mathematics

MDPI

Article

Deep Reinforcement Learning-Based Impact Angle-Constrained Adaptive Guidance Law

Zhe Hu [1,*], Wenjun Yi [1], Liang Xiao [2]

[1] National Key Laboratory of Transient Physics, Nanjing University of Science and Technology, Nanjing 210094, China; wenjunyi@njust.edu.cn
[2] School of Computer Science and Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; xiaoliang@mail.njust.edu.cn
* Correspondence: hz@njust.edu.cn

Abstract: This study presents an advanced second-order sliding-mode guidance law with a terminal impact angle constraint, which ingeniously combines reinforcement learning algorithms with the nonsingular terminal sliding-mode control (NTSM) theory. This hybrid approach effectively mitigates the inherent chattering issue commonly associated with sliding-mode control while maintaining high levels of control system precision. We introduce a parameter to the super-twisting algorithm and subsequently improve an intelligent parameter-adaptive algorithm grounded in the Twin-Delayed Deep Deterministic Policy Gradient (TD3) framework. During the guidance phase, a pre-trained reinforcement learning model is employed to directly map the missile's state variables to the optimal adaptive parameters, thereby significantly enhancing the guidance performance. Additionally, a generalized super-twisting extended state observer (GSTESO) is introduced for estimating and compensating the lumped uncertainty within the missile guidance system. This method obviates the necessity for prior information about the target's maneuvers, enabling the proposed guidance law to intercept maneuvering targets with unknown acceleration. The finite-time stability of the closed-loop guidance system is confirmed using the Lyapunov stability criterion. Simulations demonstrate that our proposed guidance law not only meets a wide range of impact angle constraints but also attains higher interception accuracy and faster convergence rate and better overall performance compared to traditional NTSM and the super-twisting NTSM (ST-NTSM) guidance laws, The interception accuracy is less than 0.1 m, and the impact angle error is less than 0.01°.

Keywords: deep reinforcement learning; Lyapunov; guidance law; impact angle constraint

MSC: 93D05

Academic Editor: Jonathan Blackledge

Received: 13 February 2025
Revised: 11 March 2025
Accepted: 14 March 2025
Published: 17 March 2025

Citation: Hu, Z.; Yi, W.; Xiao, L. Deep Reinforcement Learning-Based Impact Angle-Constrained Adaptive Guidance Law. *Mathematics* **2025**, *13*, 987. https://doi.org/10.3390/math13060987

1. Introduction

The traditional proportional navigation guidance (PNG) law is simple in form and easy to realize and has been widely used in engineering [1,2]. However, with modern warfare development, targets have become more and more defensive and maneuverable, necessitating missiles that not only achieve precise target strikes but also meet a specified terminal impact angle [3,4]. This is crucial for scenarios such as intercepting missiles with minimal impact angles for direct collision or striking vulnerable sections of targets like ground tanks and aircraft wings with optimal angles.

Current research into guidance laws with impact angle constraints primarily focuses on methods such as proportional navigation guidance's additional bias term, optimal

control theory, and sliding-mode variable structure control theory [5–7] among others. The basic idea of the proportional guidance method with an additional bias term [8–10] is to introduce an additional bias parameter to design a guidance law that meets the constraints of the miss distance and impact angle. The guidance law designed by this method is simple and easy to implement, but it mainly attacks stationary or slow-moving targets and lacks efficacy against rapid and highly maneuverable targets [11]. The optimal control method is also the most widely studied method; the idea is to find the guidance law with terminal impact angle constraint by minimizing the performance index. In [12], a guidance law with impact angle constraint that can hit maneuvering targets with constant acceleration is proposed based on the linear quadratic optimization theory. Reference [13] proposes a modeling method with the desired line of sight as the control target and designs an optimal guidance law for ballistic forming on the basis of constraining the relative motion direction of the terminal projectile and ballistic overload. The above optimal control methods achieve satisfactory control accuracy, but all of them require an estimation of the time-to-go, which can be challenging to ascertain in real-world scenarios.

Sliding-mode variable structure control has the characteristics of high precision and strong robustness, so it is widely used in the design of guidance laws with constraints [14,15]. In [16–18], the authors use a terminal sliding-mode (TSM) control method to design the guidance law, which can guarantee that the impact angle converges to the expected impact angle and the line-of-sight (LOS) angle rate converges to zero in finite time, and they apply the finite-time control theory to solve the specific convergence time. However, the existing TSM and fast terminal sliding-mode (FTSM) controllers have a common shortcoming, that is, the singular problem occurs easily on the terminal sliding-mode surface when the error approaches zero. To solve this problem, NTSM was proposed. NTSM can not only achieve finite time convergence but also avoid singularity problems; references [19–22] verify the above theory. It should be pointed out that in [20,22], in order to eliminate chattering, a special function is used to replace the sign function, which introduces steady-state errors, and the guidance laws above are all for non-maneuvering targets. For maneuvering targets, the authors of [23] achieve impact angle convergence by selecting the missile's lateral acceleration to enforce the terminal sliding mode on a switching surface designed using nonlinear engagement dynamics, but they do not take into account the singularity that the proposed design may exhibit in its implementation. A guidance law based on NTSM is proposed in [19], but the maneuvering upper limit of targets needs to be estimated, and there are chattering problems. In order to suppress high-frequency chattering, the boundary-layer method, reaching law method, and observer method are all effective solutions. The authors in [24] used continuous saturation function to approximate the sign function, which reduced the chattering, but this method led to a decrease in the robustness of the system as the boundary layer increased. To solve this problem, reference [25] designed a second-order sliding-mode guidance law with continuous characteristics based on the high-order sliding-mode algorithm. The super-twisting algorithm was adopted as the reaching law of the sliding-mode control to eliminate the discontinuous term in the guidance law and thus weaken the chattering. However, the traditional super-twisting algorithm has the shortcomings of a slow convergence speed when the system state is far from the equilibrium point and cannot make full use of the missile overload capacity [26]. Nowadays, with powerful nonlinear approximation and data representation methods, data-driven reinforcement learning(RL) methods have attracted considerable attention in the design of guidance law, and model-free RL has been implemented in several algorithms for finding optimal solutions, making complex decisions, and in self-learning systems which improve behavior based on previous experiences [27]. Traditional reinforcement learning methods are suited for discrete environments with small action and sample spaces.

However, real-world tasks, such as guidance and control challenges, involve large state spaces and continuous action domains with complex, nonlinear dynamics, which conventional methods find difficult to manage. Deep reinforcement learning (DRL) addresses these limitations by integrating deep neural networks with reinforcement learning, thereby enabling a more robust approach to handling the complexities of continuous and nonlinear systems. In recent years, many authors [28–32] have used Q-learning, deep Q-leaning network (DQN), deep deterministic policy gradient (DDPG), twin-delayed deep deterministic policy gradient (TD3), and other RL algorithms to train neural network models. The parameters of the missile's LOS rate, position vector, and so on are used as state variables in a Markov decision as inputs, then the trained neural network can directly map the guidance command, which provides a new idea for the design of guidance law under multi-constraint conditions. Gaudet et al. [33] improved the Proximal Policy Optimization (PPO) algorithm and proposed a three-dimensional guidance law design framework based on DRL for interceptors trained with the PPO algorithm. A comparative analysis showed that the deduced guidance law also had better performance and efficiency than the extended Zero-Effort-Miss (ZEM) policy. In a similar way, Gaudet et al. [34] developed a new adaptive guidance system using reinforcement meta-learning with a recursive strategy and a value function approximator to complete the safe landing of an agent on an asteroid with unknown environmental dynamics. This is also a good method to solve the continuous control problem, but the PPO algorithm uses a random policy to explore and exploit rewards, and this method to estimate the accurate gradient requires a large number of random actions to find the accurate gradient. Therefore, such random operation reduces the algorithm's convergence speed. In [35], an adaptive neuro-fuzzy inference sliding-mode control (ANFSMC) guidance law with impact angle constraint is proposed by combining the sliding-mode control method with an adaptive neuro-fuzzy inference system (ANFIS), and the ANFIS is introduced to adaptively update the additional control command and reduce the high-frequency chatter of the sliding-mode control (SMC), which enhances the robustness and reduces the chattering of the system; numerical simulations also verify the effectiveness of the proposed guidance law. In [36], the author proposes a three-dimensional (3D) intelligent impact-time control guidance (ITCG) law based on nonlinear relative motion relationship with the field of view (FOV) strictly constrained. By feeding back the FOV error, the modified time bias term including a guidance gain which can accelerate the convergence rate is incorporated into 3D-PNG; then, the guidance gain is obtained by DDPG in the RL framework, which makes the proposed guidance law achieve a smaller time error and have less energy loss. But in [37], the authors pointed out that the DDPG had the problem of an overestimation of the Q value, which may not give the optimal solution. Moreover, the DDPG uses a replay buffer to remove the correlations existing in the input experience (i.e., the experience replay mechanism) [38,39] and uses the target network method to stabilize the training process. As an essential part of the DDPG, experience replay significantly affects the performance and speed of the learning process by choosing the experience used to train the neural network.

Based on the above problems, this paper designs a novel adaptive impact angle-constrained guidance law based on DRL and the super-twisting algorithm. It introduces a parameter γ to replace the fractional power equal to one half in the conventional super-twisting algorithm, which improves the guidance performance and the convergence rate of the system state by adaptively adjusting this parameter. Drawing on the insights from [37,40], we apply the preferential sampling or prioritized experience replay mechanism [41] to enhance the TD3 algorithm and then obtain the adaptively optimal parameters γ using our improved DRL algorithm. Compared to the sliding-mode guidance law based on the traditional super-twisting algorithm, our algorithm is more effective in reducing

chattering, has better guidance accuracy, and achieves better constraint effects on the terminal impact angle. Moreover, the trained neural network exhibits strong generalization capabilities in unfamiliar scenarios. Finally, we design a GSTESO, different from the modified ESO in [42] that only uses nonlinear terms; the proposed GSTESO can provide faster convergence rate when the estimation error is far away from the equilibrium, and it does not depend on an upper bound of uncertainty. Furthermore, in comparison to the modified ESOs incorporating both linear and nonlinear terms discussed in [43], the GSTESO we propose does not introduce any additional parameters that require adjustment.

2. Preliminaries and Problem Setup

2.1. Relative Kinematics and Guidance Problem

The missile–target relative motion relationship is established in the inertial coordinate system, as shown in Figure 1, where Oxy is the ground inertial coordinate system, r is the missile–target relative distance, q is the missile–target LOS angle, M and T are the interceptor and the target, σ and σ_T are the flight path angle of the missile and the target, respectively, V and V_T represent the velocity of the missile and the target, respectively, a_M and a_T represent the normal acceleration of the missile and the target, respectively, η and η_T are the parameters of leading angles of the missile and the target. It is stipulated that all angles in Figure 1 are positive counterclockwise.

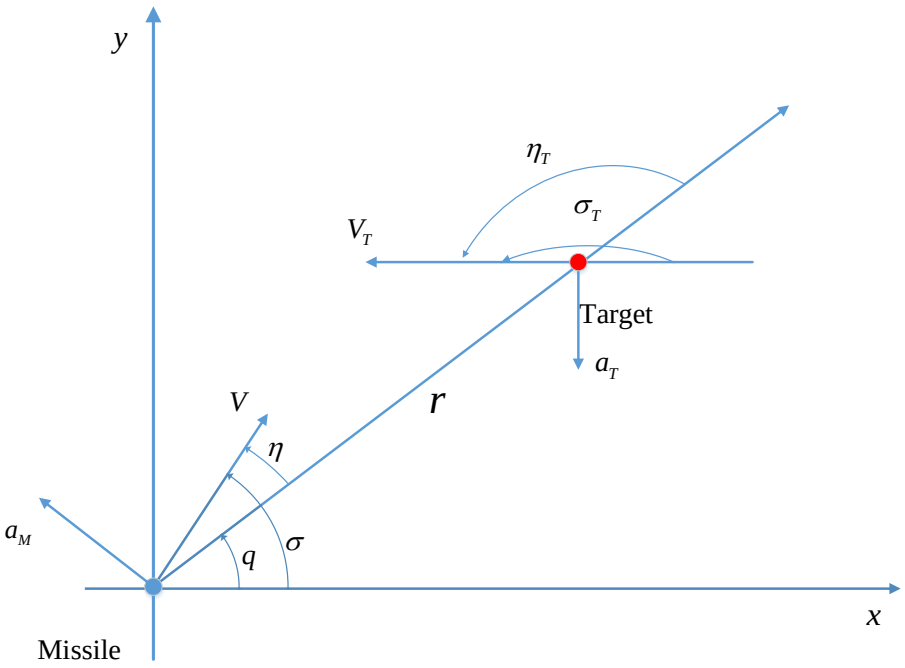


Figure 1. Relative motion model (a_M is perpendicular to V, a_T is perpendicular to V_T).

According to the missile–target relative motion relationship in Figure 1, the dynamic model of the missile terminal guidance phase is obtained as

$$\dot{r} = -V_T \cos(\sigma_T - q) - V \cos(\sigma - q) \tag{1}$$

$$\dot{q} = [V_T \sin(\sigma_T - q) - V \sin(\sigma - q)]/r \tag{2}$$

$$\dot{\sigma} = \frac{a_M}{V} \tag{3}$$

$$\dot{\sigma}_T = \frac{a_T}{V_T} \tag{4}$$

where $\dot{r}$ is the relative speed between the missile and the target; $\dot{q}$ is the LOS rate. According to the constant-bearing course, a guidance law is designed to make the LOS angle rate converge in a finite time to ensure that the missile can accurately attack the target.

The impact angle is defined as the angle between the missile velocity vector and the target velocity vector at the interception of the guidance phase during the missile's attack on the target. The moment when the missile intercepts the target is denoted as t_f, and the desired impact angle of the missile in the guidance process is denoted as σ_f.

Lemma 1 ([44]). *For a pre-designated target with an expected impact angle, there always exists a unique desired terminal LOS angle. Hence, the control of the intercept angle can be transformed into the control of the final LOS angle.*

Assuming that the desired LOS angle is q_f, the objective of this paper translates into designing a continuous guidance law that achieves $q \to q_f$, $\dot{q} \to 0$ in finite time.

2.2. Kinematic Equations for LOS Angle Tracking Error

We take the first derivative of Equation (1) and Equation (2), respectively, to obtain

$$\ddot{r} = r\dot{q}^2 + a_{Tr} - a_r \tag{5}$$

$$\ddot{q} = \frac{-2\dot{r}\dot{q}}{r} - \frac{a_q}{r} + \frac{a_{Tq}}{r} \tag{6}$$

where a_r, a_{Tr} is the acceleration of the missile and the target along the LOS angle, respectively, $a_r = a\sin(q-\sigma)$, $a_{Tr} = a_T\sin(q-\sigma_T)$; a_q, a_{Tq} is the acceleration of the missile and the target normal to the LOS angle direction, respectively. For most tactical missiles controlled by aerodynamic forces, the axial acceleration along the velocity direction is often uncontrollable, so this paper only designs the guidance law based on Equation (6).

Let e be the LOS angle tracking error, and the LOS angle rate $\dot{e} = \dot{q}$. Then, the LOS angle second-order dynamic error equation can be obtained as:

$$\ddot{e} = \frac{-2\dot{r}\dot{q}}{r} - \frac{a_q}{r} + \frac{a_{Tq}}{r} \tag{7}$$

As shown in Equation (7), when $q - \sigma = \pm\frac{\pi}{2}\text{rad}$, $a_q = 0\text{m/s}^2$, the missile cannot be controlled by the controller at this point, so it is regarded as a singular point. In order to avoid this situation, a_q is used as a virtual control input, and Equation (7) can be rewritten as follows:

$$\ddot{e} = \frac{-2\dot{r}\dot{q}}{r} - \frac{a}{r} + \frac{a - a_q + a_{Tq}}{r} \tag{8}$$

2.3. Deep Reinforcement Learning

The deep reinforcement learning algorithm is an algorithm based on the actor–critic network structure, which can update policy and value function at the same time and is suitable for solving decision problems in the continuous task space. Using the DDPG [45] algorithm as an example, the actor network uses policy functions to generate action $a = \pi(s)$, which is executed by agents to interact with the environment. The critic network uses the state-action value function $Q(s,a)$ to evaluate the performance of the action and instruct the actor network to choose a better action for the next stage.

The actor network uses a policy gradient algorithm to update parameter ϕ:

$$\nabla_{\phi\pi} J = \mathrm{E}[\nabla_a Q(s,a)|_{a=\pi(s)} \nabla_{\theta\pi}\pi(s)] \tag{9}$$

The actor network uses the estimated value $Q(s,a)$ obtained by the critic network to select actions, so the estimate should be as close to the target value as possible, which is defined by:

$$y_t = R_t + \tau Q'(s_{t+1}, \pi'(s_{t+1})) \tag{10}$$

where R_t represents the current reward, s_{t+1} represents the next state, $Q'(\cdot)$ is the state-action estimate obtained by the critic target network, and $\pi'(s_{t+1})$ is the estimated action obtained by the actor target network. All target networks use the "soft update" method to update network parameters.

The training of the critic network is based on minimizing the following loss function $L(w)$ to update the new parameter w:

$$L(w) = \mathrm{E}[(y_t - Q_w(s_t, a_t))^2] \tag{11}$$

When the two networks obtain the optimal parameters, the whole algorithm can obtain the optimal strategy, which can be regarded as a mapping from state to action, so that the missile can select the most valuable action according to the current environment state.

3. Guidance Law Design with Impact Angle Constraint

3.1. Disturbance Differential Observer Design and Stability Analysis

The observer proposed in this paper is the GSTESO, whose role is to observe the disturbance in real time. Its input is the measurable system state parameters, and its output is the observed value of the disturbance, which is used for system compensation.

Consider the following system:

$$\dot{z} = u + d(t) \tag{12}$$

where z is the controlled variable, u is the control variable, and d is the disturbance.

Assuming that the upper limit of target acceleration a_T is known [46], and due to the limitation of physical conditions, there is always an upper limit of $\dot{a}_T$. Therefore, assuming that $\dot{a}_T$ is known, it is easy to know that the disturbance term a_{Tq} also has an upper limit, that is:

$$a_M - a_q + a_{Tq} < L \tag{13}$$

Let system disturbance $d(t) = a_M - a_q + a_{Tq}$; Equation (6) becomes

$$\ddot{q} = \ddot{e} = \frac{-2\dot{r}\dot{q}}{r} - \frac{a_M}{r} + \frac{d(t)}{r} \tag{14}$$

Let $z = r\dot{q}$; Equation (6) can be written as

$$\dot{z} = -\dot{r}\dot{q} - a_M + d(t) \tag{15}$$

According to Equations (12) and (15), the proposed GSTESO can be constructed as follows:

$$\begin{cases} \dot{\hat{z}} = \hat{d}(t) - \dot{r}\dot{q} - a_M + k_3\zeta_1(e_1) \\ \dot{\hat{d}}(t) = k_4\zeta_2(e_1) \end{cases} \tag{16}$$

where $e_1 = \hat{z} - z$, and the nonlinear function is designed as

$$\begin{cases} \zeta_1(e_1) = \lfloor e_1 \rceil^{\frac{1}{2}} + e_1 \\ \zeta_2(e_1) = \frac{3}{2}\lfloor e_1 \rceil^{\frac{1}{2}} + \frac{1}{2} \cdot \text{sign}(e_1) + e_1 \end{cases} \tag{17}$$

where the symbol $\lfloor \cdot \rceil^x$ is defined as $|\cdot|^x \cdot \text{sgn}(\cdot)$, and the parameter of GSTESO can be selected as $k_3 = 2\varepsilon_0$, $k_4 = \varepsilon_0^2$.

Similarly, let $e_2 = \hat{d}(t) - d(t)$; the estimation error system can be written as:

$$\begin{cases} \dot{e}_1 = k_3\zeta_1(e_1) + e_2 \\ \dot{e}_2 = k_4\zeta_2(e_1) - \dot{d}(t) \end{cases} \tag{18}$$

Next, we verify the stability of the GSTESO. Firstly, let us introduce a new state vector as follows

$$\eta^{\mathrm{T}} = [\zeta(e_1), e_2] \tag{19}$$

The Lyapunov function is constructed as

$$V_1 = \eta^{\mathrm{T}} P \eta \tag{20}$$

where P is a symmetric and positive definite matrix, and the derivative of η is

$$\begin{aligned} \dot{\eta} &= \omega(e_1) \begin{bmatrix} k_3\zeta(e_1) + e_2 \\ k_4\zeta(e_1) - \dot{d}(t)/\omega(e_1) \end{bmatrix} \\ &= \omega(e_1)(A\eta - B\Lambda) \end{aligned} \tag{21}$$

where $\omega(e_1) = 0.5|e_1|^{-0.5} + 1$, $\Lambda = \dot{d}(t)/\zeta(e_1)$

$$A = \begin{bmatrix} k_1 & 1 \\ k_2 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \tag{22}$$

Inspired by [43], we set $\Delta(\zeta, \Lambda) = \begin{bmatrix} \eta \\ \Lambda \end{bmatrix}^T \begin{bmatrix} C & D \\ D & -\theta \end{bmatrix} \begin{bmatrix} \eta \\ \Lambda \end{bmatrix}$, where $\theta \geq 0$; then, it can be guaranteed that $\Delta(\zeta, \Lambda) \geq 0$ by choosing proper matrices C and D .

Theorem 1 ([43]). *Suppose there exists a positive definite symmetric matrix P and positive constant κ satisfying the following matrix inequality:*

$$\begin{bmatrix} A^T P + PA + \kappa P + C & PB + D^T \\ B^T P + D & -\theta \end{bmatrix} \leq 0 \tag{23}$$

System (18) is finite-time stable.

$$\begin{aligned}\dot{V}_1 &= \omega(e_1)\left[\eta^T(A^TP+PA)\eta+\Lambda B^TP\eta+\eta^TPB\Lambda\right]\\ &= \omega(e_1)\begin{bmatrix}\eta\\ \Lambda\end{bmatrix}^T\begin{bmatrix}A^TQ+QA & QB\\ B^TQ & 0\end{bmatrix}\begin{bmatrix}\eta\\ \Lambda\end{bmatrix}\\ &\le \omega(e_1)\left\{\begin{bmatrix}\eta\\ \Lambda\end{bmatrix}^T\begin{bmatrix}A^TQ+QA & QB\\ B^TQ & 0\end{bmatrix}\begin{bmatrix}\eta\\ \Lambda\end{bmatrix}+\Delta(\eta,\Lambda)\right\}\\ &\le \omega(e_1)\begin{bmatrix}\eta\\ \Lambda\end{bmatrix}^T\begin{bmatrix}-\kappa P & 0\\ 0 & 0\end{bmatrix}\begin{bmatrix}\eta\\ \Lambda\end{bmatrix}\\ &= -\frac{\kappa}{2|e_1|^{1/2}}V-\kappa V\end{aligned} \tag{24}$$

According to Equation (20), we can obtain the final result as follows:

$$\psi_{\min}\{P\}\|\eta\|^2 \le \eta^TP\eta \le \psi_{\max}\{P\}\|P\|^2 \tag{25}$$

where $\psi\{P\}$ is the eigenvalue of matrix P, and $\|\eta\|^2 = |e_1| + 2|e_1|^{(3/2)} + e_1^2 + e_2^2$.

Therefore, $|e_1|^{(1/2)} \le (V_1^{(1/2)})/(\psi_{\min}^{(1/2)}\{P\})$ holds, and Equation (24) can be rewritten as

$$\dot{V}_1 \le -\frac{\kappa\psi_{\min}^{1/2}\{P\}}{2}V^{1/2}-\kappa V \tag{26}$$

According to the finite-time stability criterion mentioned in [47], the error system (18) achieves finite-time stability.

3.2. Guidance Law Design

The sliding-mode control can be divided into two phases: the first phase is the reaching period, during which the system state converges from the initial state to the sliding-mode surface under the action of the auxiliary control term. The second phase is the sliding phase, in which the system state slides along the sliding-mode surface to the equilibrium point under the action of the equivalent control term to achieve state convergence.

Terminal sliding-mode control uses a nonlinear function as a sliding surface, which can make the system state converge in finite time, but this method has a singularity problem. In order to avoid the singularity problem and obtain higher control accuracy and faster convergence speed [48], this paper uses the non-singular terminal sliding-mode surface to design the guidance law. The sliding surface is designed by:

$$c = e + \frac{1}{\beta}|\dot{e}|^{\alpha}\mathrm{sgn}(\dot{e}) \tag{27}$$

where α, β are design parameters, $\beta > 0, 1 < \alpha < 2$.

The super-twisting algorithm is a second-order sliding-mode algorithm, which has been widely used in control problems because of its superior characteristics such as effectively reducing the chattering, strong robustness, and high-precision control.

The adaptive super-twisting algorithm is as follows:

$$\begin{cases}\dot{y}_1 = -k_1|y_1|^{1-1/\gamma}\mathrm{sgn}(y_1)+y_2\\ \dot{y}_2 = -k_2\mathrm{sgn}(y_1)\end{cases} \tag{28}$$

The algorithm converges in finite time. The proof can be found in reference [49,50].

According to Equations (27) and (28) and reference [51], the proposed RLST-NTSM is given by:

$$a_M = -2\dot{r}\dot{q} + \frac{\beta r}{\alpha}|\dot{e}|^{2-\alpha}\mathrm{sgn}(\dot{e}) + z_1 + k_1|c|^{(1-1/\gamma)}\mathrm{sgn}(c) + k_2\int\frac{\mathrm{sgn}(c)}{r}dt \tag{29}$$

where k_1, k_2 are design parameters, γ is the introduced adaptive parameter, and z_1 is the output of the GSTESO Equation (16).

Remark 1. *While the traditional super-twisting method directly sets the parameter to 2, we introduced a free parameter γ in Equation (28). Various papers [26,52–57] have proposed different adaptive methods to improve the super-twisting sliding-mode guidance law, each with its own specific form. However, most of these methods involve an adaptive gain adjustment of the parameter k_1. The study [50] already showed that for the traditional super-twisting algorithm, as long as k_1 was sufficiently large, the algorithm could converge. As for the impact of parameter variation involved in the equation $1-1/\gamma$ on control effectiveness, there is currently no literature studying this aspect, with only a few references (e.g., [58]) mentioning choosing a value slightly greater than 0.5 for $1-1/\gamma$. This suggests that some researchers may have already found that directly selecting the value as $1-1/\gamma = 0.5$ may not be appropriate. Therefore, this paper employed DRL algorithms to adaptively adjust the selection of this parameter through model-free learning to achieve adaptive tuning.*

3.3. Stability Analysis

Before delving into the analysis, let us introduce Lemma 2 [59]:

Lemma 2 ([59])**.** *Suppose $v(x)$, defined on $U \in \mathrm{R}^n$, C^1, is a smooth positive definite function, and $v(x) + \beta_1 v^{\beta 2}(x)$ is a negative semi-definite function on $U \in \mathrm{R}^n$ for $\beta_2 \in (0,1)$ and $\beta_1 \in \mathrm{R}^+$; then, there exists an area $U_0 \in \mathrm{R}_n$ such that any $v(x)$ which starts from U_0 can reach $v(x) = 0$ in finite time T_r, and the settling time T_r can be estimated by*

$$T_r \leq \frac{v^{1-\beta_2}(x_0)}{\beta_1(1-\beta_2)} \tag{30}$$

where $v(x_0)$ is the initial value of $v(x)$.

Taking the derivative of Equation (27) with respect to time, we obtain

$$\dot{c} = \dot{e} + \frac{\alpha}{\beta}|\dot{e}|^{\alpha-1}\ddot{e} = \dot{e} + \frac{\alpha}{\beta}|\dot{e}|^{\alpha-1}\left(\frac{-2\dot{r}\dot{q}}{r} - \frac{a}{r} + \frac{f}{r}\right) \tag{31}$$

Substituting Equation (27) into Equation (29), we obtain

$$\begin{aligned}\dot{c} &= \dot{e} + \tfrac{\alpha}{\beta}|\dot{e}|^{\alpha-1}\left(\tfrac{f}{r} - \tfrac{z_1}{r} - \tfrac{\beta}{\alpha}|\dot{e}|^{2-\alpha}\mathrm{sgn}(\dot{e}) - \tfrac{k_1}{r}|c|^{1-1/\gamma}\mathrm{sgn}(c) - \tfrac{k_2}{r}\int\tfrac{\mathrm{sgn}(c)}{r}dt\right)\\ &= \tfrac{\alpha}{\beta r}|\dot{e}|^{\alpha-1}\left(f - z_1 - k_1|c|^{1-1/\gamma}\mathrm{sgn}(c) - k_2\int\tfrac{\mathrm{sgn}(c)}{r}dt\right)\end{aligned} \tag{32}$$

According to Section 3.1, the estimation error of the GSTESO in (16) is dynamically finite-time stable, when the observer observation error tends to 0, and Equation (32) can be simplified as follows

$$\dot{c} = \frac{\alpha}{\beta r}|\dot{e}|^{\alpha-1}\left(-k_1|c|^{1-1/\gamma}\mathrm{sgn}(c) - k_2\int\frac{\mathrm{sgn}(c)}{r}dt\right) \tag{33}$$

Referring to reference [49], we introduce a new state vector as follows

$$\xi = [|x_1|^{1-1/\gamma}\mathrm{sgn}(x_1), x_2]^{\mathrm{T}} \tag{34}$$

where $x_1 = c$, $x_2 = -\frac{\alpha k_2}{\beta}\int \frac{\mathrm{sgn}(c)}{r}dt + \frac{\alpha}{\beta}(f - z_1)$.

Taking the derivative of Equation (34) with respect to time, we obtain

$$\dot{\xi} = \frac{|\dot{e}|^{\alpha-1}}{r}|x_1|^{-\frac{1}{\gamma}}A\xi \tag{35}$$

$$A = \begin{bmatrix} -\frac{\alpha}{\beta}(1-\frac{1}{\gamma}) & 1-\frac{1}{\gamma} \\ -\frac{\alpha k_2}{\beta} & 0 \end{bmatrix} \tag{36}$$

Since $k_1 > 0, k_2 > 0, \beta > 0, 1 < \alpha < 2, \gamma > 1$, it is easy to prove that A is a Hurwitz matrix. Thus, there exists a unique matrix $P = P^{\mathrm{T}}$ and P is the solution of the following Lyapunov equation.

$$A^{\mathrm{T}}P + PA = -Q \tag{37}$$

where every $Q = Q^{\mathrm{T}} > 0$.

For the stability analysis of Equation (35), we choose the following Lyapunov function:

$$V = \xi^{\mathrm{T}}P\xi \tag{38}$$

Taking the derivative of Equation (38) and substituting Equations (34) and (35), we obtain

$$\begin{aligned} \dot{V} &= \dot{\xi}^{\mathrm{T}}P\xi + \xi^{\mathrm{T}}P\dot{\xi} \\ &= \frac{|\dot{e}|^{\alpha-1}}{r}|x_1|^{-1/\gamma}\xi^{\mathrm{T}}(A^{\mathrm{T}}P + PA)\xi \\ &= -\frac{|\dot{e}|^{\alpha-1}}{r}|x_1|^{-1/\gamma}\xi^{\mathrm{T}}Q\xi \end{aligned} \tag{39}$$

From the definition of ξ, we know that $|x_1|^{1-1/\gamma} \le \|\xi\|_2$, where $\|\cdot\|_2$ denotes the L2-norm of the vector; then, Equation (39) becomes

$$\begin{aligned} \dot{V} &\le \frac{|\dot{e}|^{\alpha-1}}{r}\|\xi\|_2^{-\frac{1}{\gamma-1}}\xi^{\mathrm{T}}Q\xi \\ &\le -\lambda_{\min}(Q)\frac{|\dot{e}|^{\alpha-1}}{r}\|\xi\|_2^{2-\frac{1}{\gamma-1}} \\ &\le -\lambda_{\min}(Q)\lambda_{\max}^{-[1-\frac{1}{2(\gamma-1)}]}(P)\frac{|\dot{e}|^{\alpha-1}}{r}V^{1-\frac{1}{2(\gamma-1)}} \\ &\le -\lambda_{\min}(Q)\lambda_{\max}^{-[1-\frac{1}{2(\gamma-1)}]}(P)\frac{|\dot{e}|^{\alpha-1}}{r_{\max}}V^{1-\frac{1}{2(\gamma-1)}} \end{aligned} \tag{40}$$

where $\gamma_{\min}(\cdot)$ and $\gamma_{\max}(\cdot)$ are the minimum eigenvalue and maximum eigenvalue of matrix $(\cdot)$, $\lambda_{\min}(Q)\lambda_{\max}^{-[1-\frac{1}{2(\gamma-1)}]}(P)\frac{|\dot{e}|^{\alpha-1}}{r_{\max}} \ge 0$, $1 - 1/[2(\gamma-1)] \in (0,1)$. According to Lemma 3, when $|\dot{e}| \ne 0$, the equilibrium point of the system in Equation (35) is finite-time stable, and the convergence time T_c is satisfied

$$T_c \le T_r + \frac{V^{1-\frac{1}{2(\gamma-1)}}(T_r)}{\lambda_{\min}(Q)\lambda_{\max}^{-[1-\frac{1}{2(\gamma-1)}]}(P)\frac{|\dot{e}|^{\alpha-1}}{r_{\max}}\frac{1}{2(\gamma-1)}} \tag{41}$$

where $V(T_r)$ denotes the value of V at time t_r. It follows from the definition of $\dot{\xi}$ that Equation (27) for the nonsingular terminal sliding-mode surface can also converge in finite time. Once the sliding-mode surface is reached, Equation (27) is obtained

$$e + \frac{1}{\beta}|\dot{e}|^{\alpha}\mathrm{sgn}(\dot{e}) = 0 \tag{42}$$

In summary, the system in Equation (34) is finite-time stable.

3.4. Algorithm Design for Reinforcement Learning

Reinforcement learning considers the paradigm of interaction between an agent and its environment, and its goal is to make the agent take the behavior that maximizes the benefit, so as to obtain the optimal policy. Most reinforcement learning algorithms are based on a Markov decision process (MPD). The MPD model is often used to establish the corresponding mathematical model for a decision problem with uncertain state transition probability and state space and action space and to solve the reinforcement learning problem by solving the mathematical model. If the simulation covers all the state spaces encountered by the guidance law in practice and there is sufficient sampling density, the resulting guidance law is the optimal one with respect to the modeled missile and environment dynamics.

To better solve the problem in this paper, we chose the LOS angle rate, LOS angle tracking error, as well as the sliding-mode value as the state space $S : [\dot{q}, e, c]$, which can also cover the whole guidance process. Then, the neural network was used to map the state of each step out of the parameter γ in Equation (28). Through Equation (40), we know $1 - 1/[2(\gamma - 1)] \in (0, 1)$, so $\gamma > 1.5$ can be obtained; thus, we designed the action space as $A : [\gamma] \in (1.6, 3.6)$. Considering the overload problem in real flight, we restricted the action space to 200 $\mathrm{m/s^2}$.

The reward function acts as a feedback system to indicate from the environment whether the missile performance is good or bad. In order to further reduce the chattering problem and improve the guidance accuracy, the reward function designed in this work was divided into two parts.

We shaped the shaping reward function as follows:

$$R_{\text{shaping}} = \begin{cases} \frac{0.0001}{|c|}, & \text{if dr} < 0 \\ 0, & \textit{otherwise} \end{cases} \tag{43}$$

$$R_{\text{terminal}} = \begin{cases} \frac{100}{|r|}, & \text{if } r < 10 \\ 0, & \text{if otherwise} \end{cases} \tag{44}$$

$$R = \omega R_{\text{shaping}} + R_{\text{terminal}} \tag{45}$$

where $\omega > 1$ is the weight parameter, and the maximum values of R_{shaping} and R_{terminal} were set to 100 and 10,000, respectively.

In standard reinforcement learning, the aim is to maximize the long-term reward for the agent's interaction with the environment. The goal of the agent is to learn a policy $\pi : S \to A$ that maximizes the long-term reward: $G_t = \sum_{i=t}^{T} \tau^{i-t} R(s_i, a_i)$, where T is the termination time step, $R(s_i, a_i)$ is the reward the agent would receive for performing action a_i in state s_i, and τ is the discount factor $\tau \in (0, 1)$. The action value function is used to represent the expected long-term reward for performing action a_i in state s_i:

$$Q(s_t, a_t) = \mathrm{E}[G_t | s = s_t, a = a_t] = E[\sum_{i=t}^{T} \tau^{i-t} R(s_i, a_i)] \tag{46}$$

Using the Bellman equation to compute the optimal behavioral value function, the above equation becomes

$$Q*(s_t, a_t) = \mathrm{E}[R(s_t, a_t) + \tau \max_{\pi} Q*(s_{t+1}, a_{t+1})] \tag{47}$$

The reinforcement learning algorithm used in this paper was based on the TD3 algorithm. Compared with the DDPG algorithm mentioned in Section II, this algorithm has three differences: (1) it uses two critic network modules to estimate the Q value, (2) it has delayed policy updates, and (3) it uses target policy smoothing.

The first is also known as Clipped Double-Q learning. TD3 has two critic networks and two critic target networks. Both target networks produce a value and use the smaller one as the new target.

$$y = R + \tau \min_{i=1,2} Q_{ti}(s', a(s')) \tag{48}$$

where Q_{t1}, Q_{t2} are the Q values generated by the two critic target networks, respectively, and s' represents the next state. When updating the policy, because the actor network module selects the maximum Q value in the current state, Equation (48) can effectively reduce the overestimation error. Then, we take the above equation and calculate the mean squared error (MSE) of the values generated by the two critic networks, and we can obtain the loss function of the critic network.

$$L(\theta_{e=1,2}) = \mathrm{E}_{(s,a,r,s')\sim D}[(y - Q_{e=1,2}(s, a))^2] \tag{49}$$

where $\theta_{e=1,2}$ stands for the parameters of the two critic networks, D stands for the replay buffer, $Q_1(\cdot)$ and $Q_2(\cdot)$ are the Q values produced by the two critic models, respectively.

The next point is the target policy smoothing; if the D estimate produces an incorrect peak, the policy network is immediately affected, and then the performance of the whole algorithm is affected, so clipping noise was added to the target policy generation operation. Then, the action was clipped within the valid range, and Equation (48) was formulated as follows:

$$y = R + \tau \min_{i=1,2} Q_{ti}(s', \mathrm{clip}(\pi_\phi(s') + \mathrm{clip}(\varepsilon, -c, c), a_{\mathrm{low}}, a_{\mathrm{high}})) \tag{50}$$

where $\pi_\phi(\cdot)$ is the policy generated by the actor network, $\mathrm{clip}(\varepsilon, -c, c)$ is the clipping noise, and a_{low} and a_{high} refer to the upper and lower bounds of the action space, respectively. Adding noise at the target value is equivalent to adding a regularization term to the critic. In the process of sampling multiple times, a' obtains slightly different values according to different sampling noise, and y evaluates the average of the value functions of these sampling points. Therefore, y reflects the overall situation of the value functions around a'. This way, it is not affected by some extreme value functions, and the stability of critic learning is enhanced.

The last point is to delay policy update. DDPG and TD3 both focus on how to learn a good critic, because the more accurate the critic estimates, the better the policy obtained by actor training. Therefore, if the critic is poorly learned and its output has large errors, the actor is also affected by preorder errors in the learning process. TD3 proposes that the critic should be updated more frequently than the actor, that is, the policy should be updated after getting an accurate estimate of the value function, and in this paper, the value function was updated five times before the policy function was updated.

3.5. Prioritized Experience Replay

The core concept of prioritized experience replay is to replay more frequently experiences associated with very successful attempts or extremely poor performance. Therefore, the key issue lies in defining the criteria for determining experience value. In most reinforcement learning algorithms, the TD error is commonly used to update the estimation of the action value function $Q(s, a)$. The magnitude of the TD error acts as a correction for the estimation and implicitly reflects the agent's ability to learn from experiences. A larger absolute TD error implies a more significant correction to the expected action value. Furthermore, experiences with high TD errors are more likely to possess a high value and be associated with highly successful attempts. Similarly, experiences with large negative TD errors represent situations where the agent performs poorly, and these conditions are learned by the agent. Prioritized experience replay helps gradually realize the actual consequences of incorrect behavior in corresponding states and prevents repeating those mistakes under similar circumstances, thereby enhancing overall performance, so these unfavorable experiences are also considered valuable. In this paper, we adopted the absolute TD error $|\delta|$ as an index for evaluating experience value. The TD error δ_j of experience j can be calculated as:

$$\delta_j = y - Q_{e=1,2}(s, a) \tag{51}$$

where y is obtained from Equation (49), and $Q_{e=1,2}(\cdot)$ is the Q value generated by the two critic models, respectively.

We defined the probability of sampling experience j as

$$P(j) = \frac{D_j^\rho}{\sum_k D_k^\rho} \tag{52}$$

where $D_j = \frac{1}{\text{rank}(j)} > 0$, $k > 0$, $\text{rank}(j)$ represents the priority of experience j in replay buffer, and the parameter ρ is used to adjust the priority level. The definition of sampling probability can be regarded as a method to add a random factor when selecting experiences, because even experiences with a low TD error can still have the probability of replay, which ensures the diversity of sampling experiences. This diversity helps to prevent any overfitting of the neural network.

Because our priority experience replay changed the sampling method, the importance sampling weight was used to correct the error introduced by the priority replay, and the loss function of the network for gradient training was calculated to reduce the error rate of the model. The importance sampling weights were calculated as follows:

$$\omega_j = (N \cdot P(j))^{-\zeta} \tag{53}$$

where ζ is the adjustment factor of ω_j.

Based on the several algorithms introduced above, an adaptive integrated guidance algorithm with impact angle constraint based on deep reinforcement learning is presented in the following Figure 2.

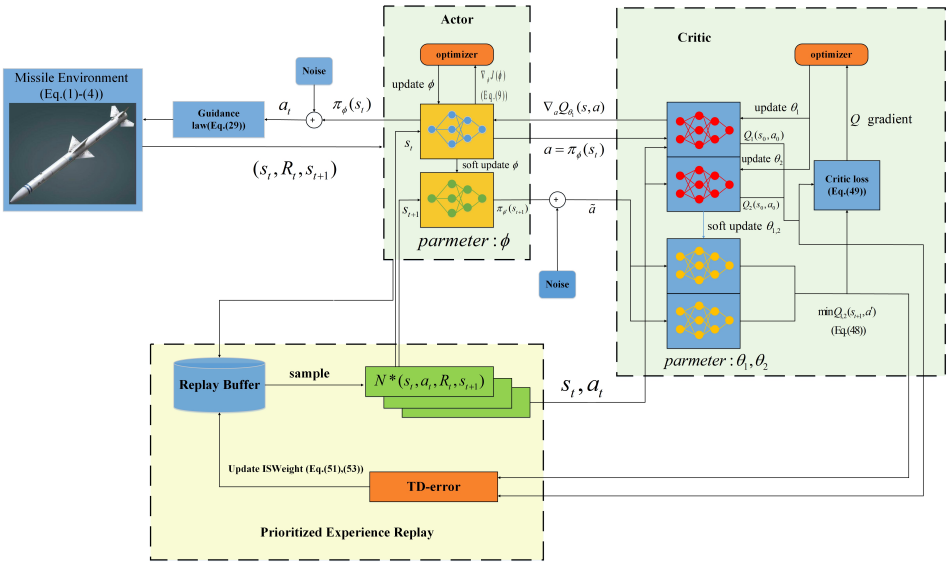


Figure 2. Schematic of the PERTD3-based RL guidance law (a_t represents the adaptive parameter to be output at time t, s_t represents the $[\dot{q}, e, c]$, $\dot{q}$ captured by the seeker at time t, in addition to the step of environmental interaction, which includes the update of parameters such as position, angle, sliding variable, etc., the rest of the equation number is one step, and each step can be found in the pseudo-code).

The TD3 algorithm with prioritized experience replay (PERTD3) is shown in Algorithm 1.

Algorithm 1 Guidance law's learning algorithm based on the PERTD3 algorithm

Initialize critic networks $Q_{\theta 1}, Q_{\theta 2}$, and actor network π_ϕ
with random parameters θ_1, θ_2, ϕ
Initialize target networks parameters $\theta_1' \leftarrow \theta_1, \theta_2' \leftarrow \theta_2, \phi' \leftarrow \phi$
Initialize replay buffer $\mathcal{D}$
for t = 1 to T **do**
  $ENV.reset()$, reset the environment initial value
  while True **do**
    Sample experience j with probability $P(j)$
    $\tilde{a} \leftarrow \pi_{\phi'}(s) + \varepsilon, \varepsilon \sim \mathrm{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
    if $t > M$ **then**
      for $j = 1, K$ **do**
        $\tilde{a} \leftarrow \pi_{\phi'}(s) + \varepsilon, \varepsilon \sim \mathrm{clip}(\mathcal{N}(0, \tilde{\sigma}), -c, c)$
        Obtain Q values by the two critic target networks as Equation (48)
        $y = R + \tau \min_{i=1,2} Q_i(s', a(s'))$
        Calculate corresponding importance-sampling weight ω_j and the TD error
        δ_j as Equations (51) and (53)
        $\omega_j = (N \cdot P(j))^{-\zeta}$
        $\delta_j = y_j - Q_\theta(s, a)$
      end for
      Calculate the loss function $L(\theta_{e=1,2})$ as Equation (49)
      $L(\theta_{e=1,2}) = \mathrm{E}_{(s,a,r,s')\sim D}[(y - Q_{e=1,2}(s,a))^2]$
      Update the critic network using gradient descent as in reference [32]
      $\nabla_{\theta_e} L(\theta_e) = \frac{1}{N} \sum_{i=1}^{N} \delta_i \nabla_{\theta_e} Q_{\theta_e}(s_i, a_i), \theta_e' = \theta_e + \alpha_{\theta_e} \nabla_{\theta_e} L(\theta_e)$
      if t mod d **then**
        Update ϕ by the deterministic policy gradient as Equation (9):
        $\nabla_\phi J(\pi_\phi) = N^{-1} \sum \nabla_a Q_{\theta_1}(s,a)|_{a=\pi_\phi(s)} \nabla_\phi \pi_\phi(s)$
        $\phi' = \phi + \alpha_\phi \nabla_\phi J(\pi_\phi)$
        Update target networks as in reference [32]:
        $\theta_e' \leftarrow \tau\theta_e + (1-\tau)\theta_e'$
        $\phi' \leftarrow \tau\phi + (1-\tau)\phi'$
      end if
    end if
  end while
end for

4. Simulation and Analysis

In this section, we considered two maneuver modes, a sinusoidal maneuver and a constant maneuver, and then tested them from three aspects: flight trajectory, normal acceleration, and LOS angle rate. Finally, the performance of the three guidance laws was analyzed in terms of miss distance.

First, when the target was on a sinusoidal maneuver, the PERTD3 and TD3 algorithms were both trained 10,000 times using the parameters in Tables 1 and 2. Both actor network and critic network were implemented by a three-layer fully connected neural network, Critic1 and Critic2 used the same network structure; the network structure is shown in Figure 3. Except for the output layer of critic, the neurons in all other layers were activated by a ReLu function, i.e.,

$$\mathrm{ReLu}(x) = \begin{cases} x, \text{ if } x \geq 0 \\ 0, \text{ if } x < 0 \end{cases} \tag{54}$$

The output layer of the critic network was activated by the tanh function, which is defined as

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{55}$$

The policy and value functions were periodically updated during optimization after accumulating trajectory rollouts of the replay buffer size.

Table 1. Dynamic model's initial simulation parameters.

Parameters	Value
Missile position (m)	(0, 0)
Missile speed (m/s)	500
Target speed (m/s)	300
Relative range (m)	15,000
LOS angle (°)	45
Missile flight path angle (°)	60
Target flight path angle (°)	180
Impact angle (°)	Random (0, 90)

Table 2. Hyperparameters for TD3 algorithm [32].

Parameters	Value
Number of hidden layers	2
BATCH_SIZE	32
Size of experience buffer	50,000
Learning rate for actor	10^{-5}
Learning rate for critic	2×10^{-5}
Policy noise	0.2
Noise bound	0.5
Soft replacement τ	0.99
Discounting factor γ	0.95
Delay steps	5
Gradient optimizer	Adam

As shown in Figure 4, our improved deep reinforcement learning algorithm was better than two other existing deep reinforcement learning algorithms in our written dynamic environment. The guidance algorithm based on PERTD3 and TD3 basically converged when approaching 800 steps, but the TD3 algorithm with prioritized experience replay could achieve higher scores and had higher stability, while the DDPG algorithm did not converge at the end.

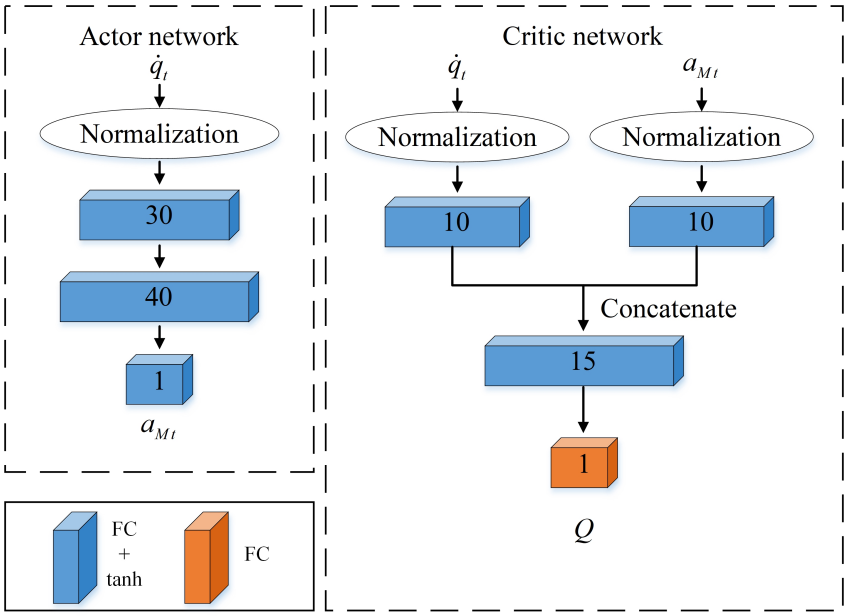


Figure 3. The network structure of PERTD3.

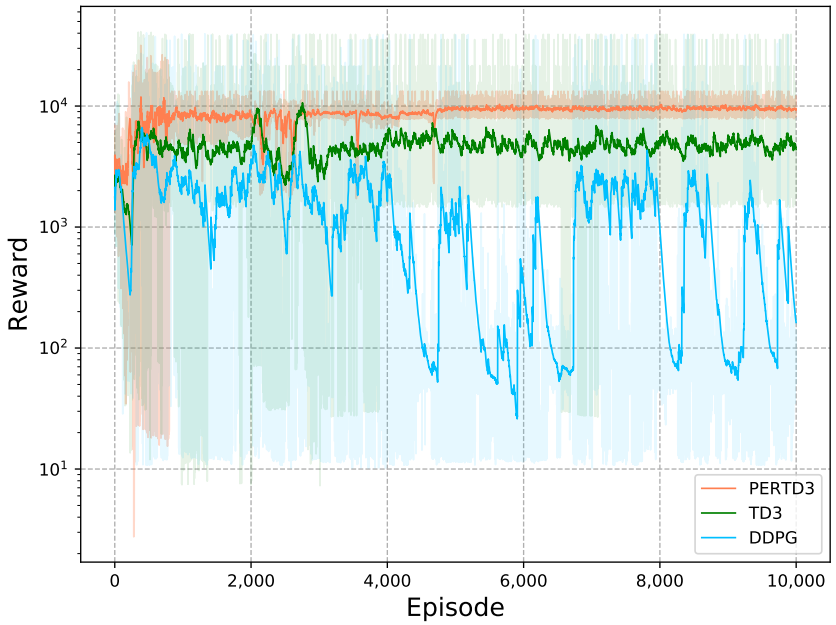


Figure 4. Comparison of learning curves of the three algorithms.

To verify the robustness of our algorithm, we took the following two maneuvers and the case of the required impact angle constraint, and case 1 was not used during training but was used to test the generalization of our reinforcement learning algorithms.

Case 1: The target performed a constant maneuver $a_T = 20\,\mathrm{m/s^2}$, i.e., the impact angle constraint was $0^\circ < q_f < 90^\circ$ for every 30°.

Case 2: The target performed a constant maneuver $a_T = 50\sin(0.5t)\mathrm{m/s^2}$, i.e., the impact angle constraint was $0^\circ < q_f < 90^\circ$ for every 30°.

As shown in Figures 5 and 6, the simulation results in case 1 are shown in Figure 5, and the simulation results in case 2 are shown in Figure 6. From Figures 5a,c,d and 6a,c,d, Tables 3 and 4, it can be seen that the missile could accurately hit the target with the desired terminal LOS angle in a limited time, the LOS angle rate converged to $0^\circ/\mathrm{s}$, and the LOS angle converged to the expected value under different desired LOS angles, which verified the superstrong control ability of the guidance law for the miss distance and the impact

angle. From Figures 5b and 6b, it can be seen that the missile acceleration command was smooth. Since the control law was continuous with respect to the sliding variable, it can be seen that the sliding variable converged to zero continuously and smoothly in finite time in Figures 5e and 6e, and the chattering phenomenon did not appear in the control channel. It can be seen from Figures 5f and 6f that the disturbance differential observer could achieve a great estimation of the system uncertainty. As can be seen from the overall Figure 6, when facing untrained scenes, our improved reinforcement learning algorithm could still hit the target, and the LOS angle converged, which proves that our algorithm has generalization and practical significance.

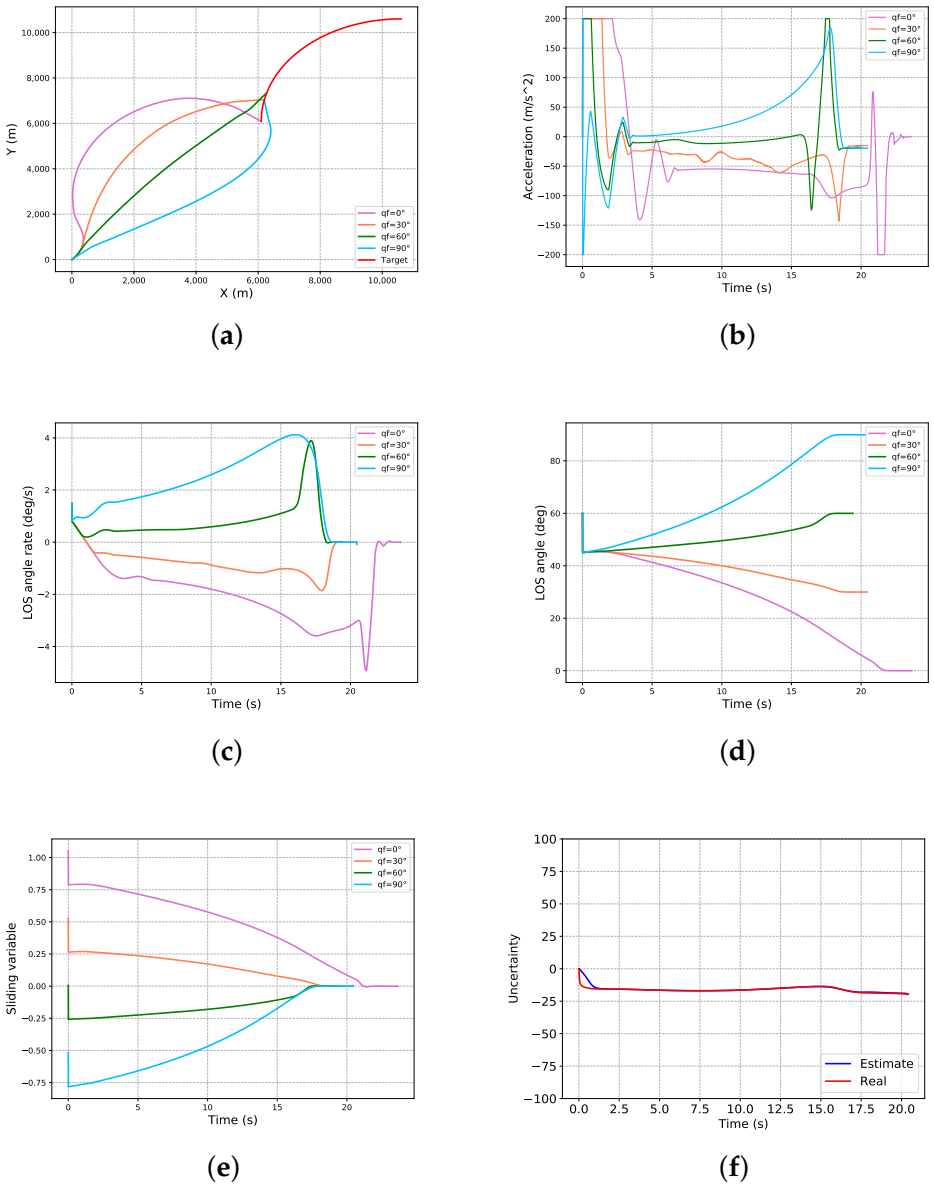


Figure 5. Simulation results for case 1: (**a**) trajectories of missile and target; (**b**) missile acceleration commands; (**c**) LOS angle rate; (**d**) LOS angle; (**e**) sliding-variable profiles; (**f**) GSTESO estimation.

Table 3. Statistics of guidance simulation results in case 1.

LOS Angle Expectation (°)	0	30	60	90
Miss distance (m)	0.0173	0.0173	0.0047	0.0174
LOS angle error (°)	−0.00001	−0.00006	−0.00006	0.00011

Table 4. Statistics of guidance simulation results in case 2.

LOS Angle Expectation (°)	0	30	60	90
Miss distance (m)	0.0071	0.0103	0.0152	0.0080
LOS angle error (°)	0.00014	−0.00004	0.00014	0.00043

In Figure 7, Figure 7a is the parameter change for case 1, and Figure 7b is the parameter change for case 2. It can be seen from the figure that the parameter change finally converged to between two and three, and the change process was smooth, which is of practical engineering significance.

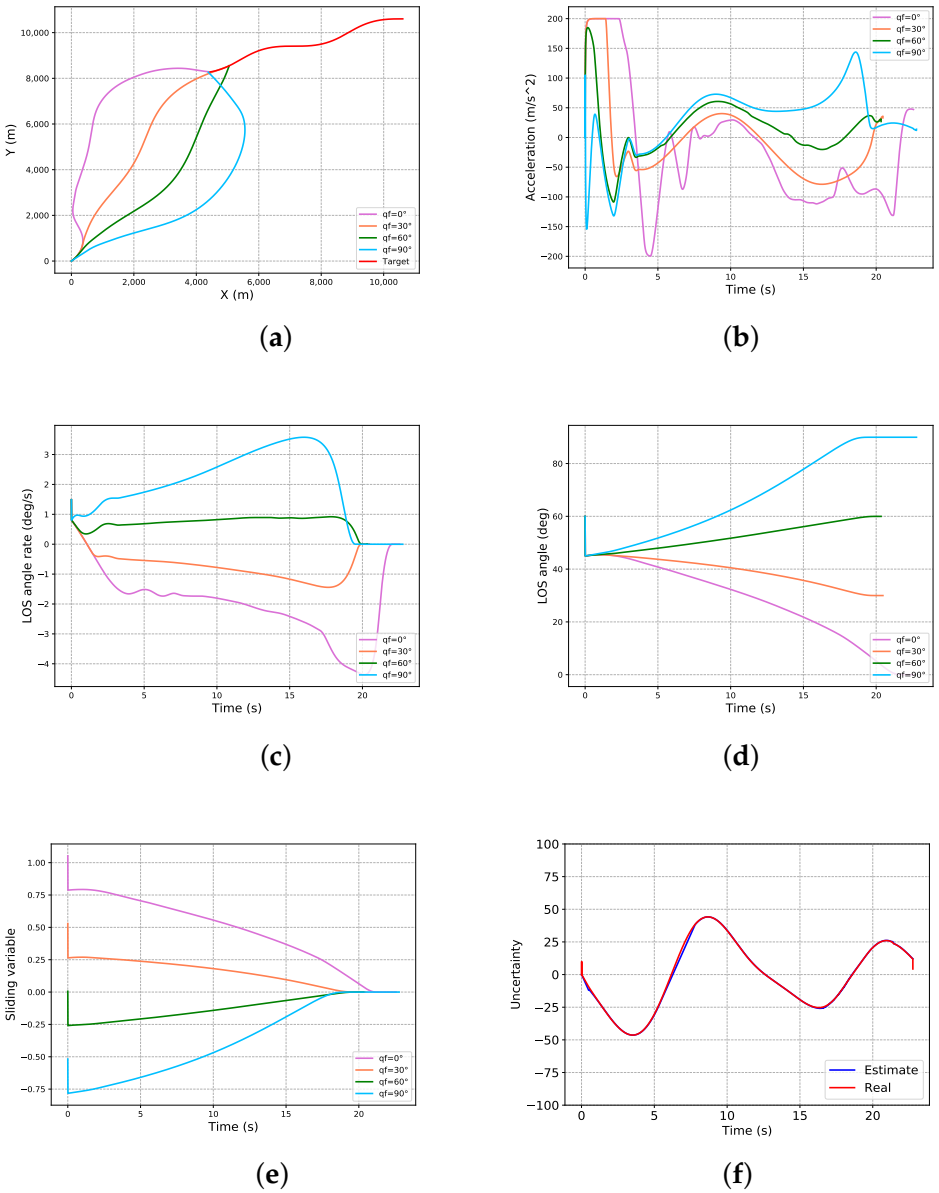


Figure 6. Simulation results for case 2: (**a**) trajectories of missile and target; (**b**) missile acceleration commands; (**c**) LOS angle rate; (**d**) LOS angle; (**e**) sliding-variable profiles; (**f**) GSTESO estimation.

In order to show the advantages of our guidance law, we compared with ST-NSTM and NSTM guidance laws, and we set the miss distance to less than 0.1 m. Our algorithm could call the RLST-NTSM guidance law. The parameters of Equation (20) were borrowed from the literature [58], where $\beta = 1$, $\alpha = 1.4$, and then we chose $k_1 = 1200$, $k_2 = 800$, after the comparison. The α, β, k_1, k_2 in the ST-NTSM guidance law was the same as before,

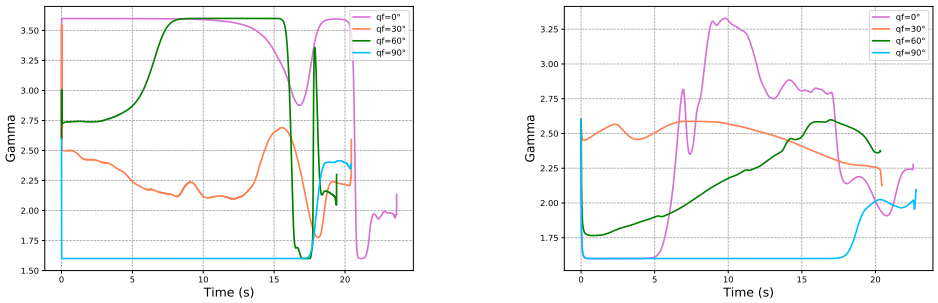
and the traditional super-twisting non-singular terminal sliding-mode guidance law (ST-NTSM [51]) was formulated as:

$$a = -2\dot{r}\dot{q} + \frac{\beta r}{\alpha}|\dot{e}|^{2-\alpha}\mathrm{sgn}(\dot{e}) + z_1 + k_1|c|^{1/2}\mathrm{sgn}(c) + k_2\int\frac{\mathrm{sgn}(c)}{r}dt \tag{56}$$

where z_1 is the output of the finite-time convergence disturbance observer (FTDO). According to the reference [58], the specific form of the FTDO [60] introduced by the author in his article is as follows:

$$\begin{aligned}
\dot{z}_0 &= v_0 - \dot{r}\dot{q} - a \\
v_0 &= -\lambda_2 L^{1/3}|z_0 - x|^{2/3}\mathrm{sgn}(z_0 - \theta) + z_1 \\
\dot{z}_1 &= v_1, v_1 = -\lambda_1 L^{1/2}|z_1 - v_0|^{1/2}\mathrm{sgn}(z_1 - v_0) + z_2 \\
\dot{z}_2 &= -\lambda_0 L\mathrm{sgn}(z_2 - v_1)
\end{aligned} \tag{57}$$

where the choice of λ in the equation can be obtained from [58]. The parameter L is used to adjust the transient performance of the estimation process and can be set to 200, which makes the observer output z_1 converge to $d(t)$ in finite time.



(**a**) (**b**)

Figure 7. The variation trend of parameter γ: (**a**) curve of parameter γ for case 1; (**b**) curve of parameter γ for case 2.

In addition, another extended state observer based on the traditional super-twisting extended state observer (STESO) [42] was also added for comparison; it was formulated as:

$$\begin{cases} \dot{\hat{z}} = \hat{d}(t) - rq - a + k_3|e_1|^{1/2}\mathrm{sgn}(e_1) \\ \dot{\hat{d}(t)} = k_4\mathrm{sgn}(e_1) \end{cases} \tag{58}$$

where k_3, k_4 were the same as in the GSTESO.

And the NSTM guidance law was of the form:

$$a = -2\dot{r}\dot{q} + \frac{\beta r}{\alpha}|\dot{e}|^{2-\alpha}\mathrm{sgn}(\dot{e}) + K\mathrm{sgn}(c) \tag{59}$$

where the values of α, β were also the same as before, and the value of the switching gain K was 400 according to reference [58].

As shown in Table 5 and Figure 8, we simulated different γ values corresponding to ST-NTSM for case 2. It can be seen from Table 5 that no matter how the parameter γ value was selected, the accuracy of ST-NTSM was not as high as our algorithm, only the values of 2.3 and 2.6 qualified, given that the miss distance had to be less than 0.1 m. Figure 8 also shows that ST-NTSM still had a small number of chattering problems. And as the parameter γ increased, the convergence rate of the guidance law was slower, while our guidance law converged faster, and the chattering phenomenon did not occur.

Table 5. Statistics of simulation results for case 2 with different parameter γ values.

Parameter γ	1.7	2.3	2.6	2.9	3.2
LOS angle expectation (°)	90	90	90	90	90
Miss distance (m)	0.1223	0.0834	0.2577	0.0513	0.1666
LOS angle error (°)	0.0040	0.00001	0.000005	0.0006	0.0028
Sliding variable	0.3430	0.0015	0.0002	−0.0187	−0.0004

The simulation results for case 2 with $q_f = 90°$ are shown in Figure 9 and Table 6. The RLST-NTSM proposed in this paper was better than the other two guidance laws in terms of accuracy and convergence of the final LOS angle error, and Figure 9b shows that the missile guided by NTSM did not hit the target, while the performance of ST-NTSM was relatively good. However, according to Figure 9a,d, there was still a certain chattering phenomenon, and the proposed algorithm further reduced the system chattering. It can also be seen from Figure 9c that RLST-NTSM reached the equilibrium state of the system earlier than ST-NTSM. From Figure 9d,e, it can be seen that the sliding-mode variable and LOS angle rate of ST-NTSM and NTSM did not converge well, which would greatly affect their practical engineering application. Finally, it should be noted that although the three observers could be directly added to any guidance law, it can be seen from Figure 9f–h that the GSTESO performed better than the other two observers, as its initial uncertainty observation error was minimal.

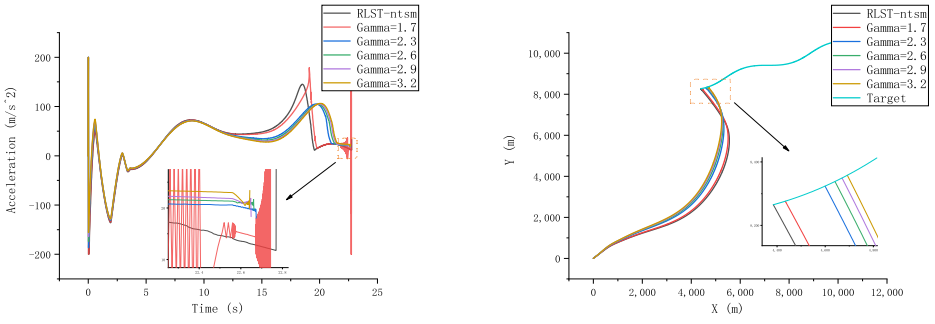
Table 6. Statistics of simulation results for case 2 with different parameter γ values.

	RLST-NTSM	ST-NTSM	NTSM
LOS angle expectation (°)	90	90	90
Miss distance (m)	0.0080	0.042	92.64
LOS angle error (°)	0.00004	0.0020	−104.2
Sliding variable	−0.0001	−20.55	−18.01

As overload saturation emerged in the simulation, we eventually decided to consider the constraints of the autopilot in a simplified manner; combined with Equation (14), we studied the following augmented system.

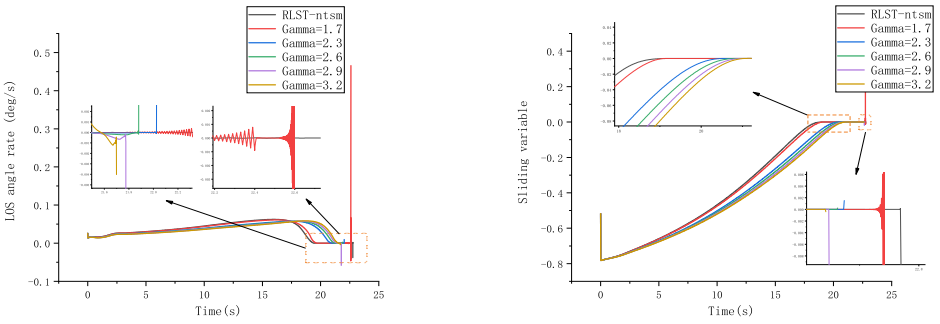
$$\begin{cases} \dot{a} = x_4 \\ \dot{x}_4 = -\omega_n^2 a + \omega_n^2 u + h \\ \dot{h} = H \end{cases} \tag{60}$$

where $h = d_a - 2\xi_n\omega_n\dot{a}$, d_a is the modeling error, ξ_n and ω_n are the damping ratio and natural frequency of the autopilot.



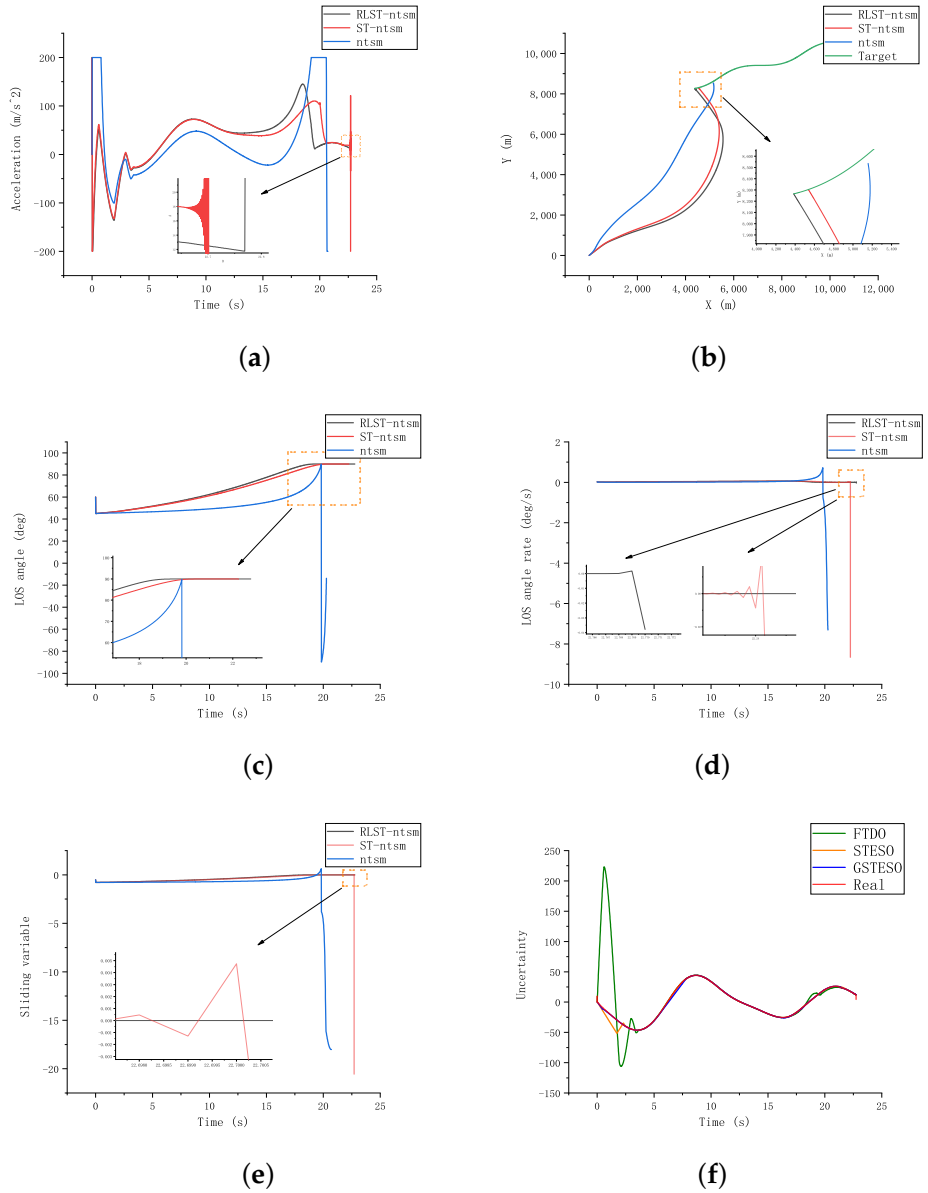
(**a**) (**b**)

Figure 8. *Cont.*



(**c**) (**d**)

Figure 8. Comparison of guidance laws with different parameters γ for case 2: (**a**) missile acceleration commands; (**b**) trajectories of missile and target; (**c**) LOS angular rate; (**d**) sliding-variable profiles.

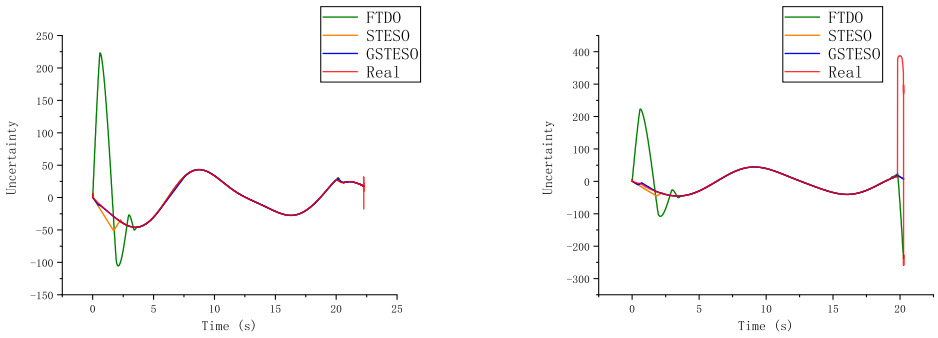


(**a**) (**b**)

(**c**) (**d**)

(**e**) (**f**)

Figure 9. *Cont.*



(g) **(h)**

Figure 9. Comparison for three guidance law and three observer under case 2: (**a**) missile acceleration commands; (**b**) trajectories of missile and target; (**c**) LOS angle; (**d**) LOS angular rate; (**e**) sliding-variable profiles; (**f**) three observers' estimation in RLST-NTSM. (**g**) three observers' estimation in ST-NTSM; (**h**) three observers' estimation in NTSM.

This paper employed an autopilot approach similar to the one described in reference [61], and its formulation is given by

$$\begin{aligned} u &= \omega_n^{-2}[\omega_n^2 a - z_{23} + L_{22}\mathrm{fal}(e_{21}, \vartheta_{21}, \eta) - L_{21}^2 e21 \\ &- H_2 + \dot{\varphi}_2 - m_1 s_2 - m_2 sig^{\chi} s_2] \\ s_2 &= \varepsilon_2 + \alpha_2\varepsilon_1 + \alpha_2\beta_2(\varepsilon_1) \\ \beta_2(\varepsilon_1) &= \begin{cases} sig^{\chi}\varepsilon_1 \mathrm{if}|\varepsilon_1| \geq \eta_2 \\ b_1\varepsilon_1 + b_2 sig^2\varepsilon_1 \mathrm{if}|\varepsilon_1| < \eta_2 \end{cases} \end{aligned} \tag{61}$$

where $H_2 = \alpha_2\varepsilon_2 + \alpha_2\dot{\beta}_2(\varepsilon_1)$, $\varepsilon_1 = z_{21} - \varphi_1, \varepsilon_2 = \dot{z}_{21} - \varphi_2$.

We selected the virtual control signal a_c as in Equation (29). The first and second derivatives of a_c were estimated using the following filter instructions:

$$\begin{aligned} \dot{\varphi}_1 &= \varphi_2 \\ \dot{\varphi}_2 &= -2\xi\omega\varphi_2 - \omega^2(\varphi_1 - a) \end{aligned} \tag{62}$$

where $b_1 = (2-\chi)\eta^{\chi-1}, b_2 = (1-\chi)\eta^{\chi-2}$, $\xi = 0.707$ and $\omega = 35$ are the damping ratio and natural frequency of the command filter, respectively, φ_1 and φ_2 are the estimate and the corresponding derivative, respectively, α_2 and η are positive constants. The system was observed using the following ESO

$$\begin{cases} z_{21} = z_{22} - L_{21}e_{21} \\ z_{22} = -\omega_n^2 a + \omega_n^2 u + z_{23} - L_{22}\mathrm{fal}(e_{21}, \vartheta_{21}, \eta) \\ z_{23} = -L_{23}\mathrm{fal}(e_{21}, \vartheta_{22}, \eta) \\ e_{21} = z_{21} - a \end{cases} \tag{63}$$

where $e_{21} = z_{21} - a, e_{22} = z_{22} - x_4$ are the estimate errors, L_{21}, L_{22}, and L_{23} are the positive observer gains to be designed, $0 < \vartheta_{21}, \vartheta_{22} < 1$, and η is a small positive constant.

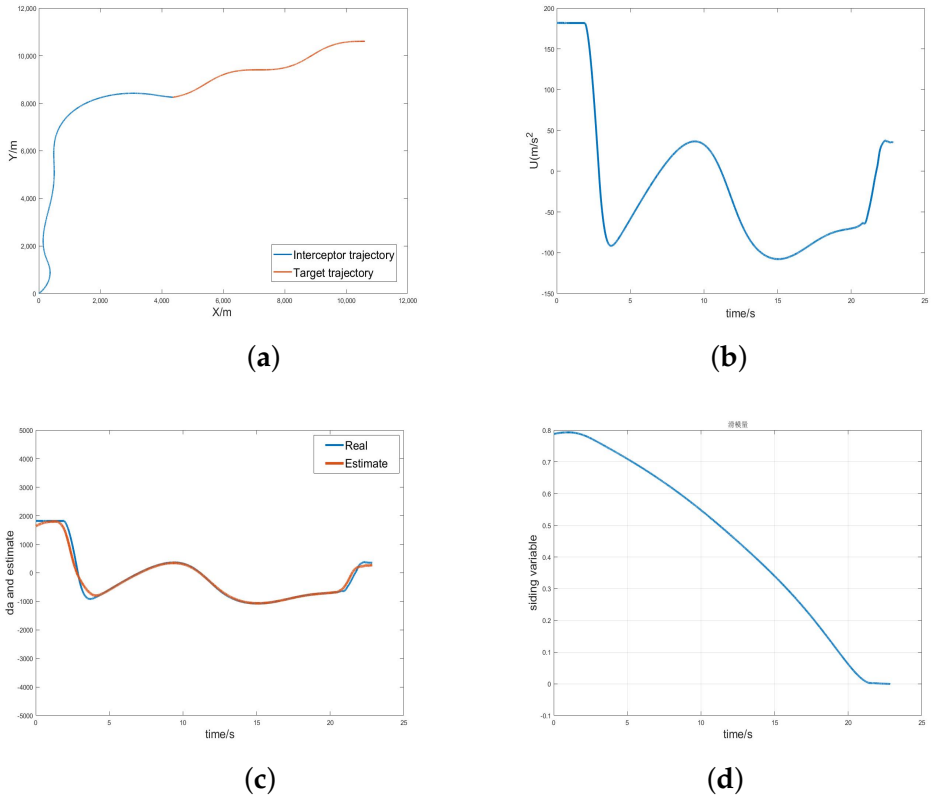
The stability of the autopilot component can be established by consulting reference [61], and a detailed analysis was not conducted. The autopilot parameters were $\omega_n = 35$ rad/s, $\xi_n = 0.8$, $d_a = 0.1\omega_n^2 u$. The design parameters for implementing guidance law (61) were set to $\alpha_2 = 0.5, \chi = 0.6, \eta_2 = 0.8, m_1 = 200$, and $m_2 = 50$. The parameters of the ESOs were selected as $L_{21} = 80, L_{22} = 800, L_{23} = 20,000, \vartheta_{21} = 0.7, \eta = 0.02$.

As illustrated in Figure 10 and Table 7, this paper focused on simulating the scenario that produced the maximum normal overload, specifically, $q_f = 90^\circ$ in case 2; the results

showed that considering the autopilot lag characteristics did not affect the performance of the proposed guidance law.

Table 7. Simulation results considering the autopilot lag for case 2.

	RLST-NTSM Considering Autopilot Lag
LOS angle expectation (°)	0
Miss distance (m)	0.045
LOS angle error (°)	−0.0305
Sliding variable	−0.0013



(**a**) (**b**)

(**c**) (**d**)

Figure 10. Simulation results considering the autopilot lag: (**a**) trajectories of missile and target; (**b**) missile acceleration commands; (**c**) autopilot error estimate; (**d**) LOS angular rate.

5. Conclusions

In this study, we proposed a novel guidance law, named RLST-NTSM, which combined deep reinforcement learning with sliding-mode guidance. That guidance law was specifically designed for the interception of maneuvering targets with terminal LOS angle constraints. The RLST-NTSM guidance law enhanced the traditional ST-NTSM by introducing an adaptive gamma γ parameter, which was optimized through a DRL algorithm to achieve optimal adaptive control. Through the design of the reward function, it effectively reduced the chattering phenomenon and improved the convergence rate of the system state; this is beneficial to improve the stability of the missile body during the actual flight. Furthermore, a GSTESO was employed to estimate and compensate for system disturbances. Distinguished from conventional STESO and FTDO, the GSTESO provided stronger robustness and disturbance rejection ability. Utilizing Lyapunov's stability theory, we proved the finite-time convergence of the closed-loop guidance system. Simulation results showed that our algorithm greatly improved the overall performance compared with the traditional ST-NTSM and NTSM, and the interception accuracy was improved by at least five times. In response to the emergence of overload saturation within our simulations, we considered

the delay characteristics of the autopilot in the concluding segment of our study. Our analysis revealed that these constraints did not impair the guidance performance. In the future, the primary objective of the subsequent phase of research will be to expand the guidance environment into a three-dimensional space. Following this, we intend to deploy the guidance algorithm in a real-world setting to conduct experimental trials.

Author Contributions: Conceptualization, Z.H. and W.Y.; methodology, Z.H.; investigation, Z.H.; writing—original draft preparation, L.X.; writing—review and editing, L.X.; funding acquisition, W.Y. and L.X. All authors have read and agreed to the published version of the manuscript.

Funding: This work has been supported by the National Natural Science Foundation of China (Grant No. 62471235).

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

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

Nomenclature

MPD	Markov Decision Process
RL	Reinforcement learning
s	State vector in the MDP
a	Action vector in the MDP
θ	The parameter of the critic network
ϕ	The parameter of the actor network
$R(s_t, a_t)$	Reward for being in states when selecting the corresponding action a at time t
G_t	The cumulative sum of subsequent rewards after time point t
$p(x)$	The probability density p associated with random variable x
$Q^{\pi}(s_t, a_t)$	Expectation of $f(x)$, i.e., $\int_{-\infty}^{+\infty} p(x)f(x)dx$
$\mathrm{E}[f(x)]$	The action value for an agent taking action a by following policy π in the state at time t

References

1. Ristic, B.; Beard, M.; Fantacci, C. An overview of particle methods for random finite set models. *Inf. Fusion* **2016**, *31*, 110–126. [CrossRef]
2. Liu, D.; Wang, Z.; Liu, Y.; Alsaadi, F.E. Extended Kalman filtering subject to random transmission delays: Dealing with packet disorders. *Inf. Fusion* **2020**, *60*, 80–86. [CrossRef]
3. Wang, X.; Zhang, Y.; Wu, H. Distributed cooperative guidance of multiple anti-ship missiles with arbitrary impact angle constraint. *Aerosp. Sci. Technol.* **2015**, *46*, 299–311. [CrossRef]
4. Ratnoo, A.; Ghose, D. Impact angle constrained interception of stationary targets. *J. Guid. Control Dyn.* **2008**, *31*, 1817–1822. [CrossRef]
5. Wenjie, Z.; Shengnan, F.U.; Wei, L.I.; Qunli, X. An impact angle constraint integral sliding mode guidance law for maneuvering targets interception. *J. Syst. Eng. Electron.* **2020**, *31*, 168–184.
6. Shima, T.; Golan, O.M. Head pursuit guidance. *J. Guid. Control. Dyn.* **2007**, *30*, 1437–1444. [CrossRef]
7. Wei, J.H.; Sun, Y.L.; Zhang, J. Integral sliding mode variable structure guidance law with landing angle constrain of the guided projectile. *J. Ordnance Equip. Eng.* **2019**, *40*, 103–105. (In Chinese)
8. Zhou, H.; Wang, X.; Bai, B.; Cui, N. Reentry guidance with constrained impact for hypersonic weapon by novel particle swarm optimization. *Aerosp. Sci. Technol.* **2018**, *78*, 205–213. [CrossRef]
9. Zhang, Y.A.; Ma, G.X.; Wu, H.L. A biased proportional navigation guidance law with large impact angle constraint and the time-to-go estimation. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2014**, *228*, 1725–1734. [CrossRef]
10. Ma, S.; Wang, X.; Wang, Z.; Yang, J. BPNG law with arbitrary initial lead angle and terminal impact angle constraint and time-to-go estimation. *Acta Armamentarii* **2019**, *40*, 71–81.
11. Behnamgol, V.; Vali, A.R.; Mohammadi, A. A new adaptive finite time nonlinear guidance law to intercept maneuvering targets. *Aerosp. Sci. Technol.* **2017**, *68*, 416–421. [CrossRef]

12. Taub, I.; Shima, T. Intercept angle missile guidance under time varying acceleration bounds. *J. Guid. Control Dyn.* **2013**, *36*, 686–699. [CrossRef]
13. Li, H.; She, H.P. Trajectory shaping guidance law based on ideal line-of-sight. *Acta Armamentarii* **2014**, *35*, 1200–1204. (In Chinese)
14. Chen, S.F.; Chang, S.J.; Wu, F. A sliding mode guidance law for impact time control with field of view constraint. *Acta Armamentarii* **2019**, *40*, 777–787. (In Chinese)
15. Liang, Z.S.; Feng, X.A.; Xue, Y. Design on adaptive weighted guidance law for underwater intelligent vehicle tracking target. *J. Northwestern Polytech. Univ.* **2021**, *39*, 302–308. (In Chinese) [CrossRef]
16. Zhao, J.L.; Zhou, J. Strictly convergent nonsingular terminal sliding mode guidance law with impact angle constraints. *Opt. J. Light Electron Opt.* **2016**, *127*, 10971–10980. [CrossRef]
17. Yu, J.Y.; Xu, Q.J.; Zhi, Y. A TSM control scheme of integrated guidance/autopilot design for UAV. In Proceedings of the 3rd International Conference on Computer Research and Development, Shanghai, China, 11–13 March 2011; pp. 431–435.
18. Zhang, Y.X.; Sun, M.W.; Chen, Z.Q. Finite-time convergent guidance law with impact angle constraint based on sliding-mode control. *Nonlinear Dyn.* **2012**, *70*, 619–625. [CrossRef]
19. Kumar, S.R.; Rao, S.; Ghose, D. Nonsingular terminal sliding mode guidance with impact angle constraints. *J. Guid. Control Dyn.* **2014**, *37*, 1114–1130. [CrossRef]
20. Feng, Y.; Bao, S.; Yu, X.H. Design method of non-singular terminal sliding mode control systems. *Control Decis.* **2002**, *17*, 194–198. (In Chinese)
21. Xiong, S.; Wang, W.; Liu, X.; Wang, S.; Chen, Z. Guidance law against maneuvering targets with intercept angle constraint. *ISA Trans.* **2014**, *53*, 1332–1342. [CrossRef]
22. Zhang, X.J.; Liu, M.Y.; Li, Y. Nonsingular terminal slidingmode-based guidance law design with impact angle constraints. *Iran. J. Sci. Technol. Electr. Eng.* **2019**, *43*, 47–54.
23. Kumar, S.R.; Rao, S.; Ghose, D. Sliding-mode guidance and control for all-aspect interceptors with terminal angle constraints. *J. Guid. Control Dyn.* **2012**, *35*, 1230–1246. [CrossRef]
24. Shima, T.; Golan, O.M. Exo-atmospheric guidance of an accelerating interceptor missile. *J. Frankl. Inst.* **2012**, *349*, 622–637. [CrossRef]
25. He, S.; Lin, D. Guidance laws based on model predictive control and target manoeuvre estimator. *Trans. Inst. Meas. Control* **2016**, *38*, 1509–1519. [CrossRef]
26. Yang, F.; Zhang, K.; Yu, L. Adaptive Super-Twisting Algorithm-Based Nonsingular Terminal Sliding Mode Guidance Law. *J. Control Sci. Eng.* **2020**, *2020*, 1058347. [CrossRef]
27. Russell, S.J.; Norvig, P. *Artificial Intelligence: A Modern Approach*; Pearson: London, UK, 2016.
28. Chithapuram, C.; Jeppu, Y.; Kumar, C.A. Artificial Intelligence learning based on proportional navigation guidance. In Proceedings of the 2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Mysore, India, 22–25 August 2013; pp. 1140–1145.
29. Chithapuram, C.U.; Cherukuri, A.K.; Jeppu, Y.V. Aerial vehicle guidance based on passive machine learning technique. *Int. J. Intell. Comput. Cybern.* **2016**, *9*, 255–273. [CrossRef]
30. Wu, M.Y.; He, X.J.; Qiu, Z.M.; Chen, Z.H. Guidance law of interceptors against a high-speed maneuvering target based on deep Q-Network. *Trans. Inst. Meas. Control* **2022**, *44*, 1373–1387. [CrossRef]
31. He, S.; Shin, H.S.; Tsourdos, A. Computational missile guidance: A deep reinforcement learning approach. *J. Aerosp. Inf. Syst.* **2021**, *18*, 571–582. [CrossRef]
32. Hu, Z.; Xiao, L.; Guan, J.; Yi, W.; Yin, H. Intercept Guidance of Maneuvering Targets with Deep Reinforcement Learning. *Int. J. Aerosp. Eng.* **2023**, *2023*, 7924190. [CrossRef]
33. Gaudet, B.; Furfaro, R.; Linares, R. Reinforcement learning for angle-only intercept guidance of maneuvering targets. *Aerosp. Sci. Technol.* **2020**, *99*, 105746. [CrossRef]
34. Gaudet, B.; Linares, R.; Furfaro, R. Adaptive guidance and integrated navigation with reinforcement meta-learning. *Acta Astronaut.* **2020**, *169*, 180–190. [CrossRef]
35. Li, Q.; Zhang, W.; Han, G.; Yang, Y. Adaptive neuro-fuzzy sliding mode control guidance law with impact angle constraint. *IET Control Theory Appl.* **2015**, *9*, 2115–2123. [CrossRef]
36. Wang, N.; Wang, X.; Cui, N.; Li, Y.; Liu, B. Deep reinforcement learning-based impact time control guidance law with constraints on the field-of-view. *Aerosp. Sci. Technol.* **2022**, *128*, 107765. [CrossRef]
37. Fujimoto, S.; Hoof, H.V.; Meger, D. Addressing Function Approximation Error in Actor-Critic Methods. *arXiv* **2018**, arXiv:1802.09477.
38. Lin, L.J. *Reinforcement Learning for Robots Using Neural Networks*; Carnegie Mellon University: Pittsburgh, PA, USA, 1993.
39. Adam, S.; Busoniu, L.; Babuska, R. Experience replay for real-time reinforcement learning control. *IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.)* **2011**, *42*, 201–212. [CrossRef]

40. Hou, Y.; Liu, L.; Wei, Q.; Xu, X.; Chen, C. A novel DDPG method with prioritized experience replay. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; pp. 316–321.
41. Schaul, T.; Quan, J.; Antonoglou, I.; Silver, D. Prioritized experience replay. *arXiv* **2015**, arXiv:1511.05952.
42. Sheikhbahaei, R.; Khankalantary, S. Three-dimensional continuous-time integrated guidance and control design using model predictive control. *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.* **2023**, *237*, 503–515. [CrossRef]
43. Zhao, L.; Gu, S.; Zhang, J.; Li, S. Finite-time trajectory tracking control for rodless pneumatic cylinder systems with disturbances. *IEEE Trans. Ind. Electron.* **2021**, *69*, 4137–4147. [CrossRef]
44. Kim, S.; Lee, J.G.; Han, H.S. Biased PNG law for impact with angular constraint. *IEEE Trans. Aerosp. Electron. Syst.* **1998**, *34*, 277–288.
45. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous control with deep reinforcement learning. *arXiv* **2019**, arXiv:1509.02971.
46. He, S.; Lin, D. A robust impact angle constraint guidance law with seeker's field-of-view limit. *Trans. Inst. Meas. Control* **2015**, *37*, 317–328. [CrossRef]
47. Zhao, L.; Gu, S.; Zhang, J.; Li, S. Continuous finite-time control for robotic manipulators with terminal sliding mode. *Automatic* **2005**, *41*, 1957–1964.
48. Feng, Y.; Yu, X.; Man, Z. Non-singular terminal sliding mode control of rigid manipulators. *Automatica* **2002**, *38*, 2159–2167. [CrossRef]
49. Moreno, J.A.; Osorio, M. Strict Lyapunov functions for the super-twisting algorithm. *IEEE Trans. Autom. Control* **2012**, *57*, 1035–1040. [CrossRef]
50. Edwards, C.; Shtessel, Y.B. Adaptive continuous higher order sliding mode control. *Automatica* **2016**, *65*, 183–190. [CrossRef]
51. He, S.; Wang, W.; Wang, J. Discrete-time super-twisting guidance law with actuator faults consideration. *Asian J. Control* **2017**, *19*, 1854–1861. [CrossRef]
52. Utkin, V.I.; Poznyak, A.S. Adaptive sliding mode control with application to super-twist algorithm: Equivalent control method. *Automatica* **2013**, *49*, 39–47. [CrossRef]
53. Castillo, I.; Fridman, L.; Moreno, J.A. Super-twisting algorithm in presence of time and state dependent perturbations. *Int. J. Control* **2018**, *91*, 2535–2548. [CrossRef]
54. Yang, F.; Wei, C.; Cui, N.; Xu, J. Adaptive generalized super-twisting algorithm based guidance law design. In Proceedings of the 2016 14th International Workshop on Variable Structure Systems (VSS), Nanjing, China, 1–4 June 2016; pp. 47–52.
55. Kumar, S.; Soni, S.K.; Sachan, A.; Kamal, S.; Bandyopadhyay, B. Adaptive super-twisting guidance law: An event-triggered approach. In Proceedings of the 2022 16th International Workshop on Variable Structure Systems (VSS), Rio de Janeiro, Brazil, 11–14 September 2022; pp. 190–195.
56. Shtessel, Y.B.; Moreno, J.A.; Plestan, F.; Fridman, L.M.; Poznyak, A.S. Super-twisting adaptive sliding mode control: A Lyapunov design. In Proceedings of the 49th IEEE Conference on Decision and Control (CDC), Atlanta, GA, USA, 15–17 December 2010; pp. 5109–5113.
57. Kumar, S.; Sharma, R.K.; Kamal, S. Adaptive super-twisting guidance law with extended state observer. In Proceedings of the IECON 2021–47th Annual Conference of the IEEE Industrial Electronics Society, Toronto, ON, Canada, 13–16 October **2021**; pp. 1–6.
58. He, S.; Lin, D.; Wang, J. Continuous second-order sliding mode based impact angle guidance law. *Aerosp. Sci. Technol.* **2015**, *41*, 199–208. [CrossRef]
59. Bhat, S.P.; Bernstein, D.S. Finite-time stability of continuous autonomous systems. *SIAM J. Control Optim.* **2000**, *38*, 751–766. [CrossRef]
60. Levant, A. Higher-order sliding modes, differentiation and output-feedback control. *Int. J. Control* **2003**, *76*, 924–941. [CrossRef]
61. Wang, X.; Lu, H.; Huang, X.; Zuo, Z. Three-dimensional terminal angle constraint finite-time dual-layer guidance law with autopilot dynamics. *Aerosp. Sci. Technol.* **2021**, *116*, 106818. [CrossRef]