Fuzzy Logic Theory-based Adaptive Reward Shaping for Robust Reinforcement Learning (FARS)

Hürkan Şahin, Van Huyen Dang, Erdi Sayar, Alper Yegenoglu, and Erdal Kayacan

***Abstract*—Reinforcement learning (RL) often struggles in real-world tasks with high-dimensional state spaces and long horizons, where sparse or fixed rewards severely slow down exploration and cause agents to get trapped in local optima. This paper presents a fuzzy-logic-based reward shaping method that integrates human intuition into RL reward design. By encoding expert knowledge into adaptive, interpretable terms, fuzzy rules promote stable learning and reduce sensitivity to hyperparameters. The proposed method leverages these properties to adapt reward contributions based on the agent's state, enabling smoother transitions between fast motion and precise control in challenging navigation tasks. The extensive simulation results on autonomous drone racing benchmarks show stable learning behavior and consistent task performance across scenarios of increasing difficulty. The proposed method achieves faster convergence and reduced performance variability across training seeds in more challenging environments, with success rates improving by up to approximately 5% compared to non-fuzzy reward formulations.**

***Index Terms*—Reinforcement Learning, Reward Shaping, Fuzzy Logic, Autonomous Navigation**

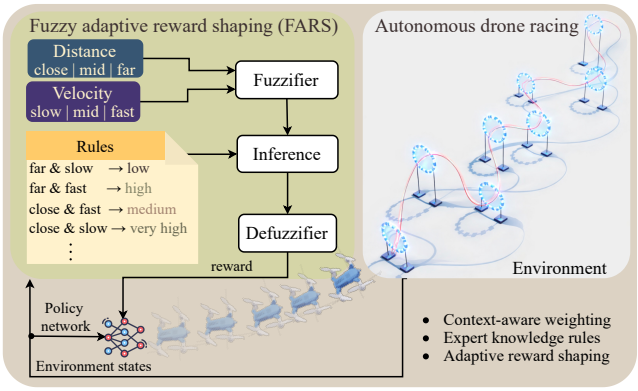


Fig. 1. Schematic overview of the fuzzy adaptive reward shaping (FARS) method. Whereas conventional RL design typically relies on manually tuned and crisp input terms, FARS introduces a fuzzy logic-based method that adaptively shapes reward as a function of distance and velocity using human-interpretable linguistic rules. The fuzzy reward is integrated into the RL objective to provide context-aware feedback, enabling faster convergence, reduced variance, and improved stability.

I. Introduction

Deep reinforcement learning (RL) has delivered strong performance across diverse domains by learning policies directly from high-dimensional sensory inputs, such as raw pixels or sensor data. Key examples include robotic manipulation [1], [2] and autonomous drone navigation [3], [4]. Despite these successes, RL performance in complex tasks remains highly sensitive to the design of the reward function, as sparse or poorly balanced rewards can hinder exploration and destabilize training. While reward shaping [5] offers denser feedback, conventional approaches rely on manually tuned reward combinations that are often brittle and sensitive to task variations.

Fuzzy logic has been widely adopted in unmanned aerial vehicles (UAVs) [6]–[11] to handle uncertainties and rapidly changing operating conditions. Recent studies have explored integrating fuzzy logic into RL frameworks to improve robustness. A control framework based on proximal policy optimization (PPO) [12] is introduced in [13]. The authors incorporate a Lyapunov-inspired fuzzy reward and an adjustable policy learning rate to improve training stability. A PPO-based actor–critic framework is presented in [14], where manually tuned classical rewards are replaced with a fuzzy logic–based reward formulation. This results in robust navigation performance in both static and dynamic environments. These works collectively indicate that fuzzy logic can provide structured, interpretable, and stabilizing reward signals for RL.

Autonomy in fast and agile flight has emerged as a powerful benchmark in robotics, as it simultaneously challenges perception, planning, and control. Furthermore, fast and agile flight is crucial for enabling autonomous drones to tackle time-critical real-world tasks; e.g. rapidly searching cluttered disaster zones for survivors. However, at high speeds with rapid rotations, nonlinear system dynamics are excited, providing a demanding testbed for advanced navigation methods [15]–[22]. As a challenging benchmark for reward design, autonomous drone racing requires an aerial robot to navigate at high speed through a sequence of gates under long-horizon and real-time constraints. Recent works have explored deep RL for drone racing, primarily focusing on generalization and robustness. An adaptive environment-shaping framework is proposed in [23], where progressively challenging training environments are generated by a secondary RL agent, enabling a single racing policy to generalize to unseen and dynamic tracks. A two-phase, curriculum-based RL framework is introduced in [24], in which aggressive flight and collision avoidance are balanced to achieve robust vision-based drone racing performance.

Recent work on fuzzy-logic–based RL has shown improved stability and interpretability through structured reward formulations. This work explores reward-level abstraction for autonomous drone racing by encoding human-inspired heuristics—such as maintaining high speed over long distances

*This work was partially supported by the Horizon Europe Grant Agreement No. 101136056 and No. 101119774.

Hürkan Şahin, Van Huyen Dang, Erdi Sayar, Alper Yegenoglu, and Erdal Kayacan are with the Automatic Control Group (RAT), Paderborn University, 33098 Paderborn, Germany {hursah, van.huyen.dang, erdi.sayar, alper.yegenoglu, erdal.kayacan}@upb.de

and decelerating near gates—directly into the reward function using fuzzy linguistic rules, as illustrated in Fig. 1. This approach yields a simple, interpretable reward formulation that generalizes across racing scenarios without task-specific reward engineering.

The main contributions of this paper are as follows:

- We propose an interpretable fuzzy-logic reward shaping method called FARS. It automatically balances fast movement when far from gates and slow, precise movement when close to gates. This results in smoother and more intelligent rewards than traditional non-fuzzy (crisp) shaping methods.
- We compare our proposed fuzzy logic-based reward shaping method using Mamdani and Sugeno defuzzification against a well-known potential field-based method [5] on drone racing tasks of increasing difficulty (Easy, Medium, and Hard zigzag courses). We analyze and discuss the main differences in smoothness, training stability, and final performance.
- We show that fuzzy rewards help the drone switch smoothly between fast flying and careful slowing in long and difficult drone racing tasks. By using velocity and distance in a natural way, fuzzy rewards lead to faster learning and higher success rates compared to non-fuzzy methods.

The remainder of this paper is organized as follows: Section I introduces the problem and reviews related work. Section II presents the proposed fuzzy logic–based reward shaping method. Section III describes the experimental setup, while Section IV reports and discusses the experimental results. Finally, Section V concludes the paper and outlines directions for future work.

II. Methodology

To demonstrate the effectiveness of the proposed fuzzy-based reward shaping method, we employ a deep RL framework to solve a drone racing task, as illustrated in Fig. 2.

This task is modeled as a Markov decision process defined by the tuple (S, A, P, r, γ), where S and A denote the state and action spaces, P is the transition matrix, r is the reward function, and $\gamma \in [0, 1)$ is the discount factor [25]. Accordingly, the objective is to learn a policy π_θ that maximizes the expected discounted return:

$$\max_{\theta} \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r_t\right]. \tag{1}$$

Since reward design plays a crucial role in learning efficiency, we next present the proposed fuzzy logic–based reward shaping approach and compare it with a commonly used potential-based reward shaping method [5].

A. Potential field-based reward shaping (PFBRS) for drone racing

In principle, a reward function is designed to guide the agent's learning process as it navigates a sequence of gates and hovers at the end point. We define a reward function, including multiple terms:

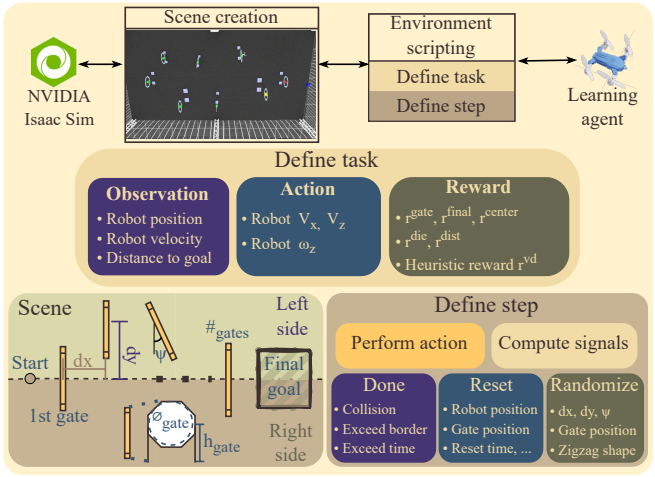


Fig. 2. Overview of our proposed training environment and the task design in IsaacLab [26]. The top row illustrates the task formulation, including observations, continuous actions, and the composite reward structure. The middle row shows the interaction between Isaac Sim–based scene generation, environment scripting, and the learning agent. The bottom row depicts the zigzag gate racing scenario with randomized gate positions and orientations, ending in a final goal region.

1) Gate-passed reward: The gate-passed reward r^{gate} increases cumulatively with the number of gates successfully passed, providing positive reinforcement for sequential progress.

$$r^{\text{gate}} = \begin{cases} c_1, & \text{if the agent passes a gate at time } t, \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

2) Final goal hovering reward: After the agent passes the final gate, the task switches to a goal-hovering phase. A dense distance-based reward encourages precise positioning at the final goal. Let p_t and p_{goal} denote the agent and goal positions, respectively. The reward is defined as

$$r^{\text{final}} = c_2 \exp\left(-\|p_t - p_{\text{goal}}\|_2\right) \Delta t \tag{3}$$

3) Distance-to-gate reward: A dense shaping reward is applied based on the Euclidean distance to the currently active gate (or the final goal after the last gate):

$$r^{\text{dist}} = \|p_t - p_{\text{goal}}\|_2 \, \Delta t \tag{4}$$

4) Gate alignment reward: The gate center alignment reward r^{center}, which encourages the agent to align with the gate center for easier gate passing.

$$r^{\text{center}} = \exp(-|\mathbf{n} \cdot \mathbf{v}_{\text{agent}}|) \tag{5}$$

where $|\mathbf{n} \cdot \mathbf{v}_{\text{agent}}|$ describes the alignment between the agent-to-gate moving direction $\mathbf{v}_{\text{agent}}$ and the direction of the normal vector crossing the gate center $\mathbf{n}$.

5) Collision penalty: The collision penalty r^{die} is a large negative reward applied if the drone collides with gates.

$$r^{\text{die}} = \begin{cases} -c_3, & \text{if a collision occurs at time } t, \\ 0, & \text{otherwise.} \end{cases} \tag{6}$$

We apply potential-based shaping to the terms r^{dist} and r^{center} to provide intermediate feedback and prevent training divergence. In general, the potential field function is constructed as

$$r_{\text{aux}} = \gamma \times r(s_t) - r(s_{t-1}) \tag{7}$$

where $r(s_{t-1})$ is the previous reward value, and $r(s_t)$ is the current value. Hence, the overall reward function with a potential-field-based approach is defined as

$$r_t = r^{\text{gate}} + r^{\text{final}} + r^{\text{dist}}_{\text{aux}} + r^{\text{center}}_{\text{aux}} + r^{\text{die}} \tag{8}$$

The remaining terms are retained unchanged, as they represent single-event rewards or penalties. The selection of hyperparameters, c_1, c_2, c_3, is commonly performed via trial and error and is highly reliant on the practitioner's experience with reward shaping.

B. Reward shaping enhancement with fuzzy-logic theory

A fuzzy-logic–based reward formulation is introduced to explicitly address the trade-off between approach speed and precision during gate traversal. Specifically, a velocity–distance reward r^{vd} is designed by defining intuitive linguistic rules over distance and velocity states, which enables smooth interpolation of the reward signal to promote the desired behaviours.

Within the fuzzy-based formulation, the classical distance-based auxiliary reward $r^{\text{dist}}_{\text{aux}}$ is replaced by the fuzzy velocity–distance reward, while all other reward components remain unchanged. Consequently, the total reward in the fuzzy-based approach is defined as follows:

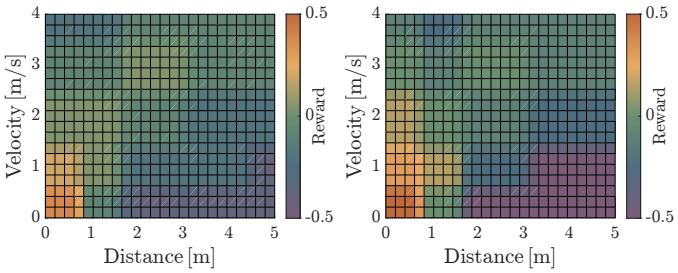
$$r_t = r^{\text{gate}} + r^{\text{final}} + r^{\text{vd}} + r^{\text{center}}_{\text{aux}} + r^{\text{die}}. \tag{9}$$

The fuzzy-logic-based reward depends on two inputs: the distance to the currently active gate and the magnitude of the drone's linear velocity. The velocity-distance reward is defined using two alternative formulations, each encoding the same underlying heuristic rather than introducing distinct behavioural objectives: (i) Mamdani fuzzy inference [27], (ii) Sugeno fuzzy inference [28], with rule-based surfaces shown in Fig. 3a and Fig. 3b, In all cases, the same distance and velocity normalization, membership functions, and reward logic are employed. The intuition behind the fuzzy logic is as follows:

$$\text{If } v \text{ is } A_i \text{ and } d \text{ is } B_i, \text{ then } z \text{ is } C_i,$$

for example: if *"the drone is far from the target gate"*, and *"it flies at a high velocity"*, then this behavior is encouraged to promote rapid progress. Similarly, if *"the drone approaches the gate"*, and *"its velocity is slow"*, then this behavior is also rewarded due to safe, stable, and centered traversal.

At comparable velocities, closer proximity to the gate yields a higher reward than maintaining a greater distance, thereby explicitly encouraging rapid convergence toward the gate rather than slow movement at greater distances. This design is essential to prevent conservative behaviours, such as premature speed reduction without meaningful progress. Fig. 3 illustrates how this principle is realised by the three velocity–distance reward formulations. Although the Mamdani and Sugeno variants differ in terms of smoothness and representation, both maintain the same qualitative structure. The Mamdani formulation produces a smooth, continuous reward surface, while the Sugeno formulation displays piecewise regions due to constant rule outputs.



(a) Mamdani r^{vd} (b) Sugeno r^{vd}

Fig. 3. Velocity–distance reward surfaces r^{vd} derived from (a) Mamdani and (b) Sugeno fuzzy inference systems. The surfaces provide a continuous representation of the fuzzy rule base, mapping distance–velocity inputs to reward outputs. Higher reward values correspond to higher velocities at larger distances and lower velocities near the target.

III. Experiments

The navigation policy is evaluated on a zigzag gate circuit designed to enforce alternating lateral maneuvers while maintaining forward motion, as illustrated in Fig. 2 scene section. Gates are arranged sequentially along the x-axis, with the first and last gates centered on the y-axis. Intermediate gates alternate strictly between left ($y > 0$) and right ($y < 0$) placements, forming a zigzag trajectory. Lateral offsets are sampled from $0.5\,\text{m},\ 0.75\,\text{m},\ 1.0\,\text{m},\ 1.25\,\text{m},\ 1.5\,\text{m},\ 1.75\,\text{m}$ with a uniform perturbation of up to 0.1m.

The Easy scenario focuses on basic lateral alternation and forward motion. The course consists of 6 gates with fixed height ($1.0\ \text{m}$) and diameter ($0.60\ \text{m}$). The longitudinal distance between consecutive gates is sampled from $1.5\ \text{m}, 1.75\ \text{m}, 2.0\ \text{m}$. Small yaw perturbations are applied to the gates, uniformly sampled from $[-10°, 10°]$. This configuration provides generous clearance and limited variability, serving as a baseline for navigation performance.

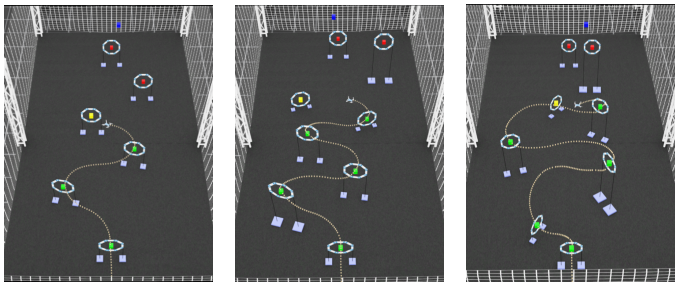


Fig. 4. Randomly generated zigzag environments in Isaac Sim for the Easy, Medium, and Hard scenarios (left to right). The environments are confined within predefined spatial boundaries. The dotted curve shows the drone trajectory. Green markers denote passed gates, the yellow marker indicates the active target gate, and red markers represent inactive gates. Each episode is limited to 15s and terminates immediately upon collision.

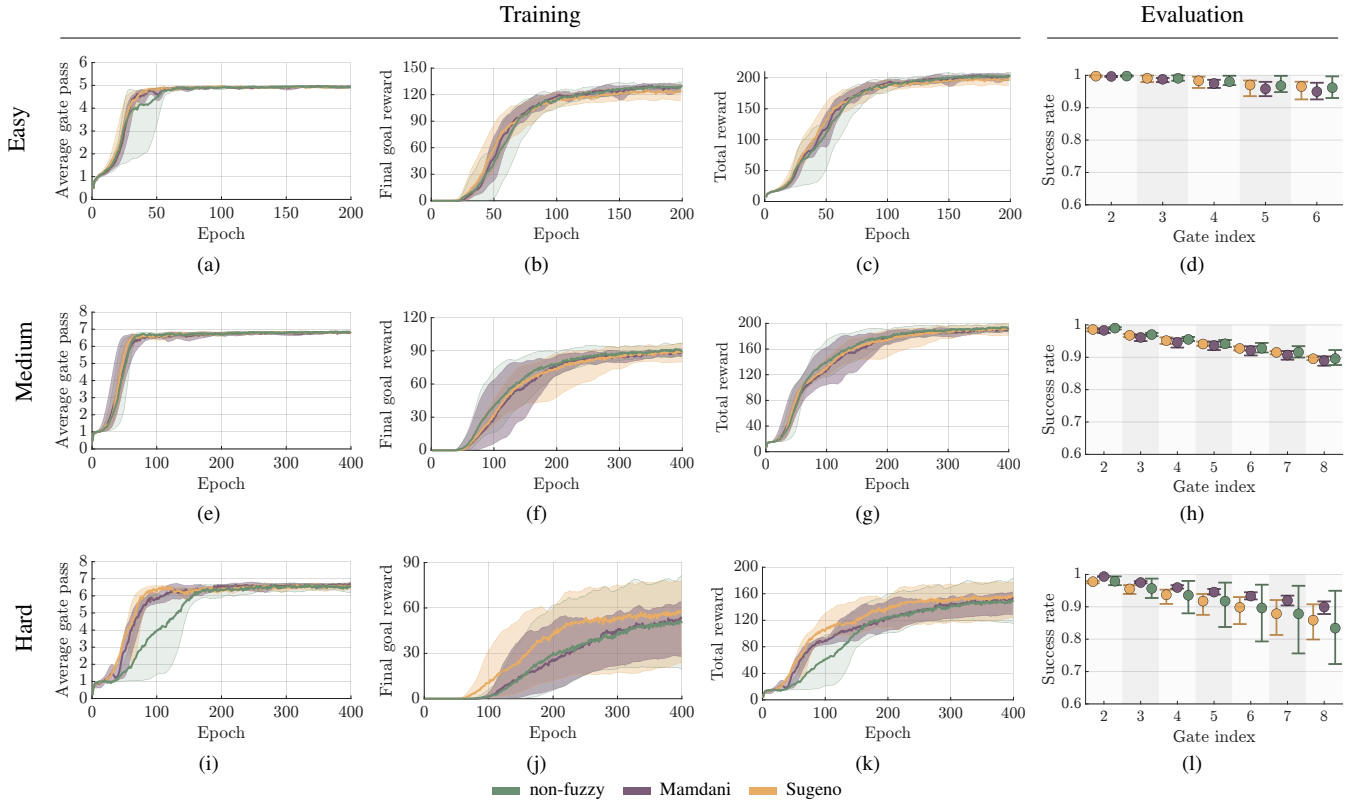


Fig. 5. Training and evaluation results for the Easy, Medium, and Hard scenarios. The left three columns (a–c, e–g, i–k) report training performance, averaged over five random seeds. Thick solid lines denote the mean performance across seeds, while the semi-transparent shaded regions indicate variability across seeds. The rightmost column (d, h, l) presents evaluation results, where each trained policy is tested over 5000 evaluations. Circular markers denote the mean gate-passing success rate at each gate index, while vertical lines indicate the minimum–maximum range observed across all seeds. Success rates are normalized between 0 and 1, representing the probability of successfully passing each gate. Results are shown for three reward configurations: non-fuzzy (green), Mamdani (purple), and Sugeno (yellow).

The Medium scenario increases complexity by introducing tighter longitudinal spacing and vertical variability. The number of gates is increased to eight, while the distance between gates is reduced to $1.0\ \mathrm{m},\ 1.25\ \mathrm{m}, 1.5\ \mathrm{m}$. The gate diameters remain unchanged, but the gate heights are $0.5\ \mathrm{m},\ 1.0\ \mathrm{m},\ 1.5\ \mathrm{m},\ 2.0\ \mathrm{m}$. This scenario requires the policy to handle increased vertical variation. The Hard scenario further increases difficulty by reducing the gate diameter to $0.45\mathrm{m}$ and introducing larger yaw perturbations sampled from $[-60°, 60°]$. Consequently, the Hard scenario imposes substantially greater demands on precise trajectory control, accurate yaw alignment, and robust perception during rapid alternating maneuvers. Example environments for each difficulty level are visualized in Fig. 4. In simulation, the UAV is modeled as an X-configuration quadrotor with 5 inch propellers, having a propeller tip–to–tip footprint of $0.325\,\mathrm{m} \times 0.325\,\mathrm{m}$ and a height of $0.08\,\mathrm{m}$.

All experiments are carried out using an actor–critic framework (PPO) with continuous action spaces, trained in IsaacLab [26] with 2048 parallel simulation environments and a horizon length of 128. The policy and value networks share a multilayer perceptron architecture with hidden layers $[128, 128]$. Training employs an adaptive learning rate with an initial value of 4×10^{-4}, clipping $\epsilon = 0.2$, and entropy regularization with coefficient 0.01. Policies are trained for up to 400 epochs. All experiments are executed on a system equipped with an Intel® Core™ i7-14700K CPU and a NVIDIA GeForce RTX 4080 GPU.

IV. Results

We evaluate the proposed fuzzy reward-shaping strategy on three scenarios of increasing difficulty (Easy, Medium, and Hard). For each scenario, we report training curves and evaluate the trained policies in the corresponding environment under a unified protocol. The protocol is summarized as follows:

- Three reward formulations—non-fuzzy PFBRS, Sugeno fuzzy-based reward shaping, and Mamdani fuzzy-based reward shaping—are evaluated across three scenarios (Easy, Medium, and Hard).
- All reward formulations described in Section III are applied without any environment-specific re-tuning. In particular, fuzzy logic-based velocity–distance reward r^{vd} is kept unchanged across all scenarios and difficulty levels, ensuring that observed performance differences are not influenced by scenario-specific tuning.
- For each scenario and reward formulation, training is repeated with multiple random seeds[1], and results are reported as

[1]Seeds used: {5, 8, 16, 32, 36}

the mean and variance across seeds. Figure 5 decomposes the total reward into its components—gate-pass reward (Figs. 5a, 5e, 5i), hovering reward (Figs. 5b, 5f, 5j), and total reward (Figs. 5c, 5g, 5k)—during training. For evaluation, we execute each trained policy and report gate-passing performance as the number (and per-gate rate) of successfully traversed gates (Figs. 5d, 5h, 5l).

A. Easy scenario

In the Easy scenario, all reward configurations exhibit rapid convergence, as illustrated in Fig. 5a–5c. As shown in Fig. 5a, gate-passing performance converges within approximately 50 training epochs, with agents successfully traversing approximately 5 out of 6 gates on average across all seeds. This rapid convergence can be attributed to the reduced environmental stochasticity inherent in this scenario, including uniform gate heights and larger inter-gate spacing. Under these conditions, fuzzy-based reward formulations are observed to converge slightly earlier than their non-fuzzy counterparts for this metric; however, differences between reward formulations are not clearly distinguishable through the gate-passing metric alone. As shown in Fig. 5b, the final goal-hovering reward differs only marginally across the non-fuzzy, Mamdani, and Sugeno formulations. By the end of training, all three achieve comparable mean reward and variance, and the total accumulated reward in Fig. 5c is similarly aligned, indicating equivalent performance in the Easy scenario.

Evaluation results for gate traversal success rate shown in Fig. 5d, further indicates that all three reward formulations achieve very similar final-gate success rates in the Easy scenario. On average, the probability of passing the final gate exceeds 95% across methods, reflecting the overall simplicity of the task. While the non-fuzzy formulation reaches 100% success in some individual test runs, its average performance remains closely aligned with the Sugeno-based formulation, with only marginal differences observed across seeds.

B. Medium scenario

In the Medium scenario, all reward formulations exhibit stable convergence across all seeds, with learning dynamics closely resembling those observed in the Easy case, albeit with increased variability. The gate-pass reward in Fig. 5e converges within approximately 60 training epochs, with agents traversing about 7 out of 8 gates on average across all seeds. Although all methods converge, the final goal-hovering reward in Fig. 5f is lower than in the Easy scenario, consistent with the increased task difficulty and the reduced time available for hovering at the final goal. Figure 5g reports the total reward, showing convergence for all methods within approximately 300 training epochs. The gate traversal success rate, as shown in Fig. 5h, is used for evaluation in this scenario and indicates that all reward formulations achieve high success rates. The probability of passing the final gate exceeds 90% for all three methods, demonstrating robust task completion. Among them, the non-fuzzy formulation attains a slightly higher final success rate, although the overall differences remain marginal.

C. Hard scenario

In the Hard scenario, increased yaw perturbations and narrower gate geometries lead to clearer performance differences between reward formulations. While all methods converge across all seeds, fuzzy-based rewards achieve faster and more consistent gate-passing convergence, stabilizing after approximately 100 training epochs, compared to around 150 epochs for the non-fuzzy formulation (Fig. 5i). All methods converge to an average gate-passing value above 6, although the non-fuzzy approach exhibits higher variability across seeds during early training.

Despite achieving rapid gate traversal after convergence, the non-fuzzy formulation progresses more slowly in the final goal hovering task than the Mamdani and Sugeno formulations (Fig. 5j). Among all methods, the Sugeno-based reward demonstrates the most stable learning behavior and achieves the highest total accumulated reward in the Hard scenario (Fig. 5k).

The gate traversal success rate for the Hard scenario is illustrated in Fig. 5l. The Mamdani formulation achieves the highest average success rate, exceeding 90%, with the lowest variance across seeds. Although the non-fuzzy formulation reaches up to 95% success in some runs, its performance fluctuates substantially across seeds. Overall, these results indicate that fuzzy-based reward formulations provide improved robustness under increased task difficulty.

V. Conclusion

This paper introduces FARS, a fuzzy logic–based reward shaping method that incorporates interpretable, human-inspired heuristics into RL. By encoding domain knowledge through linguistic rules, the proposed approach adaptively modulates velocity–distance and task-related reward components based on the agent's state, providing an alternative to fixed, hand-crafted reward formulations. The method is evaluated on autonomous drone racing scenarios of increasing difficulty using three reward formulations: non-fuzzy, Mamdani-based, and Sugeno-based fuzzy inference. In simpler environments, all three methods achieve comparable performance, with only marginal differences observed and, in some cases, slightly better results obtained by the non-fuzzy formulation. As task complexity increases, fuzzy-based reward formulations tend to converge faster and exhibit more stable behavior across training seeds, resulting in more consistent gate-passing performance. These findings highlight the potential of fuzzy logic–based reward shaping as an effective design choice in complex and variable settings, while maintaining competitive performance in simpler scenarios.

Future work will focus on real-time evaluation to assess robustness beyond simulation and analyze sim-to-real transfer performance. We will also investigate unifying different reward formulations into a compact fuzzy framework, along with learning-based adaptation of membership functions and rule sets during training.

REFERENCES

[1] E. Sayar, G. Iacca, and A. Knoll, "Multi-objective evolutionary hindsight experience replay for robot manipulation tasks," in *Proceedings of the Genetic and Evolutionary Computation Conference*, ser. GECCO '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 403–411. [Online]. Available: https://doi.org/10.1145/3638529.3654045

[2] E. Sayar, Z. Bing, C. D'Eramo, O. S. Oguz, and A. Knoll, "Contact energy based hindsight experience prioritization," in *2024 IEEE International Conference on Robotics and Automation (ICRA)*, 2024, pp. 5434–5440.

[3] V. H. Dang, A. Redder, H. X. Pham, A. Sarabakha, and E. Kayacan, "Vds-nav: Volumetric depth-based safe navigation for aerial robots–bridging the sim-to-real gap," *IEEE Robotics and Automation Letters*, vol. 10, no. 10, pp. 11 038–11 045, 2025.

[4] H. I. Ugurlu, A. Redder, and E. Kayacan, "Lyapunov-inspired deep reinforcement learning for robot navigation in obstacle environments," in *2025 IEEE Symposium on Computational Intelligence on Engineering/Cyber Physical Systems (CIES)*, 2025, pp. 1–8.

[5] A. Y. Ng, D. Harada, and S. Russell, "Policy invariance under reward transformations: Theory and application to reward shaping," in *Icml*, vol. 99. Citeseer, 1999, pp. 278–287.

[6] E. Kayacan and R. Maslim, "Type-2 fuzzy logic trajectory tracking control of quadrotor vtol aircraft with elliptic membership functions," *IEEE/ASME Transactions on Mechatronics*, vol. 22, no. 1, pp. 339–348, 2017.

[7] C. Fu, A. Sarabakha, E. Kayacan, C. Wagner, R. John, and J. M. Garibaldi, "Input uncertainty sensitivity enhanced nonsingleton fuzzy logic controllers for long-term navigation of quadrotor uavs," *IEEE/ASME Transactions on Mechatronics*, vol. 23, no. 2, pp. 725–734, 2018.

[8] A. Sarabakha and E. Kayacan, "Online deep fuzzy learning for control of nonlinear systems using expert knowledge," *IEEE Transactions on Fuzzy Systems*, vol. 28, no. 7, pp. 1492–1503, 2020.

[9] A. Sarabakha, C. Fu, and E. Kayacan, "Intuit before tuning: Type-1 and type-2 fuzzy logic controllers," *Applied Soft Computing*, vol. 81, p. 105495, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494619302650

[10] E. Camci, D. R. Kripalani, L. Ma, E. Kayacan, and M. A. Khanesar, "An aerial robot for rice farm quality inspection with type-2 fuzzy neural networks tuned by particle swarm optimization-sliding mode control hybrid algorithm," *Swarm and Evolutionary Computation*, vol. 41, pp. 1–8, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650217302407

[11] E. Camci and E. Kayacan, "Game of drones: Uav pursuit-evasion game with type-2 fuzzy logic controllers tuned by reinforcement learning," in *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, 2016, pp. 618–625.

[12] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," 2017. [Online]. Available: https://arxiv.org/abs/1707.06347

[13] M. Chen, H. K. Lam, Q. Shi, and B. Xiao, "Reinforcement learning-based control of nonlinear systems using lyapunov stability concept and fuzzy reward scheme," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 67, no. 10, pp. 2059–2063, 2020.

[14] M. C. Bingol, "A safe navigation algorithm for differential-drive mobile robots by using fuzzy logic reward function-based deep reinforcement learning," *Electronics*, vol. 14, no. 8, 2025.

[15] M. Faessler, A. Franchi, and D. Scaramuzza, "Differential flatness of quadrotor dynamics subject to rotor drag for accurate tracking of high-speed trajectories," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 620–626, 2018.

[16] H. I. Ugurlu, X. H. Pham, and E. Kayacan, "Sim-to-real deep reinforcement learning for safe end-to-end planning of aerial robots," *Robotics*, vol. 11, no. 5, 2022. [Online]. Available: https://www.mdpi.com/2218-6581/11/5/109

[17] Z. Qiao, X. H. Pham, S. Ramasamy, X. Jiang, E. Kayacan, and A. Sarabakha, "Continual learning for robust gate detection under dynamic lighting in autonomous drone racing," in *2024 International Joint Conference on Neural Networks (IJCNN)*, 2024, pp. 1–8.

[18] H. X. Pham, A. Sarabakha, M. Odnoshyvkin, and E. Kayacan, "Pencilnet: Zero-shot sim-to-real transfer learning for robust gate perception in autonomous drone racing," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 847–11 854, 2022.

[19] K. F. Andersen, H. X. Pham, H. I. Ugurlu, and E. Kayacan, "Event-based navigation for autonomous drone racing with sparse gated recurrent network," in *2022 European Control Conference (ECC)*, 2022, pp. 1342–1348.

[20] H. X. Pham, H. I. Ugurlu, J. Le Fevre, D. Bardakci, and E. Kayacan, "Chapter 15 - deep learning for vision-based navigation in autonomous drone racing," in *Deep Learning for Robot Perception and Cognition*, A. Iosifidis and A. Tefas, Eds. Academic Press, 2022, pp. 371–406.

[21] H. X. Pham, I. Bozcan, A. Sarabakha, S. Haddadin, and E. Kayacan, "Gatenet: An efficient deep neural network architecture for gate perception using fish-eye camera in autonomous drone racing," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 4176–4183.

[22] T. Morales, A. Sarabakha, and E. Kayacan, "Image generation for efficient neural network training in autonomous drone racing," in *2020 International Joint Conference on Neural Networks (IJCNN)*, 2020, pp. 1–8.

[23] H. Wang, J. Xing, N. Messikommer, and D. Scaramuzza, "Environment as policy: Learning to race in unseen tracks," 2026.

[24] F. Yu, Y. Hu, Y. Su, Y. Deng, L. Zhang, and D. Zou, "Mastering diverse, unknown, and cluttered tracks for robust vision-based drone racing," 2025.

[25] M. L. Puterman, "Chapter 8 markov decision processes," in *Stochastic Models*, ser. Handbooks in Operations Research and Management Science. Elsevier, 1990, vol. 2, pp. 331–434. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0927050705801720

[26] M. Mittal, P. Roth, J. Tigue, A. Richard, O. Zhang, P. Du, A. Serrano-Munoz, X. Yao, R. Zurbrügg, N. Rudin *et al.*, "Isaac lab: A gpu-accelerated simulation framework for multi-modal robot learning," *arXiv preprint arXiv:2511.04831*, 2025.

[27] E. Mamdani, "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the Institution of Electrical Engineers*, vol. 121, pp. 1585–1588, 1974. [Online]. Available: https://digital-library.theiet.org/doi/abs/10.1049/piee.1974.0328

[28] M. Sugeno, *Industrial Applications of Fuzzy Control*. USA: Elsevier Science Inc., 1985.