

Interception of adversarial drones in urban spaces using a modified deep reinforcement learning approach

Nicolas Lucotte, Adolfo Perrusquía*, Antonios Tsourdos, Weisi Guo, Hyo-Sang Shin

Faculty of Engineering and Applied Sciences, Cranfield University, Bedford MK43 0AL, UK

HIGHLIGHTS

- A TD3-APF formulation that allows faster convergence to an optimal solution and improves interception success.
- Local minima are avoided of traditional DRL approaches such that the interception time is minimised.
- The approach shows resilience and robustness under different number of obstacles and shapes.

ARTICLE INFO

Keywords:

Target interception
Twin-delayed deep deterministic policy gradient (TD3)
Artificial potential field (APF)
Collision avoidance
Urban environments
Drones

ABSTRACT

The proliferation of drone technologies in urban spaces has increased the threat space on potential attacks of key assets and people. The early disruption of adversarial drones is crucial to ensuring safety and the smooth operation of transportation services. Previous approaches use target interception algorithms for well-known trajectory profiles such that the interception problem is reduced to solving a geometric task. However, the unpredictable nature of the decision-making of drones poses significant challenges to traditional interception algorithms. To this end, this paper proposes a tactical and dynamic path planning approach that maximises the interception probability of adversarial drones in the presence of obstacles, whilst minimising the interception time. This is achieved by modifying a twin-delayed deep deterministic policy gradient (TD3) algorithm with an artificial potential field (APF) that increases the likelihood of success for the interception in a reduced time, whilst avoiding local minima issues. Extensive experiments are conducted to demonstrate the feasibility of the approach and future research directions.

1. Introduction

The fast proliferation of drones is becoming a challenge in security and defence [1,2]. The diverse applications of drones in urban spaces pose critical risks to security, privacy, and protection of both people and key assets [3]. The interception of these drones plays a pivotal role in ensuring the smooth operation for services. Therefore, there is a need of designing novel algorithmic tools or methodologies to support enforcers in the interception of drones that do not comply with safety and security regulations [4–6]. Traditional drone neutralisation approaches (e.g., missiles [7]) are not appropriate in urban areas due to the risk of damage to the surrounding area [8]. Recent research investments are focusing on using autonomous interceptor drones [9,10] that catch the drone and tow it into a safe area instead of neutralising it. In this case, the interceptor must adapt its mission profile and predict the movement of the target to intercept it as quickly as possible.

1.1. Related work

Solid-state LiDAR has been used to fuse two simultaneous scans with higher and lower frequencies to estimate the position of a drone [11]. The approach consists of first detecting the target using YOLOv5 over ranges of images generated from the LiDAR point cloud. Then, two simultaneous tracking estimators on two different scan frequencies are run in parallel, both using an extended Kalman filter (EKF) [12,13].

Proportional Navigation (PN) is one of the traditional techniques for target interception [14]. This algorithm is commonly used in missile systems to intercept moving targets. Reinforcement learning (RL) has been used to improve the capabilities of PN by modifying the navigation ratio in real-time. However, PN algorithms are designed for missile interception at higher altitudes, i.e., under clear skies which makes PN unsuitable for urban environments that contain obstacles.

* Corresponding author.

Email address: adolfo.perrusquia-guzman@cranfield.ac.uk (A. Perrusquía).

Multi-task algorithms are designed to alleviate the aforementioned issue. A fusion between the Dynamic Window Approach (DWA) and proportional guidance (PG) using RL was proposed in Ref. [15]. Here, a proximal policy optimisation (PPO) algorithm is used to switch between the DWA and PG algorithms. Artificial potential field (APF) is widely used in multi-task missions due to its high computational efficiency [16,17]. This method leverages the concept of virtual forces to guide robots toward a goal while avoiding obstacles. Here, the robot is treated as a particle moving in a potential field influenced by attractive and repulsive forces [18]. Also purely RL models [19,20] have been used for multi-tasking. In Refs. [21,22] a deep deterministic policy gradient (DDPG) is used to intercept drones with obstacle avoidance capabilities by designing specific reward functions for each task. However, the learning time is large. Additionally, DDPG algorithms have been improved to deal with high-density and non-deterministic environments in path planning approaches [23–26]. Also PPO has been used for target interception in both open-sky environments and in congested environments without considering interception time as a key metric [27,28].

A multi experience pool mechanism is used with a Twin-Delayed DDPG (TD3) algorithm [29] to improve the convergence time. This technique is quite useful when the training fails in a local minimum. In this case, learning from a single experience pool is not optimal (as the pool is filled with non-optimal experiences). In order to avoid this situation, three experience pools are created: the failure one, the success one, and the temporary one. If the episode is a success, the data from the temporary pool are saved in the success pool. If the episode is a failure, it is saved in the failure pool. The data sampled to update the networks are taken following a certain proportion from the success and failure pools. However, this approach requires high-computational resources which increases the interception time. There are recent approaches that combine the benefits of heuristic methods with RL algorithms [30–32]; however, these approaches are constrained to path planning approaches that cannot be easily migrated to target interception problems under random target manoeuvres and highly dense environments [33,34].

1.2. Contributions

In view of the above, this paper proposes a target interception algorithm that combines the advantages of the artificial potential field (APF) with the twin-delayed DDPG (TD3) algorithm. Here, the TD3 algorithm is used to approach the target as fast as possible, whilst the APF algorithm is implemented when the interceptor is close to the target within a threshold. The contributions of the approach are twofold: 1) the TD3-APF algorithm converges faster to an optimal solution and improves the interception success and 2) local minima are avoided such that the interception time is minimised. Extensive simulation studies are carried out under simulated environments with random obstacles of different shapes, sizes and locations. The results verify the advantages of the proposed model over the state-of-the-art DDPG and TD3 algorithms. Table 1 shows a summary of the variables used in this work.

2. Problem statement and methodology

The interception problem is depicted in Fig. 1, which consists of an interceptor drone detected in a restricted urban area, a target drone, and a set of obstacles. Here, an interceptor drone (interceptor) is deployed to intercept the drone without collisions and as quickly as possible using a data-driven target interception algorithm. To this end and without loss of generality, we consider that the following assumptions are fulfilled,

Assumption 1. The interceptor has access to real-time measurements of its surrounding environment within a finite sensing range, including relative obstacle distances and the target position within its field of view.

Assumption 2. Disturbances and unmodeled terms are compensated for by the low-level controller, which is not discussed in this paper. In addition, the low-level control is in charge of executing the high-level reference commands of the proposed interception planner.

Table 1
Variables used in this work.

Symbol	Description
p	Position vector
v	Velocity vector
v_x, v_y	Velocity components
a_x, a_y	Acceleration components
θ	Orientation of the yaw angle
B	angle between agents
$F_a^T, F_a^I, F_r^T, F_r^I, F_{r'}^T, F_{r'}^I$	APF attractive and repulsive forces
l_i	LIDAR scan
$v_{\max}, a_{\max}$	Maximum velocity and acceleration magnitudes
x	Observation vector
u	Action vector
$Q(x_k, u_k \phi_i)$	Q-network
ϕ_i	Neural weights of Q-network i
τ	Soft update hyperparameter
σ	Random noise
$k_t, k_d, k_o, k_i k_v, k_f, k_s$	Reward gains
k_a^I, k_a^I	Attractive force gains
k_{urg}^I, k_{urg}^I	Repulsive force gains in emergency zones
$k_{rep1}^I, k_{rep2}^I, k_{rep}^I$	Repulsive force gains for agents and obstacles
$d_{\min}, d_{th}$	Repulsive thresholds
s_{th}	Switching threshold
$d_{f_{\text{rout}}}, d_{rep}$	Interceptor distance thresholds

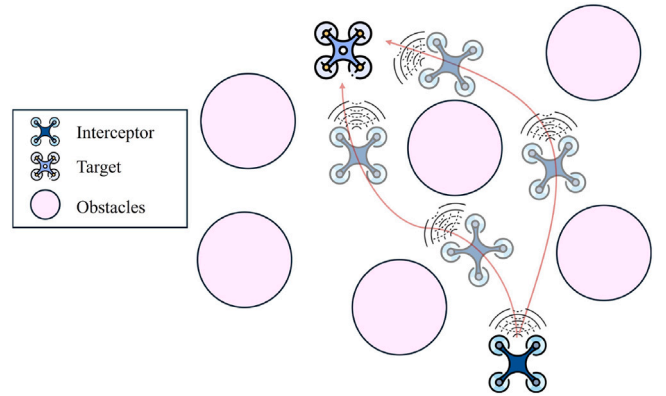


Fig. 1. Target interception problem: A drone interceptor aims to intercept the trajectory of a drone target by navigating through an environment with obstacles.

Assumption 3. The scenario takes place in an intelligent city infrastructure equipped with multi-frequency LiDAR sensors that provide accurate estimates of the positions of both the intruder and the interceptor. These measurements are assumed to be noise-free, instantaneous, and without range or coverage limitations, and are communicated to the interceptor without delay. Consequently, the interceptor operates under a fully observable and closed-loop setting.

Assumption 1 defines the sensing capabilities at a modeling level, namely the availability of environmental measurements within a finite range and field of view. In this work, these capabilities are instantiated using LiDAR-based sensing with a range of 20 m and a field of view of π radians. Single-beam LiDAR sensors are a common practical choice due to their low cost, accuracy, and scalability [35]. **Assumption 2** is acceptable (to some extent) by considering a robust low-level control design that attenuates external disturbances, e.g., PID, sliding mode, or novel model predictive controllers (MPCs) [36–38]. **Assumption 3** is motivated by recent developments in smart city sensing infrastructures, which enable real-time localization and tracking of aerial agents, and can be reasonably approximated as providing accurate state information in the considered scenario [39].

Based on these assumptions and Fig. 1, the target receives information about the interceptor from the smart city, while the interceptor receives the LiDAR measurements and the position of the target.

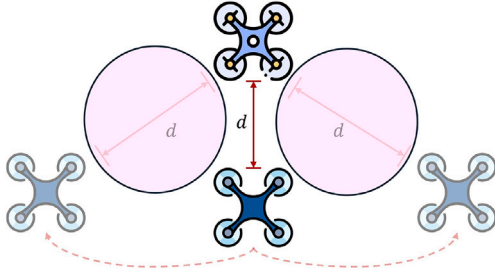


Fig. 2. Local minima challenge of previous data-driven interception algorithms.

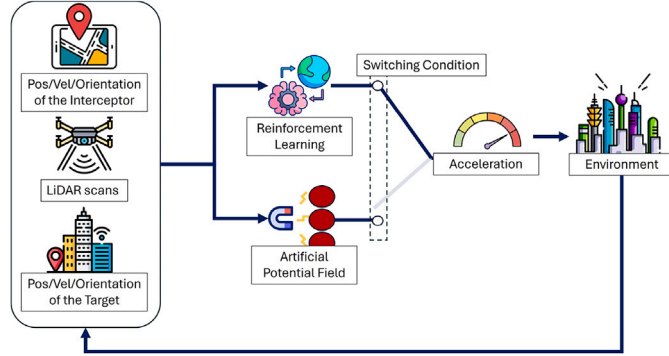


Fig. 3. High-level diagram of the proposed TD3-APF approach.

The target will attempt to evade the interceptor by applying evasion maneuvers. The main goal of the interceptor is to safely intercept the target (without colliding with obstacles) as quickly as possible to prevent its advance into the restricted zone. One of the main challenges in the interception problem arises when the interceptor gets too close to the drone target in the presence of obstacles. Fig. 2 illustrates an example scenario of this challenge. Here, the interceptor and target are separated by a distance d , but the interceptor cannot get closer to the target due to two nearby obstacles. The interceptor must navigate around the obstacles to effectively intercept the target, which increases the distance between the drones. This creates a cycle-limit problem because every time the interceptor moves away from the target, the interception algorithm incentivizes the interceptor to approach the drone target, returning to the same previous position. Previous RL-based interception algorithms, such as DDPG and TD3, attempt to mitigate this problem by increasing the interception distance, which is undesirable since the target could escape. Furthermore, these approaches struggle when the target is close to the interceptor, leading to an increase in interception time.

To address this challenge, this paper proposes a complementary TD3-APF algorithm that takes advantage of the complementary attributes of both the TD3 and APF algorithms to successfully intercept the target in reduced time. The overall architecture of the proposed approach is depicted in Fig. 3.

Fig. 3 shows how the observations of the interceptor and target feed both the TD3 reinforcement learning and APF algorithm. The key idea is to use TD3 as the main algorithm to approach the target as quickly as possible. A switching condition is designed to switch from TD3 to a modified APF algorithm that allows the interceptor to get closer to the target and avoid local minima issues by fulfilling a proximity condition.

2.1. Drone motion model

We consider a simple 2D kinematic model for both drones with a constant altitude that satisfies aviation regulations. With this particular model, the roll and pitch angles are neglected and we just consider the yaw angle as the main orientation angle of each drone. For simplicity,

we assume that the yaw angle matches the orientation of the velocity vector v .

Remark 1. We focus on a 2D environment to evaluate the proposed method under a simplified yet representative setting. Extending the framework to urban 3D operation is non-trivial because altitude becomes an additional optimisation variable constrained by aviation regulations, geofencing requirements, no-fly zones, obstacle avoidance, and dynamic airspace policies. These factors require richer environmental modeling and additional constraints within the optimisation process, significantly increasing problem complexity. A comprehensive treatment of regulatory-aware 3D planning is therefore left for future work.

We consider the following kinematic model,

$$\mathbf{x}_k = \mathbf{x}_{k-1} + \mathbf{b}\Delta t, \quad (1)$$

where $\mathbf{x}_k = [p_x, p_y, v_x, v_y]^T \in \mathbb{R}^4$ is the state vector that describes the linear position, velocity, and acceleration of the drone, $\mathbf{b} = [v_x, v_y, a_x, a_y]^T \in \mathbb{R}^4$ and Δt is a sampling time between measurements in the time index $k-1$ and k . In this study, we measure the position of the target drone in the global frame (measured from the LiDARs of the smart city), whilst the position of the interceptor is measured in its local frame that moves in the direction of the velocity vector. The velocity magnitudes of both the interceptor and the target are described by $v^I = \sqrt{(v_x^I)^2 + (v_y^I)^2}$ and $v^T = \sqrt{(v_x^T)^2 + (v_y^T)^2}$, respectively. The superscripts T, I denote the target drone and the interceptor drone, respectively. This model uses the acceleration components as control inputs, which are decomposed as follows

$$\begin{bmatrix} a_x \\ a_y \end{bmatrix} = a \begin{bmatrix} \cos(\theta) \\ \sin(\theta) \end{bmatrix}, \quad (2)$$

where a is the acceleration magnitude and θ is the orientation of the yaw angle. We add the following kinematic constraints for both the target and interceptor drones: maximum velocities of $v_{\max}^I = 20$ m/s and $v_{\max}^T = 18$ m/s, and a maximum acceleration of $a_{\max} = 5$ m/s² for both drones. The sampling time is $\Delta t = 0.05$ seconds. Therefore, the magnitude of the acceleration is $a \in [-5, 5]$ m/s² and $\theta \in [-\pi, \pi]$ radians.

2.2. Environment

A two-dimensional square environment of size 200×200 m² is considered in this work. Circular obstacles [15,21,40] with varying radii are randomly placed within the environment to enhance generalization. The obstacle locations are resampled at the beginning of each training episode of the RL-based interceptor. Obstacles are positioned within the region spanning 50–150 meters from the origin, and each obstacle radius is uniformly sampled between 3 and 10 meters. The initial positions of both the interceptor and target drones are randomly assigned within the central area of the environment. We consider $N = 10$ obstacles in total.

2.3. LiDAR observations

The interceptor drone is equipped with 19 single-beam LiDAR sensors as the main measurement source with a range of 20 meters. These LiDARs are uniformly located with an interval of $\frac{\pi}{18}$ radians to cover an angle of π radians. The length of the line of each LiDAR will be shorter when it intersects an obstacle. In order to create a more realistic environment, noise is added to the measurements. An SF20/C Lightware [41] linear LiDAR is considered with a range of 100 meters and a worst-case error of 10 cm. An exponential distribution will be used to represent the noise of these LiDARs. Fig. 4 shows a sketch of this exponential distribution using 1000 samples.

This noise is randomly set as positive or negative such that we obtain different noise in each iteration. Since the maximum range of the LiDARs is $l_{\max} = 20$ meters, data above 20 will be set to 20, and negative data will be set to 0. The final LiDAR data are given by the vector $\mathbf{l} = [l_1, l_2, \dots, l_{19}]^T \in \mathbb{R}^{19}$.

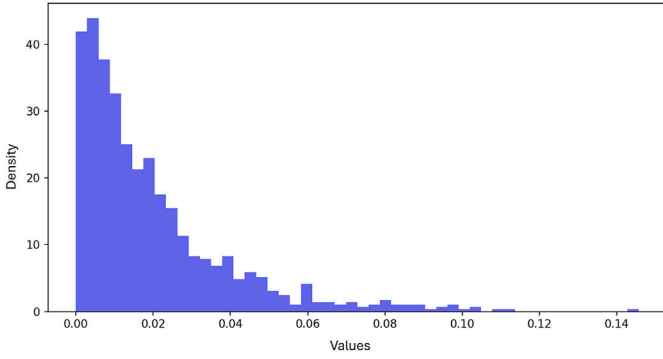


Fig. 4. Noise distributions of the LiDAR sensors.

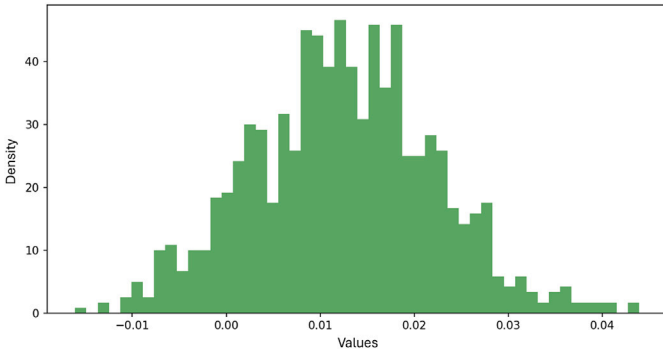


Fig. 5. Noise distributions of the target position.

Assumption 3 states that the city is equipped with a sensing infrastructure capable of accurately estimating the positions of both the interceptor and the target. The interceptor's position is assumed to be noise-free, as it is part of the smart-city system and is therefore fully observable. In contrast, the target drone is considered an external intruder, and its position estimate is subject to measurement uncertainty. To model this uncertainty, we adopt the sensing performance reported in [11], where a multi-frequency LiDAR system achieves a root-mean-squared position error (RMSE) of 0.014 meters when tracking a moving UAV. Since the error is expressed as an RMSE, this work adopts a Gaussian distributed noise centered on 0.014 meters with a standard deviation of 0.01 meters. Fig. 5 shows the Gaussian distribution of the position error of the target drone. This noise is sampled at each measurement and added to the true target position, with both positive and negative deviations occurring randomly.

We consider two categories of LiDARs: critical L_{critic} and the side LiDARs L_{side} which are used to design different repulsive forces of the APF algorithm discussed in subsequent sections. Here, we consider as critical LiDARs the 5 scans located in the middle, whilst the rest of the scans are considered as side LiDARs. Fig. 6 shows a simple visualization of the two categories of LiDARs.

3. Reinforcement learning design

From the previous section, we can easily observe that we are dealing with a Partially Observable Markov Decision Process (POMDP) with continuous observations and continuous actions. Therefore, this section is dedicated to formulating a deep reinforcement learning (DRL) architecture that can deal with these continuous observation-action spaces.

3.1. Observation and action space modeling

To construct the observation space for this particular problem we consider the data obtained from: the LiDARs, the position of the

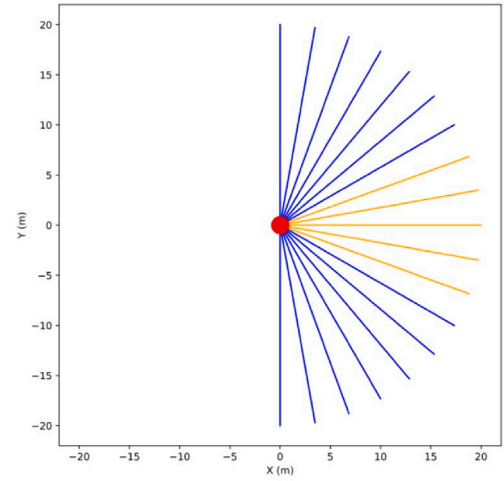


Fig. 6. LiDARs categories. Critical LiDARs are depicted in Orange, whilst the side LiDARs are shown in blue colour.

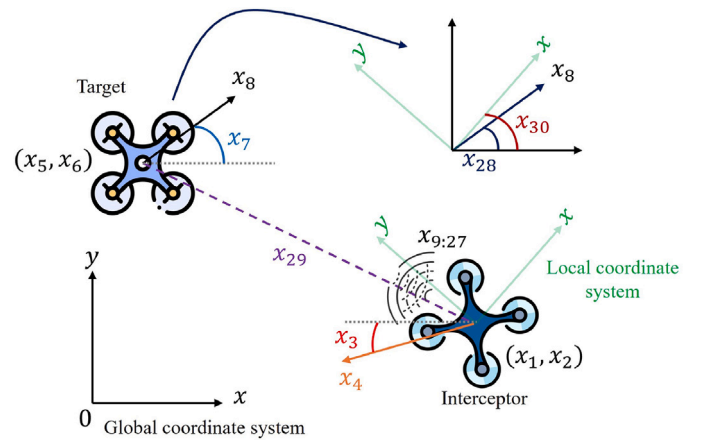


Fig. 7. Observation space: (x_1, x_2) defines the position of the interceptor, x_3 the orientation of the velocity vector of the interceptor measured from the global frame, x_4 the ratio of the velocity of the interceptor, (x_5, x_6) the position of the target, x_7 the orientation of the velocity vector of the target measured from the global frame, x_8 the ratio of the velocity of the target, $x_{9:27}$ the 19 LiDARs' measurements, x_{28} the direction of the target in the global frame, x_{29} the distance between the interceptor and the target, and x_{30} the direction of the target with respect to the local frame.

interceptor, and the estimated position of the target [42]. These data are used to infer other observations such as the velocity of the agents. We used a small sample time of 0.05 s. We additionally normalised the observations to avoid scale issues. The state space is denoted as $\mathbf{x} = [x_1, x_2, \dots, x_{30}]^T$ and includes the following: **1)** the position of the interceptor $x_1, x_2 \in [0, 1]$, **2)** the orientation of the velocity vector of the interceptor in the global frame $x_3 \in [-1, 1]$, **3)** the ratio velocity of the interceptor $x_4 = \frac{v^I}{v_{\max}^I} \in [0, 1]$, **4)** the position of the target $x_5, x_6 \in [0, 1]$, **5)** the orientation of the velocity vector of the target in the global frame $x_7 \in [-1, 1]$, **6)** the ratio velocity of the target $x_8 = \frac{v^T}{v_{\max}^T} \in [0, 1]$, **7)** the 19 LiDARs' measurements $x_j \in [0, 1]$ with $j = 9, 10, \dots, 27$, **8)** the direction of the target in the global frame $x_{28} \in [-1, 1]$, **9)** the distance between the target and the interceptor $x_{29} \in [0, 1]$, and **10)** the direction of the target in the local frame $x_{30} \in [-1, 1]$. Fig. 7 shows a simple schematic of the observation variables obtained from the geometry of the target-interception problem.

The action space is also normalised between -1 and 1 to improve stability and convergence in the training stage. Therefore, the final action space is $a_{RL} = [-1, 1]$ and $\theta_{RL} \in [-1, 1]$, and hence $u = [a_{RL}, \theta_{RL}]^T$.

3.2. Reward function

The reward function is carefully designed and decomposed into 7 main components. Let $k_t, k_d, k_o, k_l, k_v, k_f, k_s > 0$ denote constant gains associated with each term. First, the terminal reward r_e depends on the episode outcome: $r_e = k_t$ if the target is intercepted, $r_e = -k_t$ if the interceptor collides with an obstacle, and $r_e = 0$ if the maximum step limit is reached. The distance term $r_d = k_d \sqrt{200^2 + 200^2(x_{29}^{k-1} - x_{29}^k)}$ encourages reduction of the target distance, yielding a positive value when the interceptor approaches the target and negative otherwise. The orientation component $r_o = k_o(0.5 - x_{30})$ promotes alignment of the interceptor's heading toward the target direction. To discourage approaches to obstacles, the LiDAR-based penalty $r_l = k_l \sum_{j=9}^{27} \min\left(\frac{1-x_j}{x_{j\max}}, 1.5\right)$ is introduced, providing stronger penalties when LiDAR readings indicate close obstacles. The velocity reward $r_v = k_v v_{\max}^l (1 - x_4)$ motivates fast interception behavior by favoring higher speeds. A safety term $r_f = \min\left(\frac{k_f(x_{18\max} - d_s)}{d_s}, 0\right)$ further prevents unsafe forward motion, penalizing the agent when the LiDAR ray aligned with the velocity direction falls below the defined safe distance d_s . Finally, a step cost $r_s = -k_s$ imposes a constant negative reward each timestep to promote efficient task completion. Collecting all terms, the total reward at step k is given by $r_k = r_e + r_d + r_o + r_l + r_v + r_f + r_s$. Each constant gain is manually adjusted until the best performance is achieved.

3.3. Twin-delayed deep deterministic policy gradient (TD3) algorithm

We adopt the twin-delayed deep deterministic policy gradient (TD3) algorithm as the underlying learning framework, owing to its ability to learn deterministic policies in continuous action spaces while mitigating overestimation bias. TD3 has demonstrated strong performance in target interception problems [29,43,44], making it a suitable choice for our setting. The algorithm incorporates three key techniques that enhance training stability and robustness: (i) *clipped double Q-learning*, where two critic networks are employed to reduce value overestimation; (ii) *delayed policy updates*, in which the actor is updated less frequently than the critics to ensure reliable value estimates; and (iii) *target policy smoothing*, where noise is injected into the target action to prevent exploitation of sharp value gradients.

The critic networks are parameterized by a set of weights denoted as $\phi_i, i = 1, 2$ that aim to minimise their respective loss function given by [45]

$$\mathcal{L}(\phi_i) = \mathbb{E}_h[(Q(x_k, u_k | \phi_i) - y_k^i)^2], \quad (3)$$

where $y_k^i = r_k + \gamma Q^h(x_{k+1}, h(x_{k+1}) | \phi_i)$ is the temporal difference (TD) target, r_k is the reward function and γ is the discount factor. The actor network with parameters θ aims to learn the optimal policy $h(x|\theta)$ that maximises the Q network $Q(x, u | \phi_i)$ as $\max_{\theta} \mathbb{E}_{x \sim \mathcal{D}}[Q_{\phi_i}(x, h(x|\theta))]$, where $\mathcal{D}$ is a replay buffer that stores the trajectories (x_k, u_k, r_k, x_{k+1}) in each iteration k . Both the actor and critic networks use a small batch sampled from the replay buffer.

The target network is a copy of the main network and is composed of an actor network $h(x|\theta')$ and a critic Q network $Q(x, u | \phi'_i)$. A soft-update is used to maintain stable performance while learning [45], i.e.,

$$\theta' = \tau\theta + (1 - \tau)\theta', \quad \phi'_i = \tau\phi_i + (1 - \tau)\phi'_i, \quad (4)$$

where $\tau \in [0, 1]$ is the soft update rate. Random noise σ is added to the selected action, i.e., $u' = u + \sigma$, to improve the generalisation of the learned policy.

Remark 2. Stochastic RL agents, such as proximal policy optimisation (PPO) or soft actor critic (SAC), are not commonly used in target

interception tasks. This is because interception requires precision, predictability, and stability, such that stochasticity tends to degrade all three of these factors. This is demonstrated in the related work mentioned above, where the majority of the work adopts DDPG/TD3 as the main RL agent.

4. Proposed approach: TD3-APF algorithm

Although DDPG demonstrates strong performance in many interception scenarios, it struggles when the interceptor approaches obstacles, as illustrated in Fig. 2. In such situations, the agent tends to remain stationary because it does not learn an appropriate avoidance maneuver that would allow it to bypass the obstacle and continue pursuing the target. This behavior is expected: navigating around the obstacle temporarily increases the distance to the target, resulting in a short-term reduction in reward, which conflicts with the reinforcement learning objective of immediate reward maximization. On the other hand, TD3 tends to reduce the velocity such that it reduces collisions, but increases the likelihood of not intercepting the target.

To address this limitation, we incorporate an Artificial Potential Field (APF) mechanism into the TD3 framework. The key idea is to employ TD3 to guide the interceptor toward the target up to a predefined proximity and within the LiDAR sensing range. Once this condition is satisfied, control is seamlessly transferred to the APF module, which then guides the agent to intercept the target while ensuring collision-free navigation and avoiding local minima. The details of the proposed APF-based strategy are presented in the following subsection.

4.1. Artificial potential field (APF)

The APF algorithm will be used for both agents: the target (as an evasion strategy) and the interceptor.

4.1.1. Target APF

To prevent collisions with obstacles, we designed an APF algorithm that considers a security distance between the target and the obstacles, such that if the target is too close to the obstacles, it will cancel its attractive force and amplify the repulsive force of the obstacle. To this end, we design the attractive force of the platform as $F_a^T = k_a^T(p_{platform} - p_{target})$, where k_a^T is the attractive coefficient of the target, $p_{platform}$ is the position vector of the platform, and $p_{target} = [x_5, x_6]^T$ is the position vector of the target. We consider three scenarios for the repulsive force of obstacles: (1) when the distance between the target and the obstacles is equal to or less than a safe security distance $d_{min} > 0$, (2) when the distance between the target and the obstacle is less than a repulsive threshold $d_{th} > 0$, and (3) when the distance between the target and the obstacles is greater than the repulsive threshold $d_{th} > 0$ which means that there is no repulsive force. This is formally described as follows

$$F_{r_n}^T = \begin{cases} k_{urg}^T \left(\frac{1}{d_n} - \frac{1}{d_{min}} \right) \frac{p_{target} - C_n}{d_n^3} & \text{if } d_n < d_{min} \\ k_{rep1}^T \left(\frac{1}{d_n} - \frac{1}{d_{th}} \right) \frac{p_{target} - C_n}{d_n^3} & \text{if } d_n < d_{th} \\ \mathbf{0}_{2 \times 1} & \text{otherwise,} \end{cases} \quad (5)$$

where $k_{urg}^T, k_{rep1}^T > 0$ are the security and threshold gains. The total repulsive force of obstacles is given by

$$F_{r_o}^T = \sum_{n=1}^N F_{r_n}^T, \quad (6)$$

where N denotes the total number of obstacles. The variable d_n corresponds to the distance between the target position p_{target} and the nearest point of obstacle C_n . The repulsive force from the interceptor is applied just when the following two conditions are satisfied: $d_n < d_{th}$ and $d_n > d_{min}$, i.e., that the target is not in an emergency zone. Then, the

repulsive force from the interceptor is given by

$$\mathbf{F}_{r^I}^T = k_{rep_2}^T \left(\frac{1}{x_{29}} - \frac{1}{d_{th}} \right) \frac{\mathbf{p}_{target} - \mathbf{p}_{interceptor}}{x_{29}^3}, \quad (7)$$

where $\mathbf{p}_{interceptor} = [x_1, x_2]^T$ is the position vector of the interceptor and $k_{rep_2}^T$ represents the repulsive gain applied to the interceptor. The total artificial force acting on the target is computed as $\mathbf{F}_T = \mathbf{F}_a^T + \mathbf{F}_{r^o}^T + \mathbf{F}_{r^I}^T$, and this force is used as the input target velocity $\mathbf{v}^T = \mathbf{F}_T$.

4.1.2. Interceptor APF

The interceptor is designed to reach the target while avoiding collisions with obstacles. We assume that the interceptor operates under noisy and partial observations, consisting of measurements from 19 linear LiDAR sensors, the target position, and its own position. The attractive force acting on the interceptor is defined as $\mathbf{F}_a^I = k_a^I (\mathbf{p}_{target} - \mathbf{p}_{interceptor})$, where k_a^I denotes the attractive gain. Since the interceptor does not have complete information about the obstacles, repulsive forces are generated based on LiDAR measurements. A repulsive force is assigned to each LiDAR scan that detects a potential obstacle, i.e., when the detected distance is less than 19 meters (with a maximum sensing range of 20 meters). Similar to the APF of the target drone, we consider different repulsive forces based on different thresholds and the LiDAR category (critic or side LiDAR). To this end, we define $d_{front} > 0$ as the repulsive threshold of the critic LiDARs L_{critic} , $d_{rep} > 0$ is the interceptor repulsive threshold, and we use the same security threshold d_{min} , then the interceptor repulsive force is described by

$$\mathbf{F}_{r_j}^I = \begin{cases} k_{urg}^I \left(\frac{1}{x_j l_{max}} - \frac{1}{d_{min}} \right) \frac{\mathbf{p}_{interceptor} - \mathbf{p}_l}{(x_j l_{max})^3} & \text{if } x_j l_{max} < d_{min} \\ 5k_{urg}^I \left(\frac{1}{x_j l_{max}} - \frac{1}{d_{front}} \right) \frac{\mathbf{p}_{interceptor} - \mathbf{p}_l}{(x_j l_{max})^3} & \text{if } x_j l_{max} < d_{front} \text{ and } j \in L_{critic} \\ k_{rep}^I \left(\frac{1}{x_j l_{max}} - \frac{1}{d_{rep}} \right) \frac{\mathbf{p}_{interceptor} - \mathbf{p}_l}{(x_j l_{max})^2} & \text{if } x_j l_{max} < d_{rep} \\ \mathbf{0}_{2 \times 1} & \text{otherwise,} \end{cases} \quad (8)$$

where $k_{urg}^I, k_{rep}^I > 0$ are the security and threshold gains of the interceptor and $\mathbf{p}_j$ defines the intersection point of the LiDAR j with the obstacle or target. The total repulsive force is given by

$$\mathbf{F}_r^I = \sum_{j=9}^{27} \mathbf{F}_{r_j}^I. \quad (9)$$

The total artificial force acting on the interceptor is $\mathbf{F}_I = \mathbf{F}_a^I + \mathbf{F}_r^I$, and the resulting interceptor velocity is $\mathbf{v}^I = \mathbf{F}_I$.

4.2. TD3-APF algorithm

Although the APF formulation improves obstacle avoidance, it exhibits similar behavior to the TD3 policy when the interceptor approaches obstacles: the magnitude of the resultant force diminishes due to the proximity of both the target and surrounding obstacles. To mitigate this effect and enhance interception performance, we introduce a simple switching condition. Let B denote the angle between the interceptor's velocity vector $\mathbf{x}_4$ and the displacement vector $\mathbf{p}_{target} - \mathbf{p}_{interceptor}$, and also consider the switching condition $s_{th} > 0$ that allows the switch between the TD3 and APF algorithms. Assuming a critical scenario, i.e., when the interceptor is close to the obstacles, then if $\|\mathbf{p}_{target} - \mathbf{p}_{interceptor}\| < s_{th}$ and $|B| < 0.5\pi$, the following indices are computed

$$I_{up} = \left\lceil \frac{18B}{\pi} \right\rceil + 10, \quad I_{down} = \left\lfloor \frac{18B}{\pi} \right\rfloor + 10, \quad (10)$$

where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceiling and floor functions, respectively, and I_j are the LiDAR measurements with $j = 1, \dots, 19$. Then, if the conditions $I_{up} > \|\mathbf{p}_{target} - \mathbf{p}_{interceptor}\|$ and $I_{down} > \|\mathbf{p}_{target} - \mathbf{p}_{interceptor}\|$ are satisfied, we set the interceptor's velocity as $\mathbf{v}^I = \mathbf{F}_I = \mathbf{F}_a^I$.

Algorithm 1 TD3-APF.

Require: Position of the target $\mathbf{p}_{target}$, position of the interceptor $\mathbf{p}_{interceptor}$, thresholds s_{th}, d_{min} , angle B , and maximum number of iterations N

- 1: **for** $n = 1$ **to** N **do**
- 2: APF=False
- 3: **if** $\|\mathbf{p}_{target} - \mathbf{p}_{interceptor}\| < s_{th}$ and $|B| < 0.5\pi$ **then**
- 4: Compute the indices I_{up} and I_{down} using (10)
- 5: **if** $I_{up} > \|\mathbf{p}_{target} - \mathbf{p}_{interceptor}\|$ and $I_{down} > \|\mathbf{p}_{target} - \mathbf{p}_{interceptor}\|$ **then**
- 6: APF=True
- 7: **end if**
- 8: **end if**
- 9: **for** $j = 1$ **to** L **do**
- 10: **if** ($j \in L_{critic}$ and $I_j < d_{min}$) or $I_j < 0.5$ **then**
- 11: APF=True
- 12: **end if**
- 13: **end for**
- 14: **if** APF=True **then**
- 15: Actions are chosen by the APF algorithm.
- 16: **else**
- 17: Actions are chosen by the TD3 algorithm.
- 18: **end if**
- 19: **end for**

The proposed condition works as follows: if the target is within a radius of 19 meters from the interceptor and in the line of sight of the LiDARS field, then it checks if there are obstacles between the interceptor and the target. If no obstacle is detected, then the velocity vector will be a straight vector in the direction of the target. This is the interception part of the APF which allows for quickly intercepting the target in situations where obstacles are close. The proposed TD3-APF algorithm is summarized in Algorithm 1.

We adapt the forces of the APF to comply with the maximum velocity and acceleration constraints for both the target and the interceptor. For the acceleration constraint, we have that the inequality $\|\mathbf{v}_k - \mathbf{v}_{k-1}\| \leq a_{max} \Delta t$ has to be satisfied. If the above inequality is not satisfied, then the velocity vector is modified as follows

$$\begin{bmatrix} b_x \\ b_y \end{bmatrix} = \frac{\mathbf{v}_k - \mathbf{v}_{k-1}}{\Delta t}, \quad \alpha = \arctan\left(\frac{b_y}{b_x}\right) \quad (11)$$

$$\mathbf{a} = a_{max} \begin{bmatrix} \cos(\alpha) \\ \sin(\alpha) \end{bmatrix}, \quad \mathbf{v}_k = \mathbf{v}_{k-1} + \mathbf{a} \Delta t,$$

where $\mathbf{v}_k = [v_x, v_y]^T \in \mathbb{R}^2$ is either the velocity vector of the target or the interceptor. This process allows to obtain a new velocity vector $\mathbf{v}$ that satisfies the maximum acceleration constraint. To deal with the velocity constraint, we follow a similar approach. If $\|\mathbf{v}\| \geq v_{max}$ (for either the target or interceptor) then we compute

$$B = \arctan\left(\frac{v_y}{v_x}\right), \quad \begin{bmatrix} v_x \\ v_y \end{bmatrix} = v_{max} \begin{bmatrix} \cos(B) \\ \sin(B) \end{bmatrix}. \quad (12)$$

Then, the next position of the agent can be calculated using the simple formula $\mathbf{p}_k = \mathbf{p}_{k-1} + \mathbf{v}_k \Delta t$, where $\mathbf{p}_k = [p_x, p_y] \in \mathbb{R}^2$.

Remark 3. The overall computational complexity of the proposed algorithm is dominated by the TD3 algorithm, whose computational complexity is given by $\mathcal{O}(B_s T(4n_c + (1/d)n_a))$, where B_s is the batch size, n_c, n_a are the computational costs of the critic and actor networks, respectively, in terms of parameters/operations, d corresponds to the delays in the actor updates, and T is the total number of steps. Similar results were reported in Ref. [46].

Remark 4. The proposed switching condition is independent of the number of LiDAR beams, as it relies solely on the geometric property that the target direction is bracketed by two adjacent rays. Therefore,

the condition remains valid for different sensor resolutions. However, lower angular resolutions increase the separation between consecutive beams, which may reduce the conservativeness of obstacle detection in borderline cases.

Remark 5. The switching to the APF controller is performed only when a clear line-of-sight to the target is detected, which mitigates the occurrence of local minima in the terminal phase. Nevertheless, concave obstacle configurations may still challenge the TD3 policy during earlier navigation stages, as such geometries are not explicitly represented in the training environments.

5. Experimental studies

The proposed approach is tested and compared with state-of-the-art implementations which include DDPG and TD3. In this implementation a random environment is created and used in each episode to ensure generalisation across diverse interception scenarios. We use 8000 episodes for training (approximately 4–5 hours of training time). For all models, the target uses the APF algorithm as an evasion strategy. Each episode will have a maximum number of steps of 500. The algorithms are compared based on the average time of interception and the success rate.

The hyperparameters of the APF are set to $k_a^T = 0.5$, $d_{\min} = 2$, $d_{th} = 50$, $k_{rep1}^T = 9000$, $k_{rep2}^T = 1000$, $k_{urg}^T = 10,000$, $d_{front} = 3$, $d_{rep} = 10$, $k_a^I = 6$, $k_{urg}^I = 5000$ and $k_{rep}^I = 1800$. In this initial experiment we used a switching threshold of $s_{th} = 7$ meters, which will be tested in the ablation studies subsection.

For the reward function we use the following values that achieve the best performance for all models: $k_t = 100$, $k_d = k_o = 0.5$, $k_l = -0.1$, $k_v = -0.8$, $k_f = 0.1$, $d_s = 15$, and $k_k = 0.2$. Both the DDPG and TD3 algorithms use the following hyperparameter values: $\tau = 0.005$, $\gamma = 0.99$, $\sigma = 0.1$, learning rate $lr = 0.0003$, batch size of 256, and buffer size of 1,000,000. Additionally, for the TD3 we set the noise clip to 0.5 and the policy noise to 0.2. Both the actor and critic networks consist of two layer fully connected network with 400 and 300 neurons respectively with the activation function ReLU, where the main difference is in the input vector, i.e., 32 inputs for the critic and 30 for the actor. The total number of parameters is 133,801 and 133,001, respectively (approximately 266k floating-point operations (FLOPs)) with approximately 1ms per inference.

5.1. Comparison results

Table 2 summarizes the results obtained for each algorithm. We have four metrics based on the success rate of interception, the collision rate (safety), the unsuccessful rate that occurs when the agent is not capable of intercepting the target within the pre-defined number of steps, and the average time that each algorithm takes to intercept the target.

The convergence results of each algorithm are shown in Fig. 8. The results show that the proposed TD3-APF algorithm outperforms both DDPG and TD3 algorithms in intercepting the target in minimum time and with a 100% success rate. Notice that the DDPG algorithm exhibits more collisions, whilst TD3 has more unsuccessful missions. This is caused by the interception challenge when the obstacles are too close, as well as its main benefit of avoiding the overestimation of actions. Here, the action overestimation helps the DDPG algorithm escape from

this issue in some cases, which is reflected in the number of unsuccessful missions. On the other hand, the proposed TD3-APF exhibits stable performance and is capable of completing the mission every time in minimum average time.

Fig. 9 shows how the interception task looks in the proposed environment. Notice that the DDPG agent reduces the velocity in the presence of obstacles that are too close and during the interception phase. This can cause the interceptor to fail to intercept the target in several cases. TD3 effectively overcomes the velocity issue, however it still has the interception issue. On the other hand, the TD3-APF exhibits smooth trajectories and a faster interception time. The algorithm exploits the reward constraints of the TD3, i.e., safe distance and prediction of the interception point; along with the APF smooth target interception. The TD3-APF is faster than the other algorithms and maintains safe performance across all the random environments.

5.2. Ablation studies

We conduct three different ablation studies: (1) modify the TD3-APF algorithm to show how the APF and switching condition contribute to the target interception task, (2) study how different thresholds in the switching condition affect the effectiveness of the interception, and (3) examine how different sensor noise levels compromise the interception results [47].

5.2.1. TD3-APF algorithm ablation

We conduct an ablation study of the TD3-APF algorithm by considering a naive APF variant that omits the switching conditions (Lines 10–12 of Algorithm 1). The corresponding results are summarized in Table 3.

The results presented in Table 3 demonstrate the effectiveness of integrating the modified APF and TD3 strategies for target interception and safe navigation. The naive APF algorithm achieves a relatively high target-reaching rate of 92.9%; however, it still exhibits a collision rate of 5.8% and a 1.3% failure rate. These shortcomings are mainly attributed to the well-known local minima problem of conventional APF methods, which can trap the agent in suboptimal positions and prevent successful target interception. By contrast, the modified APF significantly improves performance, increasing the success rate to 99.9% while completely eliminating collisions. This result confirms that the proposed modifications effectively address the local minima issue and enhance navigation safety. Nevertheless, the average interception time slightly increases from 6.57 s to 6.62 s, indicating that the additional corrective behaviors introduced to improve safety may lead to marginally longer trajectories.

The TD3-APF (Naive) approach improves upon both the naive APF and TD3 methods. Specifically, it increases the target-reaching rate to 94.2%, reduces the interception time to 6.32 s, and completely eliminates failure cases. Although collisions still occur in 5.8% of the trials, the results indicate that combining reinforcement learning with APF guidance enhances both efficiency and robustness. The proposed TD3-APF framework achieves the best overall performance, attaining a 100% success rate with zero collisions and zero failed missions. Furthermore, it records the shortest average interception time of 5.70 s, outperforming all other approaches. These results demonstrate that the synergy between the improved APF guidance and the TD3 learning strategy not only guarantees safe navigation but also enables faster target interception. While the modified APF alone already provides excellent safety and reliability, the incorporation of TD3 further improves decision-making efficiency, resulting in an optimal balance between safety, robustness, and interception speed. This is particularly important in security and surveillance applications, where rapid and reliable responses are critical for protecting people and infrastructure.

5.2.2. Switching condition threshold ablation

As an ablation study we test different switching thresholds to observe how they affect the effectiveness of the proposed approach. We used the

Table 2

Success rate of the agents. Best results are in bold.

Model	Reach the target	Collision (Safety)	Not success	Average time success case
DDPG	92.50%	7.20%	0.30%	7.10 s
TD3	91.20%	6.10%	2.70%	8.16 s
TD3-APF	100%	0%	0%	5.70 s

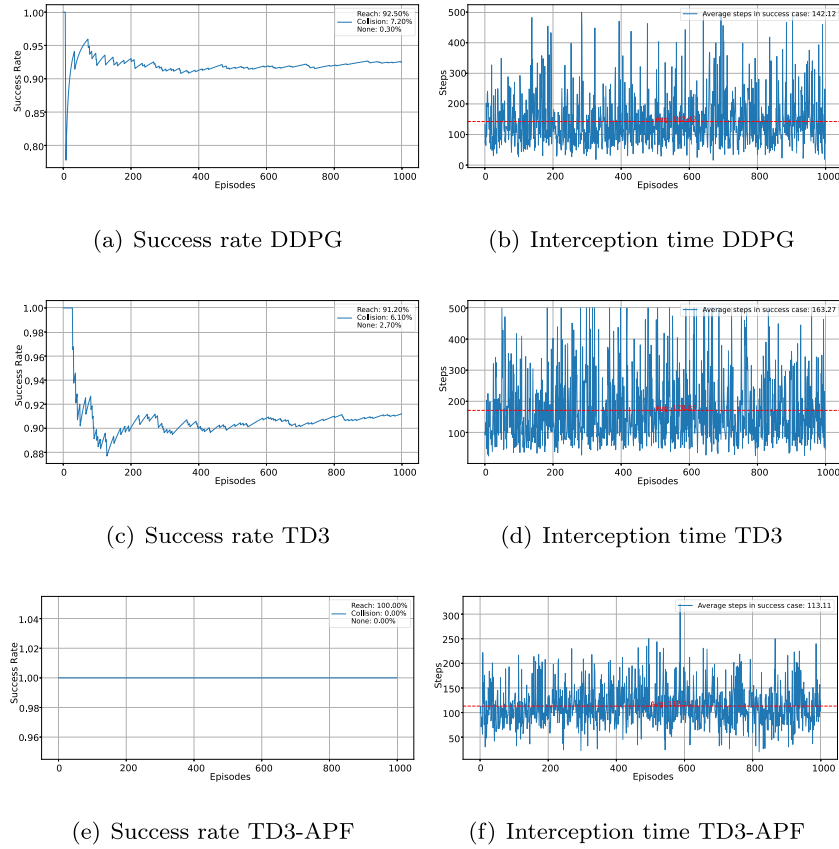


Fig. 8. Comparison results between DDPG, TD3, and TD3-APF.

same hyperparameters under 1000 random environments. The results are summarised in Table 4.

The results highlight that the switching threshold of 3 meters achieves the best interception time, whilst a switching threshold of 1.5 meters slightly reduces the success rate to 99.4% with 0.5% collisions and 0.1% of unsuccessful interceptions. This shows that the TD3-APF approach maintains its performance under different switching threshold conditions.

5.2.3. Noise ablation

We further test how the noise intensity affects the proposed TD3-APF approach. In this experiment the noise applied to each measurement is scaled using different factor gains. We test four factor gains: 0, 2, 5, and 10. The results obtained from 1000 random environments are given in Table 5.

The results without noise, i.e., with a scaling factor of 0, show that the obtained metrics are approximately equivalent to the baseline with noise. However, when the noise level is scaled by factors of 2 and 5, a slight degradation in the interception time is observed. This can be attributed to the agent perceiving obstacles as being closer than their actual positions, leading to overly conservative behavior such as premature or excessive slowing down. Importantly, safety remains unaffected across these settings: the collision rate stays at 0% even under increased noise levels. When the noise is further amplified by a factor of 10, the interception time becomes more pronounced, following the same underlying mechanism. Nevertheless, the system continues to maintain a 0% collision rate, indicating that the TD3-APF algorithm preserves strong safety guarantees even under substantial sensory noise.

5.3. Robustness

This section addresses the robustness of the TD3-APF algorithm under data loss. This is simulated by assigning an independent

probability of loss to each measurement at every time step. The considered measurements include the target position, the agent position, and the 19 LiDAR measurements. At the beginning of each time step, all measurements are computed. Then, for each individual measurement, a Boolean variable is randomly sampled according to a predefined probability distribution representing the likelihood of data loss. This process is repeated independently for every measurement at each time step. If the sampled Boolean variable is true, the corresponding measurement is considered lost. In such a case, the missing value is replaced with the most recently available measurement.

Remark 6. Data loss does not occur simultaneously across all measurements at a given time step. Instead, the process is applied independently to each sensor.

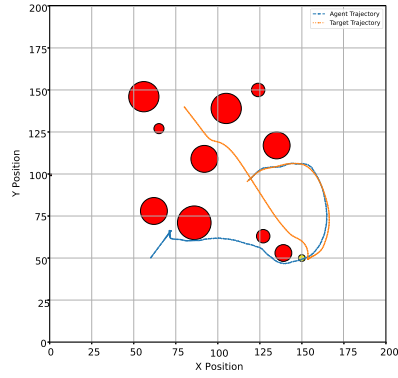
Since both TD3 and APF compute accelerations, velocity is obtained through direct integration. Therefore, the agent velocity observation is not affected by measurement loss. However, the target velocity estimation depends on the difference between two consecutive target position measurements divided by the elapsed time interval. As a result, target velocity estimation is directly impacted by missing data. To address this issue, when a target position measurement is lost, the target velocity observation is also replaced with its previously estimated value. Otherwise, the computed velocity would incorrectly become zero.

When a valid target measurement becomes available again after n consecutive missing steps, the target velocity is estimated as

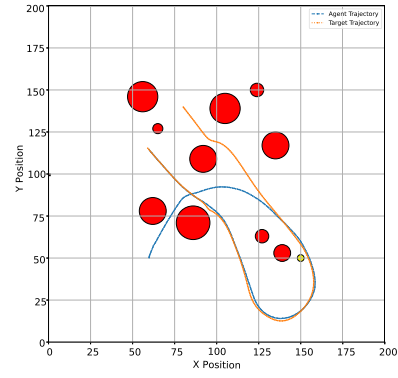
$$v_{\text{target}} = \frac{P_{\text{target},k+n} - P_{\text{target},k}}{t_{k+n} - t_k}.$$

The results under 1000 random environments and information loss are summarised in Table 6.

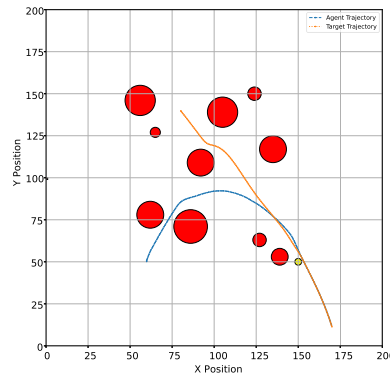
From the results of Table 6, we can observe that the TD3-APF approach is robust to data loss with minor percentage of collisions when



(a) DDPG



(b) TD3



(c) TD3-APF

Fig. 9. Interception Results.

Table 3

Success rate of the agents. Best results are in bold.

Model	Reach the target	Collision (Safety)	Not success	Average time success case
APF (Naive)	92.90%	5.80%	1.30%	6.57 s
APF	99.9%	0%	0.10%	6.62 s
TD3	91.20%	6.10%	2.70%	8.16 s
TD3-APF(Naive)	94.20%	5.80%	0%	6.32 s
TD3-APF	100%	0%	0%	5.70 s

Table 4

Interception results under different switching conditions. Best average time in bold.

Switching Threshold (m)	Success Rate	Collision	None	Average time of interception (s)
1.5	99.4	0.5	0.1	5.8
3	100	0	0	5.43
5	100	0	0	5.79
7	100	0	0	5.9
10	100	0	0	6.31

Table 5

Interception results under different noise intensities. Best average time in bold.

Noise gain	Success rate	Collision	None	Average time of Interception (s)
0	100	0	0	5.88
2	100	0	0	6.05
5	100	0	0	5.99
10	100	0	0	6.17

Table 6

Interception results under different noise intensities. Best average time in bold.

% of data loss	Success rate	Collision	None	Average time of Interception (s)
1	100	0	0	5.92
5	100	0	0	5.98
10	99.7	0.3	0	6.15
30	98.3	1.5	0.2	7.11

the data loss increases more than 10%. Here, the interception time increases proportionally to the percentage of data loss. This performance can be enhanced by incorporating sensor-fusion methods. However, we can observe consistent interception results with a minor number of collisions.

5.4. Generalisation

We perform additional experiments to show the generalisation of the TD3-APF when the number of obstacles increases, as well as considering obstacles with different shapes [48]. We consider 1000 environments with random obstacle configurations and different initial conditions for

both the target and the interceptor. For the 15 obstacles case, we obtain 99.90% success, 0.00% collisions, and 0.10% interception failure; with an average time of 5.92 s. For the 20 obstacles case, we obtain 99.90% success, 0.00% collisions, and 0.10% interception failure; with an average time of 5.73 s. Notice that in both cases, the interceptor fails to intercept the target in one single random environment setting. This is because the agent relies more on the APF rather than the TD3 algorithm due to the highly dense environment. This implies that the interceptor needs to reduce its velocity causing slower interception trajectories which results in failing the interception task. Nevertheless, both results show that the proposed approach is robust and maintains a good balance between speed and safety.

Fig. 10 shows the distance between agents and the interceptor velocity for both the 15 and 20 obstacle cases. Notice that in both scenarios, the TD3-APF algorithm can effectively reduce the distance between the agents and modulate its velocity to intercept the target as quickly as possible without collisions. Fig. 11 shows some sample trajectories of the 15 and 20 obstacle cases using the TD3-APF algorithm, where we can observe smooth and safe trajectories.

In Table 4 we observed that the switching threshold of 3 meters provides the best interception time results for environments with

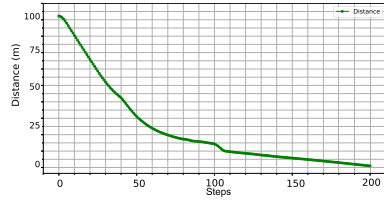
Table 7

Results in higher density environments with different threshold conditions. Best results are in bold.

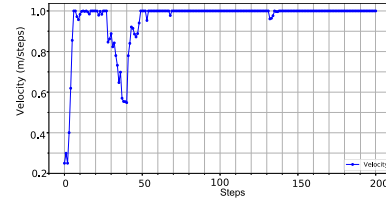
Switching Threshold s_{th} (m)	Success rate	Collision	None	Average time of interception (s)
3	99.6	0.3	0.1	5.63
7	100	0	0	6.22

10 obstacles. We test this switching threshold under a higher dense environment with 20 obstacles. The results are shown in Table 7.

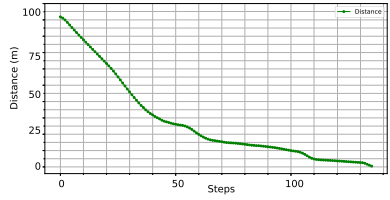
These results indicate that reducing the switching threshold in environments with high obstacle density compromises system safety. We slightly modify line 10 of Algorithm 1 to consider the switching threshold of 3 meters and in addition change a LiDAR scan from 0.5 to 2 to provide more space for manoeuvrability. The results are summarised in Table 8 which shows that this slight change ensures a 100% success rate with no collisions and almost homogeneous interception time in environments with different obstacle densities. Based on these results we can observe that when the switching condition is small the minimum distance of the LiDAR scan should be increased to ensure the interceptor has enough manoeuvrability to avoid collisions and ensure fast interception.



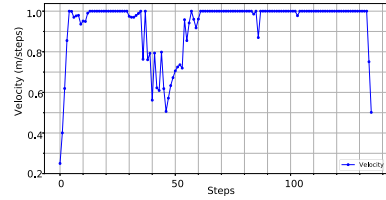
(a) Distance between agents: 15 obstacles



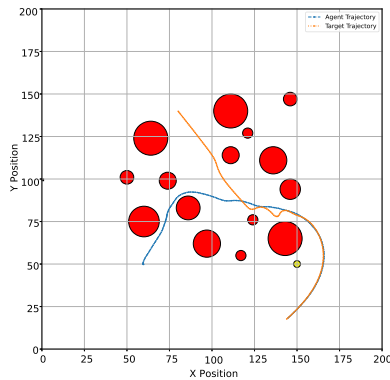
(b) Velocity interceptor: 15 obstacles



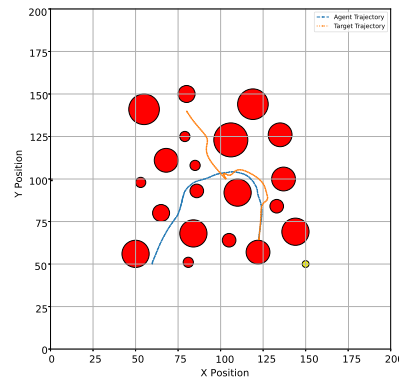
(c) Distance between agents: 20 obstacles



(d) Velocity interceptor: 20 obstacles

Fig. 10. TD3-APF generalisation with 15 and 20 obstacles.

(a) 15 obstacles



(b) 20 obstacles

Fig. 11. TD3-APF trajectory results.

Table 8

Results in higher density environments under different obstacle densities, $s_{th} = 3$ and $l_j = 2$.

Obstacle Density	Success rate	Collision	None	Average time of interception (s)
10	100	0	0	5.59
20	100	0	0	5.96

We additionally test the approach under polygonal obstacles. Fig. 12 shows the obtained results using 1000 random environments with 10 polygonal obstacles. We obtain a 98.30% success rate, 0.50% collisions, and 1.20% interception failure, with an average time of 5.99 s. We can observe that the interceptor is capable of anticipating the movement of the target regardless of the shape of the obstacles in most cases. The velocity profile adapts to avoid collisions due to the different obstacle shapes. However, there exists a risk of collision when switching to the APF since the obstacle shapes were not considered in the policy training. Here, the reward needs to be refined by considering the shape of the obstacles as well as the interception condition proposed in this paper. Nevertheless, the algorithm shows robustness and generalisation capabilities to different numbers of obstacles and shapes.

5.5. Failure cases and limitations

The proposed framework exhibits performance degradation in environments with increased obstacle density and more complex geometries (e.g., polygonal or concave shapes), which are not represented in the training distribution. In such scenarios, two primary failure modes are observed. First, the TD3 policy may generate suboptimal trajectories due to limited generalization, leading to inefficient navigation or collisions. This can be mitigated by modifying the switching threshold condition and the minimum distance of a LiDAR scan to have a more conservative switching condition that allows intercepting the target faster and

without collisions. Second, the switching mechanism may be triggered prematurely or under marginal line-of-sight conditions, particularly in the presence of irregular obstacle boundaries, resulting in unsafe transitions to the APF controller. These limitations highlight the sensitivity of the approach to distribution shift and geometric complexity. Future work will focus on improving robustness by enriching the training environments, incorporating safety margins in the switching condition, and extending the perception and control framework to better handle complex obstacle configurations.

6. Conclusions

This paper reports an interception approach for adversarial drones in urban zones using deep reinforcement learning. The evasion manoeuvres applied by the adversarial drone are governed by an artificial potential field (APF), whilst the interceptor is trained under a deep reinforcement learning architecture with APF in an urban zone with obstacles of varying densities. A simple condition for target interception is designed to ensure successful interception when the agent gets close to the target. In addition, a sparse reward function is designed to optimise the interception time, collision avoidance, and interception point prediction. Extensive experiments are conducted to demonstrate the benefits and challenges of the proposed model.

Future work will focus on improving the robustness and realism of the proposed framework. In particular, adversarial learning will be explored to enhance the interceptor's performance in dynamic scenarios, while imitation and inverse reinforcement learning will be used to model more realistic and adaptive target behaviors. Further extensions will also consider more complex obstacle environments, partial observability, and fully three-dimensional settings.

CRedit authorship contribution statement

Nicolas Lucotte: Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization.

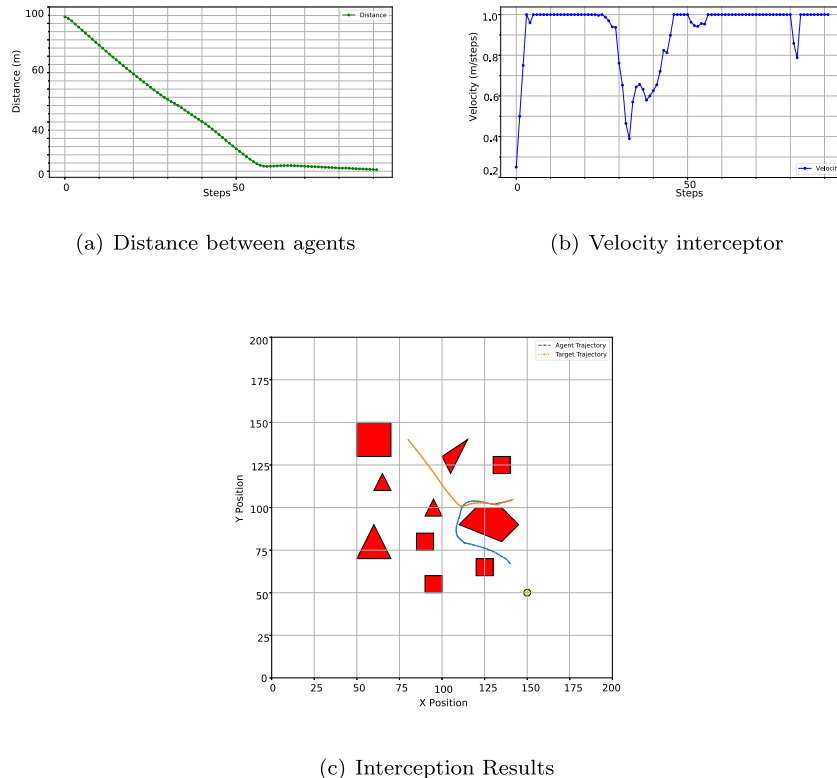


Fig. 12. TD3-APF generalisation under polygonal obstacles.

Adolfo Perrusquía: Writing – original draft, Supervision, Resources, Project administration, Methodology, Investigation, Conceptualization. **Antonios Tsourdos:** Writing – original draft, Supervision, Resources, Project administration, Funding acquisition, Conceptualization. **Weisi Guo:** Writing – original draft, Software, Resources, Funding acquisition. **Hyo-Sang Shin:** Visualization, Validation, Supervision, Funding acquisition.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data were used for the research described in the article.

References

- [1] J.-P. Yaacoub, H. Noura, O. Salman, A. Chehab, Security analysis of drones systems: attacks, limitations, and recommendations, *Internet Things* 11 (2020) 100218.
- [2] A. Perrusquía, W. Guo, B. Fraser, Z. Wei, Uncovering drone intentions using control physics informed machine learning, *Commun. Eng.* 3 (1) (2024) 36.
- [3] A.M. Ali, A. Perrusquía, W. Guo, A. Tsourdos, Flight plan optimisation of unmanned aerial vehicles with minimised radar observability using action shaping proximal policy optimisation, *Drones*, 2024.
- [4] S.N. Ashraf, S. Manickam, S.S. Zia, A.A. Abro, M. Obaidat, M. Uddin, M. Abdelhaq, R. Alsaqour, IoT empowered smart cybersecurity framework for intrusion detection in internet of drones, *Sci. Rep.* 13 (1) (2023) 18422.
- [5] Z. Oudina, M. Dourdour, A. Dib, M.M. Bouhamed, Empirical analysis of the security threats and risks that drones face, represent, and mitigation, in: 2024 6th International Conference on Pattern Analysis and Intelligent Systems (PAIS), IEEE, 2024, pp. 1–8.
- [6] U. Çelik, A. Perrusquía, UAV pursuit–evasion using a complementary scientific machine learning with control barrier functions approach, *IEEE J. Ocean. Eng.* 51 (2) (2026) 1075–1087.
- [7] E. Bildik, A. Tsourdos, A. Perrusquía, G. Inalhan, Decoys deployment for missile interception: a multi-agent reinforcement learning approach, *Aerospace* 11 (8) (2024) 684.
- [8] D. Gallacher, Drone applications for environmental management in urban spaces: a review, *International Journal of Sustainable Land Use and Urban Planning* 3 (4) (2016).
- [9] J. Rothe, M. Strohmeier, S. Montenegro, A concept for catching drones with a net carried by cooperative UAVs, in: 2019 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), IEEE, 2019, pp. 126–132.
- [10] M. Vrba, D. Heft, M. Saska, Onboard marker-less detection and localization of non-cooperating drones for their safe interception by an autonomous aerial system, *IEEE Robot. Autom. Lett.* 4 (4) (2019) 3402–3409.
- [11] I. Catalano, H. Sier, X. Yu, T. Westerlund, J.P. Queralta, UAV tracking with solid-state lidars: dynamic multi-frequency scan integration, in: 2023 21st International Conference on Advanced Robotics (ICAR), IEEE, 2023, pp. 417–424.
- [12] M. Khan, N. Salman, A. Ali, A. Khan, A. Kemp, A comparative study of target tracking with Kalman filter, extended Kalman filter and particle filter using received signal strength measurements, in: International Conference on Emerging Technologies (ICET), 2015, pp. 1–6.
- [13] B. Fraser, A. Perrusquía, D. Panagiotakopoulos, W. Guo, A deep mixture of experts network for drone trajectory intent classification and prediction using non-cooperative radar data, in: 2023 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2023, pp. 1–6.
- [14] C.-H. Lee, T.-H. Kim, M.-J. Tahk, Interception angle control guidance using proportional navigation with error feedback, *J. Guid. Control Dyn.* 36 (5) (2013).
- [15] Z. Huo, S. Dai, M. Yuan, X. Chen, X. Zhang, A reinforcement learning based multiple strategy framework for tracking a moving target, in: 2020 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), IEEE, 2020, pp. 1292–1297.
- [16] J.-F. Duhé, S. Victor, P. Melchior, Contributions on artificial potential field method for effective obstacle avoidance, *Fract. Calc. Appl. Anal.* 24 (2021) 421–446.
- [17] X. Meng, B. Sun, D. Zhu, Harbour protection: moving invasion target interception for multi-AUV based on prediction planning interception method, *Ocean Eng.* 219 (2021) 108268.
- [18] J. Xu, D. Panagopoulos, A. Perrusquía, W. Guo, A. Tsourdos, Generalising rescue operations in disaster scenarios using drones: a lifelong reinforcement learning approach, *Drones* 9 (6) (2025) 409.
- [19] W. Yu, A. Perrusquía, Human-Robot Interaction Control Using Reinforcement Learning, John Wiley & Sons, 2021.
- [20] K. Tamanakijprasart, A. Perrusquía, S. Mondal, A. Tsourdos, Autonomous path selection of unmanned aerial vehicle in dynamic environment using reinforcement learning, in: AIAA SCITECH 2025 Forum, 2025, pp. 1998.
- [21] C. Zhang, P. Cheng, B. Lin, W. Zhang, W. Xie, DRL-based target interception strategy design for an underactuated USV without obstacle collision, *Ocean Eng.* 280 (2023) 114443.
- [22] D. Fan, H. Shen, L. Dong, Switching-aware multi-agent deep reinforcement learning for target interception, *Appl. Intell.* 53 (7) (2023) 7876–7891.
- [23] J. Xue, S. Zhang, Y. Lu, X. Yan, Y. Zheng, Bidirectional obstacle avoidance enhancement-deep deterministic policy gradient: a novel algorithm for mobile-robot path planning in unknown dynamic environments, *Adv. Intell. Syst.* 6 (4) (2024) 2300444.
- [24] J. Xue, M. He, J. Chen, B. Dong, Y. Zheng, Improved DDPG based on enhancing decision evaluation for path planning in high-density environments, *Expert Syst. Appl.* 279 (2025) 127378.
- [25] J. Xue, J. Chen, S. Zhang, Action-curiosity-based deep reinforcement learning algorithm for path planning in a nondeterministic environment, *Intell. Comput.* 4 (2025) 0140.
- [26] C. Huang, Q. Zeng, J. Xu, Research on space proximity pursuit-evasion interception decision-making based on deep reinforcement learning, *Mechatronics* 110 (2025) 103387.
- [27] T. Gavin, S. Lacroix, M. Bronz, Agile interception of a flying target using competitive reinforcement learning, *arXiv preprint arXiv:2603.16279*, 2026.
- [28] X. Wang, H. Zhang, J. Xu, S. Wang, M. Guay, Reinforcement learning in pursuit-evasion differential game: safety, stability and robustness, *arXiv preprint arXiv:2507.19516*, 2025.
- [29] Z. Hu, L. Xiao, J. Guan, W. Yi, H. Yin, Intercept guidance of maneuvering targets with deep reinforcement learning, *Int. J. Aerosp. Eng.* 2023 (1) (2023) 7924190.
- [30] X. Li, S. Yu, Obstacle avoidance path planning for AUVs in a three-dimensional unknown environment based on the C-APF-TD3 algorithm, *Ocean Eng.* 315 (2025) 119886.
- [31] Y. Bi, X. Fang, A hybrid path planning framework integrating deep reinforcement learning and variable-direction potential fields, *Mathematics* 13 (14) (2025) 2312.
- [32] X. Chen, S. Liu, J. Zhao, H. Wu, J. Xian, J. Montevka, Autonomous port management based AGV path planning and optimization via an ensemble reinforcement learning framework, *Ocean Coast. Manag.* 251 (2024) 107087.
- [33] E. Thellier, A. Perrusquía, A. Tsourdos, Instance-specific heuristic learning for scalable UAV path planning, in: AIAA Aviation Forum and Ascend 2025, 2025, pp. 3397.
- [34] B. Wang, X. Shao, Y. Gao, J. Liu, Reinforcement learning-based enhanced fixed-time guidance law with no-fly zone constraints, *Defence Technology*, 2026.
- [35] X. Zhang, J. Yin, Z. Lin, C. Zhang, A positioning and orientation method based on the usage of INS and single-beam LiDAR, *Optik* 126 (22) (2015) 3376–3381.
- [36] D. Bhattacharjee, A. Chakravarthy, K. Subbarao, Nonlinear model predictive control based missile guidance for target interception, in: AIAA SciTech 2020 Forum, 2020, pp. 0865.
- [37] Y. Kartal, P. Kolaric, V. Lopez, A. Dogan, F. Lewis, Backstepping approach for design of PID controller with guaranteed performance for micro-air UAV, *Control Theory Technol.* 18 (1) (2020) 19–33.
- [38] J.A. Flores-Campos, C.R. Torres-San-Miguel, J.C. Paredes-Rojas, A. Perrusquía, Prescribed time interception of moving objects' trajectories using robot manipulators, *Robotics* 13 (10) (2024) 145.
- [39] A. Gohari, A.B. Ahmad, R.B.A. Rahim, A.S.M. Supa'at, S. Abd Razak, M.S.M. Gismalla, Involvement of surveillance drones in smart cities: a systematic review, *IEEE Access* 10 (2022) 56611–56628.
- [40] G. Xu, W. Jiang, Z. Wang, Y. Wang, Autonomous obstacle avoidance and target tracking of UAV based on deep reinforcement learning, *J. Intell. Robot. Syst.* 104 (4) (2022) 60.
- [41] S. Systems, Inertial solutions for UAV LiDAR & photogrammetry surveying, 2025, <https://www.sbg-systems.com/geospatial/uav-lidar-photogrammetry/> (Accessed: 2 November 2025).
- [42] V. Sonntag, A. Perrusquía, A. Tsourdos, W. Guo, A COLREGs compliance reinforcement learning approach for USV manoeuvring in track-following and collision avoidance problems, *Ocean Eng.* 316 (2025) 119907.
- [43] L. Xu, T. Wang, J. Wang, J. Liu, C. Sun, Attention-based policy distillation for UAV simultaneous target tracking and obstacle avoidance, *IEEE Transactions on Intelligent Vehicles*, 2023.
- [44] L. Linxiang, B. Xiaojun, Z. Di, Reinforcement learning-based guidance law for highly-maneuvering targets, *J. Syst. Eng. Electron.* (2026) 1–15, Early access.
- [45] S. Fujimoto, H. Hoof, D. Meger, Addressing function approximation error in actor-critic methods, in: International Conference on Machine Learning, PMLR, 2018, pp. 1587–1596.
- [46] B. Hazarika, K. Singh, S. Biswas, S. Mumtaz, C.-P. Li, Multi-agent DRL-based task offloading in multiple RIS-aided IoT networks, *IEEE Trans. Veh. Technol.* 73 (1) (2023) 1175–1190.
- [47] D.K. Panda, A. Perrusquía, W. Guo, Curriculum-guided antifragile reinforcement learning for secure UAV deconfliction under observation-space attacks, *arXiv preprint arXiv:2506.21129*, 2025.
- [48] A.J. Ellery, R.E. Baker, S.W. McCue, M.J. Simpson, Modeling transport through an environment crowded by a mixture of obstacles of different shapes and sizes, *Phys. Stat. Mech. Appl.* 449 (2016) 74–84.



DSPACE

<https://dspace.org/>

Interception of adversarial drones in urban spaces using a modified deep reinforcement learning approach

Lucotte, Nicolas; Perrusquía, Adolfo; Tsourdos, Antonios; Guo, Weisi; Shin, Hyo-Sang

2026-10

Elsevier

<https://doi.org/10.1016/j.asoc.2026.115797>

Lucotte N, Perrusquía A, Tsourdos A, et al., (2026) Interception of adversarial drones in urban spaces using a modified deep reinforcement learning approach. Applied Soft Computing, Volume 202, Part A, October 2026, Article number 115797