

## Article

# Deep Reinforcement Learning for Collision-Aware Leader–Follower Formation Control

Sławomir Romaniuk \*  and Jakub Budnik

Department of Automatic Control and Robotics, Faculty of Electrical Engineering, Białystok University of Technology, 15-351 Białystok, Poland; jakub.budnik2@gmail.com

\* Correspondence: s.romaniuk@pb.edu.pl

## Abstract

This work tackles leader–follower navigation in scenarios where rapid, unpredictable leader maneuvers can trigger unsafe proximity or collisions. We apply Proximal Policy Optimization (PPO) to learn adaptive follower behavior capable of tracking the leader’s direction while considering a minimum-distance safety objective. A task-specific reward function—capturing formation constraints, collision avoidance, control-effort regularization, and responsiveness to sudden directional changes—enables effective policy learning. The results show that the PPO-based follower provides a favorable safety–tracking trade-off compared with the considered A2C and PID baselines under the simulated test conditions, although collision-free operation is not guaranteed in all episodes.

**Keywords:** deep reinforcement learning; proximal policy optimization; leader–follower formation; collision avoidance; mobile robotics; MuJoCo; autonomous robots

## 1. Introduction

Leader–follower formation control is a fundamental problem in cooperative mobile robotics, where a follower robot must maintain a prescribed relative configuration with respect to a moving leader. This paradigm is relevant to coordinated navigation, autonomous transportation, surveillance, warehouse robotics, and multi-robot exploration, where stable formation maintenance and safe inter-robot distances are essential [1,2]. In practice, abrupt changes in leader heading or velocity can increase tracking error and reduce the safety margin between robots.

Classical formation-control methods include trajectory-tracking and distance-bearing feedback control, PID-like controllers, LQR-based methods, consensus protocols, Lyapunov-based control, artificial potential fields, model predictive control, and control barrier function formulations [3–9]. They offer interpretability, computational efficiency, and, in many cases, formal stability or safety arguments. Recent leader–follower UAV studies also address collision avoidance and cooperative tracking under communication constraints [10,11]. However, compact local observations, nonlinear dynamics, abrupt maneuvers, and competing safety–tracking objectives can make the design of a single analytical controller difficult.

Reinforcement learning provides a data-driven framework for learning control policies through interaction with the environment [12,13]. Instead of relying solely on an explicitly derived control law, an RL agent optimizes a long-term objective defined by a reward function. Deep reinforcement learning has therefore gained increasing attention in robotic continuous-control, mapless-navigation, and collision-avoidance tasks, especially



Academic Editor: Egidio D’Amato

Received: 23 April 2026

Revised: 22 June 2026

Accepted: 29 June 2026

Published: 1 July 2026

**Copyright:** © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

when the system is nonlinear, partially observable, or difficult to model precisely [14–18]. In multi-agent and swarm settings, reinforcement learning has also been investigated for decentralized coordination and collective behavior learning [19,20]. Among policy-gradient methods, Proximal Policy Optimization (PPO) is commonly used due to its stable policy updates and practical effectiveness in continuous-action problems [21]. Advantage Actor–Critic methods, such as A2C, are also frequently used as reference algorithms for evaluating policy-gradient approaches [22].

This paper investigates a compact-observation, safety-aware reinforcement learning framework for leader–follower formation control of a differential-drive mobile robot. The proposed framework is based on PPO, but the contribution of the work is not a modification of the PPO algorithm itself. Instead, the focus is placed on the design and evaluation of a modular reward function that combines formation tracking, collision avoidance, orientation alignment, and actuator-level control regularization. The follower receives a six-dimensional observation vector containing relative geometric and velocity-related information and produces two continuous wheel-control commands. The task is evaluated in a MuJoCo-based simulation environment [23], where the leader follows multiple trajectories containing both smooth and abrupt motion patterns. Although the proposed reward terms do not provide formal constraint-satisfaction guarantees, their safety-oriented design is motivated by the broader literature on safe and constrained reinforcement learning [24,25].

No physical equipment was used in this study; all experiments were conducted in simulation. The software version numbers were not available in the manuscript source. Therefore, the corresponding software websites and access dates are provided as requested by the journal: MuJoCo (<https://mujoco.org>, accessed on 28 June 2026), Gymnasium (<https://gymnasium.farama.org>, accessed on 28 June 2026), and Stable-Baselines3 (<https://stable-baselines3.readthedocs.io>, accessed on 28 June 2026).

The study is intended to clarify how individual reward components influence the behavior of the learned follower policy. For this purpose, several reward-weight variants are examined, allowing the relationship between tracking accuracy, safety, and control smoothness to be analyzed systematically. The evaluation considers not only average formation error but also safety violation rate, safety violation severity, collision count, velocity-tracking error, control smoothness, and post-maneuver recovery behavior. In addition, the learned PPO policy is compared with A2C and with a classical geometry-based leader–follower controller under the same observation, action, and trajectory conditions. The experimental analysis is further extended by robustness-oriented tests involving observation noise, actuation delay, and wheel-slip or action-scaling uncertainty. These tests are not intended to claim full sim-to-real validation, but rather to provide an initial assessment of the learned policy under more realistic disturbances, in line with the common observation that simulation-trained policies may be sensitive to the reality gap and to deployment-time uncertainty [26–29].

The main contributions of this work are as follows. The paper proposes a compact-observation reinforcement-learning framework for the collision-aware leader–follower formation control of a differential-drive robot under abrupt leader maneuvers. It also introduces a modular safety-aware reward formulation and evaluates it through a controlled reward-weight sensitivity study. The evaluation protocol combines formation accuracy, safety-violation rate and severity, velocity synchronization, control regularity, and post-maneuver recovery behavior, which makes it possible to analyze the learned policies beyond the accumulated reward alone. In addition, PPO is compared with A2C and with a classical distance–bearing PID controller under identical observation, action, and trajectory conditions. Finally, the study includes an observation-space ablation analysis

and a test-time robustness evaluation under observation noise, actuation delay, and wheel-slip-like action scaling.

The remainder of this paper is organized as follows. Section 2 defines the leader–follower control problem, including the robot model, observation space, action space, and task assumptions. Section 3 presents the reinforcement learning methodology, reward formulation, neural-network architecture, and training configuration. Section 4 reports the experimental results, including reward-weight sensitivity analysis, baseline comparisons, generalization tests, and robustness evaluation. Finally, Section 5 concludes the paper and discusses limitations and directions for future work.

## 2. Problem Formulation

The considered task is a leader–follower formation-control problem in which a differential-drive mobile robot, hereafter referred to as the follower, must track a desired position behind a moving leader while maintaining a safe distance and producing feasible wheel-actuator commands. The task is formulated explicitly as a continuous-control decision-making problem with limited observations, rather than only as a descriptive simulation scenario. The formulation separates the full simulator state, the compact observation available to the policy, the continuous wheel-command action space, the leader-motion model, and the safety-oriented objective.

The control problem is represented as

$$\mathcal{M} = (\mathcal{S}, \mathcal{O}, \mathcal{A}, P, h, r, \gamma, T), \quad (1)$$

where the following hold:

- $\mathcal{S}$ —full MuJoCo simulator state space;
- $\mathcal{O}$ —compact observation space available to the learning policy;
- $\mathcal{A}$ —continuous action space;
- $P(s_{t+1}|s_t, a_t)$ —transition model induced by the MuJoCo physics engine and the imposed leader trajectory;
- $h: \mathcal{S} \rightarrow \mathcal{O}$ —observation function;
- $r(o_t, a_t)$ —reward function;
- $\gamma$ —discount factor;
- $T$ —finite episode horizon.

At each time step  $t$ , the policy receives the observation  $o_t = h(s_t)$ , selects an action  $a_t$ , and the simulator advances the physical state according to the robot dynamics, contact interactions, actuator limits, and externally imposed leader motion. The goal is to learn a stochastic control policy

$$\pi_\theta(a_t|o_t) \quad (2)$$

where the following hold:

- $\pi_\theta$ —stochastic control policy;
- $a_t$ —action selected at time step  $t$ ;
- $o_t$ —observation available at time step  $t$ ;
- $\theta$ —policy-network parameters.

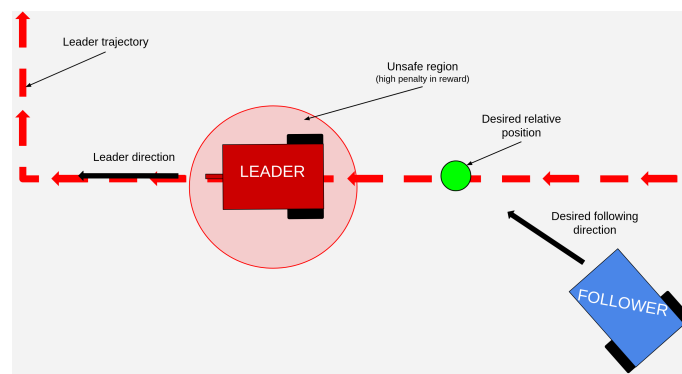
The policy is optimized by maximizing the expected discounted return

$$J(\theta) = \mathbb{E}_{\pi_\theta} \left[ \sum_{t=0}^{T-1} \gamma^t r(o_t, a_t) \right]. \quad (3)$$

where the following hold:

- $J(\theta)$ —expected discounted return optimized by the policy;
- $\theta$ —policy-network parameters;
- $\mathbb{E}_{\pi_\theta}[\cdot]$ —expectation under policy  $\pi_\theta$ ,
- $\gamma$ —discount factor;
- $r(o_t, a_t)$ —reward obtained from observation  $o_t$  and action  $a_t$ .

The policy does not receive the complete MuJoCo state vector, global contact state, or all body poses and velocities. Instead, it operates on a six-dimensional vector of relative geometric and velocity-related quantities. Therefore, the implemented task can be interpreted as a compact-observation, partially observable leader–follower control problem. This compact representation is intentional: it reduces dependence on absolute world coordinates and forces the controller to react to the relative leader–follower configuration. The leader–follower task geometry is illustrated in Figure 1.



**Figure 1.** Leader–follower formation-control task. The follower is required to track the green reference point located behind the leader while avoiding the unsafe region around the leader.

### 2.1. Simulated Objects and Physical Environment

The simulation environment was implemented in MuJoCo [23] and integrated with the Gymnasium interface [30]. The environment contains a differential-drive follower robot, a kinematic leader object, and an auxiliary formation reference point placed behind the leader. The follower is dynamically simulated, whereas the leader is controlled kinematically through a MuJoCo mocap body. This choice makes it possible to impose repeatable leader trajectories and to isolate the decision-making problem of the follower.

The MuJoCo model uses a fixed simulation time step of 0.01 s. In the environment implementation, each control step consists of five internal MuJoCo integration substeps, resulting in an effective control interval of  $\Delta t = 0.05$  s. Gravity is set to  $9.81 \text{ m/s}^2$ , and the ground is modeled as a horizontal plane with sliding, torsional, and rolling friction coefficients equal to 0.8, 0.01, and 0.001, respectively. This provides a controlled and repeatable physics-based testbed for evaluating the learned control policy.

The main geometric parameters of the simulated objects are summarized in Table 1. The dimensions are reported as full physical dimensions derived from the MuJoCo model.

**Table 1.** Geometric and physical parameters of the simulated leader–follower environment.

| Object                     | MuJoCo Type            | Dimensions/Parameters                                  | Mass    |
|----------------------------|------------------------|--------------------------------------------------------|---------|
| Follower chassis           | box                    | $0.40 \times 0.30 \times 0.10 \text{ m}$               | —       |
| Drive wheel ( $\times 2$ ) | cylinder               | radius $r_w = 0.07 \text{ m}$ , width $0.04 \text{ m}$ | 1.5 kg  |
| Lateral wheel spacing      | —                      | $L \approx 0.346 \text{ m}$                            | —       |
| Passive support wheel      | sphere                 | radius $0.02 \text{ m}$                                | 0.01 kg |
| Leader body                | kinematic mocap object | visual body with wheel-like markers                    | —       |
| Formation reference point  | mocap sphere           | radius $0.05 \text{ m}$                                | —       |
| Wheel actuator range       | motor                  | $u_L, u_R \in [-10, 10]$                               | —       |

## 2.2. Differential-Drive Follower Model

The follower is a differential-drive platform moving on a planar surface. Let

$$q_F(t) = [x_F(t), y_F(t), \psi_F(t)]^\top \quad (4)$$

where the following hold:

- $q_F(t)$ —planar pose of the follower;
- $x_F(t)$ —longitudinal position of the follower body;
- $y_F(t)$ —lateral position of the follower body;
- $\psi_F(t)$ —yaw angle of the follower.

The ideal differential-drive kinematics are expressed as

$$\dot{x}_F = v_F \cos \psi_F, \quad \dot{y}_F = v_F \sin \psi_F, \quad \dot{\psi}_F = \omega_F, \quad (5)$$

where the following hold:

- $v_F$ —linear velocity of the follower;
- $\omega_F$ —angular velocity of the follower;
- $\psi_F$ —yaw angle of the follower.

These velocities are related to the left and right wheel angular velocities,  $\omega_L$  and  $\omega_R$ , by

$$v_F = \frac{r_w}{2}(\omega_R + \omega_L), \quad \omega_F = \frac{r_w}{L}(\omega_R - \omega_L), \quad (6)$$

where the following hold:

- $r_w$ —wheel radius, equal to 0.07 m;
- $L$ —lateral distance between the two drive wheels, approximately 0.346 m;
- $\omega_L$ —angular velocity of the left wheel;
- $\omega_R$ —angular velocity of the right wheel.

The above kinematic equations clarify the relationship between wheel commands and planar motion. In the actual implementation, however, the follower motion is generated by MuJoCo through numerical integration of the full physical model, including body inertia, gravity, wheel-ground contact, friction, joint damping, and actuator constraints. The policy therefore controls the robot through wheel actuators, while the resulting motion is determined by the simulator dynamics rather than by a purely kinematic update.

## 2.3. Leader Motion and Formation Reference

The leader position in the horizontal plane is denoted by

$$p_L(t) = [x_L(t), y_L(t)]^\top. \quad (7)$$

where the following hold:

- $p_L(t)$ —planar leader position at time step  $t$ ;
- $x_L(t)$ —longitudinal leader position;
- $y_L(t)$ —lateral leader position.

During training, the leader follows a set of predefined but diverse trajectories. These trajectories include circular, elliptical, sinusoidal, spiral, square, polygonal, and Lissajous-like paths. They are designed to expose the follower to both smooth and abrupt leader maneuvers, including changes in curvature, heading, and speed. The leader is not dynamically affected by the follower, and no direct communication between the leader and the follower is assumed.

The instantaneous direction of leader motion is estimated from consecutive leader positions:

$$\hat{d}_L(t) = \frac{p_L(t) - p_L(t - \Delta t)}{\|p_L(t) - p_L(t - \Delta t)\|_2}, \quad (8)$$

where the following hold:

- $\hat{d}_L(t)$ —unit vector of the instantaneous leader-motion direction,
- $p_L(t)$ —current leader position,
- $p_L(t - \Delta t)$ —leader position at the previous control step,
- $\Delta t$ —control interval,
- $\|\cdot\|_2$ —Euclidean norm.

This expression is used when the leader displacement is nonzero. The desired formation point is located behind the leader along the opposite direction of motion:

$$p_{\text{ref}}(t) = p_L(t) - d_{\text{ref}}\hat{d}_L(t), \quad (9)$$

where the following hold:

- $p_{\text{ref}}(t)$ —desired reference position behind the leader;
- $p_L(t)$ —leader position in the horizontal plane;
- $d_{\text{ref}}$ —desired behind-leader offset, equal to 1.0 m;
- $\hat{d}_L(t)$ —unit vector of the instantaneous leader-motion direction.

The corresponding formation error is

$$e_{\text{form}}(t) = \|p_F(t) - p_{\text{ref}}(t)\|_2, \quad (10)$$

where the following hold:

- $e_{\text{form}}(t)$ —instantaneous formation error;
- $p_F(t)$ —planar follower position;
- $p_{\text{ref}}(t)$ —desired reference position behind the leader;
- $\|\cdot\|_2$ —Euclidean norm.

Safety is defined with respect to the physical distance between the follower and the leader:

$$d_L(t) = \|p_F(t) - p_L(t)\|_2. \quad (11)$$

The follower is considered to violate the safety constraint when

$$d_L(t) < d_{\text{safe}}, \quad (12)$$

where the following hold:

- $d_L(t)$ —direct leader–follower distance;
- $d_{\text{safe}}$ —minimum admissible leader–follower distance, equal to 0.5 m in the nominal environment configuration.

The learning objective therefore combines formation tracking with avoidance of excessive proximity to the leader.

#### 2.4. Random-Seed Training Stability

To assess the repeatability of the learning process, an additional random-seed training stability test was performed. The purpose of this experiment was not to replace the full nominal benchmark, but to verify whether the learning trend observed for the selected PPO configuration is preserved under different random initializations. The test was conducted

for the safety-oriented PPO\_R3 variant, which was previously identified as one of the representative reward configurations.

Three independent shortened PPO training runs were performed using random seeds

$$s \in \{0, 1, 2\}. \quad (13)$$

where the following hold:

- $s$ —base random seed used for a given training run.

In each run, the reward weights, policy architecture, observation vector, action space, PPO hyperparameters, and training trajectories were kept unchanged. Only the random seed was modified. The seed affects the initial policy parameters, stochastic action sampling during training, minibatch ordering during PPO optimization, and randomized initial follower poses at environment reset. For parallel training environments, the environment-specific seed is shifted by the environment index:

$$s_i = s + i, \quad (14)$$

where the following hold:

- $s_i$ —seed assigned to the  $i$ -th parallel environment;
- $s$ —base random seed used for the training run;
- $i$ —index of the parallel environment.

Because this experiment was intended as a supplementary repeatability check, a shortened training horizon was used. Each run was trained for approximately  $5.08 \times 10^5$  environment steps, corresponding to 336 logged training episodes. Therefore, the results should be interpreted as evidence of qualitative training repeatability rather than as a full statistical validation of final policy performance.

The training reward is negative because the reward function is dominated by penalty terms. Thus, values closer to zero indicate a lower accumulated weighted penalty and better optimization of the corresponding training objective. For each seed, the initial reward was computed as the mean reward over the first 50 logged episodes, whereas the final reward was computed as the mean reward over the last 50 logged episodes. The final formation error, velocity-tracking error, and smoothness were computed from the last 10% of logged training-metric samples.

Table 2 shows that all three shortened PPO\_R3 training runs improved the training objective. The mean final reward increased from  $-12,421.2$  to  $-6829.1$ , corresponding to an average relative improvement of  $44.4\% \pm 6.7\%$ . Since all three runs show a positive reward improvement, the result suggests that the observed learning trend is not limited to a single favorable random seed.

**Table 2.** Random-seed training stability for shortened PPO\_R3 training runs with seeds  $s \in \{0, 1, 2\}$ . Reward values are negative because the objective is dominated by penalty terms; therefore, less negative values indicate better training performance.

| Seed       | Initial Reward     | Final Reward    | Improvement [%] | Final Error [m] | Velocity-Tracking Error [m/s] | Smoothness    |
|------------|--------------------|-----------------|-----------------|-----------------|-------------------------------|---------------|
| 0          | −11,335.6          | −6065.4 ± 159.2 | 46.5            | 1.727           | 1.315                         | 1.668         |
| 1          | −10,901.4          | −6876.6 ± 562.2 | 36.9            | 1.728           | 1.269                         | 1.620         |
| 2          | −15,026.8          | −7545.2 ± 504.4 | 49.8            | 2.003           | 1.383                         | 1.600         |
| Mean ± std | −12,421.2 ± 2266.9 | −6829.1 ± 741.0 | 44.4 ± 6.7      | 1.819 ± 0.159   | 1.322 ± 0.058                 | 1.630 ± 0.035 |

The final reward still exhibits noticeable variation across seeds. The best final reward was obtained for seed 0, whereas seed 2 resulted in a lower final reward and a higher final formation error. This indicates that the shortened training process remains sensitive to stochastic initialization and sampling. Therefore, the experiment should not be interpreted as proof of full quantitative convergence across seeds.

The formation-error results show moderate variability. Seeds 0 and 1 resulted in almost identical final formation errors, equal to 1.727 m and 1.728 m, respectively. Seed 2 produced a higher final error of 2.003 m. The across-seed mean formation error was  $1.819 \pm 0.159$  m. This suggests that the training process produces qualitatively similar, although not identical, formation-tracking behavior under the considered shortened training horizon.

The velocity-tracking error and smoothness metrics are more consistent across seeds. The velocity-tracking error remains in the range from 1.269 m/s to 1.383 m/s, while the smoothness metric remains between 1.600 and 1.668. This indicates that the actuator-level behavior is relatively repeatable in the shortened training runs. However, the velocity-tracking error remains higher than in the fully trained nominal models, which confirms that the shortened random-seed experiment should be treated as a repeatability check rather than as a final performance benchmark.

Overall, the random-seed training experiment indicates that PPO\_R3 exhibits a repeatable qualitative learning trend across the tested seeds  $s \in \{0, 1, 2\}$ . All runs improved the reward objective, and the velocity-related metrics remained relatively consistent. At the same time, the differences in final reward and formation error show that shortened training still introduces moderate variability. Consequently, this experiment supports the reproducibility of the learning process at the trend level, while the main task-level conclusions should remain based on the fully trained policies evaluated in the nominal comparison.

### 2.5. Observation Space

The policy receives a compact six-dimensional observation vector

$$o_t = [\alpha_t, \bar{d}_{L,t}, \bar{\omega}_{F,t}, \bar{v}_{F,t}, \bar{v}_{L,t}, \Delta\psi_t]^\top \in \mathcal{O} \subset \mathbb{R}^6. \quad (15)$$

where the following hold:

- $o_t$ —compact observation vector at time step  $t$ ;
- $\alpha_t$ —relative bearing angle from the follower heading to the leader position;
- $\bar{d}_{L,t}$ —normalized follower–leader distance;
- $\bar{\omega}_{F,t}$ —normalized yaw rate of the follower;
- $\bar{v}_{F,t}$ —normalized forward velocity of the follower;
- $\bar{v}_{L,t}$ —normalized leader speed;
- $\Delta\psi_t$ —relative direction of leader motion with respect to the follower heading.

The components correspond directly to the quantities returned by the environment implementation. The normalized follower–leader distance is

$$\bar{d}_{L,t} = \text{clip}\left(\frac{d_L(t)}{d_{\max}}, 0, 1\right), \quad (16)$$

where the following hold:

- $d_L(t)$ —Euclidean leader–follower distance;
- $d_{\max}$ —maximum distance used for normalization, equal to 5.0 m;
- $\text{clip}(\cdot)$ —clipping operator limiting the value to the stated interval.

The normalized yaw rate of the follower is

$$\bar{\omega}_{F,t} = \text{clip}\left(\frac{\omega_F(t)}{\omega_{\max}}, -1, 1\right), \quad (17)$$

where the following hold:

- $\omega_F(t)$ —yaw rate of the follower;
- $\omega_{\max}$ —maximum yaw-rate magnitude used for normalization, equal to 5.0 rad/s.

The normalized forward velocity of the follower is

$$\bar{v}_{F,t} = \text{clip}\left(\frac{v_F(t)}{v_{F,\max}}, -1, 1\right), \quad (18)$$

where the following hold:

- $v_F(t)$ —forward velocity of the follower;
- $v_{F,\max}$ —maximum follower velocity used for normalization, equal to 3.0 m/s.

The normalized leader speed is

$$\bar{v}_{L,t} = \text{clip}\left(\frac{v_L(t)}{v_{L,\max}}, 0, 1\right), \quad (19)$$

where the following hold:

- $v_L(t)$ —leader speed;
- $v_{L,\max}$ —maximum leader speed used for normalization, equal to 3.0 m/s.

The relative direction of leader motion with respect to the follower heading is

$$\Delta\psi_t = \text{atan2}(\sin(\psi_L(t) - \psi_F(t)), \cos(\psi_L(t) - \psi_F(t))), \quad (20)$$

where the following hold:

- $\Delta\psi_t$ —wrapped relative direction of leader motion with respect to the follower heading;
- $\psi_L(t)$ —direction of leader motion;
- $\psi_F(t)$ —follower yaw angle.

The resulting observation space is bounded by

$$\mathcal{O} = [-\pi, \pi] \times [0, 1] \times [-1, 1] \times [-1, 1] \times [0, 1] \times [-\pi, \pi]. \quad (21)$$

This observation vector intentionally excludes absolute world coordinates and full simulator state variables. As a result, the learned policy must infer suitable wheel commands from relative leader–follower geometry and velocity information only.

## 2.6. Action Space and Actuator Mapping

The action selected by the policy is a two-dimensional continuous vector

$$a_t = [a_{L,t}, a_{R,t}]^T, \quad (22)$$

where the following hold:

- $a_t$ —continuous two-dimensional action vector;
- $a_{L,t}$ —normalized command for the left wheel actuator;
- $a_{R,t}$ —normalized command for the right wheel actuator.

The action space is

$$\mathcal{A} = [-1, 1]^2. \quad (23)$$

Before being applied to the MuJoCo actuators, the normalized action is clipped to  $[-1, 1]$  and linearly mapped to the actuator-control range:

$$u_{i,t}^{\text{phys}} = u_i^{\min} + \frac{a_{i,t} + 1}{2} (u_i^{\max} - u_i^{\min}), \quad i \in \{L, R\}. \quad (24)$$

where the following hold:

- $u_{i,t}^{\text{phys}}$ —physical actuator command applied to wheel  $i$ ;
- $a_{i,t}$ —normalized policy output for wheel  $i$ ;
- $u_i^{\min}$ —lower actuator-control limit for wheel  $i$ ;
- $u_i^{\max}$ —upper actuator-control limit for wheel  $i$ ;
- $i \in \{L, R\}$ —index denoting the left or right wheel actuator.

For the implemented robot model, the actuator limits are

$$u_L^{\min} = u_R^{\min} = -10, \quad u_L^{\max} = u_R^{\max} = 10. \quad (25)$$

where the following hold:

- $u_L^{\min}$ —lower control limit of the left wheel actuator;
- $u_R^{\min}$ —lower control limit of the right wheel actuator;
- $u_L^{\max}$ —upper control limit of the left wheel actuator;
- $u_R^{\max}$ —upper control limit of the right wheel actuator.

This explicit mapping ensures that the policy output remains normalized, while the simulator receives physically bounded actuator commands.

## 2.7. Transition Dynamics

The transition model  $P(s_{t+1}|s_t, a_t)$  is not specified analytically, because the physical evolution of the system is computed by the MuJoCo simulator. Let  $s_t$  denote the full simulator state, including body poses, velocities, joint states, contact states, and internal MuJoCo variables. The transition can be written compactly as

$$s_{t+1} = f_{\text{MJ}}(s_t, u_t^{\text{phys}}, p_L(t)), \quad (26)$$

where the following hold:

- $s_t$ —full simulator state at time step  $t$ ;
- $u_t^{\text{phys}}$ —physical actuator command vector applied to the follower;
- $p_L(t)$ —externally imposed leader position;
- $f_{\text{MJ}}(\cdot)$ —numerical MuJoCo integration procedure.

The observation used by the policy is then obtained through

$$o_t = h(s_t, p_L(t)). \quad (27)$$

where the following hold:

- $o_t$ —compact observation vector available to the policy;
- $h(\cdot)$ —observation function;
- $s_t$ —full simulator state;
- $p_L(t)$ —leader position.

This distinction is important because the simulator internally evolves a high-dimensional physical state, whereas the policy only observes the compact vector defined above.

## 2.8. Observation and Actuation Uncertainty

The nominal training environment uses direct observations and direct execution of wheel commands. However, to define the robustness evaluation considered in the revised experimental protocol, observation and actuation uncertainties can be introduced at test time. A noisy observation can be expressed as

$$\tilde{o}_t = o_t + \epsilon_t, \quad (28)$$

where the following hold:

- $\tilde{o}_t$ —noisy observation vector;
- $o_t$ —nominal observation vector;
- $\epsilon_t$ —observation-noise vector.

In particular, distance and bearing measurements can be perturbed as

$$\tilde{d}_L(t) = d_L(t) + \epsilon_d, \quad \epsilon_d \sim \mathcal{N}(0, \sigma_d^2), \quad (29)$$

$$\tilde{\alpha}_t = \alpha_t + \epsilon_\alpha, \quad \epsilon_\alpha \sim \mathcal{N}(0, \sigma_\alpha^2). \quad (30)$$

where the following hold:

- $\tilde{d}_L(t)$ —noisy leader–follower distance;
- $d_L(t)$ —nominal leader–follower distance;
- $\epsilon_d$ —distance-measurement noise;
- $\sigma_d$ —standard deviation of distance noise;
- $\tilde{\alpha}_t$ —noisy relative bearing angle;
- $\alpha_t$ —nominal relative bearing angle;
- $\epsilon_\alpha$ —bearing-measurement noise;
- $\sigma_\alpha$ —standard deviation of bearing noise.

Actuator delay can be modeled by executing an earlier action instead of the currently selected command:

$$a_t^{\text{exec}} = a_{t-k}, \quad (31)$$

where the following hold:

- $a_t^{\text{exec}}$ —action actually executed by the simulator at time step  $t$ ;
- $a_{t-k}$ —action generated  $k$  control steps earlier;
- $k$ —delay in control steps.

Wheel-slip-like actuation uncertainty can be approximated by action scaling:

$$a_t^{\text{exec}} = \lambda_t a_t, \quad \lambda_t \sim \mathcal{U}(\lambda_{\min}, \lambda_{\max}). \quad (32)$$

where the following hold:

- $a_t^{\text{exec}}$ —scaled action executed by the simulator;
- $a_t$ —nominal action generated by the policy;
- $\lambda_t$ —random action-scaling coefficient;
- $\mathcal{U}(\lambda_{\min}, \lambda_{\max})$ —uniform distribution over the scaling interval.

These perturbation models are not part of the nominal training setting, but they formalize the noise, delay, and actuator-effectiveness disturbances used to evaluate robustness of the learned policy.

### 2.9. Task Assumptions

The formulation assumes that the follower moves on a planar surface and is controlled only through two wheel actuators. The leader motion is imposed by the simulator and is not dynamically affected by the follower, which makes the leader trajectory repeatable across training and evaluation runs. No direct communication between the leader and the follower is available. Consequently, the policy receives only the compact six-dimensional observation vector defined above and does not have access to the full MuJoCo simulator state, global body poses, or contact information.

The desired formation point is located behind the leader along the direction opposite to its instantaneous motion. Safety is defined by a minimum admissible leader–follower distance  $d_{\text{safe}}$ . The nominal training process is conducted in a controlled MuJoCo environment, whereas the robustness evaluation introduces additional observation noise, action delay, and action-scaling uncertainty. These assumptions make the task sufficiently constrained for repeatable simulation-based learning while still preserving the central difficulty of compact-observation leader–follower control under changing leader maneuvers.

This formulation provides a rigorous and reproducible basis for the proposed safety-aware reinforcement-learning controller. It also explicitly identifies the state hidden inside the simulator, the limited observation available to the policy, the continuous wheel-command interface, the leader-motion assumptions, and the safety constraint that governs the leader–follower interaction.

## 3. Methodology

This section describes the reinforcement-learning methodology used to train the follower robot in the leader–follower formation-control task. The environment, observation space, action space, and transition model were defined in Section 2. Therefore, the present section focuses on the learning architecture, action-to-actuator mapping, reward formulation, training setup, and evaluation metrics. The main controller was trained using Proximal Policy Optimization (PPO), while Advantage Actor–Critic (A2C) was used as a reference actor–critic algorithm in the comparative experiments [21,22].

### 3.1. Policy Architecture

The control policy receives a compact six-dimensional observation vector and returns two continuous wheel-actuator commands. The observation vector used by the neural network is defined as

$$o_t = [\alpha_t, \bar{d}_{L,t}, \bar{\omega}_{F,t}, \bar{v}_{F,t}, \bar{v}_{L,t}, \Delta\psi_t]^\top \in \mathbb{R}^6. \quad (33)$$

where the following hold:

- $o_t$ —compact observation vector at time step  $t$ ;
- $\alpha_t$ —relative angle from the follower heading to the leader;
- $\bar{d}_{L,t}$ —normalized leader–follower distance;
- $\bar{\omega}_{F,t}$ —normalized yaw rate of the follower;
- $\bar{v}_{F,t}$ —normalized forward velocity of the follower;
- $\bar{v}_{L,t}$ —normalized leader speed;
- $\Delta\psi_t$ —relative direction of leader motion with respect to the follower heading.

The angular components are provided directly in radians within  $[-\pi, \pi]$ . The remaining scalar quantities are clipped and normalized as follows:

$$\bar{d}_{L,t} = \text{clip}\left(\frac{d_L(t)}{d_{\text{max}}}, 0, 1\right), \quad d_{\text{max}} = 5.0 \text{ m}. \quad (34)$$

$$\bar{\omega}_{F,t} = \text{clip}\left(\frac{\omega_F(t)}{\omega_{\max}}, -1, 1\right), \quad \omega_{\max} = 5.0 \text{ rad/s.} \quad (35)$$

$$\bar{v}_{F,t} = \text{clip}\left(\frac{v_F(t)}{v_{F,\max}}, -1, 1\right), \quad v_{F,\max} = 3.0 \text{ m/s.} \quad (36)$$

$$\bar{v}_{L,t} = \text{clip}\left(\frac{v_L(t)}{v_{L,\max}}, 0, 1\right), \quad v_{L,\max} = 3.0 \text{ m/s.} \quad (37)$$

where the following hold:

- $\bar{d}_{L,t}$ —normalized leader–follower distance;
- $d_L(t)$ —Euclidean leader–follower distance;
- $d_{\max}$ —maximum distance used for normalization;
- $\bar{\omega}_{F,t}$ —normalized yaw rate of the follower;
- $\omega_F(t)$ —yaw rate of the follower;
- $\omega_{\max}$ —maximum yaw-rate magnitude used for normalization;
- $\bar{v}_{F,t}$ —normalized forward velocity of the follower;
- $v_F(t)$ —forward velocity of the follower;
- $v_{F,\max}$ —maximum follower velocity used for normalization;
- $\bar{v}_{L,t}$ —normalized leader speed;
- $v_L(t)$ —leader speed;
- $v_{L,\max}$ —maximum leader speed used for normalization;
- $\text{clip}(\cdot)$ —clipping operator limiting each value to the stated interval.

The relative direction of leader motion is computed using a wrapped angular difference:

$$\Delta\psi_t = \text{atan2}(\sin(\psi_L(t) - \psi_F(t)), \cos(\psi_L(t) - \psi_F(t))). \quad (38)$$

where the following hold:

- $\Delta\psi_t$ —wrapped relative direction of leader motion with respect to the follower heading;
- $\psi_L(t)$ —instantaneous direction of leader motion;
- $\psi_F(t)$ —follower yaw angle;
- $\text{atan2}(\cdot)$ —four-quadrant arctangent used to keep the angular difference wrapped.

The policy output is a two-dimensional continuous action vector

$$a_t = [a_{L,t}, a_{R,t}]^\top \in [-1, 1]^2, \quad (39)$$

where the following hold:

- $a_t$ —continuous two-dimensional action vector;
- $a_{L,t}$ —normalized left-wheel command before actuator scaling;
- $a_{R,t}$ —normalized right-wheel command before actuator scaling.

The controller was implemented using the MlpPolicy architecture from Stable-Baselines3 [31]. Since no custom policy architecture was specified, the default feed-forward actor–critic multilayer perceptron was used. The actor and critic use separate multilayer perceptrons with two hidden layers of 64 units each and hyperbolic tangent activation functions. The actor maps the six-dimensional observation vector to the parameters of a diagonal Gaussian continuous action distribution, whereas the critic estimates the scalar value function. The log standard deviation of the action distribution is learned together with the policy parameters. No recurrent layers, memory modules, or image encoders were used. Table 3 summarizes the neural-network input, output, and policy representation.

**Table 3.** Neural-network input, output, and policy representation.

| Element              | Configuration                                                |
|----------------------|--------------------------------------------------------------|
| Policy type          | Feed-forward actor–critic multilayer perceptron              |
| Implementation       | Stable-Baselines3 MlpPolicy                                  |
| Input dimension      | 6                                                            |
| Output dimension     | 2                                                            |
| Actor hidden layers  | [64, 64] fully connected layers                              |
| Critic hidden layers | [64, 64] fully connected layers                              |
| Activation function  | Hyperbolic tangent, Tanh                                     |
| Actor output         | Parameters of a diagonal Gaussian wheel-command distribution |
| Critic output        | Scalar value estimate $V_\phi(o_t)$                          |
| Optimizer            | Adam                                                         |
| Recurrent layers     | Not used                                                     |
| Image encoder        | Not used                                                     |
| Action range         | Normalized $[-1, 1]^2$                                       |

### 3.2. Action-to-Actuator Mapping

The neural network does not directly output physical wheel torques or angular velocities. Instead, it generates normalized actions that are clipped to  $[-1, 1]$  and then linearly mapped to the actuator control range defined in the MuJoCo model. This mapping is implemented as

$$u_{i,t}^{\text{phys}} = u_i^{\min} + \frac{a_{i,t} + 1}{2} (u_i^{\max} - u_i^{\min}), \quad i \in \{L, R\}. \quad (40)$$

where the following hold:

- $u_{i,t}^{\text{phys}}$ —physical actuator command applied to wheel  $i$ ;
- $a_{i,t}$ —normalized policy output for wheel  $i$ ;
- $u_i^{\min}$ —lower actuator-control limit for wheel  $i$ ;
- $u_i^{\max}$ —upper actuator-control limit for wheel  $i$ ;
- $i \in \{L, R\}$ —index denoting the left or right wheel actuator.

For the simulated robot, both wheel actuators use the same control interval:

$$u_L^{\min} = u_R^{\min} = -10, \quad u_L^{\max} = u_R^{\max} = 10. \quad (41)$$

where the following hold:

- $u_L^{\min}$ —lower control limit of the left wheel actuator;
- $u_R^{\min}$ —lower control limit of the right wheel actuator;
- $u_L^{\max}$ —upper control limit of the left wheel actuator;
- $u_R^{\max}$ —upper control limit of the right wheel actuator.

Thus, for the present actuator range, the mapping is equivalent to  $u_{i,t}^{\text{phys}} = 10a_{i,t}$ . This formulation keeps the reinforcement-learning action space normalized while ensuring that all commands applied to the MuJoCo model remain within the physical actuator limits.

### 3.3. Policy Optimization

PPO was selected as the main learning algorithm because it limits excessively large policy updates through a clipped surrogate objective. This property is useful in the considered task, where unstable updates may lead to abrupt changes in wheel commands and unsafe behavior near the leader. The PPO objective is written as

$$L^{\text{CLIP}}(\theta) = \mathbb{E}_t [\min(\rho_t(\theta) \hat{A}_t, \text{clip}(\rho_t(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_t)]. \quad (42)$$

where the following hold:

- $L^{\text{CLIP}}(\theta)$ —clipped PPO surrogate objective;
- $\theta$ —parameters of the policy network;
- $\rho_t(\theta)$ —probability ratio between the updated and previous policies;
- $\hat{A}_t$ —estimated advantage at time step  $t$ ;
- $\epsilon$ —clipping parameter limiting the policy update;
- $\mathbb{E}_t[\cdot]$ —empirical expectation over sampled rollout transitions.

The probability ratio used in the PPO objective is defined as

$$\rho_t(\theta) = \frac{\pi_{\theta}(a_t|o_t)}{\pi_{\theta_{\text{old}}}(a_t|o_t)}. \quad (43)$$

where the following hold:

- $\rho_t(\theta)$ —policy probability ratio at time step  $t$ ;
- $\pi_{\theta}(a_t|o_t)$ —probability of action  $a_t$  under the updated policy;
- $\pi_{\theta_{\text{old}}}(a_t|o_t)$ —probability of action  $a_t$  under the previous policy;
- $o_t$ —observation available to the policy;
- $a_t$ —action selected by the policy.

A2C was used only as a reference actor–critic method. Its actor update is based on the advantage estimate

$$A_t = R_t - V_{\phi}(o_t). \quad (44)$$

where the following hold:

- $A_t$ —advantage estimate at time step  $t$ ;
- $R_t$ —discounted return;
- $V_{\phi}(o_t)$ —critic estimate of the value function;
- $\phi$ —parameters of the critic network.

In contrast to PPO, A2C does not constrain the policy update using a clipped probability ratio. This makes it a useful baseline for assessing whether the more conservative update mechanism of PPO improves training stability and safety in the considered leader–follower task.

### 3.4. Classical Distance–Bearing PID Baseline

The classical baseline used in this study is a distance–bearing PID controller, following the general feedback-control principle of PID regulation and classical tracking control for mobile robots [8,9]. It represents a closed-loop, error-feedback type of leader–follower tracking: at every control step, the follower computes its position error with respect to the moving reference point located behind the leader and then generates wheel commands intended to reduce this error. In contrast to the learned PPO and A2C policies, the PID controller does not optimize a reward function, does not adapt its gains during evaluation, and does not predict future leader motion. Its behavior is therefore reactive and is determined by the current and past tracking errors together with fixed controller gains.

The basic PID principle combines three terms. The proportional term reacts to the current error, the integral term accumulates past error and can reduce steady-state offset, and the derivative term reacts to the rate of error change, which can damp oscillatory responses. In the present distance–bearing formulation, two error signals are used: the distance error to the desired behind-leader point and the bearing error between the follower heading and the direction toward that point. Let

$$e_d(t) = \|p_{\text{ref}}(t) - p_F(t)\|_2, \quad e_{\alpha}(t) = \text{atan2}(\sin(\alpha(t) - \psi_F(t)), \cos(\alpha(t) - \psi_F(t))), \quad (45)$$

where  $\alpha(t) = \text{atan2}(y_{\text{ref}}(t) - y_F(t), x_{\text{ref}}(t) - x_F(t))$  is the direction from the follower to the desired reference point. The linear and angular commands are then computed as

$$v_{\text{PID}}(t) = K_{P,d}e_d(t) + K_{I,d} \sum_{\tau=0}^t e_d(\tau)\Delta t + K_{D,d} \frac{e_d(t) - e_d(t-1)}{\Delta t}, \quad (46)$$

$$\omega_{\text{PID}}(t) = K_{P,\alpha}e_\alpha(t) + K_{I,\alpha} \sum_{\tau=0}^t e_\alpha(\tau)\Delta t + K_{D,\alpha} \frac{e_\alpha(t) - e_\alpha(t-1)}{\Delta t}. \quad (47)$$

The commanded linear and angular velocities are converted to left and right differential-drive wheel commands using the standard inverse kinematic relation and are clipped to the same actuator limits as the learning-based controllers. In qualitative terms, this controller performs geometric point following: it attempts to drive the follower toward the instantaneous reference point behind the leader while aligning the follower with that direction. This makes it a useful interpretable non-learning baseline, but it also means that the controller is sensitive to gain tuning and may lag behind abrupt leader maneuvers because it responds only after the distance and bearing errors have appeared.

### 3.5. Reward Function

The reward function follows the implementation used in the environment and combines formation tracking, safety, control-effort regularization, and bearing-angle shaping. The reference point is located behind the leader along the opposite direction of leader motion:

$$p_{\text{ref}}(t) = p_L(t) - b\hat{d}_L(t), \quad b = 1.0 \text{ m}. \quad (48)$$

where the following hold:

- $p_{\text{ref}}(t)$ —desired reference position behind the leader;
- $p_L(t)$ —leader position at time step  $t$ ;
- $b$ —desired behind-leader offset;
- $\hat{d}_L(t)$ —unit vector representing the instantaneous direction of leader motion.

The distance to this point is

$$d_{\text{behind}}(t) = \|p_F(t) - p_{\text{ref}}(t)\|_2. \quad (49)$$

where the following hold:

- $d_{\text{behind}}(t)$ —distance between the follower and the reference point behind the leader;
- $p_F(t)$ —follower position at time step  $t$ ;
- $p_{\text{ref}}(t)$ —desired reference position behind the leader;
- $\|\cdot\|_2$ —Euclidean norm.

The distance used for safety evaluation is the direct leader–follower distance:

$$d_L(t) = \|p_F(t) - p_L(t)\|_2. \quad (50)$$

where the following hold:

- $d_L(t)$ —direct distance between the follower and the leader;
- $p_F(t)$ —follower position at time step  $t$ ;
- $p_L(t)$ —leader position at time step  $t$ ;
- $\|\cdot\|_2$ —Euclidean norm.

The reward at time step  $t$  is defined as

$$r_t = -w_f d_{\text{behind}}(t) - w_c \max(0, d_{\text{safe}} - d_L(t)) - w_a \sum_{i \in \{L,R\}} a_{i,t}^2 - w_\theta |\alpha_t|. \quad (51)$$

where the following hold:

- $r_t$ —reward value at time step  $t$ ;
- $w_f$ —weight of the formation-tracking penalty;
- $d_{\text{behind}}(t)$ —distance from the follower to the desired reference point behind the leader;
- $w_c$ —weight of the safety penalty;
- $d_{\text{safe}}$ —minimum admissible leader–follower distance;
- $d_L(t)$ —direct leader–follower distance;
- $w_a$ —weight of the control-effort regularization penalty;
- $a_{i,t}$ —normalized action applied to wheel actuator  $i$ ;
- $w_\theta$ —weight of the bearing-angle shaping term;
- $\alpha_t$ —relative bearing angle from the follower heading to the leader.

The first term penalizes deviation from the desired position behind the leader. The second term is activated only when the follower enters the unsafe region around the leader. The third term discourages excessive wheel-command magnitudes and should therefore be interpreted as control-effort regularization rather than a direct smoothness penalty. The final term provides an additional bearing-angle shaping signal. The nominal safety threshold is

$$d_{\text{safe}} = 0.5 \text{ m.} \quad (52)$$

where the following hold:

- $d_{\text{safe}}$ —minimum admissible distance between the follower and the leader.

The reward variants R1–R7 are treated as a controlled reward-weight sensitivity study. In this interpretation, the variants are not independent manual tunings, but systematic modifications of individual reward components. R1 is the baseline configuration, R2 and R3 increase the safety penalty, R4 increases the formation-tracking weight, R5 and R6 modify the control-effort regularization term, and R7 increases the bearing-angle shaping weight. The corresponding reward-weight variants are listed in Table 4.

**Table 4.** Reward-function variants used in the sensitivity study.

| Variant | $w_f$ | $w_c$ | $w_a$ | $w_\theta$ | Interpretation                         |
|---------|-------|-------|-------|------------|----------------------------------------|
| R1      | 2.0   | 12.0  | 0.30  | 0.01       | Baseline configuration                 |
| R2      | 2.0   | 16.0  | 0.30  | 0.01       | Increased safety penalty               |
| R3      | 2.0   | 20.0  | 0.30  | 0.01       | Strong safety penalty                  |
| R4      | 3.0   | 12.0  | 0.30  | 0.01       | Stronger formation tracking            |
| R5      | 2.0   | 12.0  | 0.60  | 0.01       | Stronger control-effort regularization |
| R6      | 2.0   | 12.0  | 0.15  | 0.01       | Weaker control-effort regularization   |
| R7      | 2.0   | 12.0  | 0.30  | 0.03       | Stronger bearing-angle shaping         |

### 3.6. Training Setup

Training was performed in sixteen parallel MuJoCo environments. Each environment used a different leader trajectory, with path identifiers from 0 to 15. This setup exposes the policy to circular, elliptical, sinusoidal, spiral, polygonal, figure-eight, and Lissajous-like leader motions. The leader trajectory is imposed by the simulation, while the follower is controlled only through the learned wheel-command policy.

Each PPO model was trained for  $3 \times 10^6$  environment steps. The main training parameters are summarized in Table 5. Parameters that were not explicitly set in the training script were left at the default values of the Stable-Baselines3 implementation [31].

**Table 5.** PPO training configuration used in the experiments.

| Parameter                                  | Value                     |
|--------------------------------------------|---------------------------|
| Algorithm                                  | PPO                       |
| Policy                                     | MlpPolicy                 |
| Number of parallel environments            | 16                        |
| Training path identifiers                  | 0–15                      |
| Total time steps per reward variant        | $3 \times 10^6$           |
| Rollout length, $n\_steps$                 | 1024 per environment      |
| Rollout samples per update                 | $16 \times 1024 = 16,384$ |
| Batch size                                 | 512                       |
| Number of optimization epochs, $n\_epochs$ | 10                        |
| Learning rate                              | $3 \times 10^{-4}$        |
| Discount factor, $\gamma$                  | 0.99                      |
| GAE parameter, $\lambda$                   | 0.95                      |
| PPO clipping range                         | 0.20                      |
| Entropy coefficient                        | 0.0                       |
| Value-function coefficient                 | 0.5                       |
| Maximum gradient norm                      | 0.5                       |
| Advantage normalization                    | Enabled                   |
| Optimizer                                  | Adam                      |
| Maximum episode length                     | 1500 environment steps    |
| Safety threshold                           | 0.5 m                     |

The same observation vector, action space, and evaluation metrics were used when comparing PPO with A2C. This ensures that the comparison reflects differences between the optimization algorithms rather than differences in the underlying control task.

### 3.7. Evaluation Metrics

The evaluation uses metrics for formation accuracy, safety, dynamic synchronization, and actuator regularity. The basic formation-tracking error is as follows:

$$e_{\text{form}}(t) = d_{\text{behind}}(t) = \|p_F(t) - p_{\text{ref}}(t)\|_2. \quad (53)$$

where the following hold:

- $e_{\text{form}}(t)$ —instantaneous formation-tracking error;
- $d_{\text{behind}}(t)$ —distance between the follower and the reference point behind the leader;
- $p_F(t)$ —follower position;
- $p_{\text{ref}}(t)$ —desired reference position behind the leader;
- $\|\cdot\|_2$ —Euclidean norm.

For an episode of length  $T$ , the mean and maximum formation errors are computed as

$$\bar{e}_{\text{form}} = \frac{1}{T} \sum_{t=1}^T e_{\text{form}}(t), \quad e_{\text{form}}^{\max} = \max_t e_{\text{form}}(t). \quad (54)$$

where the following hold:

- $\bar{e}_{\text{form}}$ —mean formation error over an episode;
- $e_{\text{form}}^{\max}$ —maximum formation error over an episode;
- $T$ —episode length;
- $e_{\text{form}}(t)$ —instantaneous formation-tracking error.

Safety is evaluated using the direct leader–follower distance  $d_L(t)$ . The safety-violation rate is defined as

$$V_{\text{safe}} = \frac{1}{T} \sum_{t=1}^T \mathbb{I}(d_L(t) < d_{\text{safe}}), \quad (55)$$

where the following hold:

- $V_{\text{safe}}$ —safety-violation rate;
- $T$ —episode length;
- $\mathbb{I}(\cdot)$ —indicator function;
- $d_L(t)$ —direct leader–follower distance;
- $d_{\text{safe}}$ —minimum admissible leader–follower distance.

In tables, this quantity is reported as a percentage. The cumulative severity of intrusions into the unsafe zone is as follows:

$$S_{\text{unsafe}} = \sum_{t=1}^T \max(0, d_{\text{safe}} - d_L(t)). \quad (56)$$

where the following hold:

- $S_{\text{unsafe}}$ —cumulative safety-violation severity,
- $T$ —episode length,
- $d_{\text{safe}}$ —minimum admissible leader–follower distance,
- $d_L(t)$ —direct leader–follower distance.

Velocity synchronization is measured using the absolute difference between the follower forward velocity and the leader speed:

$$E_v = \frac{1}{T} \sum_{t=1}^T |v_F(t) - v_L(t)|. \quad (57)$$

where the following hold:

- $E_v$ —mean absolute velocity-tracking error;
- $T$ —episode length;
- $v_F(t)$ —follower forward velocity;
- $v_L(t)$ —leader speed.

Control smoothness is evaluated at the actuator-command level using the change in normalized actions between consecutive time steps:

$$S_u = \frac{1}{T-1} \sum_{t=2}^T \|a_t - a_{t-1}\|_2. \quad (58)$$

where the following hold:

- $S_u$ —control-smoothness metric;
- $T$ —episode length;
- $a_t$ —normalized action vector at time step  $t$ ;
- $a_{t-1}$ —normalized action vector at the previous time step;
- $\|\cdot\|_2$ —Euclidean norm.

Here, lower  $E_v$  and  $S_u$  indicate better velocity matching and smoother wheel-command generation, respectively.

For additional analysis of abrupt leader maneuvers, the recovery time after a sudden change of leader direction can be defined as

$$T_{\text{rec}} = \min\{t : e_{\text{form}}(t) < e_{\text{thr}} \text{ for } N \text{ consecutive steps}\}. \quad (59)$$

where the following hold:

- $T_{\text{rec}}$ —recovery time after a sudden leader-direction change;
- $e_{\text{form}}(t)$ —instantaneous formation-tracking error;
- $e_{\text{thr}}$ —error threshold defining recovered formation tracking;
- $N$ —number of consecutive steps required below the threshold.

This metric is used in the generalization scenario to assess recovery after sharp turns and zigzag-like motion. Overall, the metrics separate tracking, safety, synchronization, and actuator regularity instead of relying only on cumulative reward.

## 4. Experiments and Results

This section presents the experimental evaluation of the proposed reinforcement-learning-based leader–follower controller. In contrast to the preliminary evaluation, which focused mainly on training curves and selected trajectory visualizations, the revised protocol provides a broader and more systematic assessment of the learned policies. The experiments include nominal testing, reward-weight sensitivity analysis, comparison with a classical distance–bearing PID baseline, robustness evaluation under observation and actuation disturbances, observation-space ablation, and generalization testing on an unseen maneuver sequence referred to as Scenario 17.

The evaluation was designed to answer four main questions. First, the reward-weight sensitivity analysis investigates how individual reward components affect safety, formation accuracy, velocity synchronization, and control smoothness. Second, the nominal comparison evaluates PPO against A2C and the classical PID baseline under the same observation and action spaces. Third, the robustness tests determine whether the trained policy remains reliable when observation noise, action delay, and wheel-slip-like actuation uncertainty are introduced at test time. Finally, Scenario 17 evaluates whether the learned controller can generalize to a trajectory containing abrupt leader maneuvers that differ from the nominal training paths.

The results are reported using task-level metrics rather than cumulative reward alone. The main evaluation criteria include collision occurrence, safety-violation rate, safety-violation severity, minimum leader–follower distance, mean and maximum formation error, velocity-tracking error, control smoothness, control jerk, recovery time, and success rate. Unless stated otherwise, the reported values are aggregated over multiple evaluation episodes and presented as mean values with standard deviations.

### 4.1. Experimental Protocol

The experimental evaluation used a separate test pipeline built around the trained policies and the MuJoCo leader–follower environment. Four test suites were considered: nominal evaluation on trajectories 0–15, robustness tests with disturbances introduced only at evaluation time, observation-space ablation, and Scenario 17 as an unseen composite trajectory.

In the nominal evaluation, each controller was tested on the standard trajectory set using 30 episodes per condition and a maximum episode length of 1500 steps. Scenario 17 uses a 6000-step horizon and a nominal leader speed of 1.0 m/s. PPO, A2C, and the distance–bearing PID baseline all use the same compact observation vector and normalized two-dimensional action space.

Unless stated otherwise, reported standard deviations quantify episode-level variability, including differences between trajectories and initial conditions. They should not be interpreted as variability across independently trained random seeds. Robustness disturbances were introduced only during testing and include Gaussian observation noise, action

delay, wheel-slip-like action scaling, and their combined form. The experimental test suites are summarized in Table 6.

**Table 6.** Overview of the experimental test suites used in the evaluation.

| Test Suite  | Objective                                               | Output                                                                                                           |
|-------------|---------------------------------------------------------|------------------------------------------------------------------------------------------------------------------|
| Nominal     | Standard leader trajectories.                           | Reward sensitivity, PPO/A2C/PID comparison.                                                                      |
| Robustness  | Test-time noise, delay, and wheel-slip-like scaling.    | Safety, error, smoothness, and success under disturbances.                                                       |
| Ablation    | Masked observation components.                          | Performance drop after removing leader speed, leader direction, follower velocity, or using distance–angle only. |
| Scenario 17 | Unseen composite path with smooth and abrupt maneuvers. | Trajectory plots and segment-level formation, safety, and recovery metrics.                                      |

The ablation study masks selected observation entries while keeping the trained policy unchanged. The tested variants remove leader speed, leader direction, follower velocity, or retain only distance and bearing. Scenario 17 then evaluates generalization on six maneuver segments, including a sharp turn and zigzag motion, which require rapid correction of bearing and formation distance. The following subsections report reward-weight sensitivity, baseline comparisons, robustness, ablation, and Scenario 17 results.

#### 4.2. Reward-Weight Sensitivity Analysis

The first experiment evaluates how reward-weight selection affects learning and final leader–follower behavior. Seven PPO variants, R1–R7, modify the weights of formation tracking, safety-distance preservation, control-effort regularization, and bearing-angle shaping, while all other experimental settings remain unchanged. R1 is the baseline, R2–R3 emphasize safety, R4 emphasizes tracking, R5–R6 modify control-effort regularization, and R7 strengthens bearing-angle shaping.

Figure 2 shows the smoothed mean episode reward during training. Since the reward is dominated by penalty terms, values closer to zero indicate better optimization. All variants improve rapidly at the beginning of training and then exhibit variant-specific convergence behavior.



**Figure 2.** Mean episode reward during PPO training for reward variants R1–R7. Higher values, i.e., values closer to zero, indicate better policy performance because the reward function is composed primarily of penalty terms.

The stability of the learning process was additionally assessed using the standard deviation of the reward signal. Let  $R_i$  denote the reward obtained in episode  $i$ , and let  $\bar{R}$  be the mean reward over  $N$  episodes. The reward variability is defined as

$$\sigma_R = \sqrt{\frac{1}{N} \sum_{i=1}^N (R_i - \bar{R})^2}. \quad (60)$$

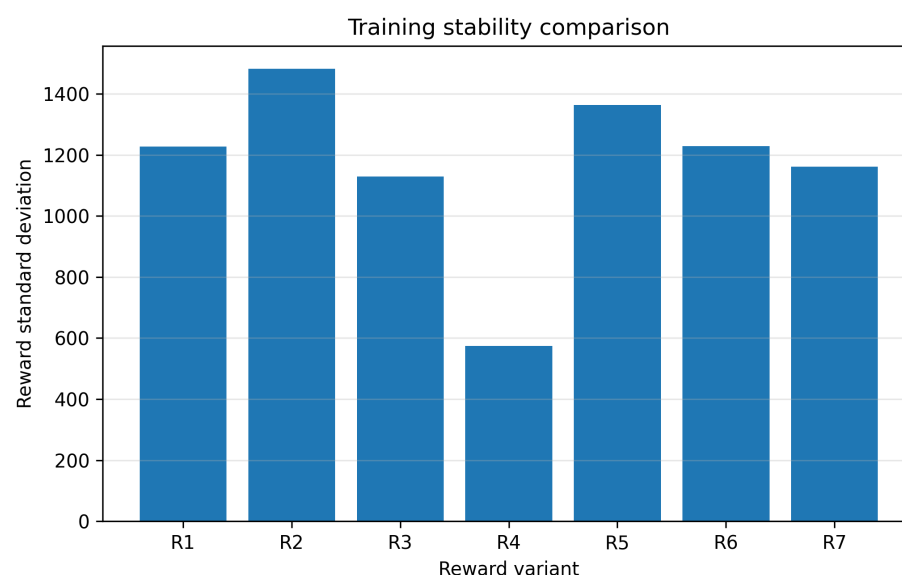
where the following hold:

- $\sigma_R$ —standard deviation of the reward signal;
- $R_i$ —reward obtained in episode  $i$ ;
- $\bar{R}$ —mean reward over the analyzed set of episodes;
- $N$ —number of episodes used to compute the statistic.

Lower  $\sigma_R$  indicates smaller reward oscillations. Figure 3 shows that R4 has the lowest reward variability, but this alone does not imply the best task behavior, because safety and formation accuracy must also be considered.

Table 7 summarizes task-level results over 10 evaluation episodes per variant. Because the reward is a weighted sum of penalties, less negative values indicate better optimization within a given formulation. However, raw reward values are not directly comparable across all variants and must be interpreted together with safety, tracking, synchronization, and smoothness metrics.

R6 achieved the highest final reward (−3351), the lowest mean formation error (0.331 m), and the lowest velocity-tracking error (0.087 m/s). However, it did not provide the lowest safety-violation rate. From the safety perspective, R2 and R3 both reached a safety-violation rate of 0.50% and avoided collisions, with R3 offering lower tracking and velocity errors than R2.



**Figure 3.** Reward variability during PPO training for reward variants R1–R7. Lower values indicate smaller fluctuations in the reward signal and therefore more stable learning dynamics.

R4 illustrates the limitation of using reward stability or control smoothness as the sole criterion for selecting a reward configuration. Although R4 produced the smoothest control signal, it also resulted in the highest safety-violation rate and collision count. Overall, R6 is preferable when average formation accuracy is dominant, whereas R3 provides the more conservative safety-oriented compromise. The formation-error evolution in

Figure 4 supports this interpretation by showing intermittent error peaks during difficult trajectory fragments.

**Table 7.** Reward-weight sensitivity analysis for PPO variants R1–R7 under nominal evaluation conditions. Values are reported as mean  $\pm$  standard deviation over 10 evaluation episodes.

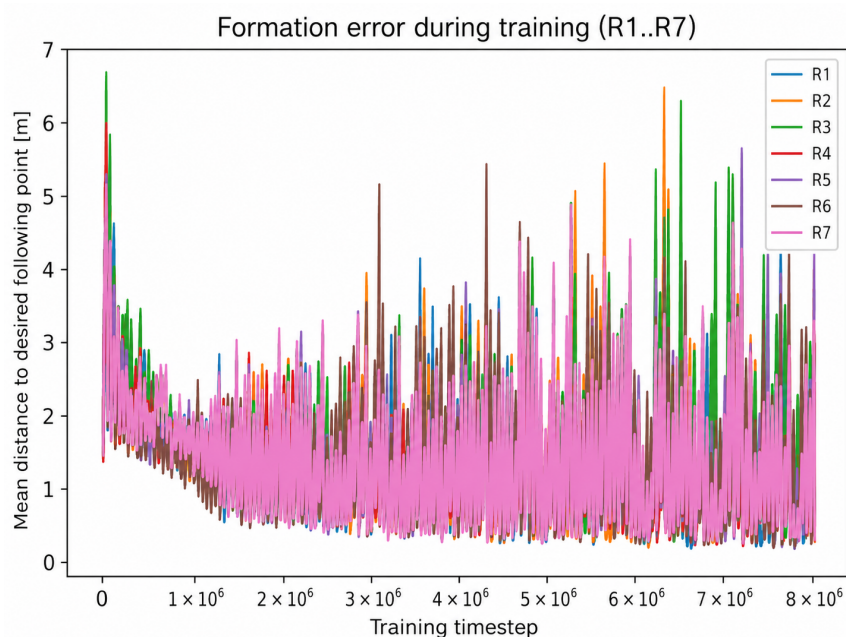
| Variant | Collision Count | Safety Violation Rate [%] | Mean Error [m]    | Max Error [m]     | Velocity Error [m/s] | Final Reward |
|---------|-----------------|---------------------------|-------------------|-------------------|----------------------|--------------|
| R1      | 0.40 $\pm$ 1.26 | 1.70 $\pm$ 4.20           | 0.684 $\pm$ 0.005 | 1.478 $\pm$ 0.002 | 0.114 $\pm$ 0.001    | −5729        |
| R2      | 0.00 $\pm$ 0.00 | 0.50 $\pm$ 1.10           | 0.612 $\pm$ 0.005 | 1.278 $\pm$ 0.001 | 0.114 $\pm$ 0.001    | −6708        |
| R3      | 0.00 $\pm$ 0.00 | 0.50 $\pm$ 1.10           | 0.527 $\pm$ 0.002 | 0.912 $\pm$ 0.000 | 0.095 $\pm$ 0.001    | −4496        |
| R4      | 0.50 $\pm$ 1.27 | 12.30 $\pm$ 37.30         | 0.564 $\pm$ 0.004 | 1.150 $\pm$ 0.018 | 0.116 $\pm$ 0.002    | −4551        |
| R5      | 0.10 $\pm$ 0.32 | 3.20 $\pm$ 6.00           | 0.602 $\pm$ 0.004 | 1.314 $\pm$ 0.007 | 0.118 $\pm$ 0.002    | −6264        |
| R6      | 0.00 $\pm$ 0.00 | 3.30 $\pm$ 8.90           | 0.331 $\pm$ 0.005 | 1.254 $\pm$ 0.006 | 0.087 $\pm$ 0.002    | −3351        |
| R7      | 0.10 $\pm$ 0.32 | 3.50 $\pm$ 9.40           | 0.646 $\pm$ 0.007 | 1.462 $\pm$ 0.004 | 0.119 $\pm$ 0.003    | −5234        |

Figure 5 shows that the follower speed gradually approaches the leader-speed range during training, indicating improved dynamic synchronization in addition to geometric tracking.

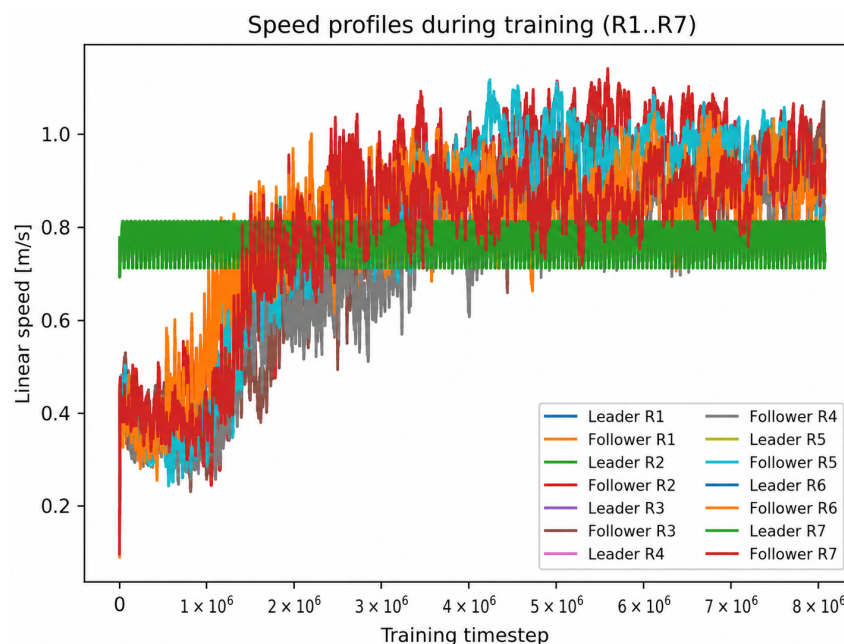
In summary, reward design strongly affects the safety–accuracy–regularity trade-off. R3 is the most conservative and balanced configuration, whereas R6 is preferable when minimizing average formation error is the dominant objective.

#### 4.3. Nominal Comparison with PPO, A2C, and PID Baseline

The nominal comparison evaluates PPO, A2C, and the distance–bearing PID controller under the standard leader trajectories. Based on the reward-weight sensitivity analysis, R3 was selected as the safety-oriented PPO/A2C configuration and R6 as the accuracy-oriented configuration. All controllers were evaluated using the same observation vector, action space, trajectory set, actuator limits, and task metrics.



**Figure 4.** Mean distance to the desired following point during PPO training for reward variants R1–R7. Lower values indicate more accurate formation tracking.



**Figure 5.** Leader and follower speed profiles during PPO training for reward variants R1–R7. The follower speed gradually approaches the leader-speed range, indicating improved dynamic synchronization during training.

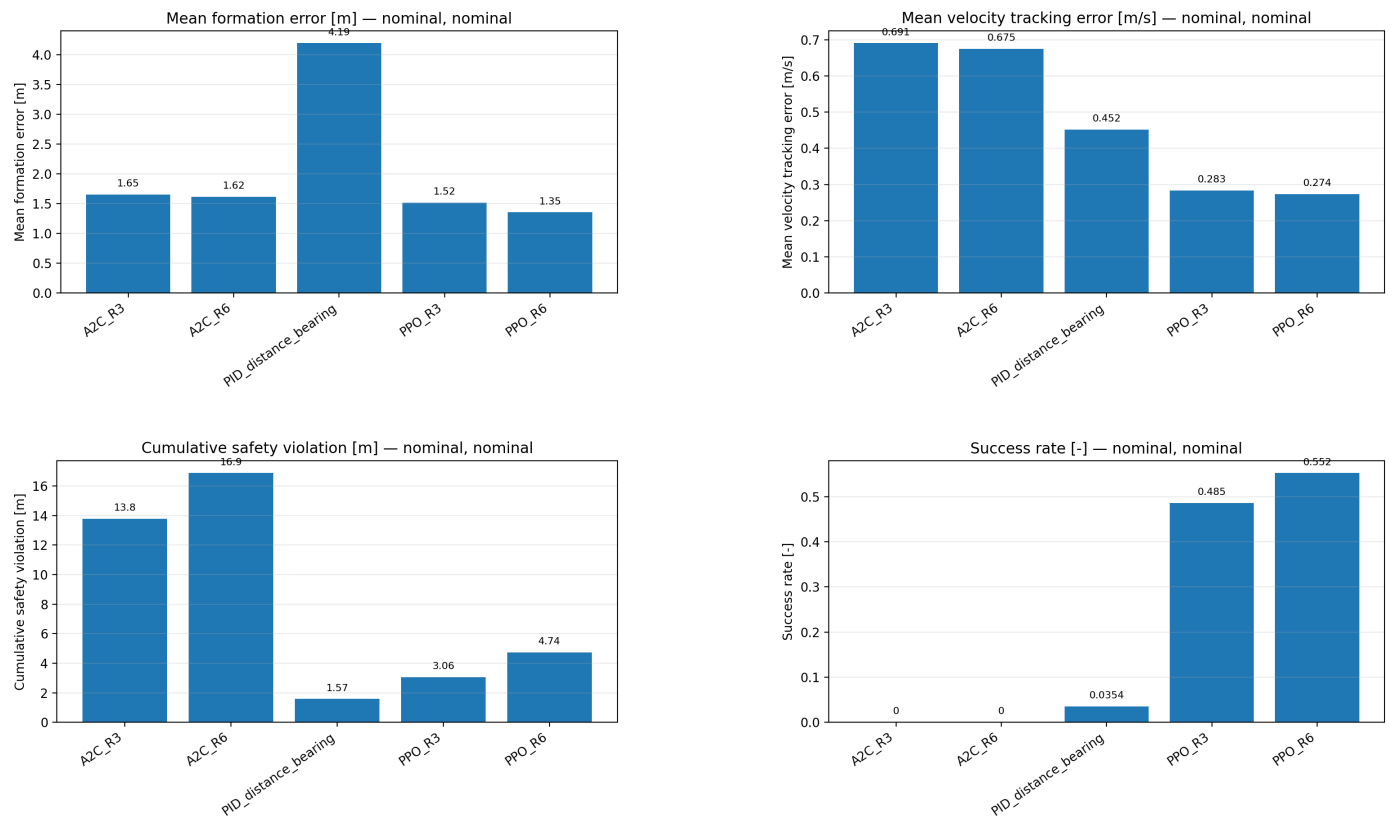
The comparison was performed over 480 nominal evaluation episodes for each controller. The reported metrics include collision rate, safety-violation rate, cumulative safety-violation severity, mean formation error, velocity-tracking error, and success rate. An episode is considered successful only if no collision occurs, the safety-violation rate remains below 1%, and the mean formation error remains below 0.75 m; therefore, success rate is a strict task-level measure rather than a reward-based score.

Table 8 shows that the PPO controllers provide the most balanced task-level performance under nominal conditions. PPO\_R6 obtains the lowest mean formation error, 1.353 m, and the lowest velocity-tracking error, 0.274 m/s. PPO\_R3 gives slightly weaker tracking accuracy, but lower cumulative safety-violation severity than PPO\_R6. The A2C variants reach comparable mean formation errors, but their collision and safety-violation metrics are much worse, leading to zero success rate. The PID baseline remains conservative with respect to safety-violation severity, but this comes at the cost of substantially larger formation error.

Figure 6 summarizes the same trend visually: PPO\_R6 is the most accurate, PPO\_R3 is the more conservative PPO variant, A2C fails the strict success criterion, and PID reduces violation severity but does not maintain the desired formation reliably.

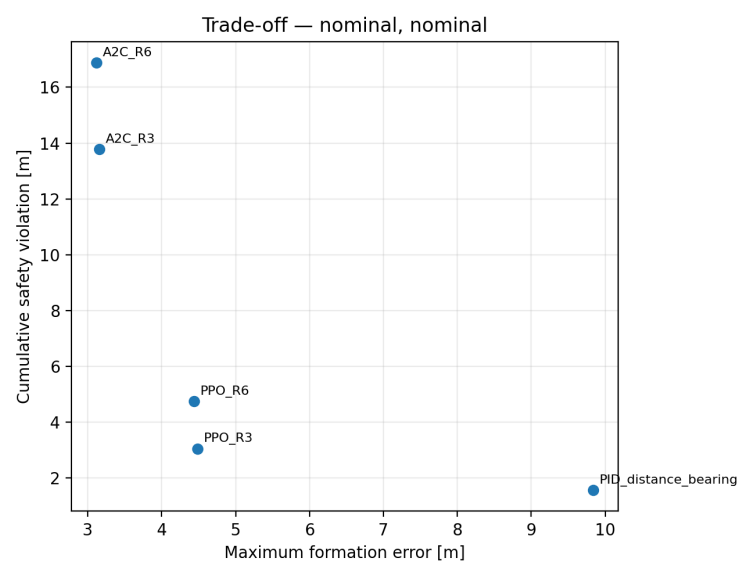
**Table 8.** Nominal comparison of PPO, A2C, and the distance–bearing PID baseline. Results are reported as mean  $\pm$  standard deviation over 480 evaluation episodes. Lower values are preferable for all metrics except success rate.

| Controller | Collision Rate [-] | Safety-Violation Rate [%] | Cumulative Safety-Violation Severity [m] | Mean Formation Error [m] | Velocity-Tracking Error [m/s] | Success Rate [-]  |
|------------|--------------------|---------------------------|------------------------------------------|--------------------------|-------------------------------|-------------------|
| A2C_R3     | $0.856 \pm 0.351$  | $11.801 \pm 16.635$       | $13.783 \pm 26.513$                      | $1.654 \pm 0.897$        | $0.691 \pm 0.462$             | $0.000 \pm 0.000$ |
| A2C_R6     | $0.883 \pm 0.321$  | $14.137 \pm 19.189$       | $16.875 \pm 29.640$                      | $1.616 \pm 0.848$        | $0.675 \pm 0.465$             | $0.000 \pm 0.000$ |
| PID        | $0.342 \pm 0.475$  | $0.782 \pm 2.048$         | $1.575 \pm 3.646$                        | $4.193 \pm 6.978$        | $0.452 \pm 0.341$             | $0.035 \pm 0.185$ |
| PPO_R3     | $0.456 \pm 0.499$  | $3.508 \pm 10.240$        | $3.056 \pm 8.700$                        | $1.518 \pm 4.168$        | $0.283 \pm 0.183$             | $0.485 \pm 0.500$ |
| PPO_R6     | $0.423 \pm 0.495$  | $4.251 \pm 10.724$        | $4.742 \pm 11.601$                       | $1.353 \pm 3.233$        | $0.274 \pm 0.236$             | $0.552 \pm 0.498$ |



**Figure 6.** Nominal evaluation of A2C, PPO, and the distance–bearing PID baseline. The bar plots summarize formation accuracy, velocity synchronization, cumulative safety-violation severity, and success rate.

The trade-off plot in Figure 7 illustrates the multi-objective nature of the task. The PID baseline occupies a low-violation but high-error region, the A2C policies occupy a high-violation region, and the PPO variants remain closer to the desirable compromise region. Thus, PPO\_R3 is preferable when safety-related criteria are prioritized, whereas PPO\_R6 is preferable when formation accuracy and dynamic synchronization dominate.



**Figure 7.** Trade-off between maximum formation error and cumulative safety-violation severity under nominal conditions. The preferred region is the lower-left part of the plot, where both tracking error and safety violation are low.

#### 4.4. Observation-Space Ablation Study

The observation-space ablation study evaluates how individual components of the compact six-dimensional observation vector affect the trained PPO policies. Selected observation entries were set to zero only during evaluation, while the environment state and task-level metrics were computed from the unmodified simulation state.

The full observation vector is defined as

$$o_t = [\alpha_t, \bar{d}_{L,t}, \bar{\omega}_{F,t}, \bar{v}_{F,t}, \bar{v}_{L,t}, \Delta\psi_t]^\top. \quad (61)$$

The Full 6D condition uses all components. The ablated variants remove leader speed  $\bar{v}_{L,t}$ , leader direction  $\Delta\psi_t$ , follower-velocity components  $\bar{v}_{F,t}$  and  $\bar{\omega}_{F,t}$ , or retain only relative distance and bearing. PPO\_R3 and PPO\_R6 were selected as the safety-oriented and accuracy-oriented policies; A2C is not emphasized because it remained non-competitive in the nominal comparison.

Table 9 shows that the full 6D observation is the most reliable condition. PPO\_R6 achieves the highest success rate and lowest mean formation error, whereas PPO\_R3 retains lower cumulative safety-violation severity. The leader-direction component  $\Delta\psi_t$  is the most safety-critical channel: masking it increases the safety-violation rate to 44.242% for PPO\_R3 and 29.399% for PPO\_R6.

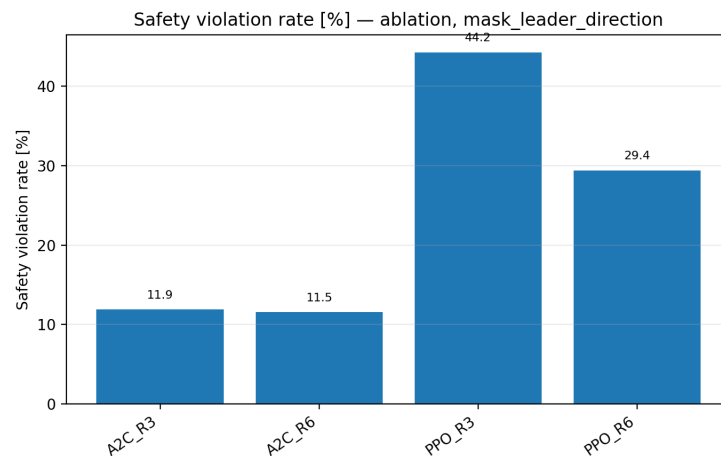
Removing follower velocity mainly degrades formation accuracy and synchronization. The mean formation error increases to 5.129 m for PPO\_R3 and 4.529 m for PPO\_R6, confirming the importance of proprioceptive velocity feedback. The main ablation effects are visualized in Figures 8–10.

**Table 9.** Observation-space ablation results for PPO\_R3 and PPO\_R6. Results are reported as the mean  $\pm$  standard deviation over 480 evaluation episodes per condition. Lower values are preferable for safety violation, severity, formation error, and velocity error, whereas higher values are preferable for success.

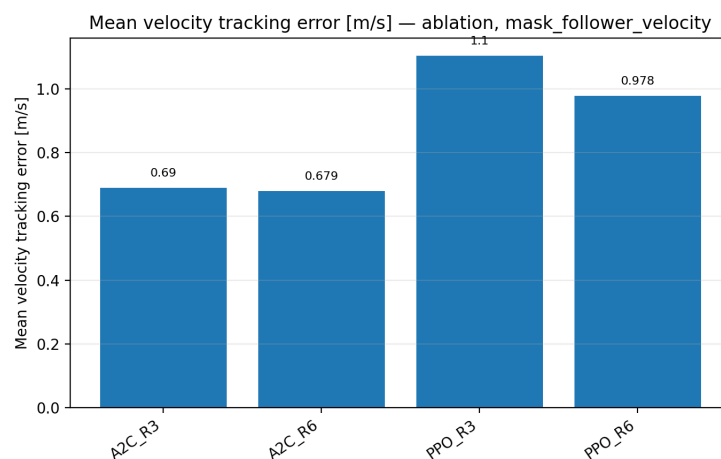
| Policy | Observation Condition | Safety Viol. [%]    | Severity [m]        | Mean Error [m]    | Velocity Error [m/s] | Success [-]       |
|--------|-----------------------|---------------------|---------------------|-------------------|----------------------|-------------------|
| PPO_R3 | Full 6D               | 3.508 $\pm$ 10.240  | 3.056 $\pm$ 8.700   | 1.518 $\pm$ 4.168 | 0.283 $\pm$ 0.183    | 0.485 $\pm$ 0.500 |
| PPO_R3 | No leader speed       | 0.837 $\pm$ 2.001   | 1.435 $\pm$ 3.947   | 1.812 $\pm$ 2.231 | 0.369 $\pm$ 0.231    | 0.000 $\pm$ 0.000 |
| PPO_R3 | No leader direction   | 44.242 $\pm$ 36.854 | 62.355 $\pm$ 57.285 | 4.479 $\pm$ 6.955 | 0.426 $\pm$ 0.238    | 0.000 $\pm$ 0.000 |
| PPO_R3 | No follower velocity  | 4.279 $\pm$ 11.177  | 4.956 $\pm$ 10.226  | 5.129 $\pm$ 5.570 | 1.104 $\pm$ 0.534    | 0.046 $\pm$ 0.209 |
| PPO_R3 | Distance–angle only   | 4.044 $\pm$ 9.791   | 6.062 $\pm$ 17.244  | 4.639 $\pm$ 2.936 | 1.296 $\pm$ 0.843    | 0.000 $\pm$ 0.000 |
| PPO_R6 | Full 6D               | 4.251 $\pm$ 10.724  | 4.742 $\pm$ 11.601  | 1.353 $\pm$ 3.233 | 0.274 $\pm$ 0.236    | 0.552 $\pm$ 0.498 |
| PPO_R6 | No leader speed       | 6.593 $\pm$ 12.966  | 9.554 $\pm$ 25.322  | 1.388 $\pm$ 1.030 | 0.379 $\pm$ 0.285    | 0.044 $\pm$ 0.205 |
| PPO_R6 | No leader direction   | 29.399 $\pm$ 30.130 | 28.785 $\pm$ 43.494 | 2.714 $\pm$ 4.205 | 0.388 $\pm$ 0.213    | 0.000 $\pm$ 0.000 |
| PPO_R6 | No follower velocity  | 4.118 $\pm$ 9.596   | 3.675 $\pm$ 9.413   | 4.529 $\pm$ 5.209 | 0.978 $\pm$ 0.570    | 0.006 $\pm$ 0.079 |
| PPO_R6 | Distance–angle only   | 5.428 $\pm$ 18.033  | 10.738 $\pm$ 41.560 | 7.005 $\pm$ 3.961 | 1.597 $\pm$ 0.864    | 0.000 $\pm$ 0.000 |

Leader speed has a weaker effect on mean formation error, but it contributes to the combined success criterion. The distance–angle-only condition confirms that relative bearing and distance alone are insufficient for this dynamic task, as both PPO variants obtain zero success rate.

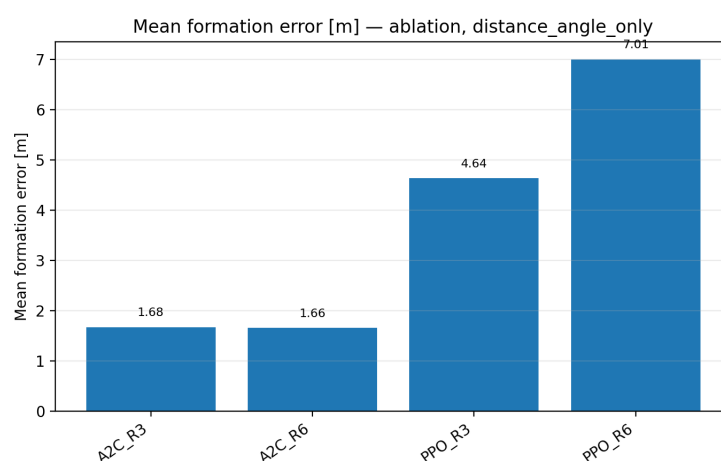
Overall, the compact observation vector is low-dimensional but not arbitrary: leader direction supports safety, follower velocity supports synchronization and accuracy, and leader speed helps satisfy the combined success criterion.



**Figure 8.** Effect of masking the leader-direction component on the safety-violation rate. Removing  $\Delta\psi_t$  substantially increases safety violations, which indicates that the relative direction of leader motion is critical for maintaining a safe leader–follower configuration.



**Figure 9.** Effect of masking follower-velocity components on velocity-tracking error. Removing proprioceptive velocity information increases the mismatch between follower and leader motion, confirming its importance for dynamic synchronization.



**Figure 10.** Formation error under the distance–angle-only observation condition. Retaining only relative distance and bearing is insufficient for reliable formation tracking, especially for the PPO policies that rely on velocity and leader-direction information.

#### 4.5. Robustness Under Observation and Actuation Disturbances

The robustness evaluation examines PPO\_R3 and PPO\_R6 under relaxed observation and actuation assumptions. The tests are motivated by sim-to-real and domain-randomization

studies, where observation uncertainty, dynamics mismatch, and actuator-effectiveness changes commonly degrade performance [26–28]. Disturbances were introduced only during evaluation, without retraining or fine-tuning, and the distance-bearing PID controller was retained as a classical reference.

The suite includes one nominal reference and seven disturbed conditions: low/high Gaussian observation noise, one-/three-step action delay, mild/strong wheel-slip-like action scaling, and a combined uncertainty condition. Each controller was evaluated over 480 episodes per condition. The disturbances introduced during robustness evaluation can be formalized as follows. Observation noise is modeled as an additive Gaussian perturbation applied to the compact observation vector:

$$\tilde{o}_t = o_t + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma_o^2 I). \quad (62)$$

where the following hold:

- $\tilde{o}_t$ —disturbed observation vector used during evaluation;
- $o_t$ —nominal observation vector returned by the environment;
- $\epsilon_t$ —Gaussian observation-noise vector;
- $\sigma_o$ —standard deviation of the observation noise;
- $I$ —identity matrix matching the observation dimension.

Actuation delay is modeled by executing an action selected in a previous control step:

$$a_t^{\text{exec}} = a_{t-k}. \quad (63)$$

where the following hold:

- $a_t^{\text{exec}}$ —action actually executed by the simulator at time step  $t$ ;
- $a_{t-k}$ —policy action generated  $k$  control steps earlier;
- $k$ —number of delayed control steps.

Wheel-slip-like actuation uncertainty is approximated by random scaling of the executed action:

$$a_t^{\text{exec}} = \lambda_t a_t, \quad \lambda_t \sim \mathcal{U}(\lambda_{\min}, \lambda_{\max}). \quad (64)$$

where the following hold:

- $a_t^{\text{exec}}$ —scaled action executed by the simulator;
- $a_t$ —nominal action generated by the policy;
- $\lambda_t$ —random action-scaling coefficient;
- $\mathcal{U}(\lambda_{\min}, \lambda_{\max})$ —uniform distribution over the scaling interval.

Table 10 reports the main robustness results for PPO\_R3 and PPO\_R6. To keep the table readable, three representative task-level metrics are shown: cumulative safety-violation severity, mean formation error, and success rate. These metrics summarize the main safety-accuracy trade-off under non-nominal observation and actuation conditions. Lower values are preferable for safety-violation severity and formation error, whereas higher values are preferable for success rate. Additional metrics, including collision rate, safety-violation rate, velocity-tracking error, control smoothness, and control jerk, were also computed and used in the interpretation.

Table 10 indicates that both PPO variants retain comparable aggregate quality under the tested disturbances, but the large standard deviations show that performance remains trajectory-dependent. The results should therefore be interpreted as an aggregate comparison over the evaluated trajectory set, not as evidence of uniform robustness in every episode.

**Table 10.** Robustness evaluation of PPO\_R3 and PPO\_R6 under observation and actuation disturbances. Results are reported as mean  $\pm$  standard deviation over 480 evaluation episodes per condition.

| Cond.       | R3 Cumulative Safety Violation Severity [m] | R3 Mean Formation Error [m] | R3 Success Rate   | R6 Cumulative Safety Violation Severity [m] | R6 Mean Formation Error [m] | R6 Success Rate   |
|-------------|---------------------------------------------|-----------------------------|-------------------|---------------------------------------------|-----------------------------|-------------------|
| Nominal     | 3.056 $\pm$ 8.700                           | 1.518 $\pm$ 4.168           | 0.485 $\pm$ 0.500 | 4.742 $\pm$ 11.601                          | 1.353 $\pm$ 3.233           | 0.552 $\pm$ 0.498 |
| Noise low   | 2.838 $\pm$ 8.278                           | 1.436 $\pm$ 3.955           | 0.487 $\pm$ 0.500 | 4.698 $\pm$ 11.441                          | 1.484 $\pm$ 3.523           | 0.556 $\pm$ 0.497 |
| Noise high  | 2.641 $\pm$ 7.740                           | 1.494 $\pm$ 4.143           | 0.475 $\pm$ 0.500 | 4.850 $\pm$ 11.806                          | 1.346 $\pm$ 3.179           | 0.562 $\pm$ 0.497 |
| Delay 1     | 2.665 $\pm$ 7.885                           | 1.501 $\pm$ 4.225           | 0.502 $\pm$ 0.501 | 4.149 $\pm$ 9.785                           | 1.205 $\pm$ 2.720           | 0.554 $\pm$ 0.498 |
| Delay 3     | 2.129 $\pm$ 7.258                           | 1.506 $\pm$ 4.063           | 0.563 $\pm$ 0.497 | 3.039 $\pm$ 6.328                           | 1.660 $\pm$ 2.858           | 0.475 $\pm$ 0.500 |
| Slip mild   | 2.347 $\pm$ 7.026                           | 1.461 $\pm$ 3.997           | 0.492 $\pm$ 0.500 | 4.511 $\pm$ 10.747                          | 1.291 $\pm$ 3.066           | 0.565 $\pm$ 0.496 |
| Slip strong | 1.442 $\pm$ 6.375                           | 1.516 $\pm$ 3.976           | 0.525 $\pm$ 0.500 | 4.533 $\pm$ 10.998                          | 1.182 $\pm$ 2.742           | 0.552 $\pm$ 0.498 |
| Combined    | 1.546 $\pm$ 6.557                           | 1.267 $\pm$ 3.449           | 0.527 $\pm$ 0.500 | 4.240 $\pm$ 9.336                           | 1.221 $\pm$ 2.708           | 0.569 $\pm$ 0.496 |

Table 10 reports the cumulative safety-violation severity, mean formation error, and success rate for PPO\_R3 and PPO\_R6. These metrics summarize the main safety–accuracy trade-off under non-nominal observation and actuation conditions.

PPO\_R3 generally maintains lower cumulative safety-violation severity than PPO\_R6; under combined uncertainty, the values are 1.546 m and 4.240 m, respectively. PPO\_R6, in turn, often preserves lower mean formation error, including 1.182 m under strong wheel-slip-like scaling and 1.221 m under combined uncertainty. Thus, the same safety–accuracy trade-off observed in nominal testing remains visible under disturbances.

Action delay and observation noise expose different weaknesses. A three-step delay increases the PPO\_R6 mean formation error to 1.660 m, while high observation noise increases control-smoothness values to 0.166 for PPO\_R3 and 0.194 for PPO\_R6. The PID baseline remains comparatively conservative but inaccurate, with mean formation error of approximately 4.15–4.75 m and near-zero success. Overall, the robustness tests support cautious conclusions: the PPO policies do not collapse under the evaluated disturbances, but no direct sim-to-real claim can be made without physical robot validation.

#### 4.6. Generalization Test Scenario

In order to evaluate the generalization capability of the trained policies, an independent testing scenario was designed. The purpose of this experiment is to verify whether the follower robot is able to maintain the desired formation when the leader executes a trajectory that differs from those used during the training phase.

The scenario consists of a sequence of motion primitives forming a closed trajectory executed by the leader robot. During the test the leader moves with a constant forward velocity and performs a series of maneuvers combining curvilinear motion, straight-line segments, and oscillatory movement patterns. Throughout the experiment the follower robot is required to continuously adapt its motion in order to maintain the desired relative position behind the leader.

As illustrated in Figure 11, the trajectory of the leader is composed of several motion primitives designed to introduce different types of motion patterns and dynamic conditions for the follower robot. The trajectory consists of several motion primitives listed below.

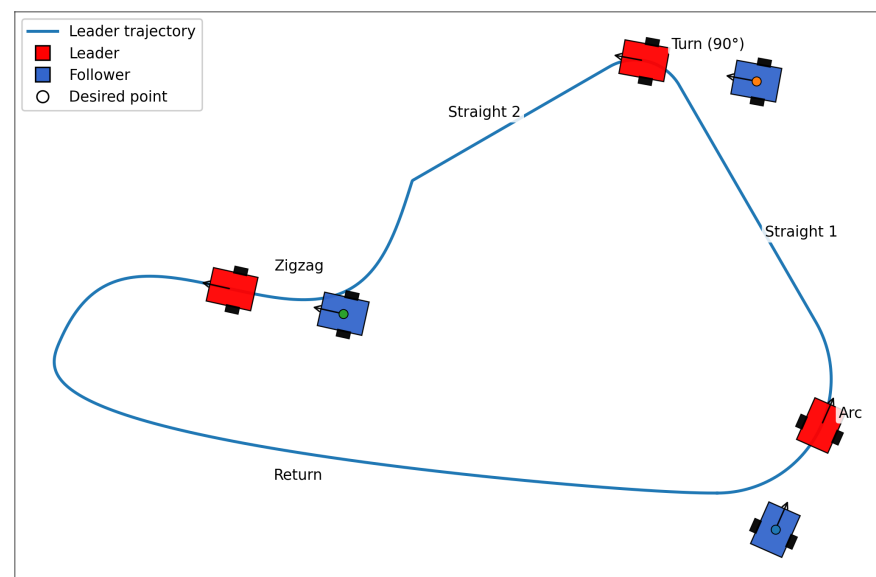
1. Circular arc—the leader follows a fragment of a circular trajectory corresponding to approximately 120° of rotation. This segment evaluates the ability of the follower to track smooth curvilinear motion while maintaining the desired formation geometry.
2. Straight segment—the leader moves along a straight-line trajectory with constant velocity, allowing the follower robot to stabilize its position relative to the desired reference point in the formation.

3. Sharp turn—the leader performs a turn of approximately  $90^\circ$  along a short circular arc with a smaller radius. This maneuver introduces a rapid change of heading that requires the follower to react promptly in order to maintain the desired relative configuration.
4. Second straight segment—another straight motion segment is executed after the turning maneuver. This part of the trajectory is used to evaluate the ability of the follower to regain a stable formation after a sudden change in the leader's heading.
5. Zigzag motion—the leader moves forward while simultaneously performing lateral oscillations described by a sinusoidal function. This segment introduces periodic lateral disturbances and provides a more challenging tracking task for the follower robot.
6. Return segment—finally, the trajectory smoothly returns to the initial position using a fifth-order polynomial interpolation that ensures continuity of both position and velocity profiles.

The complete trajectory forms a closed motion cycle, which allows the experiment to be repeated for multiple episodes without introducing discontinuities in the leader motion. Such structure ensures consistent testing conditions across different trained models.

During each evaluation episode the follower robot starts from a randomized initial pose. This procedure prevents the learned policies from overfitting to a specific initial configuration and enables a more reliable assessment of their generalization capability.

The performance of the trained models is evaluated using the average distance between the follower robot and the desired reference point located behind the leader, as well as the distance between the follower and the leader itself. These metrics provide quantitative indicators of both formation accuracy and operational safety during the execution of the trajectory.



**Figure 11.** Leader trajectory used in the generalization experiment (Scenario 17).

#### 4.7. Comparison of PPO and A2C

An auxiliary experiment compared proximal policy optimization (PPO) with advantage actor–critic (A2C) in the considered leader–follower task. The goal was not to provide an exhaustive reinforcement-learning benchmark, but to verify whether the choice of PPO as the primary optimization algorithm is justified. Both algorithms were trained and evaluated using the same environment, observation and action spaces, and baseline reward variant R1.

Both methods belong to the actor–critic family and can handle continuous-control problems. PPO differs mainly through its clipped policy-update mechanism, which is intended to limit excessively large policy changes. This property is relevant here because unstable policy updates may cause oscillatory motion, degraded formation tracking, or more frequent safety-zone violations.

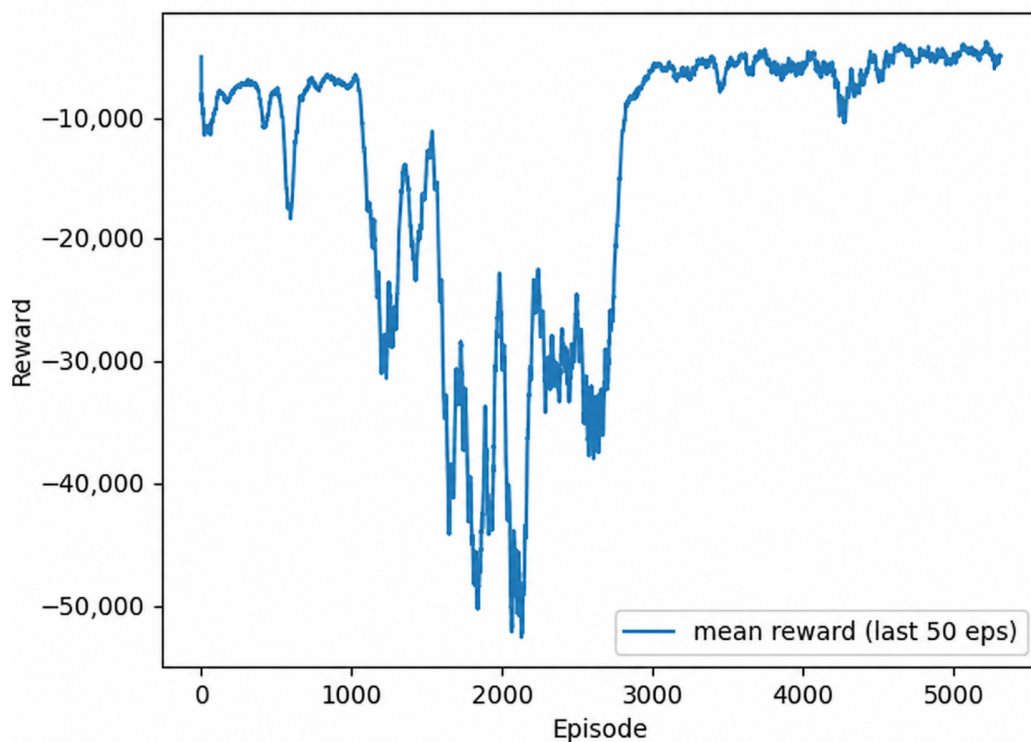
The comparison used the same task-oriented indicators as the reward-weight sensitivity analysis, including collision count, safety violation rate, mean formation error, velocity-tracking error, control smoothness, and reward variability. The A2C training curves are shown in Figures 12–16.

The reward curve indicates substantial variability during the intermediate phase of A2C training, followed by recovery and gradual stabilization at a higher reward level. This suggests that A2C can discover a useful control strategy, but its convergence path is less regular.

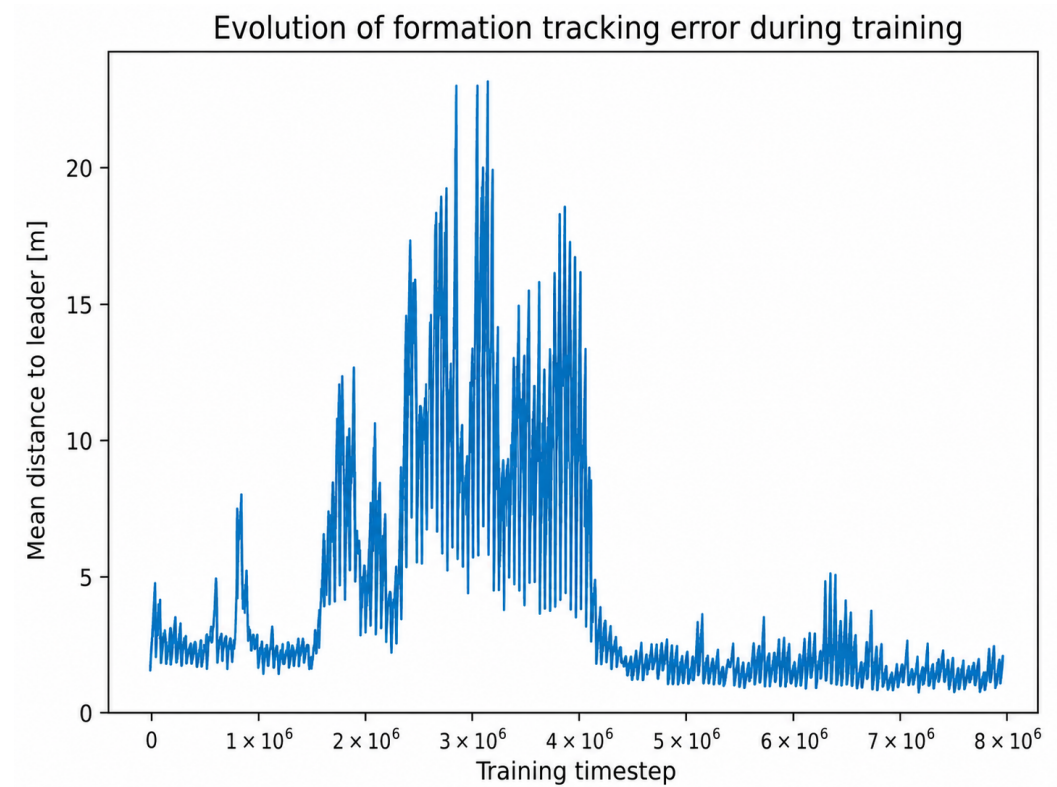
Figure 13 shows that the formation error decreases in the final training phase, although pronounced intermediate oscillations remain visible. This behavior is undesirable in a safety-sensitive control task, where stable learning dynamics are important.

The control-smoothness metric improves over training, but local fluctuations indicate that abrupt wheel-command changes are still present in parts of the learning process.

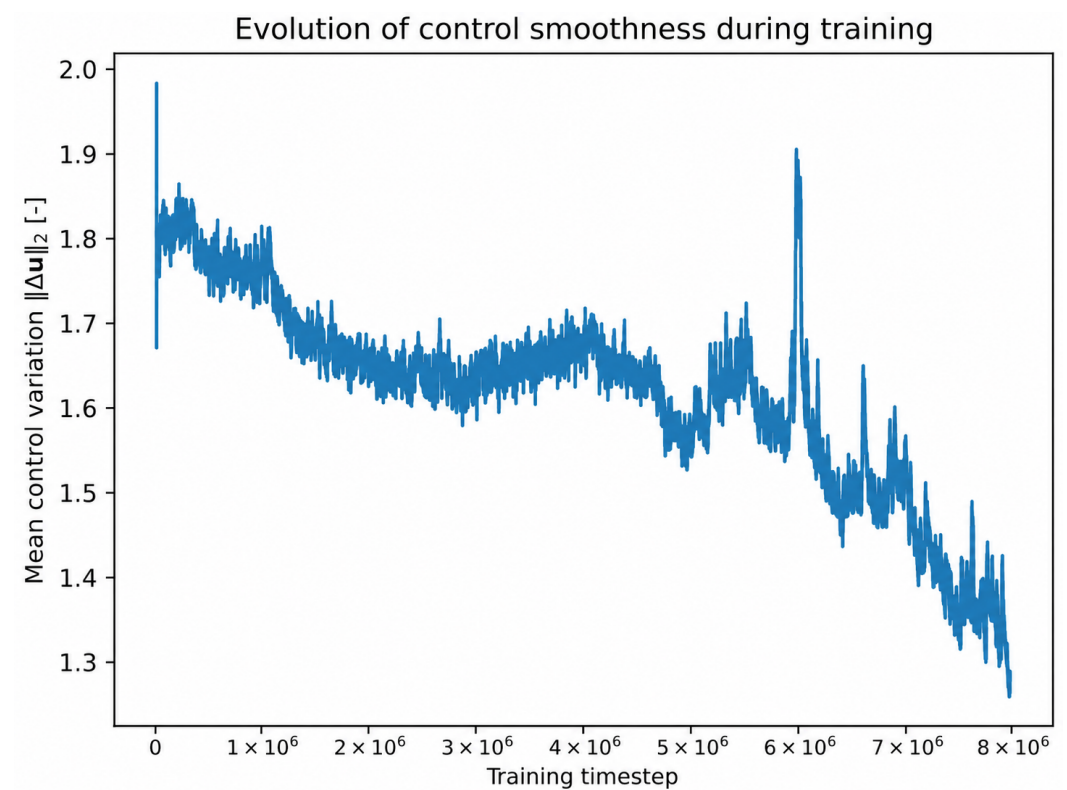
The velocity plots show gradual improvement in follower–leader synchronization, but also confirm substantial fluctuations before the final training phase. Overall, A2C improves reward, formation maintenance, smoothness, and velocity matching over time, yet the learning process is noticeably oscillatory. PPO was therefore adopted in the main experiments because its clipped update provides a more conservative optimization mechanism, which is better aligned with the safety-sensitive and trajectory-changing nature of the task. A2C remains a meaningful baseline, but the comparison supports PPO as the primary algorithm used in the study.



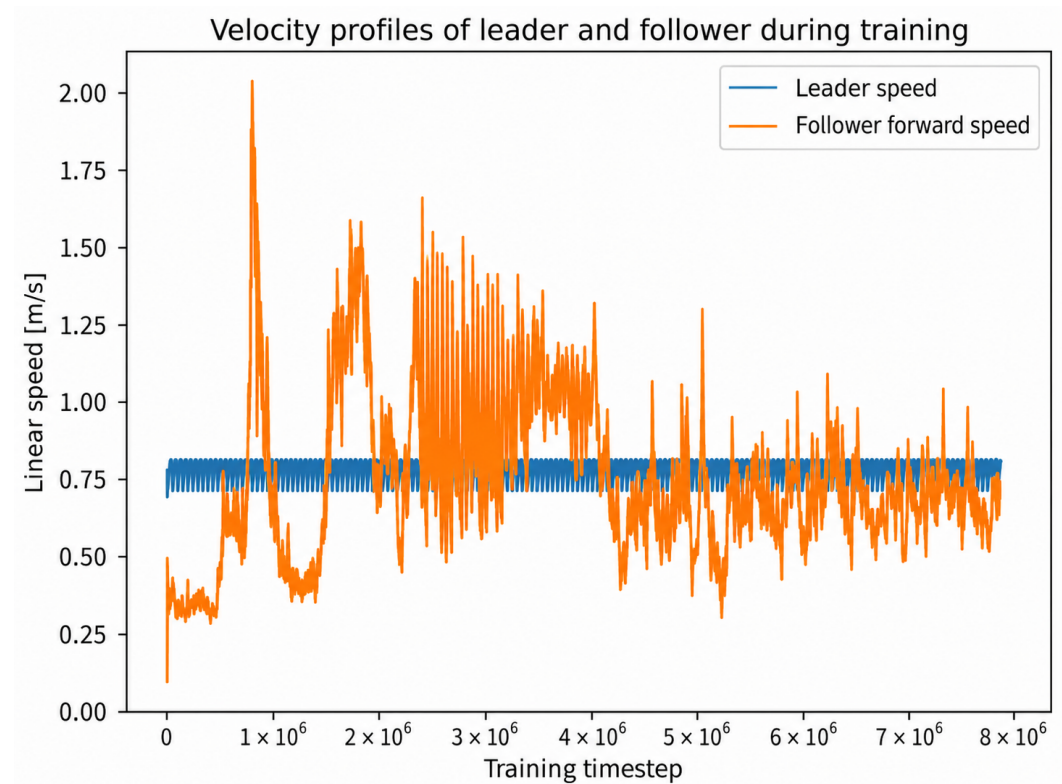
**Figure 12.** Mean episode reward during A2C training for the baseline reward variant R1. The plotted value represents the moving average over the last 50 episodes.



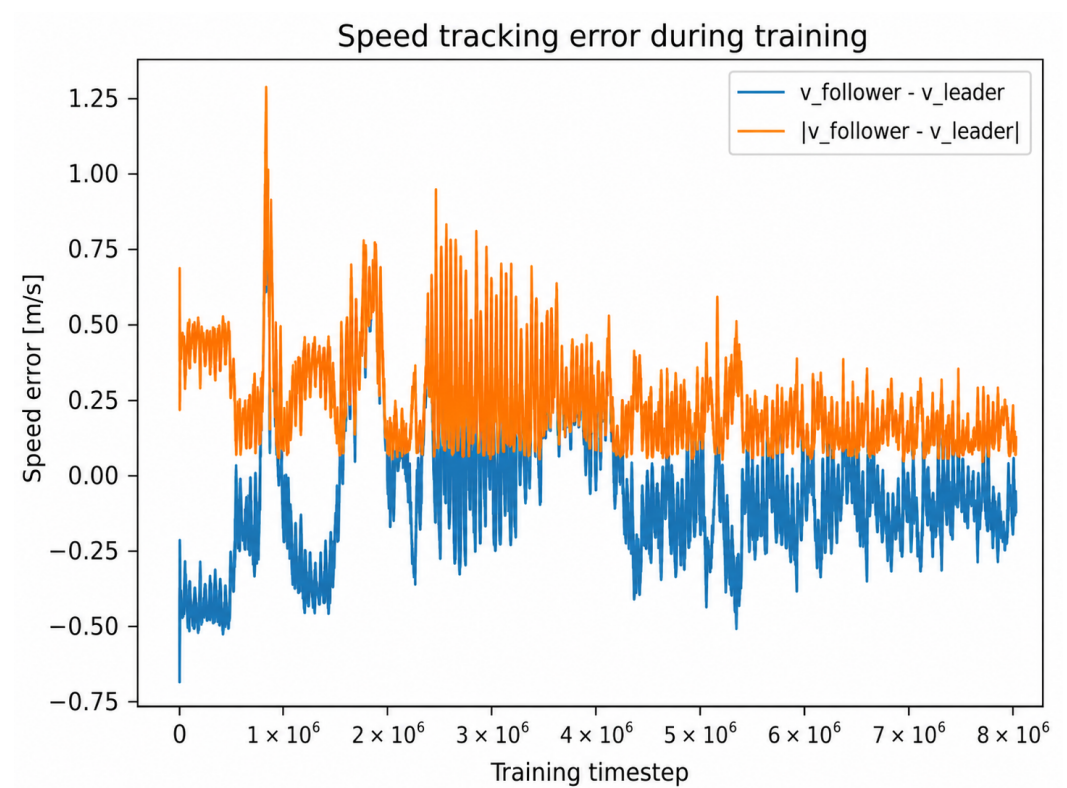
**Figure 13.** Evolution of formation tracking error during A2C training for the baseline reward variant R1.



**Figure 14.** Evolution of control smoothness during A2C training for the baseline reward variant R1. Lower values correspond to smoother control signals.



**Figure 15.** Velocity profiles of the leader and follower during A2C training for the baseline reward variant R1.



**Figure 16.** Speed tracking error during A2C training for the baseline reward variant R1. The plot shows the discrepancy between follower and leader velocity over the course of training.

## 5. Conclusions

This paper addressed leader–follower navigation for a wheeled mobile robot under dynamic leader maneuvers. A PPO-based follower was trained in a MuJoCo environment to maintain a desired position behind the leader while penalizing unsafe proximity.

The main contribution is a modular reward formulation combining formation tracking, collision avoidance, control-effort regularization, and bearing-angle shaping. The reward-weight study showed that no variant dominates all criteria: R3 provides a more conservative safety-oriented trade-off, whereas R6 improves average formation accuracy and velocity synchronization.

In the simulated tests, PPO achieved a more favorable safety–tracking trade-off than the considered A2C variants and the classical distance–bearing PID baseline. However, collision-free behavior was not guaranteed in all episodes, so the results should be interpreted as empirical simulation evidence rather than proof of deployment-ready safety.

The ablation and robustness studies indicate that the learned behavior depends strongly on the compact observation structure, especially the relative direction of leader motion, and that test-time disturbances produce trajectory-dependent variability. Future work will address multi-robot formations, sample efficiency, across-seed statistical validation, and sim-to-real transfer through systematic domain randomization and real-robot experiments [26–28].

**Author Contributions:** Conceptualization, S.R. and J.B.; methodology, S.R. and J.B.; software, J.B.; validation, S.R. and J.B.; formal analysis, S.R. and J.B.; investigation, S.R. and J.B.; resources, S.R. and J.B.; data curation, J.B.; writing—original draft preparation, S.R. and J.B.; writing—review and editing, S.R. and J.B.; visualization, J.B.; supervision, S.R.; project administration, S.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Ministry of Science and Higher Education in Poland at the Bialystok University of Technology under research subsidy No. WZ/WE-IA/5/2026.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Data and code supporting the findings of this study are available from the corresponding author upon reasonable request.

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

1. Oh, K.-K.; Park, M.-C.; Ahn, H.-S. A survey of multi-agent formation control. *Automatica* **2015**, *53*, 424–440. [\[CrossRef\]](#)
2. Olfati-Saber, R.; Fax, J.A.; Murray, R.M. Consensus and cooperation in networked multi-agent systems. *Proc. IEEE* **2007**, *95*, 215–233. [\[CrossRef\]](#)
3. Desai, J.P.; Ostrowski, J.; Kumar, V. Modeling and control of formations of nonholonomic mobile robots. *IEEE Trans. Robot. Autom.* **2001**, *17*, 905–908. [\[CrossRef\]](#)
4. Fax, J.A.; Murray, R.M. Information flow and cooperative control of vehicle formations. *IEEE Trans. Autom. Control* **2004**, *49*, 1465–1476. [\[CrossRef\]](#)
5. Ren, W.; Beard, R.W. *Distributed Consensus in Multi-Vehicle Cooperative Control*; Springer: London, UK, 2008.
6. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [\[CrossRef\]](#)
7. Ames, A.D.; Xu, X.; Grizzle, J.W.; Tabuada, P. Control barrier function based quadratic programs for safety critical systems. *IEEE Trans. Autom. Control* **2017**, *62*, 3861–3876.
8. Kanayama, Y.; Kimura, Y.; Miyazaki, F.; Noguchi, T. A stable tracking control method for an autonomous mobile robot. In Proceedings of the IEEE International Conference on Robotics and Automation, Cincinnati, OH, USA, 13–18 May 1990; pp. 384–389.
9. Åström, K.J.; Hägglund, T. *PID Controllers: Theory, Design, and Tuning*, 2nd ed.; ISA: Research Triangle Park, NC, USA, 1995.

10. Wang, J.; Liu, J.; Chen, P.; Chen, K. Distributed observer-based formation control for leader-follower multi-UAVs with collision avoidance. *IEEE Trans. Veh. Technol.* **2025**, *74*, 11951–11962. [[CrossRef](#)]
11. Tang, M.; Feng, C.; Quek, T.Q.S. Decentralized semantic communication and cooperative tracking control for a UAV swarm over wireless MIMO fading channels. *IEEE Trans. Veh. Technol.* **2025**, *75*, 3354–3359. . [[CrossRef](#)]
12. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018.
13. Kober, J.; Bagnell, J.A.; Peters, J. Reinforcement learning in robotics: A survey. *Int. J. Robot. Res.* **2013**, *32*, 1238–1274. [[CrossRef](#)]
14. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. In Proceedings of the International Conference on Learning Representations, San Juan, Puerto Rico, 2–4 May 2016.
15. Haarnoja, T.; Zhou, A.; Abbeel, P.; Levine, S. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 1861–1870.
16. Tai, L.; Paolo, G.; Liu, M. Virtual-to-real deep reinforcement learning: Continuous control of mobile robots for mapless navigation. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017; pp. 31–36.
17. Kahn, G.; Villafior, A.; Pong, V.; Abbeel, P.; Levine, S. Uncertainty-aware reinforcement learning for collision avoidance. *arXiv* **2017**, arXiv:1702.01182.
18. Long, P.; Fan, T.; Liao, X.; Liu, W.; Zhang, H.; Pan, J. Towards optimally decentralized multi-robot collision avoidance via deep reinforcement learning. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018; pp. 6252–6259.
19. Matarić, M.J. Reinforcement learning in the multi-robot domain. *Auton. Robot.* **1997**, *4*, 73–83. [[CrossRef](#)]
20. Hüttenrauch, M.; Sosc, A.; Neumann, G. Deep reinforcement learning for swarm systems. *J. Mach. Learn. Res.* **2019**, *20*, 1–31.
21. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal policy optimization algorithms. *arXiv* **2017**, arXiv:1707.06347.
22. Mnih, V.; Badia, A.P.; Mirza, M.; Graves, A.; Lillicrap, T.; Harley, T.; Silver, D.; Kavukcuoglu, K. Asynchronous methods for deep reinforcement learning. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 19–24 June 2016; pp. 1928–1937.
23. Todorov, E.; Erez, T.; Tassa, Y. MuJoCo: A physics engine for model-based control. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, Portugal, 7–12 October 2012; pp. 5026–5033.
24. García, J.; Fernández, F. A comprehensive survey on safe reinforcement learning. *J. Mach. Learn. Res.* **2015**, *16*, 1437–1480.
25. Achiam, J.; Held, D.; Tamar, A.; Abbeel, P. Constrained policy optimization. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 22–31.
26. Tobin, J.; Fong, R.; Ray, A.; Schneider, J.; Zaremba, W.; Abbeel, P. Domain randomization for transferring deep neural networks from simulation to the real world. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017; pp. 23–30.
27. Peng, X.B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P. Sim-to-real transfer of robotic control with dynamics randomization. In Proceedings of the IEEE International Conference on Robotics and Automation, Brisbane, Australia, 21–25 May 2018; pp. 3803–3810.
28. Zhao, W.; Queralta, J.P.; Westerlund, T. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In Proceedings of the IEEE Symposium Series on Computational Intelligence, Canberra, Australia, 1–4 December 2020; pp. 737–744.
29. Dulac-Arnold, G.; Mankowitz, D.; Hester, T. Challenges of real-world reinforcement learning. *arXiv* **2019**, arXiv:1904.12901.
30. Gymnasium Contributors. Gymnasium: A Standard API for Reinforcement Learning Environments. 2023. Available online: <https://gymnasium.farama.org> (accessed on 30 March 2026).
31. Raffin, A.; Hill, A.; Gleave, A.; Kanervisto, A.; Ernestus, M.; Dormann, N. Stable-Baselines3: Reliable reinforcement learning implementations. *J. Mach. Learn. Res.* **2021**, *22*, 1–8.

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.