Beyond Visual Range Missile Evasion with Model-based Reinforcement Learning

Ahmet Talha Çetin[1] and Emre Koyuncu[1]

***Abstract*— This paper presents a model-based reinforcement learning framework for autonomous aircraft evasion against missile threats in Beyond Visual Range (BVR) engagements. We propose a solution utilizing Dual Heuristic Programming (DHP), a model-based reinforcement learning architecture that approximates the gradient of the value function (costate) rather than the scalar value, thereby providing explicit directional guidance for control optimization. To learn the model, an online Recursive Least Squares (RLS) estimator is employed to identify local system Jacobians in real-time using only relative position observations. The control policy is optimized against a bounded, differentiable reward function that represents objectives of beyond visual range evasion. Monte Carlo simulations demonstrate that the proposed method achieves a higher survival rate and larger average miss distance compared to standard fixed-structure heuristic maneuvers across a diverse range of engagement geometries.**

I. Introduction

Autonomous missile evasion in Beyond Visual Range (BVR) air combat is a high-stakes control problem under limited situational awareness and severe time constraints. The aircraft–missile interaction is often modeled as a non-cooperative zero-sum differential game, with the evader maximizing survivability while the missile minimizes intercept distance [1]. A central difficulty is that the missile guidance law, seeker behavior, and aerodynamic characteristics are typically unobservable to the evader. Fixed maneuver templates or assumed missile models can therefore be brittle against modern adaptive threats, motivating online learning controllers that do not require prior knowledge of the adversary's internal logic.

To ensure robust performance in the presence of such parameter variations and unknown initial states, an adaptation mechanism is essential. Adaptive Optimal Control (AOC) provides a framework for deriving controllers that can handle these uncertainties while optimizing a performance index. Theoretically, the AOC problem can be solved via Dynamic Programming (DP) [2], which provides the exact solution to the Hamilton-Jacobi-Bellman (HJB) equation. However, the computational complexity of DP suffers from the *curse of dimensionality* making it intractable for real-time continuous systems. To overcome this, Reinforcement Learning (RL) based approaches [3], [4] and Approximate Dynamic Programming (ADP) have been developed. In ADP, function approximators—such as Artificial Neural Networks or linear basis structures—are employed to estimate the control policy and value function recurrently [5].

[1]Aerospace Research Center, Istanbul Technical University, Türkiye (cetinah16@itu.edu.tr, emre.koyuncu@itu.edu.tr)

Several ADP architectures solve the HJB equation forward in time [6], [7], [8], [9], [10], which have demonstrated stability in regulation and tracking problems. Within actor–critic frameworks, these methods are commonly classified into Heuristic Dynamic Programming (HDP), Dual Heuristic Programming (DHP), Globalized Dual Heuristic Programming (GDHP), and Action-Dependent (AD) architectures [11].

Among them, HDP directly approximates the value function and may converge slowly in complex or high-dimensional state spaces. DHP instead estimates the value-function gradient, or costate, providing directional information for policy improvement. This motivates a DHP architecture augmented with online system identification, where local system Jacobians are estimated in real time for adaptive missile evasion under unknown and time-varying dynamics.

For BVR air combat, reinforcement learning has been used to construct control barrier functions from learned value functions that bound the minimum future aircraft–missile separation, providing explicit survivability guarantees during missile evasion [12]. Complementary work in [13] has further demonstrated that RL-based risk estimation can be used as a decision-support tool for thrust-aided BVR evasion. [14] proposed an autonomous UCAV evasive maneuvering framework for BVR air-to-air missile engagements, formulating missile evasion as a hierarchical multi-objective optimization problem solved using a multi-objective evolutionary algorithm. Game-theoretic approaches have also been applied to missile-aware aircraft navigation, where safe BVR maneuvering is formulated as a zero-sum differential game between the aircraft and an adversarial missile–target coalition, yielding closed-loop controllers that ensure survivability in the presence of energy-bleeding coasting missiles [15].

Despite these advances, model-free RL methods require extensive training data and struggle with sample efficiency. Evolutionary and game-theoretic approaches often rely on known or assumed threat models, limiting their applicability against adversaries employing unknown or adaptive guidance laws.

To address these limitations, this paper develops a model-based reinforcement learning framework for autonomous aircraft evasion in BVR missile engagements with unknown and time-varying threat dynamics. The evasion task is formulated as an optimal control problem and solved using a Dual Heuristic Programming framework that approximates the gradient of the value function (costate) to guide policy optimization. Online Recursive Least Squares is employed to identify local system Jacobians directly from relative position observations, enabling adaptation without prior knowledge of

the missile guidance law or full state observability. Coupled with a bounded, differentiable reward balancing line-of-sight rate generation and separation, the resulting policy adapts to engagement geometry and missile behavior, achieving improved survivability and miss distance compared to representative fixed-structure evasion strategies.

The remainder of this paper is organized as follows. Section II presents the aircraft and missile models and controllers. Section III formalizes the optimal evasion task. Section IV details the DHP architecture, RLS-based system identification, and bounded-rational reward design. Section V reports numerical results, and Section VI concludes the paper.

II. PRELIMINARIES

A. Notation

Throughout this paper, $\mathbb{R}$ denotes the real numbers and $\mathbb{Z}_{\geq 0}$ the non-negative integers. Vectors and matrices are denoted by bold lowercase and uppercase symbols, respectively; $(\cdot)^\top$ denotes transpose and $\|\cdot\|$ the Euclidean norm.

At discrete time $k \in \mathbb{Z}_{\geq 0}$, $\mathbf{s}_k \triangleq \mathbf{s}(t_k)$. The operator $\nabla_{\mathbf{x}}(\cdot)$ denotes the gradient with respect to $\mathbf{x}$, and the Jacobian of $\mathbf{f} : \mathbb{R}^n \to \mathbb{R}^m$ is $\mathbf{J}_{\mathbf{f}}(\mathbf{x}) \triangleq \frac{\partial \mathbf{f}}{\partial \mathbf{x}} \in \mathbb{R}^{m \times n}$.

We use the North-East-Down inertial frame $\mathcal{F}_i$, aircraft wind frame $\mathcal{F}_a$, and missile wind frame $\mathcal{F}_m$. The matrix $\mathbf{R}_a^b$ rotates vectors from $\mathcal{F}_a$ to $\mathcal{F}_b$, and $[\cdot]_\times$ satisfies $\mathbf{u} \times \mathbf{v} = [\mathbf{u}]_\times \mathbf{v}$.

B. Aircraft Model

1) Aircraft Dynamics: The aircraft translational dynamics are defined in the North-East-Down (NED) inertial reference frame $\mathcal{F}_i$. Let $\mathbf{r}_A = [x_A,\, y_A,\, z_A]^\top \in \mathbb{R}^3$ and $\mathbf{v}_A = [v_{Ax},\, v_{Ay},\, v_{Az}]^\top \in \mathbb{R}^3$ denote the aircraft position and velocity in the $\mathcal{F}_i$. The equations of motion are:

$$\dot{\mathbf{r}}_A = \mathbf{v}_A \tag{1a}$$

$$\dot{\mathbf{v}}_A = \mathbf{g} + \mathbf{R}_a^i \mathbf{a}_a \tag{1b}$$

$$\dot{\mathbf{R}}_a^i = \mathbf{R}_a^i [\boldsymbol{\omega}_a]_\times \tag{1c}$$

where $\mathbf{g} = [0,\, 0,\, g]^\top$ is the gravity vector expressed in $\mathcal{F}_i$. The vector $\mathbf{a}_a = [a_{ax}, 0, a_{az}]^\top$ represents the non-gravitational force expressed in the aircraft wind frame $\mathcal{F}_a$. We assume a coordinated flight condition with zero sideslip ($\beta = 0$), which implies zero lateral acceleration in the wind frame ($a_{a2} = 0$). The attitude dynamics are driven by $\boldsymbol{\omega}_a$, the angular velocity of $\mathcal{F}_a$ relative to $\mathcal{F}_i$, where $[\boldsymbol{\omega}_a]_\times$ denotes the corresponding skew-symmetric matrix and $\mathbf{R}_a^i$ is the rotation matrix transforming coordinates from $\mathcal{F}_a$ to $\mathcal{F}_i$.

Assuming coordinated flight with zero sideslip ($\beta = 0$), the aerodynamic and propulsive forces are resolved in the wind frame $\mathcal{F}_a$. Let T denote the thrust magnitude, D the drag, L the lift, and α the angle of attack. The specific force vector in $\mathcal{F}_a$ is given by

$$\mathbf{a}_a = \frac{1}{m} \begin{bmatrix} T\cos\alpha - \frac{1}{2}\rho V^2 S C_D(\alpha) \\ 0 \\ -T\sin\alpha - \frac{1}{2}\rho V^2 S C_L(\alpha) \end{bmatrix} \tag{2}$$

where ρ is air density, V is airspeed, S is reference area, and $C_L(\alpha)$ and $C_D(\alpha)$ are lift and drag coefficients. The x-axis component is thrust projected along the velocity direction minus drag, and the z-axis component is due to vertical thrust and aerodynamic lift.

Adopting the *Pseudo-Six-Degree-of-Freedom (Pseudo-6DOF)* formalism of [16], translational dynamics are force-driven while rotational motion is propagated through kinematic angular rates. The aircraft state is

$$\mathbf{x}_A = \begin{bmatrix} \mathbf{r}_A^\top & \mathbf{v}_A^\top & \mathbf{q}_A^\top \end{bmatrix}^\top \in \mathbb{R}^{10}. \tag{3}$$

where $\mathbf{q}_A$ is aircraft orientation represented in quaternions.

2) Aircraft Controllers: High-level guidance commands $\mathbf{a}_k$ are tracked by low-level PID controllers that generate wind-frame specific forces $\mathbf{a}_a$ and angular rates $\boldsymbol{\omega}_a$ for the Pseudo-6DOF aircraft model in eq. (1).

a) Thrust and Aerodynamic Force Computation: Airspeed and altitude are regulated via independent PID loops on $e_V = V_{\text{cmd}} - \|\mathbf{v}_A\|$ and $e_h = h_{\text{cmd}} - h$, producing the commanded thrust T and vertical load factor $n_{z,\text{cmd}}$, respectively. The angle of attack α is obtained from the wind-frame normal force balance by enforcing $a_{az} = -n_{z,\text{cmd}}\, g$, which, using eq. (2), yields

$$-T\sin\alpha - \frac{1}{2}\rho V^2 S C_L(\alpha) + n_{z,\text{cmd}}\, g = 0. \tag{4}$$

b) Angular Velocity Command Generation: Under the Pseudo-6DOF assumption, angular rates $\boldsymbol{\omega}_a = [p, q, r]^\top$ are generated kinematically to enforce coordinated turning ($\beta = 0$). A yaw–to–roll PID loop generates the commanded bank angle ϕ_{cmd} from the wrapped heading error, with roll dynamics modeled as a first-order lag

$$p = \frac{\phi_{\text{cmd}} - \phi}{\tau_{\text{roll}}} \tag{5}$$

The pitch and yaw rates required to sustain the commanded load factor are given by

$$q = \frac{g}{\|\mathbf{v}_A\|} \left(n_{z,\text{cmd}} - \cos\phi\cos\theta\right), \tag{6}$$

$$r = \frac{g}{\|\mathbf{v}_A\|} \sin\phi\cos\theta. \tag{7}$$

C. Missile Model

1) Missile Dynamics: Translational motion of the missile is expressed in $\mathcal{F}_i$, while control accelerations are applied in $\mathcal{F}_m$.

The missile state vector is defined as

$$\mathbf{x}_M = \begin{bmatrix} \mathbf{r}_M^\top & \mathbf{v}_M^\top & m_M \end{bmatrix}^\top \in \mathbb{R}^7. \tag{8}$$

where $\mathbf{r}_M = [x_M, y_M, z_M]^\top$, $\mathbf{v}_M = [v_{Mx}, v_{My}, v_{Mz}]^\top$, and m_M are missile position, velocity, and mass. The equations of motion are

$$\dot{\mathbf{r}}_M = \mathbf{v}_M, \tag{9a}$$

$$\dot{\mathbf{v}}_M = \mathbf{g} + \mathbf{R}_m^i \mathbf{a}_m, \tag{9b}$$

$$\dot{m}_M = -\dot{m}_{\text{prop}}\, \delta(t), \tag{9c}$$

where $\mathbf{R}_m^i$ rotates from missile wind frame to inertial frame, and $\delta(t)$ is a boost-sustain-coast throttle command:

$$\delta(t) = \begin{cases} \delta_{\max} & 0 \le t < t_{\text{boost}} \\ \delta_{\min} & t_{\text{boost}} \le t < t_{\text{boost}} + t_{\text{sustain}} \\ 0 & t \ge t_{\text{boost}} + t_{\text{sustain}} \end{cases} \tag{10}$$

where t_{boost} and t_{sustain} are boost and sustain durations.

Under skid-to-turn dynamics, the wind-to-inertial rotation matrix is computed from missile flight-path angle $\gamma_M = \arcsin(-v_{Mz}/\|\mathbf{v}_M\|)$ and heading $\xi_M = \arctan(v_{My}, v_{Mx})$:

$$\mathbf{R}_m^i = \left(\mathbf{R}_y(\gamma_M)\,\mathbf{R}_z(\xi_M)\right)^\top \tag{11}$$

where $\mathbf{R}_y(\cdot)$ and $\mathbf{R}_z(\cdot)$ are standard right-handed rotations about the y- and z-axes.

The axial wind-frame acceleration is given by

$$a_{m1} = \frac{T_M - D_M}{m_M}, \tag{12}$$

with thrust T_M and drag D_M modeled as

$$T_M = I_{\text{sp}} g_0 \dot{m}_{\text{prop}} \delta(t), \tag{13}$$

$$D_M = \frac{1}{2}\rho V_M^2 S_{\text{ref}} C_D(M), \tag{14}$$

where $V_M = \|\mathbf{v}_M\|$ and $C_D(M)$ is interpolated with Mach number $M = V_M/a$.

2) Proportional Navigation Guidance: Missile guidance uses true Proportional Navigation (PN) with skid-to-turn dynamics. Lateral commands are generated from line-of-sight (LOS) angular rate, while axial acceleration is governed by propulsion and drag.

Define the relative position $\mathbf{r}_{\text{rel}}$ and velocity $\mathbf{v}_{\text{rel}}$ as

$$\mathbf{r}_{\text{rel}} = \mathbf{r}_A - \mathbf{r}_M, \quad \mathbf{v}_{\text{rel}} = \mathbf{v}_A - \mathbf{v}_M. \tag{15}$$

The LOS unit vector $\hat{\boldsymbol{\ell}}$ and LOS rate $\dot{\boldsymbol{\ell}}$ are defined as

$$\hat{\boldsymbol{\ell}} = \frac{\mathbf{r}_{\text{rel}}}{\|\mathbf{r}_{\text{rel}}\|}, \qquad \dot{\boldsymbol{\ell}} = \frac{\mathbf{r}_{\text{rel}} \times \mathbf{v}_{\text{rel}}}{\|\mathbf{r}_{\text{rel}}\|^2} \tag{16}$$

With closing velocity $V_c = -\mathbf{v}_{\text{rel}}^\top \hat{\boldsymbol{\ell}}$, the inertial-frame PN command is

$$\mathbf{a}_{\text{cmd}}^i = N\, V_c\, \dot{\boldsymbol{\ell}} \times \frac{\mathbf{v}_M}{\|\mathbf{v}_M\|} - \mathbf{g} \tag{17}$$

where N is the navigation constant. Transforming to the missile wind frame gives

$$\mathbf{a}_m = (\mathbf{R}_m^i)^\top \mathbf{a}_{\text{cmd}}^i. \tag{18}$$

Only the lateral components of $\mathbf{a}_m$ are used for guidance; the axial component is governed by propulsion and drag.

III. Problem Formulation

Consider the continuous-time nonlinear dynamics of the aircraft-missile engagement given by:

$$\dot{\mathbf{x}}(t) = \mathcal{F}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{d}(t)) \tag{19}$$

$$\mathbf{s}(t) = \mathcal{H}(\mathbf{x}(t)) \tag{20}$$

where $\mathbf{x}(t) = [\mathbf{x}_A^\top,\ \mathbf{x}_M^\top]^\top \in \mathbb{R}^{n_{\mathbf{x}}}$ is the latent aircraft-missile state, $\mathbf{s}(t) \in \mathbb{R}^n$ is the observation available to the agent, $\mathbf{u}(t) \in \mathbb{R}^m$ is the evader control, and $\mathbf{d}(t) \in \mathbb{R}^p$ is the disturbance induced by missile guidance. The map $\mathcal{H}(\cdot)$ defines the observable features.

With sampling interval Δt, the learning agent observes

$$\mathbf{x}_{k+1} = \mathbf{f}(\mathbf{x}_k, \mathbf{u}_k, \mathbf{d}_k) \tag{21}$$

$$\mathbf{s}_k = \mathcal{H}(\mathbf{x}_k) \tag{22}$$

The missile is assumed to use an unknown guidance law, e.g., Proportional Navigation, acting on the true relative geometry, approximated by $\mathbf{d}_k \approx \pi_M(\mathbf{x}_k)$. Thus, the closed-loop system depends implicitly on the adversary's policy.

The evasion task is formulated as an MDP. The policy $\mathbf{u}_k = \pi(\mathbf{s}_k)$ maximizes discounted reward while balancing separation, LOS-rate generation, and control effort.

Problem 1 (Optimal Evasion): Consider the system eq. (21) where the pursuer dynamics $\mathbf{d}_k$ are unknown. Find the optimal policy π^* that maximizes the value function:

$$V^\pi(\mathbf{s}_k) = \sum_{j=k}^{\infty} \gamma^{j-k} r(\mathbf{s}_j, \pi(\mathbf{s}_j)) \tag{23}$$

subject to the system dynamics, where $\gamma \in (0, 1)$ is the discount factor and $r(\cdot)$ is a bounded rational reward function. The policy must remain effective over diverse aspect angles, closing velocities, and relative altitudes.

We use an observation vector containing relative position and velocity:

$$\mathbf{s} = \begin{bmatrix} \mathbf{r}_{\text{rel}} \\ \mathbf{v}_{\text{rel}} \end{bmatrix} = \begin{bmatrix} \mathbf{r}_A - \mathbf{r}_M \\ \mathbf{v}_A - \mathbf{v}_M \end{bmatrix} \in \mathbb{R}^6. \tag{24}$$

The action space comprises high-level aircraft commands:

$$\mathbf{u} = \begin{bmatrix} \Delta\psi_{\text{cmd}} & h_{\text{cmd}} & V_{\text{cmd}} \end{bmatrix}^\top \in \mathbb{R}^3, \tag{25}$$

representing heading-angle change, altitude, and airspeed commands tracked by the controllers in Section II-B.2.

IV. Methodology

This section presents the model-based RL method used to solve eq. (23). We use DHP, an adaptive critic architecture that approximates the value-function gradient to satisfy the Hamilton–Jacobi–Bellman (HJB) optimality conditions.

The architecture contains a Critic, an Actor, and an online model learner. The Critic in Section IV-A approximates the costate $\hat{\boldsymbol{\lambda}}_{\mathbf{w}_c}(\mathbf{s})$, the gradient of value with respect to the observable state. The Actor in Section IV-B represents the policy $\pi_{\mathbf{w}_a}(\mathbf{s}_k)$ and updates its parameters using Critic gradients. Since analytical dynamics are unavailable, the RLS learner in Section IV-C estimates local system Jacobians from observed transitions; Section IV-D gives the bounded reward.

A. The Critic: Value Gradient Approximation

The core of the DHP architecture is the Critic network, which estimates the costate vector $\boldsymbol{\lambda}(\mathbf{s})$. The costate is defined as the gradient of the value function $V(\mathbf{s})$ with respect to the system state [11]:

$$\boldsymbol{\lambda}(\mathbf{s}) \triangleq \nabla_{\mathbf{s}} V(\mathbf{s}) = \left[\frac{\partial V}{\partial s_1}, \ldots, \frac{\partial V}{\partial s_n}\right]^\top \in \mathbb{R}^n. \tag{26}$$

The value function satisfies the discrete-time Bellman optimality equation

$$V(\mathbf{s}_k) = r(\mathbf{s}_k, \mathbf{u}_k) + \gamma V(\mathbf{s}_{k+1}) \tag{27}$$

where the control input is generated by the policy $\mathbf{u}_k = \pi_{\mathbf{w}_a}(\mathbf{s}_k)$. By differentiating the Bellman equation, we obtain the recursive relationship for the costate:

$$\boldsymbol{\lambda}(\mathbf{s}_k) = \frac{\partial r_k}{\partial \mathbf{s}_k} + \left(\frac{\partial \mathbf{u}_k}{\partial \mathbf{s}_k}\right)^\top \frac{\partial r_k}{\partial \mathbf{u}_k} + \gamma \left(\frac{\partial \mathbf{s}_{k+1}}{\partial \mathbf{s}_k}\right)^\top \boldsymbol{\lambda}(\mathbf{s}_{k+1}). \tag{28}$$

where $r_k = r(\mathbf{s}_k, \mathbf{u}_k)$. This equation relates local reward gradients to future value gradients backpropagated through the closed-loop transition Jacobian $\frac{\partial \mathbf{s}_{k+1}}{\partial \mathbf{s}_k}$. Applying the chain rule yields:

$$\frac{\partial \mathbf{s}_{k+1}}{\partial \mathbf{s}_k} = \mathbf{A}_k + \mathbf{B}_k \frac{\partial \mathbf{u}_k}{\partial \mathbf{s}_k} \tag{29}$$

where $\mathbf{A}_k = \nabla_{\mathbf{s}} \mathbf{f}$ and $\mathbf{B}_k = \nabla_{\mathbf{u}} \mathbf{f}$ are the state and action Jacobians.

The Critic network, parameterized by weights $\mathbf{w}_c$, approximates the mapping $\hat{\boldsymbol{\lambda}}_{\mathbf{w}_c}(\mathbf{s}_k) \approx \boldsymbol{\lambda}(\mathbf{s}_k)$. To enhance training stability, a target critic network with weights $\mathbf{w}_c'$ is used to compute the temporal difference error vector:

$$\begin{aligned}\mathbf{e}_c(k) = \hat{\boldsymbol{\lambda}}_{\mathbf{w}_c}(\mathbf{s}_k) - \Bigg[&\frac{\partial r_k}{\partial \mathbf{s}_k} + \left(\frac{\partial \mathbf{u}_k}{\partial \mathbf{s}_k}\right)^\top \frac{\partial r_k}{\partial \mathbf{u}_k} \\ &+ \gamma \left(\frac{\partial \mathbf{s}_{k+1}}{\partial \mathbf{s}_k}\right)^\top \hat{\boldsymbol{\lambda}}_{\mathbf{w}_c'}(\mathbf{s}_{k+1})\Bigg].\end{aligned} \tag{30}$$

The weights are updated with

$$\mathbf{w}_c \leftarrow \mathbf{w}_c - \eta_c \left(\frac{\partial \hat{\lambda}_{\mathbf{w}_c}(\mathbf{s}_k)}{\partial \mathbf{w}_c}\right)^\top \mathbf{e}_c(k) \tag{31}$$

where η_c is learning rate of the critic network. The target critic network weights are computed with a soft update rule with a time constant $\tau \ll 1$:

$$\mathbf{w}_c' \leftarrow \tau \mathbf{w}_c + (1-\tau)\mathbf{w}_c'. \tag{32}$$

B. The Actor: Policy Update

The Actor maintains the policy $\mathbf{u}_k = \pi_{\mathbf{w}_a}(\mathbf{s}_k)$ and uses the Critic costate estimate to select actions that maximize the performance measure J.

The policy weights $\mathbf{w}_a$ are updated by gradient ascent in the direction of the value gradient with respect to the action:

$$\mathbf{w}_a \leftarrow \mathbf{w}_a + \eta_a \left(\frac{\partial \pi_{\mathbf{w}_a}}{\partial \mathbf{w}_a}\right)^\top \left(\frac{\partial r_k}{\partial \mathbf{u}_k} + \gamma \mathbf{B}_k^\top \hat{\boldsymbol{\lambda}}_{\mathbf{w}_c}(\mathbf{s}_{k+1})\right) \tag{33}$$

where η_a is the actor learning rate. The second parenthesis is the performance sensitivity to $\mathbf{u}_k$, obtained by projecting the future costate $\hat{\boldsymbol{\lambda}}(\mathbf{s}_{k+1})$ backward through $\mathbf{B}_k$.

C. Model Learning

Unknown dynamics in Eq. (28) and (29) are handled with online Recursive Least Squares (RLS):

$$\hat{\mathbf{s}}_{k+1} = \mathbf{W}_k^\top \boldsymbol{\zeta}_k, \tag{34}$$

where $\boldsymbol{\zeta}_k = [\mathbf{s}_k^\top, \mathbf{u}_k^\top, 1]^\top \in \mathbb{R}^{n+m+1}$ is the augmented regressor, and $\mathbf{W}_k \in \mathbb{R}^{(n+m+1)\times n}$ contains one coefficient column per state dimension.

At each time step, $\mathbf{W}_k$ is updated to minimize the one-step prediction error $\mathbf{e}_k = \mathbf{s}_{k+1} - \mathbf{W}_{k-1}^\top \boldsymbol{\zeta}_k$ using exponentially-weighted RLS:

$$\mathbf{K}_k = \frac{\mathbf{P}_{k-1}\boldsymbol{\zeta}_k}{\lambda_{\text{RLS}} + \boldsymbol{\zeta}_k^\top \mathbf{P}_{k-1}\boldsymbol{\zeta}_k} \tag{35}$$

$$\mathbf{W}_k = \mathbf{W}_{k-1} + \mathbf{K}_k \mathbf{e}_k^\top, \tag{36}$$

$$\mathbf{P}_k = \lambda_{\text{RLS}}^{-1}(\mathbf{I} - \mathbf{K}_k \boldsymbol{\zeta}_k^\top)\mathbf{P}_{k-1} \tag{37}$$

where $\lambda_{\text{RLS}} \in (0, 1]$ is the forgetting factor that exponentially down-weights older observations, and $\mathbf{P}_k$ is the inverse correlation matrix.

The Jacobians are extracted by partitioning $\mathbf{W}_k = [\mathbf{W}_k^{(s)} \mid \mathbf{W}_k^{(u)} \mid \mathbf{w}_k^{(b)}]$, where the blocks correspond to state, action, and bias terms:

$$\mathbf{A}_k \approx (\mathbf{W}_k^{(s)})^\top, \quad \mathbf{B}_k \approx (\mathbf{W}_k^{(u)})^\top. \tag{38}$$

Thus, actor and critic updates use the observed system response without analytical dynamics.

D. Reward Function Design

The evasion reward combines LOS-rate, separation, and control-penalty terms:

$$r(\mathbf{s}, \mathbf{u}) = \alpha_{\text{los}}\, r_{\text{los}}(\omega_{\text{los}}) + \alpha_{\text{dist}}\, r_{\text{dist}}(R) - \alpha_{psi}(\Delta\psi_{\text{cmd}})^2 \tag{39}$$

where $\alpha_{\text{los}} > 0$ and $\alpha_{\text{dist}} > 0$ weight LOS-rate generation and separation, $R = \|\mathbf{r}_{\text{rel}}\|$, $\omega_{\text{los}} = \|\mathbf{r}_{\text{rel}} \times \mathbf{v}_{\text{rel}}\|/R^2$, and $\alpha_{psi} > 0$ is the yaw-change penalty.

Rational saturation gives bounded rewards and smooth gradients:

$$r_{\text{los}}(\omega_{\text{los}}) = \frac{\omega_{\text{los}}}{\sigma_{\text{los}} + \omega_{\text{los}}}, \qquad r_{\text{dist}}(R) = \frac{R}{\sigma_{\text{dist}} + R}. \tag{40}$$

The scalar derivatives of the saturation functions are

$$\frac{\partial r_{\text{los}}}{\partial \omega_{\text{los}}} = \frac{\sigma_{\text{los}}}{(\sigma_{\text{los}} + \omega_{\text{los}})^2}, \qquad \frac{\partial r_{\text{dist}}}{\partial R} = \frac{\sigma_{\text{dist}}}{(\sigma_{\text{dist}} + R)^2} \tag{41}$$

The analytic gradients of the total reward w.r.t. states are

$$\begin{aligned}\nabla_{\mathbf{r}_{\text{rel}}} r = \alpha_{\text{dist}} &\frac{\partial r_{\text{dist}}}{\partial R}\frac{\mathbf{r}_{\text{rel}}}{R} \\ &+ \alpha_{\text{los}} \frac{\partial r_{\text{los}}}{\partial \omega_{\text{los}}}\left(\frac{\mathbf{v}_{\text{rel}} \times \mathbf{h}}{\|\mathbf{h}\| R^2} - \frac{2\omega_{\text{los}}\mathbf{r}_{\text{rel}}}{R^2}\right),\end{aligned} \tag{42}$$

$$\nabla_{\mathbf{v}_{\text{rel}}} r = \alpha_{\text{los}} \frac{\partial r_{\text{los}}}{\partial \omega_{\text{los}}} \frac{\mathbf{h} \times \mathbf{r}_{\text{rel}}}{\|\mathbf{h}\| R^2} \tag{43}$$

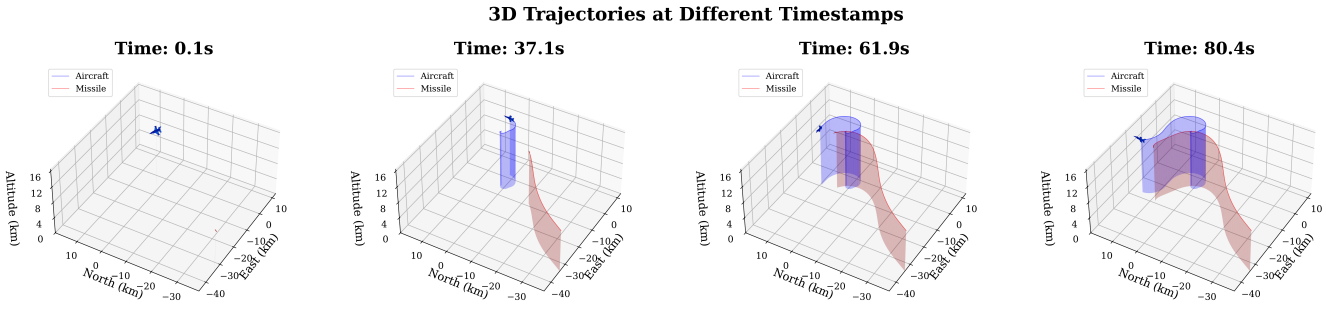

Fig. 1: 3D Trajectories of Missile and Aircraft for Engagement

where $\mathbf{h} = \mathbf{r}_{\text{rel}} \times \mathbf{v}_{\text{rel}}$. The state-gradient is

$$\frac{\partial r}{\partial \mathbf{s}} = \begin{bmatrix} \nabla_{\mathbf{r}_{\text{rel}}} r \\ \nabla_{\mathbf{v}_{\text{rel}}} r \end{bmatrix} \tag{44}$$

Similarly, $\frac{\partial r}{\partial \mathbf{u}} = [-2\alpha_\psi \Delta\psi_{\text{cmd}}, 0, 0]^\top$. These analytic gradients are used in Eq. (31) and (33).

V. Results

A. Training Details

The actor and critic are fully-connected neural networks. The critic approximates the costate with one 64-neuron hidden layer and hyperbolic tangent activation. The actor uses a 32-neuron hidden layer, with hyperbolic tangent activation at the hidden and output layers to enforce bounded commands. Network weights are initialized from a normal distribution with standard deviation $\sigma_w = 0.1$.

Training used 1000 episodes, critic and actor learning rates $\eta_c = 1 \times 10^{-4}$ and $\eta_a = 5 \times 10^{-4}$, discount factor $\gamma = 0.99$, and target-network soft update parameter $\tau = 10^{-3}$. Gradient clipping norms were 0.2 for the critic and 0.1 for the actor.

Exploration used additive Gaussian noise with initial standard deviation $\sigma_0 = 0.25$, exponential decay rate $\lambda = 0.995$ per episode, and minimum $\sigma_{\min} = 10^{-3}$. The online RLS model used forgetting factor $\lambda_{\text{RLS}} = 0.9995$ and initial covariance scaling of 1.0.

The reward weights were $\alpha_{\text{los}} = 0.015$, $\alpha_{\text{dist}} = 50$, and $\alpha_\psi = 0.013\ deg^{-2}$, with saturation parameters $\sigma_{\text{los}} = 12\ deg/s$ and $\sigma_{\text{dist}} = 35000\ meters$.

B. Illustrative Scenario

The proposed framework was evaluated in a BVR air-to-air engagement using the dynamical models in Sections II-B and II-C. The RL agent executes evasive maneuvers against an incoming threat; the initial conditions are summarized in Table I. The aircraft–missile trajectories are shown in Fig. 1, and the corresponding aircraft altitude, airspeed, and attitude trajectories are given in Fig. 2.

TABLE I: Initial Conditions for Simulation in Figs. 1 and 2

Parameter	Aircraft	Missile
Position [m]	(-4800, 1400, -14300)	(-33900, -38400, -10593)
Altitude [km]	14.28	10.59
Velocity [m/s]	(-179.3, 114.2, 0.0)	(194.3, 292.8, 0.0)
Airspeed [m/s]	212.68	351.41
Heading	147.50°	56.43°
Engagement Geometry		
Initial Range: 49.46 km — Bearing: 53.88° — Aspect: -2.55°		

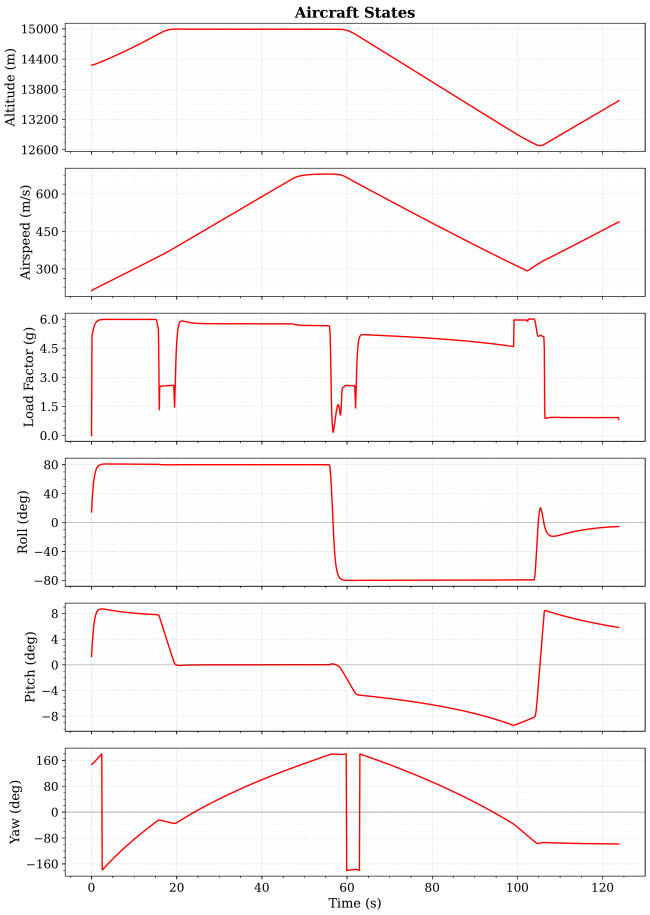

Fig. 2: Aircraft State history for Evasion in Fig. 1

C. Monte Carlo Analysis

Robustness was evaluated over 1,000 independent BVR engagements with randomized aircraft heading, speed, altitude, relative bearing, and initial separation (20–60 km). The missile used true proportional navigation with $N \in [2.5, 5]$ and maximum speed Mach 4. A significant fraction of cases

lie within the nominal 50–60 km no-escape zone, where evasion is unlikely without substantial guidance error.

Performance was assessed by *Survival Rate*, the percentage of episodes with minimum miss distance above the 20 m lethal radius ($d_{miss} \geq 20$ m), and *Average Miss Distance*. Baselines are: *Diving LOS-Rate*, which maximizes $\omega_{\text{los}} = \|\mathbf{r}_{\text{rel}} \times \mathbf{v}_{\text{rel}}\|/\|\mathbf{r}_{\text{rel}}\|^2$ subject to $a_{Az} < 0$; *Level LOS-Rate (Beam)*, which maximizes ω_{los} while enforcing $\mathbf{v}_A^\top \hat{\boldsymbol{\ell}} \approx 0$ and $a_{Az} \approx 0$; and *Turn-Away (Drag)*, which enforces $\mathbf{v}_A \parallel -\hat{\boldsymbol{\ell}}$ to maximize $\dot{R} = -\mathbf{v}_{\text{rel}}^\top \hat{\boldsymbol{\ell}}$.

Results are summarized in Table II. The diving and level LOS-rate maneuvers yield the lowest survival rates, showing that LOS angular rate alone is insufficient against a fast missile at long range and high closing speed. The turn-away maneuver improves miss distance by reducing closing velocity, but its low LOS-rate generation remains susceptible when the missile retains energy and guidance authority.

TABLE II: Monte Carlo Results

Agent	Survival Rate %	Avg Miss Distance
Diving LOS-Rate	8.7	429.52
Level LOS-Rate	12.3	1043.07
Turn-Away (Drag)	15.2	1647.19
Proposed	22.1	1782.40

The proposed DHP-based policy outperforms all baseline maneuvers in survival rate and average miss distance. Its advantage stems from adaptively trading LOS-rate generation and separation instead of pursuing a fixed geometric objective. The resulting survival rates and miss distances indicate that the policy can degrade interceptor effectiveness even within the nominal no-escape zone.

VI. Conclusion

This paper presented a novel model-based reinforcement learning framework for autonomous aircraft missile evasion in BVR missile engagements with unknown and time-varying threat dynamics. The evasion problem was formulated as a MDP and addressed using a Dual Heuristic Programming architecture that approximates the value gradient (costate) rather than the value function itself. By integrating online Recursive Least Squares identification of local system Jacobians and a bounded, differentiable reward function balancing separation and line-of-sight rate generation, the proposed method enables adaptive policy learning without prior knowledge of the missile guidance law. Simulation results demonstrate improved survivability and miss distance compared to representative fixed-structure evasion maneuvers across a wide range of engagement geometries.

Future work will focus on extending the proposed framework to more complex and realistic air combat scenarios. Planned directions include multi-threat engagements with coordinated missile salvos, learning-versus-learning settings where both evader and pursuer adapt online, and the incorporation of sensor uncertainty and communication delays.

References

[1] A. N. Costa, J. P. A. Dantas, E. Scukins, F. L. L. Medeiros, and P. Ögren, "Simulation and machine learning in beyond visual range air combat: A survey," *IEEE Access*, vol. 13, pp. 76 755–76 774, 2025.

[2] D. Bertsekas, *Dynamic programming and optimal control: Volume I*. Athena scientific, 2012, vol. 1.

[3] R. S. Sutton, A. G. Barto, and R. J. Williams, "Reinforcement learning is direct adaptive optimal control," *IEEE control systems magazine*, vol. 12, no. 2, pp. 19–22, 1992.

[4] F. L. Lewis and D. Vrabie, "Reinforcement learning and adaptive dynamic programming for feedback control," *IEEE circuits and systems magazine*, vol. 9, no. 3, pp. 32–50, 2009.

[5] D. A. White and D. A. Sofge, *Handbook of intelligent control: neural, fuzzy, and adaptive approaches*. Van Nostrand Reinhold Company, 1992.

[6] S. Ferrari and R. F. Stengel, "Online adaptive critic flight control," *Journal of Guidance, Control, and Dynamics*, vol. 27, no. 5, pp. 777–786, 2004.

[7] F. L. Lewis and K. G. Vamvoudakis, "Reinforcement learning for partially observable dynamic processes: Adaptive dynamic programming using measured output data," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 41, no. 1, pp. 14–25, 2010.

[8] B. Kiumarsi, F. L. Lewis, M.-B. Naghibi-Sistani, and A. Karimpour, "Optimal tracking control of unknown discrete-time linear systems using input-output measured data," *IEEE transactions on cybernetics*, vol. 45, no. 12, pp. 2770–2779, 2015.

[9] Y. Zhou, E.-J. v. Kampen, and Q. Chu, "Nonlinear adaptive flight control using incremental approximate dynamic programming and output feedback," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 2, pp. 493–496, 2016.

[10] A. Keshavarz and S. Boyd, "Quadratic approximate dynamic programming for input-affine systems," *International Journal of Robust and Nonlinear Control*, vol. 24, no. 3, pp. 432–449, 2014.

[11] J. Si, A. G. Barto, W. B. Powell, and D. Wunsch, "Model-based adaptive critic designs," 2004.

[12] E. Scukins and P. Ögren, "Using reinforcement learning to create control barrier functions for explicit risk mitigation in adversarial environments," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, 2021, pp. 10 734–10 740.

[13] E. Scukins, M. Klein, and P. Ögren, "Enhancing situation awareness in beyond visual range air combat with reinforcement learning-based decision support," in *2023 International Conference on Unmanned Aircraft Systems (ICUAS)*, 2023, pp. 56–62.

[14] Z. Yang, D. Zhou, H. Piao, K. Zhang, W. Kong, and Q. Pan, "Evasive maneuver strategy for ucav in beyond-visual-range air combat based on hierarchical multi-objective evolutionary algorithm," *IEEE Access*, vol. 8, pp. 46 605–46 623, 2020.

[15] D. Alkaher and A. Moshaiov, "Game-based safe aircraft navigation in the presence of energy-bleeding coasting missile," *Journal of Guidance, Control, and Dynamics*, vol. 39, no. 7, pp. 1539–1550, 2016.

[16] P. H. Zipfel, *Modeling and simulation of aerospace vehicle dynamics*. Aiaa, 2000.