

USV Pursuit-Evasion using a Complementary Scientific Machine Learning with Control Barrier Functions Approach

Ugurcan Çelik, Adolfo Perrusquía, *Member, IEEE*

Abstract—The maritime pursuit-evasion (PE) problem is increasingly relevant to autonomous robotics and naval operations, particularly for security, surveillance, search and rescue, and environmental monitoring. Effective pursuit requires accurate evader behaviour prediction combined with robust obstacle avoidance in cluttered maritime environments. Traditional methods, including differential game theory and heuristic planning, often neglect realistic complexities and provide limited safety guarantees. Recent reinforcement learning approaches improve flexibility but struggle with generalization and formal safety assurance in complex scenarios. To bridge this gap, we propose a novel integration of Scientific Machine Learning (SciML) with Control Barrier Functions (CBFs), enabling provably safe pursuit and navigation under realistic vessel dynamics, partial observability, and non-convex obstacle constraints. Simulations validate ability of the proposed methods to achieve safe and effective pursuit in challenging maritime environments.

Index Terms—Cluttered maritime environments, autonomous decision making, pursuit evasion, scientific machine learning, control barrier functions

I. INTRODUCTION

In recent years, the maritime pursuit-evasion (PE) problem has gained increasing attention within the fields of autonomous robotics and naval operations, largely due to its applicability in security operations, surveillance, search and rescue missions, and environmental monitoring [1]–[4]. Such scenarios demand autonomous decision-making from USVs operating in cluttered, dynamic environments. Effective pursuit strategies in these settings require not only the ability to predict the evader’s behaviour but also to navigate safely around static and dynamic obstacles such as islands, other vessels, and restricted zones. Traditional pursuit-evasion approaches often neglect these real-world complexities [5]. A central challenge lies in balancing accurate interception with safe manoeuvring under environmental constraints.

A. Related Work

Although maritime pursuit-evasion has only recently attracted significant attention, it is a long-standing challenge in domains such as robotics, aerospace, and autonomous systems [6], [7]. To address collision avoidance in pursuit-evasion, various methods have been explored—ranging from potential field

approaches and velocity obstacles to sampling-based planners [8]–[10]. While effective in simple environments, these methods lack robustness in cluttered or dynamic settings and often provide no formal safety guarantees [11]. More recently, Control Barrier Functions (CBFs) have gained popularity for their ability to enforce formal safety constraints in non-linear control systems [12], [13]. CBFs provide a rigorous framework to ensure that system trajectories remain within safe sets, making them especially appealing for applications that require provable safety guarantees during execution. Applications in UAVs and robotic systems demonstrate their effectiveness in cluttered environments, though their integration with PE strategies remains under-explored.

Game theory has found widespread application in pursuit-evasion scenarios, often approached as a min-max optimization problem [6]. In [14], the authors examined a two-aircraft pursuit-evasion scenario in a horizontal plane and introduced a method using extremal trajectory maps. Meanwhile, the study in [15] tackled the problem by incorporating control input constraints in a numerical problem. The work in [16] explored a multi-agent pursuit-evasion game framework involving a prey, predator, and protector, aimed at addressing collision avoidance. A similar problem has been considered for two satellites, taking into account their reachable domains under continuous thrust conditions [17]. A further example of multiplayer differential game theory can be found in [18], which focuses on coordinating multiple UAVs to engage with multiple targets. In the maritime context, [19] discusses the application of game theory in various tasks related to USVs. While exact analytical solutions are sometimes obtainable under simplifying assumptions, implementing these solutions in real-time is often impractical due to its complexity.

In parallel, the maritime domain has seen a shift toward machine learning-based approaches, particularly reinforcement learning (RL), to manage the complexities of dynamic environments and underactuated vessel dynamics. RL represents a formalized learning approach based on trial and error, where a machine autonomously learns the optimal behaviour or policy for a specific task within a defined environment. This learning process involves maximizing or minimizing cumulative rewards or costs. An application of RL on maritime applications is to design a path following controller as the USV dynamics are usually not highly accurate and expose to high level of uncertainties [20]–[24]. RL is also utilized for collision avoidance tasks, overcoming optimization methods similarly due to unrealistic assumptions or imprecise vessel

U. Çelik, and A. Perrusquía and are with the Faculty of Engineering and Applied Sciences, Cranfield University, Bedford, UK (e-mail: {ugurcan.celik, adolfo.perrusquia-guzman}@cranfield.ac.uk}).

This work was supported by the Engineering and Physical Sciences Research Council under Grant 220124

models [25]–[28]. For the aim of this study, the pursuit evasion problem, there have been several applications for training a pursuer, training an evader, co-operative hunting [29]–[32]. RL for pursuit evasion game is also widely studied for robotics and UAVs for single/multi agent use cases [33]–[35]. One advantage of RL methods is their suitability for fast real-time inference. Furthermore, these methods enable the optimization of control loops, guidance loops, and obstacle avoidance loops collectively, mitigating performance degradation arising from interactions between subsystems [36], [37]. However, a notable drawback of RL methods lies in their training efficiency, which is highly contingent on the chosen reward functions [38]. Thus, considerable attention must be given to reward engineering. Additionally, for continuous systems, the agent’s exploration of the state space expands rapidly, making the search for global optimum solutions the most challenging aspect of Reinforcement Learning (RL)-based approaches [39].

Scientific Machine Learning (SciML), which embeds physical priors into learning architectures, offers improved generalization and interpretability, as demonstrated in tasks like precision landing and nonlinear control [40], [41]. However, like RL, SciML-based methods

B. Contributions

Based on the previous literature review, three main challenges are identified: (1) RL generalization to unseen and safety-critical scenarios remains limited, especially under complex dynamics and non-convex obstacles, (2) RL requires reward engineering to cover all multiple scenarios in the pursuer-evasion problem under congested maritime environments, and (3) both RL and SciML architectures often lack formal safety guarantees. To deal with the aforementioned gaps, we propose a novel framework that integrates SciML with Control Barrier Functions to enforce safety guarantees in a maritime pursuit-evasion problem. To best of our knowledge, this is the first attempt to integrate safety mechanisms into the SciML design in an adversarial scenario. The proposed method enables provably safe, real-time obstacle avoidance during pursuit, even under partial observability and occlusion, while accounting for realistic USV dynamics and supporting non-convex obstacle representations.

This study builds on our previous work [42], where we applied SciML to PE by embedding known USV dynamics into a universal ODE and learning unknown control terms through neural networks. Whilst previous approach focused on inter-agent collision avoidance and time-optimal capture, it did not address navigation in obstacle-rich environments under safety constraints. Here, we extend previous work with a CBF-based safety layer to ensure provably safe navigation in complex, real-world maritime settings while maintaining pursuit performance. The synergy between SciML’s generalization and CBF’s formal safety assurances results in a robust and scalable approach to autonomous pursuit in cluttered maritime domains. The main contributions of this work are twofold

- 1) A framework for USV Pursuit-Evasion that integrates CBF’s into the SciML formulation to enforce safety navigation in cluttered maritime environments.
- 2) A reinforcement learning intelligent evader is designed to compete with the SciML pursuer to improve the resilience and robustness of the final control policy in diverse scenarios and initial conditions.
- 3) An obstacle avoidance algorithm capable to deal effectively with convex and non-convex obstacles with different sizes and shapes.

The outline of the paper is organized as follows: Section II defines the pursuit-evasion problem and outlines the SciML architecture. Section 3 describes the obstacles avoidance formulation using CBFs. Section IV presents simulation results in varied scenarios. Finally, Section V concludes the paper.

II. PROBLEM STATEMENT AND METHODOLOGY

In this section, we present the methodology and framework for developing AI-aided defense tactics for a pursuer USV against an evader in a multi-obstacle environment. The baseline pursuing policy is formulated using scientific machine learning, and obstacle avoidance is integrated using the control barrier function approach.

A. Unmanned Surface Vehicle Model

To maintain tractability in the pursuit-evasion framework, each USV is modeled as a 2-DOF point mass with planar position, heading, and velocity states. Control inputs are longitudinal and rotational accelerations, constrained by physical limits (e.g., thruster capabilities). The point-mass dynamics is described by

$$\begin{aligned} \dot{x} &= V \cos \theta, & \dot{y} &= V \sin \theta, \\ \dot{V} &= a_v, & \dot{\theta} &= \omega, \end{aligned} \quad (1)$$

where $x, y \in \mathbb{R}$ denote the position coordinates, $V \in \mathbb{R}$ is the linear velocity magnitude, and $\theta \in \mathbb{R}$ is the heading angle. The control inputs a_v and ω represent the longitudinal acceleration and angular velocity, respectively.

Both the pursuer and the evader are subject to velocity-dependent maneuver limits: as speed rises, admissible surge acceleration and turn rate drop sharply. We calibrated these limits with proprietary Pacific 24 sea trial data (not publicly releasable); only the functional relationship - higher speed means lower agility - is required for reproduction of the results.

B. Scientific Machine Learning (SciML) preliminaries

Scientific machine learning (SciML) offers an alternative approach to classical machine learning control by embedding the agent’s governing physics directly into the training loop. Instead of learning a control policy purely from data or rewards, SciML treats the unknown portion of a physical model (in this case, the pursuer’s decision-making/control) as a learnable component within a system of differential equations. Formally, the known USV motion dynamics form the “mechanistic” part of the ordinary differential equation (ODE), while a neural network (NN) approximates the unknown terms.

In our previous work [42], we cast the pursuer dynamics as a Universal Differential Equation (UDE) such that: the

pursuer's state evolves through a combination of i) a physics-based model (referring to mechanistic model) and (ii) a NN-based policy $\pi_{SciML}(\cdot)$ that outputs the desired acceleration and angular velocity, i.e.,

$$\pi_{SciML}(\mathbf{X}(t), W_{NN}) = \begin{bmatrix} a_{vREF}(t) \\ w_{REF}(t) \end{bmatrix} \approx \mathcal{NN}(\mathbf{X}(t), W_{NN}), \quad (2)$$

where $\mathcal{NN}(\cdot)$ is a neural network, $\mathbf{X}(t) \in \mathbb{R}^D$ is the input feature vector, W_{NN} denotes the weights of the network. The UDE framework is described as:

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{V} \\ \dot{\theta} \end{bmatrix} = \underbrace{\begin{bmatrix} V \cos \theta \\ V \sin \theta \\ 0 \\ 0 \end{bmatrix}}_{\mathbf{g}_k(\mathbf{X}(t))} + \underbrace{\begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \mathcal{NN}^{(1)}(\mathbf{X}(t), W_{NN}) \\ \mathcal{NN}^{(2)}(\mathbf{X}(t), W_{NN}) \end{bmatrix}}_{\mathbf{g}_u(\mathbf{X}(t), W_{NN})}, \quad (3)$$

where the terms $\mathbf{g}_k(\cdot)$ and $\mathbf{g}_u(\cdot)$ represent the known and unknown components of the system dynamics which are consistent with the SciML formulation in [43], and $\mathbf{X}(t) = [x, y, V, \theta]^\top \in \mathbb{R}^4$ is the state vector. The term, and $\mathcal{NN}^{(i)}(\cdot)$ indicates the i th output of the policy network.

After formulating the UDE, we optimise the network weights by considering a physics-informed composite loss function comprising: tracking loss ($\mathcal{L}_{TT}$) to minimize deviation from the desired tracking range, control effort loss ($\mathcal{L}_{CE}$) to penalize high angular velocity, state constraint loss ($\mathcal{L}_{SC}$) to enforce state upper and lower limits, collision avoidance loss ($\mathcal{L}_{CA}$) to maintain safe distance, and astern approach loss ($\mathcal{L}_{AA}$) to guide pursuit from behind the evader. The composite physics-informed loss is given by

$$\mathcal{L} = \lambda_{TT} \mathcal{L}_{TT} + \lambda_{CE} \mathcal{L}_{CE} + \lambda_{SC} \mathcal{L}_{SC} + \lambda_{CA} \mathcal{L}_{CA} + \lambda_{AA} \mathcal{L}_{AA}, \quad (4)$$

where

$$\begin{aligned} \mathcal{L}_{TT} &= \sum_{k=0}^N |||\tilde{\mathbf{r}}_{p \rightarrow e}(k)| - D_r|, \\ \mathcal{L}_{CE} &= \frac{1}{N} \sum_{k=0}^N \omega_{ref}^2(k), \\ \mathcal{L}_{SC} &= \sum_{k=0}^N [||\text{ReLU}(\theta_{lb} - \theta(k))|| + ||\text{ReLU}(\theta(k) - \theta_{ub})||], \\ \mathcal{L}_{CA} &= \sum_{k=0}^N \text{ReLU}(|(D_{min} - ||\tilde{\mathbf{r}}_{p \rightarrow e}(k)||)|), \\ \mathcal{L}_{AA} &= \sum_{k=0}^N ||\text{ReLU}(-\tilde{r}_x^{EBF}(k))|| + \mathcal{M} ||\text{ReLU}(D_{SD} - |\tilde{r}_y^{EBF}(k)|)||, \end{aligned}$$

here k indexes the simulation steps ($N+1$ in total), $\tilde{\mathbf{r}}_{p \rightarrow e}$ is the relative position vector between the pursuer position $\mathbf{p}_p = [x_p, y_p]^\top$ and the evader position $\mathbf{p}_e = [x_e, y_e]^\top$, D_r is the desired tracking range, ω_{ref} denotes the reference yaw rate, θ is the pursuer's heading angle and θ_{lb}, θ_{ub} describe the lower and upper limits of the heading angle, D_{min} denotes the desired collision avoidance threshold, and $(\tilde{r}_x^{EBF}, \tilde{r}_y^{EBF})$ defines the evader body-frame coordinates used for astern pursuit, and

D_{SD} is the desired separation distance. $\mathcal{M}(\cdot)$ is a mask function which assigns a value of 0 when the pursuer is positioned behind the evader, and a value of 1 when the pursuer is ahead of the evader. The terms $\lambda_{TT}, \lambda_{CE}, \lambda_{SC}, \lambda_{CA}, \lambda_{AA} > 0$ are weights of each loss term.

The SciML policy training pseudo-algorithm is depicted in Fig. 1 and in Algorithm 1.

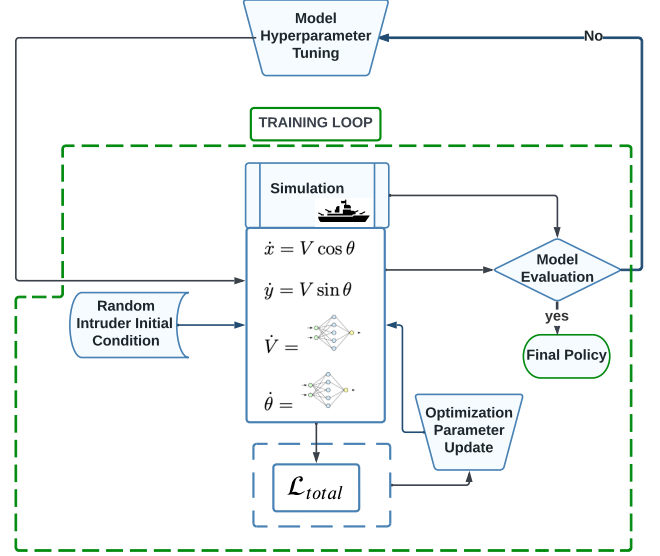


Fig. 1. Policy development flowchart for SciML

Algorithm 1 begins with a predefined grid of pursuer initial conditions. The neural network that represents the control policy is randomly initialised, after which the grid is split into mini-batches. For each batch, an evader state is sampled—heading and speed held constant to emulate a non-maneuvring target—and, with fixed network weights, the resulting UDE is integrated in parallel to generate trajectory roll-outs. The composite loss (4) is then evaluated on every trajectory, and its gradient with respect to the network parameters is obtained by back-propagating through the differentiable solver. A first-order optimiser updates the weights, a new set of mini-batches is drawn, and the process repeats until the loss converges. The resulting policy satisfies the pursuit-evasion objectives while respecting dynamic and safety constraints. Details of grid construction, hyper-parameter settings, and other implementation choices follow exactly those reported in previous work [42].

1) *Policy architecture*: We design a feature vector $\mathbf{X}(t) \in \mathbb{R}^{11}$ composed of the following terms

$$\mathbf{X}(t) = \left[\tilde{\mathbf{r}}_{p \rightarrow e}^I, \tilde{\mathbf{r}}_{p \rightarrow e}^{EBF}, \dot{\mathbf{r}}_{p \rightarrow e}, ||V_p||, V_p, V_e, \dot{\lambda}_{LOS_{p \rightarrow e}} \right]^\top,$$

where $\tilde{\mathbf{r}}_{p \rightarrow e}^I(t)$, $\tilde{\mathbf{r}}_{p \rightarrow e}^{EBF}(t)$ denote the relative distance between the pursuer and the evader expressed in the inertial reference frame and the evader body fixed reference frame, respectively. $\dot{\mathbf{r}}_{p \rightarrow e}(t)$ indicates the time derivative of the relative position vector, highlighting the dynamic distance changes between pursuer and evader over time. The velocity vectors of the pursuer and evader are denoted as $V_p(t)$ and $V_e(t)$, with $||V_p(t)||$ specifying the magnitude of the pursuer's velocity.

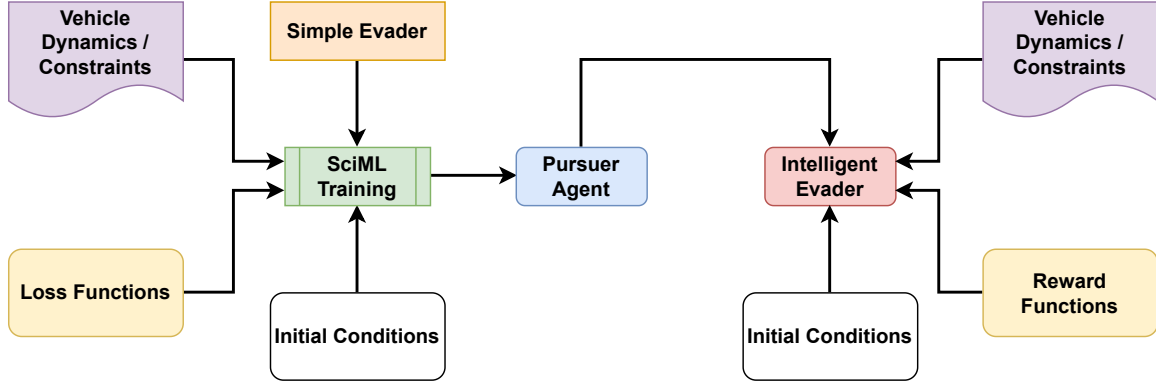


Fig. 2. Policy development flowchart Intelligent Evader

Algorithm 1 SciML minibatch training algorithm

```

1: Input: ODE Solver(ODE, IC's), Epoch Size: N, Mini-
   Batch Size: M
2: Data: Initial Conditions Grid for Pursuer
3: Initialization: Initilization of Network Weights,  $W_{NN}$ 
4: Optimization Variable:  $W_{NN}$ 
5: for  $epoch = 1, 2, \dots, N$  do
6:   Generate Minibatches
7:   for  $Minibatch = 1, 2, \dots$  do
8:     for Pursuer IC's in Minibatch do
9:        $x_e(t_0) \leftarrow$  Sample the Evader's Initial Condition
10:       $x(t)_i, u(t)_i \leftarrow simulation(IC_i, x_e(t_0), W_{NN})$ 
11:      Compute Loss,  $\mathcal{L}_i \leftarrow x_i(t)$ 
12:      Compute Gradient,
13:       $\left(\frac{d\mathcal{L}_i}{dW_{NN}}\right)_i \leftarrow \mathcal{L}_i, x_i(t), u_i(t), W_{NN}$ 
14:    end for
15:    for  $i$  in Minibatch do
16:      Gradient Descent with Adam Optimizer :
17:       $W_{updated} \leftarrow W_{NN} + \alpha \left(\frac{d\mathcal{L}}{dW_{NN}}\right)_i$ 
18:    end for
19:  end for

```

Lastly, $\dot{\lambda}_{LOS_{p \rightarrow e}}(t)$ represents the rate of change of the line-of-sight angle between the pursuer and the evader, capturing the angular dynamics of their engagement.

TABLE I
NEURAL NETWORK POLICY ARCHITECTURE. NA-TANH STANDS TO A NEURONWISE ADAPTIVE TANH FUNCTION.

Layer	Type	# units	Activation
Input	Scaling	11	-
Layer 1	Dense	16	Tanh
Layer 2	Dense	16	Tanh
Layer 3	Dense	16	NA-Tanh
Output	Re-Scaling	2	-

The control policy is realised by the feed-forward network reported in Table I, which employs adaptive-tanh activations. In earlier work, the resulting policy achieved reliable interception and target-collision avoidance across a broad set of initial

conditions, yet it lacked guarantees in cluttered waters. To address this limitation, we augment the policy with a control-barrier-function (CBF) layer that enforces provable obstacle avoidance without materially altering baseline performance.

C. Intelligent Evader

In this paper, we design an intelligent evader to compete against a pursuer agent guided by a SciML-based policy, aiming to assess and refine the effectiveness of the pursuer agent's strategy.

The proposed intelligent evader training framework is shown in Fig. 2. This framework consists of two main components: The left side of the flowchart outlines the policy training process for a pursuer trained via scientific machine learning to counter a simple evader. This training results in the development of an intelligent pursuer. This pursuer then forms the basis for training an intelligent evader. The right side of the flowchart focuses on the policy optimization process for the evader, highlighting that the evader shares the same system dynamics and constraints as the pursuer.

The evader training is based on a Proximal Policy Optimisation (PPO) algorithm which is a widely used policy gradient method in reinforcement learning that addresses the stability and efficiency of policy updates.

Remark 1: The pursuer's SciML policy can be optionally fine-tuned by training it against the intelligent evader policy in an adversarial learning setup. This process, similar to classical adversarial training, helps the pursuer adapt to the evader's evolving tactics. Although this step was not required in the present work, preliminary trials informed the feature vector design used in the main experiments.

1) *Reinforcement Learning Environment:* In the RL environment, we employed the same dynamics for both the pursuer and evader, as detailed in (3). This time, the pursuer operates under a neural policy optimized using SciML, which is kept frozen. Meanwhile, the evader begins with a randomly initialized policy that is progressively refined through RL. The complete flow of the environment is illustrated in Figure 3.

The flowchart illustrates that the evader and pursuer's system dynamics are updated simultaneously at each time step. The evader executes actions—commanded longitudinal

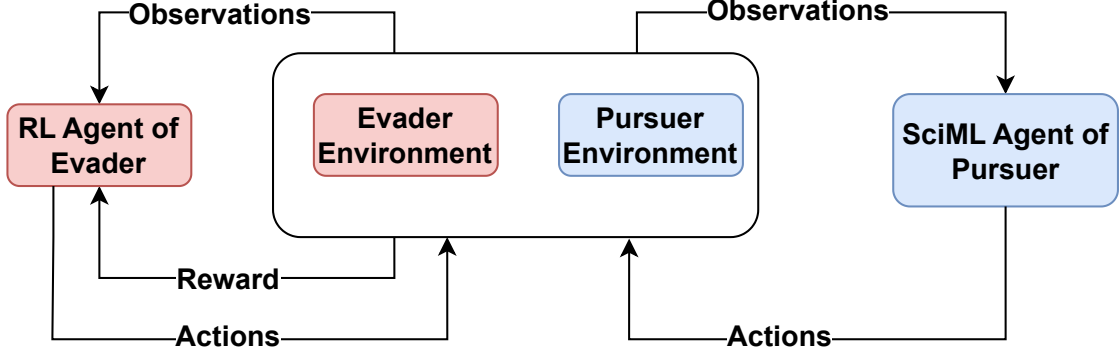


Fig. 3. Intelligent Evader RL Environment

accelerations and angular velocity based on the relative state vector with the pursuer. Simultaneously, the pursuer acts based on its relative states. The evader then receives rewards based on environment observations, detailed in the next section. At each trajectory's end, rewards are compiled and fed into the PPO algorithm to update the evader's policy, whereas the pursuer's policy remains unchanged.

2) *Reward Function*: The design of the reward function is crucial in optimizing the reinforcement learning policy, especially for implementing the evading strategy. To facilitate this strategy, we have formulated a total reward function, denoted as r_{total} , which consists of two primary components: a running range reward r_{RR} , which pertains to the evader's distance from the pursuer, and r_{inf} , which is tied to the evader's distance from the infrastructure.

$$r_{total} = r_{RR} + r_{inf}. \quad (5)$$

The reward r_{RR} acts as a penalty when the evader enters the danger zone of being captured by the pursuer. Specifically, when the evader is within the running range (RR), it incurs a continuous reward of -1 , signalling increased risk of capture.

$$r_{RR} = \begin{cases} -1, & \text{if } \|\tilde{\mathbf{r}}_{p \rightarrow e}\| \leq RR \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

The reward r_{inf} is designed to motivate the evader to approach and potentially hit the infrastructure, granting a $+100$ reward when the evader is within the hitting range (HR), thereby completing its mission. Conversely, being outside the allowable hitting range results in quadratic penalties, emphasizing the importance of mission completion.

$$r_{inf} = \begin{cases} +100, & \text{if } \|\tilde{\mathbf{r}}_{e \rightarrow inf}\| \leq HR \\ -\max\left(\left(\frac{\|\tilde{\mathbf{r}}_{e \rightarrow inf}\| - HR}{100}\right)^2, 2\right), & \text{otherwise.} \end{cases} \quad (7)$$

We use the $\max(\cdot)$ function to prioritize evasion of the infrastructure over the risk of capture. This approach ensures that penalties for failing to evade the infrastructure are more significant, reinforcing the primary objective of mission success while managing the threat posed by the pursuer.

III. OBSTACLE AVOIDANCE WITH CONTROL BARRIER FUNCTIONS

To ensure obstacle avoidance, we have integrated our baseline policy, derived using SciML, with the control barrier function approach. This combination allows us to enforce safety constraints dynamically.

Considering a non-linear affine system of the form,

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \quad \mathbf{x} \in \mathbb{R}^n, \mathbf{u} \in \mathbb{R}^m. \quad (8)$$

Let $\mathcal{S} = \{\mathbf{x} : h(\mathbf{x}) \geq 0\}$ be the safe set. h is a *control barrier function* (CBF) if $\exists \alpha \in \mathcal{K}_\infty$ such that [12], [13]

$$\sup_{\mathbf{u}} L_{\mathbf{f}}h(\mathbf{x}) + L_{\mathbf{g}}h(\mathbf{x})\mathbf{u} + \alpha(h(\mathbf{x})) \geq 0, \quad (9)$$

where $L_{\mathbf{f}}h$ and $L_{\mathbf{g}}h$ are Lie derivatives of h along $\mathbf{f}$ and $\mathbf{g}$. Safety is enforced via the quadratic program

$$\min_{\mathbf{u}} (\mathbf{u} - \mathbf{u}_b)^\top \mathbf{H}(\mathbf{u} - \mathbf{u}_b) \quad \text{s.t. (9)}. \quad (10)$$

From the CBF definition and the quadratic program (QP) formulation, it becomes clear that when $L_{\mathbf{g}}h(\mathbf{x}) = 0$ it is impractical to find a safe controller using (9) or to implement a QP as defined in (10). This issue arises because the chosen function $h(\mathbf{x})$ possesses a high relative degree with respect to the system dynamics given in (8). A function h that is continuously differentiable is said to have a control input relative degree m with respect to system if the first occurrence of the control input $\mathbf{u}$ in the derivatives of h along the system's dynamics happens in its m th derivative [44].

High Order Control Barrier Function (HOCBF) were introduced to manage high relative degree barrier functions [45]. This method enables the synthesis of a controller that ensures system forward invariance, denoted as $\forall \mathbf{x} \in \mathcal{S}$ for $t > t_0$, using a time-invariant HOCBF. The synthesis process effectively maintains the system's state within the safe set $\mathcal{S}$ beyond the initial time t_0 .

$$\sup_{\mathbf{u}} L_{\mathbf{f}}^m h + L_{\mathbf{g}} L_{\mathbf{f}}^{m-1} h \mathbf{u} + O(h(\mathbf{x})) + \alpha_m(\psi_{m-1}) \geq 0, \quad (11)$$

with the recursive sequence

$$\begin{aligned} \psi_0(\mathbf{x}) &= h(\mathbf{x}), \\ \psi_{i+1}(\mathbf{x}) &= \dot{\psi}_i(\mathbf{x}) + \alpha_{i+1}(\psi_i(\mathbf{x})), \quad i = 0, \dots, m-1, \end{aligned} \quad (12)$$

here $\alpha_i \in \mathcal{K}_\infty$ and m is the relative degree.

A. Obstacle Avoidance for Pursuit-Evasion Problem with Convex Circular Obstacles

The functions $\mathbf{f}(\mathbf{x})$ and $\mathbf{g}(\mathbf{x})$ in (3) are defined as follows $\mathbf{f}(\mathbf{x}) = [V \cos \theta, V \sin \theta, 0, 0]^\top$ and $\mathbf{g}(\mathbf{x}) = [0_2, \mathbf{I}_2]^\top$.

For convex circular obstacles, the safe set can be mathematically formulated as follows. Consider an obstacle represented as a perfect circle with a centre at (c_x, c_y) and radius r . The safe set $\mathcal{S}$ for a point $(x_p, y_p) = (x, y)$ representing the position of the pursuer in relation to this obstacle can be defined by the inequality:

$$\mathcal{S} = \{(x, y) \in \mathbb{R}^2 : h(x, y) > 0\}, \quad (13)$$

where $h(x, y) = (x - c_x)^2 + (y - c_y)^2 - r^2$.

The condition $h(x, y) > 0$ ensures that the pursuer remains outside the obstacle, maintaining safety by avoiding collisions. However, upon analysing the chosen CBF candidate, it becomes apparent that $L_{\mathbf{g}}h(x, y) = 0$, this characteristic categorizes the barrier function as a "high relative degree" function. Specifically, the barrier function is of the second order relative to the system, with $m = 2$. Hence, we have followed to HOCBF approach and implemented QP-HOCBF to derive a safe controller.

The core objective of this QP-HOCBF optimization is to identify a control commands (reference longitudinal and angular accelerations) that minimally deviates from the reference accelerations commanded from our baseline policy, π , while simultaneously ensuring that the system remains within the safe set defined by HOCBF, $\mathcal{S}$, thus avoiding obstacles.

Furthermore, we have extended QP-HOCBF formulation to accommodate multi-obstacle scenarios by incorporating additional constraints for each obstacle, $h_1(x, y), h_2(x, y), \dots, h_n(x, y)$, where n denotes the total number of obstacles in the environment.

Additionally, in safety-critical systems, accounting for input constraints is vital, as saturation may impair the performance of automatic control subsystems. Nevertheless, incorporating control bounds into a QP can occasionally yield infeasible solutions. To mitigate this, we relaxed the QP-HOCBF optimization by introducing slack variables, ensuring the feasibility of solutions under all conditions.

The overall QP-HOCBF optimization problem is given by

$$\begin{aligned} \min_{\mathbf{u}, \mathbf{s}_u, \mathbf{s}_l} \quad & J = (\mathbf{u} - \pi_{\text{SciML}})^\top \mathbf{H}(\mathbf{u} - \pi_{\text{SciML}}) + \lambda_u \|\mathbf{s}_u\|^2 + \lambda_l \|\mathbf{s}_l\|^2 \\ \text{s.t.} \quad & L_{\mathbf{f}}^2 h_1(x, y) + L_{\mathbf{g}} L_{\mathbf{f}} h_1(x, y) \mathbf{u} + \alpha_2(\dot{h}_1(x, y) + \alpha_1(h_1(x, y))) \geq 0 \\ & L_{\mathbf{f}}^2 h_2(x, y) + L_{\mathbf{g}} L_{\mathbf{f}} h_2(x, y) \mathbf{u} + \alpha_2(\dot{h}_2(x, y) + \alpha_1(h_2(x, y))) \geq 0 \\ & \vdots \\ & L_{\mathbf{f}}^2 h_n(x, y) + L_{\mathbf{g}} L_{\mathbf{f}} h_n(x, y) \mathbf{u} + \alpha_2(\dot{h}_n(x, y) + \alpha_1(h_n(x, y))) \geq 0 \\ & \mathbf{u} < \mathbf{u}_{\text{upper}} + \mathbf{s}_u \\ & \mathbf{u}_{\text{lower}} < \mathbf{u}_{\text{lower}} + \mathbf{s}_l, \end{aligned} \quad (14)$$

where $\mathbf{H}$ is a positive definite matrix that typically serves to weight the terms in the control inputs, ensuring a convex formulation. The variables $\mathbf{s}_u \in \mathbb{R}^2$ and $\mathbf{s}_l \in \mathbb{R}^2$ are slack variables used to relax the upper and lower constraints on controls, respectively, allowing a feasible solution space when strict control bounds might not be directly achievable. λ_l, λ_u are the coefficients used to balance the importance of

the corresponding components within the objective function. $\mathbf{u}_{\text{upper}} \in \mathbb{R}^2$ and $\mathbf{u}_{\text{lower}} \in \mathbb{R}^2$ denotes the upper and lower bounds for the controls.

B. Obstacle Avoidance for Non-Convex Shapes

The HOCBF formulation (and the QP-based HOCBF in (14)) assumes circular obstacle shapes. However, in real maritime environments, obstacles often deviate from circular forms, and their locations may be initially unknown. For real-time detection, our USVs are equipped with a LiDAR sensor. This sensor emits laser rays in a known angular configuration, and each ray's intersection point provides the obstacle's range and bearing.

As illustrated in Fig.4, collecting these intersection points across all rays yields a discrete representation of the obstacle boundaries surrounding the vessel.

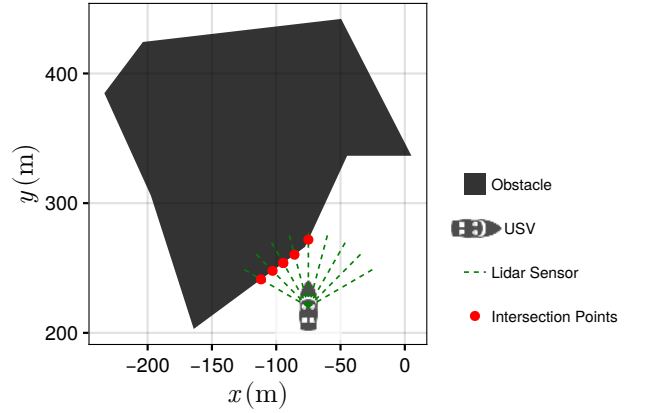


Fig. 4. Lidar Sensor Illustration

In this study, as a proof of concept, we have assumed that USVs are equipped with a LiDAR sensor which has a detection range of 60 meters with an angular configuration of $\pm 60^\circ$ laser rays. As a practical solution for unknown, non-convex obstacles, the agent constructs virtual circular barriers centered at each laser-ray intersection point, and applies the same QP-HOCBF formulation from (10). The radius of these virtual obstacles, r_{virtual} is chosen based on the USV's dynamic characteristic; here, it is set to half the sensor range (30m), providing a practical safety margin for real-time obstacle avoidance.

IV. RESULTS

In this section we provide extensive simulation studies to demonstrate the feasibility of the approach for evader pursuing with and without obstacles. All the simulation studies were developed in Julia using an 32 GB-RAM ThinkPad with Intel(R) Core(TM) Ultra 7 155H with 1.40 GHz endowed with an NVIDIA RTX 2000 Ada Generation GPU.

The hyper-parameters of the SciML policy were optimised by using a grid search algorithm. The complete training set-up and the optimized hyper-parameter values are summarized in Table II.

TABLE II
TRAINING SETUP AND HYPERPARAMETERS

Object	Component/Description
ODE Solver	Tsit5 (Tsitouras 5/4 Runge-Kutta method)
	Adjoint Method $\rightarrow$ Interpolating Adjoint
	$\Delta t \rightarrow$ Variable
	Time Span=[0, 80], s
Cost Coefficients	Absolute/Relative Tolerances = 10^{-5}
	$\lambda_{TT} = \frac{t}{8}$
	$\lambda_{CE} = 0.05, \lambda_{CA} = 1$
	$\lambda_{SC} = 100, \lambda_{AA} = 0.3$
Cost Parameters	$D_r = 20, m, D_{min} = 10, m$
	$D_{SD} = 20, m$
Optimizer	ADAM
	Mini-Batch Size, $M = 1$
	Learning Rate, $\zeta = 0.0003$
	Momentum decay $\beta = (0.9, 0.999)$

A. Evasion Scenario

In this section, we present the simulation outcomes for a pursuer governed by the proposed SciML-based policy engaging an intelligent evader that strategically approaches a protected national infrastructure, assumed to be located at the origin. Two representative scenarios are examined. In the first, the pursuer and evader possess comparable dynamic capabilities, allowing for a balanced engagement. In the second, the evader exhibits superior performance, as the pursuer's maximum velocity is deliberately constrained to be lower, highlighting the policy's robustness under disadvantageous conditions.

1) *Scenario 1: Intelligent Evader with Equal Capability:* In this scenario, both agents possess comparable dynamic capabilities. The initial conditions are set as $\mathbf{x}_p(0) = [-100, 40, 0, 140^\circ]^\top$ for the pursuer and $\mathbf{x}_e(0) = [-400, 540, 17, -120^\circ]^\top$ for the evader.

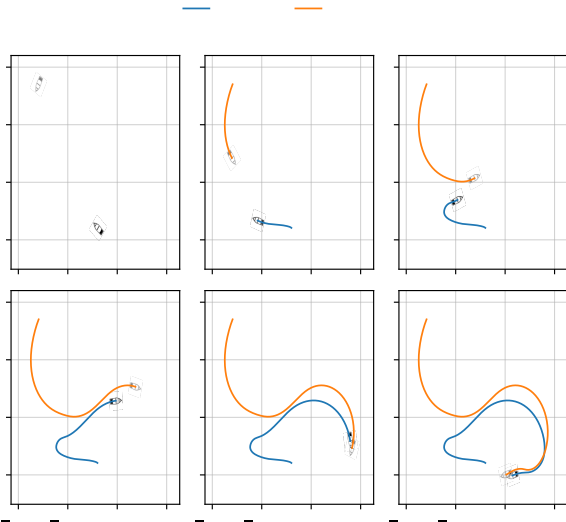


Fig. 5. Pursuer and intelligent evader elapsed time trajectories

Fig. 5 traces the intelligent evader (orange) driving toward the infrastructure and the pursuer (blue) executing a lead-turn with a tighter curvature to cut across the evader's path. After

the intercept geometry is achieved, the pursuer slides astern and maintains line-of-sight, ensuring continuous tracking without overshoot. The range history in Fig. 6 shows a smooth, steadily decreasing separation that levels off inside the capture envelope, confirming timely interception and sustained track throughout the engagement.

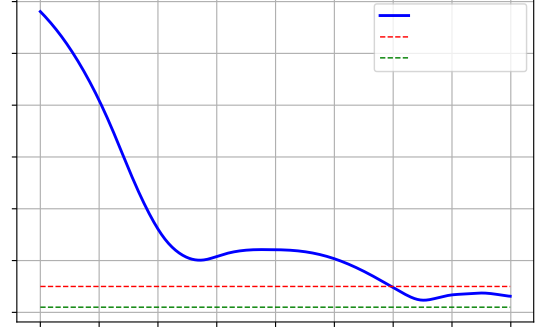


Fig. 6. Relative distance between pursuer and evader for evasion scenario

Fig. 7 plots the normalized longitudinal-acceleration and yaw-rate commands for both agents. The left column (blue) corresponds to the pursuer, the right column (orange) to the intelligent evader; in each subplot the red dashed envelope denotes the admissible range of $[-1, 1]$. Because all variables are nondimensionalised for confidentiality reasons, ± 1 map to the vehicle-specific actuator limits.

The pursuer repeatedly drives each channel to its extrema, switching sign in a bang-bang fashion that is characteristic of time-optimal interception. In contrast, the evader's profiles remain closer to mid-range for most of the engagement, then swing rapidly to the positive limit during its terminal escape manoeuvre. Throughout the scenario neither agent saturates beyond the admissible envelope, confirming that both learned policies honour the embedded dynamic constraints while executing their respective guidance strategies.

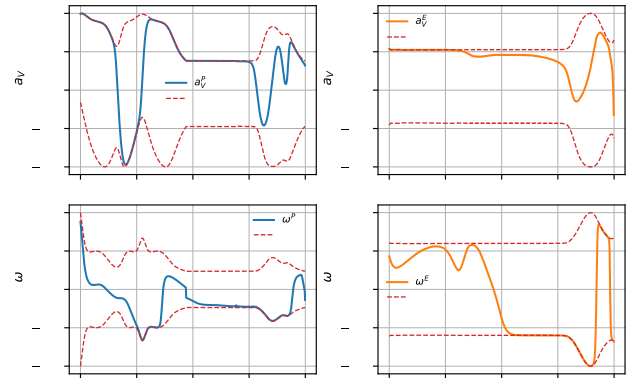


Fig. 7. SciML Policy control actions over time for evasion scenario

2) *Scenario 2: Intelligent Evader with Superior Capability:* In this case, the evader is dynamically superior, with its maximum velocity exceeding that of the pursuer by approximately

25%. The pursuer's maximum velocity is therefore limited to about 75% of the evader's. The initial conditions are $\mathbf{x}_p(0) = [0, 64, 0, 20^\circ]^\top$ and $\mathbf{x}_e(0) = [265, 551, 12, -100^\circ]^\top$.

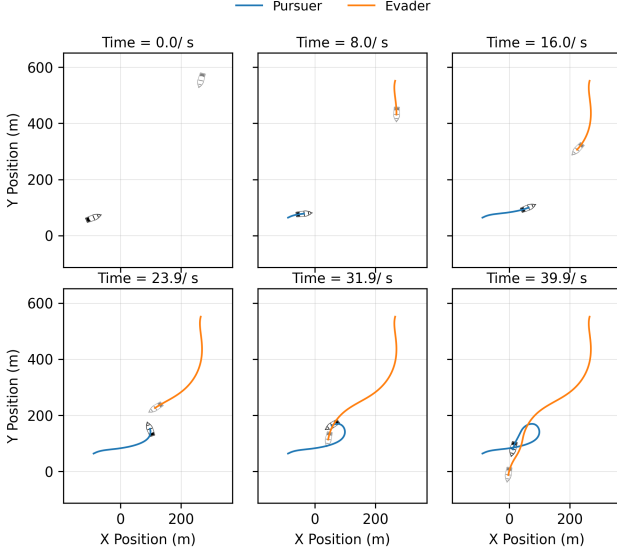


Fig. 8. Pursuer and superior evader elapsed time trajectories

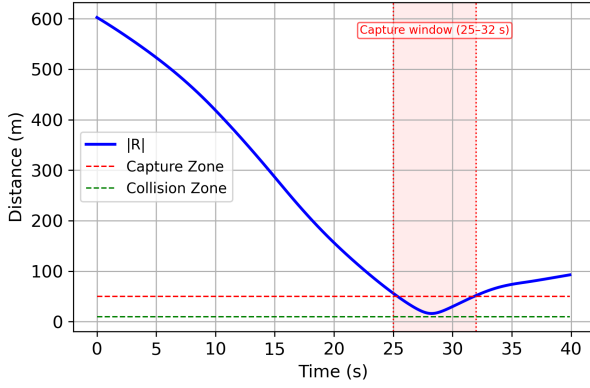


Fig. 9. Relative distance between pursuer and superior evader for evasion scenario

As illustrated in Fig. 8, when the evader possesses superior kinematic capability, the pursuer executes a sequence of well-timed lead-turn and intercept maneuvers to exploit transient geometric advantages. The capture window indicated in Fig. 9 (25–32 s) highlights a brief interval during which the relative distance falls within the capture envelope before increasing again as the evader accelerates toward the protected infrastructure. If interception is not achieved within this interval, the pursuer loses any further opportunity for capture due to its velocity limitation (approximately 75 % of the evader's maximum speed).

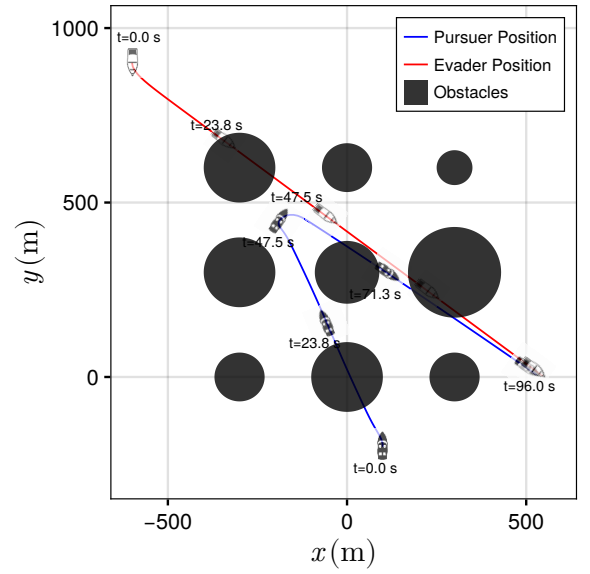
B. Collision Avoidance

This section presents the experimental results evaluating the proposed avoidance strategy in multi-obstacle environments. Unlike the previous scenarios that focused on protecting a

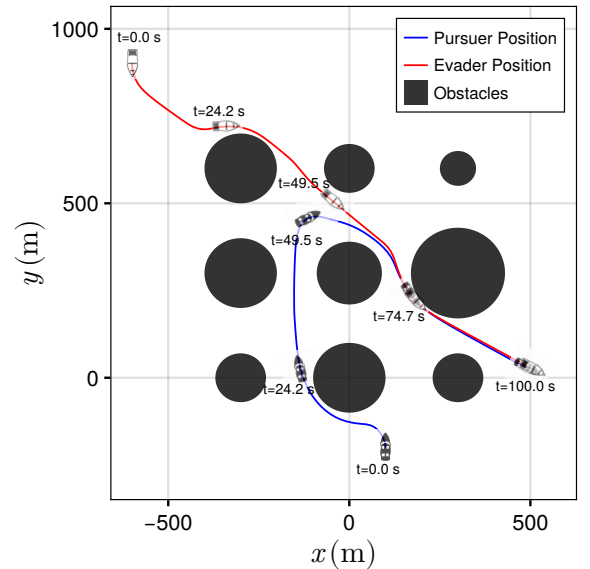
fixed infrastructure, the objective here is to assess the pursuer's ability to capture an evader while safely navigating through cluttered regions. Two representative cases are considered: (1) pursuit in the presence of initially known, convex obstacles; and (2) pursuit amid initially unknown, irregularly shaped obstacles detected online.

1) Scenario 1: Pursuit with Known Convex Obstacles:

In this scenario, nine circular-shaped obstacles are placed in known locations. The SciML policy is combined with a HOCBF layer, configured such that the pursuer considers only the two nearest obstacles at any given time which reduces computational overhead for real-time applications. This allows the agent to navigate safely through the environment while maintaining its objective of capturing the evader.



(a) Trajectory without CBF Layer



(b) Trajectory with CBF Layer

Fig. 10. Trajectories for Scenario-1 with and without CBF

Fig.10 compares the pursuer-evader trajectories under two settings: in Fig.10(a), the pursuer uses only the baseline SciML policy, while in Fig.10(b), a HOCBF layer is integrated for obstacle avoidance. The evader employs a HOCBF-enhanced P-controller for illustration. Without the HOCBF layer, the SciML policy guides the pursuer through obstacles due to the absence of explicit obstacle avoidance, except for inter-agent collision avoidance encoded during policy training. In contrast, with the HOCBF layer, the pursuer adjusts its trajectory to avoid obstacles while effectively tracking the evader, even in complex environments. As shown in Fig.11, the relative distance between the pursuer and evader steadily decreases and stabilizes within the safe capture zone, confirming successful and safe interception under known multi-obstacle conditions.

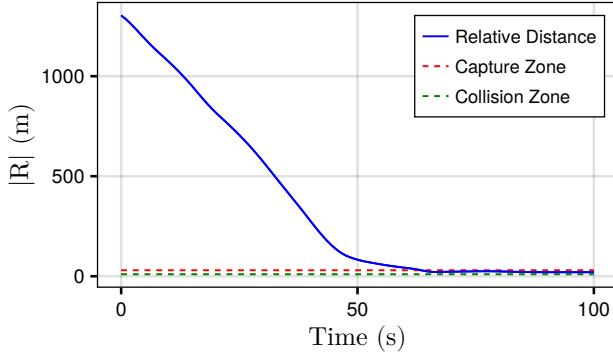


Fig. 11. Relative distance between pursuer and evader over time for scenario-1

Fig.12 shows the time evolution of both the SciML policy's commanded reference accelerations (dashed blue lines) and the accelerations adjusted by the HOCBF layer (solid blue lines) for obstacle avoidance. The dashed red lines denote the vehicle's dynamic control limits. Whenever the pursuer nears an obstacle, the HOCBF layer deviates from the SciML outputs, while still remaining within the feasible acceleration bounds. Otherwise, it closely follows the original policy commands, demonstrating that the HOCBF layer intervenes only as needed to ensure safe navigation.

To evaluate the proposed algorithm's robustness under varying initial conditions and enable fair comparison with conventional methods, we conducted an extensive Monte Carlo study with 500 randomized trials. The baseline Artificial Potential Field (APF) method was carefully tuned through systematic parameter optimization to ensure equitable comparison grounds.

Remark 2: Recent studies [46] have shown that the Artificial Potential Field (APF) method achieves more reliable collision avoidance performance in cluttered pursuit-evasion scenarios than several state-of-the-art reinforcement learning (RL) approaches. This performance gap largely stems from the sensitivity of RL methods to reward design and training distribution, which often limits their generalization in dense or highly dynamic environments. Motivated by these findings, we employ the APF method as a strong and interpretable baseline to benchmark the proposed SciML-CBF framework under comparable conditions.

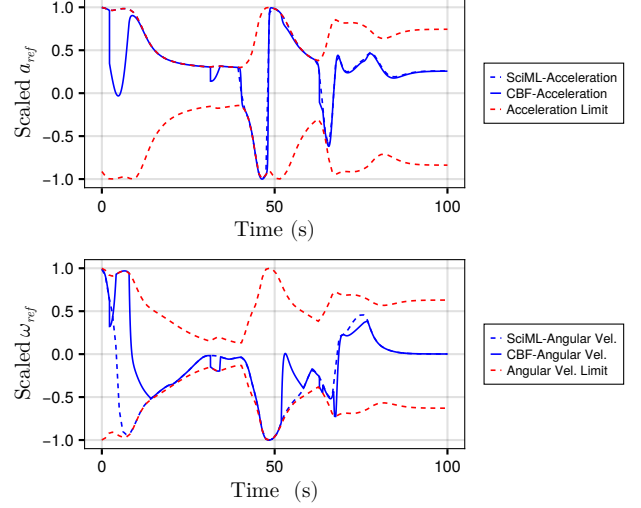


Fig. 12. Policy control actions over time for Scenario-1

The artificial potential field (APF) formulation integrates target attraction and velocity-adaptive repulsion through:

$$\mathbf{F}_{\text{APF}} = \underbrace{k_{\text{att}} \tilde{\mathbf{r}}_{p \rightarrow e}}_{\text{Attraction}} + \sum_{i=1}^N \underbrace{\frac{k_{\text{rep}}}{(d_i - r_i^{\text{eff}})^{\gamma}} \left[\frac{\tilde{\mathbf{r}}_{p \rightarrow o_i}}{d_i} + \lambda_t \frac{\boldsymbol{\omega}_i}{\|\boldsymbol{\omega}_i\|} \right]}_{\text{Repulsion}}, \quad (15)$$

where $\tilde{\mathbf{r}}_{p \rightarrow e}$ and $\tilde{\mathbf{r}}_{p \rightarrow o_i}$ represent unit vectors from pursuer to evader and i th obstacle respectively, k_{att} and k_{rep} govern attraction/repulsion balance, r_i^{eff} denotes effective obstacle radius with safety margin, $\gamma > 1$ controls repulsion gradient steepness, and $\lambda_t \frac{\boldsymbol{\omega}_i}{\|\boldsymbol{\omega}_i\|}$ introduces velocity-dependent tangential guidance for smooth obstacle circumvention. Control inputs for velocity and heading are derived from the resultant potential field through proportional-derivative (PD) techniques, ensuring smooth trajectory generation.

Fig.13 visualizes the Monte-Carlo trajectories (solid lines denote the pursuer trajectories), while Table III summarizes key metrics.

TABLE III
PERFORMANCE COMPARISON BETWEEN PROPOSED AND BASELINE METHODS

Metric	SciML + HOCBF	Pure APF
Number of Collisions	0	275
Mean Capture Time (s)	48.3	41.2

Note: Results averaged over 500 Monte Carlo trials with randomized initial conditions

Fig.13 and Table III show that the SciML + HOCBF policy outperforms the baseline APF. Across 500 Monte Carlo trials, it achieved zero collisions, whereas the APF-based controller failed in nearly half the runs. Although the SciML+HOCBF approach exhibits a slightly higher average capture time, this is expected since the statistics are computed only over successful captures. The gain stems from two factors: (i) the SciML training embeds USV dynamics and actuation limits, allowing the controller to slow down before high-rate turns, and (ii)

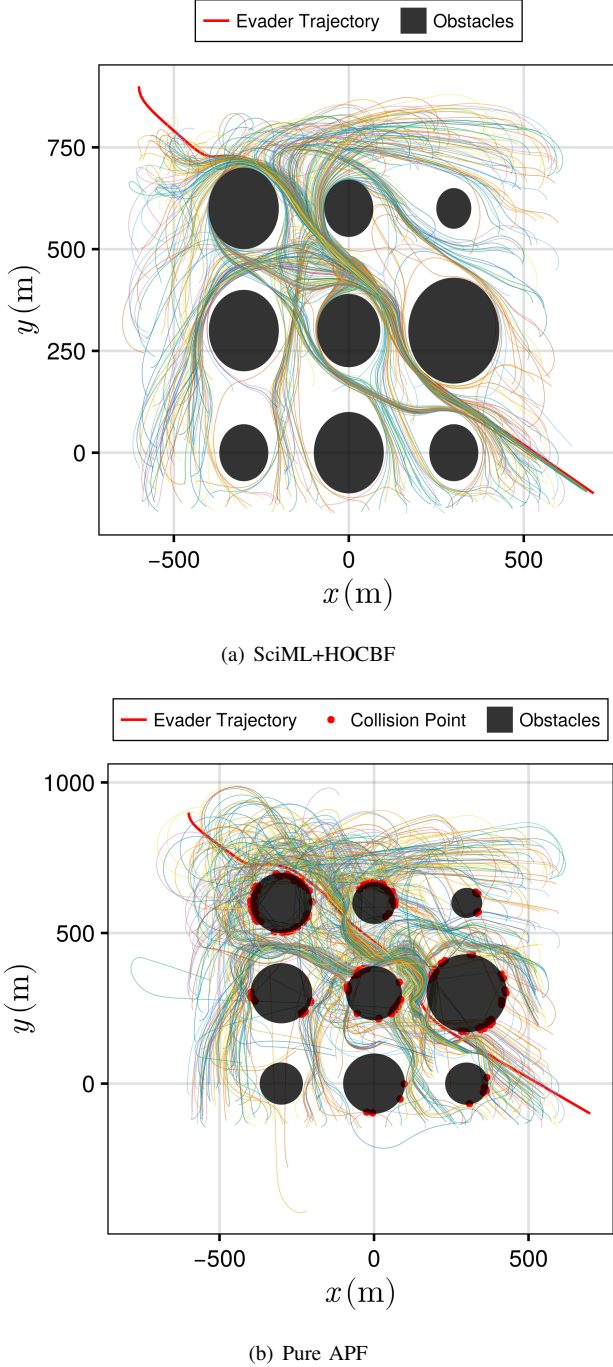


Fig. 13. Monte carlo simulation trajectories for known convex obstacles

the HOCBF layer projects these commands onto a safety-invariant set with minimal deviation. In contrast, the pure APF—lacking both dynamic awareness and formal safety guarantees—accelerates aggressively in cluttered scenes, causing repulsive-force deadlocks and collisions in roughly half the trials.

2) *Scenario 2: Pursuit with Unknown Irregular Obstacles:* In this scenario, two nonconvex-shaped obstacles are placed in the environment with unknown locations. The pursuer relies on an onboard LiDAR sensor for obstacle detection. Similar to Case 1, after identifying the intersection points

between LiDAR rays and obstacles, only the two nearest points are used to reduce the computational load of the HOCBF. Virtual circular obstacles are then defined at these locations for avoidance.

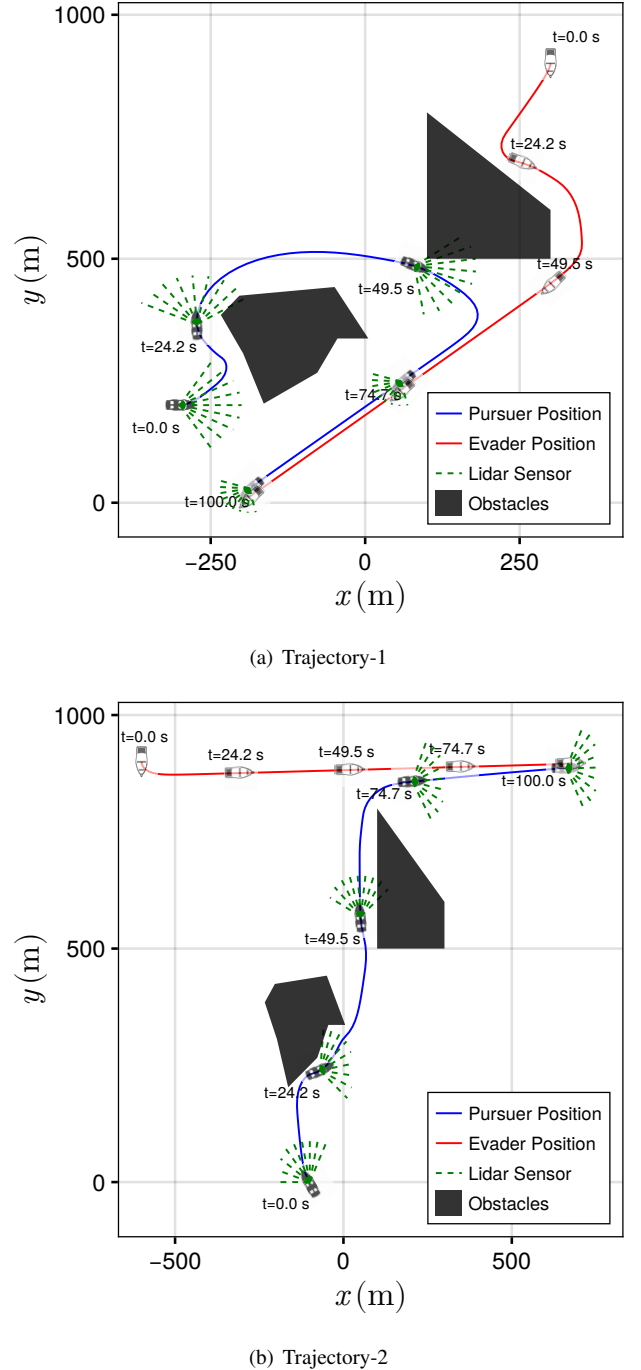


Fig. 14. Representative pursuit-evasion trajectories in environments with unknown irregular obstacles under two different initial conditions

Fig.14 highlights the effectiveness of our proposed approach in a dynamic maritime environment containing irregular unknown obstacles. Despite having no prior knowledge of the obstacles' locations or shapes, it continuously interprets sensor data to create virtual circular obstacles at critical detection points. This enables smooth, collision-free path adjustments

while maintaining an effective pursuit velocity toward the evader. As highlighted in Fig. 14, the proposed controller demonstrates consistent capture performance across different initial conditions, validating its robustness and adaptability under uncertain and cluttered maritime environments.

To further quantify the framework's performance, Fig. 15 and Table IV summarize the results of 500 Monte Carlo trials in environments containing multiple irregular polygon-shaped obstacles. Each trial involved randomized initial positions for the pursuer, yielding statistically significant metrics for capture success and safety. The combined SciML + HOCBF framework achieved reliable capture behavior, with only five collision incidents out of 500 trials and a mean capture time of approximately 51.7 s.

These five failures were attributed not to policy misbehavior but to intrinsic vehicle dynamic limitations. When operating at high speeds, the pursuer cannot instantly execute high-curvature maneuvers without first reducing its velocity; this dynamic constraint occasionally prevented successful avoidance when obstacle detection occurred too late in the engagement. In practice, such cases can be mitigated through the use of higher-range LiDAR or radar sensors, enabling earlier obstacle detection and allowing sufficient time for controlled deceleration before initiating aggressive avoidance or interception maneuvers.

The baseline APF policy was omitted from this analysis, as it failed to maintain collision-free trajectories even in simpler convex-obstacle cases. Consequently, only the hybrid SciML + HOCBF results are reported, underscoring the essential contribution of the safety layer in ensuring reliable and dynamically feasible collision avoidance across stochastic initial conditions.

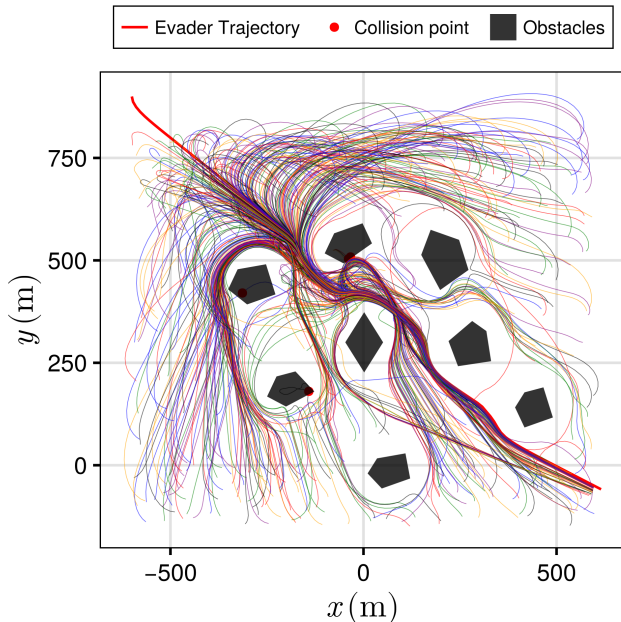


Fig. 15. Monte Carlo Simulation Trajectories for Unknown Polygon-Shaped Obstacles

In summary, the results across both known and unknown obstacle scenarios demonstrate the versatility and robustness

TABLE IV
MONTE CARLO SIMULATION RESULTS FOR POLYGON SHAPED OBSTACLES

Metric	SciML + HOCBF
Number of Collisions	5
Mean Capture Time (s)	51.7

Note: Results averaged over 500 Monte Carlo trials with randomized initial conditions

of the proposed SciML–CBF framework in cluttered maritime environments. For environments with known convex obstacles, the integration of the HOCBF layer successfully compensated for the limitations of the baseline APF policy, which otherwise failed to ensure safe interception. In the more challenging case of unknown, irregularly shaped obstacles, the framework generalized effectively to unseen configurations, leveraging onboard sensing and learned dynamics to maintain safety and pursuit efficiency.

As discussed in the Monte Carlo analysis (Fig. IV), the small number of failed trajectories arose from inherent vehicle dynamic constraints rather than deficiencies in the policy itself. The inability to perform sharp turns at high speeds underscores the importance of aligning sensing range with dynamic feasibility. This limitation can be alleviated by employing longer-range perception systems, which extend the available planning horizon and allow for earlier deceleration and smoother high-curvature maneuvers.

Collectively, these results validate the SciML–CBF framework's ability to achieve collision-free interception under real-world constraints, demonstrating both its physical interpretability and readiness for deployment in autonomous maritime operations where uncertainty and dynamic limitations coexist.

V. CONCLUSION

This work introduced a novel framework that integrates SciML with CBFs to enable safe and effective maritime pursuit–evasion in cluttered environments. By embedding known USV dynamics into a universal differential equation and augmenting the learned policy with a HOCBF layer, the proposed approach achieves both physics-consistent decision making and formal safety guarantees under partial observability and irregular shaped obstacle constraints.

Extensive simulations demonstrate that the SciML–CBF framework consistently achieves collision-free interception against intelligent evaders, even in environments containing irregularly shaped or dynamically detected obstacles. The HOCBF layer effectively compensates for the baseline policy's limitations, ensuring dynamically feasible trajectories that respect safety constraints. Furthermore, Monte Carlo analyses confirm the method's robustness to random initial conditions, with only a few failures attributed to inherent vehicle dynamic constraints. These cases highlight the coupling between sensing range and maneuverability—limitations that can be mitigated in practice through higher-range LiDAR or radar sensing to allow earlier deceleration and tighter curvature maneuvers.

From a computational perspective, both components of the framework are well-suited for real-time embedded deployment. The SciML policy, implemented as a compact feed-forward neural network, achieves inference times on the order of tens of microseconds on a Raspberry Pi 4, corresponding to a control update rate of approximately 100–200 Hz. The accompanying HOCBF safety layer requires solving a small-scale convex quadratic program, which can also be executed within the same control cycle on embedded processors. Together, these properties ensure that the complete inference–optimization loop comfortably meets real-time requirements for onboard autonomy without dedicated GPU hardware.

Overall, the proposed SciML–CBF framework establishes a principled, real-time capable, and safety-certified solution for maritime pursuit–evasion under uncertainty. Future research will focus on hardware validation using autonomous surface and aerial platforms, as well as extensions to multi-agent cooperative and adversarial interactions, thereby bridging the gap between scientific machine learning and field-deployable autonomous systems.

REFERENCES

- [1] X. Qu, L. Zeng, S. Qu, F. Long, and R. Zhang, “An overview of recent advances in pursuit–evasion games with unmanned surface vehicles,” *Journal of Marine Science and Engineering*, vol. 13, no. 3, 2025. [Online]. Available: <https://www.mdpi.com/2077-1312/13/3/458>
- [2] Z. Peng, J. Wang, D. Wang, and Q.-L. Han, “An overview of recent advances in coordinated control of multiple autonomous surface vehicles,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 2, pp. 732–745, 2021.
- [3] V. Sonntag, A. Perrusquía, A. Tsourdos, and W. Guo, “A colregs compliance reinforcement learning approach for usv manoeuvring in track-following and collision avoidance problems,” *Ocean Engineering*, vol. 316, p. 119907, 2025.
- [4] J. Xu, D. Panagopoulos, A. Perrusquía, W. Guo, and A. Tsourdos, “Generalising rescue operations in disaster scenarios using drones: A lifelong reinforcement learning approach,” *Drones*, vol. 9, no. 6, p. 409, 2025.
- [5] J. A. Flores-Campos, C. R. Torres-San-Miguel, J. C. Paredes-Rojas, and A. Perrusquía, “Prescribed time interception of moving objects’ trajectories using robot manipulators,” *Robotics*, vol. 13, no. 10, p. 145, 2024.
- [6] I. E. Weintraub, M. Pachter, and E. Garcia, “An introduction to pursuit–evasion differential games,” *arXiv preprint arXiv:2003.05013*, 2020. [Online]. Available: <https://arxiv.org/abs/2003.05013>
- [7] T. H. Chung, G. A. Hollinger, and V. Isler, “Search and pursuit–evasion in mobile robotics: A survey,” *Autonomous robots*, vol. 31, pp. 299–316, 2011.
- [8] F. Janabi-Sharifi and D. Vinke, “Integration of the artificial potential field approach with simulated annealing for robot path planning,” in *Proceedings of 8th IEEE International Symposium on Intelligent Control*, 1993, pp. 536–541.
- [9] P. Fiorini and Z. Shiller, “Motion planning in dynamic environments using velocity obstacles,” *The International Journal of Robotics Research*, vol. 17, no. 7, pp. 760–772, 1998. [Online]. Available: <https://doi.org/10.1177/027836499801700706>
- [10] S. Karaman and E. Frazzoli, “Sampling-based algorithms for optimal motion planning,” *The International Journal of Robotics Research*, vol. 30, no. 7, pp. 846–894, 2011. [Online]. Available: <https://doi.org/10.1177/0278364911406761>
- [11] K. Tamanakijprasart, A. Perrusquía, S. Mondal, and A. Tsourdos, “Autonomous path selection of unmanned aerial vehicle in dynamic environment using reinforcement learning,” in *AIAA SCITECH 2025 Forum*, 2025, p. 1998.
- [12] A. D. Ames, S. Coogan, M. Egerstedt, G. Notomista, K. Sreenath, and P. Tabuada, “Control barrier functions: Theory and applications,” in *2019 18th European Control Conference (ECC)*, 2019, pp. 3420–3431.
- [13] A. D. Ames, J. W. Grizzle, and P. Tabuada, “Control barrier function based quadratic programs with application to adaptive cruise control,” in *53rd IEEE Conference on Decision and Control*, 2014, pp. 6271–6278.
- [14] N. Rajan, U. Prasad, and N. Rao, “Pursuit–evasion of two aircraft in a horizontal plane,” *Journal of Guidance and Control*, vol. 3, no. 3, pp. 261–267, 1980. [Online]. Available: <https://doi.org/10.2514/3.55982>
- [15] B. Jarmark and C. Hillberg, “Pursuit–evasion between two realistic aircraft,” *Journal of Guidance, Control, and Dynamics*, vol. 7, no. 6, pp. 690–694, 1984. [Online]. Available: <https://doi.org/10.2514/3.19914>
- [16] D. W. Oyler, P. T. Kabamba, and A. R. Girard, “Pursuit–evasion games in the presence of obstacles,” *Automatica*, vol. 65, pp. 1–11, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0005109815004823>
- [17] H. Gong, S. Gong, and J. Li, “Pursuit–evasion game for satellites based on continuous thrust reachable domain,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 56, no. 6, pp. 4626–4637, 2020.
- [18] J. M. Reimann, “Using multiplayer differential game theory to derive efficient pursuit–evasion strategies for unmanned aerial vehicles,” PhD thesis, Georgia Institute of Technology, May 2007, available at <https://repository.gatech.edu/entities/publication/0a078dbb-c05a-42e1-b0ec-57efb140a8f0>.
- [19] M. Cococcioni, L. Fiaschi, and P. F. J. Lermusiaux, “Game theory for unmanned vehicle path planning in the marine domain: State of the art and new possibilities,” *Journal of Marine Science and Engineering*, vol. 9, no. 11, 2021. [Online]. Available: <https://www.mdpi.com/2077-1312/9/11/1175>
- [20] N. Wang, Y. Gao, H. Zhao, and C. K. Ahn, “Reinforcement learning-based optimal tracking control of an unknown unmanned surface vehicle,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 7, pp. 3034–3045, 2021.
- [21] J. Woo, C. Yu, and N. Kim, “Deep reinforcement learning-based controller for path following of an unmanned surface vehicle,” *Ocean Engineering*, vol. 183, pp. 155–166, 2019. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801819302203>
- [22] A. Gonzalez-Garcia, D. Barragan-Alcantar, I. Collado-Gonzalez, and L. Garrido, “Adaptive dynamic programming and deep reinforcement learning for the control of an unmanned surface vehicle: Experimental results,” *Control Engineering Practice*, vol. 111, p. 104807, 2021. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0967066121000848>
- [23] N. Wang, Y. Gao, and X. Zhang, “Data-driven performance-prescribed reinforcement learning control of an unmanned surface vehicle,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 12, pp. 5456–5467, 2021.
- [24] X. Lin, J. McConnell, and B. Englot, “Robust unmanned surface vehicle navigation with distributional reinforcement learning,” in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2023, pp. 6185–6191.
- [25] C. Pan, Z. Peng, L. Liu, and D. Wang, “Data-driven distributed formation control of under-actuated unmanned surface vehicles with collision avoidance via model-based deep reinforcement learning,” *Ocean Engineering*, vol. 267, p. 113166, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801822024490>
- [26] E. Meyer, A. Heiberg, A. Rasheed, and O. San, “Colreg-compliant collision avoidance for unmanned surface vehicle using deep reinforcement learning,” *IEEE Access*, vol. 8, pp. 165 344–165 364, 2020.
- [27] J. Woo and N. Kim, “Collision avoidance for an unmanned surface vehicle using deep reinforcement learning,” *Ocean Engineering*, vol. 199, p. 107001, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801820300792>
- [28] E. Meyer, H. Robinson, A. Rasheed, and O. San, “Taming an autonomous surface vehicle for path following and collision avoidance using deep reinforcement learning,” *IEEE Access*, vol. 8, pp. 41 466–41 481, 2020.
- [29] J. Xia, Y. Luo, Z. Liu, Y. Zhang, H. Shi, and Z. Liu, “Cooperative multi-target hunting by unmanned surface vehicles based on multi-agent reinforcement learning,” *Defence Technology*, vol. 29, pp. 80–94, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S221491472200215X>
- [30] X. Qu, W. Gan, D. Song, and L. Zhou, “Pursuit–evasion game strategy of usv based on deep reinforcement learning in complex multi-obstacle environment,” *Ocean Engineering*, vol. 273, p. 114016, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0029801823004006>
- [31] W. Gan, X. Qu, D. Song, and P. Yao, “Multi-usv cooperative chasing strategy based on obstacles assistance and deep reinforcement learning,”

IEEE Transactions on Automation Science and Engineering, pp. 1–16, 2023.

- [32] X. Liang, Y. Yang, Q. Wang, W. Wang, and W. Han, “Research on anti-pursuit evasion strategy of unmanned surface vehicle based on t-d3qn,” in *2022 5th International Conference on Intelligent Autonomous Systems (ICoIAS)*, 2022, pp. 172–178.
- [33] B. Vlahov, E. Squires, L. Strickland, and C. Pippin, “On developing a uav pursuit-evasion policy using reinforcement learning,” in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2018, pp. 859–864.
- [34] J. Liu, S. Liu, H. Wu, and Y. Zhang, “A pursuit-evasion algorithm based on hierarchical reinforcement learning,” in *2009 International Conference on Measuring Technology and Mechatronics Automation*, vol. 2, 2009, pp. 482–486.
- [35] Y. Kartal, K. Subbarao, A. Dogan, and F. Lewis, “Optimal game theoretic solution of the pursuit-evasion intercept problem using on-policy reinforcement learning,” *International Journal of Robust and Nonlinear Control*, vol. 31, no. 16, pp. 7886–7903, 2021. [Online]. Available: <https://onlinelibrary.wiley.com/doi/abs/10.1002/rnc.5719>
- [36] K. Wan, D. Wu, Y. Zhai, B. Li, X. Gao, and Z. Hu, “An improved approach towards multi-agent pursuit-evasion game decision-making using deep reinforcement learning,” *Entropy*, vol. 23, no. 11, 2021. [Online]. Available: <https://www.mdpi.com/1099-4300/23/11/1433>
- [37] Z. Zhou and H. Xu, “Decentralized optimal large scale multi-player pursuit-evasion strategies: A mean field game approach with reinforcement learning,” *Neurocomputing*, vol. 484, pp. 46–58, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0925231221015769>
- [38] A. Perrusquía and W. Guo, “Uncovering reward goals in distributed drone swarms using physics-informed multiagent inverse reinforcement learning,” *IEEE Transactions on Cybernetics*, 2024.
- [39] E. Bildik, A. Tsourdos, A. Perrusquía, and G. Inalhan, “Decoys deployment for missile interception: A multi-agent reinforcement learning approach,” *Aerospace*, vol. 11, no. 8, p. 684, 2024.
- [40] U. Çelik and M. U. Demirezen, “Optimal reusable rocket landing guidance: A cutting-edge approach integrating scientific machine learning and enhanced neural networks,” *IEEE Access*, vol. 12, pp. 16 805–16 829, 2024.
- [41] M. Uzun, U. Celik, G. Guner, O. Ozdemir, and G. Inalhan, “A certifiable ai-based braking control framework for landing using scientific machine learning,” in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, 2024, pp. 1–10.
- [42] U. Celik, M. Uzun, G. Inalhan, and M. Woods, “Scientific machine learning based pursuit-evasion strategy in unmanned surface vessel defense tactics,” in *2024 AIAA DATC/IEEE 43rd Digital Avionics Systems Conference (DASC)*, 2024, pp. 1–11.
- [43] C. Rackauckas, Y. Ma, J. Martensen, C. Warner, K. Zubov, R. Supekar, D. Skinner, and A. J. Ramadhan, “Universal differential equations for scientific machine learning,” *CoRR*, vol. abs/2001.04385, 2020. [Online]. Available: <https://arxiv.org/abs/2001.04385>
- [44] S. Yaghoubi, G. Fainekos, and S. Sankaranarayanan, “Training neural network controllers using control barrier functions in the presence of disturbances,” in *2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.
- [45] W. Xiao and C. Belta, “Control barrier functions for systems with high relative degree,” in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 474–479.
- [46] N. Lucotte, A. Perrusquía, A. Tsourdos, W. Guo, and H.-S. Shin, “Tactical planning interception enhancement using expert learning - twin delayed deep deterministic policy gradient,” *IEEE Transactions on Intelligent Vehicles*, pp. 1–11, 2025.



at Cranfield University, U.K. His research interests include scientific machine learning, reinforcement learning, optimal control, and trajectory optimization for intelligent air and maritime vehicles.



driven control nonlinear control, machine learning and system identification.

Ugurcan Çelik received the B.Sc. degree in aerospace engineering from Middle East Technical University, Ankara, Türkiye, in 2017, and the M.Sc. degree in aerospace engineering from KTH Royal Institute of Technology, Stockholm, Sweden, in 2019. From 2019 to 2023 he worked in industry as a System Design Engineer and later as an Artificial Intelligence Engineer, focusing on autonomous-systems software and mission-level optimization. He is currently working toward the Ph.D. degree in artificial-intelligence-enabled autonomous systems

Adolfo Perrusquía received the M.Sc. and Ph.D. degrees, both in Automatic Control from the Automatic Control Department at the CINVESTAV-IPN in 2016 and 2020, respectively. He is currently a lecturer at Cranfield University, and a RAEng alumni. He is a member of the IEEE Computational Intelligence Society and Associate Editor of the IEEE TNNLS and IEEE TASE. He is a member of the Early Career Advisory Group of the Control Engineering Practice. His main research of interest focuses on reinforcement learning, inverse reinforcement learning, data-



DSPACE

<https://dspace.org/>

USV pursuit–evasion using a complementary scientific machine learning with control barrier functions approach

Çelik, Ugurcan; Perrusquía, Adolfo

2026-12-31

Institute of Electrical and Electronics Engineers (IEEE)

<https://doi.org/10.1109/joe.2025.3634663>

Çelik U, Perrusquía A. (2026) USV pursuit–evasion using a complementary scientific machine learning with control barrier functions approach. IEEE Journal of Oceanic Engineering, Available online 3 February 2026