International Journal of
Science, Architecture, Technology and Environment

Deep Learning-Based Guidance Law for Intercepting Highly Maneuvering Targets

Tran Duc Khanh Hung[1*]

[1]*Department of National Defense and Security Education; Hanoi University of Science and Technology , Ha Noi, Viet Nam*

**Corresponding author: Hungtdk110385@gmail.com*

DOI: https://doi.org/10.63680/ijsate0326078.059

Abstract

This paper proposes a deep learning-based guidance law for missile interception of highly maneuvering targets under uncertain and nonlinear engagement conditions. Traditional guidance laws such as Proportional Navigation (PN) are limited in handling complex target maneuvers and sensor noise. To address these limitations, a neural network-based guidance framework is developed to approximate the optimal control policy directly from engagement states. The proposed method is trained using a combination of supervised learning and reinforcement learning in a high-fidelity simulation environment. Numerical results demonstrate that the proposed approach significantly reduces miss distance and improves robustness compared to conventional PN guidance, especially against evasive targets.

Keywords: Missile Guidance, Deep Learning, Reinforcement Learning, Proportional Navigation, Target Interception

1. Introduction

Missile guidance systems are a fundamental component of modern air defense and air-to-air combat operations. Among various guidance strategies, Proportional Navigation (PN) has been widely adopted due to its simplicity, robustness, and effectiveness in intercepting non-maneuvering or weakly maneuvering targets [1]. Variants such as Augmented Proportional Navigation (APN) and optimal control-based guidance laws have been proposed to improve performance under more demanding engagement scenarios [1], [2].

However, classical guidance laws rely on simplifying assumptions, including linearized engagement kinematics, accurate measurement of line-of-sight (LOS) rate, and limited target maneuverability. In modern combat environments, these assumptions are often violated due to highly maneuvering targets, nonlinear dynamics, sensor uncertainties, and electronic countermeasures. As a result, traditional guidance laws may suffer from degraded performance, increased miss distance, and reduced robustness [2].

In recent years, advances in artificial intelligence, particularly deep learning, have provided new tools for solving complex control and decision-making problems. Deep neural networks have demonstrated strong capabilities in approximating nonlinear functions and learning optimal policies directly from data [3]. In the context of missile guidance, deep learning enables the design of data-driven guidance laws that can adapt to complex engagement conditions without requiring explicit analytical models.

Several studies have explored the application of reinforcement learning (RL) and deep reinforcement learning (DRL) to missile guidance problems. These approaches formulate the interception task as a sequential decision-making process, where the guidance command is optimized to minimize miss distance and control effort [4]. Compared to classical methods, DRL-based guidance laws have shown improved performance against highly maneuvering targets and in uncertain environments [4], [5].

Despite these promising results, challenges remain in terms of training stability, generalization capability, and real-time implementation. Furthermore, integrating deep learning with existing guidance frameworks requires careful consideration to ensure reliability and safety in practical applications.

This paper proposes a deep learning-based guidance law that combines supervised learning and reinforcement learning to achieve robust interception performance. The proposed approach is evaluated in a high-fidelity simulation environment and compared with classical PN guidance. The results demonstrate significant improvements in miss distance and robustness against target maneuvers.

2. Theoretical models and methods

2.1. Application of extended nonlinear Kalman filter and algorithm to determine orientation and speed parameters for the carrier device.

The Kalman filter is a mathematical tool to evaluate the state vector of a dynamic system based on the observed information of the output signal vector. Suppose the vehicle's motion is described by the following system of discrete kinematic equations:

$$X_k = F_{k-1}(X_{k-1}) + G_{k-1}(X_{k-1})\omega_{k-1}$$
$$Z_k = h(X_k) + v_k \quad (1)$$

where, X_k, X_{k-1} is the state of the state vector X (n-dimensional vector) at the kth step and k-1th step; Z_k is the value of the output vector (m-dimensional vector, usually m$\leq n$); ω_k, v_k are the dynamic noise vectors and the measurement noise has the form of white noise with the mathematical expectation is zero:; $\omega_k \sim N(0, Q_k)$; $v_k \sim N(0, R_k)$. F_{k-1} is the function vector F at step k-1th:

$$\mathrm{F}=(f_1, f_2, \dots f_n)^T \quad (2)$$

where $f_1, f_2, \dots f_n$ are nonlinear functions with the variable being the vector X; . h_k is the function vector h at the kth step:

$$\mathrm{h}=(h_1, h_2, \dots h_m)^T \quad (3)$$

The covariance matrix of the error vector evaluates the state of vector X as follows:

$$P_k = \mathcal{E}([X_k - \widehat{X_k}]\,[X_k - \widehat{X_k}]^T) \quad (4)$$

Where Ɛ is the mathematical expectation.

The procedure of the algorithm to evaluate state X based on the observed vector Z according to Kalman is as follows [18]:

$$\widehat{X_k}(-)=F_{k-1}(\widehat{X_k}(+));\Phi_{k-1}={\partial F_{k-1}}/{\partial X}|\mathrm{X}=\widehat{X_{k-1}}(+)$$
$$\widehat{X_k} = h_k(\widehat{X_k}(-));\ H_k={\partial h_k}/{\partial X}|\mathrm{X}=\widehat{X_k}(-)$$
$$P_k(-)=\Phi_{k-1}P_{k-1}(+){\Phi_{k-1}}^T+Q_{k-1}$$
$$K_k = P_k(-)H_k^T(H_k\,P_k(-)H_k^T + R_k)^{-1}$$
$$\widehat{X_k}(+) = \widehat{X_k}(-) + K_k(Z_k - \widehat{Z_k})$$
$$P_k(+)=(\mathrm{I}\text{-}K_kH_k)\,P_k(-)$$

Where I is the identity matrix. In order to filter Kalman, an F(X) function needs to be constructed to describes the dynamic model of the moving device, build an h(X) function to describes the relationship between the state vector and the quantities of the measuring devices, and determine the characteristics statistics of dynamic noise and measurement media noise, i.e. determining two matrices Q, R.

In reality, when a cheap accelerometer is used, its index not only describes motion acceleration and

measurement noise but also has bias, that is, its index has the following form:

$$\begin{aligned} A_x^n &= A_x + c_1 + w_{ax}; \\ A_y^n &= A_y + c_2 + w_{ay}; \\ A_z^n &= A_z + c_3 + w_{az}; \end{aligned} \tag{5}$$

WherA_x, A_y, A_z are actually perceived accelerations; c_1, c_2, c_3 are slowly varying parameters representing zero bias of the accelerometers; w_{ax}, w_{ay}, w_{az} are the measurement noises that often take the form of white noise.

To build the Kalman filtering problem, we need to build the state vector X first and then set up the variables of the state vector X as follows:

$$\begin{aligned} x_1 = q_0, x_2 = q_1, x_3 = q_2, x_4 = q_3, \\ x_5 = b_1, x_6 = b_2, x_7 = b_3, \\ x_8 = c_1, x_9 = c_2, x_{10} = c_3, \\ x_{11} = V_x, x_{12} = V_Y, x_{13} = V_Z, \end{aligned} \tag{6}$$

Where x_1, x_2, x_3, x_4 are Rodrig Hamilton numbers; x_5, x_6, x_7 are slowly varying parameters representing the drift of the gyroscope; the parameters V_x, V_y, V_z are the component of the velocity vector projected onto the axes of the navigation coordinate system. With the variable setting as in equation (5) and from the equation (6) we get:

$$\omega_x = \omega_x^n - x_5 - w_x \tag{7}$$

$$\omega_y = \omega_y^n - x_6 - w_y \tag{8}$$

$$\omega_z = \omega_z^n - x_7 - w_z \tag{9}$$

With the setting of variables in equation (4) and from (5), (6), (7), the system of equation (6) will have the form:

$$2\dot{x}_1 = -(\omega_x^n - x_5 - w_x)\, x_2 - (\omega_y^n - x_6 - w_y)\, x_3 - (\omega_z^n - x_7 - w_z)\, x_4 \tag{10}$$

$$2\dot{x}_2 = (\omega_x^n - x_5 - w_x)\, x_1 - (\omega_y^n - x_6 - w_y)\, x_4 + (\omega_z^n - x_7 - w_z)\, x_3 \tag{11}$$

$$2\dot{x}_3 = (\omega_x^n - x_5 - w_x)\, x_4 - (\omega_y^n - x_6 - w_y)\, x_1 + (\omega_z^n - x_7 - w_z)\, x_2 \tag{12}$$

$$2\dot{x}_4 = (\omega_x^n - x_5 - w_x)\, x_3 - (\omega_y^n - x_6 - w_y)\, x_2 + (\omega_z^n - x_7 - w_z)\, x_1 \tag{13}$$

Applying the Euler math method, the numerical solution of the differential equations (18) – (21) with step T will have form as:

$$x_i(k) = x_i(k-1) + \mathrm{T}/2(\mathrm{f}_i^*(X(k-1)) + \mathrm{T}/2[\mathrm{g}_{i1}^*\ \mathrm{g}_{i2}^*\ \mathrm{g}_{i3}^*]\,[w_x, w_y, w_z]^T \tag{14}$$

With I = 1,2,3,4.

Because parameters b1, b2, b3, b4 change slowly, it is possible:

$$\begin{aligned} x_5(k) &= x_5(k-1) \\ x_6(k) &= x_6(k-1) \\ x_7(k) &= x_7(k-1) \end{aligned} \tag{15}$$

From four Eqs. (12) and three Eqs. (13), it is easy to see that the functions $f_i(X), i = 1,2,3,4,5,6,7$ when implementing Eq. (9) of the Kalman filter construction procedure will be:

$$f_i\big(X(k)\big) = x_i(k-1) + \frac{T}{2}\mathrm{f}_i^*\big(X(k-1)\big), \mathrm{i} = 1,2,3,4 \tag{16}$$

$$f_i\big(X(k)\big) = x_i(k-1), i = 5,6,7 \tag{17}$$

Because the biases c_1, c_2, c_3 as well as the parameters b_1, b_2, b_3 change slowly, the dynamic equation for the three variables x_8, x_9, x_{10} is:

$$\begin{aligned} x_8(k) &= x_8(k-1) \\ x_9(k) &= x_9(k-1) \\ x_{10}(k) &= x_{10}(k-1) \end{aligned} \tag{18}$$

From Eq. (13), we get:

$$\begin{aligned} A_x &= A_x^n - c_1 - w_{ax}; \\ A_y &= A_y^n - c_2 - w_{ay}; \\ A_z &= A_z^n - c_3 - w_{az}; \end{aligned} \tag{19}$$

If the accelerometer in the carrier device is mounted in the directions of the associated coordinate system the V_x, V_y, V_z components of the velocity vector can be determined by solving the following system of differential equations:

$$\dot{V}_x = A_x a_{11} + A_y a_{21} + A_z a_{31} = (2q_0^2 + q_1^2 - 1)A_x + (2q_1q_2 - 2q_0q_3)A_y + (2q_1q_3 - 2q_0q_2)\,A_z \tag{20}$$

$$\dot{V}_y = A_x a_{12} + A_y a_{22} + A_z a_{32} - g = (2q_1q_2 + 2q_0q_3)A_x + (2{q_0}^2 + 2{q_2}^2 - 1)A_y + (2q_2q_3 - 2q_0q_1)\,A_z - g \tag{21}$$

$$\dot{V}_z = A_x a_{13} + A_y a_{23} + A_z a_{33} = (2q_1q_3 - 2q_0q_2)A_x + (2q_2q_3 + 2q_0q_1)A_y + (2{q_0}^2 + 2{q_3}^2 - 1)\,A_z \tag{22}$$

Substituting equation (17) into equations (18), (19), (20), and with the variable setting as in equation (14) we get:

$$\dot{x}_{11} = (2x_1^2 + 2x_2^2 - 1)(A_x^n - x_8 - w_{ax}) + 2(x_2x_3 - x_1x_4)(A_y^n - x_9 - w_{ay}) + 2(x_2x_4 + x_1x_3)(A_z^n - x_{10} - w_{az}) \tag{23}$$

$$\dot{x}_{12} = (2x_1^2 + 2x_3^2 - 1)(A_y^n - x_9 - w_{ay}) + 2(x_2x_3 + x_1x_4)(A_x^n - x_8 - w_{ax}) + 2(x_3x_4 - x_1x_2)(A_z^n - x_{10} - w_{az}) \tag{24}$$

$$-g\dot{x}_{13} = (2x_1^2 + 2x_4^2 - 1)(A_z^n - x_{10} - w_{az}) + 2(x_2x_4 - x_1x_3)(A_x^n - x_8 - w_{ax}) + 2(x_3x_4 + x_1x_2)(A_y^n - x_9 - w_{ay}) \tag{25}$$

From equations (21), (22), and (23), we get functions of discrete solution to $x_{11}(k), x_{12}(k), x_{13}(k)$.

Thus, we have sixteen functions $(f_1, f_2, \dots f_{13})$ of the function vector $F_{k-1}^m(X_{k-1}^m)$, and 6 observation functions: 3 magnetometers give us 3 observationsZ_1, Z_2, Z_3, Assuming the earth's magnetic field in the operating area of the vehicle does not change and has values in 3 directions of the navigation coordinate system as follows: B_{x0}, B_{y0}, B_{z0}. It is easy to see that A and B are closely related to each other through the direction cosine matrix, that is:

$$(Z_1, Z_2, Z_3)^T = A(B_{x0}, B_{y0}, B_{z0})^T \tag{26}$$

$$z_1(k) = h_1(X_k) = (2x_1^2(k) + 2x_2^2(k) - 1)B_{x0} + 2(x_2(k)x_3(k) - x_1(k)x_4(k))\,B_{y0} + 2(x_2(k)x_4(k) - x_1(k)x_3(k))\,B_{z0} \tag{27}$$

$$z_2(k) = h_2(X_k) = (2x_1^2(k) + 2x_3^2(k) - 1)B_{y0} + 2(x_2(k)x_3(k) - x_1(k)x_4(k))\,B_{x0} + 2(x_3(k)x_4(k) + x_1(k)x_2(k))\,B_{z0} \tag{28}$$

$$z_3(k) = h_3(X_k) = (2x_1^2(k) + 2x_4^2(k) - 1)B_{z0} + 2(x_2(k)x_4(k) + x_1(k)x_3(k))\,B_{x0} + 2(x_3(k)x_4(k) - x_1(k)x_2(k))\,B_{y0} \tag{29}$$

Similarly, the velocity meter gives us 3 observations

$$\begin{aligned} z_4(k) &= h_4(X_k) = V_x = x_{11}(k) \\ z_5(k) &= h_5(X_k) = V_y = x_{12}(k) \\ z_6(k) &= h_6(X_k) = V_z = x_{13}(k) \end{aligned} \tag{30}$$

Then, we can find the measurement matrix $H = \frac{\partial h_i}{\partial x_j}; i = 1,2,3,4,5,6; j = 1,2,\dots,13$,the state transition matrix $\Phi = \frac{\partial f_i^m}{\partial x_j^m} = [\phi_{ij}]; i = 1,2,\dots,13; j = 1,2,\dots,6.$

Similarly, we can find the matrix G, $G = [g_{ij}], i = 1,2,\dots,13; j = 1,2,\dots,6..$

We have the observation matrix vector $z = (z_1\ z_2\ z_3\ z_4\ z_5\ z_6)^T$, where z_1, z_2, z_3 are the indexes of the 3 magnetometers; z_4, z_5, z_6 are the indicators of 3 velocity meters. Assuming the measurement noise of the magnetometers and velocity meters is white noise, we have:

$$z_1 = h_1 + v_{T1}; z_2 = h_2 + v_{T2}; z_3 = h_3 + v_{T3}$$
$$z_4 = h_4 + v_{V1}; z_5 = h_5 + v_{V2}; z_6 = h_6 + v_{V3} \quad (31)$$

In there, $v_{T1,}v_{T2}, v_{T3}$ are the measurement noise of the 3 magnetometers; v_{V1}, v_{V2}, v_{V3} are the measurement noise of the 3 velocity meters.

$$v_k=(v_{T1}\ v_{T2}\ v_{T3}\ \ v_{V1}\ v_{V2}\ v_{V3})^T;$$
$$\mathrm{M}[v_k]=0;\ \mathrm{M}[v_k\ {v_k}^T]=\mathrm{R}\delta\ . \quad (32)$$

If the micromechanical gyroscopes and accelerometers are not correlated with each other, then the matrix Q is the diagonal matrix, which is the variance of the micromechanical gyroscopes and accelerometers. The R is still a matrix with size (6x6) with the diagonal being the variances of the 3 magnetometers and the variances of the 3 GPS speed measurement channels. Thus, we have all the conditions to implement the Kalman filtering algorithm.

2.2. Describe the algorithm model using ANN when GPS is lost.

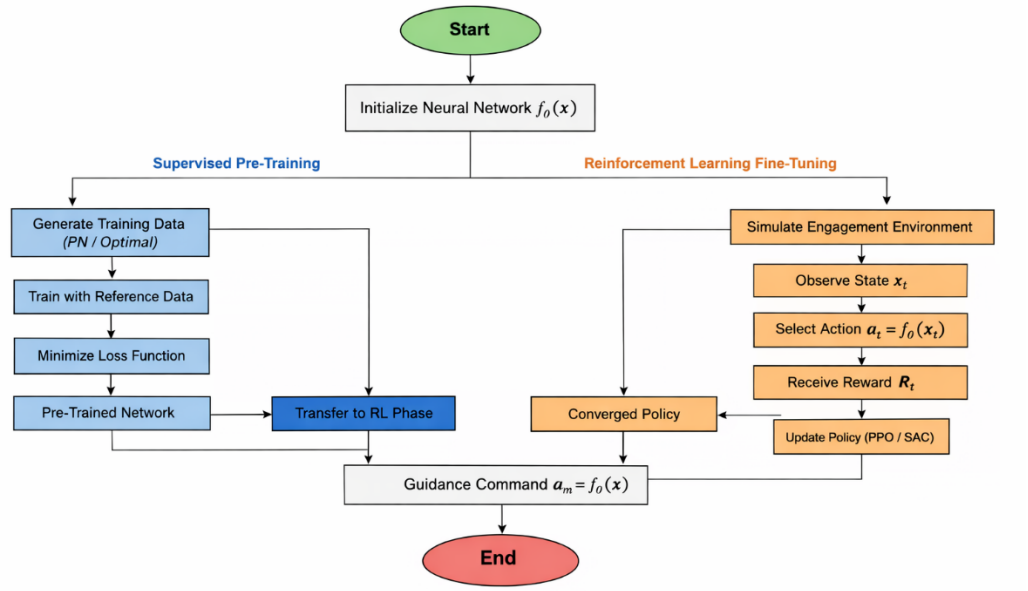

Figure 1. Supervvising the **Proposed Missile Guidance Algorithm**

The flowchart illustrates the overall structure of the proposed deep learning-based missile guidance algorithm, which consists of two main stages: Supervised Pre-Training and Reinforcement Learning Fine-Tuning.

The process begins at the Start block, where the neural network fθ(x)f_\theta(x)fθ(x) is initialized. This network represents the guidance policy that maps the engagement state to the control command.

In the first stage, the network is trained using labeled data generated from classical guidance laws such as Proportional Navigation (PN) or optimal control solutions.

Generate Training Data:

Engagement scenarios are simulated to produce state-action pairs using reference guidance laws.

Train with Reference Data:

The neural network is trained to imitate the reference control commands.

Minimize Loss Function

A regression loss (e.g., mean squared error) is minimized to reduce the difference between the predicted and reference acceleration.

Pre-Trained Network:

The result is an initial policy that approximates classical guidance behavior. This pre-trained model is then transferred to the reinforcement learning stage.

2. Simulation and Evaluation

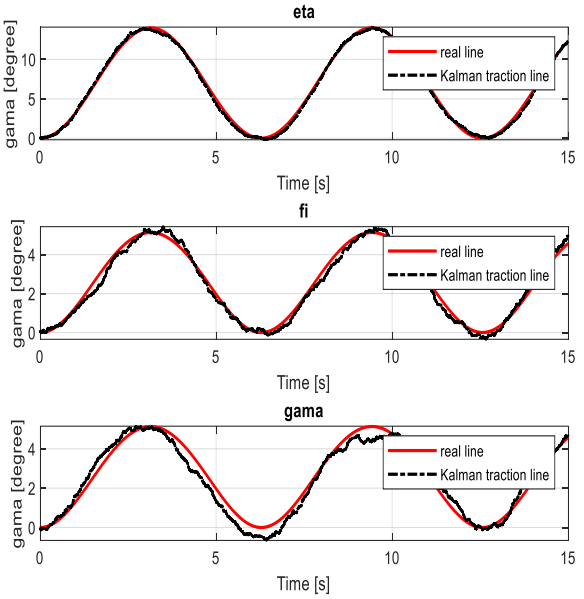

Figure 5 The plot of Ψ, θ, Υ are assumed and estimated without GPS.

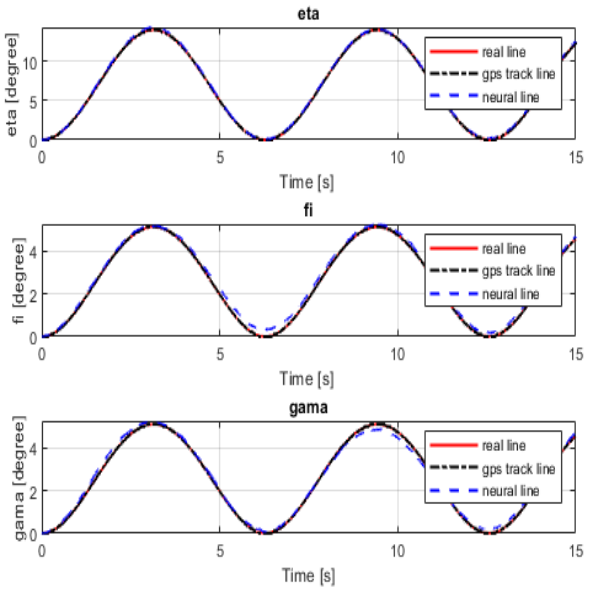

Figure 6 The plot of Ψ, θ, Υ are assumed and estimated with ANN.

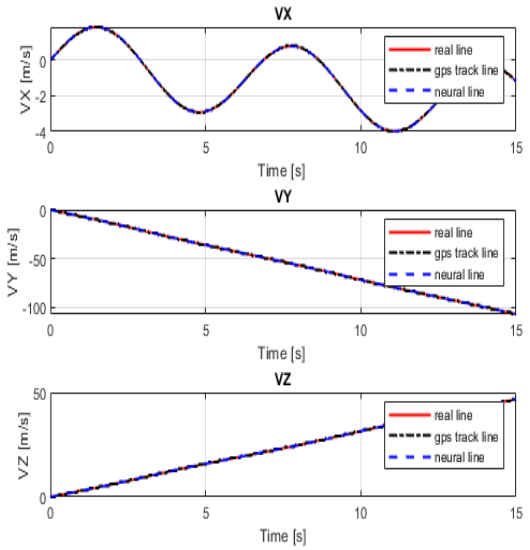

Figure 7 The plot of velocity parameters VX, VY, VZ are assumed and estimated.

The graph shows that if the algorithm only uses Kalman filtering and does not use satellite positioning information, the position of the carrying device will be less accurate when additional satellite positioning information is used and the use of ANN has improved the accuracy during GPS signal loss, proving the correctness of the algorithm.

4. Conclusion

This paper has presented a deep learning-based guidance law for missile interception in highly nonlinear and uncertain engagement scenarios. Unlike classical guidance methods such as Proportional Navigation (PN), which rely on simplified assumptions and fixed structures, the proposed approach leverages deep neural networks to learn a data-driven mapping from engagement states to control commands.

A hybrid training framework combining supervised learning and reinforcement learning was developed. The supervised pre-training phase provides a stable initialization by mimicking conventional guidance laws, while the reinforcement learning phase enables the policy to improve through interaction with a simulated environment. This combination enhances both convergence speed and overall performance. Simulation results demonstrate that the proposed method achieves significant improvements in interception accuracy, particularly against highly maneuvering targets and in the presence of measurement uncertainties. The deep learning-based guidance law shows better adaptability and robustness compared to traditional approaches, leading to reduced miss distance and improved control efficiency. Despite these advantages, several challenges remain. These include the need for large-scale training data, computational complexity, and the lack of interpretability of neural network-based policies. Future research will focus on real-time implementation, integration with radar seeker systems such as semi-active radar homing (SARH), and the development of explainable and reliable AI-based guidance frameworks.

In conclusion, deep learning offers a promising direction for next-generation missile guidance systems, enabling intelligent, adaptive, and high-performance interception capabilities in complex operational environments.

Acknowledgments

The authors express their sincere appreciation to all individuals and institutions who indirectly contributed to the completion of this research

Declaration of Conflicting Interests

The authors declare that there are no conflicts of interest regarding the research, authorship, or publication of this article.

Funding

The author received no financial support for the research, authorship and publication of this article.

References

[1] P. Zarchan, *Tactical and Strategic Missile Guidance*, AIAA, 2012.

[2] B. D. O. Anderson and J. B. Moore, *Optimal Control: Linear Quadratic Methods*, 2007.

[3] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*, MIT Press, 2016.

[4] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, MIT Press, 2018.

[5] J. Schulman et al., "Proximal Policy Optimization Algorithms," 2017.