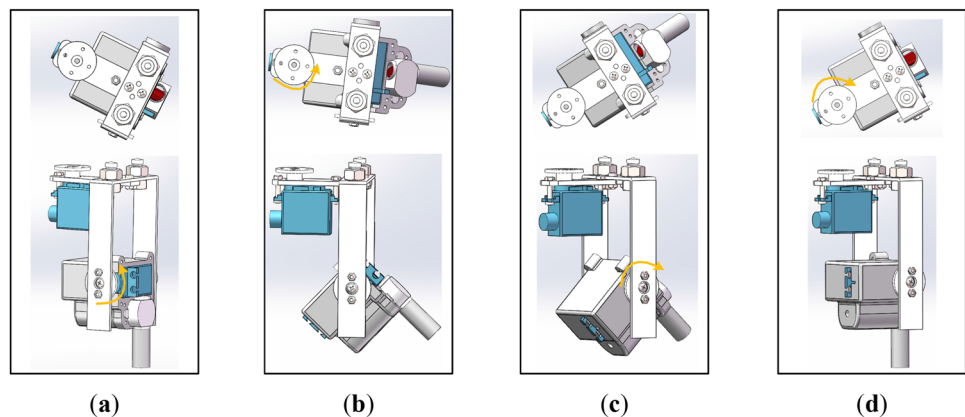
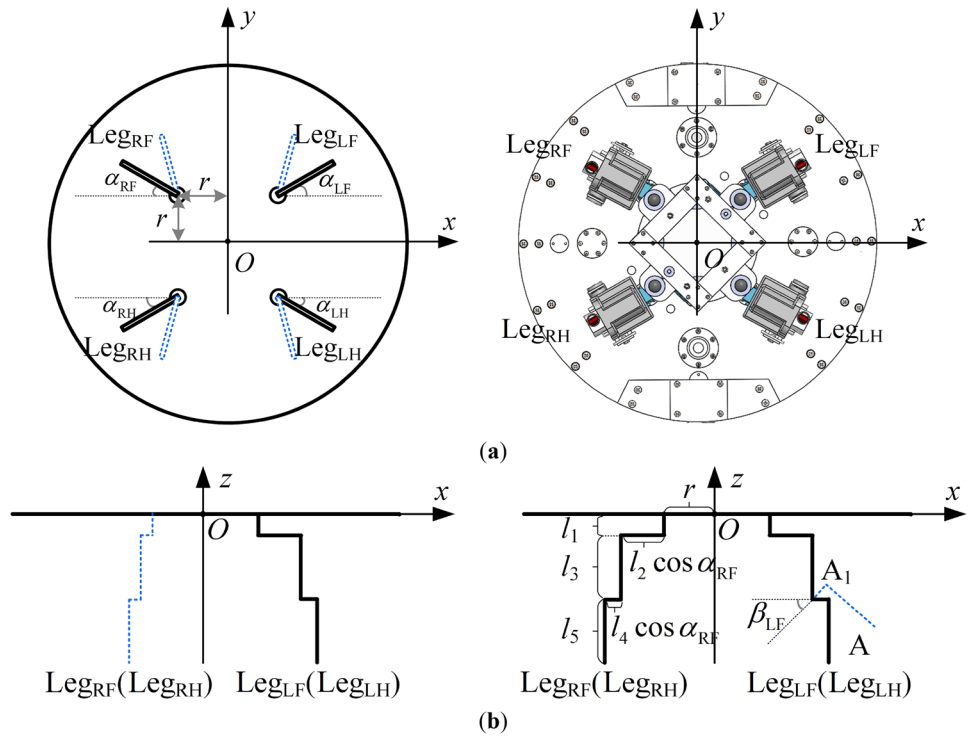


Fig. 4 Simplified mechanical model of the amphibious spherical robot. **a** Bottom view of the amphibious spherical robot; **b** Left-side view of the amphibious spherical robot



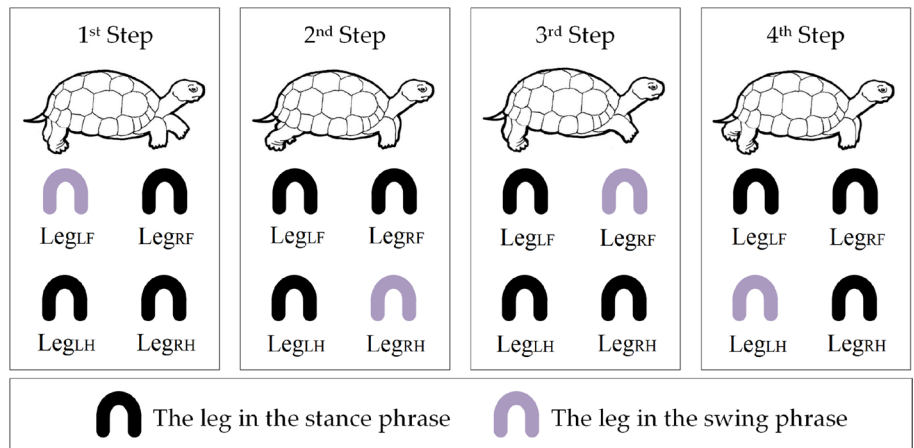
$$\begin{cases} x_{A1} = r + (l_2 + l_4 \cos \beta_{LF}) \cos \alpha_{LF} \\ y_{A1} = r + (l_2 + l_4 \cos \beta_{LF}) \sin \alpha_{LF} \\ z_{A1} = -l_1 - l_3 + l_4 \sin \beta_{LF}. \end{cases} \quad (2)$$

The commonly used classical gait of quadruped robot can be divided into five categories: walk, trot, pace, bound, and gallop [30]. Bound and gallop gaits have significant advantages in obstacle climbing ability and movement speed respectively, but they need the driving system to output strong explosive force to realize the jumping and taking off of the robot body. The gallop and bound gaits are not feasible because our spherical robot has limited power resources

and they require a strong driving force. As the robot is symmetrical and its center of gravity is on the z-axis, the pace and trot gaits may cause the robot to roll. To maintain stability, the body of the robot should be supported by three legs. We selected a walking gait to ensure that the robot remains stable when carrying counter weights and working in adverse environments.

Over one walking cycle, the robot's four legs move in the order "Leg_{LF}-Leg_{RH}-Leg_{RF}-Leg_{LH}", as shown in Fig. 5. More than half of the legs are in the stance phase at any point during the walking cycle. The legs in the swing phase followed the sequence of operations shown in Fig. 3. Motion is induced as the robot's center of gravity is gradually pushed forward. We can adjust the direction of the steps

Fig. 5 Diagram of the walking gaits



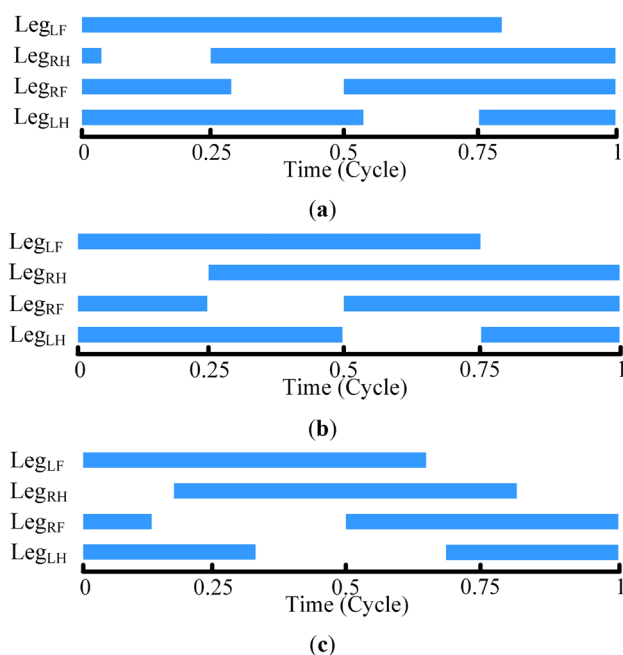


Fig. 6 Movement sequence charts for the designed gaits. **a** Stable walking gait; **b** standard walking gait; and **c** fast walking gait

Table 1 Overview of the designed gaits

Parameter	Stable walk gait	Standard walk gait	Fast walk gait
Time of an operation	0.1 s	0.1 s	0.1 s
Time of swing phase	0.4 s	0.4 s	0.4 s
Duty factor	80.0%	75.0%	66.7%
Time of a cycle	2.0 s	1.6 s	1.2 s
Maximum legs in stance phase	4	3	3
Minimum legs in stance phase	3	3	2
Unstable time	0 s	0 s	0.204 s
Very stable time	0.1 s	0 s	0 s

to turn the robot left or right, or allow it to move straight ahead. Thus, the four legs adopt stance phases and swing phases as they progress through each walking cycle. We can adapt the walking gait to different scenarios by varying the time spent in these phases. We designed the walking gaits used by the amphibious spherical robot by adjusting the duty factor of the stance phase, as shown in Fig. 6 and Table 1 [31]. The robot moves with the stable, standard, and fast walking gaits, when the duty factor is set to 80.0%, 75.0% and 66.7%, respectively. The four legs alternate between the swing and stance phases as they walk with the standard gait. This ensures that there are always three legs in the stance phase. The stable and fast walking gaits are achieved by, respectively, advancing and postponing the phase switch

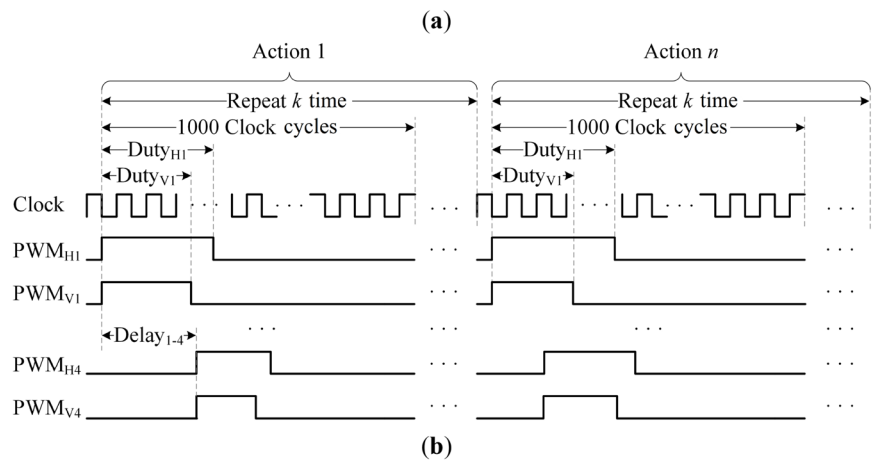
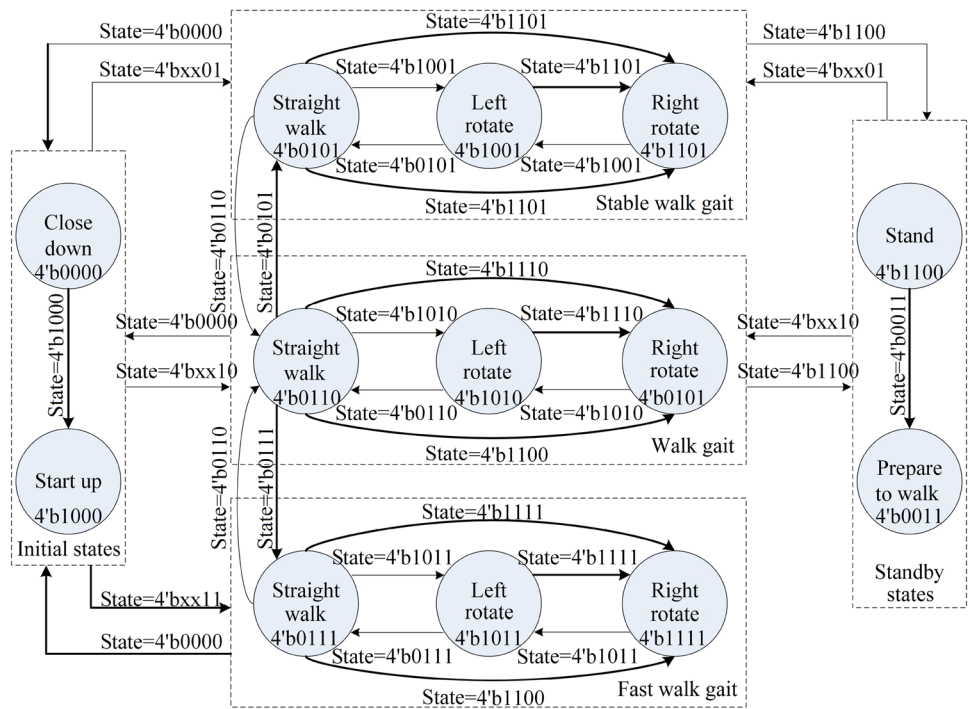
operations. The robot moves, respectively, 20% more slowly or quickly when it walks with these gaits instead of the standard gait. The stable gait keeps all four of the robot's legs in the stance phase during the phase switch operations. This improves the robot's stability on rugged terrains. In contrast, the fast gait makes the robot sway because its body is only supported by two legs during some parts of the fast walking cycle. Hence, the fast gait is only suitable for flat terrains.

The walking gaits can be implemented by configuring timers that output-specific sequences of PWM signals. As we needed to control eight servo motors in parallel, we implemented the walking gait system on digital circuits and deployed it to the FPGA section of the Xilinx XC7Z020 SoC. The walking gait system is stored on an Intellectual Property (IP) core that is connected to the ARM section of the SoC through an Advanced eXtension Interface (AXI) bus. The main robot control program runs on the ARM-Linux system and configures the gait system via the state register [32]. Figure 7a shows the state transition diagram of the gait system. After starting up, it can select the type of gait (stable walk gait, walk gait or fast walk gait) and direction of motion (walking straight, turning left or turning right). The robot can stop moving and stand still if it detects something interesting. As the robot cannot stand steadily on two legs in the stance phrase, it cannot transition directly from standby to the fast gait. The fast gait is only used for racing. Figure 7b shows the output control signals of the gait system. We use timers and frequency dividers to generate the eight PWM signals that control the servo motors. The frequencies of the clock signal and the PWM waves are 100 MHz and 100 Hz, respectively. We vary the gait by adjusting the durations of the intervals between the PWM signals (i.e., Delay_{1-4}). We can also change the speed of the motor and the motion of the robot by varying the number of output pulses from each leg operation, k . The robot can recognize terrains to adjust its walking gaits. It also detects and tracks targets using cameras, which were introduced in the references [18] and [19].

3.2 Adjustments for Climbing Slopes

There is not enough friction for the robot to climb steadily up a slope in the gaits described above, as shown in Fig. 8a. The robot would slip downwards as it steps, so its body would make continuous pitching motions. The frequent tilting motions may interfere with the normal operation of the devices installed inside its body. Hence, we must adjust the walking gait to ensure that the robot remains stable as it climbs up slopes. We keep the robot platform horizontal and lowered its center of gravity by varying the rotations of the vertical motors, as shown in Fig. 8b. The rotations of the vertical motors can be incremented by adjustment angles,

Fig. 7 Implementation of the robotic gait system. **a** State transition diagram of the robotic gait system; and **b** signal sequence diagram of the robotic gait system



φ_{LF} , φ_{RF} , φ_{LH} , and φ_{RH} . From their initial positions, the left foreleg and left hind leg land at positions *A* and *D*, which are calculated using Eqs. (3) and (4):

$$\begin{cases} x_A = r + [l_2 + l_4 \cos(\beta_{LF} + \varphi_{LF}) + l_5 \sin(\beta_{LF} + \varphi_{LF})] \cos \alpha_{LF} \\ y_A = r + [l_2 + l_4 \cos(\beta_{LF} + \varphi_{LF}) + l_5 \sin(\beta_{LF} + \varphi_{LF})] \sin \alpha_{LF} \\ z_A = -l_1 - l_3 + l_4 \sin(\beta_{LF} + \varphi_{LF}) - l_5 \cos(\beta_{LF} + \varphi_{LF}), \end{cases} \quad (3)$$

$$\begin{cases} x_D = -r - [l_2 + l_4 \cos(\beta_{LH} + \varphi_{LH}) + l_5 \sin(\beta_{LH} + \varphi_{LH})] \cos \alpha_{LH} \\ y_D = -r - [l_2 + l_4 \cos(\beta_{LH} + \varphi_{LH}) + l_5 \sin(\beta_{LH} + \varphi_{LH})] \sin \alpha_{LH} \\ z_D = -l_1 - l_3 + l_4 \sin(\beta_{LH} + \varphi_{LH}) - l_5 \cos(\beta_{LH} + \varphi_{LH}). \end{cases} \quad (4)$$

where $\beta_{LF} = \beta_{LH} = 0$, $\alpha_{LF} = \pi/4$, and $\alpha_{LH} = -\pi/4$. As \overline{AD} was vertical with the axis *y*, the coordinates of *A* and *D* should satisfy the condition defined in Eq. (5).

$$z_A - z_D = (y_A - y_D) \tan \delta. \quad (5)$$

δ , which represents the angle of the slope, can be measured approximately online by the MEMS IMU. The adjustment angles can be determined by solving Eqs. (3), (4) and (5). We added a slight random disturbance to δ during the calculation process to reflect the fact that most slopes do not have constant gradients. We minimized the error in Eq. (5) by selecting adjustment angles corresponding to different gradients, as shown in Fig. 9. The FPGA-based gait system

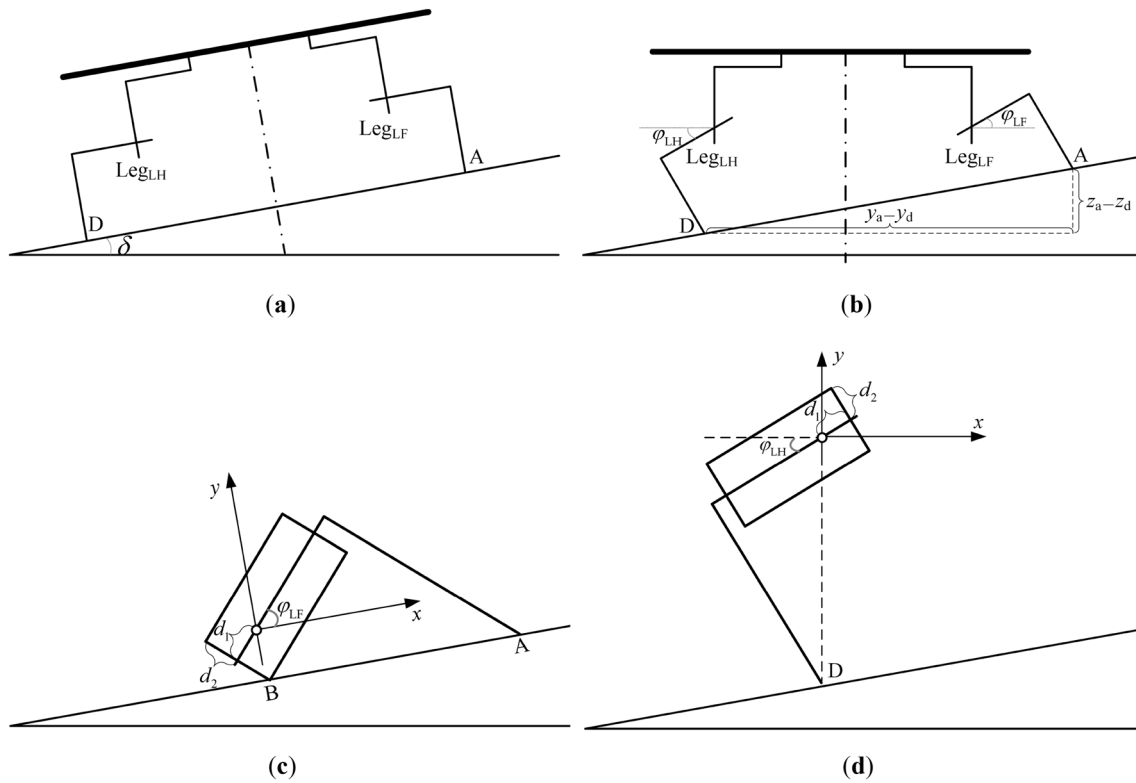


Fig. 8 Analysis of the slope climbing motion. **a** Slope climbing without gait adjustment; **b** slope climbing with gait adjustment; **c** critical case with the left foreleg in slope climbing mode; and **d** critical case with the left foreleg in slope climbing mode

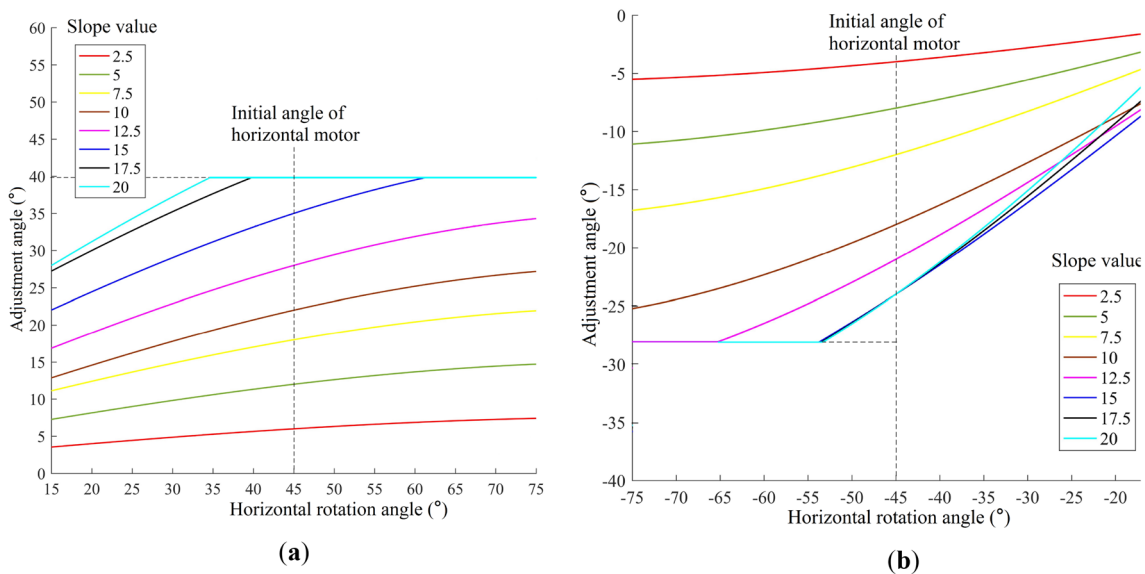


Fig. 9 Adjustment angles of the legs toward different slopes. **a** Adjustment angle of the forelegs toward different slopes; and **b** adjustment angles of the hind legs toward different slopes

uses a look-up table compiled from the curves in Fig. 9 to adjust the motions of the robot’s legs so that it can climb slopes successfully.

Limitations in the mechanical structure of the legs mean that the robot can only climb gentle slopes. Figure 8c shows an example of a critical limitation, where the left foreleg

is lying on the ground. The coordinates of A and B can be calculated using Eqs. (6) and (7):

$$\begin{cases} x_A = l_4 \cos \varphi_{LF} + l_5 \sin \varphi_{LF} \\ y_A = l_4 \sin \varphi_{LF} - l_5 \cos \varphi_{LF} \end{cases}, \tag{6}$$

$$\begin{cases} x_B = -\sqrt{d_1^2 + d_2^2} \cdot \cos[\varphi_{LF} + \arctan(d_2/d_1)] \\ y_B = -\sqrt{d_1^2 + d_2^2} \cdot \sin[\varphi_{LF} + \arctan(d_2/d_1)] \end{cases}, \tag{7}$$

where $d_1=21.5$ mm, $d_2=20$ mm, and $y_A=y_B$. Figure 8c shows another critical limiting case where the robot is only supported by its left hind leg. The maximum adjustment angles of Leg_{LF} and Leg_{LH} are $\varphi_{LFmax}=33.7^\circ$ and $\varphi_{LH}=21.8^\circ$, respectively. Form the Fig. 8, we can obtain the maximum angle of the slope $\delta_{max}=[\varphi_{LFmax}, \varphi_{LHmax}]_{min}=21.8^\circ$. Considering the structural deformation error of the robot, the flatness of the slope, and potential subsidence, the maximum angle of the slope δ_{max} should be less than 21.8° . So, to avoid the front leg joint contacting the ground, we set the maximum climbing angle of the robot as 20° .

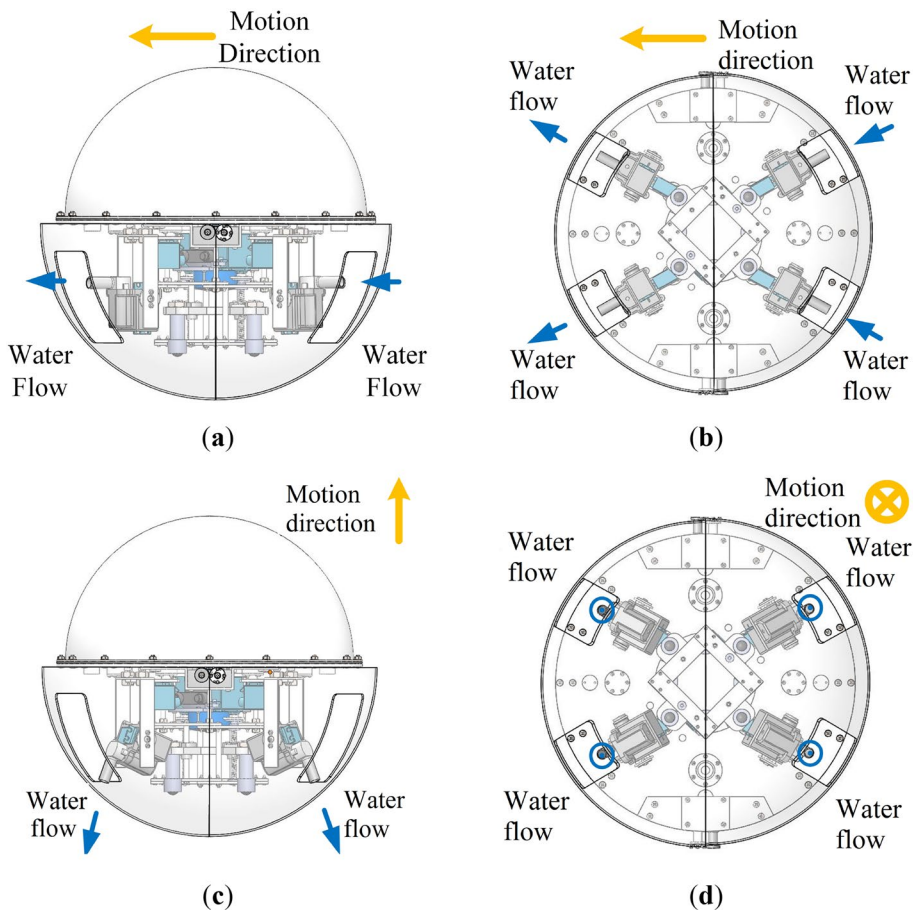
4 Generalized Prediction Control for Motion in Water

4.1 Simplified Underwater Dynamic Model

Unlike most autonomous underwater vehicles, our amphibious spherical robot has four vectored thrusters. These give it 4-DoFs of underwater movement. The characteristics of the robot’s motion vary because it has a deformable mechanical structure and carries variable loads. Moreover, as the robot is ball-shaped, its motion is easily affected by the flow of the surrounding water [33]. Thus, the parameters governing the robot’s dynamic model vary continuously. It is difficult to control robots robustly using classical PID algorithms, as they are highly reliant on the robots’ system transfer functions. It is possible that adaptive motion control algorithms can enable robots to respond to environmental variations and disturbances [34–36].

We simplified the control process using combinations of the horizontal and vertical motions as the 4-DoFs of underwater motion, as shown in Fig. 10. In the horizontal motion mode, we decouple the four legs by crossing them at certain angles. The four water-jet motors spurt or pump water to provide horizontal

Fig. 10 Principles of the amphibious spherical robot’s underwater motion. **a** Horizontal motion (lateral view); **b** horizontal motion (bottom view); **c** vertical motion (lateral view); and **d** vertical motion (lateral view)



thrusts in specified directions. We translate and rotate the robot by adjusting the speed and rotary direction of the motors. In the vertical motion mode, the robot surfaces and dives by making vertical thrusts with the water-jet motors.

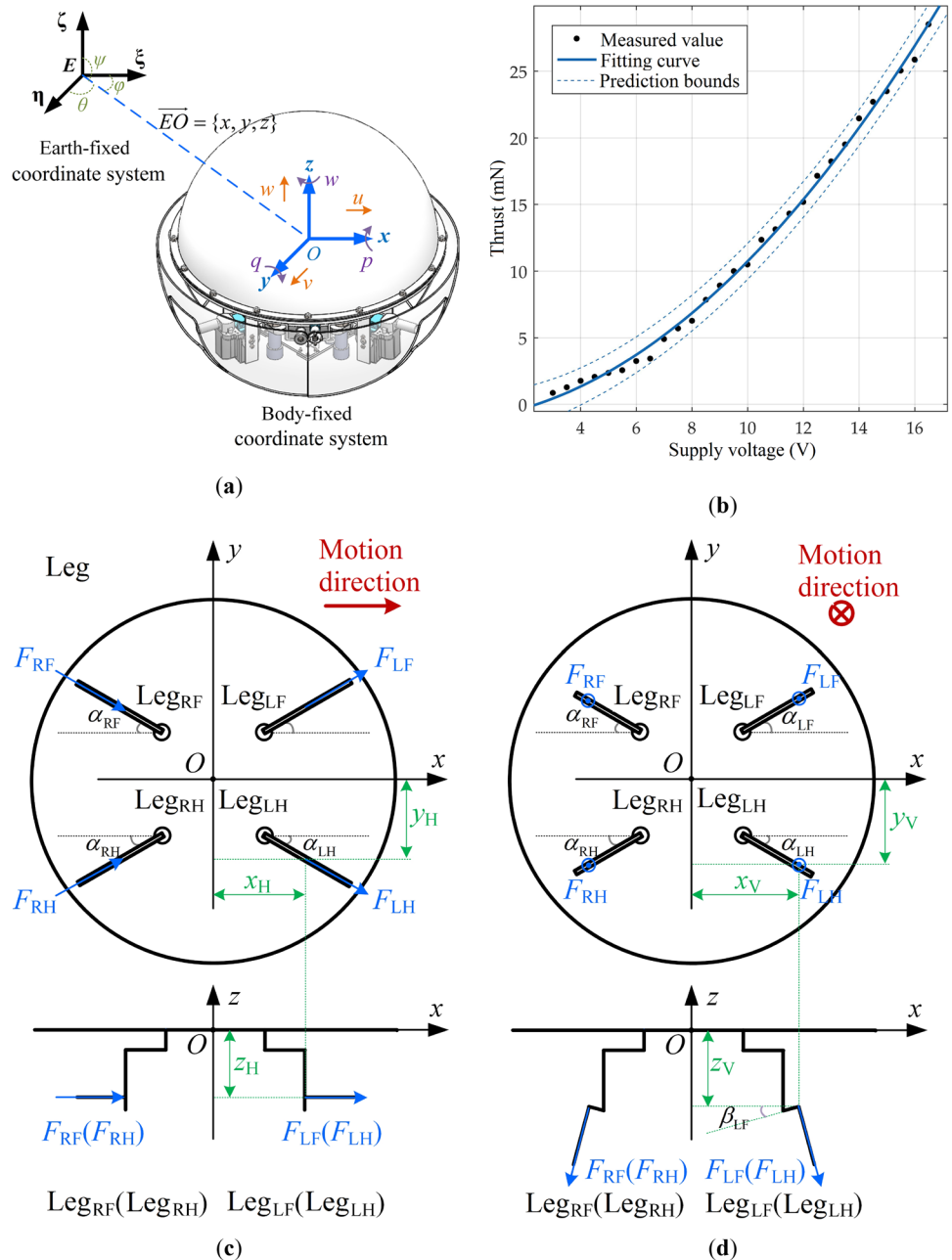
We developed a dynamic model of the robot so that we could explore the form of its system transfer function. As shown in Fig. 11a, we defined a coordinate system (called the earth-fixed coordinate system) with respect to the earth and a coordinate system (called the body-fixed coordinate system) with respect to the robot itself. We took the center of the ball as the origin of the body-fixed coordinate system.

The coordinates of the center of mass and the center of buoyancy are, respectively, $r_B = [0, 0, z_B]^T = [0, 0, -0.0045]^T$ and $r_G = [0, 0, z_G]^T = [0, 0, 0.062]^T$. The location and the inclination of the robot can be calculated using Eq. (8):

$$\eta = \{x, y, z, \varphi, \theta, \psi\}, \tag{8}$$

where x , y , and z represent the Cartesian position in the earth-fixed coordinate system, φ is the roll angle, θ is the pitch angle, and ψ is the yaw angle. The state of the robot's motion in the body-fixed coordinate system is:

Fig. 11 Kinematic model of the amphibious spherical robot in underwater environments. **a** Kinetic coordinate system and static coordinate system; **b** the thrust forces induced by the water-jets; **c** layout of the thrust forces as the robot moves horizontally; and **d** layout of the thrust forces when the robot moves vertically



$$v = \{u, v, w, p, q, r\}, \tag{9}$$

where $u, v,$ and w are the components of the linear velocity in the surge, sway, and heave directions, respectively. And $p, q,$ and r are the components of the angular velocities in the roll, pitch, and yaw, directions, respectively. Equation (10) is a dynamic model of the robot with 6-DoFs:

$$(\mathbf{M}_{RB} + \mathbf{M}_A)\dot{v} + (\mathbf{C}_{RB}(v) + \mathbf{C}_A(v))v + (\mathbf{D}_l + \mathbf{D}_q(v))v + \mathbf{g}(\eta) = \tau, \tag{10}$$

where $\mathbf{M}_{RB} \in \mathbb{R}^{6 \times 6}$ is the matrix representing the mass of the rigid body, $\mathbf{M}_A \in \mathbb{R}^{6 \times 6}$ is the matrix representing the mass added to the robot, $\mathbf{C}_{RB}(v) \in \mathbb{R}^{6 \times 6}$ is the Coriolis force matrix of the rigid body, $\mathbf{C}_A(v) \in \mathbb{R}^{6 \times 6}$ is the Coriolis force matrix of the added mass, $\mathbf{D}_l \in \mathbb{R}^{6 \times 6}$ is the linear damping matrix, $\mathbf{D}_q(v) \in \mathbb{R}^{6 \times 6}$ is the nonlinear damping matrix, $\mathbf{g}(\eta) \in \mathbb{R}^{6 \times 1}$ is the gravity and buoyancy vector, and $\tau \in \mathbb{R}^{6 \times 1}$ is the vector representing the thrust generated by the water-jet motors.

$$\mathbf{M}_A = \begin{bmatrix} -X_{\dot{u}} & 0 & 0 & 0 & 0 & 0 \\ 0 & -Y_{\dot{v}} & 0 & 0 & 0 & -Y_{\dot{r}} \\ 0 & 0 & -Z_{\dot{w}} & 0 & -Z_{\dot{q}} & 0 \\ 0 & 0 & 0 & -K_p & 0 & 0 \\ 0 & 0 & -M_{\dot{w}} & 0 & -M_{\dot{q}} & 0 \\ 0 & -N_{\dot{v}} & 0 & 0 & 0 & -N_{\dot{r}} \end{bmatrix} \tag{12}$$

$$= \begin{bmatrix} 10.300 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10.300 & 0 & 0 & 0 & 0 \\ 0 & 0 & 10.300 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.088 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.088 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.065 \end{bmatrix},$$

where m represents the mass of the robot. The terms of \mathbf{M}_{RB} are determined by the mass distribution of the robot and the terms of \mathbf{M}_A are determined by the geometrical shape of the robot and the density of the fluid. We used finite element analysis tools to calculate the numerical solutions of \mathbf{M}_A and \mathbf{M}_{RB} .

$\mathbf{C}_{RB}(v)$ and $\mathbf{C}_A(v)$ are defined by:

$$\mathbf{C}_{RB} = \begin{bmatrix} 0 & 0 & 0 & mz_G r & -m(x_G q - w) & -m(x_G r + v) \\ 0 & 0 & 0 & -mw & m(z_G r + x_G p) & mu \\ 0 & 0 & 0 & m(v - z_G p) & -m(z_G q + u) & mx_G p \\ -mz_G r & mw & m(z_G p - v) & 0 & -(I_{zx} p - I_z r) & -I_y q \\ m(x_G q - w) & -m(z_G r + x_G p) & m(z_G q + u) & I_{zx} p - I_z r & 0 & I_x p - I_{zx} r \\ m(x_G r + v) & -mu & -mx_G p & I_y q & I_{zx} r - I_x p & 0 \end{bmatrix}, \tag{13}$$

$$\mathbf{C}_A = \begin{bmatrix} 0 & 0 & 0 & 0 & -(Z_{\dot{w}} w + Z_{\dot{q}} q) & Y_{\dot{v}} v + Y_{\dot{r}} r \\ 0 & 0 & 0 & Z_{\dot{w}} w + Z_{\dot{q}} q & 0 & -X_{\dot{u}} u \\ 0 & 0 & 0 & -(Y_{\dot{v}} v + Y_{\dot{r}} r) & X_{\dot{u}} u & 0 \\ 0 & -(Z_{\dot{w}} w + Z_{\dot{q}} q) & Y_{\dot{v}} v + Y_{\dot{r}} r & 0 & -(N_{\dot{v}} v + N_{\dot{r}} r) & M_{\dot{w}} w + M_{\dot{q}} q \\ Z_{\dot{w}} w + Z_{\dot{q}} q & 0 & -X_{\dot{u}} u & N_{\dot{v}} v + N_{\dot{r}} r & 0 & -K_p p \\ -(Y_{\dot{v}} v + Y_{\dot{r}} r) & X_{\dot{u}} u & 0 & -(M_{\dot{w}} w + M_{\dot{q}} q) & K_p p & 0 \end{bmatrix}. \tag{14}$$

As the mechanical structure of the robot is symmetrical and the pitch and yaw movements are passive, \mathbf{M}_{RB} and \mathbf{M}_A are defined as follows:

The terms of $\mathbf{C}_{RB}(v)$ and $\mathbf{C}_A(v)$ depend on the terms of \mathbf{M}_A and \mathbf{M}_{RB} . As the cruising speed of the robot is relatively low, we were able to use Stokes' law to calculate \mathbf{D}_l :

$$\mathbf{M}_{RB} = \begin{bmatrix} m & 0 & 0 & 0 & mz_G & 0 \\ 0 & m & 0 & -mz_G & 0 & mx_G \\ 0 & 0 & m & 0 & -mx_G & 0 \\ 0 & -mz_G & 0 & I_x & 0 & -I_{zx} \\ mz_G & 0 & -mx_G & 0 & I_y & 0 \\ 0 & mx_G & 0 & -I_{zx} & 0 & I_z \end{bmatrix} = \begin{bmatrix} 9.408 & 0 & 0 & 0 & 0.0423 & 0 \\ 0 & 9.408 & 0 & -0.0423 & 0 & 0 \\ 0 & 0 & 9.408 & 0 & 0 & 0 \\ 0 & -0.0423 & 0 & 0.0664 & 0 & 0 \\ 0.0423 & 0 & 0 & 0 & 0.0664 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.0826 \end{bmatrix}, \tag{11}$$

$$D_1 v = C_l \times \text{diag}\{u, v, w, p, q, r\} = 1.01 \times 10^{-4} \times \text{diag}\{u, v, w, p, q, r\}, \tag{15}$$

where $C_l = 1.01 \times 10^{-4}$ is the linear viscosity coefficient of liquid at 20 °C. $D_q(v)$ was calculated from the drag:

$$D_q(v) = \rho v^2 C_d A / 2, \tag{16}$$

where C_d is the drag coefficient, which is determined by the robot’s geometry and the characteristics of the flow, ρ is the mass density of the fluid, v is the velocity of the flow relative to the robot, and A is the reference area. The numerical solutions of $D_q(v)$ were calculated by analyzing the Reynolds number in Eq. (17).

$$D_q(v)v = \frac{1}{2} C_d \rho V_r^2 A = \begin{bmatrix} 0.471 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.471 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.499 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \end{bmatrix}. \tag{17}$$

$g(\eta)$, which is relevant to the design of the counterweight, is calculated using Eq. (18):

$$g(\Theta) = \begin{bmatrix} 0^{3 \times 1} \\ (z_G W - z_B B) \cos \theta \sin \phi \\ (z_G W - z_B B) \sin \theta + (x_G W - x_B B) \cos \theta \cos \phi \\ -(x_G W - x_B B) \cos \theta \sin \phi \end{bmatrix} = \begin{bmatrix} 0^{3 \times 1} \\ -0.9856 \cos \theta \sin \phi \\ -0.9856 \sin \theta \\ 0 \end{bmatrix}, \tag{18}$$

where W and B represent the gravity and the buoyancy, respectively.

The thrust vector was determined by analyzing the layout and output force from the water-jet motors as follows:

$$\tau = LU, \tag{19}$$

where L is the layout parameter matrix, and $U = [F_{LF}, F_{RF}, F_{RH}, F_{LH}]$ is the thrust vector generated by the water-jet motors. As shown in Fig. 11b, the thrust of a water-jet motor, which is controlled by the supply voltage, is defined by an empirical formula based on data from a force measuring system as per Eq. (20):

$$F = 0.094U^2 + 0.249U - 1.167. \tag{20}$$

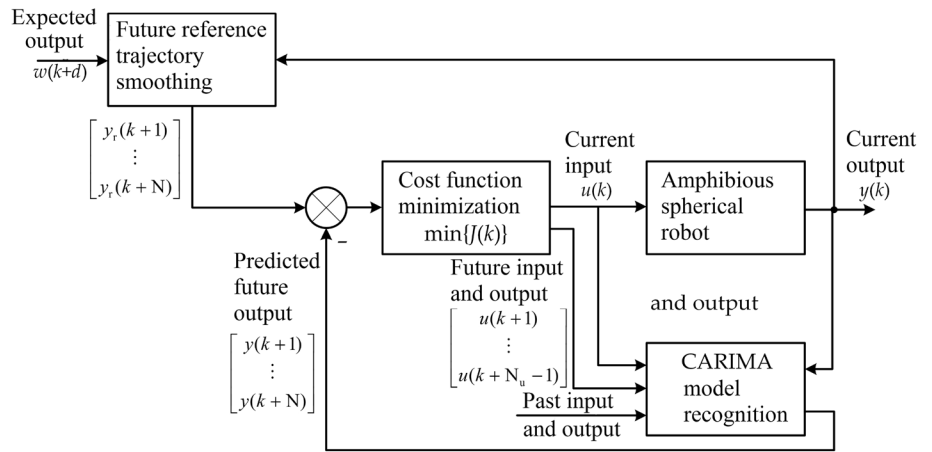
Figure 11c, d shows the layouts of the thrusts provided by the water-jet motors. In the horizontal motion mode, the layout parameter matrix is approximately equal to Eq. (21):

$$L = \begin{bmatrix} \cos \alpha_{LF} & \cos \alpha_{RF} & \cos \alpha_{RH} & \cos \alpha_{LH} \\ \sin \alpha_{LF} & -\sin \alpha_{RF} & \sin \alpha_{RH} & -\sin \alpha_{LH} \\ 0 & 0 & 0 & 0 \\ (z_G + z_H) \sin \alpha_{LF} & -(z_G + z_H) \sin \alpha_{RF} & (z_G + z_H) \sin \alpha_{RH} & -(z_G + z_H) \sin \alpha_{LH} \\ -(z_G + z_H) \cos \alpha_{LF} & -(z_G + z_H) \cos \alpha_{RF} & (z_G + z_H) \cos \alpha_{RH} & -(z_G + z_H) \cos \alpha_{LH} \\ x_H \sin \alpha_{LF} - y_H \cos \alpha_{LF} & x_H \sin \alpha_{RF} - y_H \cos \alpha_{RF} & y_H \cos \alpha_{RH} - x_H \sin \alpha_{RH} & y_H \cos \alpha_{LH} - x_H \sin \alpha_{LH} \end{bmatrix}, \tag{21}$$

where x_H , y_H , and z_H are the coordinates of the water-jet motors in the body-fixed coordinate system. In the vertical motion mode, the layout parameter matrix is approximately equal to Eq. (22).

$$L = \begin{bmatrix} \sin \beta_{LF} \cos \alpha_{LF} & \sin \beta_{LF} \sin \alpha_{LF} & -\cos \beta_{LF} & -y_v \cos \beta_{LF} & x_v \cos \beta_{LF} & x_v \sin \beta_{LF} \sin \alpha_{LF} - y_v \sin \beta_{LF} \cos \alpha_{LF} \\ -\sin \beta_{RF} \cos \alpha_{RF} & \sin \beta_{RF} \sin \alpha_{RF} & -\cos \beta_{RF} & -y_v \cos \beta_{RF} & -x_v \cos \beta_{RF} & y_v \sin \beta_{RF} \cos \alpha_{RF} - x_v \sin \beta_{RF} \sin \alpha_{RF} \\ -\sin \beta_{RH} \cos \alpha_{RH} & -\sin \beta_{RH} \sin \alpha_{RH} & -\cos \beta_{RH} & y_v \cos \beta_{RH} & -x_v \cos \beta_{RH} & x_v \sin \beta_{RH} \sin \alpha_{RH} - y_v \sin \beta_{RH} \cos \alpha_{RH} \\ \sin \beta_{LH} \cos \alpha_{LH} & -\sin \beta_{LH} \sin \alpha_{LH} & -\cos \beta_{LH} & y_v \cos \beta_{LH} & x_v \cos \beta_{LH} & y_v \sin \beta_{LH} \cos \alpha_{LH} - x_v \sin \beta_{LH} \sin \alpha_{LH} \end{bmatrix}^T. \tag{22}$$

Fig. 12 The conceptual basis of the adaptive generalized prediction system for controlling the robot in underwater environments



We can use the dynamic model to determine the system transfer function of the robot. The simplified underwater dynamic model suggests that the system transfer function is second order. The hydraulic resistance and inertia effects mean that the robot is a delay system. In theory, the parameters of the dynamic model can be acquired either numerically or experimentally. However, some key parameters, such as the damping matrix, are determined by the current motion status of the robot and environmental factors that vary continuously. Thus, the parameters of the control system should be adjusted online as the system transfer function changes.

4.2 Generalized Prediction Control Mechanism

From the existing adaptive control algorithms, the generalized prediction control method derives advantages from model predictive control and dynamic matrix control [37]. Moreover, it is robust to unmatched system models and is capable of overcoming system lag [38]. These characteristics are required from the control system of the amphibious spherical robot [39–44]. Figure 12 shows the main concepts used in the adaptive generalized prediction control algorithm. This algorithm was adopted to control the course and depth of the robot in underwater environments.

Let us consider the course control of the robot in the following example. As the robot moves slowly underwater, the parameters of its system transfer function vary slowly. Thus, we implemented online system recognition with the forgetting factor least squares algorithm. We used the underwater dynamic model established in Sect. 4.1 to derive the controlled auto-regressive integrated moving average model (CARIMA) defined by Eq. (23). This model provides us with the following system model of the robot in the underwater horizontal motion mode:

$$(1 + a_1z^{-1} + a_2z^{-2})y(k) = z^{-d}(b_0 + b_1z^{-1})u(k) + \xi(k)/(1 - z^{-1}), \tag{23}$$

where $y(k)$ represents the k th system output or the course angle of the robot, $u(k)$ is the k th system input or the controlled water-jet motor variable, and $\xi(k)$ is a random disturbance term from a Gaussian distribution. To simplify the control process, the robot’s four water-jet motors were divided into two sets. The two motors in each set are controlled to output the same force, which is $F_{LF} = F_{RF}$ and $F_{LH} = F_{RH}$ in the scene shown in Fig. 11c. That is to say that $u(k)$ is a 2 by 1 vector, and each of its elements control a set of motors. The k th loop is represented by Eq. (24):

$$\begin{cases} \hat{\theta}(k) = \hat{\theta}(k - 1) + K(k)[y(k) - \varphi^T(k)\hat{\theta}(k - 1)] \\ K(k) = \frac{P(k-1)\varphi(k)}{\lambda + \varphi^T(k)P(k-1)\varphi(k)} \\ P(k) = \frac{1}{\lambda} [I - K(k)\varphi^T(k)]P(k - 1) \end{cases}, \tag{24}$$

where $\lambda \in [0.9, 1]$ is the forgetting factor, $\varphi(k) = [-\Delta y(k - 1), -\Delta y(k - 2), -\Delta u(k - 1), -\Delta u(k - 2)]^T$ is the past system input and output increment vector, $\hat{\theta}(k) = [a_1(k), a_2(k), b_0(k), b_1(k)]^T$ is the estimated model parameter vector, and $K(k)$ and $P(k)$ are intermediate results.

The generalized prediction controller uses the calculated system transfer function to predict the robot’s future course angle sequence ($[y(k + 1), \dots, y(k + N)]$) using past, current, and future control inputs, and past and current course angles. Then, it selects future control inputs ($[u(k), \dots, u(k + N_u - 1)]$) to ensure that the course angle will finally reach the expected output w . Suppose that the robot is a d th-order-delay system and is set to follow the predefined course angle $w(k + d)$. We guarantee that the course angle makes smooth transitions by pre-processing the reference trajectory using the first-order smoothing model in Eq. (25):

$$\begin{cases} y_r(k) = y(k) \\ y_r(k + j) = \alpha y_r(k + j - 1) + (1 - \alpha)w, k = 1, 2, \dots, N, \end{cases} \tag{25}$$

where $\alpha \in [0, 1)$ is the softness factor, N is the prediction step, and $y_r(k)$ is the future reference trajectory. We used a

gradient descent optimization approach to find the future control inputs that minimize the cost function J :

$$\min J = \min E \left\{ \sum_{j=N_1}^{N_2} [y(k+j) - y_r(k+j)]^2 + \sum_{j=1}^{N_u} [\gamma_i \Delta u(k+j-1)] \right\}, \quad (26)$$

where N_1 and N_2 are, respectively, the maximum and minimum prediction lengths, γ_i is the weighting coefficient, and $\Delta u(k+j-1)$ is the future control input increment. The search process is equivalent to solving the following Diophantine equation:

$$\begin{cases} C(z^{-1})y^*(k+j|k) = G_j(z^{-1})y(k) + F_j(z^{-1})\Delta u(k+j-1) \\ C(z^{-1}) = \bar{A}(z^{-1})E_j(z^{-1}) + z^{-j}G_j(z^{-1}) \\ F_j(z^{-1}) = B_j(z^{-1})E_j(z^{-1}), \end{cases} \quad (27)$$

where $y^*(k+j|k)$ represents the values expected to minimize the cost function, and $\bar{A}(z^{-1})$, $B_j(z^{-1})$, $C(z^{-1})$, $E_j(z^{-1})$, $F_j(z^{-1})$ and $G_j(z^{-1})$ are determined by the system transfer function. The corresponding control input is defined as follows:

$$u(k) = u(k-1) + [1, 0, \dots, 0](F_1^T F_1 + \Gamma)^{-1} F_1^T [Y_r - F_2 \Delta U(k-j) - \mathbf{G}Y(k)], \quad (28)$$

where F_1 , F_2 , and \mathbf{G} are the solutions of the Diophantine equation, $\Gamma = \text{diag}(\gamma_1, \gamma_2, \gamma_{N_u})$ is the weighting coefficient vector, $Y_r = [y_r(k+N_1), y_r(k+N_1+1), \dots, y_r(k+N_2)]^T$ is the smoothed reference trajectory, and $Y = [y(k+N_1), y(k+N_1+1), \dots, y(k+N_2)]^T$ is the predicted course angle vector.

In our real-world experiments, we set the sample interval of the generalized prediction controller to 100 ms and its parameters to $\alpha = 0.8$, $\lambda = 0.95$, $N = 5$, $N_1 = d = 3$, $N_2 = 10$, and $N_u = 1$. The values of $\mathbf{K}(0)$ and $\mathbf{P}(0)$ were set to $10^6 \mathbf{I}$ and 0.05, respectively. The controller was implemented as a soft-real-time program running in the PS of the SoC.

5 Evaluation and Experimental Results

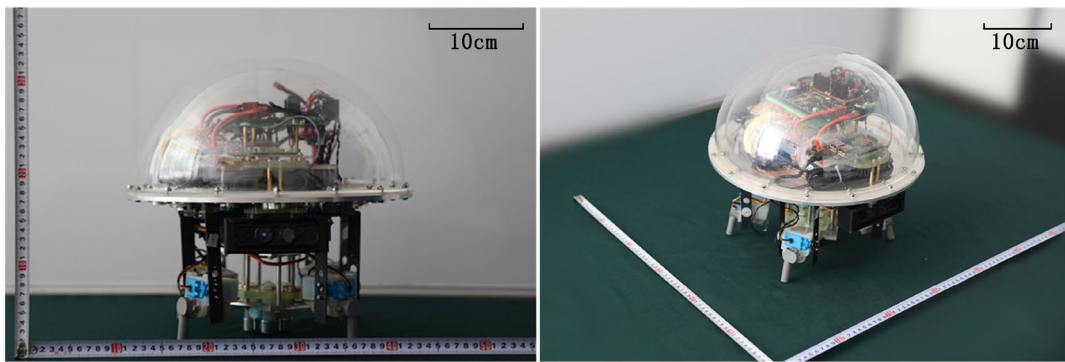
Figure 13a shows a prototype of our amphibious spherical robot. Only the upper hemisphere of the robot is designed to be waterproof. So, we balanced the robot's gravity and buoyancy and the robot could be suspended in the water without driving. We evaluated the performance of the robot and the proposed adaptive motion control system by conducting experiments in three phases, both on land and underwater.

(1) In the motion performance evaluation phase, the amphibious spherical robot was configured to walk on land and swim in water, as shown in Fig. 13b–e. In the

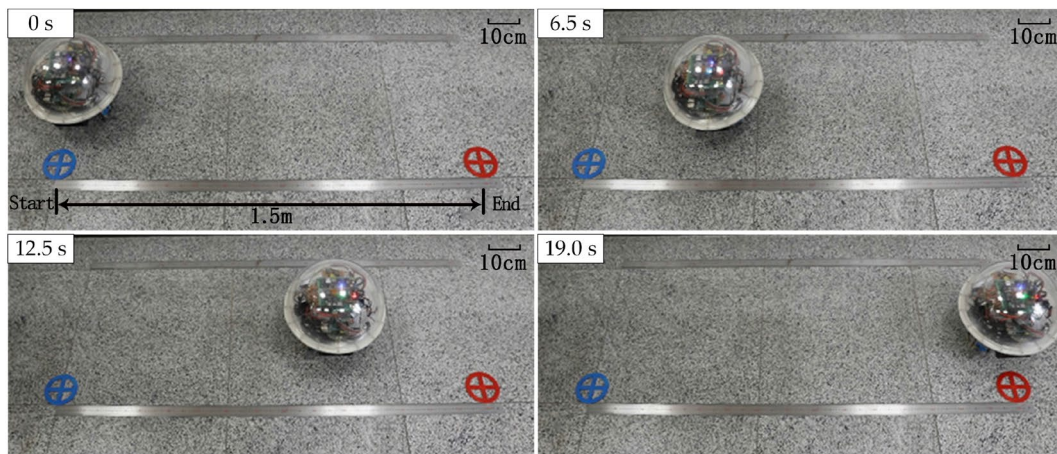
land test, we evaluated three gaits and eleven speed grades. Figure 13f shows the walking speed, which we measured with a ruler and a timer. This shows that the robot is able to walk at speeds up to 8.5 cm/s (0.34 body length per second). The fast walking gait is, on average, 12.1% faster than the standard walking gait and the stable walking gait is an average of 14.3% slower than the standard gait. In the underwater test, the robot was configured to swim in a straight line in a $2.8 \times 2.0 \times 1.0$ m tank. We measured the speed of the robot as it was driven by two or four water-jet motors. Figure 13g shows the underwater speed of the robot, which was again measured with a ruler and timer. The robot is able to swim at speeds of up to 16.1 cm/s (0.64 body length per second) when the water-jet motors work at the maximum speed grade. As shown in Fig. 14, when the robot walks on a relatively flat road, it can reduce the height of the lifting platform, use the omni-directional wheel to support the body, and the legs slide the ground to generate the initial speed. Using the inertial motion of the robot, this process greatly increases the mobile efficiency and endurance of the robot. The robot can glide with an average speed of 16.7 cm/s.

(2) In the slope climbing evaluation phase, the robot was configured to climb up a slope with a gradient of five degrees. We installed an NDI Polaris Vicra, a real-time three-dimensional measurement system with a precision of 0.5 mm, at the top of the slope to monitor the displacement of the robot. We fixed three magnetic markers onto the surface of the robot to determine its spatial orientation, as shown in Fig. 15a. The height of the slope did not include in the vertical displacement data. So, we just consider the vibration amplitude and vertical displacement difference between start point and end point. Figure 15b, c show the vertical displacement of the robot as it tries to climb up the slope. Without gait adjustment, the fluctuations of the robot's body have amplitudes of nearly 20 mm, and the robot slipped frequently, as shown in Fig. 15b. After adding the gait adjustment mechanism, the amplitudes of the fluctuations decreased to less than 10 mm and the robot climbed the slope steadily, as shown in Fig. 15c. This shows that the gait adjustment mechanism meets the stability and mobility requirements.

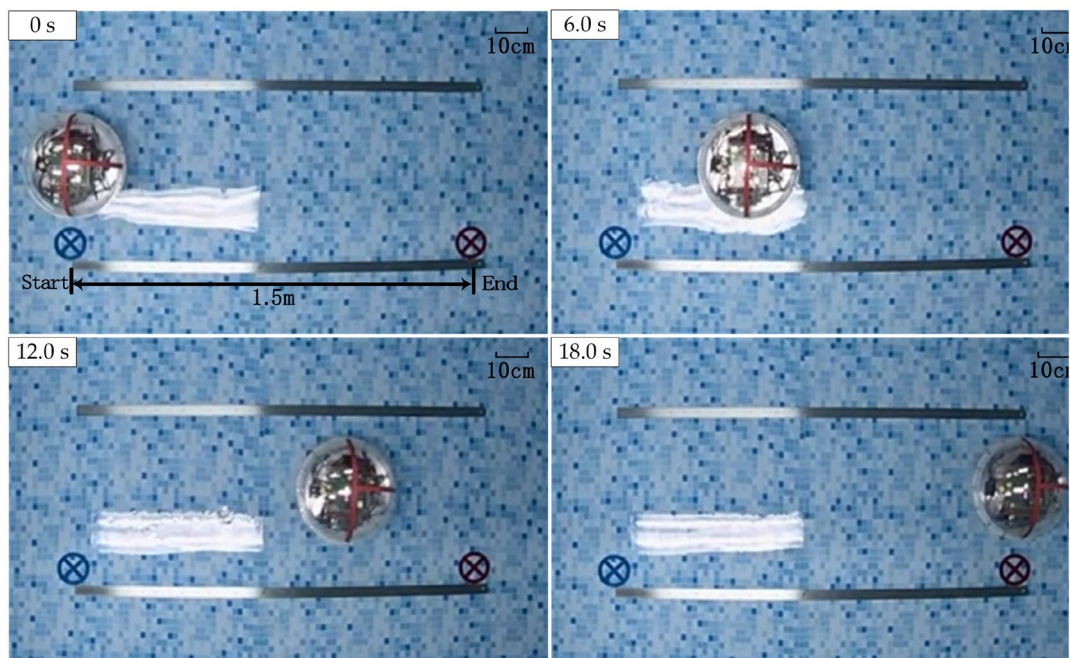
(3) In the underwater course control test phase, the robot was configured to move in a straight line, then make a turn. The heading of the robot was controlled by the generalized prediction controller. We measured the attitude and motion parameters of the robot with a MEMS IMU, which consisted of a gyroscope (ITG-3200), an accelerometer (ADXL345), and an electronic compass (HMC5883L). We used the quaternion algorithm with the Kalman filtering mechanism to calculate the angle



(a)



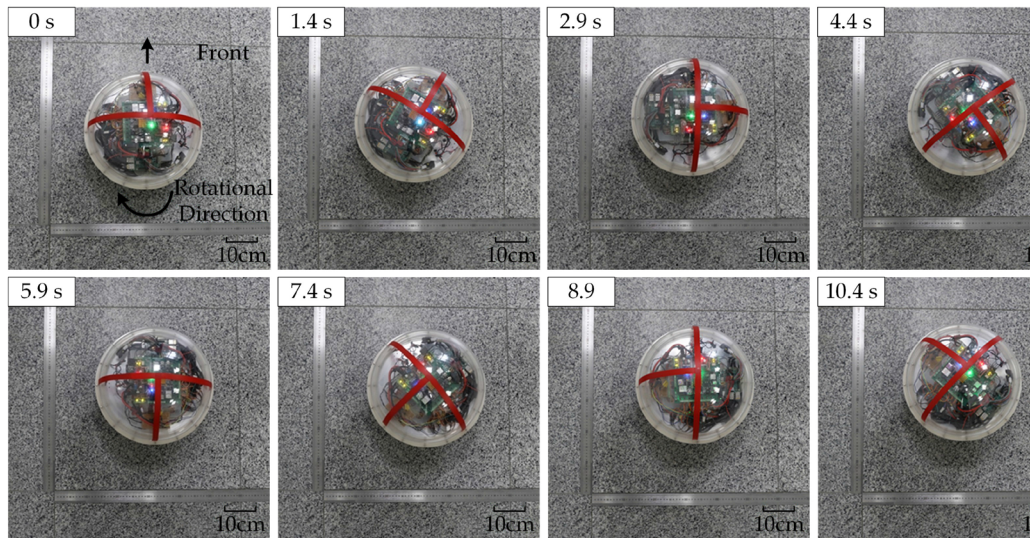
(b)



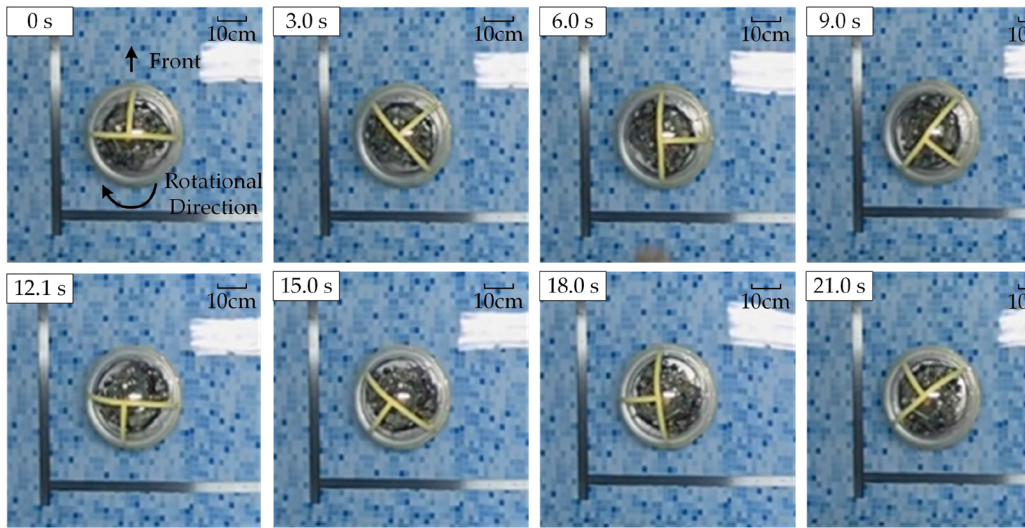
(c)

Fig. 13 Results from the evaluation of the robot’s motion performance. **a** Pictures of the prototype of the amphibious spherical robot; **b** Pictures of the translational motion test on land; **c** Pictures of the rotational motion test on land; **d** Pictures of the translational motion test in water; **e** Pictures

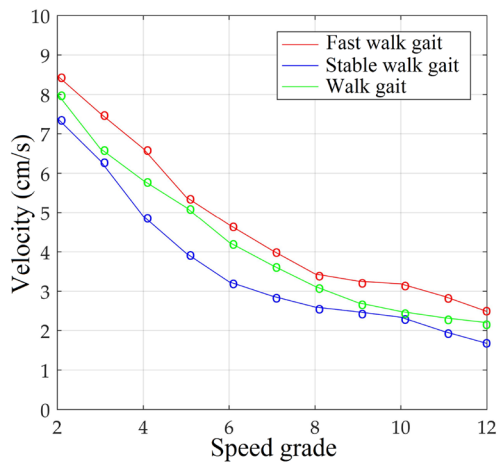
of the rotational motion test in water; **f** Speed of the amphibious spherical robot as it walks with different gaits on land; and **g** Speed of the amphibious spherical robot driven by two and four water-jets in water



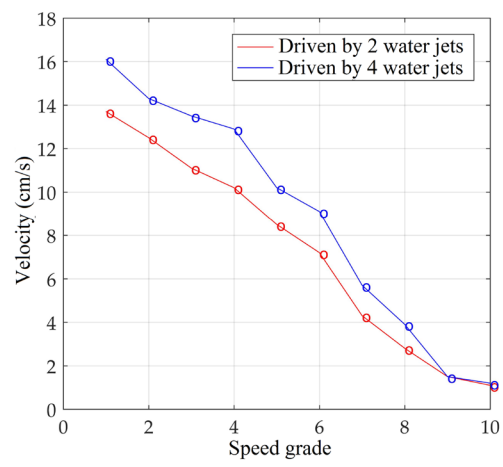
(d)



(e)



(f)



(g)

Fig. 13 (continued)

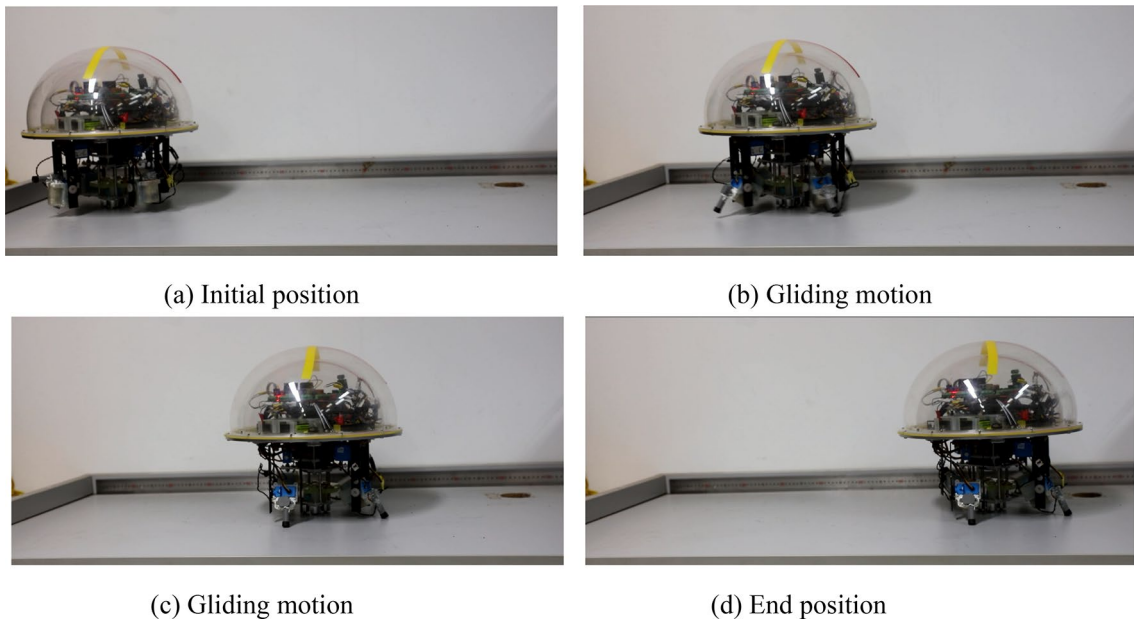


Fig. 14 Passive gliding experiment

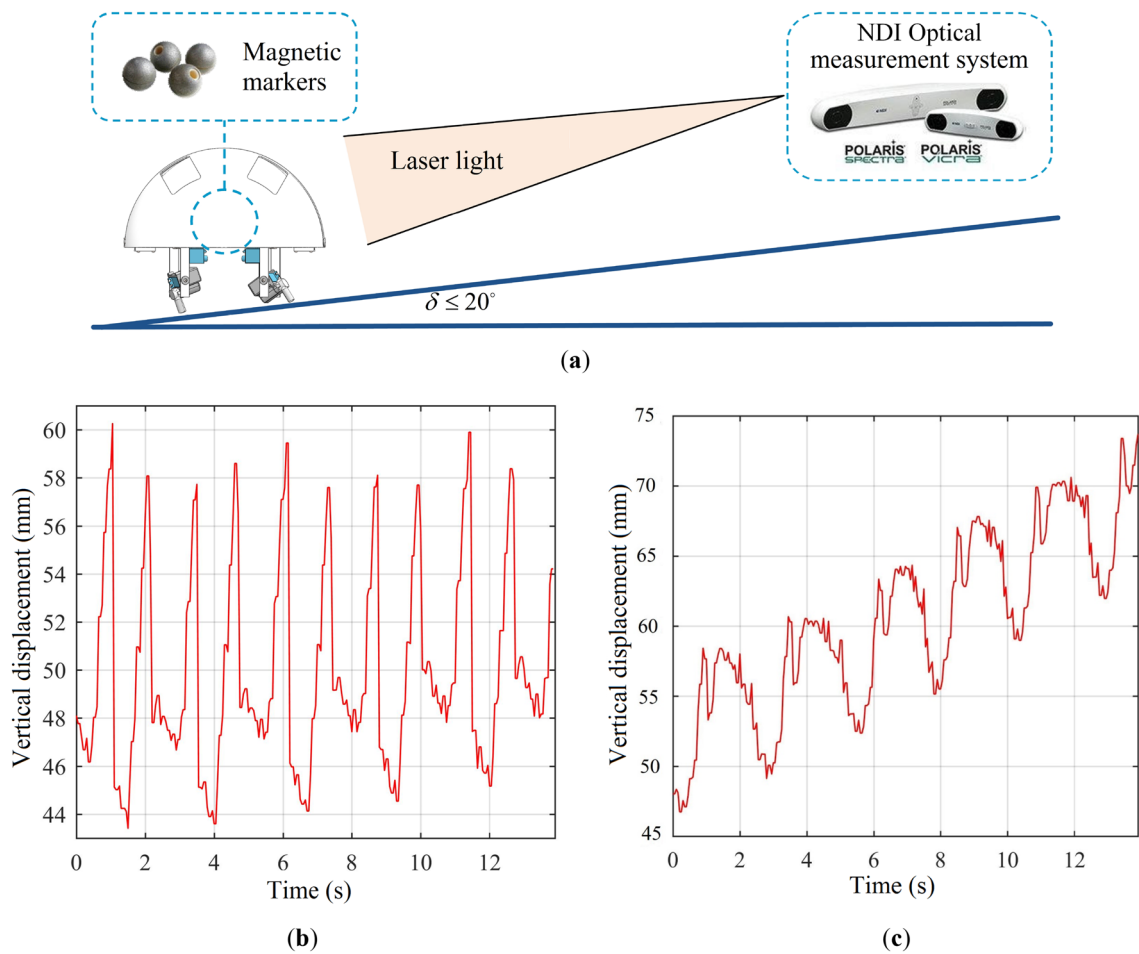


Fig. 15 Evaluation of the robot's slope climbing ability. **a** Principle used to test the slope climbing ability; **b** vertical displacement of the robot without gait adjustment; and **c** vertical displacement of the robot with gait adjustment

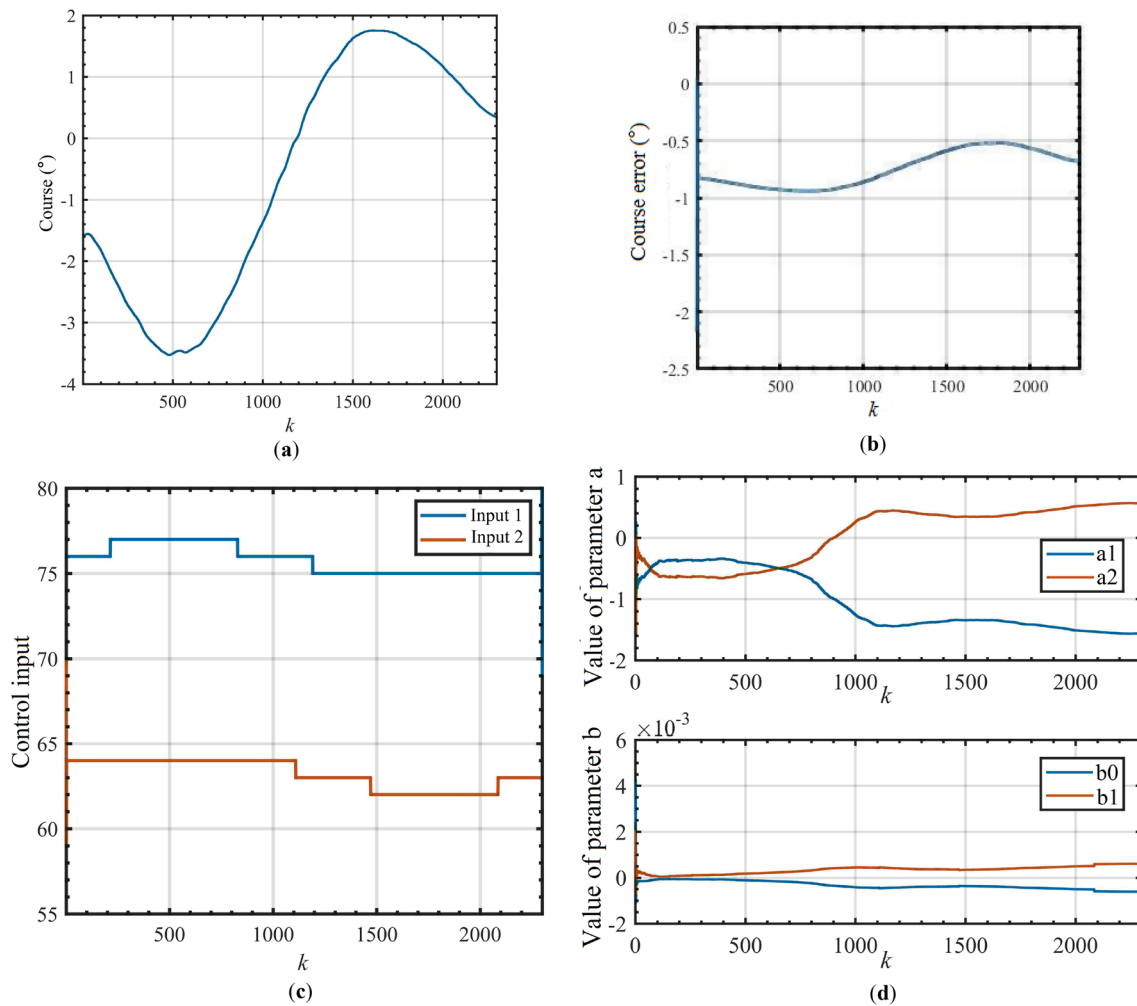


Fig. 16 Results of the underwater rectilinear motion test. **a** The robot's course; **b** course control error **c** control input to the generalized prediction control system; and **d** parameters of the calculated system transfer function

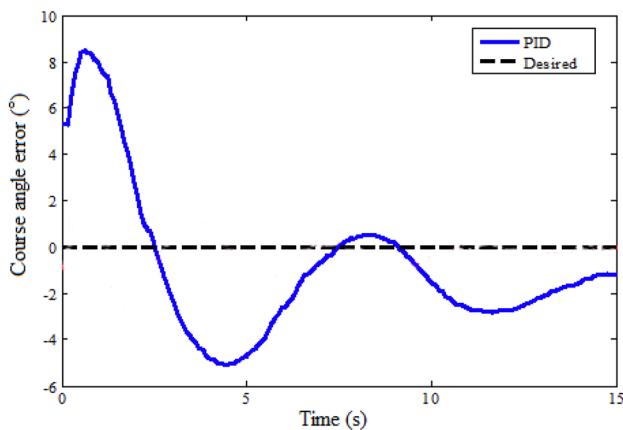


Fig. 17 The robot's course by PID controller

of the robot's course, which acted as the feedback signal to the controller. The angle of the robot's course was set to 0° in the rectilinear motion test. We used a program to record the control parameters, as shown in Fig. 16. The error between the robot's heading and the expected value was less than $\pm 3^\circ$, as shown in Fig. 16a. The parameters of the system transfer function change gradually as the robot's motion status changes, then stabilize as the robot enters a state of uniform motion, as shown in Fig. 16d. This indicates that the generalized prediction control algorithm is suitable for controlling systems that vary slowly with time. In contrast, the classical PID controller, which relies on a static system transfer function, failed to stabilize the robot's heading. The course error increased gradually and the system did not eventually converge. When the robot moves forward in a straight line, the maximum error of the PID control algorithm is approximately 8.45° , as shown in Fig. 17.

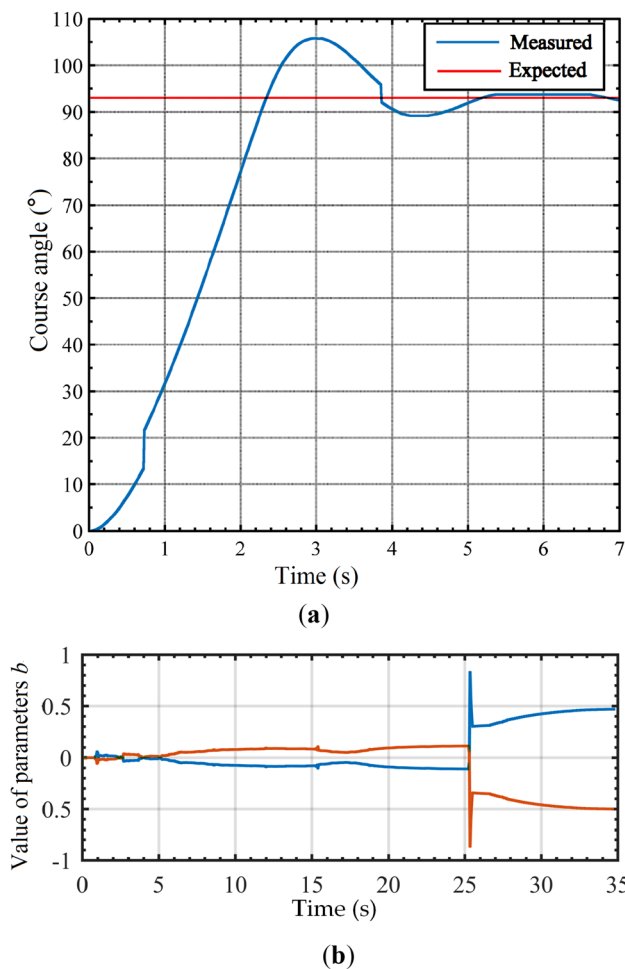


Fig. 18 Results of the underwater turning motion test. **a** The robot's course; and **b** parameters of the calculated system transfer function

In the turning motion test, the angle of the robot's course was first set to 0° , and then adjusted to 90° . Figure 18 shows the control parameters recorded during the turning motion test. The course error was less than ± 3 degrees and the turning operation lasted for 6 s. This demonstrates that our algorithm can control the robot dynamically, as shown in Fig. 18a. The parameters of the system transfer function change continually during the turning process, as shown in Fig. 18b.

6 Conclusions

In this study, we designed and built an improved version of our previously reported amphibious spherical robot with the aim of making the robot capable of completing exploration tasks in varied amphibious environments. We then designed an adaptive motion control system with 4-DoFs. First, we

developed simplified mechanical and dynamic models so that we could analyze the motions of the robot on land and in water, respectively. We designed a set of walking gaits so that the robot could move flexibly on various terrains. These gaits provided different stabilities and speeds. We then developed an online gait adjustment mechanism to keep the robot stable as it climbs up slopes with small gradients. As the underwater motion of the robot is very susceptible to environmental variations, we implemented an online system recognition with a forgetting factor least squares algorithm. The underwater motion was controlled by a generalized prediction control algorithm. The proposed motion control system was implemented on digital circuits and software. We used a MEMS IMU to measure the status of the robot's motions and provide feedback signals to the controller. We evaluated the performance of the robot and the proposed motion control system experimentally. The adaptive robotic control system enhanced the stability and mobility of the amphibious spherical robot, such that they meet the requirements for applications in complex amphibious environments.

Acknowledgements This work was supported by National Natural Science Foundation of China (61773064, 61503028). Xianqiang Bao, Xui Xiao, Zhan Chen, Yanlin He, and Xiaojuan Cai also contributed to the fabrication work of this amphibious spherical robot.

Data availability All data generated or analyzed during this study are included in this published article.

Declarations

Conflict of interest We declare that there is no conflict of interest pertaining to this publication.

References

- Shi, L.W., Pan, S.W., Guo, S.X., Tang, K., Guo, P., Xiao, R., & He, Y.L. (2016). Design and evaluation of quadruped gaits for amphibious spherical robots. In *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 13–18). IEEE
- Wu, Z. X., Yu, J. Z., Yuan, J., & Tan, M. (2019). Towards a gliding robotic dolphin: Design, modeling, and experiments. *IEEE/ASME Transactions on Mechatronics*, *24*(1), 260–270.
- Wang, W., Dai, X., Li, L., Gheneti, B. H., Ding, Y., Yu, J. Z., & Xie, G. M. (2018). Three-dimensional modeling of a fin-actuated robotic fish with multimodal swimming. *IEEE/ASME Transactions on Mechatronics*, *23*(4), 1641–1652.
- Shi, Q., Ishii, H., Sugahara, Y., Takahashi, A., Huang, Q., & Fukuda, T. (2014). Design and control of a biomimetic robotic rat for interaction with laboratory rats. *IEEE/ASME Transactions on Mechatronics*, *20*(4), 1832–1842.
- Shi, Q., Gao, Z. H., Jia, G. L., & Li, C. (2021). Implementing rat-like motion for a small-sized biomimetic robot based on extraction of key movement joints. *IEEE Transactions on Robotics*, *37*(3), 747–762.
- Shi, Q., Gao, J. H., Wang, S. J., Quan, X. L., Jia, G. L., Huang, Q., & Fukuda, T. (2022). Development of a small-sized quadruped

- robotic rat capable of multimodal motions. *IEEE Transactions on Robotics*. <https://doi.org/10.1109/TRO.2022.3159188>
7. Li, W. B., Zhang, W. M., Zou, H. X., Peng, Z. K., & Meng, G. (2018). A fast rolling soft robot driven by dielectric elastomer. *IEEE/ASME Transactions on Mechatronics*, 23(4), 1630–1640.
 8. Guo, X., Ma, S. G., Li, B., & Fang, Y. C. (2018). A novel serpentine gait generation method for snakelike robots based on geometry mechanics. *IEEE/ASME Transactions on Mechatronics*, 23(3), 1249–1258.
 9. Shi, Q., Li, C., Li, K., Huang, Q., Ishii, H., Takanishi, A., & Fukuda, T. (2018). A modified robotic rat to study rat-like pitch and yaw movements. *IEEE/ASME Transactions on Mechatronics*, 23(5), 2448–2458.
 10. Tanaka, M., Nakajima, M., Suzuki, Y., & Tanaka, K. (2018). Development and control of articulated mobile robot for climbing steep stairs. *IEEE/ASME Transactions on Mechatronics*, 23(2), 531–541.
 11. Cui, L., Cheong, P., Adams, R., & Johnson, T. (2014). AmBot: A bio-inspired amphibious robot for monitoring the swan-canning estuary system. *Journal of Mechanical Design*, 136(11).
 12. Vogel, A. R., Kaipa, K. N., Krummel, G. M., Bruck, H. A., & Gupta, S. K. (2014). Design of a compliance assisted quadrupedal amphibious robot. In *2014 IEEE International Conference on Robotics and Automation (ICRA)* (pp. 2378–2383). IEEE.
 13. Zhong, B., Zhou, Y. C., Li, X. X., Xu, M., & Zhang, S. W. (2016). Locomotion performance of the amphibious robot on various terrains and underwater with flexible flipper legs. *Journal of Bionic Engineering*, 13(4), 525–536.
 14. Kaznov, V., & Seeman, M. (2010). Outdoor navigation with a spherical amphibious robot. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems* (pp. 5113–5118). IEEE.
 15. Jia, L. C., Hu, Z. Q., Geng, L. G., Yang, Y., & Wang, C. (2016). The concept design of a mobile amphibious spherical robot for underwater operation. In *2016 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)* (pp. 411–415). IEEE.
 16. Guo, S. X., Mao, S. L., Shi, L. W., & Li, M. X. (2012). Development of an amphibious mother spherical robot used as the carrier for underwater microrobots. In *2012 ICME International Conference on Complex Medical Engineering (CME)* (pp. 758–762). IEEE.
 17. Guo, S. X., He, Y. L., Shi, L. W., Pan, S. W., Xiao, R., Tang, K., & Guo, P. (2018). Modeling and experimental evaluation of an improved amphibious robot with compact structure. *Robotics and Computer-Integrated Manufacturing*, 51, 37–52.
 18. Guo, S. X., Pan, S. W., Li, X. Q., Shi, L. W., Zhang, P. Y., Guo, P., & He, Y. L. (2017). A system on chip-based real-time tracking system for amphibious spherical robots. *International Journal of Advanced Robotic Systems*, 14(4), 1729881417716559.
 19. Guo, S. X., Pan, S. W., Shi, L. W., Guo, P., He, Y. L., & Tang, K. (2017). Visual detection and tracking system for an amphibious spherical robot. *Sensors*, 17(4), 1–21.
 20. Yi, Y., Geng, Z., Jianqing, Z., Siyuan, C., & Mengyin, F. (2015). Design, modeling and control of a novel amphibious robot with dual-swing-legs propulsion mechanism. In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (pp. 559–566). IEEE.
 21. Yue, M., Liu, B. Y., An, C., & Sun, X. J. (2014). Extended state observer-based adaptive hierarchical sliding mode control for longitudinal movement of a spherical robot. *Nonlinear Dynamics*, 78(2), 1233–1244.
 22. Karavaev, Y. L., & Kilin, A. A. (2015). The dynamics and control of a spherical robot with an internal omnivheel platform. *Regular and Chaotic Dynamics*, 20(2), 134–152.
 23. Guo, J., Guo, S. X., & Li, L. G. (2017). Design and characteristic evaluation of a novel amphibious spherical robot. *Microsystem Technologies*, 23(6), 1999–2012.
 24. Li, M. X., Guo, S. X., Hirata, H., & Ishihara, H. (2017). A roller-skating/walking mode-based amphibious robot. *Robotics and Computer-Integrated Manufacturing*, 44, 17–29.
 25. Pan, S. W., Shi, L. W., & Guo, S. X. (2015). A kinect-based real-time compressive tracking prototype system for amphibious spherical robots. *Sensors*, 15(4), 8232–8252.
 26. Guo, S. X., He, Y. L., Shi, L. W., Pan, S. W., Tang, K., Xiao, R., & Guo, P. (2017). Modal and fatigue analysis of critical components of an amphibious spherical robot. *Microsystem Technologies*, 23(6), 2233–2247.
 27. He, Y. L., Shi, L. W., Guo, S. X., Pan, S. W., & Wang, Z. (2016). Preliminary mechanical analysis of an improved amphibious spherical father robot. *Microsystem Technologies*, 22(8), 2051–2066.
 28. Guo, J., Wu, G. Q., & Guo, S. X. (2015). Fuzzy PID algorithm-based motion control for the spherical amphibious robot. In *2015 IEEE International Conference on Mechatronics and Automation (ICMA)* (pp. 1583–1588). IEEE.
 29. Wang, Z., Guo, S. X., Shi, L. W., Pan, S. W., & He, Y. L. (2014). The application of PID control in motion control of the spherical amphibious robot. In *2014 IEEE International Conference on Mechatronics and Automation* (pp. 1901–1906). IEEE.
 30. An, L., Heng, W., & YongZheng, L. (2013). Gait transition of quadruped robot using rhythm control and stability analysis. In *2013 IEEE International Conference on Robotics and Biomimetics (ROBIO)* (pp. 2535–2539). IEEE.
 31. Li, M. X., Guo, S. X., Hirata, H., & Ishihara, H. (2015). Design and performance evaluation of an amphibious spherical robot. *Robotics and Autonomous Systems*, 64, 21–34.
 32. Pan, S. W., Shi, L. W., Guo, S. X., Guo, P., He, Y. L., & Xiao, R. (2015). A low-power SoC-based moving target detection system for amphibious spherical robots. In *2015 IEEE International Conference on Mechatronics and Automation (ICMA)* (pp. 1116–1121). IEEE.
 33. Prasad, B., Agrawal, A., Viswanathan, V., Chowdhury, A. R., Kumar, R., & Panda, S. K. (2015). A visually guided spherical underwater robot. In *2015 IEEE Underwater Technology (UT)* (pp. 1–6). IEEE.
 34. Hernández-Alvarado, R., García-Valdovinos, L. G., Salgado-Jiménez, T., Gómez-Espinosa, A., & Fonseca-Navarro, F. (2016). Neural network-based self-tuning PID control for underwater vehicles. *Sensors*, 16(9), 1429.
 35. Shi, Q., Yang, Z., Guo, Y. N., Wang, H. P., Sun, L. N., Huang, Q., & Fukuda, T. (2017). A vision-based automated manipulation system for the pick-up of carbon nanotubes. *IEEE/ASME Transactions on Mechatronics*, 22(2), 845–854.
 36. Shi, Q., Li, C., Wang, C. B., Luo, H. B., Huang, Q., & Fukuda, T. (2017). Design and implementation of an omnidirectional vision system for robot perception. *Mechatronics*, 41, 58–66.
 37. Krug, R., & Dimitrov, D. (2015). Model predictive motion control based on generalized dynamical movement primitives. *Journal of Intelligent & Robotic Systems*, 77(1), 17–35.
 38. Tran, Q. N., Özkan, L., & Backx, A. (2015). Generalized predictive control tuning by controller matching. *Journal of Process Control*, 25, 1–18.
 39. Xing, H. M., Shi, L. W., Tang, K., Guo, S. X., Hou, X. H., Liu, Y., & Hu, Y. (2019). Robust RGB-D camera and IMU fusion-based cooperative and relative close-range localization for multiple turtle-inspired amphibious spherical robots. *Journal of Bionic Engineering*, 16(3), 442–454.
 40. Shi, L. W., Hu, Y., Su, S. X., Guo, S. X., Xing, H. M., Hou, X. H., & Xia, D. B. (2020). A fuzzy PID algorithm for a novel miniature spherical robots with three-dimensional underwater motion control. *Journal of Bionic Engineering*, 17(5), 959–969.

41. Xing, H. M., Guo, S. X., Shi, L. W., He, Y. L., Su, S. X., Chen, Z., & Hou, X. H. (2018). Hybrid locomotion evaluation for a novel amphibious spherical robot. *Applied Sciences*, 8(2), 156.
42. Shi, L. W., Guo, S. X., Mao, S. L., Yue, C. F., Li, M. X., & Asaka, K. (2013). Development of an amphibious turtle-inspired spherical mother robot. *Journal of Bionic Engineering*, 10(4), 446–455.
43. Hou, X. H., Guo, S. X., Shi, L. W., Xing, H. M., Liu, Y., Liu, H. K., & Li, Z. (2019). Hydrodynamic analysis-based modeling and experimental verification of a new water-jet thruster for an amphibious spherical robot. *Sensors*, 19(2), 259.
44. Xing, H. M., Shi, L. W., Hou, X. H., Liu, Y., Hu, Y., Xia, D. B., Li, Z., & Guo, S. X. (2021). Design, modeling and control of a miniature bio-inspired amphibious spherical robot. *Mechatronics*, 77(1), 102574.
45. Shi, L. W., Hu, Y., Su, S. X., Guo, S. X., Xing, H. M., Hou, X. H., Liu, Y., Chen, Z., Li, Z., & Xia, D. B. (2020). A fuzzy PID algorithm for a novel miniature spherical robots with three-dimensional underwater motion control. *Journal of Bionic Engineering*, 17(5), 959–969.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.