aerospace

MDPI

Article

A Real-Time and Optimal Hypersonic Entry Guidance Method Using Inverse Reinforcement Learning

Linfeng Su, Jinbo Wang * and Hongbo Chen

School of Systems Science and Engineering, Sun Yat-sen University, Guangzhou 510006, China
* Correspondence: wangjinbo@mail.sysu.edu.cn

Abstract: The mission of hypersonic vehicles faces the problem of highly nonlinear dynamics and complex environments, which presents challenges to the intelligent level and real-time performance of onboard guidance algorithms. In this paper, inverse reinforcement learning is used to address the hypersonic entry guidance problem. The state-control sample pairs and state-rewards sample pairs obtained by interacting with hypersonic entry dynamics are used to train the neural network by applying the distributed proximal policy optimization method. To overcome the sparse reward problem in the hypersonic entry problem, a novel reward function combined with a sophisticated discriminator network is designed to generate dense optimal rewards continuously, which is the main contribution of this paper. The optimized guidance methodology can achieve good terminal accuracy and high success rates with a small number of trajectories as datasets while satisfying heating rate, overload, and dynamic pressure constraints. The proposed guidance method is employed for two typical hypersonic entry vehicles (Common Aero Vehicle-Hypersonic and Reusable Launch Vehicle) to demonstrate the feasibility and potential. Numerical simulation results validate the real-time performance and optimality of the proposed method and indicate its suitability for onboard applications in the hypersonic entry flight.

Keywords: hypersonic entry; inverse reinforcement learning; few datasets; autonomous guidance; real-time optimal control

Citation: Su, L.; Wang, J.; Chen, H. A Real-Time and Optimal Hypersonic Entry Guidance Method Using Inverse Reinforcement Learning. *Aerospace* **2023**, *10*, 948. https://doi.org/10.3390/aerospace10110948

Academic Editor: Giuseppe Pezzella

Received: 13 October 2023
Revised: 3 November 2023
Accepted: 4 November 2023
Published: 7 November 2023

1. Introduction

A hypersonic vehicle is a specific type of vehicle that traverses the atmosphere at speeds exceeding Mach 5. In recent years, the prominence of gliding hypersonic vehicles has increased significantly due to their remarkable capabilities for long-range and cross-range flights, as well as their ability to achieve high-precision targeting in both military and civilian domains [1]. However, when operating in complex flight environments characterized by factors such as heat, pressure, and overload, the system dynamics of a hypersonic vehicle become coupled, uncertain, and highly nonlinear [2]. In order to ensure the success of flight missions, the entry guidance algorithm for a hypersonic vehicle necessitates enhanced robustness and autonomy [3]. Therefore, the online real-time trajectory optimization algorithm is particularly necessary to be developed [4]. Nevertheless, it is still a significant challenge to design an optimal or near-optimal guidance strategy for onboard applications with guaranteed stability and real-time performance [5]. In this paper, a novel entry guidance algorithm based on inverse reinforcement learning is proposed to generate optimal or near-optimal control in real-time under complex hypersonic flight environments.

The optimal guidance can be described as a trajectory optimization or an optimal control problem (OCP), which aims to optimize a performance index while satisfying complex constraints. Traditionally, OCP algorithms can be classified into two main types: indirect methods and direct methods [6,7]. Based on Pontryagin's minimum principle, indirect methods transform the OCP problem into a two-point boundary value problem [8].

Numerous indirect methods have been developed to solve OCP problems, offering high-precision solutions [9–11]. However, due to the main drawbacks of indirect methods in convergence difficulty and solving path constraints, direct methods have gained broader application. Direct methods involve transforming OCP problems into finite-dimensional parameter optimization problems through discretization methods, subsequently solved using nonlinear solvers. By combining convex optimization theory and the pseudospectral method, direct methods offer advantages in real-time performance and solution accuracy and have been successfully applied to solving many OCP problems [12–14]. Ref. [15] developed a two-stage trajectory optimization framework using convex optimization and the pseudospectral method to solve the hypersonic vehicle entry problem, improving computational efficiency. Additionally, a Chebyshev pseudospectral method based on differential flatness theory was applied to the hypersonic vehicle entry problem, demonstrating that the guidance algorithm can reduce the solution time for a single trajectory [16]. Unfortunately, modeling the constraints of the trajectory planning problem into a convex format losslessly is difficult work, particularly for hypersonic dynamic systems with highly nonlinear dynamics and constraints. Moreover, the computational cost escalates rapidly with an increase in discrete points, and the number of iterations becomes unpredictable when aiming for a high-precision solution [17,18]. Consequently, for the OCP algorithms, the above shortcomings limit its online application.

In recent years, artificial intelligence (AI)-based guidance algorithms have gained significant attention in the aerospace field, primarily due to their real-time performance and adaptable capabilities. Ref. [19] proposed that these algorithms can be broadly classified into two implementations: supervised learning (SL) and reinforcement learning (RL). In supervised learning, neural networks are trained using extensive datasets of optimal trajectories generated by OCP algorithms. Several SL-based guidance algorithms have been proposed for onboard applications [20]. For instance, Ref. [21] presented a deep neural network (DNN)-based guidance framework for planetary landing, capable of predicting fuel optimal controls from raw images captured by an onboard optical camera. Ref. [22] introduced a DNN-based guidance method for two-degree-of-freedom (2DOF) entry trajectory planning of hypersonic vehicles, and numerical simulations demonstrated its ability to provide stable and real-time control instructions for maximizing terminal velocity. Ref. [23] proposed a real-time DNN-based algorithm to solve the 3DOF entry problem of hypersonic vehicles, and the results showed its capability to generate optimal onboard controls. Similarly, Ref. [24] proposed a DNN-based controller to map optimal control commands from the state, and a hard-ware-in-the-loop (HIL) system was developed to support the real-time performance conclusion of the controller. However, both Ref. [23] and Ref. [24] required generating a large number of datasets before the training process, which was extremely costly in practical applications. Consequently, ensuring the convergence accuracy of existing SL-based algorithms for hypersonic entry problems necessitates constructing a large number of datasets to cover all scenarios, which remains a drawback for these algorithms when the missions are time-sensitive.

On the other hand, reinforcement learning offers an alternative approach that does not rely on existing datasets. RL algorithms continuously update model parameters through interactions with the environment, leading to improved generalization and robustness. RL has also shown promising results in addressing aerospace problems [25,26]. In comparison to traditional guidance algorithms, RL-based guidance algorithms exhibit strong anti-disturbance capabilities and real-time performance [27–30]. Ref. [31] proposed an RL-based adaptive real-time guidance algorithm for the 3DOF entry problem of hypersonic vehicles, and numerical simulation demonstrated that the proposed algorithm achieved a higher terminal success rate compared to the Linear Quadratic Regulator (LQR) method. The convergence of RL-based algorithms heavily relies on the design of the reward function. In the implementation of Ref. [31], dense rewards were provided by tracking a human-designed guidance law, which made it challenging for the model to search for the global

optimal solution. Hence, in order to generate optimal control commands, it is key to design an improved reward function for RL-based algorithms.

In hypersonic entry flight environments, the reward signal is often sparse, meaning that the agent receives a reward only after completing a mission. To address this challenge, a reward shaping function needs to be designed to provide dense rewards throughout the learning process, motivating the agent to learn continuously. A reasonable reward shaping function is hard to complicate manually. Fortunately, the IRL method is one potential solution for solving this problem. The IRL algorithm represents an innovative approach within the realm of the RL method. Diverging from the traditional RL algorithm, the IRL method aims to infer a potential reward function from observed expert examples. Furthermore, IRL can be thought of as an inverse problem where the objective is to understand the motivation behind expert behavior rather than directly learning a policy.

This paper presents a novel guidance algorithm based on Inverse Reinforcement Learning (IRL) to address the guidance problem during the entry phase of hypersonic vehicles. In comparison to other AI-based algorithms and traditional optimal control algorithms explored in previous works, the proposed algorithm's controller can generate optimal actions that meet the requirements of onboard applications using only a few trajectories as datasets. To the best of our knowledge, there have been few studies reported on the generation of optimal actions for hypersonic vehicles via a well-trained DNN-based controller supported by a few trajectories as datasets. Therefore, the concern is attempted to be addressed in this work. In our work, the guidance algorithm is implemented as a policy neural network updated through simulated experience over an interaction of a hypersonic entry simulated environment. In the proposed IRL framework, a customized version of Proximal Policy Optimization Algorithms (PPO) [32] is used to optimize the policy network. In particular, a generative adversarial neural network is designed to distinguish between the agent trajectories and the optimal datasets provided by the optimal control theory, which can effectively address the sparse reward problem while maintaining optimality. It is worth noting that the optimal dataset is only served by a few trajectories. After model optimization, the policy can offer high-frequency closed-loop guidance commands for onboard applications. To fully demonstrate the applicability of the proposed algorithm, numerical simulations are conducted on two typical hypersonic vehicles: the Common Aero Vehicle-Hypersonic (CAV-H) [33] and the Reusable Launch Vehicle (RLV). The two hypersonic vehicles correspond to different flight conditions, which is sufficient to illustrate the generalization of the proposed algorithm.

This paper is structured as follows. Section 2 provides an introduction to the entry problem for hypersonic vehicles, highlighting its characteristics of highly nonlinear dynamics. The Inverse-Reinforcement-Learning-based (IRL-based) guidance method is detailed in Section 3, including the algorithm framework, reward function design, and network structures utilized in the approach. Section 4 verifies the effectiveness and optimality of the proposed algorithm by performing a number of simulations through comparisons with General Pseudo-Spectral Optimal Control Software (GPOPS) [34]. The conclusion of this paper is given in Section 5.

2. Problem Formulation

2.1. The 3DOF Dynamic Model for Hypersonic Entry

The Earth is modeled as a uniform sphere, taking into account the effects of Earth's rotation. During the entry phase of hypersonic vehicles, the control of the vehicle is achieved through the manipulation of the bank angle and attack angle, assuming no flight sideslip angle. The dynamic model for the entry phase is formulated in a 3-degree-of-freedom (3DOF) format, and the parameters of the dynamic model used in this paper are defined within the geocentric fixed coordinate system. These parameters are further elaborated in Equations (1)–(3), and these expressions are derived from Refs. [16,35].

$$\begin{cases} \frac{\mathrm{d}r}{\mathrm{d}t} = v\sin(\gamma) \\ \frac{\mathrm{d}\theta}{\mathrm{d}t} = \frac{v\cos(\gamma)\sin(\psi)}{r\cos(\varphi)} \\ \frac{\mathrm{d}\varphi}{\mathrm{d}t} = \frac{v\cos(\gamma)\cos(\psi)}{r} \\ \frac{\mathrm{d}v}{\mathrm{d}t} = -\frac{D}{m} - g\sin(\gamma) + \Omega_v \\ \frac{\mathrm{d}\gamma}{\mathrm{d}t} = \frac{1}{v}[\frac{L\cos(\sigma)}{m} + (\frac{v^2}{r} - g)\cos(\gamma)] + \Omega_\gamma \\ \frac{\mathrm{d}\psi}{\mathrm{d}t} = \frac{1}{v}[\frac{L\sin(\sigma)}{m\cos(\gamma)} + \frac{v^2}{r}\cos(\gamma)\sin(\psi)\tan(\varphi)] + \Omega_\psi \end{cases} \tag{1}$$

with

$$\begin{aligned} \Omega_v &= \omega^2 r\cos(\varphi)[\sin(\gamma)\cos(\varphi) - \cos(\gamma)\sin(\varphi)\cos(\psi)] \\ \Omega_\gamma &= 2\omega\cos(\varphi)\sin(\psi) + \frac{\omega^2 r\cos(\varphi)}{v}[\cos(\gamma)\cos(\varphi) + \sin(\gamma)\sin(\varphi)\cos(\psi)] \\ \Omega_\psi &= 2\omega[\cos(\varphi)\tan(\gamma)\cos(\psi) - \sin(\varphi) + \frac{\omega^2 r}{v\cos(\gamma)}\sin(\varphi)\cos(\varphi)\sin(\psi)] \end{aligned} \tag{2}$$

where r represents the distance from the center of the earth to the hypersonic vehicle, θ and φ denote the longitude and latitude, respectively. v represents the velocity of the vehicle relative to the earth. γ describes the flight angle of the velocity versus the local horizontal plane angle. ψ is the heading angle. ω represents the angular velocity of the earth's rotation. σ denotes the bank angle. g represents the gravitational acceleration, defined as μ/r^2 where μ is the gravitational constant. L and D represent the aerodynamic lift and the drag, respectively, which can be expressed as:

$$\begin{aligned} L &= \tfrac{1}{2}\rho v^2 S_{ref} C_L \\ D &= \tfrac{1}{2}\rho v^2 S_{ref} C_D \end{aligned} \tag{3}$$

in which the reference area of the vehicle is denoted by S_{ref}, the atmospheric density $\rho = \rho_0 e^{-(r-R_e)/h_s}$ is given by the equation that is the function of altitude and reference sea level density. Here, ρ_0 is the reference density at sea level, R_e is the earth's radius, and h_s is the density scale height. The lift coefficient C_L, and the drag coefficient C_D, are both functions of the attack angle and the Mach number.

2.2. Problem Statement

The paper addresses the trajectory planning problem for hypersonic vehicles, which can be formulated as an optimization control problem. The objective is to generate a sequence of optimal control commands that minimize a given objective function, subject to various constraints, including boundary, path, and control constraints. Using the dynamic model, a typical optimization problem for hypersonic entry vehicles can be defined as follows [35]:

$$\begin{aligned} &\min J \\ &s.t.\ \dot{x} = f(x,u) \\ &x(t_0) = x_0, x(t_f) = x_f \\ &x \in [x_{\min}, x_{\max}] \\ &u_{\min} \le u \le u_{\max} \\ &\dot{Q} = K_Q \rho^{0.5} v^{3.15} \le \dot{Q}_{max} \\ &q = \tfrac{1}{2}\rho v^2 \le q_{max} \\ &n = S_{ref}\frac{q\sqrt{{C_L}^2 + {C_D}^2}}{mg} \le n_{max} \end{aligned} \tag{4}$$

where the objective function is denoted as J, the dynamics system is represented by the equation $\dot{x} = f(x,u)$, x is a six-dimensional state vector given by $[r, \theta, \varphi, v, \gamma, \psi]^T$. Additionally, the heat rate at the stagnation point is denoted as $\dot{Q}$, the dynamic pressure as q, the

overload as n, and K_Q is a constant parameter related to the curvature radius of the vehicle. The initial state is represented as x_0 and the mission target as x_f. The boundary constraints for the states are denoted as $[x_{min}, x_{max}]$. It is important to note that the initial states x_0 are randomly generated to simulate actual flight conditions. The control command vector u is determined by the vehicle model and mission requirements. The minimum and maximum values of the control commands are represented by $[u_{min}, u_{max}]$.

3. Inverse-Reinforcement-Learning-Based Guidance Method

In this section, the proposed framework based on the IRL method is introduced. Our framework describes a novel training process for a DNN model, which achieves high accuracy even with a few optimal trajectories as datasets. The RL problem formulation, reward function design, and the architecture of the neural network are also discussed in this section.

3.1. IRL-Based Guidance Framework

Different from traditional RL algorithms, the IRL method requires the dataset and can enable the learning of a reward shaping function from expert demonstrations, such as optimal trajectories generated by the OCP algorithms, allowing the agent to generalize from limited data and generate dense rewards. By using IRL, the reward shaping function can be learned automatically, relieving the need for manual and complex reward design. When the expert demonstration does not cover the scene, the IRL method can still optimize the agent to learn a decent policy. Several IRL methods have been proposed, such as extracting reward functions using the maximum-margin method [36], using the Gaussian method [37], and using decision trees [38].

In this paper, the Generative Adversarial Imitation Learning (GAIL) algorithm [39] is utilized as a form of the IRL algorithms to train a DNN-based model. The proposed guidance framework based on IRL is designed to achieve effective training of the model using a small number of expert demonstrations. The process of model training is depicted in Figure 1, illustrating the steps involved in training the model.

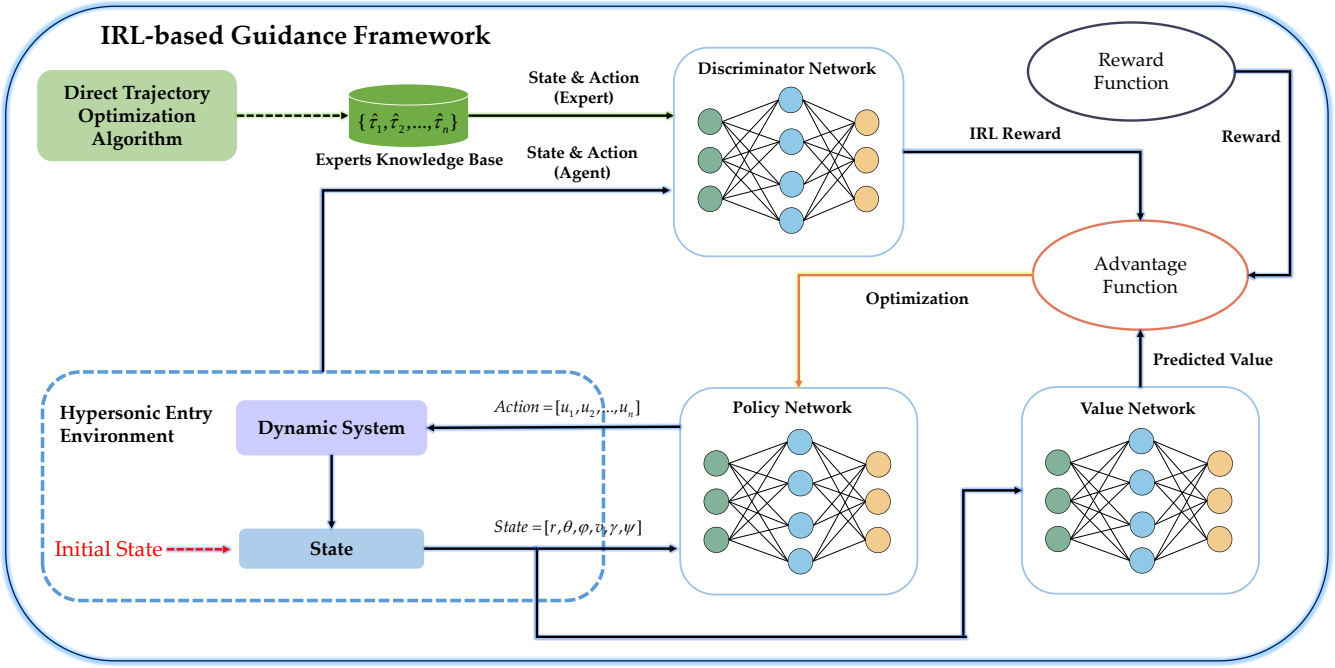

Figure 1. IRL-based guidance framework.

Prior to the model training phase, expert demonstrations are generated using a sophisticated direct trajectory optimization algorithm to solve the hypersonic entry problem. It is important to note that the initial states of the trajectories in the expert demonstrations only cover a limited range. However, the subsequent numerical experiment demonstrates that even with a few trajectories, the proposed algorithm is effective and capable of achieving good guidance.

The proposed IRL-based guidance method incorporates three different neural networks: the Policy Neural Network (Actor), the Value Neural Network (Critic), and the Discriminator Neural Network. The training phase of the IRL-based method proceeds as follows: (1) The policy network interacts with a high-fidelity hypersonic entry environment, as described in Section 2, and generates trajectories online. All initial states are generated randomly and contain all state space for the hypersonic entry problems. (2) State-action pairs from the generated trajectories and the expert demonstration are randomly sampled to train the discriminator network. The goal is to maximize the expert reward while minimizing the agent reward, enabling the discriminator to distinguish between the expert behaviors and the agent behaviors. (3) The advantage function is calculated by combining the IRL reward generated by the discriminator network, the reward computed by the reward function, and the value predicted by the value network. Then, the advantage function is used to optimize the policy network, enabling it to generate improved control commands.

In our IRL-based framework, we utilize the PPO algorithm, which is a popular Advantage Actor-Critic (A2C) method and is widely used in various complex problems. The PPO algorithm is a policy gradient algorithm based on the Trust Region Policy Optimization (TRPO) method. It can dynamically adjust the maximum updated step size by constraining Kullback–Leibler (KL) divergence between the new and old policy. The PPO objective function can be expressed as follows:

$$\max J(\theta) = \mathbb{E}[\min(r_t(\theta), clip(r_t(\theta), 1-\varepsilon, 1+\varepsilon))\, A_{\pi,t}(s,a)] \tag{5}$$

where $r_t(\theta)$ represents the probability ratio $\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$ between the new and old policy. The advantage function is denoted by $A_{\pi,t}(s,a)$, which captures the advantage or benefit of taking action a in state s. The function $clip(x, 1-\varepsilon, 1+\varepsilon)$ is a clipped function that limits the value of x within the range $[1-\varepsilon, 1+\varepsilon]$, where ε is the clipping ratio.

To reduce the computational cost of the model training phase, the Distributed Proximal Policy Optimization (DPPO) algorithm [40] is implemented in this paper, which utilizes a multi-process mechanism to accelerate exploration efficiency. The combination of IRL with DPPO is described in Algorithm 1. According to the recommendations from prior research [25,32], adjusting the clipping ratio and the policy learning rate during the model training phase dynamically can improve performance. The approach is also employed in our work, as shown in line 17 of Algorithm 1, and can be described as follows:

$$\varepsilon = \begin{cases} \min(\varepsilon_{\max}, 1.5\varepsilon) \text{ if kl } < \frac{1}{2}\text{kl}_{targ} \\ \max(\varepsilon_{\min}, \frac{1}{1.5}\varepsilon) \text{ if kl } > 2\text{kl}_{targ} \end{cases} \tag{6}$$

$$\alpha_\theta = \begin{cases} \min(\alpha_{\theta_{max}}, 1.5\alpha_\theta), \text{ if kl } < \frac{1}{2}\text{kl}_{targ} \text{ and } \varepsilon > \frac{1}{2}\varepsilon_{\max} \\ \max(\alpha_{\theta_{min}}, \frac{1}{1.5}\alpha_\theta), \text{ if kl } > 2\text{kl}_{targ} \text{ and } \varepsilon < 2\varepsilon_{\min} \end{cases} \tag{7}$$

Algorithm 1 DPPO-IRL algorithm

Input: Expert trajectories $\tau_E \sim \pi_E$, iteration number I, DPPO worker process number M, buffer size K, discount factor Γ, initial policy, value network, and discriminator parameters $\theta_0, \phi_0, \omega_0$.
Generate the DPPO worker process $[\text{worker}_1, \text{worker}_2, \ldots, \text{worker}_M]$.
for $i = 1, 2, 3, \ldots, I$ **do**
  for $j = 1, 2, \ldots, M$ **do**
    Send the current policy π_{θ_i} to the process worker_j, and run the policy π_{θ_i} for K/M timesteps to collect online trajectories.
  end for
  Sample agent trajectories $\tau_i \sim \pi_{\theta_i}$ and expert trajectories $\tau_E \sim \pi_E$.
  Update the discriminator parameter from ω_i to ω_{i+1} with the gradient:
$$\mathbb{E}_{\tau_{\theta_i}}[\nabla_\omega \log(D_\omega(s,a))] + \mathbb{E}_{\tau_E}[\nabla_\omega \log(1 - D_\omega(s,a)]$$
  The final reward R_t is calculated as the sum of the cost generated by the cost function $\log(1 - D_{\omega_{i+1}}(s,a))$ and the reward is computed by the reward function.
  Discount the reward using the factor Γ and estimate advantages via $A_{\pi,t} = \hat{R}_t - V_{\phi_i}(s_t)$.
  Update the policy parameter from θ_i to θ_{i+1} and value network parameter from ϕ_i to ϕ_{i+1}, using the PPO algorithm.
  Adjust the policy learning rate and clipping ratio according to the approximate KL divergence.
end for

3.2. RL Problem Formulation

An episode in the training process will be terminated prematurely if the range angle increases, indicating that the vehicle has deviated from the target point. Additionally, the episode will also be terminated if the path constraints (such as heat rate, dynamic pressure, or overload) are violated. After a certain number of steps have been accumulated, the policy, value function, and discriminator network are updated once using the IRL-based method. During the model optimization, the observation is represented by a vector given in Equation (8). As mentioned in Equation (9), the action space is defined differently for various problems. For the CAV-H entry problem and the RLV entry problem, the action space consists of [generalized lift coefficient λ, bank angle σ] and [bank angle σ], respectively.

$$\text{obs} = [r\ \theta\ \varphi\ v\ \gamma\ \psi] \tag{8}$$

$$\text{action} = \begin{cases} [\lambda, \sigma] \in \mathbb{R}^2, \text{ when CAV} - \text{H Entry Problem} \\ [\sigma] \in \mathbb{R}^1, \text{ when RLV Entry Problem} \end{cases} \tag{9}$$

The position and velocity of observations used in this paper are normalized before being fed into the model. The definition of normalization is $[\overline{r} = r/R_e, \overline{v} = v/\sqrt{g_0 R_e}]$. Furthermore, each element in the action space is independently normalized to the range $[-1, 1]$ using the Equation (10):

$$u(i)_{norm} = \frac{2 * (u(i) - u(i)_{min})}{u(i)_{max} - u(i)_{min}} - 1, i = \begin{cases} [\lambda, \sigma]\ , \text{ when CAV} - \text{H Entry Problem} \\ [\sigma]\ , \text{ when RLV Entry Problem} \end{cases} \tag{10}$$

3.3. Reward Function Design

In the field of hypersonic entry problems, the issue of sparse reward is a challenge, and designing a reasonable and dense reward function has been a focal point for researchers. However, to the best of our knowledge, there is little literature available on the design of the reward function in the field of hypersonic entry problems. One potential solution is to follow predetermined guidance law. The design of the tracking guidance law resulted in a loss of trajectory optimality. To overcome this limitation, in this paper, a novel reward function is introduced in Equation (11) that combines the discriminator network with several designed terms.

$$\begin{aligned} r &= r_{IRL} + r_{shaping} + r_{penalty} + r_{bonus} + \eta \\ r_{shaping} &= r_{shaping_h} + r_{shaping_{heat}} + r_{shaping_{pressure}} + r_{shaping_{overload}} \\ r_{penalty} &= r_{penalty_\theta} + r_{penalty_\varphi} + r_{penalty_h} + r_{penalty_\gamma} + r_{penalty_\psi} \end{aligned} \tag{11}$$

where the variables mentioned above are defined as follows:

(1) In contrast to other classical RL methods, r_{IRL} is a term generated by the discriminator network at each step, which provides incentives for the agent to learn an optimal policy that aligns with the expert demonstrations.

(2) $r_{shaping}$ is a punishment term for undesired states when the agent approaches the boundary, such as altitude, heat rate, dynamic pressure, and overload. As shown in Equation (12), the value of $r_{shaping}$ is determined by an exponential function, where the punishment increases as the agent gets closer to the boundary. This method enables the agent to quickly learn the solution that does not violate the path constraints.

$$\begin{aligned} r_{shaping_h} &= \begin{cases} \alpha_h \exp\left(-\left\| h - h_{boundary_{min}} \right\| / h_{scale}\right), & \text{if } h > h_{limit_{max}} \\ \beta_h \exp\left(-\left\| h - h_{boundary_{max}} \right\| / h_{scale}\right), & \text{if } h < h_{limit_{min}} \\ 0, & \text{otherwise} \end{cases} \\ r_{shaping_{heat}} &= \begin{cases} \alpha_{heat} \exp\left(-\left\| \dot{Q} - \dot{Q}_{max} \right\| / \dot{Q}_{max}\right), & \text{if } \dot{Q} > \beta_{heat} \dot{Q}_{max} \\ 0, & \text{otherwise} \end{cases} \\ r_{shaping_{pressure}} &= \begin{cases} \alpha_{pressure} \exp\left(-\| q - q_{max} \| / q_{max}\right), & \text{if } q > \beta_{pressure} q_{max} \\ 0, & \text{otherwise} \end{cases} \\ r_{shaping_{overload}} &= \begin{cases} \alpha_{overload} \exp\left(-\| n - n_{max} \| / n_{max}\right), & \text{if } n > \beta_{overload} n_{max} \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{12}$$

(3) To continuously incentivize the agent to improve terminal accuracy, we introduce the term $r_{penalty}$ which measures the accuracy of the terminal state and is only provided at the end of an episode. The specific formulations of $r_{penalty}$ are described as follows:

$$\begin{aligned} r_{penalty_\theta} &= \begin{cases} \zeta_\theta \| \theta - \theta_{target} \|, & \text{if done} \\ 0, & \text{otherwise} \end{cases} \\ r_{penalty_\varphi} &= \begin{cases} \zeta_\varphi \| \varphi - \varphi_{target} \|, & \text{if done} \\ 0, & \text{otherwise} \end{cases} \\ r_{penalty_h} &= \begin{cases} \zeta_h \min\left(\| h - h_{target_{min}} \|, \| h - h_{target_{max}} \|\right), & \text{if done and } h \notin \left[h_{target_{min}}, h_{target_{max}}\right] \\ 0, & \text{otherwise} \end{cases} \\ r_{penalty_\gamma} &= \begin{cases} \zeta_\gamma \min\left(\| \gamma - \gamma_{target_{min}} \|, \| \gamma - \gamma_{target_{max}} \|\right), & \text{if done and } \gamma \notin \left[\gamma_{target_{min}}, \gamma_{target_{max}}\right] \\ 0, & \text{otherwise} \end{cases} \\ r_{penalty_\psi} &= \begin{cases} \zeta_\psi \min\left(\| \psi - \psi_{target_{min}} \|, \| \psi - \psi_{target_{max}} \|\right), & \text{if done and } \psi \notin \left[\psi_{target_{min}}, \psi_{target_{max}}\right] \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{13}$$

(4) As defined in Equation (14), r_{bonus} is a bonus given at the end of an episode if the agent satisfies all terminal constraints and the range angle dx is less than the specified tolerance dx_{lim}.

$$r_{bonus} = \begin{cases} \kappa, \text{ if done and } dx < dx_{\text{lim}} \text{ and } x_f \in [x_{target_{min}}, x_{target_{max}}] \\ 0, \text{ otherwise} \end{cases} \tag{14}$$

(5) η is a positive constant that encourages the agent to continue exploring. This is necessary because the agent might tend to terminate early if all rewards are negative.

3.4. Neural Network Architecture

Two opposing neural networks are required for the IRL-based algorithm, called the generator and the discriminator. In the implementation of the algorithm described in this article, the policy plays the role of the generator, which is composed of a four-layer neural network. The input to the policy is a six-dimensional vector representing the state, and the output is a vector whose dimension depends on the action definition. Each hidden layer of the policy uses a hyperbolic tangent activation function. The value function network estimates the advantage value, which is a one-dimensional value representing the expected advantage of taking a specific action in a given state. The output of the value function network is a single value that represents the estimated advantage value. The discriminator network takes as input a concatenated vector of the observation and action. Figure 2 provides a summary of three network structures.

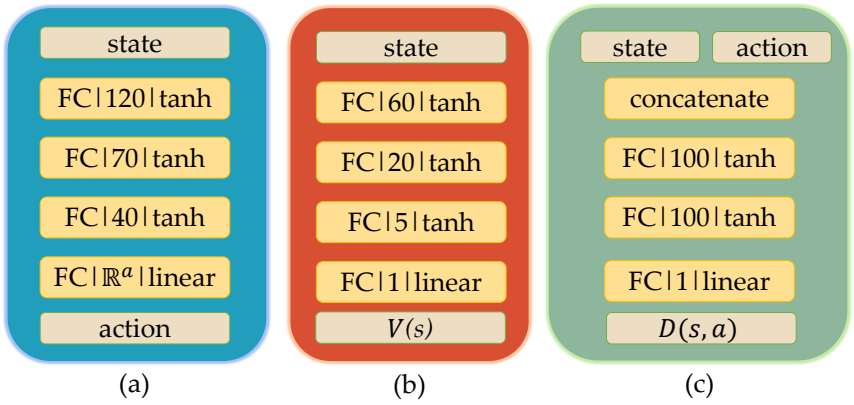

Figure 2. Architecture of the three neural networks. (**a**) Policy network (**b**) Value function network (**c**) Discriminator network.

4. Experiments

First, this section provides an overview of two vehicle models and missions considered in this study. The characteristics and parameters of the hypersonic vehicles are described. Next, the process of generating expert demonstrations and optimizing the model is presented. Furthermore, numerical trajectories of the IRL-based guidance algorithm are given in this section. Additionally, a comparison between the IRL-based algorithm and the GPOPS solver is provided. It is important to note that the model optimization and all numerical experiments were finished on a personal computer with an Intel Core i9-9900 CPU @ 3.10GHz, 16.0 GB RAM, and Windows 10 operating system. The Python 3.7 environment with PyTorch 1.10 was used for implementing the IRL-based algorithm, while the GPOPS software was executed in MATLAB.

4.1. Vehicle Model and Mission

4.1.1. CAV-H Entry Problem

Referring to the article [20], the first vehicle model used in this paper is CAV-H, which exhibits a high lift-drag ratio during hypersonic entry flight. Without loss of generality, the drag coefficient C_D of CAV-H can be assumed to follow the equation of C_L, and the expression for the lift-to-drag ratio can be obtained through a corresponding equation. Assuming the vehicle maintains the maximum lift-to-drag ratio, the lift coefficient and drag coefficient of the vehicle can be defined as follows:

$$\begin{aligned} {C_L}^* &= \sqrt{\frac{C_{D0}}{K}} \\ {C_D}^* &= 2C_{D0} \end{aligned} \tag{15}$$

Therefore, the maximum lift-to-drag ratio coefficient E^* can be expressed as $E^* = C_L^*/{C_D}^* = 1/2\sqrt{K \cdot C_{D0}}$. In this problem, the vehicle always maintains the maxi-

mum lift-to-drag ratio during flight, and the generalized coefficient λ is used as the control command instead of the traditional attack angle. The generalized coefficient λ is defined as $\lambda = C_L/{C_L}^*$. As a result, the lift and drag coefficients can be redefined as follows:

$$\begin{aligned} C_L &= \lambda {C^*}_L \\ C_D &= \frac{{C_L}^*(1+\lambda^2)}{2E^*} \end{aligned} \tag{16}$$

The generalized lift coefficient λ and bank angle σ used as the control command in this CAV-H entry problem are limited within a certain range. The parameters for the CAV-H can be found in Table 1. The initial and terminal states of the vehicle are provided in Table 2. The entry mission is to reach a target location defined by a specific longitude, latitude, and final altitude range. Generally, in hypersonic missions, there are various performance indexes that can be optimized. Due to the CAV-H's classification as a weapon missile, the minimization of flight time is considered imperative. The objective function can be defined as $\min J = t_f$.

Table 1. The parameters of the CAV-H.

Parameter	Value	Parameter	Value
m (kg)	907	$\dot{Q}_{max}$ (kW/m^2)	2000
S_{ref} (m^2)	0.4839	q_{max} (kN/m^2)	300
E^* (-)	3.24	n_{max} (g_0)	3.0
${C_L}^*$ (-)	0.45	K_Q (-)	1.688×10^{-5}
λ_{min} (-)	0	σ_{min} (deg)	-80
λ_{max} (-)	2	σ_{max} (deg)	80

Table 2. Boundary constraints for the CAV-H entry problem.

Boundary \ Item	h (km)	θ (deg)	φ (deg)	v (m/s)	γ (deg)	ψ (deg)
Initial condition	$41 \le h \le 46$	$-0.5 \le \theta \le 0.5$	$-0.5 \le \varphi \le 0.5$	$5300 \le v \le 5500$	$-0.5 \le \gamma \le 0.5$	$89.9 \le \psi \le 90.1$
Terminal condition	$30 \le h \le 40$	39.3	20	–	–	–

4.1.2. RLV Entry Problem

Similar to the assumption in reference [41], an RLV model is used for numerical demonstrations in this work. The RLV is a winged-body vehicle for vertical takeoff and horizontal landing. The trajectory optimization in this paper considers the approximated aerodynamic coefficients regime as described in reference [42], with the following expressions:

$$\begin{aligned} C_L &= 0.0002602\alpha^2 + 0.016292\alpha - 0.041065 \\ C_D &= 0.86495{C_L}^2 - 0.03026C_L + 0.080505 \end{aligned} \tag{17}$$

where α is in degrees and is scheduled based on the velocity profile as given below:

$$\alpha = \begin{cases} 40, & \text{if } v > \ 4570 \text{ m/s} \\ 40 - 0.20705(v - 4570)^2/340^2, & \text{otherwise} \end{cases} \tag{18}$$

Profiles of the angle of attack and aerodynamic coefficients are shown in Figure 3. Consequently, in the RLV entry problem of this paper, the bank angle σ is the only control command, and the rate of the bank angle is limited to 10 deg/s. The parameters for the RLV can be found in Table 3. Similar to the CAV-H entry problem, the parameters of the initial and terminal points are listed in Table 4. The free-flight-time entry is considered in this paper. For the RLV, it is significant to minimize the total heat load during entry.

Therefore, the objective function for the RLV mission is to minimize the total heat load, as expressed as $\min J = \int_{t_0}^{t_f} \dot{Q}\, d_t$.

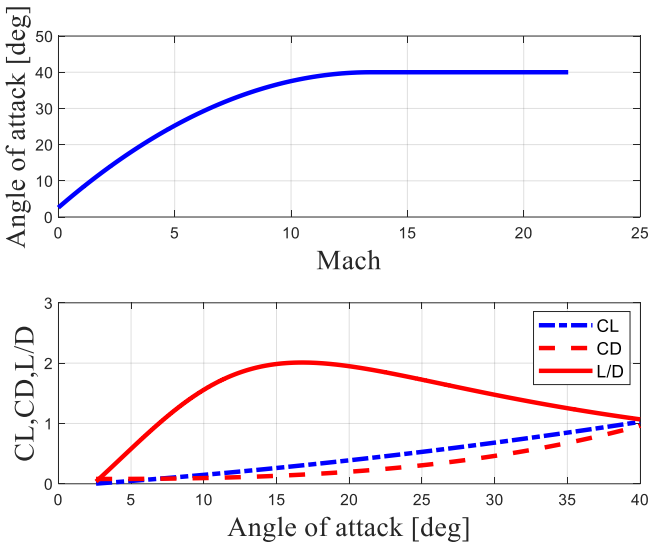

Figure 3. Profiles of angle of attack and aerodynamic coefficients.

Table 3. The parameters of the RLV.

Parameter	Value	Parameter	Value
m (kg)	104,305	$\dot{Q}_{max}$ (kW/m^2)	1800
S_{ref} (m^2)	391.22	q_{max} (kN/m^2)	20
σ_{min} (deg)	-80	n_{max} (g_0)	3.0
σ_{max} (deg)	80	K_Q (-)	1.65×10^{-4}

Table 4. Boundary constraints for the RLV entry problem.

Boundary \ Item	h (km)	θ (deg)	φ (deg)	v (m/s)	γ (deg)	ψ (deg)
Initial condition	$99 \le h \le 101$	$-0.2 \le \theta \le 0.2$	$-0.2 \le \varphi \le 0.2$	7450	-0.5	0
Terminal condition	$20 \le h \le 30$	12	70	–	$-20 \le \gamma \le 0$	$80 \le \psi \le 100$

4.2. Expert Demonstrations Generation Strategy

The GPOPS software, which is based on a pseudospectral method, is used in this paper as the OCP solver to generate the expert demonstrations. The environment parameters used in the simulations and dataset generation are reported in Table 5.

Table 5. Environment parameters.

Parameter	Value
Atmosphere scale height h_s (m)	7500 (CAV-H), 7200 (RLV)
Surface air density ρ_0 (kg/m^3)	1.2 (CAV-H), 1.225 (RLV)
Earth redius R_e (m)	6.378×10^6
Gravitational acceleration at Earth redius g_0 (m/s^2)	9.81

For both the CAV-H and RLV entry problems, 50 trajectories are randomly selected. The profiles of the 50 trajectories for the CAV-H and RLV entry problem are plotted in

Figures 4 and 5, respectively. After the generation of the trajectories, with the aim of augmenting the dataset, the 50 trajectories were linearly interpolated at intervals of step = 1 s. It should be noted that if a well-trained network is optimized using supervised learning methods, the number of samples required would typically be two orders of magnitude larger than the dataset used in this paper [22–24].

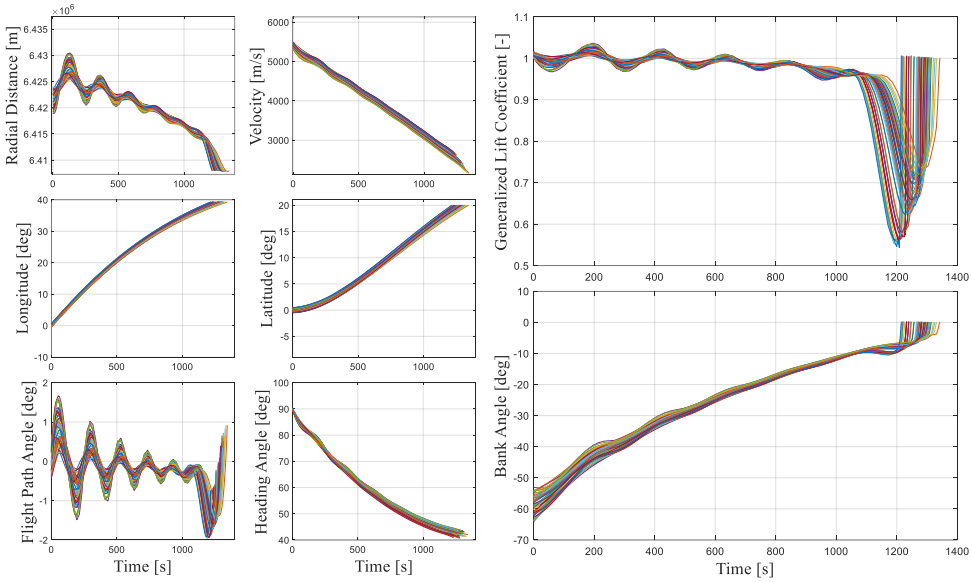

Figure 4. Dataset of the expert demonstrations for the CAV-H entry problem.

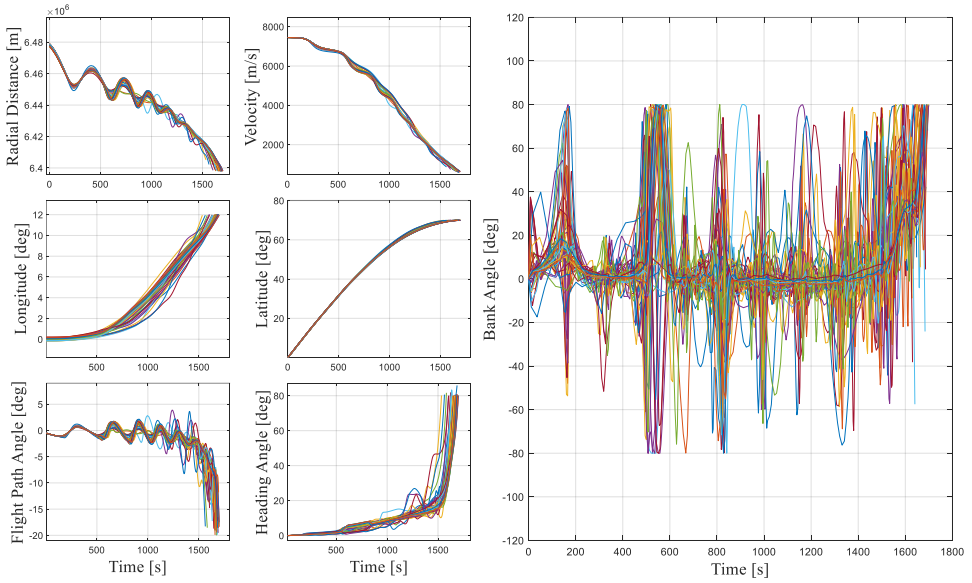

Figure 5. Dataset of the expert demonstrations for the RLV entry problem.

4.3. Model Optimization

The initial state range used in this model optimization can be obtained in Tables 2 and 4. At the beginning of model optimization, the initial learning rates of policy, value function, and discriminator network are set to 0.0002, 0.0025, and 0.001, respectively. All of the hyperparameters during the model optimization are listed in Table 6. These hyperparameters have been elaborately determined based on the mission objectives, constraints, and empirical knowledge, with the aim of rescaling rewards across all components to sensible ranges. During the model optimization, the guidance period is set to 2.5 s, and the integration period is 0.5 s. For both the CAV-H and RLV entry problems, a total of 3000 model iterations are performed, which takes approximately 15 h to complete.

Table 6. Hyperparameters settings.

Parameter	CAV-H Entry Problem	RLV Entry Problem
α_i , $i \in [h, heat, pressure, overload]$	$[-1, -1, -1, -1]$	$[-1, -2.5, -2.5, -5]$
$\beta_i, i \in [h, heat, pressure, overload]$	$[-0.5, 0.98, 0.98, 0.98]$	$[-1, 0.96, 0.96, 0.96]$
$\zeta_i, i \in [\theta, \varphi, h, \gamma, \psi]$	$[-10, -10, -0.1, 0, 0]$	$[-5, -5, -2, -0.5, -0.5]$
$h_i, i \in [boundary_{min}, boundary_{max}, limit_{min}, limit_{max}, scale]$	$[25, 55, 30, 50, 0.1]$	$[20, 120, 20, 120, 0.1]$
κ	150	100
η	0.01	0.01
dx_{lim}	0.25	0.5
ε	0.1	0.1
Γ	0.99	0.99
K	32,768	32,768
M	6	6
I	3000	3000

After applying smoothing with a window size of 5, the reward curves and terminal range angle curves for the CAV-H and RLV entry problems are plotted in Figures 6 and 7, respectively. The left y-axis represents the reward curve, while the right y-axis describes the terminal range angle curve. At the beginning of the model optimization, the agent violated the path constraints, and the terminal range angle was large. As the model optimization progressed, the control commands generated by the agent gradually became similar to the expert demonstrations, leading to a rapid increase in total rewards. Finally, the agent learned how to satisfy all constraints and to continuously receive the terminal bonus. After approximately 1500 epochs of updating, both the policy and the discriminator network reached convergence, indicating that the algorithm had effectively learned the optimal guidance strategy.

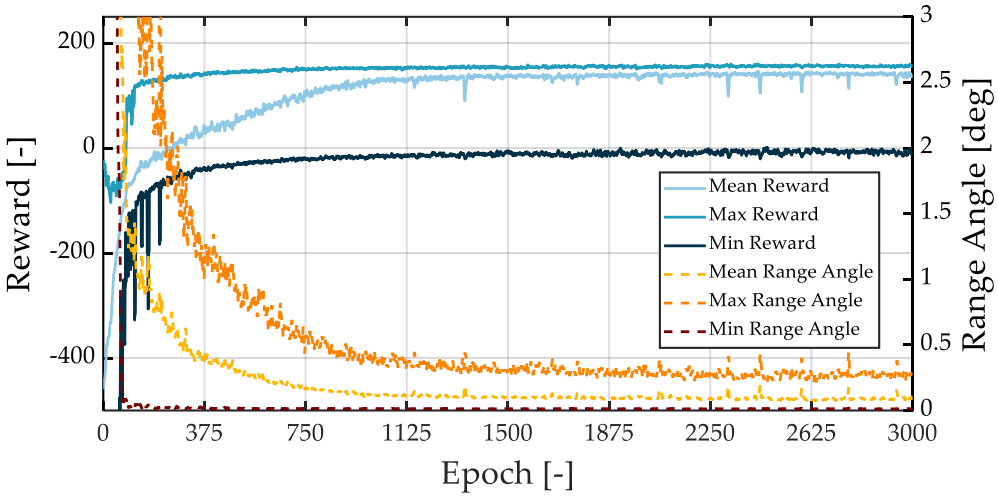

Figure 6. Optimization reward curve and range angle curve for the CAV-H entry problem.

It is important to note that while the total reward curve increased continuously during the model training process, the reward generated by the discriminator network might not followed the same trend. This can be attributed to the limited number of trajectories in the expert demonstrations, which can introduce compounding errors in the control sequence. This observation highlights the advantage of using the IRL-based algorithm compared to supervised learning methods, where a large amount of data is typically required to ensure model generalization. In the case of the RLV entry mission, the reward generated by the

discriminator network exhibited an upward and then downward trend. This phenomenon resulted in a slight overall decrease in the total reward, as evidenced in Figure 7. This indicates that the agent was able to learn a different strategy from the expert demonstrations through the IRL-based algorithm, demonstrating its ability to explore alternative solutions.

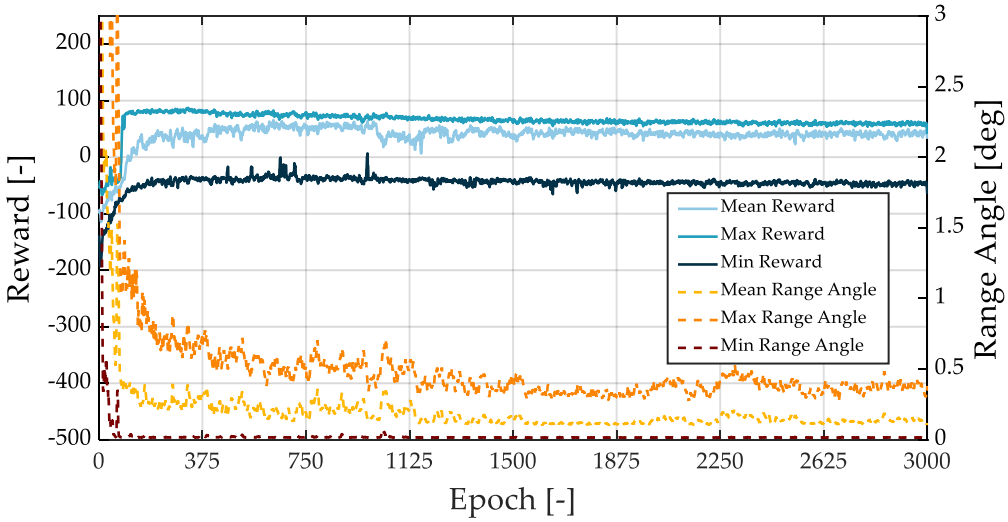

Figure 7. Optimization reward curve and range angle curve for the RLV entry problem.

4.4. Terminal Guidance Accuracy of the IRL-Based Guidance Method

In this subsection, in order to fully evaluate the performance of the IRL-based guidance method, 1000 trajectories are served in numerical simulations. The state variables of the vehicle are randomly initialized, and real-time closed-loop guidance is performed using IRL-based controllers introduced above. The statistics for the terminal state are used to measure the performance of the IRL-based algorithm, which are tabulated in Table 7. The mission is considered successful if the trajectory satisfies all constraints and the terminal range angle error is less than a certain threshold, dx_{lim} degrees. For the CAV-H mission, the threshold is set to 0.25 degrees, while for the RLV mission, it is set to 0.5 degrees due to the greater difficulty of finding a viable solution. The results show that the proposed algorithm achieves a success rate of 99.6% for the CAV-H mission, with the maximum range angle well controlled below 0.27 degrees. Even for the more challenging RLV mission, the success rate is still high at 99.2%. Table 8 provides statistics for the heating rate, dynamic pressure, and overload, demonstrating that all 1000 trajectories generated by the IRL-based method strictly stratify the path constraints. Furthermore, the terminal state distributions of the two vehicles are plotted in Figures 8 and 9, respectively, providing a visual representation of the achieved performance.

Table 7. Terminal Accuracy Statistics.

Parameter	CAV Entry Problem			RLV Entry Problem		
	Min	**Mean**	**Max**	**Min**	**Mean**	**Max**
Range Angle (deg)	0.01	0.07	0.27	0.00	0.17	0.46
Latitude (deg)	39.14	39.31	39.51	11.81	12.05	12.20
Longitude (deg)	19.83	19.99	20.15	69.83	70.14	70.45
Velocity (m/s)	2074.24	2456.57	2755.38	1086.91	1205.92	1407.08
Altitude (km)	30.76	33.38	35.50	28.12	28.83	31.97
Flight Path Angle (deg)		–		−9.04	−5.54	0.03
Heading Angle (deg)		–		81.36	96.05	99.46
Success Rate		99.6%			99.2%	

Table 8. Path Constraint Statistics.

Vehicle Mission	Constraints	μ	σ	Max	Limit
CAV-H Entry	Heating Rate (kW/m^2)	587.57	60.35	725.43	2000
	Dynamic Pressure (kN/m^2)	54.86	10.05	76.34	300
	Overload (g_0)	1.42	0.27	1.98	3
RLV Entry	Heating Rate (kW/m^2)	1436.14	13.84	1459.8	1800
	Dynamic Pressure (kN/m^2)	16.42	0.17	16.59	20
	Overload (g_0)	2.86	0.019	2.88	3

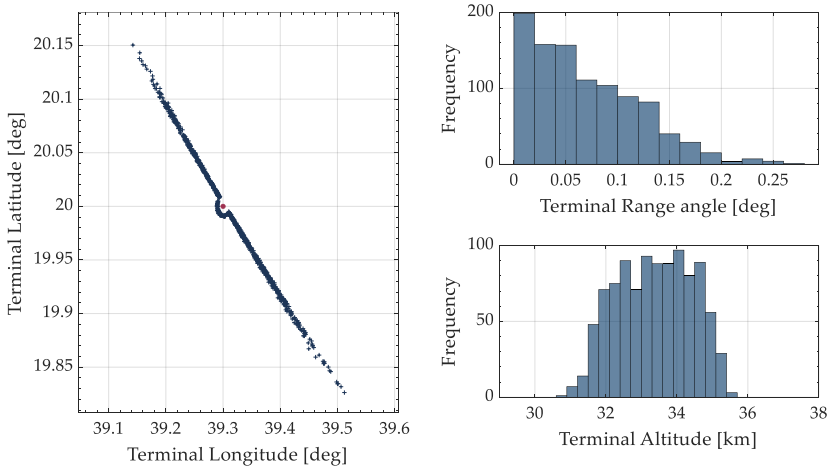

Figure 8. Statistical graph of terminal latitude, longitude, range angle, and altitude for the CAV-H entry problem.

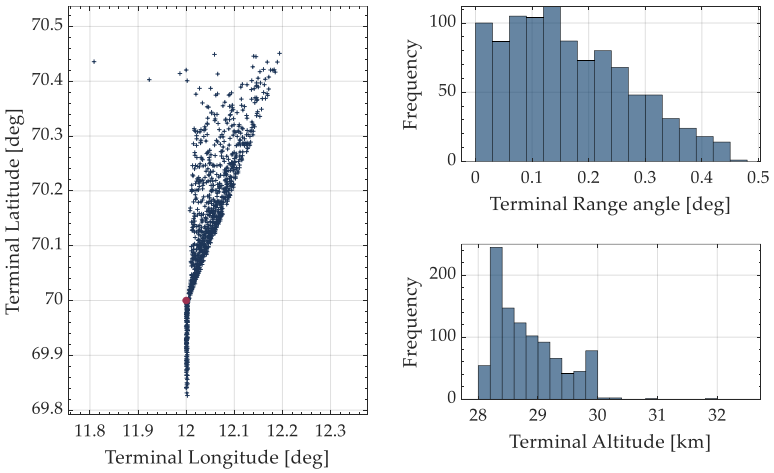

Figure 9. Statistical graph of terminal latitude, longitude, range angle, and altitude for the RLV entry problem.

4.5. Optimality Analysis and Real-Time Performance

As shown in Ref. [35] and Ref. [41], the solutions from GPOPS are typically considered the benchmark for the trajectory optimality. Therefore, in this paper, the IRL-based results are compared to the GPOPS solutions to validate their optimality. With the given initial and terminal conditions, the optimization objective for the CAV-H entry problem is the minimum flight time, and for the RLV entry problem is the minimum total heat load. Figures 10 and 11 show sample trajectories obtained using the IRL-based controller and the GPOPS method. For the CAV-H entry problem, it can be observed that the solutions

obtained from the IRL-based are similar to the results of the GPOPS method. The profiles of the generalized lift coefficient and bank angle also exhibit the same trends. When the vehicle approaches the target point, the generalized lift coefficient of the IRL-based algorithm appears to be smoother compared to the GPOPS method.

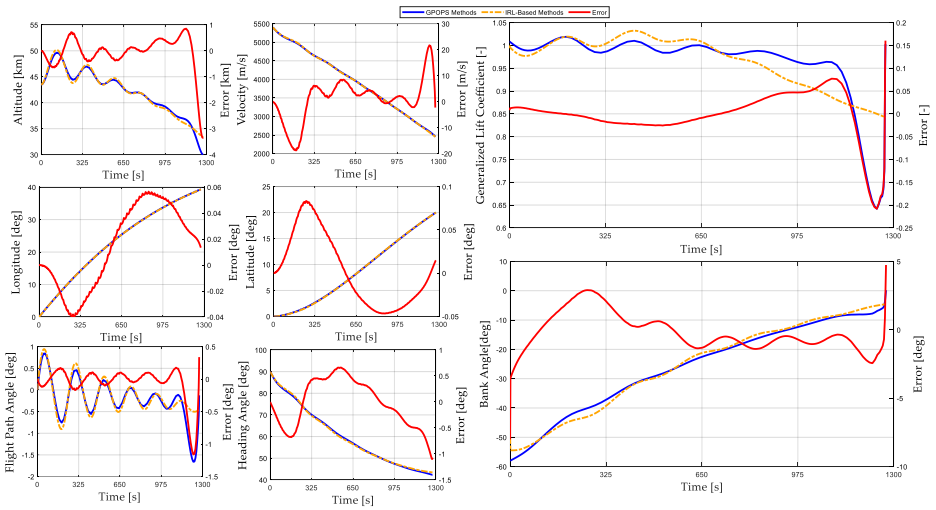

Figure 10. Comparison of sample trajectory for the CAV-H entry problem.

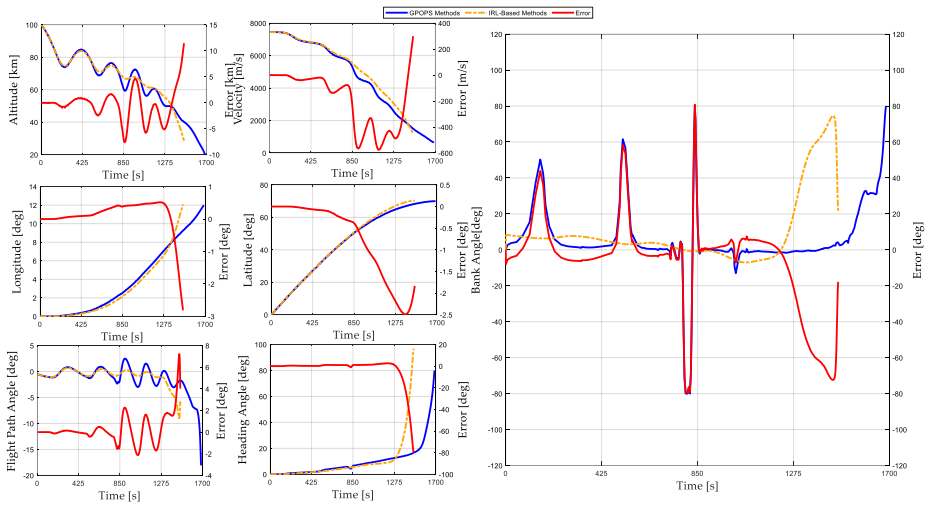

Figure 11. Comparison of sample trajectory for the RLV entry problem.

For the RLV entry problem, the objective function becomes a highly nonlinear integral function, which can make the problem infeasible or difficult to converge. As shown in Figure 5, the control profiles of the expert demonstration for the RLV mission exhibit high-frequency jitter, indicating the complexity of searching for an optimal solution using GPOPS methods. This complexity also brings challenges in the model learning phase, especially when working with a limited number of trajectories in this paper. From Figure 11, the bank angle profiles of the IRL-based method have a similar trend to that of GPOPS, but the control results of the IRL-based method are smoother, which is more conducive to the actual flight environment. One noteworthy item is that in order to reduce the total heat load, the IRL-based method chooses to reach the target point faster, while the GPOPS method tends to decelerate as much as possible. The terminal range angle of the IRL-based method is only 0.1459 degrees, which satisfies the required accuracy for the RLV entry problem.

In order to further analyze the closed-loop guidance effect and optimality of the intelligence controller, 50 trajectories are severed for evaluating the performance and the computational cost of two vehicles. The results of the comparison with the GPOPS method

are presented in Tables 9 and 10, respectively. While the training phase of the IRL-based method requires time, once the model training is completed, the online guidance frequency of the controller is high. The statistics illustrate that the IRL-based method achieves a guidance frequency of 1 s/0.000167 s $\approx$ 5988 HZ, which provides potential for future online closed-loop applications. In general, the Nonlinear Programming solver used in GPOPS is much slower compared to the IRL-based method, and the solution time of GPOPS is unpredictable because good initial guesses are required for convergence. Specially, due to the complexity of the minimum total head load, the calculation time of GPOPS increases to 28 s during the RLV entry environment. In contrast, the CPU time of the IRL-based method is only 0.167 milliseconds, demonstrating the computational advantage. Furthermore, the total heat load of the IRL-based method is only 3% higher than that of GPOPS. The result further demonstrates the optimality of the IRL-based method.

Table 9. Comparison with GPOPS for the CAV-H entry problem.

Method	Mean Flight Time (s)	Mean CPU Time (ms)
IRL-based	1264	0.163
GPOPS	1260	14328

Table 10. Comparison with GPOPS for the RLV entry problem.

Method	Mean Total Heat Load (kw/m^2)	Mean CPU Time (ms)
IRL-based	1,099,644	0.167
GPOPS	1,064,421	28,234

5. Conclusions

In this paper, an Inverse-Reinforcement-Learning-based method for hypersonic entry problems is developed to solve highly nonlinear optimal control problems, where a discriminator network is employed to implicitly capture the optimal reward information associated with expert demonstrations. On this basis, a novel reward function is proposed to address the sparse reward dilemma and provide optimal incentives, which is the main contribution of this paper. The IRL-based method has been validated on two typical hypersonic entry vehicle missions, showcasing its generalization capability. Extensive experiments have demonstrated the effectiveness of the IRL-based method in achieving real-time and high terminal precision with a small dataset. Furthermore, the optimality of the IRL-based method has been demonstrated by numerical solutions through comparison with GPOPS, and the simulation results show that the methodology proposed in this paper is suitable for online optimal guidance and has the potential for onboard implementation in practical applications.

Author Contributions: Conceptualization, J.W. and L.S.; Methodology, J.W. and L.S.; Software, L.S.; Validation, L.S.; Formal analysis, J.W.; Investigation, L.S.; Resources, J.W. and H.C.; Data Curation, L.S.; Writing—Original Draft Preparation, L.S.; Writing—Review and Editing, J.W. and H.C.; Visualization, L.S.; Supervision, J.W.; Project administration, H.C.; Funding acquisition, J.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Basic and Applied Basic Research Project of Guangzhou Science and Technology Bureau, No. 202201011187.

Data Availability Statement: All data used during the study appear in the submitted article.

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

References

1. Li, Z.; Hu, C.; Ding, C.; Liu, G.; He, B. Stochastic gradient particle swarm optimization based entry trajectory rapid planning for hypersonic glide vehicles. *Aerosp. Sci. Technol.* **2018**, *76*, 176–186. [CrossRef]
2. Conway, B.A. A Survey of Methods Available for the Numerical Optimization of Continuous Dynamic Systems. *J. Optim. Theory Appl.* **2012**, *152*, 271–306. [CrossRef]
3. Chai, R.; Tsourdos, A.; Savvaris, A.; Chai, S.; Xia, Y.; Philip Chen, C. Review of advanced guidance and control algorithms for space/aerospace vehicles. *Prog. Aerosp. Sci.* **2021**, *122*, 100696. [CrossRef]
4. Ross, I.M.; Fahroo, F. Issues in the real-time computation of optimal control. *Math. Comput. Model.* **2006**, *43*, 1172–1188. [CrossRef]
5. Wang, Z.P.; Wu, H.N.; Li, H.X. Sampled-Data Fuzzy Control for Nonlinear Coupled Parabolic PDE-ODE Systems. *IEEE Trans. Cybern.* **2017**, *47*, 2603–2615. [CrossRef]
6. Betts, J.T. Survey of Numerical Methods for Trajectory Optimization. *J. Guid. Control Dyn.* **1998**, *21*, 193–207. [CrossRef]
7. von Stryk, O.; Bulirsch, R. Direct and indirect methods for trajectory optimization. *Ann. Oper. Res.* **1992**, *37*, 357–373. [CrossRef]
8. Ozimek, M.T.; Howell, K.C. Low-Thrust Transfers in the Earth-Moon System, Including Applications to Libration Point Orbits. *J. Guid. Control Dyn.* **2010**, *33*, 533–549. [CrossRef]
9. Mansell, J.R.; Grant, M.J. Adaptive Continuation Strategy for Indirect Hypersonic Trajectory Optimization. *J. Spacecr. Rocket.* **2018**, *55*, 818–828. [CrossRef]
10. Grant, M.J.; Braun, R.D. Rapid Indirect Trajectory Optimization for Conceptual Design of Hypersonic Missions. *J. Spacecr. Rocket.* **2015**, *52*, 177–182. [CrossRef]
11. Tang, G.; Jiang, F.; Li, J. Fuel-Optimal Low-Thrust Trajectory Optimization Using Indirect Method and Successive Convex Programming. *IEEE Trans. Aerosp. Electron. Syst.* **2018**, *54*, 2053–2066. [CrossRef]
12. Wang, J.; Li, H.; Chen, H. An Iterative Convex Programming Method for Rocket Landing Trajectory Optimization. *J. Astronaut. Sci.* **2020**, *67*, 1553–1574. [CrossRef]
13. Açıkme¸se, B.; Carson, J.M.; Blackmore, L. Lossless Convexification of Nonconvex Control Bound and Pointing Constraints of the Soft Landing Optimal Control Problem. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 2104–2113. [CrossRef]
14. Wang, J.; Cui, N.; Wei, C. Optimal Rocket Landing Guidance Using Convex Optimization and Model Predictive Control. *J. Guid. Control Dyn.* **2019**, *42*, 1078–1092. [CrossRef]
15. Wang, J.; Cui, N.; Wei, C. Rapid trajectory optimization for hypersonic entry using convex optimization and pseudospectral method. *Aircr. Eng. Aerosp. Technol.* **2019**, *91*, 669–679. [CrossRef]
16. Wang, J.; Liang, H.; Qi, Z.; Ye, D. Mapped Chebyshev pseudospectral methods for optimal trajectory planning of differentially flat hypersonic vehicle systems. *Aerosp. Sci. Technol.* **2019**, *89*, 420–430. [CrossRef]
17. Yang, S.; Cui, T.; Hao, X.; Yu, D. Trajectory optimization for a ramjet-powered vehicle in ascent phase via the Gauss pseudospectral method. *Aerosp. Sci. Technol.* **2017**, *67*, 88–95. [CrossRef]
18. Lekkas, A.M.; Roald, A.L.; Breivik, M. Online Path Planning for Surface Vehicles Exposed to Unknown Ocean Currents Using Pseudospectral Optimal Control. In Proceedings of the 10th IFAC Conference on Control Applications in MarineSystemsCAMS, Trondheim, Norway, 13–16 September 2016; Volume 49, pp. 1–7.
19. Shirobokov, M.; Trofimov, S.; Ovchinnikov, M. Survey of machine learning techniques in spacecraft control design. *Acta Astronaut.* **2021**, *186*, 87–97. [CrossRef]
20. Thuruthel, T.G.; Shih, B.; Laschi, C.; Tolley, M.T. Soft robot perception using embedded soft sensors and recurrent neural networks. *Sci. Robot.* **2019**, *4*, eaav1488. [CrossRef]
21. Furfaro, R.; Bloise, I.; Orlandelli, M.; Di Lizia, P.; Topputo, F.; Linares, R. Deep learning for autonomous lunar landing. In Proceedings of the AAS/AIAA Astrodynamics Specialist Conference, Snowbird, UT, USA, 19–23 August 2018; pp. 3285–3306.
22. Shi, Y.; Wang, Z. Onboard Generation of Optimal Trajectories for Hypersonic Vehicles Using Deep Learning. *J. Spacecr. Rocket.* **2021**, *58*, 400–414. [CrossRef]
23. Wang, J.; Wu, Y.; Liu, M.; Yang, M.; Liang, H. A Real-Time Trajectory Optimization Method for Hypersonic Vehicles Based on a Deep Neural Network. *Aerospace* **2022**, *9*, 188. [CrossRef]
24. Chai, R.; Tsourdos, A.; Savvaris, A.; Xia, Y.; Chai, S. Real-Time Reentry Trajectory Planning of Hypersonic Vehicles: A Two-Step Strategy Incorporating Fuzzy Multiobjective Transcription and Deep Neural Network. *IEEE Trans. Ind. Electron.* **2020**, *67*, 6904–6915. [CrossRef]
25. Deng, T.; Huang, H.; Fang, Y.; Yan, J.; Cheng, H. Reinforcement learning-based missile terminal guidance of maneuvering targets with decoys. *Chin. J. Aeronaut.* **2023**. [CrossRef]
26. Wang, H.; Yang, Z.; Zhou, W.; Li, D. Online scheduling of image satellites based on neural networks and deep reinforcement learning. *Chin. J. Aeronaut.* **2019**, *32*, 1011–1019. [CrossRef]
27. Gaudet, B.; Linares, R.; Furfaro, R. Deep reinforcement learning for six degree-of-freedom planetary landing. *Adv. Space Res.* **2020**, *65*, 1723–1741. [CrossRef]
28. Xu, X.; Chen, Y.; Bai, C. Deep Reinforcement Learning-Based Accurate Control of Planetary Soft Landing. *Sensors* **2021**, *21*, 8161. [CrossRef] [PubMed]
29. Li, S.; Yan, Y.; Qiao, H.; Guan, X.; Li, X. Reinforcement Learning for Computational Guidance of Launch Vehicle Upper Stage. *Int. J. Aerosp. Eng.* **2022**, *2022*, 2935929. [CrossRef]

30. Furfaro, R.; Scorsoglio, A.; Linares, R.; Massari, M. Adaptive generalized ZEM-ZEV feedback guidance for planetary landing via a deep reinforcement learning approach. *Acta Astronaut.* **2020**, *171*, 156–171. [CrossRef]
31. Gaudet, B.; Drozd, K.; Furfaro, R. Adaptive Approach Phase Guidance for a Hypersonic Glider via Reinforcement Meta Learning. In Proceedings of the AIAA SCITECH 2022 Forum, San Diego, CA, USA, 3–7 January 2022.
32. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
33. Richie, G. The Common Aero Vehicle—Space delivery system of the future. In Proceedings of the Space Technology Conference and Exposition, Albuquerque, NM, USA, 28–30 September 1999.
34. Patterson, M.A.; Rao, A.V. GPOPS-II: A MATLAB Software for Solving Multiple-Phase Optimal Control Problems Using HpAdaptive Gaussian Quadrature Collocation Methods and Sparse Nonlinear Programming. *ACM Trans. Math. Softw.* **2014**, *41*, 1–37. [CrossRef]
35. Wang, Z.; Grant, M.J. Constrained Trajectory Optimization for Planetary Entry via Sequential Convex Programming. *J. Guid. Control Dyn.* **2017**, *40*, 2603–2615. [CrossRef]
36. Ng, A.Y.; Russell, S.J. Algorithms for Inverse Reinforcement Learning. In Proceedings of the Seventeenth International Conference on Machine Learning, ICML '00, San Francisco, CA, USA, 29 June–2 July 2000; pp. 663–670.
37. Levine, S.; Popovic, Z.; Koltun, V. Nonlinear inverse reinforcement learning with gaussian processes. *Adv. Neural Inf. Process. Syst.* **2011**, *24*, 19–27.
38. Bagnell, J.; Chestnutt, J.; Bradley, D.; Ratliff, N. Boosting Structured Prediction for Imitation Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Schölkopf, B., Platt, J., Hoffman, T., Eds.; MIT Press: Cambridge, MA, USA, 2006; Volume 19.
39. Ho, J.; Ermon, S. Generative Adversarial Imitation Learning. In *Proceedings of the Advances in Neural Information Processing Systems*; Schölkopf, B., Platt, J., Hoffman, T., Eds.; MIT Press: Cambridge, MA, USA, 2016; Volume 29.
40. Heess, N.; TB, D.; Sriram, S.; Lemmon, J.; Merel, J.; Wayne, G.; Tassa, Y.; Erez, T.; Wang, Z.; Eslami, S.M.A.; et al. Emergence of locomotion behaviours in rich environments. *arXiv* **2017**, arXiv:1707.02286.
41. Wang, Z.; Lu, Y. Improved Sequential Convex Programming Algorithms for Entry Trajectory Optimization. *J. Spacecr. Rocket.* **2020**, *57*, 1373–1386. [CrossRef]
42. Lu, P. Entry Guidance and Trajectory Control for Reusable Launch Vehicle. *J. Guid. Control Dyn.* **1997**, *20*, 143–149. [CrossRef]