

Article

Enhanced Adaptive QPSO-Enabled Game-Theoretic Model Predictive Control for AUV Pursuit–Evasion Under Velocity Constraints

Duan Gao, Mingzhi Chen  and Yunhao Zhang

School of Mechanical Engineering, University of Shanghai for Science and Technology, Yangpu, Shanghai 200093, China; 2235054411@st.usst.edu.cn (D.G.); 2435054827@st.usst.edu.cn (Y.Z.)

* Correspondence: mingzhichen2008@163.com

Abstract

Pursuit–evasion involves coupled, antagonistic decision-making and is prone to local-optimal behaviors when solved online under nonlinear dynamics and constraints. This study investigates a dual-AUV pursuit–evasion problem in ocean-current environments by integrating game theory with model predictive control (MPC). We formulated a game-theoretic MPC scheme that optimizes pursuit and evasion actions over a finite receding horizon, producing Nash-like responses. To solve the resulting nonconvex and multi-modal optimization problems reliably, we developed an Enhanced Adaptive Quantum Particle Swarm Optimization (EA-QPSO) method that incorporates chaos-based initialization and adaptive diversity-aware exploration with stagnation-escape perturbations. EA-QPSO is benchmarked against representative solvers, including fmincon, Differential Evolution (DE), and the Marine Predator Algorithm (MPA). Extensive 2D and 3D simulations demonstrate that EA-QPSO mitigates local-optimum trapping and yields more effective closed-loop behaviors, achieving longer escaping trajectories and more persistent pursuit until capture under the game formulation. In 3D scenarios, EA-QPSO better preserves high-speed motion while coordinating agile angular-rate adjustments, outperforming competing methods that exhibit premature deceleration or degraded maneuvering. These results validate the proposed framework for computing reliable competitive strategies in constrained underwater pursuit–evasion games.

Keywords: pursuit-evasion; autonomous underwater vehicle (AUV); model predictive control (MPC); game theory; optimization algorithm



Academic Editor: Jinfen Zhang

Received: 14 January 2026

Revised: 4 February 2026

Accepted: 5 February 2026

Published: 6 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](#)

[Attribution \(CC BY\)](#) license.

1. Introduction

The increasing deployment of autonomous underwater vehicles (AUVs) in modern naval warfare underscores the strategic importance of underwater autonomy in contested maritime environments. This trend has stimulated growing interest in robust perception, decision-making, and control for hostile marine scenarios, as highlighted in surveys of underwater multi-robot systems and marine robotics [1–3]. Within this context, pursuit–evasion has become a representative and challenging task for underwater confrontation, ranging from learning-based multi-AUV training in the Internet of Underwater Things (IoUT) [4] to neural-network-based control in marine pursuit–evasion settings [5].

Pursuit–evasion inherently features antagonistic objectives: the pursuer aims to capture while the evader attempts to maximize survivability and avoid capture. Such competition naturally aligns with zero-sum modeling, where one agent’s gain corresponds to

the other's loss. Differential game theory provides the classical analytical foundation for pursuit–evasion [6,7], and the strategic interaction can be characterized through minimax optimization under appropriate information and dynamic assumptions. Accordingly, this paper focuses on the dual-agent competitive scenario (one pursuer versus one evader), rather than multi-objective cooperative coordination.

For realistic dual-AUV pursuit–evasion, perception is the first bottleneck: the agents must localize and track each other reliably under underwater sensing constraints. Since underwater visibility is frequently degraded, sonar-based perception becomes a key modality; therefore, continuous target tracking from sonar observations (sonar visual target tracking) is essential to close the perception–decision–control loop and to provide state estimates for downstream game solving and control. In this paper, state information was assumed to be available (via sonar-based tracking).

From a method perspective, recent “smart pursuit–evasion” research can be summarized into four representative families—reinforcement learning (RL), model predictive control (MPC), artificial potential fields (APF), and Apollonius-circle-based geometric methods—each with distinct strengths and limitations, as also emphasized in recent surveys [8].

- (1) RL-based approaches learn strategies through interaction and can handle complex and partially modeled environments. RL has been increasingly applied to pursuit–evasion with obstacles and multi-agent settings [4,9–11]. However, RL typically requires extensive training and careful reward shaping. Moreover, learned policies may exhibit limited generalization when deployment conditions deviate from the training distribution [12,13]. Hybrid learning designs have also been explored, e.g., combining state estimation and actor–critic learning, but simplified evader models may weaken adversarial realism [14].
- (2) MPC-based approaches leverage rolling-horizon optimization and naturally incorporate constraints, making them attractive for interception and pursuit–evasion in dynamic systems [15,16]. Classic receding-horizon game formulations exist [17], and online MPC supervision has been demonstrated in pursuit and evasion applications [18]. Yet, MPC in adversarial settings commonly leads to nonconvex, multi-modal optimization, and performance may depend strongly on solver quality and local-optima avoidance, especially when dynamics are nonlinear and constraints are tight [19]. Therefore, improving the reliability of the online solver is critical for game-theoretic MPC in pursuit–evasion, especially under hard bounds.
- (3) The method based on artificial potential field (APF) constructs virtual attractive and repulsive fields to achieve navigation and obstacle avoidance, which is simple and can be implemented in real-time [20]. Nevertheless, APF methods can suffer from local minima and parameter sensitivity in cluttered environments [21], motivating hybridization with more principled game and control methods [22,23].
- (4) Apollonius-circle and partition-based geometric approaches provide interpretable capture and escape structures through speed ratio and geometric dominance regions, and they have been integrated with learning and partition strategies [24,25]. Although effective in certain regimes, purely geometric strategies may require augmentation when realistic constraints, obstacles, and complex dynamics are present.

In the study of chase-and-escape games under environmental disturbances (such as wind and ocean currents), flow field modeling and geometric constraints are the core elements to ensure the robustness of the strategy, and existing studies have provided important references for this. Sun et al. used an accessibility-based method in dynamic flow fields to solve the Hamilton–Jacobi–Isaacs (HJI) partial differential equation to study the multi-pursuer game strategy affected by spatiotemporal disturbances such as wind or ocean

currents, laying a theoretical foundation for flow field games [26]. In addition, Khachumov et al. proposed an intelligent geometric control theory that combines the Apollonios circular structure with artificial neural networks to solve the problem of chasing and escape path planning of small UAVs under strong wind load interference [27]. Recent work has begun to address underwater pursuit–evasion with currents and obstacles [28]. Although the above studies provide insights into the adversarial behavior in the flow field, there are still certain limitations. On the one hand, these methods often focus on offline calculations or simplified point quality models, which are difficult to meet the high-dimensional challenges caused by the coupling of attitude angles (pitch and yaw) in the 3D underwater environment. On the other hand, when dealing with highly non-convex and multi-peak online optimization problems, traditional solvers are prone to fall into local optimization. This paper draws on the previous ideas on dynamic flow field perturbation modeling and the use of geometric structure to analyze mobility boundaries and further proposes a game model predictive control (GT-MPC) framework based on the augmented adaptive quantum particle swarm algorithm (EA-QPSO). Compared with the existing accessibility or pure geometry methods, the proposed scheme achieves more robust online decision-making under 3D complex constraints by explicitly simulating the Nash equilibrium response of the opponent in the rolling time domain and using the global search capability of EA-QPSO.

Existing studies on pursuit–evasion control can be broadly grouped into (i) geometric or HJI-based offline solutions, (ii) learning-based policies trained offline, and (iii) online MPC and game-MPC schemes solved by local or population-based optimizers. Although game-theoretic MPC provides a principled way to account for strategic coupling, its online min–max subproblems are typically nonconvex and constraint-sensitive, making performance highly dependent on solver robustness. Meanwhile, QPSO/PSO variants with chaotic initialization or adaptive parameters have been widely reported in general optimization, but their direct use inside a coupled min–max receding-horizon game does not automatically guarantee reliable closed-loop behavior, especially when (a) both players optimize antagonistically, (b) feasible regions are shaped by hard bounds and obstacles, and (c) the solver must repeatedly deliver consistent solutions at each sampling instant.

The scientific novelty of this work lies in a closed-loop game-theoretic MPC framework whose solver is explicitly engineered and evaluated for repeated min–max MPC execution under velocity and turn-rate constraints and ocean-current disturbances. In particular, our Enhanced Adaptive QPSO (EA-QPSO) is not presented as a generic “better QPSO”, but as a solver design targeted at the receding-horizon pursuit–evasion game, with mechanisms that directly address (i) early-stage coverage of constrained decision spaces, (ii) diversity collapse under strong nonconvexity, and (iii) stagnation during critical maneuver-switching phases that frequently occur near capture. Accordingly, the main contributions and explicit distinctions from prior work are summarized as follows:

1. We formulated the dual-AUV interaction as a finite-horizon zero-sum min–max MPC problem with bounded surge speed and angular-rate constraints, producing Nash-like (saddle-point-seeking) receding-horizon responses in current-disturbed environments.
2. Beyond commonly used chaos initialization alone, EA-QPSO combines (i) logistic-chaos initialization for improved feasible-space coverage, (ii) a diversity-guided adaptive contraction–expansion rule that responds to the real-time swarm distribution (rather than a fixed linear schedule), and (iii) stagnation-triggered multi-strategy perturbations (e.g., Lévy-flight and elite Gaussian refinement) to escape deep local basins that arise frequently in coupled min–max MPC landscapes. This design is specifically motivated by, and validated in, the repeated online game-MPC setting rather than on standalone benchmark functions.

3. Systematic solver-level benchmarking under equal computational budgets and extension to deployable real-time querying. We benchmark EA-QPSO against representative optimizers (fmincon with SQP, DE, MPA, and GA) under matched evaluation budgets per MPC step and report closed-loop outcomes (capture time, path length, and objective evolution) in both 2D and 3D scenarios. In addition, to enable embedded deployment on lightweight onboard computers, we further constructed an offline policy table that maps relative pose and local current features to control actions, enabling real-time pursuit–evasion behavior with negligible online computational load.

The remainder of this paper is organized as follows. Section 2 presents the problem formulation and modeling. Section 3 describes the control algorithms. Section 4 analyzes the successful capture conditions. Section 5 presents the simulation results. Finally, Section 6 concludes the paper with a summary of findings and future research directions.

2. Modeling of AUV Pursuit–Evasion Scenarios with Sonar-Based State Acquisition

2.1. Modeling Assumptions and Scenario Description

2.1.1. Modeling Assumptions

This study focuses on the decision-making and control algorithms for AUVs in pursuit–evasion scenarios. The vehicle is modeled as a rigid body operating in a submerged environment. To maintain computational tractability for the real-time iterative optimization required by MPC, we adopted the following standard simplifications: the AUV is assumed to be neutrally buoyant and fully submerged; thus, surface effects such as wave interaction and water–air interface dynamics are not explicitly modeled.

2.1.2. Scenario Description

Autonomous underwater vehicles rely on multi-beam sonar to observe and track underwater targets. The process begins with acoustic–visual target tracking, which involves locating and tracking the trajectory of an escaping AUV based on sonar-generated images. This step is critical for accurately determining the target’s position and movement patterns in real-time. This paper focuses on decision-making and control; opponent states are assumed available via onboard sensing and tracking, whereas perception errors are not explicitly modeled.

Once the target’s location is established, the scenario evolves into a dual-AUV pursuit–evasion problem, where both the pursuing and escaping AUVs engage in a strategic interaction based on their known positions. This phase focuses on optimizing the pursuit strategy while considering the evader’s potential countermeasures, making it a complex and dynamic challenge in underwater vehicles.

2.2. AUV Kinematic Model

Considering a pursuit–evasion game between two AUVs in a bounded two-dimensional (2D) environment, a single pursuer AUV attempts to capture a single evader AUV. The pursuer seeks to minimize the capture time, whereas the evader aims to maximize the time to capture. We assumed that the pursuer had a higher cruising speed than the evader, whereas the evader is more agile (i.e., capable of faster maneuvering) than the pursuer. Both agents are assumed to have access to the relative state information required for decision-making (e.g., via onboard sensing and tracking). As shown in Figure 1, the kinematic models of the two AUVs in the 2D environment are given in (1).

$$\dot{\eta}_i = \begin{bmatrix} \cos \psi_i & 0 \\ \sin \psi_i & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} u_i \\ r_i \end{bmatrix} + \begin{bmatrix} V_x \\ V_y \\ 0 \end{bmatrix}, i = p, e \quad (1)$$

where $\eta_i = [x_i, y_i, \psi_i]^T$ denotes the position (x_i, y_i) and heading angle (ψ_i) of the AUV; u_i and r_i are the forward velocity and turning rate, respectively; and V_x and V_y represent the longitudinal and lateral ocean current velocities in the inertial frame. The ocean current can be obtained through DVL.

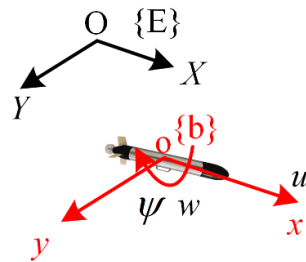


Figure 1. Coordinate frames of the AUV.

2.3. Ocean Current Model

Ocean circulation is affected by the coupling of multiple factors such as seasonal fluctuations, climate change, seabed topography and water depth differences, and presents complex spatiotemporal heterogeneity. The accurate mathematical representation of its motion laws faces significant challenges. Although the ocean current system shows significant spatiotemporal heterogeneity in the vast ocean, its flow velocity and flow direction are relatively stable in specific local ocean areas. Based on this assumption, the fluid movement within a particular grid cell is dominated by the average flow intensity. To characterize the wave patterns and mixing mechanisms of meandering jets in ocean fluids, we adopted the kinematic stream function model originally proposed by Bower [29,30]. This model has been widely utilized in the literature to describe the motion characteristics of evolving ocean fluids over time. The stream function $\vartheta(x, y, t)$ is defined as:

$$\vartheta(x, y, t) = 1 - \tanh \left(\frac{y - B(t) \cos(k(x - ct))}{(1 + \lambda^2 B(t)^2 \sin^2(k(x - ct)))^{0.5}} \right)$$

$$B(t) = B_0 + \alpha \cos(\omega_0 t + \varphi)$$

$$V_x(x, y) = -U_m \partial \vartheta(x, y, t) / \partial y$$

$$V_y(x, y) = V_m \partial \vartheta(x, y, t) / \partial x \quad (2)$$

In this paper, the key parameters of the model were set as follows: $k = 1$, $c = 0.12$, $\lambda = 0.84$, $B_0 = 0.12$, $\alpha = 0.3$, and $\omega_0 = 0.4$. The flow velocity field at the grid cell (x, y) can be expressed as $(V_x(x, y), V_y(x, y))$ and its specific mathematical expressions are shown in Equation (2).

2.4. Successful Capture Condition

The position of the pursuing AUV is defined as $p(t) = [x_p(t) \ y_p(t)]^T$, whereas the position of the escaping AUV is defined as $e(t) = [x_e(t) \ y_e(t)]^T$. The distance between them is and the condition for successful capture is described in Equation (3).

$$d_{p,e}(t_c) \leq d_c, t_c \geq 0, d_{p,e}(t) = \|p(t) - e(t)\| \quad (3)$$

Equation (3) establishes the criterion for a successful pursuit: at a certain moment $t_c \geq 0$, the distance between the two AUVs is less than a predetermined successful capture distance (d_c) [19]. Therefore, the underwater pursuit–evasion game between the two AUVs can be modeled as shown in Equation (4).

$$\begin{aligned} & \max_{U_e} \min_{U_p} t_c \\ & U_e = [u_e \quad r_e]^T \in U_e \\ & U_p = [u_p \quad r_p]^T \in U_p \end{aligned} \quad (4)$$

Equation (4) indicates that the pursuing AUV adjusts U_p to minimize the capture time t_c , whereas the escaping AUV adjusts U_e to maximize the capture time t_c . U_p and U_e represent the feasible speed ranges for the pursuing AUV and the escaping AUV, respectively.

2.5. Position Information Obtaining

The AUV is usually equipped with sonar to detect targets, as shown in Figure 2. In the sonar detection, the relative position, direction and azimuth of the two can be obtained. A learning-based detector and tracker (e.g., YOLO-type models) can be used to obtain target observations from sonar imagery; however, this paper focuses on decision-making and control. After sonar–visual target tracking, the tracked coordinates need to be converted to the inertial coordinate system. First, the coordinates are transformed into polar coordinates, represented by distance r and angle θ , as shown in Figure 3. The conversion formula is given in Equation (5), where x_m is the x -axis dimension of the sonar image.

$$\begin{cases} \theta = \arctan[(x_s - 0.5x_m)/y_s] \\ r = \sqrt{x_s^2 + y_s^2} \end{cases} \quad (5)$$



Figure 2. (a) The EDGEE-R2-EDU ROV platform. (b) Sonar imaging scene of the ROV.

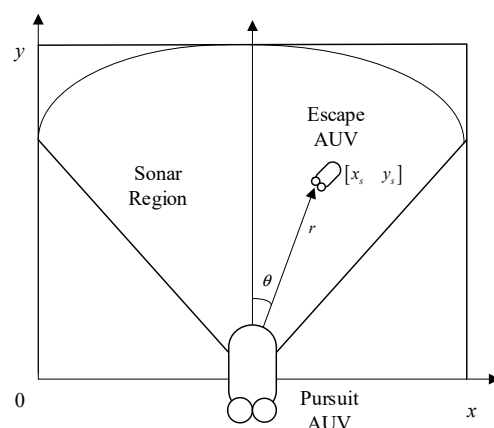


Figure 3. Conversion of the sonar observation coordinates to polar coordinates.

Next, the polar coordinates of the AUV carrier coordinate system were mapped to the inertial coordinate system. The transformation between the two coordinate systems is described by Equation (6).

$$\begin{cases} x_e = x_p + r \cdot \cos(\theta + \psi_p) \\ y_e = y_p + r \cdot \sin(\theta + \psi_p) \end{cases} \quad (6)$$

where x_e and y_e are the coordinates of the escaping AUV in the inertial coordinate system, (x_p, y_p, ψ_p) are the coordinates and heading angle of the chasing AUV in the inertial coordinate system; r is the distance between the escaping and chasing AUVs; and θ is the vector angle of the escaping AUV in the chasing AUV carrier coordinate system. The conversion schematic is illustrated in Figure 4.

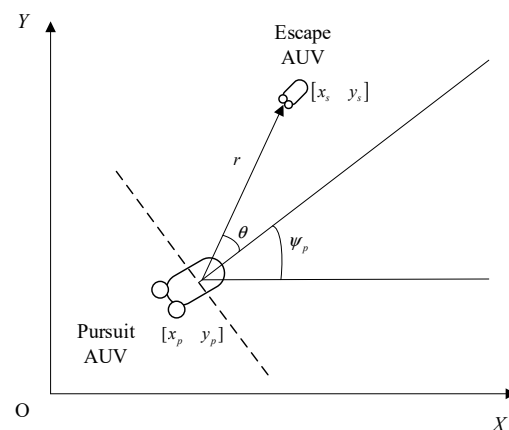


Figure 4. Schematic of the AUV carrier coordinate to inertial coordinate conversion.

2.6. Assumptions and Limitations

The modeling and strategy derivation of this paper is based on the core assumption that the relative state of the pursuer and the evader, as well as their respective physical states, are observable in real time and are directly provided by the onboard sonar visual tracking system and navigation and positioning system. Although this assumption enables research to focus on the design of game decision-making algorithms in complex dynamic environments, there are still limitations in actual engineering deployment.

Specifically, the following perceptual uncertainties were not modeled in this paper: the measurement noise of sonar sensors, the occlusion of targets in complex terrain, false detections, delays and frame drops caused by signal processing, and residual errors in ocean current estimation. In actual ENGAGEMENT, these errors at the state acquisition level can lead to biases in the prediction trajectory of MPC, which, in turn, affects the real-time judgment of capture feasibility conditions, and may cause the final generated adversarial strategy to be suboptimal rather than the ideal game balance point.

3. Control Algorithms for Dual AUV Pursuit–Evasion

Model Predictive Control is a widely utilized algorithm in industrial control, characterized by a process of predictive modeling, rolling optimization, and feedback correction. The predictive model captures the state of the controlled system and forecasts future states based on the current state, typically employing transfer functions and state-space equations. Due to inherent limitations in model accuracy during rolling optimization, future predictions can be unreliable, leading to accumulated errors over time. To address this, feedback is incorporated to adjust the predicted outputs for the next time step based on real-time information, known as feedback correction. Consequently, MPC seeks optimal solutions

within a finite horizon, continuously recalculating the optimal state of the system to enhance control performance. An important part of MPC is establishing an objective function.

3.1. Non-Game Model Predictive Control for Dual AUV Pursuit–Evasion

In the dual AUV pursuit–evasion game, the two AUVs use the kinematic model, as described in Equation (1), to predict their own states. The objective function of the model predictive controller for the pursuing AUV is proposed as follows (where k : current time step; p : prediction horizon; t : time index; and c : control horizon):

$$\begin{aligned} J_p &= \min_{U_{p,k}, U_{p,k+1}, \dots, U_{p,k+c}} d_{p,e}(k+p)_{Q_p}^2 + \sum_{t=k+1}^{k+p-1} d_{p,e}(t)_Q^2 \\ \text{s.t. } \|p(t) - cp_{obs}\| &\geq (R_{obs} + R_{AUV}), t = k+1, k+2, \dots, k+p \\ U_{p,t} &= [u_{p,t} \quad r_{p,t}]^T \in U_p, t = k, k+1, \dots, k+c \end{aligned} \quad (7)$$

Equation (7) indicates that the goal of the pursuing AUV is to optimize the control input to minimize the distance between its predicted position and that of the escaping AUV [19]. Due to the presence of obstacles in the environment, the first constraint introduced in Equation (7) is set to prevent collisions between the pursuing AUV and the obstacles, whereas the second constraint ensures that the speed of the pursuing AUV remains within the feasible range. The objective function for the escaping AUV is then proposed as follows:

$$\begin{aligned} J_e &= \min_{U_{e,k}, U_{e,k+1}, \dots, U_{e,k+c}} - [d_{p,e}(k+p)_{Q_p}^2 + \sum_{t=k+1}^{k+p-1} d_{p,e}(t)_Q^2] \\ \text{s.t. } \|e(t) - cp_{obs}\| &\geq (R_{obs} + R_{AUV}), t = k+1, k+2, \dots, k+p \\ U_{e,t} &= [u_{e,t} \quad r_{e,t}]^T \in U_e, t = k, k+1, \dots, k+c \end{aligned} \quad (8)$$

Equation (8) indicates that the objective of the escaping AUV is the opposite of that of the pursuing AUV; its goal is to optimize the control input to maximize the distance between its predicted position and the pursuing AUV [19]. Similarly, the first constraint prevents the escaping AUV from colliding with obstacles, whereas the second constraint ensures that the speed of the escaping AUV remains within the feasible range.

3.2. Game-Theoretic Model Predictive Control for the Dual AUV Pursuit–Evasion Game

The above method outlines a strategy for addressing the pursuit–evasion problem using model predictive control. In a baseline (non-game) MPC formulation, the opponent is treated as exogenous and is typically predicted using a simple motion model (or even assumed stationary), which may ignore strategic reactions. Although both AUVs receive timely feedback during the rolling optimization process of model predictive control, this does not adequately incorporate the strategies of the opposing party in the objective function optimization. When making decisions in non-game MPC, the system may even assume a stationary target by calculating the objective function using Equation (7) or (8) and minimizing it. However, this approach neglects the opponent's strategic responses. By incorporating game theory into the MPC prediction, each AUV optimizes its control sequence while explicitly accounting for the opponent's best response. Over the finite horizon, the coupled problem is solved numerically as a min–max optimization, producing Nash-like (finite-horizon approximate equilibrium) control sequences.

In the dual-AUV pursuit–evasion setting, the two players make antagonistic decisions, and each player's control action depends on the anticipated reaction of the opponent. The interaction is modeled as a zero-sum differential game over the MPC prediction horizon. Let U_p and U_e denote the pursuer's and evader's control sequences over the horizon,

respectively, and let $J(U_p, U_e)$ denote the pursuer's payoff (e.g., a distance-based cost defined in Section 3.1). Under the zero-sum assumption, the evader's payoff is the negative of the pursuer's payoff [7,19], $J_p(U_p, U_e) = J(U_p, U_e)$, $J_e(U_p, U_e) = -J(U_p, U_e)$, which implies the following:

$$J_p + J_e = 0 \quad (9)$$

Accordingly, at each MPC update, the pursuer selects U_p to minimize its payoff under the worst-case evader response, leading to the finite-horizon min–max problem.

$$(U_p^*, U_e^*) \in \arg \min_{U_p} \max_{U_e} J_p(U_p, U_e) \quad (10)$$

Symmetrically, the evader aims to maximize the pursuer's payoff (equivalently, minimize its own payoff), which can be written as follows:

$$(U_p^*, U_e^*) \in \arg \min_{U_p} \max_{U_e} J_p(U_p, U_e) \Leftrightarrow (U_p^*, U_e^*) \in \arg \min_{U_e} \max_{U_p} J_e(U_p, U_e) \quad (11)$$

The resulting solution pair (U_p^*, U_e^*) is treated as a finite-horizon Nash-like (saddle-point-seeking) approximation, and only the first control input is applied before the optimization is repeated at the next sampling instant.

3.2.1. Finite-Horizon Nash-like Equilibrium Analysis

In this work, the term “finite-horizon Nash-like equilibrium” refers to a local saddle-point approximation of the zero-sum min–max problem solved at each MPC update. Formally, for a fixed sampling instant k , prediction horizon N , and the current state $(\eta_p(k), \eta_e(k))$, we consider the finite-horizon game $\min_{U_p} \max_{U_e} J(U_p, U_e)$.

A pair (U_p^*, U_e^*) is a (global) saddle point if it satisfies $J(U_p^*, U_e) \leq J(U_p^*, U_e^*) \leq J(U_p, U_e^*)$, $\forall U_p \in U_p, \forall U_e \in U_e$. Under mild regularity conditions, a solution to the finite-horizon game exists in the sense of optimal values. In our setting, the input bounds are compact, and the cost is continuous with respect to the control sequences. Therefore, the minimization and maximization over compact sets admit at least one optimizer for each nested problem. However, the existence of a global saddle point is not guaranteed, because the resulting finite-horizon game is generally nonconvex. The nonlinear dynamics, together with obstacle-avoidance penalties and input saturation constraints, break convexity in the payoff, so classical minimax equalities and saddle-point existence guarantees do not directly apply.

Even when a saddle point exists, uniqueness is typically not expected. The pursuit–evasion payoff landscape is often multi-modal due to geometric symmetries (e.g., left and right turns) and switching maneuvers, leading to multiple local saddle candidates with similar costs. Accordingly, the “Nash-like” solution computed online should be interpreted as one of possibly many local equilibrium-like responses.

In practice, we compute (U_p^*, U_e^*) numerically using finite iterations of a metaheuristic optimizer. Hence, the implemented pair is best viewed as a numerical approximate saddle point (finite-horizon Nash-like response) rather than a provable equilibrium of the underlying continuous-time differential game. Only the first control input is applied and the optimization is repeated at the next sampling instant, which further implies that equilibrium concepts apply per-step and per-horizon, not globally over the entire engagement.

3.2.2. Sensitivity to the MPC Horizon

The computed Nash-like response generally depends on the prediction horizon N . Increasing N changes the relative importance of near-term capture geometry versus longer-term maneuvering and may introduce additional local optima. Therefore, monotonic im-

provement with larger N is not guaranteed in nonconvex min–max MPC. From a practical standpoint, we used a fixed horizon throughout the simulations and relied on receding-horizon feedback and EA-QPSO to obtain consistent closed-loop behavior.

3.3. Enhanced Adaptive QPSO Optimization Algorithm

In the game-theoretic NMPC framework for the dual-AUV pursuit–evasion problem, both agents must repeatedly solve constrained nonconvex optimization problems online over a receding horizon. Because the objective is typically multi-modal (with multiple local minima or maxima), conventional local solvers (e.g., MATLAB 2024b `fmincon`) may be sensitive to initialization and can become trapped in poor local optima, leading to premature convergence and suboptimal pursuit and evasion behaviors. To improve solution robustness, this paper adopts Quantum Particle Swarm Optimization (QPSO) as the core optimizer and further proposes an enhanced variant to strengthen global exploration in complex multi-peak search spaces.

3.3.1. Standard QPSO

QPSO is an improved particle swarm method based on a quantum-behaved model. Candidate solutions are represented as “particles” in an n -dimensional search space, and particle positions are updated through probabilistic sampling, which enhances global search capability and mitigates local-optimum trapping.

Assume a swarm of M particles in an n -dimensional space. At iteration t , the position of particle i is $\mathbf{x}_i(t) = [x_{i,1}(t), \dots, x_{i,n}(t)]^\top$. The personal best position is $\mathbf{p}_i(t)$, and the global best is $\mathbf{g}(t)$ [31]. The mean best position (mbest) is defined as follows:

$$\mathbf{m}(t) = \frac{1}{M} \sum_{i=1}^M \mathbf{p}_i(t) \quad (12)$$

A typical QPSO position update can be written as follows:

$$\mathbf{x}_i(t+1) = \mathbf{p}_i(t) \pm \alpha(t) \mid \mathbf{m}(t) - \mathbf{x}_i(t) \mid \cdot \ln\left(\frac{1}{u}\right) \quad (13)$$

where $u \sim U(0, 1)$, $\pm$ is chosen with equal probability and $\alpha(t)$ is the contraction–expansion coefficient controlling the balance between exploration and exploitation. Bound constraints were enforced via projection (component-wise saturation) after each position update. State constraints (e.g., obstacle collisions) were handled through a penalty formulation added to the payoff: infeasible solutions were assigned $+\infty$ for the pursuer and $-\infty$ for the evader. However, using a simple linear schedule for $\alpha(t)$ may not reflect the real-time swarm distribution and can reduce exploration in multi-peak environments.

3.3.2. Enhanced Adaptive QPSO (EA-QPSO): Chaos Initialization and Multi-Strategy Perturbations

To address the premature convergence and insufficient exploration of standard QPSO in complex multi-modal landscapes, we propose an Enhanced Adaptive QPSO (EA-QPSO) that integrates Logistic chaos initialization and multi-policy perturbation mechanisms. The enhanced algorithm introduces five mechanisms as follows:

(1) Logistic-chaos-based population initialization

Standard QPSO commonly initializes particles using uniform random sampling, which may produce uneven coverage and “search blind spots”. EA-QPSO uses a Logistic chaotic map to generate the initial swarm, leveraging its ergodicity and pseudo-randomness to better cover the feasible space.

Let $x_{k,d}^0$ denote the initial position of particle k in dimension d . The chaotic sequence is generated by the following:

$$z_{n+1} = \mu z_n (1 - z_n), \mu = 4, z_0 \in (0, 1) \quad (14)$$

and mapped to the bounded search interval $[x_{\min,d}, x_{\max,d}]$ as follows:

$$x_{k,d}^0 = x_{\min,d} + z_n (x_{\max,d} - x_{\min,d}) \quad (15)$$

This initialization can significantly improve early-stage global exploration and provides a higher-quality initial solution set.

(2) Diversity-guided adaptive contraction–expansion coefficient $\alpha(t)$

The coefficient $\alpha(t)$ is the key parameter controlling convergence speed and exploration radius. A linear decay schedule ignores the actual swarm distribution. EA-QPSO introduces a diversity measure $D(t)$ to adaptively tune $\alpha(t)$.

Let the population mean in each dimension be $\bar{x}_d(t) = \frac{1}{M} \sum_{i=1}^M x_{i,d}(t)$. The diversity index is defined as follows:

$$D(t) = \frac{1}{M \cdot n} \sum_{i=1}^M \sqrt{\sum_{d=1}^n (x_{i,d}(t) - \bar{x}_d(t))^2} \quad (16)$$

After normalizing $D(t)$ to $\hat{D}(t)$:

$$\alpha(t) = \alpha_{\min} + (\alpha_{\max} - \alpha_{\min}) \cdot (1 - \hat{D}(t)) \quad (17)$$

$\alpha_{\max} = 0.85$ and $\alpha_{\min} = 0.35$ are set in this paper.

(3) Lévy-flight perturbation for stagnation handling

When the global best value does not improve for S consecutive generations (stagnation), a Lévy-flight perturbation is applied to randomly selected particles as follows:

$$\mathbf{x}_{\text{new}} = \mathbf{x}_{\text{old}} + \beta \odot \text{Levy}(\lambda) \quad (18)$$

where $\odot$ denotes element-wise multiplication, $\beta = 1.5$ is a scaling factor, and $\text{Levy}(\lambda)$ produces a heavy-tailed random step (often characterized by a tail proportional to $t^{-\lambda}$).

(4) Elite Gaussian Variant (Elite Gaussian Variant)

To further improve the convergence accuracy, a small-scale Gaussian mutation attempt was made on the current global optimal particle, with a certain probability (15%) in each iteration. If the mutated fitness was better than the original optimal solution, the new position was adopted. This strategy is similar to the “local refinement” operation, which can help the global optimal solution to be fine-tuned in a very small range in the later stage of the algorithm, and significantly improves the accuracy of the final solution.

3.3.3. Comparison of Metaheuristic Solvers for Nonconvex Game-Theoretic MPC

To clarify the theoretical advantages of the proposed EA-QPSO for resolving the non-convex game-theoretic MPC problem, the key characteristics of the candidate solvers used in this study (Standard QPSO, DE, MPA, and the proposed EA-QPSO) are summarized in Table 1.

Table 1. Summary of Algorithmic Characteristics and Mechanisms.

Algorithm	Search Strategy and Core Mechanism	Diversity Maintenance and Stagnation Handling	Suitability for Game-MPC
Standard QPSO	Probabilistic sampling based on a quantum-behaved model and the mean best position (<i>mbest</i>).	Relies on a linear schedule for $\alpha(t)$, which may ignore real-time swarm distribution.	High global search capability but prone to premature convergence in multi-modal landscapes.
DE	Evolutionary optimization based on mutation, crossover, and selection operators.	Diversity is maintained through differential mutation, but can be slow to escape deep local optima.	Robust for general optimization, but may exhibit non-beneficial oscillations in 3D control signals.
MPA	Inspired by marine predator foraging strategies, using Brownian and Lévy movement.	Effective exploration but lacks explicit diversity-aware adaptive tuning.	Strong performance, yet slightly less stable than EA-QPSO during critical maneuvering phases.
EA-QPSO (Ours)	Logistic-chaos initialization and Elite Gaussian Variants for local refinement.	Diversity-guided adaptive $\alpha(t)$ and Lévy-flight perturbations for stagnation escape.	Optimized for non-convex landscapes; mitigates local-optima trapping to yield effective closed-loop evasion.

4. Theoretical Analysis of Capture Conditions

4.1. Necessary Conditions for Capture

Theoretical conditions were derived for an obstacle-free environment under uniform (or locally approximately uniform) currents, so that the current cancels out in the relative dynamics. For successful capture in an obstacle-free, infinite 2D environment, the following necessary conditions must be satisfied.

4.1.1. Velocity Advantage Condition

The pursuer's maximum velocity must be greater than the evader's:

$$u_{p,\max} > u_{e,\max} \quad (19)$$

Proof. Assume $u_{p,\max} \leq u_{e,\max}$. The evader can adopt a straight-line escape strategy ($r_e = 0$), so the distance change rate is as follows:

$$\dot{d} = \frac{(x_e - x_p)(\dot{x}_e - \dot{x}_p) + (y_e - y_p)(\dot{y}_e - \dot{y}_p)}{d} \quad (20)$$

Substituting $\dot{x}_i = u_i \cos \psi_i + V_x$ and $\dot{y}_i = u_i \sin \psi_i + V_y$, if the evader aligns its heading with the direction away from the pursuer ($\psi_e = \psi_{\text{LOS}} + \pi$), then

$$\dot{d} \geq u_{e,\max} - u_{p,\max} \geq 0 \quad (21)$$

The distance does not decrease, so capture is impossible. Thus, $u_{p,\max} > u_{e,\max}$ is a necessary condition for capture. $\square$

4.1.2. Maneuverability Constraint (Maximum Curvature)

The maximum curvature of the pursuer must be no less than a critical value related to the evader's maximum curvature. The maximum curvature is defined as follows:

$$K_{i,\max} = \frac{r_{i,\max}}{u_{i,\max}} (u_{i,\max} > 0) \quad (22)$$

where $K_{i,\max}$ reflects the tightness of the AUV's turns (smaller minimum turning radius $R_{i,\min} = 1/K_{i,\max}$ indicates stronger maneuverability). For the pursuer to track the evader's maneuver, the following condition must hold:

$$K_{p,\max} \geq \gamma K_{e,\max} \quad (23)$$

where $\gamma = \frac{u_{e,\max}}{u_{p,\max}} \in (0, 1)$ is the velocity ratio. If $K_{p,\max} < \gamma K_{e,\max}$, the evader can perform high-curvature maneuvers (e.g., circular or serpentine motion) to make the pursuer's heading adjustment unable to keep up with the LOS angle change, resulting in failure to reduce the distance.

4.2. Sufficient Condition for Capture

Combining velocity advantage and maneuverability, the sufficient condition for capture in a bounded environment is as follows:

$$\begin{cases} u_{p,\max} > u_{e,\max} \\ K_{p,\max} \geq \gamma K_{e,\max} \\ \dot{\psi}_{\text{LOS}}(t) \leq r_{p,\max} \end{cases} \quad (24)$$

where the LOS angular velocity $\dot{\psi}_{\text{LOS}}$ is approximated as follows:

$$\dot{\psi}_{\text{LOS}} \approx \frac{u_{e,\max} \sin \Delta\psi}{d(t)} \quad (25)$$

and $\Delta\psi = \psi_e - \psi_{\text{LOS}}$ is the evader's heading deviation from the LOS direction. When $\dot{\psi}_{\text{LOS}} \leq r_{p,\max}$, the pursuer can adjust its heading to track the evader, ensuring continuous distance reduction.

4.3. Impact of Optimization Algorithm on Capture Feasibility

The min-max MPC optimization problem is non-convex due to rolling optimization, discrete dynamics, and control constraints. The quality of the approximate Nash equilibrium solution depends on the optimization algorithm's ability to avoid local optima.

5. Simulation Study

A dual-AUV pursuit–evasion game simulation was conducted in a 30 * 30 m planar environment. The MPC parameters were set as follows: prediction and control horizons $N = 10$, sampling time $dt = 0.1$ s, and capture radius $d_c = 0.1$ m. All simulations in this study were implemented in MATLAB R2024b and executed on a laptop equipped with an Intel(R) Core(TM) i7-10750H CPU@2.60 GHz.

5.1. Two-Dimensional Ocean-Current, Obstacle-Free Environment

In the obstacle-free case, two AUVs engage in a pursuit–evasion game under a spatially varying current field. We compared four representative optimizers for solving the finite-horizon min-max MPC subproblem:

- (1) fmincon: Gradient-based local optimization using SQP, which is sensitive to initialization and may converge to locally optimal stationary points in nonconvex problems.
- (2) DE (Differential Evolution): Evolutionary optimization based on mutation, crossover, and selection.
- (3) MPA (Marine Predator Algorithm): A swarm-intelligence algorithm inspired by marine predator foraging strategies.
- (4) EA-QPSO: The enhanced adaptive QPSO proposed in this paper.

The AUV control constraints were set as follows: Pursuer: $u_{p,\max} = 1$, $r_p \in [-\pi/4, \pi/4]$. Evader: $r_e \in [-\pi/2, \pi/2]$, $u_{e,\max}$ was varied to represent different evasion capabilities. To ensure a fair comparison among population-based methods, DE, MPA, and EA-QPSO were run with the same population size and iteration limit (both set to 30), yielding an identical nominal sampling budget per MPC step. As a deterministic local baseline, we solve each MPC subproblem using MATLAB `fmincon` with the sequential quadratic programming (SQP) algorithm. Unless otherwise specified, the solver was run with `Display = 'none'`, `MaxIterations = 50`, `OptimalityTolerance = 1 \times 10^{-6}`, and `StepTolerance = 1 \times 10^{-6}`. We report the resulting closed-loop performance together with the per-step computational cost (runtime and/or function-evaluation counts) to enable a fair comparison against population-based metaheuristics.

To isolate solver-induced effects under a representative initial condition, we evaluate four pursuer–evader solver pairings with pursuer initial state $[15, 15, \pi/4]$, evader initial state $[17, 17, \pi/2]$, and $u_{e,\max} = 0.6$: Combo 1: (P: `fmincon`, E: `fmincon`), Combo 2: (P: `fmincon`, E: EA-QPSO), Combo 3: (P: EA-QPSO, E: `fmincon`), and Combo 4: (P: EA-QPSO, E: EA-QPSO).

5.1.1. Trajectory and Objective Evolution Under the Representative Initial Condition

Figure 5 shows the closed-loop pursuit–evasion trajectories for Combo 2 (P: `fmincon`, E: EA-QPSO) in the current field. The evader executes repeated high-curvature turning maneuvers, whereas the pursuer trajectory is comparatively smoother and responds more slowly to rapid heading changes. This behavior is consistent with the solver characteristics: in the nonconvex receding-horizon min–max problem, the gradient-based local search employed by `fmincon` may settle into locally optimal control sequences, whereas the population-based EA-QPSO is more likely to explore multiple basins and switch among distinct maneuver modes.

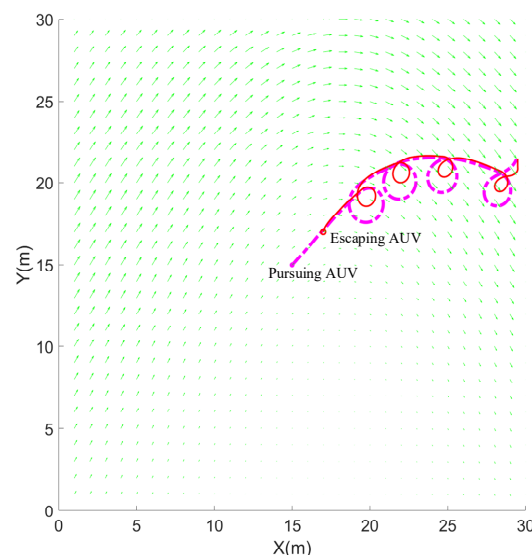
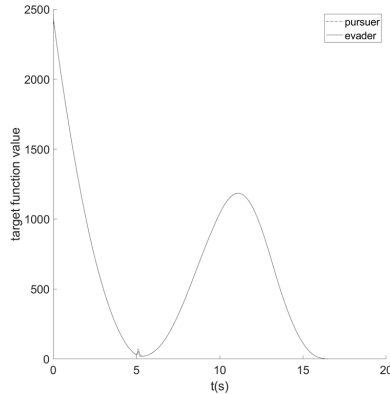
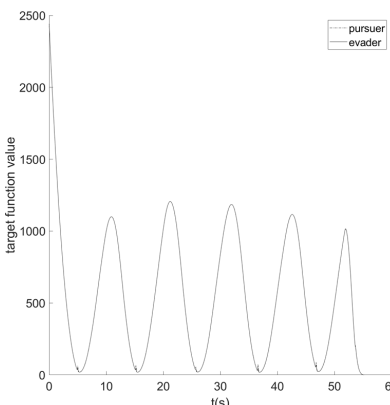
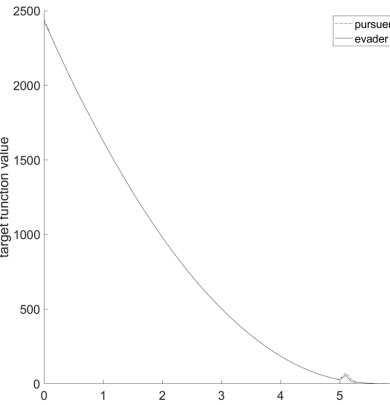
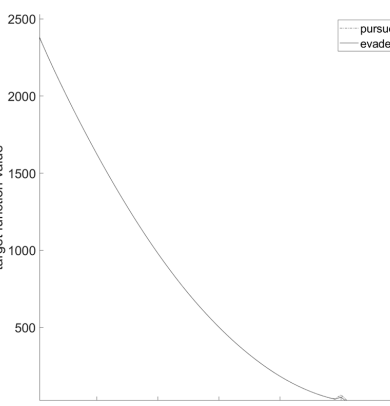


Figure 5. The pursuit and evasion trajectories of Combo 2. The pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

Table 2 summarizes the objective-function value evolution over time. As noted, the x -axis denotes simulation time, and the y -axis reports the absolute value of each agent's objective. Compared with the case where both agents use the same local solver, pairings involving EA-QPSO exhibited more pronounced nonconvex exploration, reflected by richer oscillatory patterns and more frequent switching among locally optimal maneuver modes.

Table 2. Summary of closed-loop outcomes and objective evolution.

Pursuer	Evader	Objective Function Image	Trajectory Diagram
fmincon	fmincon		Steps: 166, time: 16.50 s Distance pursuer: 15.654 m Distance evader: 9.802 m
fmincon	EA-QPSO		Steps: 553, time: 55.20 s Distance pursuer: 48.444 m Distance evader: 32.598 m
EA-QPSO	fmincon		Steps: 59, time: 5.80 s Distance pursuer: 6.128 m Distance evader: 3.503 m
EA-QPSO	EA-QPSO		Steps: 59, time: 5.80 s Distance pursuer: 6.107 m Distance evader: 3.440 m

Across the evaluated pairings, Combo 2 (P-fmincon vs. E-EA-QPSO) constitutes the most adversarial case for the pursuer and yields the most sustained evasion performance. Quantitatively, Combo 2 prolongs the engagement to 55.2 s (553 steps), compared with 16.5 s (166 steps) under Combo 1 (both pursuers using fmincon), corresponding to a +38.7 s increase (+234.5%, $3.35\times$ longer). This longer engagement was accompanied by substantially larger path lengths: the pursuer traveled 48.444 m and the evader traveled 32.598 m, which are +209.5% ($3.09\times$) and +232.6% ($3.33\times$) higher than in Combo 1, respectively. Relative to the cases where the pursuer used EA-QPSO (Combo 3/4), Combo 2 extended the game duration by 49.4 s (+851.7%, $9.52\times$) and increased the traveled distances by up to $7.9\times$ (pursuer) and $9.5\times$ (evader).

These pronounced deltas indicate that the observed behavior is not a minor numerical variation but a solver-induced strategic asymmetry: EA-QPSO enables the evader to repeatedly exploit distinct local basins of the nonconvex receding-horizon objective and to execute rapid, high-curvature maneuver switching, whereas the pursuer optimized by fmincon is more prone to local-optimum trapping and delayed reaction. Consequently, Combo 2 was adopted as a deliberate hard-case benchmark for subsequent comparisons—fixing the pursuer to fmincon and varying the evader’s optimizer—because it amplifies performance differences and better discriminates an algorithm’s ability to sustain effective evasion under strong nonconvexity and adversarial interaction.

5.1.2. Comparison Among Swarm-Based Optimizers (Evader Side)

To further examine the impact of the evader-side optimizer on the closed-loop game outcome, we conducted a controlled comparison in which the pursuer is fixed to the local solver fmincon, whereas the evader alternatively adopted EA-QPSO, DE, and MPA to solve its receding-horizon optimization at each MPC step. All experiments were performed in the same environment and with identical MPC settings and constraints, so that the only varying factor was the evader’s optimization method. The initial states were set to pursuer = $[14, 14, \frac{\pi}{4}]$ and evader = $[16, 16, \frac{\pi}{2}]$.

Figure 6 illustrates the time evolution of the evader’s objective value. Overall, EA-QPSO and MPA exhibit very similar macroscopic trends and can sustain higher objective levels than DE, suggesting that both swarm-based strategies are more effective in exploring the multi-modal, nonconvex landscape induced by the receding-horizon min–max formulation. Importantly, EA-QPSO demonstrates a more stable and persistent high-level objective profile during critical maneuvering intervals, which indicates improved robustness against premature stagnation and a stronger ability to keep discovering alternative maneuver modes as the engagement evolves.

This advantage is consistent with the design of EA-QPSO. By incorporating chaotic initialization and adaptive exploration control (together with stagnation-handling perturbations), EA-QPSO enhances population diversity and mitigates the tendency of standard metaheuristics to cluster early around suboptimal basins. As a result, EA-QPSO is more likely to generate evasive control sequences that repeatedly “reset” the pursuer’s short-horizon interception geometry, yielding stronger closed-loop evasion behavior than competing swarm optimizers with less explicit diversity management.

To further evaluate the robustness of different evader-side optimizers, we conducted 10 randomized trials under the same environment and MPC settings. In each trial, the initial positions and headings of both AUVs were randomly sampled within a 30×30 map range (the random seed was recorded for reproducibility). The pursuer always solved its MPC subproblem using MATLAB fmincon with the SQP algorithm, whereas the evader alternatively used DE, MPA, or the proposed EA-QPSO. The performance metric is the capture time (simulation time until the pursuer enters the capture radius).

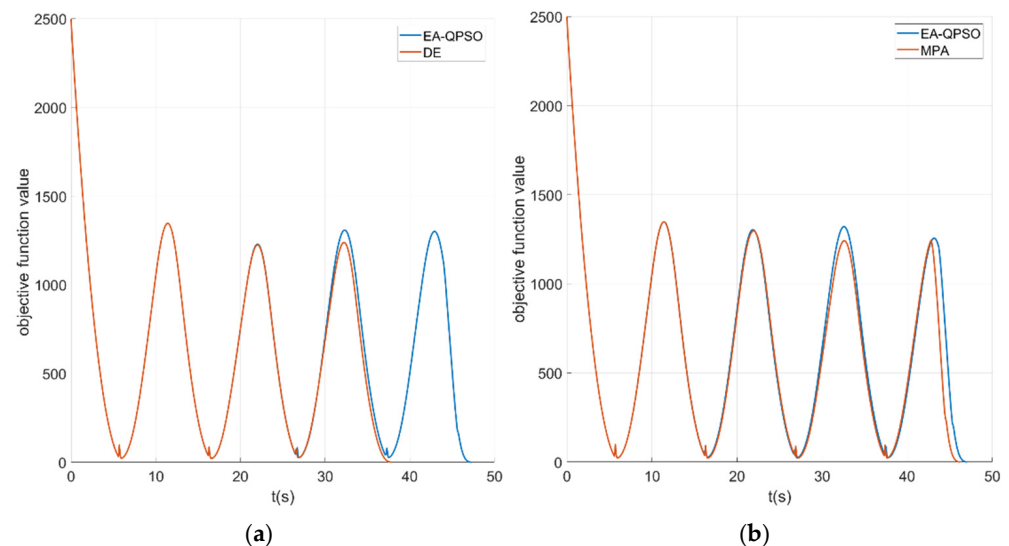


Figure 6. Trend of objective function value with different algorithms. (a) Comparison of EA-QPSO and DE. (b) Comparison of EA-QPSO and MPA.

Figure 7 reports the capture time statistics over 10 trials. The average capture times were 40.4 s (MPA), 36.2 s (DE), and 43.0 s (EA-QPSO). Overall, EA-QPSO yielded the longest engagements on average, indicating that it enables the evader to maintain more effective evasive maneuvers against a pursuer optimized by a local SQP solver. Meanwhile, the variability across trials suggests that the outcome remains sensitive to the initial relative geometry, which is expected for nonconvex receding-horizon pursuit–evasion games. Although no statistical significance test is claimed here, the mean and dispersion across trials consistently indicate that EA-QPSO tends to prolong capture time compared with DE and MPA under the same settings.

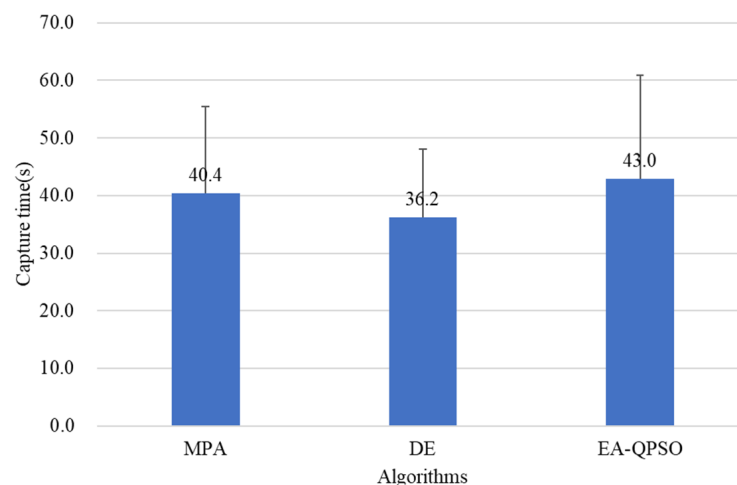


Figure 7. Capture time over 10 randomized trials with the pursuer solved by fmincon (SQP) and the evader solved by DE, MPA, or EA-QPSO. The bars denote the mean capture time and the error bars denote standard deviation.

As an illustrative example, Figure 8 visualizes the trajectories of a representative trial, showing that EA-QPSO tends to generate more frequent high-curvature maneuvers, which can repeatedly alter the interception geometry.

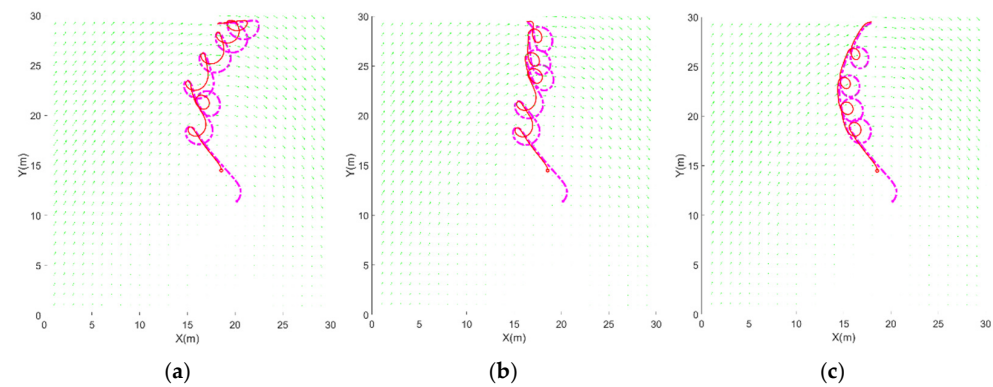


Figure 8. Representative trajectories in a selected randomized trial. The pursuer used `fmincon` (SQP), whereas the evader used (a) EA-QPSO, (b) MPA, and (c) DE. The pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

5.1.3. EA-QPSO and Original QPSO Comparison

To evaluate the benefit of the proposed EA-QPSO over the standard QPSO in solving the receding-horizon optimization of the pursuit–evasion game, we design a controlled comparison in which the pursuer always used `fmincon`, whereas the evader used either standard QPSO or EA-QPSO. In this way, the only changing factor was the evader-side optimizer, enabling an ablation-style assessment of how the EA mechanisms affect the game outcome and the closed-loop MPC behavior.

Both methods were tested under the same computational budget (identical population size and maximum iteration number) and under identical dynamic constraints and initial conditions. Specifically, the evader speed limit was fixed at 0.6 m/s, and the control bounds were kept unchanged across methods. The initial states were set to pursuer at $[3, 3, \pi/4]$ and evader at $[5, 5, \pi/2]$. Performance was assessed primarily through (i) capture/escape duration and (ii) the time history of the pursuer objective value, which reflects the evolving difficulty of the interception problem under the evader's optimized actions.

Figure 9 compares the objective-function trajectories produced by standard QPSO and EA-QPSO. The results show a clear and substantial advantage of EA-QPSO: when the evader uses standard QPSO, it was captured after 67 s, whereas with EA-QPSO the evader maintained escape for 601.6 s (approximately $9.0\times$ longer). This large increase in survival time indicates that EA-QPSO generates evasion actions that are consistently more disruptive to the pursuer's short-horizon interception plan.

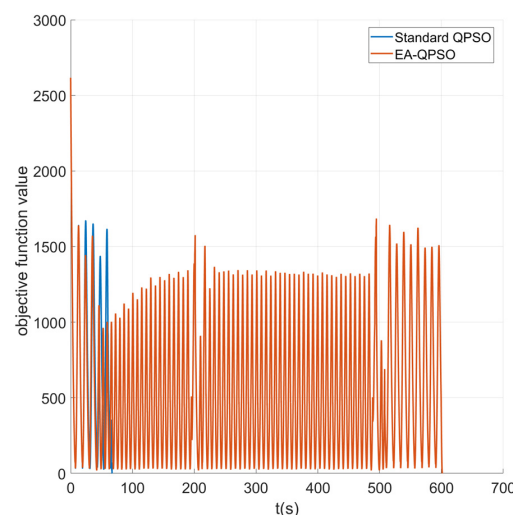


Figure 9. Objective function plot of the original QPSO and the EA-QPSO.

Moreover, the objective curves provide insight into why EA-QPSO improves performance. Although EA-QPSO may yield slightly lower peaks in some segments compared with standard QPSO, it sustains longer-lasting, repeated high-amplitude oscillations in the pursuer objective over a much longer horizon. This pattern is consistent with the following interpretation: as the pursuer approaches and the game becomes more nonconvex, EA-QPSO is more capable of escaping poor local solutions and discovering alternative maneuver modes (abrupt heading changes). As a result, the evader can repeatedly “reset” the pursuer’s predicted capture geometry, thereby delaying interception and extending the overall engagement duration.

5.2. Capture Condition Verification

To verify the capture-feasibility condition under bounded speed and bounded turning-rate constraints, we investigated how the outcome changed with the evader–pursuer speed ratio $\gamma = v_e/v_p$. In this study, the pursuer speed was normalized to $v_p \in [0, 1]$, and the evader speed was set as $v_e = \gamma v_p$. The turning-rate constraints were kept fixed, with the evader angular velocity $r_e \in [-\frac{\pi}{2}, \frac{\pi}{2}]$ and the pursuer angular velocity $r_p \in [-\frac{\pi}{4}, \frac{\pi}{4}]$. Figure 10 reports the representative case $\gamma = 0.7$.

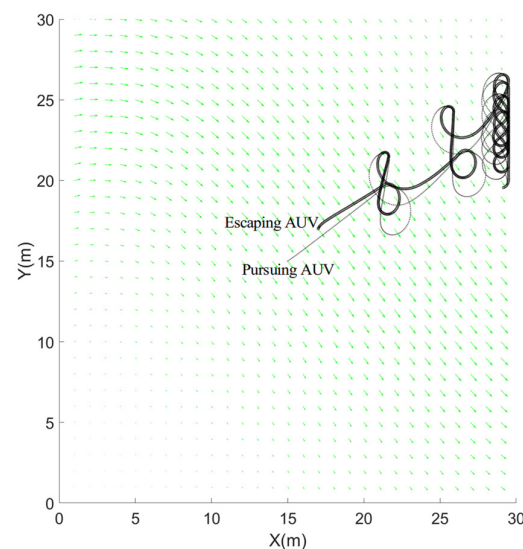


Figure 10. Failure of capture at $\gamma = 0.7$.

The simulation shows that when $\gamma = 0.7$, the pursuer failed to achieve capture; instead of converging to the evader, the pursuer trajectory gradually evolved into a quasi-periodic/limit-cycle (“dead-loop”) chasing pattern. This indicates that, under the given kinematic constraints, the pursuer cannot generate sufficient heading-rate authority to continuously reduce the relative distance once the evader maintains aggressive maneuvering.

This phenomenon can be interpreted from a maneuverability (curvature) perspective. With bounded speed and turn rate, an agent’s maximum achievable curvature satisfies approximately $K_{\max} \propto \omega_{\max}/v$. Therefore, effective capture requires the pursuer to possess a sufficiently large “relative turning capability” compared with the evader, so that it can match or exceed the evader’s maneuver-induced curvature in the relative motion. In our setting, the evader enjoys both a relatively large turning-rate bound and a higher speed ratio, leading to a condition where the pursuer’s maximum curvature becomes insufficient, $K_{p,\max} < \gamma K_{e,\max}$, and the pursuer is forced into a circular tracking pattern rather than a convergent interception trajectory.

Consequently, the results in Figure 10 confirm a non-capturable regime around $\gamma = 0.7$ for the current constraint configuration. This provides a practical capture-feasibility guideline: to guarantee capture under the same turning-rate limits, the pursuer must either (i) operate at a sufficiently higher speed advantage (smaller γ), or (ii) be given a higher turning-rate bound (larger r_p) so that its effective curvature authority exceeds that of the evader.

5.3. Sensitivity Analysis of Ocean Current Intensity

To evaluate the robustness of the proposed EA-QPSO algorithm under varying environmental dynamics, a sensitivity analysis was conducted. We introduced an intensity scaling factor to modulate the flow velocity fields. We selected three representative intensity levels: $\{0.2, 0.5, 0.8\}$. For each intensity level, five sets of randomized experiments were conducted to record the Success Rate, Average Capture Time, and Average Pursuing Distance. The statistical results are summarized in Table 3.

Table 3. Impact of Ocean Current Intensity on Pursuit Performance.

Current Intensity	Success Rate (%)	Avg. Capture Time (s)	Avg. Pursuing Distance (m)
0.2	100	17.5	16.1
0.5	100	15.2	15.1
0.8	100	18.9	18.2

The results in Table 3 reveal important insights into the interaction between environmental disturbances and the control strategy. The proposed EA-QPSO algorithm achieved a 100% success rate across all tested current intensities. This demonstrates that the algorithm effectively maintains feasibility and convergence even when the environmental parameters deviate significantly from the baseline. The capture costs (time and distance) do not follow a strictly linear trend. This phenomenon can be attributed to the spatial heterogeneity of the ocean current; in certain random configurations, a moderate current vector may partially align with the pursuer's heading, acting as a tailwind that aids acceleration. However, as the intensity increased to 0.8, both the average capture time and distance reached their maximums. This indicates that stronger currents generally impose greater disturbances, requiring the pursuer to expend more control effort to compensate for drift and maintain a valid interception trajectory.

In summary, although variations in ocean current intensity affect the specific capture metrics, the EA-QPSO framework remains reliable and effective within the tested range of environmental disturbances.

5.4. Comparison Simulation of Non-Dominant Evasion Strategy for Escaping AUV

To demonstrate the advantage of the proposed game-theoretic MPC in AUV pursuit–evasion, we compared it with the non-game MPC baseline (described in Section 3.1) under the same simulation setting. In the non-game baseline, each AUV optimizes its own MPC objective without explicitly modeling the opponent as a strategic optimizer, which often leads to more myopic and easier-to-anticipate closed-loop behaviors. In contrast, game-theoretic MPC explicitly incorporates the opponent's optimizing response within the finite-horizon receding-horizon formulation, thereby producing strategies that are more consistent with adversarial interaction and better suited to pursuit–evasion games.

For fairness, both AUVs use the same numerical optimizer (EA-QPSO) to solve their respective MPC subproblems, and the escaping AUV is constrained to be slower, with a maximum speed set to 0.6 times that of the pursuing AUV. Figure 11a presents the trajectories when both AUVs use non-game MPC, whereas Figure 11b shows the trajectories when both AUVs adopt game-theoretic MPC.

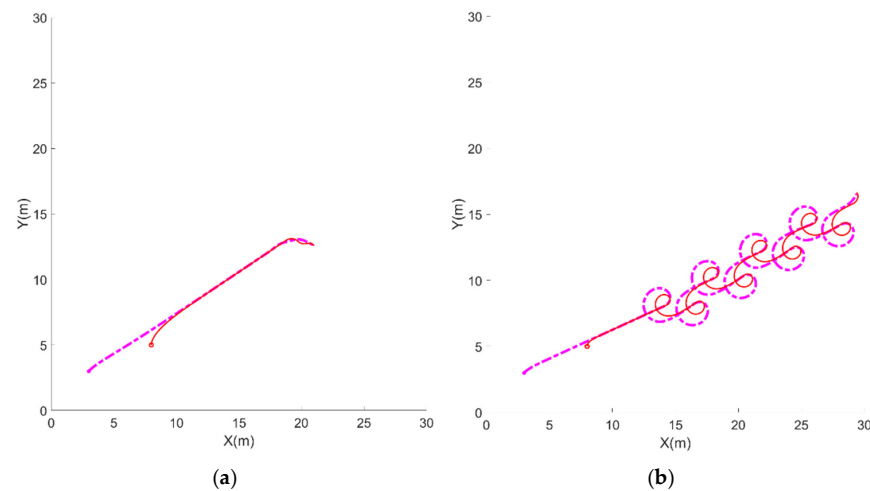


Figure 11. Simulation of the game-theoretic MPC and non-game MPC. (a) Both with non-game MPC. (b) Both with game-theoretic MPC. The pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

The quantitative results reveal a substantial difference in engagement difficulty. Under non-game MPC, the evader travels 16.032 m and is captured after 22.40 s (225 steps), whereas the pursuer travels 20.969 m. Under game-theoretic MPC, the engagement lasts 96.40 s (965 steps), during which the evader travels 57.826 m and the pursuer travels 85.393 m. Compared with the baseline, game-theoretic MPC increases the capture time by +74.0 s, i.e., $4.30\times$ longer, and increases the evader's traveled distance by $3.61\times$ (from 16.032 m to 57.826 m), indicating significantly more effective sustained evasion despite the evader's lower speed.

These results highlight the key benefit of game-theoretic MPC: by explicitly accounting for strategic coupling, it discourages “one-sided” plans that are optimal only under an implicit or inaccurate opponent model. Instead, both agents continuously replan against an anticipating opponent, which tends to generate more adversarial, less predictable maneuver sequences and makes interception more difficult in closed loop. Consequently, the proposed game-theoretic MPC yields behaviors that are closer to an approximate equilibrium response under the finite-horizon game formulation and provides a more reliable framework for pursuit–evasion decision-making in dynamic ocean environments.

5.5. Extension to 3D: Solver Robustness Verification in Constrained NMPC

It is important to explicitly address the change in problem formulation for the 3D scenarios. Although the 2D analysis utilized a game-theoretic min–max MPC to solve for approximate Nash equilibria, extending this coupled adversarial optimization directly to 3D environments presents significant challenges. The introduction of 6-DOF dynamics (specifically coupled pitch and yaw maneuvering) and spatial obstacles drastically increases the dimensionality and non-convexity of the solution space. Solving a min–max optimization problem at every time step under these conditions imposes a computational burden that currently exceeds real-time feasibility for compact AUV processors.

Therefore, as a practical methodological trade-off, we adopted a standard (non-game) NMPC formulation for the 3D experiments. In this section, the primary research objective shifts from analyzing strategic equilibrium to evaluating the robustness and stability of the proposed EA-QPSO solver when facing high-dimensional constraints. This setup allows us to isolate and verify the optimizer's ability to find feasible trajectories in complex 3D environments where traditional gradients (e.g., `fmincon`) or standard heuristics often fail. The realization of a full real-time 3D game-theoretic MPC remains a subject for future work, potentially requiring parallel computing or offline learning-based acceleration.

In 3D engagements, solving the full game-theoretic min–max MPC often leads to substantially longer interactions, and capture may become rare within a practical simulation horizon. This is primarily because the additional degree of freedom (pitch) provides the evader with more maneuvering options: by jointly adjusting yaw and pitch, the evader can continuously reshape the line-of-sight (LOS) geometry and disrupt the pursuer’s short-horizon interception plan. Moreover, when the evader has a significantly higher angular-rate authority than the pursuer, it can execute frequent high-curvature maneuvers, whereas the pursuer, limited by tighter turn-rate bounds, cannot track the induced LOS rotation sufficiently fast to ensure consistent distance reduction, which may result in non-convergent circling and quasi-periodic chase patterns.

For these reasons, repeatedly solving the full 3D min–max problem at every sampling instant would yield very long episodes and a heavy online computational burden due to both increased step count and higher dimensional nonconvex optimization. Therefore, in the 3D section, we adopt a non-game MPC formulation and perform a controlled solver comparison: the model, horizon, and constraints were fixed, and only the numerical optimizer was varied, so that solver robustness and closed-loop performance can be assessed more transparently in a challenging 3D-constrained MPC setting.

To extend to three-dimensional environments, the AUV position in the inertial frame was defined as $\mathbf{p} = [x, y, z]^T$ and the attitude was parameterized by the pitch angle θ and yaw angle ψ , both bounded within $[-\frac{\pi}{4}, \frac{\pi}{4}]$ to reflect practical motion limits and to avoid singular configurations. Under these assumptions, the 3D kinematics can be written in compact form as follows:

$$\begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \end{bmatrix} = \begin{bmatrix} J_1(\eta_2) & O_{2 \times 2} \\ O_{2 \times 1} & J_2(\eta_2) \end{bmatrix} v = J(\eta) v \quad (26)$$

$$J_1(\eta_2) = \begin{bmatrix} \cos\psi\cos\theta \\ \sin\psi\cos\theta \\ -\sin\theta \end{bmatrix} \quad (27)$$

$$J_2(\eta_2) = \begin{bmatrix} 1 & 0 \\ 0 & 1/\cos\theta \end{bmatrix} \quad (28)$$

In 3D pursuit–evasion, the control vector was selected as $v = [u, q, r]^T$, where u denotes the surge (longitudinal) speed and q and r are the pitch-rate and yaw-rate commands, respectively. Compared with the 2D case, the additional pitch channel introduced stronger coupling and a higher-dimensional, more nonconvex optimization landscape for the receding-horizon game, making the closed-loop strategy computation more sensitive to local minima and feasibility violations.

To assess the effectiveness and robustness of different solvers in this 3D MPC setting, we compared the pursuers applying the `fmincon` and the evaders using `fmincon`, EA-QPSO, DE, MPA, and GA under the same scenario (identical constraints and initialization). Table 4 summarizes the resulting evading and pursuing distances, and Figures 12–16 visualize representative 3D trajectories together with their corresponding control histories.

Table 4. Distance comparison by different optimization algorithms. N/A means not applicable.

Algorithms	Evading Distance	Pursuing Distance
<code>fmincon</code>	24.20 m	31.02 m
MPA	25.52 m	31.90 m
DE	25.19 m	31.60 m
GA	N/A	N/A
EA-QPSO	25.60 m	32.00 m

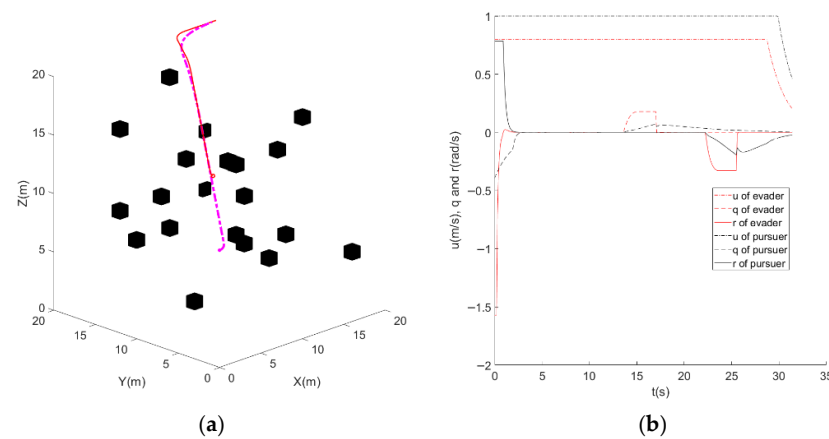


Figure 12. Simulation of the dual AUV pursuit–evasion game in a 3D environment using fmincon, showing late-stage local optimum (reduced velocities of both parties). (a) 3D pursuit–evasion game using fmincon and (b) its corresponding velocity change. In (a) the pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

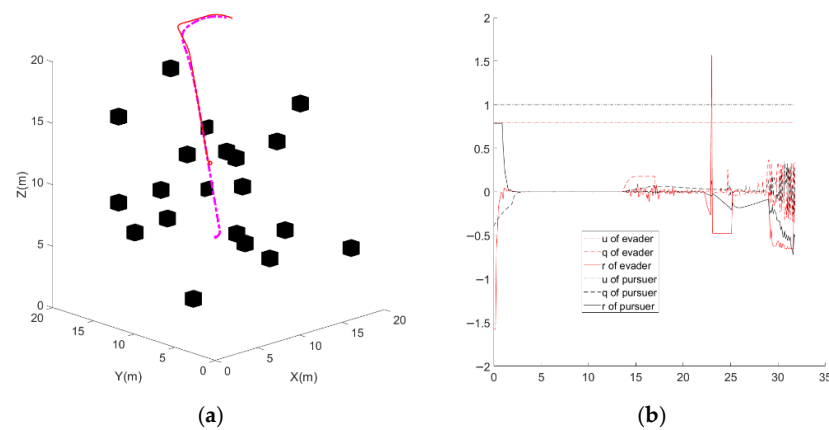


Figure 13. Simulation of the dual AUV pursuit–evasion game in a 3D environment using MPA, with less favorable angular-rate decisions compared to EA-QPSO. (a) 3D pursuit–evasion game using MPA and (b) its corresponding velocity change. In (a) the pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

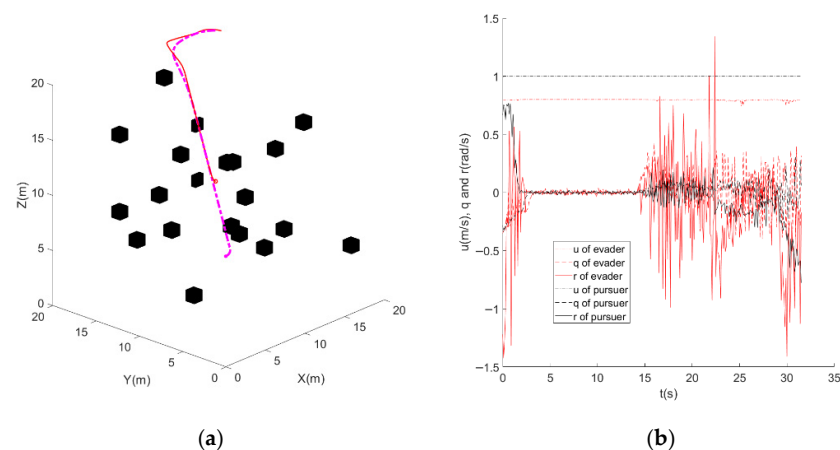


Figure 14. Simulation of the dual AUV pursuit–evasion game in a 3D environment using DE, with evader’s control signals exhibiting non-beneficial oscillations. (a) 3D pursuit–evasion game using DE and (b) its corresponding velocity change. In (a) the pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

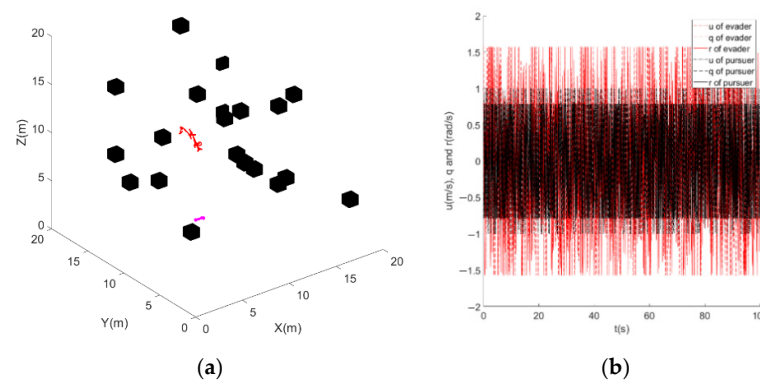


Figure 15. Simulation of the dual AUV pursuit–evasion game in a 3D environment using GA, with failure to return feasible or consistent solutions. (a) 3D pursuit–evasion game using GA and (b) its random oscillation velocity change. In (a) the pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

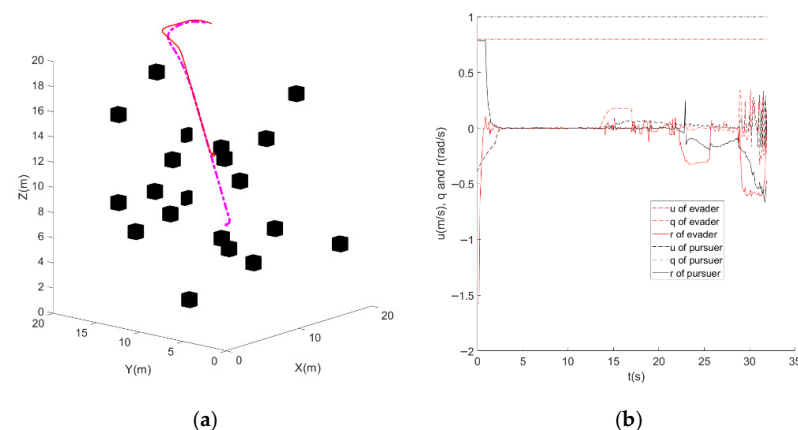


Figure 16. Simulation of the dual AUV pursuit–evasion game in a 3D environment using EA-QPSO, with the best linear and angular velocity decisions for both parties. (a) 3D pursuit–evasion game using EA-QPSO and (b) velocity change for the pursuit–evasion game. In (a) the pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

The comparison reveals distinct algorithmic behaviors in 3D. GA frequently fails to return feasible or consistent solutions, leading to unstable control sequences and ineffective pursuit–evasion behaviors. *fmincon* can provide feasible solutions early on but tends to stagnate as the engagement becomes more nonlinear, which manifests as late-stage local-optimum trapping and overly conservative control updates. DE improves robustness relative to *fmincon*, yet the evader’s control signals exhibit non-beneficial oscillations that reduce overall strategy quality. MPA and EA-QPSO achieve comparable performance in terms of maintaining aggressive speed profiles, indicating that both can explore the search space effectively; however, EA-QPSO produces more favorable angular-rate decisions during critical maneuvering phases, which translates into better closed-loop outcomes overall.

In summary, the 3D results support the suitability of EA-QPSO-type methods for solving receding-horizon pursuit–evasion games with coupled attitude dynamics and hard bounds. In particular, EA-QPSO exhibits strong global-search capability and stable performance in high-dimensional decision spaces, enabling the controller to compute more reliable responses under the finite-horizon MPC formulation. From an implementation perspective, EA-QPSO is also attractive because it involves only a small number of hyperparameters (e.g., contraction–expansion coefficient and population size), resulting in lower tuning burden and reduced sensitivity compared with GA-style operators that require careful crossover and mutation design.

5.6. Offline Policy Generation and Online Queries

AUVs are often equipped with lightweight onboard computers (e.g., Jetson Nano). In such cases, running EA-QPSO online may not reliably deliver optimization results within the required control period. To enable real-time deployment, we constructed an offline policy table in advance and performed fast online queries during operation. The overall procedure is illustrated in Figure 17.

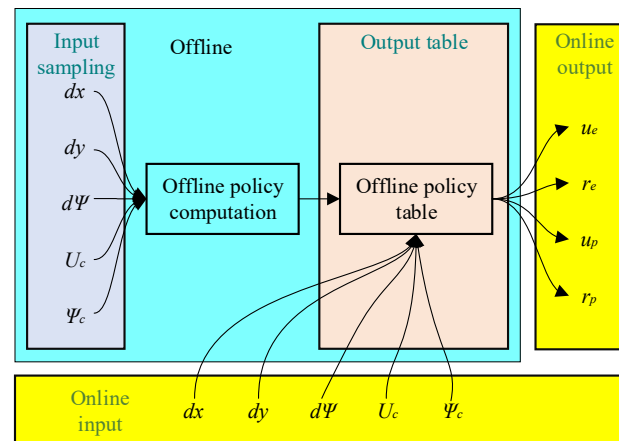


Figure 17. Offline policy table construction and online queries for the pursuit–evasion game.

Specifically, the relative pose between the pursuer and the evader (dx , dy , and $d\psi$), together with the ocean-current magnitude U_c and direction ψ_c , were selected as the inputs. These variables were discretized via grid sampling, and for each grid point we computed the corresponding control policies for both the pursuer and the evader offline. The resulting policies were stored in a lookup table.

During online execution, the current relative pose and local current estimates were used to query the table, yielding control commands for both agents with negligible computational costs. In this way, both the pursuer and the evader can operate in real time on resource-limited onboard hardware. The simulation result is shown in Figure 18. The evader is able to perform sharp turns when the pursuer approaches closely; however, near the boundary, the evader may lose feasible escape directions and is eventually captured, which is almost the same as the online computation. Overall, offline policy generation significantly reduces the online computational burden while preserving responsive behaviors in the pursuit–evasion game.

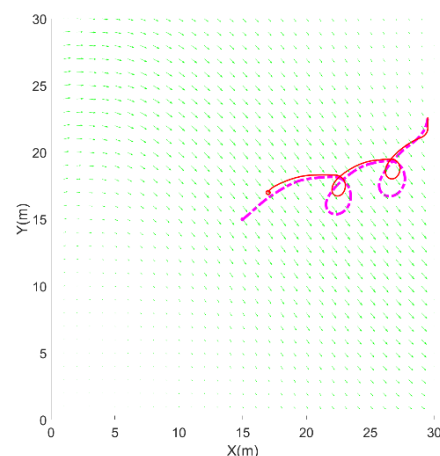


Figure 18. The pursuit–evasion game with the offline policy table. The pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

5.7. Sensitivity of MPC Horizon

To examine how the prediction horizon affects the closed-loop pursuit–evasion behavior, we conducted a sensitivity study by varying the MPC horizon while keeping all other settings (dynamics, constraints, current field, sampling time, and solver budget per step) unchanged. Figure 19 compares two representative cases with a short horizon $N = 5$ and a long horizon $N = 30$. With $N = 5$, the controller was relatively myopic and primarily reacted to the instantaneous line-of-sight geometry, resulting in smoother trajectories and faster convergence toward capture (capture time $T_c = 8.90$ s). In contrast, with $N = 30$ the controller anticipated longer-term interactions and enabled more proactive maneuver planning; the evader could exploit the extended look-ahead to perform frequent high-curvature turns and switch maneuver modes, repeatedly disrupting the pursuer’s interception plan and substantially prolonging the engagement (capture time $T_c = 55.90$ s). Overall, increasing the MPC horizon improves strategic foresight but also increases the nonconvexity and multi-modality of the underlying optimization problem, making solution quality more sensitive to the optimizer and computational budget.

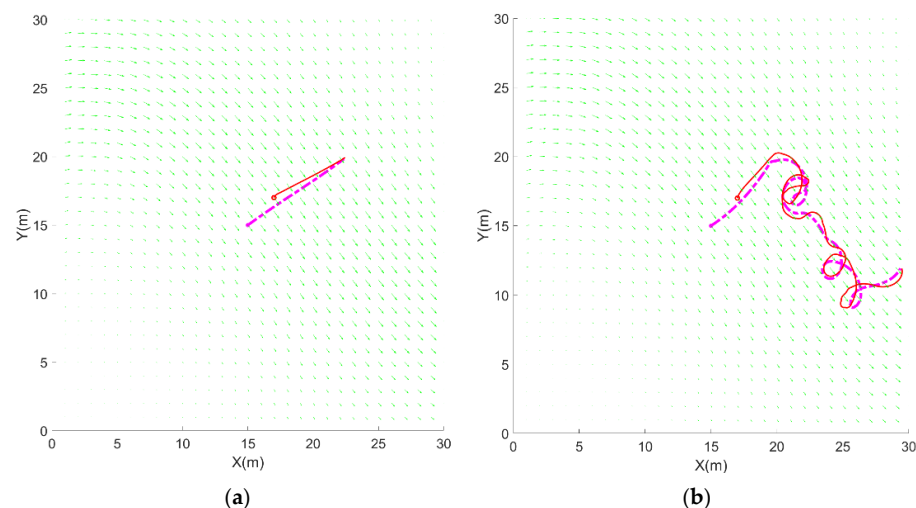


Figure 19. Sensitivity simulation of the MPC horizon: (a) MPC horizon (5) and (b) MPC horizon (30). The pursuer trajectory is shown in magenta and dash style, and the evader trajectory is shown in red and solid style.

6. Conclusions

This paper presents a game-theoretic Model Predictive Control (MPC) framework for dual-AUV pursuit–evasion in complex underwater environments with currents and obstacles. By integrating the proposed Enhanced Adaptive QPSO (EA-QPSO) solver, the framework was shown to effectively generate approximate finite-horizon saddle-point strategies under nonconvex constraints.

The simulation results demonstrate that the proposed EA-QPSO solver significantly enhances the reliability of the control strategy compared to traditional methods. Specifically, in hard-constraint scenarios, EA-QPSO increased the evader’s survival time by approximately nine times (from 67 s to 601.6 s) compared to standard QPSO. Furthermore, the sensitivity analysis confirmed that the algorithm maintains a 100% success rate in pursuit tasks under varying current intensities, effectively compensating for environmental disturbances.

However, several limitations of this study should be acknowledged. First, the perception system is assumed to be ideal, neglecting sonar noise and tracking errors, which are prevalent in real-world operations. Second, the ocean-current model is a simplified kinematic representation, and complex fluid-structure interactions (such as surface waves) were not explicitly modeled. Third, the 3D extension employed a non-game NMPC for-

mulation as a computational trade-off, focusing on solver robustness rather than full strategic equilibrium.

Future research will focus on bridging these gaps by integrating observation uncertainty into the MPC framework (e.g., Output Feedback MPC), developing computationally efficient solvers for real-time 3D gaming, and extending the scenario to multi-agent engagements.

Author Contributions: D.G.: investigation and writing—draft. M.C.: supervision, project administration, writing—code, writing—review and editing, and funding acquisition. Y.Z.: data collection and validation. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (52371331).

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: We gratefully acknowledge the support we received from the National Natural Science Foundation of China.

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

References

1. Zhou, Z.; Liu, J.; Yu, J. A survey of underwater multi-robot systems. *IEEE/CAA J. Autom. Sin.* **2021**, *9*, 1–18. [\[CrossRef\]](#)
2. Yuh, J.; West, M. Underwater robotics. *Adv. Robot.* **2001**, *15*, 609–639. [\[CrossRef\]](#)
3. Qu, X.; Zeng, L.; Qu, S.; Long, F.; Zhang, R. An Overview of Recent Advances in Pursuit–Evasion Games with Unmanned Surface Vehicles. *J. Mar. Sci. Eng.* **2025**, *13*, 458. [\[CrossRef\]](#)
4. Xu, J.; Zhang, Z.; Wang, J.; Han, Z.; Ren, Y. Multi-AUV pursuit-evasion game in the Internet of Underwater Things: An efficient training framework via offline reinforcement learning. *IEEE Internet Things J.* **2024**, *11*, 31273–31286. [\[CrossRef\]](#)
5. Guo, X.; Cui, R.; Yan, W. Pursuit-evasion games of marine surface vessels using neural network-based control. *IEEE Trans. Syst. Man Cybern. Syst.* **2024**, *55*, 18–27. [\[CrossRef\]](#)
6. Weintraub, I.E.; Pachter, M.; Garcia, E. An Introduction to Pursuit-evasion Differential Games. In Proceedings of the 2020 American Control Conference (ACC), Denver, CO, USA, 1–3 July 2020; pp. 1049–1066. [\[CrossRef\]](#)
7. Isaacs, R. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*; Courier Corporation: North Chelmsford, MA, USA, 1999.
8. Bravo, L.; Ruiz, U.; Murrieta-Cid, R. A pursuit-evasion game between two identical differential drive robots. *J. Frankl. Inst.* **2020**, *357*, 5773–5808. [\[CrossRef\]](#)
9. Qi, Q.; Zhang, X.; Guo, X. A deep reinforcement learning approach for the pursuit evasion game in the presence of obstacles. In *2020 IEEE International Conference on Real-time Computing and Robotics (RCAR)*; IEEE: Piscataway, NJ, USA, 2020; pp. 68–73. [\[CrossRef\]](#)
10. Zhang, R.; Zong, Q.; Zhang, X.; Dou, L.; Tian, B. Game of drones: Multi-UAV pursuit-evasion game with online motion planning by deep reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2022**, *34*, 7900–7909. [\[CrossRef\]](#) [\[PubMed\]](#)
11. Wang, Y.; Wang, Y.; Cui, R.; Guo, X.; Yan, W. Game of Marine Robots: USV pursuit evasion game using online reinforcement learning. In *2023 IEEE International Conference on Development and Learning (ICDL)*; IEEE: Piscataway, NJ, USA, 2023; pp. 121–126. [\[CrossRef\]](#)
12. Kaelbling, L.P.; Littman, M.L.; Moore, A.W. Reinforcement learning: A survey. *J. Artif. Intell. Res.* **1996**, *4*, 237–285. [\[CrossRef\]](#)
13. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2018. [\[CrossRef\]](#)
14. Al-Talabi, A.A.; Schwartz, H.M. Kalman fuzzy actor-critic learning automaton algorithm for the pursuit-evasion differential game. In *2016 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*; IEEE: Piscataway, NJ, USA, 2016; pp. 1015–1022. [\[CrossRef\]](#)
15. Garcia, C.E.; Prett, D.M.; Morari, M. Model predictive control: Theory and practice—A survey. *Automatica* **1989**, *25*, 335–348. [\[CrossRef\]](#)
16. Morari, M.; Lee, J.H. Model predictive control: Past, present and future. *Comput. Chem. Eng.* **1999**, *23*, 667–682. [\[CrossRef\]](#)
17. Hespanha, J.P.; Prandini, M.; Sastry, S. Probabilistic pursuit-evasion games: A one-step Nash approach. In *Proceedings of the 39th IEEE Conference on Decision and Control (Cat. No.00CH37187)*; IEEE: Piscataway, NJ, USA, 2000; Volume 3, pp. 2272–2277. [\[CrossRef\]](#)
18. Eklund, J.M.; Sprinkle, J.; Sastry, S.S. Switched and Symmetric Pursuit/Evasion Games Using Online Model Predictive Control With Application to Autonomous Aircraft. *IEEE Trans. Control Syst. Technol.* **2012**, *20*, 604–620. [\[CrossRef\]](#)

19. Sani, M.; Robu, B.; Hably, A. Pursuit-evasion Games Based on Game-theoretic and Model Predictive Control Algorithms. In *2021 International Conference on Control, Automation and Diagnosis (ICCAD)*; IEEE: Piscataway, NJ, USA, 2021; pp. 1–6. [\[CrossRef\]](#)
20. Khatib, O. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [\[CrossRef\]](#)
21. Kumar, P.B.; Rawat, H.; Parhi, D.R. Path planning of humanoids based on artificial potential field method in unknown environments. *Expert Syst.* **2019**, *36*, e12360. [\[CrossRef\]](#)
22. Dong, J.; Zhang, X.; Jia, X. Strategies of pursuit-evasion game based on improved potential field and differential game theory for mobile robots. In *2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control*; IEEE: Piscataway, NJ, USA, 2012; pp. 1452–1456. [\[CrossRef\]](#)
23. Chen, M.; Yang, C.; Zhang, X.; Li, G. Pursuit-Evasion Game of Multiple Pursuers and Evaders with Intelligent Cooperation and Obstacle Avoidance in a Complex Environment. In *Proceedings of the Chinese Conference on Swarm Intelligence and Cooperative Control, Nanjing, China, 17–19 November 2023*; Springer Nature: Singapore, 2023; pp. 222–234. [\[CrossRef\]](#)
24. Dorothy, M.R.; Maity, D.; Shishika, D.; Von Moll, A. One Apollonius Circle is enough for many pursuit-evasion games. *Automatica* **2021**, *163*, 111587. [\[CrossRef\]](#)
25. Wang, Q.; Wu, K.; Ye, J.; Wu, Y.; Xue, L. Apollonius Partitions Based Pursuit-evasion Game Strategies by Q-Learning Approach. In *2022 41st Chinese Control Conference (CCC)*; IEEE: Piscataway, NJ, USA, 2022; pp. 4843–4848. [\[CrossRef\]](#)
26. Sun, S.; Zhou, J.; Zhang, X. Multiple-Pursuer/One-Evader Pursuit-Evasion Game in Dynamic Flowfields. *J. Guid. Control Dyn.* **2017**, *40*, 1627–1639. [\[CrossRef\]](#)
27. Khachumov, I.; Khachumov, V.; Talalaev, M. Modeling the Solution of the Pursuit-Evasion Problem Based on the Intelligent-Geometric Control Theory. *Appl. Sci.* **2023**, *13*, 2038. [\[CrossRef\]](#)
28. Sun, X.; Sun, B.; Su, Z. Cooperative pursuit-evasion game for multi-AUVs in the ocean current and obstacle environment. In *Proceedings of the International Conference on Intelligent Robotics and Applications, Hangzhou, China, 5–7 July 2023*; Springer Nature: Singapore, 2023; pp. 201–213. [\[CrossRef\]](#)
29. Bower, A.S. A Simple Kinematic Mechanism for Mixing Fluid Parcels across a Meandering Jet. *J. Phys. Oceanogr.* **1991**, *21*, 173–180. [\[CrossRef\]](#)
30. Chen, M.; Zhu, D. Optimal time-consuming path planning for autonomous underwater vehicles based on a dynamic neural network model in ocean current environments. *IEEE Trans. Veh. Technol.* **2020**, *69*, 14401–14412. [\[CrossRef\]](#)
31. Sun, J.; Xu, W.; Feng, B. A global search strategy of quantum-behaved particle swarm optimization. In *IEEE Conference on Cybernetics and Intelligent Systems, 2004*; IEEE: Piscataway, NJ, USA, 2004; Volume 1, pp. 111–116. [\[CrossRef\]](#)

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