RBSAC: Rolling Balance Controller Based on Soft Actor-Critic Algorithm of the Unicycle Air Robot

Chunzheng Wang[1], Yunyi Zhang[1], Chenlong Zhang[1], Qixiang Zhao[1], and Wei Wang[1(✉)]

[1] Beihang University, Beijing, China
wangwei701@buaa.edu.cn

Abstract. Due to the complexity of the robot's dynamics model coupled with multiple outputs, most model-based or proportional-integral-derivative type controllers are unable to effectively solve the problem of attitude control of a Unicycle Air Robot (UAR) rolling in ground mode. In this paper, we formulate the attitude control problem in ground mode as a continuous-state, continuous-action Markov decision process with unknown transfer probabilities. Based on deterministic policy gradient theorems and neural network approximations, we propose RBSAC: a model-free rolling balance Reinforcement Learning (RL) controller based on Soft Actor-Critic (SAC) algorithm, which learns the state feedback controller from the attitude sampling of the ground mode. To improve the performance of the algorithm, we further integrate a batch learning method by playing back previously prioritized trajectories. We illustrate through simulations that our model-free approach RBSAC outperforms a feedback-based linear PID controller. After conducting simulation verification, it has been observed that the RBSAC controller enables stable rolling of the UAR on the ground, resulting in a 60% improvement in speed compared to the PID controller. Moreover, the RBSAC controller exhibits enhanced robustness and autonomous response to external disturbances during motion.

Keywords: Unicycle Air Robot, Reinforcement Learning, Attitude Control.

1 Introduction

Unicycle Air Robots (UARs) have been widely studied in recent years due to their higher energy efficiency compared to flight modes and higher trafficability in narrow spaces [1-5]. However, in ground mode, these robots' multiple power unit inputs generate control coupling, and their interaction with the ground forces is non-linearly complex, which poses great challenges to the attitude control.

The conventional Proportional-Integral-Derivative (PID) algorithm, renowned for its simplistic architecture, lucid principles, and straightforward implementation, has demonstrated commendable control outcomes when applied to address rudimentary control quandaries characterized by linearity and time-invariant attributes [6]. However, when the controlled system is a complex, non-linear system that is difficult to

linearise and establish an accurate mathematical model, linear PID control is often unable to achieve satisfactory control results. Furthermore, in the case of controlled systems characterized by numerous state variables, a multitude of actuators, and elevated task complexity, exemplified by the UAR system proposed in the seminal work by paper [7], the necessity arises to concurrently regulate multiple channels while considering the existence of inter-channel coupling. This intricate scenario entails the utilization of multiple PID controllers and entails a substantial number of parameters to be calibrated, rendering the task of identifying the optimal parameter combination arduous.

In the past few years, the field of Reinforcement Learning (RL) has witnessed noteworthy accomplishments in various domains, encompassing diverse applications ranging from high-level path planning and navigation [8, 9] to fine-grained attitude control [10-12]. RL emerges as a particularly promising technique for multi-rotor robots, offering valuable insights in mitigating the intricate nonlinear airflow ramifications stemming from rotor blades [13]. Additionally, RL proves advantageous in operating within uncertain environments, characterized by variables like wind patterns and obstacles. Notably, the model-free paradigm within RL holds a distinct edge, enabling agents to interact with intricate environments devoid of explicit dynamics models [14].

Wang. C. etc. [7] use the conventional method for UAR attitude control, which is the conventional proportional-integral link and controller based on model decomposition. The work in this paper is a direct continuation of that paper. We design RBSAC: a model-free rolling balance reinforcement learning controller based on Soft Actor-Critic algorithm [15], for the stable control of the ground mode rolling attitude of the UAR proposed in the literature. And we verify the excellent performance of RBSAC controller in a simulation environment.

After designing the network, we built a simulation environment based on the actual physical model parameters, sampled the UAR attitude, and trained it to avoid the possible hazards of physical UAR experiments and the problem of long and time-consuming data acquisition. To improve data efficiency, we proposed a batch learning scheme by playing back previous experiences [16].

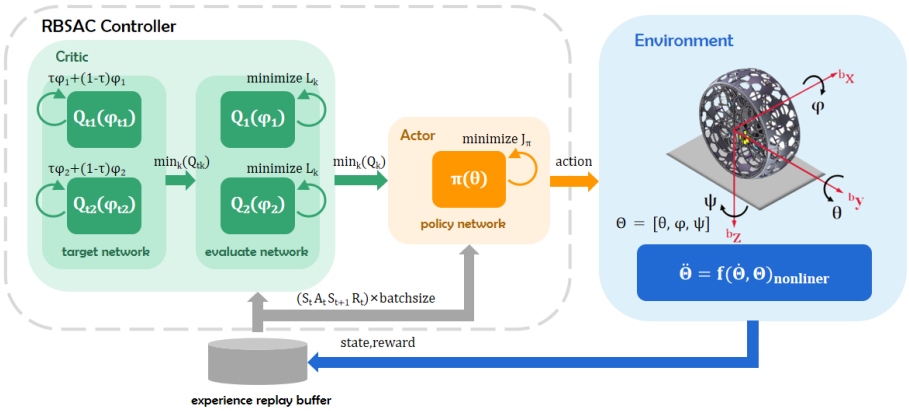

Fig. 1. The altitude control task of UAR used RBSAC controller.

The main contributions of this paper are summarized as follows:

1) Pioneeringly, we formulate the three attitude angle control problems for UAR as a Markov Decision Process (MDP), integrating meticulously crafted state and reward functions.

2) We build a realistic, testable RBSAC controller for a UAR simulation environment that can efficiently acquire observation space, state space, action space, and rewards at update rates of up to 100Hz.

3) We design an attitude controller of the UAR called RBSAC, which is based on the SAC algorithm that uses a circular experience buffer to store past experiences. The agent uses small batches of randomly sampled experiences from the buffer to update the actor and critic networks.

This paper is organized as follows: First, in Section 2, we decompose the defined UAR control problem, introduce a nonlinear dynamics model of the UAR, and transform it into a MDP; in Section 3, we propose RBSAC agent that interacts iteratively with the UAR's MDP to optimize the neural network attitude control strategy. Finally, in Section 4, we build a UAR simulation environment and evaluate the advantages of the RBSAC controller over conventional PID controllers for the UAR attitude tracking task.

2 Preliminary

2.1 Introduction of UAR

This paper innovatively integrates the multi-rotor structure with the unicycle structure and proposes a new UAR design that has the advantages of both multi-rotors and unicycle structures. The weight of the UAR is about 2.8kg, the diameter of the wheel is 460mm and the width is 170mm.

We design the inner ring of the UAR as the center of gravity eccentric structure, and install the rolling motor between the inner and outer ring, as shown in Fig. 2. The output torque of the rolling motor makes the inner ring rotate at a certain angle and overcome its gravity moment, and the reaction torque of the motor acts on the outer ring to drive it to roll. The inner ring of the system produces a torque that is directly correlated to the rotation angle. As a consequence, the reaction torque exerted on the outer ring is likewise contingent upon the rotation angle of the inner ring. Through the manipulation of the motor to regulate the magnitude of the rotation angle of the inner ring, it becomes possible to achieve control over the rolling speed of the outer ring. The rolling balance motion is schematically shown in Fig. 3.

In particular, the camera can be fixed to the inner ring, and the rotation angle of the inner ring can be kept stable by rolling motors, which means the camera can be kept facing forward to facilitate functional expansion, such as autonomous navigation and target detection, which is a better design than the robot in reference [3].

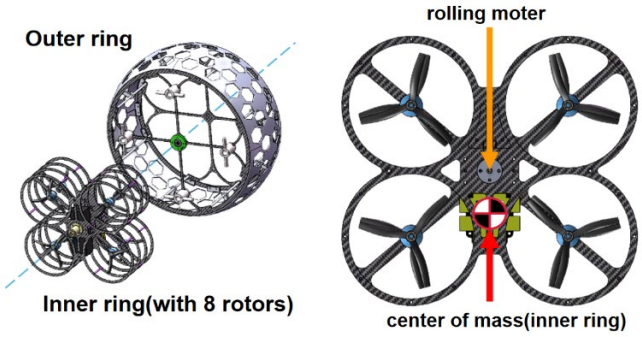

Fig. 2. Structure of the UAR

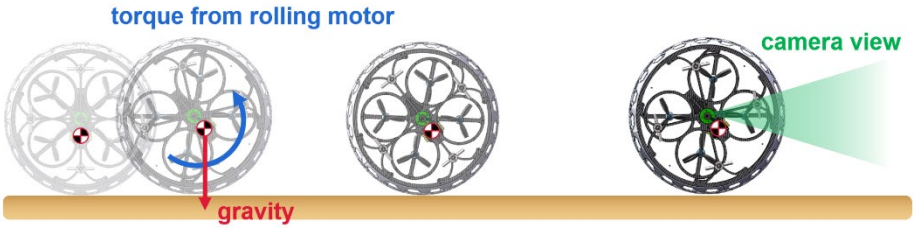

Fig. 3. Process of rolling balance motion

2.2 Task Definition

Since the control method of UAR in flight mode is similar to that of a conventional quadrotor, we will focus on the control in ground mode.

To characterize the ground motion, a coordinate system is established concerning the fixed inner circle. Euler angles are adopted to describe the attitude of the Unified Actuator-Retaining (UAR) system during the ground motion. The design of the rolling controller aims to ensure that the UAR's attitude aligns with the input command, with the expected Euler angles serving as the inputs to the controller.

2.3 Definition of Observation Space

The main onboard measurement elements of the UAR are an accelerometer, gyroscope, magnetometer, and barometer, while an encoder of the rolling motor is set to measure the angle and speed of outer rings for the control of rolling speed. According to the design goals of the controller, the desired Euler angles should also be used as the observation information provided to the agent. The defined observation space is shown in Table 1.

Table 1. Definition of the observation space

Observation space variables	Meaning
$\boldsymbol{\Theta} = \begin{pmatrix} \phi & \theta & \psi \end{pmatrix}^T$	3 measured Euler angles for the inner circle coordinate system
$\boldsymbol{\Theta}_d = \begin{pmatrix} \phi_d & \theta_d & \psi_d \end{pmatrix}^T$	3 desired Euler angles for the inner circle coordinate system
$\boldsymbol{\omega} = \begin{pmatrix} \omega_x & \omega_y & \omega_z \end{pmatrix}^T$	Tri-axial angular velocity of the inner ring
$\dot{\phi}_{\text{Outer}}$	The rotational speed of the outer ring relative to the ground around its central axis
$\mathbf{u}_{t-1}$	The intelligent body output action of the previous step

2.4 Definition of Action Space

The actuators of the UAR are the 8-rotor motors and the rolling motors between the inner and outer rings. Since there is a mapping relationship between the combined force and moment generated by the 8 rotors and the speed of each rotor represented by the power distribution matrix, it is sufficient to define the combined force and moment of the 8 rotors and the moment of the rolling motor in the action space.

Table 2. Definition of the action space

Action space variables	Meaning
F	Combined force generated by rotors
τ_x, τ_y, τ_z	Combined torque generated by rotors
T	Rolling motor torque

2.5 Definition of Reward Function

The reward function for the training process is designed according to the design goals of the controller to encourage the deviation of each Euler angle measurement from the desired value to be as small as possible, to penalize too fast rotation or vibration of the inner ring to prevent interference with the measurement elements attached to the inner ring and devices such as the camera and to penalize the agent for using too much control to obtain a reward. The reward function includes both continuous and sparse rewards and includes both reward and penalty terms. The expression of the reward function is shown below:

$$
\begin{aligned}
r_1 &= -(\varphi - \varphi_d)^2 - (\theta - \theta_d)^2 - (\psi - \psi_d)^2 \\
r_2 &= 0.25 \ (\text{if} \left|\varphi - \varphi_d\right| < 0.05) \\
r_3 &= 0.25 \ (\text{if} \left|\theta - \theta_d\right| < 0.05) \\
r_4 &= 0.25 \ (\text{if} \left|\varphi - \varphi_d\right| < 0.05) \\
r_5 &= 20 \ (\text{if} \left|\varphi - \varphi_d\right| < 0.05 \,\&\&\, \left|\theta - \theta_d\right| < 0.05 \,\&\&\, \left|\varphi - \varphi_d\right| < 0.05) \\
r_6 &= -\left\|\mathbf{u}_{t-1}\right\| \\
r_7 &= 250 \frac{T_s}{T_f} \\
r_8 &= -1000 \cdot \text{isdone} \\
R_t &= \sum_{i=1}^{8} r_i
\end{aligned}
\tag{1}
$$

The maximum training time of each round is denoted as T_f and the interval of each time step is denoted as T_f. Two parameters are set as follows:

$$T_f = 10\text{s}, \ T_s = 0.01\text{s}$$

"Isdone" means that if a control quantity deviates too much, it means that the strategy used in this round is poor and the training round needs to be terminated early. The conditions for triggering isdone are defined as:

1) Any angular deviation greater than 1 rad.

2) Inside circle instantaneous speed greater than 5 rad/s.

3 Design of the Algorithm

3.1 Algorithm Principle

The SAC algorithm is a model-free, online learning, heterogeneous strategy, reinforcement learning algorithm based on an actor-critic structure [17]. The SAC algorithm computes an optimal strategy that maximizes both the long-term desired reward and the policy entropy (policy entropy) [18]. Policy entropy is a measure of the uncertainty of a policy for a given state [19, 20]. The entropy of the policy distribution for a given state s is $\pi(a \mid s)$, the probability distribution of different actions selected by an intelligence in the action space *A*, which is defined as:

$$h(\pi(\cdot \mid s)) = -\int_A \pi(a \mid s) \ln(\pi(a \mid s)) da \tag{2}$$

Obviously, a higher entropy value promotes the agent to try more exploration in the action space. To the single-step reward R_{t+1} calculated from the reward function, add the term determined by the strategy entropy:

$$R_{t+1}^{\text{entropy}} = R_{t+1} + \alpha^{\text{entropy}} h(\pi(\cdot \mid s_t)) \tag{3}$$

And the state value function with entropy and action value function are derived according to the usual method. the SAC algorithm maintains a total of 5 deep neural networks:

Table 3. Meaning of Network Symbols

Network Symbols	Meaning
$\pi(A \mid S;\theta)$	The stochastic policy network containing the parameter θ is output as an policy (actor) to output action values in the action space according to a normal distribution.
$Q_k(S,A;\varphi_k), k=1,2$	Double Q-networks with the same structure containing parameters φ_1, φ_2 respectively, avoiding overestimation of the value function.
$Q_{tk}(S,A;\varphi_{tk}), k=1,2$	Double target networks Q_{t1} and Q_{t2} corresponding to Q_1 and Q_2 networks, respectively, improving the stability of the parameter updating process.

In the training, SAC performs the following operations:

1) Update the actor and critic attributes periodically during the learning process.

2) Estimate the mean and standard deviation of the Gaussian probability distribution of the continuous action space, and then randomly select actions based on this distribution.

3) Update the entropy weight term to balance the expected return.

4) Use a circular experience buffer to store past experiences, while the agent uses a random sampling of experiences from the buffer to update the networks.

Algorithm 1 Soft Actor-Critic Algorithm

Input: θ, φ_k and φ_{tk}

$\varphi_{tk} \leftarrow \varphi_k$, $D \leftarrow \varnothing$

for each iteration **do**

for each environment step do

$a_t \sim \pi_\theta(a_t \mid s_t)$

$s_{t+1} \sim p(s_{t+1} \mid s_t, a_t)$, p is the state transition probability.

$D \leftarrow D \cup \{(s_t, a_t, r(s_t, a_t), s_{t+1})\}$, D is the experience pool.

end for

for each gradient step **do**

$\varphi_k \leftarrow \varphi_k - \lambda_Q \hat{\nabla}_{\varphi_i} J_Q(\varphi_i)$ for $i \in \{1,2\}$, λ is learning rate, J is loss, and $\hat{\nabla}$ is gradient.

$\theta \leftarrow \theta - \lambda_\pi \hat{\nabla}_\theta J_\pi(\varphi)$

$\alpha \leftarrow \alpha - \lambda \hat{\nabla}_\alpha J(\alpha)$

$\varphi_{tk} \leftarrow \tau\varphi_k + (1-\tau)\varphi_{tk}$ for $i \in \{1,2\}$, τ is smoothing factor.

end for

end for

Output: θ, φ_k and φ_{tk}

4 Experiments and Results

4.1 Simulation Environment

In this paper, Matlab Simulink is selected as the simulation environment, and Simscape Multibody is used to build the mechanical structure of the UAR. Based on the physical test data, mathematical modeling is conducted for the input-output relationship of the UAR rotor motor, rolling motor and the aerodynamic characteristics of the propeller, and the physical interaction, sensor signals and control signals are simulated, and finally the 3D visualization simulation of the UAR motion on land and in the air is realized.

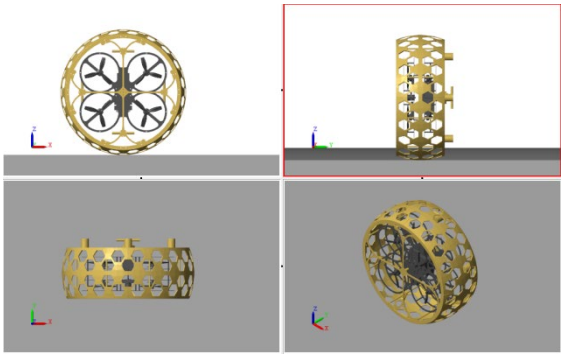

Fig. 4. 3D simulation interface

4.2 PID Controller Design

We partition the pitch and yaw angles into a single channel and assigned the roll angle to another separate channel. To achieve UAR attitude control, we develop two P-PID controllers. The architecture of these controllers is depicted below:

1) The pitch and yaw controller takes input from the measured pitch and yaw angles, as well as their angular velocities, obtained through an Inertial Measurement Unit (IMU). It outputs the desired torque in the two directions, ultimately determining the throttle for the rotor motors.

2) The pitch and yaw controller takes input from the measured pitch and yaw angles, as well as their angular velocities, obtained through an Inertial Measurement Unit (IMU). It outputs the desired torque in the two directions, ultimately determining the throttle for the rotor motors.

4.3 RBSAC Controller Design

The connection relationship between the layers of Critic and Actor networks with the number of nodes and the activation function is set as shown in Fig. 5. Critic networks fit the value of the action and Actor outputs the action according to the current state.

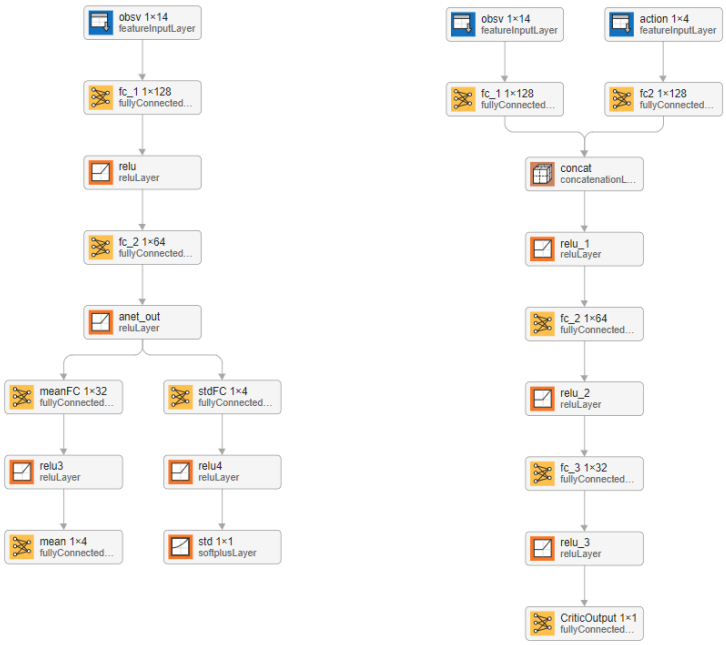

Fig. 5. Networks of RBSAC controller

4.4 Experimental Result

The following actions are designed to test the control effect of the trained reinforcement learning controller:

1) The initial state of the UAR is stationary and upright on the ground.

2) Set the desired rotation angle (i.e., roll angle) of the inner ring to -30°, and the outer ring of the UAR starts to roll on the ground under the action of the rolling motor torque.

3) After rolling for some time, the desired pitch angle of the inner ring is set to 30°, and the UAR is tilted at a certain angle by the aerodynamic force of the rotor and steadily tilted forward to roll.

4) During the tilt forward roll, a raised obstacle appears on the ground, and the UAR rolls over the obstacle while keeping the tilt angle (i.e. pitch angle) constant at 30°.

5) After the obstacle, the inner ring is set to expect the pitch angle to be 0°, and the UAR changes to an upright forward roll again.

6) The control force generated by the UAR through the rotor always keeps the yaw angle near 0° during the whole motion, i.e., it keeps the forward direction not affected by the pitch angle and obstacles, and the control torque generated by the motor always keeps the inner ring rotation angle near -30°.

The 3D visual motion simulation verification of the above motion process is shown in Fig. 6, and the video of the camera fixed on UAR is on the top left.

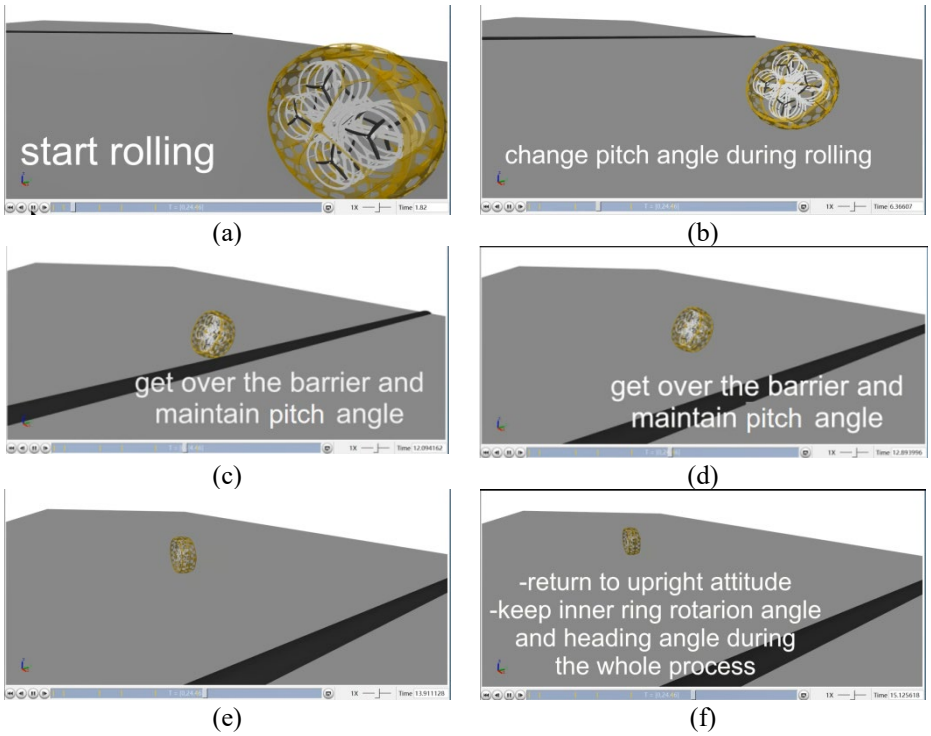

Fig. 6. Simulation of the designed motion

The variation curve of each attitude angle with time is shown in Fig. 7.

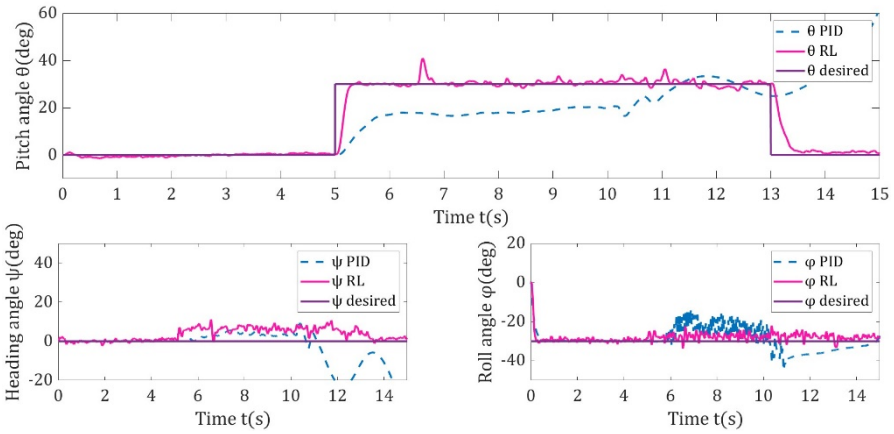

Fig. 7. The desired value and measured value of 3 Euler angles

The experiments demonstrate that the RBSAC controller can acquire an improved control strategy through a sufficient number of training iterations during fully autonomous exploration. Regarding pitch angle control, the RBSAC controller achieves

approximately 60% reduction in adjustment time compared to the PID controller, with nearly zero steady-state error, whereas the PID controller exhibits a significant steady-state error of approximately 70%. Concerning the inner ring rotation angle control, when simultaneous control of the pitch angle is required, the PID controller takes considerable oscillation, whereas the RBSAC controller consistently maintains minimal angle changes. Consequently, the designed RBSAC controller offers superior control over stable rolling and flexible attitude changes for UAR on the ground, effectively leveraging the advantages of its single-wheel structure.

In addition, when we train the network, the UAR rolls on flat terrain, but in the testing environment, roadblocks are set on the ground to verify the robustness of the controller. It was observed that at approximately 12 seconds, the UAR encountered a collision with the roadblock. As a result, the PID controller failed to maintain stability of the flywheel, leading to divergence in all Euler angles. However, it was noted that the RBSAC controller had a minimal impact on the movement of the UAR, thereby indicating the robustness of RBSAC controllers.

After extensive testing in the simulation environment, the trained policy network will be directly deployed on the physical platform in the future.

5 Conclusion

In this paper, we propose a RBSAC controller to keep the balance of UAR while rolling in ground mode. In order to simulate the motion of the UAR more realistically, this paper selects MATLAB/Simulink as the simulation environment, uses Simscape Multibody to build the mechanical structure of the UAR, and simulates physical interaction and sensor signals. The simulation is carried out to realize the 3D visualization simulation of the UAR motion on land and in the air.

This paper uses the Soft Actor-Critic algorithm as the basic principle of the controller and designs the controller for rolling motion according to the specific structure and task characteristics of the UAR, relying on the simulation platform to train the intelligent body a lot so that the agent learns the optimized control algorithm. After simulation verification, the controller designed in this paper can make the UAR roll stably on the ground and change its attitude flexibly, and at the same time, it has certain robustness and can cope with external interference to the motion autonomously, achieving the design goal.

References

1. Kawasaki, K., Zhao, M., Okada, K., Inaba, M.: MUWA: Multi-field universal wheel for air-land vehicle with quad variable-pitch propellers. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 1880-1885. IEEE, Tokyo, Japan (2013).
2. Fan, D.D., Thakker, R., Bartlett, T., et al.: Autonomous hybrid ground/aerial mobility in unknown environments. In: 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3070-3077. IEEE, Macau, China (2019).

3. Jia, H., et al.: A Quadrotor With a Passively Reconfigurable Airframe for Hybrid Terrestrial Locomotion. In: IEEE/ASME Transactions on Mechatronics, vol. 27, no. 6, pp. 4741-4751, IEEE. (2022).
4. Zhang, R., Wu, Y., Zhang, L., et al.: Autonomous and adaptive navigation for terrestrial-aerial bimodal vehicles. In: IEEE Robotics and Automation Letters 7(2), pp. 3008-3015 (2022).
5. Jia, H., Ding, R., Dong, K., Bai, S., Chirarattananon, P.: Quadrolltor: A Reconfigurable Quadrotor with Controlled Rolling and Turning. IEEE Robotics and Automation Letters, 1-8 (2023).
6. Borase, R.P., Maghade, D.K., Sondkar, S.Y. et al.: A review of PID control, tuning methods and applications. Int. J. Dynam. Control **9**, 818–827 (2021).
7. Wang, C., Zhang, Y., Li, C., Wang, W., Li, Y.: A Rotor Flywheel Robot: Land-Air Amphibious Design and Control. In: IEEE/RSJ International Conference on Intelligent Robots and System (2023).
8. Richter, D.J., Calix, R.A.: Using Double Deep Q-Learning to learn Attitude Control of Fixed-Wing Aircraft. In: 2022 16th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS), pp. 646-651, Dijon, France (2022).
9. Tong, G., Jiang, N., Biyue, L., Xi, Z., Ya, W., Wenbo, D.: Uav navigation in high dynamic environments: A deep reinforcement learning approach. Chinese Journal of Aeronautics 34(2), 479–489 (2021).
10. Hodge, V.J., Hawkins, R., Alexander, R.: Deep reinforcement learning for drone navigation using sensor data. Neural Computing and Applications **33**(6), 2015–2033 (2021).
11. Jiang, Z., Lynch A.F.: Quadrotor motion control using deep reinforcement learning. Journal of Unmanned Vehicle Systems **9**(4), 234–251 (2021).
12. Koch, W., Mancuso, R., West, R., Bestavros, A.: Reinforcement learning for uav attitude control. ACM Transactions on Cyber-Physical Systems **3**(2), 1–21 (2019).
13. Waslander, S.L., Hoffmann, G.M., Jang, J.S., Tomlin, C.J.: Multi-agent quadrotor testbed control design: Integral sliding mode vs. reinforcement learning. In: 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3712–3717. IEEE (2005).
14. Sun, Z., Wang, Z., Liu, J., Li, M., Chen, F.: Mixline: A Hybrid Reinforcement Learning Framework for Long-Horizon Bimanual Coffee Stirring Task. In: Intelligent Robotics and Applications. ICIRA 2022. Lecture Notes in Computer Science, vol 13455. Springer, Cham (2022).
15. Haarnoja, T., Zhou, A., Hartikainen, K., et al.: Soft actor-critic algorithms and applications. arXiv preprint arXiv:1812.05905 (2018).
16. Mysore, S., Mabsout, B., Mancuso, R., Saenko, K.: Regularizing Action Policies for Smooth Control with Reinforcement Learning. In: 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 1810-1816. IEEE, Xi'an, China (2021).
17. Choi, M., Filter, M., Alcedo, K., Walker, T.T., Rosenbluth, D., Ide, J.S.: Soft Actor-Critic with Inhibitory Networks for Retraining UAV Controllers Faster. In: 2022 International Conference on Unmanned Aircraft Systems, pp. 1561-1570. Dubrovnik, Croatia (2022).
18. He, L., Li, H.: Quadrotor Aerobatic Maneuver Attitude Controller based on Reinforcement Learning. In: 2022 13th Asian Control Conference, pp. 2450-2453. Jeju, Korea(2022).
19. Brunori, D., Colonnese, S., Cuomo, F., Iocchi, L.: A Reinforcement Learning Environment for Multi-Service UAV-enabled Wireless Systems. In: 2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops), pp. 251-256. Kassel, Germany (2021).
20. Liaq, M., Byun, Y.T.: Autonomous uav navigation using reinforcement learning. In: International Journal of Machine Learning and Computing, vol. 9, pp. 756–761 (2019).