doi: 10.82124/CEAS-GNC-2026-XYZ CEAS-GNC-2026-XYZ


# Reinforcement Learning for Obstacle-Aware Missile Guidance for Moving Targets in Constrained Flight Environments

**Burak Toy** *(Undergraduate Student)* Department of Aerospace Engineering, Middle East Technical University. burak.toy@metu.edu.tr

**Halil Ersin Söken** *(Associate Professor)* Department of Aerospace Engineering, Middle East Technical University. esoken@metu.edu.tr

*ABSTRACT*

**Classical missile guidance methods, such as proportional navigation (PNG) and Linear Quadratic Regulator (LQR) control, have shown good performance in structured settings but struggle when the environment is cluttered and uncertain. Reinforcement learning (RL) offers an alternative by allowing agents to learn obstacle-aware strategies directly from interaction rather than relying on predefined models. In this study, we design an RL-based guidance framework trained on randomly generated terrains where both missile and target start at varying positions, where the missile hits the moving target. The missile is modeled as a six-degree-of-freedom (6DOF) system with realistic actuation limits and full-state sensing. The learning architecture combines Soft Actor-Critic (SAC) for stable, sample-efficient training with RL² meta-learning to enable generalization across unseen environments. To improve reliability, a Model Predictive Control (MPC) layer supervises the learned policy and intervenes when unsafe actions are detected. This hybrid approach aims to achieve adaptive, robust, and safe guidance in complex flight conditions.**

**Keywords:** Reinforcement Learning, Missile Dynamic Model, Soft Actor-Critic, Meta Learning, Hybrid Control, Obstacle Avoidance

## 1 Introduction

Proportional navigation guidance (PNG) has proved to be a useful guidance technique in several surface-to-air and air-to-air missile systems for interception of airborne targets [1]. The fundamental idea behind PNG is that it issues a normal acceleration command to nullify the line-of-sight (LOS) rate, thereby forcing the interceptor to remain on the collision triangle [2]. There are different guidance methods, such as sliding mode guidance proposed by [3], which is robust to target maneuvers and missile's model uncertainties. [4] suggests using a Linear Quadratic Regulator (LQR) based controller, and approaches the missile target problem as a rendezvous problem for path planning, and solves the issue by defining an obstacle reference point and a minimum avoidance distance.

With the increasing complexity of application scenarios, however, real-world guidance problems in autonomous aerospace systems will be characterized by numerous practical constraints and highly time-varying, nonlinear dynamics [5]. Even though the defining reference points for obstacle avoidance method by [4] was highly successful, the high complexity of the real-world environment introduces the

need for a more robust obstacle avoidance system.

To address these challenges, reinforcement learning (RL) offers a data-driven approach that can generate adaptive guidance policies in environments characterized by nonlinearities and uncertainty. Unlike traditional methods that require explicit obstacle modeling, RL agents can learn obstacle-aware strategies directly through interactions with the environment. For instance, [6] propose RL-based missile guidance laws that autonomously avoid obstacles and terrains in complex environments without relying on predefined trajectories, for known and stable environments, however does not account for moving targets. Moreover, [7] optimize missile guidance controllers using RL to predict target acceleration and adjust to environmental uncertainties. These examples suggest that a mixed RL guidance solution can offer a robust missile that can understand and behave accordingly to its environment.

Thus, an RL agent is trained in order to hit the moving targets in obstacle-fitted flight environments. The training is made with SAC and $RL^2$ algorithms, with randomly generated environments. These combination of algorithms allows for the training of an agent that is highly adaptable to different environments. The missile is modeled with 6DOF, with constrains applied.

# 2 Proposed Approach

## 2.1 Mapping and Modeling

This paper proposes training an RL agent that can learn and implement the optimal policy across different environments. [6] trained an agent that can learn the optimal policy for a specific pre-defined map by rewarding in that specific environment. However, it is important to apply this concept in different mapping scenarios, since missiles will experience different environments. Thus, in each run, the agent to be trained should be faced with different environments, such as those given in Figure 1.

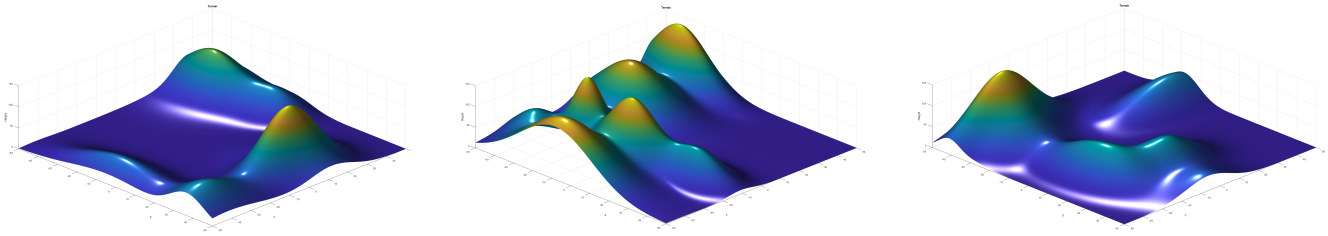

**Fig. 1 Random Terrains**

The agent and the target are spawned at random points, with the agent initialized with a velocity towards the target. If an object exists near the agent and towards the target, the initial velocity should not be perfectly aligned to the target, but should be given with respect to the gradients of the blocking object. During training, the agent has access to all of the environment data, along with the target location.

The agent is modeled as a 3D beam, and is allowed to perform actions in six degrees of freedom (6DOF) with limiting force and moment values. Its state is modeled with quaternions, angular velocity, position and velocity.

## 2.2 Moving Target Definitions

The target object is designed to follow a pre-defined path, calculated each time a new map is randomized. In order to prevent non-realistic paths such as traveling inside mountains, the environment creates a path generated with the $A^*$ algorithm. This allows the target to climb up and down the objects.

## 2.3 Reinforcement Learning Policy

For the RL model, a Soft Actor-Critic Method (SAC) is established, with the use of memory based Meta-Learning algorithms. The SAC algorithm by [8] is an off-policy method, and it has been shown to perform successfully in continuous state spaces. This method also establishes the maximum entropy formulation, which has been proven to introduce exploration and robustness to the agent [9]. With SAC algorithm, a sample efficient and stabile training can be achieved, where the agent is not greedy, and exploration does not cause instability.

Following that, the memory based meta-learning algorithm $RL^2$ by [10] is established in the training. $RL^2$ algorithm uses the combination of Recurrent Neural Network (RNN) method with RL, which proposes that RNN based memory layers with access to rewards, observations, and actions can be trained to adapt to new environments with ease. Their method is expected to be highly necessary for the task explained in this paper, since the RL model is required to guide the missile in very different environments.

The used SAC method is a model-free method, which means that it will rely solely on reward signals from the environment without modeling any dynamics. Thus, Model Predictive Control (MPC), which is a model-based method is used in order to predict the results of the actions of the model-free agent. This creates a safety shield, if safety penalties are enforced to actions that are deemed dangerous by MPC. By design, the safety guide intervenes minimally and modifies the base policy's proposed action distribution only if it inevitably leads towards an unsafe region of the state space [11].

## 2.4 Missile Modeling

For the environment simulation, the missile dynamics are integrated with 6DOF, using quaternions. Fundamental equations can be proven as follows.

### *2.4.1 Translational Dynamics*

The translation dynamics equation for the center of gravity point in body frame is given in Equation 1, and when $A_{BV}$ matrix is used, same equation can be written as Equation 2.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + A_{BV} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = m \left( \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \right) \tag{1}$$

In this equation above; X, Y and Z stands for aero-propulsive forces in body x, y, and z axes, whereas the variables u, v, and w are velocities in the same axes. The variables p, q and r are angular velocities around those axes. Since gravity applies in only NED z axis, this results in Equation 2.

$$\begin{aligned} X + mgA_{BV}(1,3) &= m\dot{u} + m\,(qw - rv) \\ Y + mgA_{BV}(2,3) &= m\dot{v} + m\,(ru - pw) \\ Z + mgA_{BV}(3,3) &= m\dot{w} + m\,(pv - qu) \end{aligned} \tag{2}$$

Re-arranging them, the equations for translational rates in Equation 3 can be found.

$$\begin{aligned}
\dot{u} &= \frac{1}{m}\left(X + mgA_{BV}(1,3)\right) - (qw - rv) \\
\dot{v} &= \frac{1}{m}\left(Y + mgA_{BV}(2,3)\right) - (ru - pw) \\
\dot{w} &= \frac{1}{m}\left(Z + mgA_{BV}(3,3)\right) - (pv - qu)
\end{aligned} \tag{3}$$

### *2.4.2 Rotational Dynamics*

The rotational dynamics equation from NED frame to body frame can be constructed as follows.

$$\sum \vec{G}_B = \left.\frac{d\vec{H}_B}{dt}\right|_{\text{Inertial Frame}} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} \tag{4}$$

Equation 4 suggest that the angular moments around body x, y and z axes, which are L, M, and N; are equal to the time rate of angular momentum in inertial frame. Note that the angular momentum in body frame can also be defined as in Equation5, where $I_B$ is the inertia matrix in body frame, and $\vec{w}_B^I$ is the angular velocity vector of body frame dissolved in inertial frame.

$$\vec{H}_B = I_B \vec{w}_B^I \tag{5}$$

Using Equation 5, the time rate of angular momentum in inertial frame can be defined as in the Equation 6.

$$\left.\frac{d\vec{H}_B}{dt}\right|_{\text{Inertial Frame}} = \dot{\vec{H}}_B + \vec{w}_B^I \times \vec{H}_B$$

$$\begin{aligned}
\text{Using} \longrightarrow \dot{\vec{H}}_B &= \dot{I}_B \vec{w}_B^I + I_B \dot{\vec{w}}_B^I \\
\text{Since} \longrightarrow \dot{I}_B &= 0
\end{aligned} \tag{6}$$

$$\left.\frac{d\vec{H}_B}{dt}\right|_{\text{Inertial Frame}} = I_B \dot{\vec{w}}_B^I + \vec{w}_B^I \times \vec{H}_B$$

Then, using proper definitions, and assuming $I_{yz} = I_{xy} = 0$ the rotational dynamic equations can be constructed as in the Equation 7.

$$\begin{aligned}
L &= I_{xx}\dot{p} - I_{xz}\left(\dot{r} + pq\right) - \left(I_{yy} - I_{zz}\right) qr \\
M &= I_{yy}\dot{q} - I_{xz}\left(r^2 - p^2\right) - \left(I_{zz} - I_{xx}\right) rp \\
N &= I_{zz}\dot{r} - I_{xz}\left(\dot{p} - qr\right) - \left(I_{xx} - I_{yy}\right) pq
\end{aligned} \tag{7}$$

Re-arranging them, the equations for rotational rates in Equation 8 can be found.

$$\dot{p} = \frac{1}{I_{xx}I_{zz} - I_{xz}^2}\left(I_{zz}L - I_{xz}N\right)$$

$$\dot{q} = \frac{1}{I_{yy}}\left(M\right) \tag{8}$$

$$\dot{r} = \frac{1}{I_{xx}I_{zz} - I_{xz}^2}\left(I_{xx}N - I_{xz}L\right)$$

### *2.4.3 Quaternion Rates*

As given in Equation 9, the quaternion rate can also be expressed as in the Equation 10. Note that quaternions are given in JPL convention. Euler angles are only used for display purposes, and will not be established in dynamics, nor be given to RL agent.

$$\dot{q} = \lim_{\Delta t \to 0} \frac{\begin{bmatrix} \frac{\Delta v}{2} \\ 0 \end{bmatrix} \otimes q(t)}{\Delta t} = \frac{1}{2}\lim_{\Delta t \to 0}\frac{\Delta v}{\Delta t} \otimes q(t) = \frac{1}{2} w \otimes q(t) \tag{9}$$

$$\begin{aligned}
\dot{q}_1 &= \frac{1}{2}\left(pq_4 + rq_2 - qq_3\right) \\
\dot{q}_2 &= \frac{1}{2}\left(qq_4 + pq_3 - rq_1\right) \\
\dot{q}_3 &= \frac{1}{2}\left(rq_4 + qq_1 - pq_2\right) \\
\dot{q}_4 &= \frac{1}{2}\left(-pq_1 - qq_2 - rq_3\right)
\end{aligned} \tag{10}$$

# References


[1] Stephen A. Murtaugh and Harry E. Criel. Fundamentals of proportional navigation. *IEEE Spectrum*, 3(12):75–85, 1966. doi: 10.1109/MSPEC.1966.5217080.

[2] Shaoming He and Chang-Hun Lee. Optimality of error dynamics in missile guidance problems. *Journal of Guidance Control and Dynamics*, 41(7):1624–1633, Feb. 2018. doi: 10.2514/1.g003343.

[3] Yuri B Shtessel and Christian H Tournes. Integrated higher-order sliding mode guidance and autopilot for dual control missiles. *Journal of guidance, control, and dynamics*, 32(1):79–94, 2009.

[4] Martin Weiss and Tal Shima. Linear quadratic optimal control-based missile guidance law with obstacle avoidance. *IEEE Transactions on Aerospace and Electronic Systems*, 55(1):205–214, 2019. doi: 10.1109/TAES.2018.2849901.

[5] Shaoming He, Hyo-Sang Shin, and Antonios Tsourdos. Computational missile guidance: A deep reinforcement learning approach. *Journal of Aerospace Information Systems*, 18(8):571–582, 2021.

[6] Daseon Hong and Sungsu Park. Avoiding obstacles via missile real-time inference by reinforcement learning. *Applied Sciences*, 12(9), 2022. ISSN: 2076-3417. doi: 10.3390/app12094142.

[7] Weifan Li, Yuanheng Zhu, and Dongbin Zhao. Missile guidance with assisted deep reinforcement learning for head-on interception of maneuvering target. *Complex & Intelligent Systems*, 8, 11 2021. doi: 10.1007/s40747-021-00577-6.

[8] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1861–1870. PMLR, Jul 2018.

[9] Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. In *Proceedings of the 23rd National Conference on Artificial Intelligence - Volume 3*, AAAI'08, page 1433–1438. AAAI Press, 2008. ISBN: 9781577353683.

[10] Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and Pieter Abbeel. $Rl^2$: Fast reinforcement learning via slow reinforcement learning. *arXiv (Cornell University)*, Jan. 2016. doi: 10.48550/arxiv.1611.02779.

[11] Samuel Pfrommer, Tanmay Gautam, Alec Zhou, and Somayeh Sojoudi. Safe reinforcement learning with chance-constrained model predictive control. In Roya Firoozi, Negar Mehr, Esen Yel, Rika Antonova, Jeannette Bohg, Mac Schwager, and Mykel Kochenderfer, editors, *Proceedings of The 4th Annual Learning for Dynamics and Control Conference*, volume 168 of *Proceedings of Machine Learning Research*, pages 291–303. PMLR, Jun 2022.