mathematics

MDPI

Article

Research on Intelligent Control Method of Launch Vehicle Landing Based on Deep Reinforcement Learning

Shuai Xue [1], Hongyang Bai [1,*], Daxiang Zhao [2] and Junyan Zhou [2]

[1] School of Energy and Power Engineering, Nanjing University of Science and Technology, Nanjing 210094, China; xueshuai72@163.com
[2] School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China; zhaodaxiang@njust.edu.cn (D.Z.); zhoujunyan@njust.edu.cn (J.Z.)
* Correspondence: hongyang@njust.edu.cn

Abstract: A launch vehicle needs to adapt to a complex flight environment during flight, and traditional guidance and control algorithms can hardly deal with multi-factor uncertainties due to the high dependency on control models. To solve this problem, this paper designs a new intelligent flight control method for a rocket based on the deep reinforcement learning algorithm driven by knowledge and data. In this process, the Markov decision process of the rocket landing section is established by designing a reinforcement function with consideration of the combination effect on the return of the terminal constraint of the launch vehicle and the cumulative return of the flight process of the rocket. Meanwhile, to improve the training speed of the landing process of the launch vehicle and to enhance the generalization ability of the model, the strategic neural network model is obtained and trained via the form of a long short-term memory (LSTM) network combined with a full connection layer as a landing guidance strategy network. The proximal policy optimization (PPO) is the training algorithm of reinforcement learning network parameters combined with behavioral cloning (BC) as the reinforcement learning pre-training imitation learning algorithm. Notably, the rocket-borne environment is transplanted to the Nvidia Jetson TX2 embedded platform for the comparative testing and verification of this intelligent model, which is then used to generate real-time control commands for guiding the actual flying and landing process of the rocket. Further, comparisons of the results obtained from convex landing optimization and the proposed method in this work are performed to prove the effectiveness of this proposed method. The simulation results show that the intelligent control method in this work can meet the landing accuracy requirements of the launch vehicle with a fast convergence speed of 84 steps, and the decision time is only 2.5 ms. Additionally, it has the ability of online autonomous decision making as deployed on the embedded platform.

Keywords: launch vehicle; landing phase; deep reinforcement learning; LSTM; imitation learning; embedded platform

MSC: 68T50; 68T07; 68T20; 68T35; 68T37

Citation: Xue, S.; Bai, H.; Zhao, D.; Zhou, J. Research on Intelligent Control Method of Launch Vehicle Landing Based on Deep Reinforcement Learning. *Mathematics* **2023**, *11*, 4276. https://doi.org/10.3390/math11204276

Academic Editors: Shuo Yu and Feng Xia

Received: 8 September 2023
Revised: 9 October 2023
Accepted: 10 October 2023
Published: 13 October 2023

1. Introduction

A launch vehicle is the main tool for exploring the vast sky and developing and utilizing space [1] and is a leader in promoting the development of space technology and industry. The demand for space transportation, the fulfillment of which with the current launch capacity (using expendable vehicles) seems unrealistic, has continuously increased [2]. For launch vehicles, it seems that reusability is the obvious path to reducing costs [3]. Therefore, a reusable launch vehicle (RLV) is an attractive option for cost- and time-effective space transportation. To this effect, the recovery and reuse of flight hardware has been a multidisciplinary problem being tackled by scientists, engineers and academics all over the world [4]. As the nerve center of the launch vehicle [5], the guidance and control

system directly affects the flight quality and comprehensive performance of the launch vehicle and determines the success of the space mission [6]. The guidance and control theory and technology involved are the hot and difficult points in the field of aircraft control nowadays, and they are cutting-edge, basic and comprehensive. Guidance and control technology has become one of the core key areas supporting the future development of China's space industry [7].

Traditional guidance and control technology of launch vehicles relies heavily on models, but it is inevitably difficult to obtain accurate control models due to factors such as insufficient wind tunnel tests and incoherence between space and Earth. In order to reduce development costs and improve the intelligence level and flight quality of launch vehicles, launch vehicles with self-learning abilities need to be developed [8]. In the literature [9], the multi-stage pseudo-spectral convex optimization method is used to transform the nonlinear optimal control problem of rocket landing guidance into a convex optimization problem that is easy to solve by convexating indexes and constraints. However, it is difficult to convexate the rocket landing trajectory with strong nonlinear and many non-convex factors, and the improved computing power is also weak in real time. The literature [10] puts forward two guidance optimization algorithms; one is single convex optimization, and the other is continuous convex optimization. The rocket flight task can be completed by using these two methods, and the landing accuracy can meet the requirements. However, the rocket model established by using them is a three-free dynamic model, which is relatively simple.

With the rapid development of artificial intelligence technology and its application in the field of autonomous control technology, the use of advanced artificial intelligence control technology can continue to improve the active adaptation and autonomous decision-making ability of launch vehicles [8]. In recent years, the rapid development of deep reinforcement learning (DRL) has made it possible to solve control problems that are difficult to overcome using traditional control methods. In 2016, the AlphaGo agent with a DRL algorithm as one of the core technologies defeated the top human professional go player Lee Sedol in the Go match, making the DRL algorithm widely recognized and deeply studied by the research community [11]. In recent years, DRL has been widely promoted and researched in many fields such as robotics, intelligent driving and electronic design [12–14]. In the literature [15], a reinforcement learning algorithm was used to design a staged reward function based on terminal constraint and fuel consumption index to train the rocket landing guidance process, and a guidance strategy with the ability to generalize the model deviation was obtained. However, the attitude change of the rocket was not considered in the flight process, and the algorithm was not optimized, so the training required a large number of steps. The adaptability of the network structure to complex environments is poor. One study [16] proposed a hierarchical reinforcement learning structure and an improved incentive function to drive punishment and designed a new missile guidance strategy, but its method was not tested on the missile-borne platform, and only the theoretical results were verified by simulation. Another study [17] proposed a new method for optimizing dynamic descent guidance, which was based on convex optimization to solve the problem of a feasible landing trajectory on Mars without reaching the target. However, the verification of its online real-time performance was insufficient. Ref. [18] proposed an HP pseudo-spectral homotopy convex optimization online trajectory planning algorithm with an optimal terminal time estimation strategy, which transformed the optimal control problem into a parameter optimization problem, and designed an optimal terminal time fast estimation strategy. However, when the initial simulation scenario was set, the rocket only moved on two coordinate axes, resulting in reduced disturbances and deviations and insufficient adaptability. Ref. [19] used a Gaussian pseudo-spectrum method to generate reference trajectories and designed a launch vehicle trajectory tracking method based on neural adaptive dynamic programming. Ref. [20] used a sparse online Gaussian process adaptive augmentation for incremental backstepping flight control. Ref. [21] used integrated guidance and control simulation to study the mechanics of the reusable retrograde flight of slender low-lift drag bodies and analyzed their performance through

aerodynamic loads and heat flux. Ref. [22] studied an online guidance algorithm based on convex optimization, which transformed the guidance problem of the rocket recovery stage into a second-order cone optimization problem. However, it customized the problem and had weak generalization ability. Ref. [23] proposed two online attitude adjustment planning methods: "three channel program angle online attitude adjustment planning" and "program quaternion online attitude adjustment planning", but their adaptive ability was insufficient.

In this paper, the deep reinforcement learning method is used to train the launch vehicle landing neural network. On the one hand, the reward function is designed through the combination of terminal constraint return and process cumulative return, and the strategy network uses an LSTM neural network to improve the adaptive ability of the rocket to the flight process environment so that the obtained strategy can meet the landing accuracy of the rocket. On the other hand, the optimization results of the classical optimal control method are used as an expert demonstration, and the training speed is improved by combining imitation learning and reinforcement learning so that the proposed method can meet the real-time requirements. In order to prove its online control ability, the method is tested on an embedded missile-borne platform to verify the performance of the proposed intelligent control method.

In summary, the intelligent control algorithm designed in this paper not only satisfies the landing accuracy of the launch vehicle and the calculation speed of online control instructions but also speeds up the convergence speed of traditional algorithms, improves the generalization ability of the network and enhances the adaptive ability of the launch vehicle under uncertain factors such as strong wind interference so that the landing of the launch vehicle does not depend on the establishment of an accurate model.

2. Launch Vehicle Landing Model

To construct the intensive learning training environment for rocket landing, it is first necessary to establish the mathematical model of the launch vehicle and define the space rectangular coordinate system as follows:

The inertial coordinate system is defined as follows: the rocket landing point O is taken as the coordinate origin, the coordinate origin of the inertial system is firmly connected with the landing point O, the OY axis points to the firing direction in the landing point horizontal plane, the OX axis is perpendicular to the horizontal plane and points upward, and the OZ axis is perpendicular to the XOY plane and constitutes the right-hand system, as shown in Figure 1.

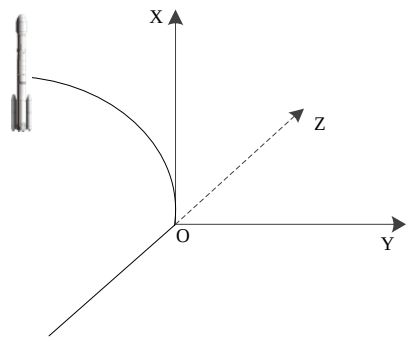

Figure 1. Inertial coordinate system.

The arrow body coordinate system is defined as follows: taking O_1, the center of mass of the rocket, as the coordinate origin, the O_1Y_1 axis is along the arrow body axis, pointing to the rocket head; the O_1X_1 axis is in the longitudinal symmetric plane of the rocket, perpendicular to the O_1Y_1 axis, pointing upward; and the O_1Z_1 axis is perpendicular to the longitudinal symmetric plane. From the tail of the arrow body, the O_1Z_1 axis points to

the left side of the arrow body, and the three axes form the right-hand system, as shown in Figure 2.

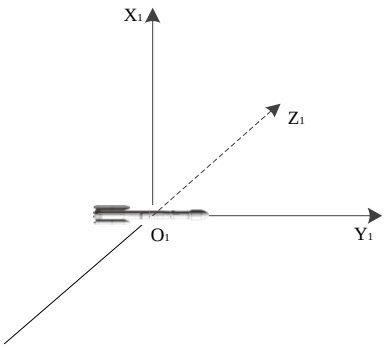

Figure 2. Arrow body coordinate system.

The kinetic equation of the rocket's center of mass in an inertial coordinate system is as follows:

$$\begin{cases} \dot{\boldsymbol{r}} = \boldsymbol{v} \\ \dot{\boldsymbol{v}} = \boldsymbol{g} + \frac{(\boldsymbol{P}+\boldsymbol{D})}{m} \\ \dot{m} = -\parallel \boldsymbol{P} \parallel / (I_{sp} g_0) \\ \boldsymbol{D} = -\frac{1}{2}\rho \parallel \boldsymbol{v} \parallel S_{ref} C_D Ma \boldsymbol{v} \end{cases} \tag{1}$$

where $\boldsymbol{r}$ is the position vector; $\boldsymbol{v}$ is the velocity vector; m is the mass of the rocket; $\boldsymbol{g}$ is the gravitational acceleration vector; $\boldsymbol{P}$ is the engine thrust vector; $\boldsymbol{D}$ is the aerodynamic drag vector; I_{sp} represents the fuel specific impulse; g_0 represents the average gravitational acceleration of the Earth at sea level; ρ is the atmospheric density; S_{ref} is the reference cross-sectional area of the rocket; C_D is the drag coefficient; and Ma is the Mach number.

The control quantity is engine thrust $\boldsymbol{P}$, and the amplitude meets the constraints

$$P_{min} \leqslant \parallel \boldsymbol{P} \parallel \leqslant P_{max} \tag{2}$$

The dynamics equation of the rocket around the center of mass in the arrow body coordinate system is as follows:

$$\begin{bmatrix} \dot{\omega}_x \\ \dot{\omega}_y \\ \dot{\omega}_z \end{bmatrix} = J^{-1} \left(\begin{bmatrix} 0 \\ M_{sty} \\ M_{stz} \end{bmatrix} + \begin{bmatrix} M_{dx} \\ M_{dy} \\ M_{dz} \end{bmatrix} + \begin{bmatrix} M_{cx} \\ M_{cy} \\ M_{cz} \end{bmatrix} - \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix} J \begin{bmatrix} \omega_x \\ \omega_y \\ \omega_z \end{bmatrix} \right) \tag{3}$$

where $\dot{\omega}_x$, $\dot{\omega}_y$ and $\dot{\omega}_z$ are the components of the angular velocity of the rocket in the three axes of the arrow body coordinate system; J is the moment of inertia vector; ω_x, ω_y and ω_z are the components of the angular velocity of the rocket in the three axes of the arrow body coordinate system; and M_{stx}, M_{sty}, M_{stz}, M_{dx}, M_{dy}, M_{dz}, M_{cx}, M_{cy} and M_{cz} are the components of the aerodynamic stabilization torque, aerodynamic damping torque and control torque acting on the rocket in the three axes of the arrow body coordinate system, respectively.

The kinematics equations of the rocket in quaternion form in the arrow body coordinate system are as follows:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{4}$$

where q_0, q_1, q_2 and q_3 are the quaternions of the rocket.

The transformation relationship between quaternion and the Euler angle is as follows:

$$\begin{cases} \gamma = \arctan\frac{2(q_0q_1+q_2q_3)}{{q_0}^2-{q_1}^2-{q_2}^2+{q_3}^2} \\ \varphi = \arcsin(2(q_0q_2-q_1q_3)) \\ \psi = \arctan\frac{2(q_0q_3+q_1q_2)}{{q_0}^2+{q_1}^2-{q_2}^2-{q_3}^2} \end{cases} \tag{5}$$

where γ is the rolling angle, φ is the pitch angle, and ψ is the yaw angle.

3. Launch Vehicle Landing Markov Decision Process

The Markov decision process (MDP) model corresponding to the rocket landing process can be described by a quintuple (S, A, T, R, γ) where S is the state space; A is the action space; T is the state transition probability; R is the reward function; and γ is a discount factor [24].

3.1. Design of State Space and Action Space

The state space is

$$S = \left[\boldsymbol{r},\ \boldsymbol{v},\ q_0,\ q_1,\ q_2,\ q_3,\ \omega_x,\ \omega_y,\ \omega_z, m\right]^T \tag{6}$$

where $\boldsymbol{r}$ is the position vector of the rocket; $\boldsymbol{v}$ is the velocity vector of the rocket; m is the mass of the rocket; q_0, q_1, q_2 and q_3 are the rocket quaternions; and ω_x, ω_y and ω_z are the components of the rocket rotation angular velocity on the three axes of the arrow body coordinate system.

The action space is

$$A = \left[\delta_y, \delta_z, \parallel \boldsymbol{P} \parallel\right]^T \tag{7}$$

where δ_y, δ_z and $\boldsymbol{P}$ are shown in Figure 3; δ_y is the angle between the thrust direction vector projection and Y axis in the YZ plane; and δ_z is the angle between the thrust direction vector projection and thrust in the YZ plane. The thrust of the engine is $\parallel \boldsymbol{P} \parallel$ in the same direction as the vertical axis of the rocket.

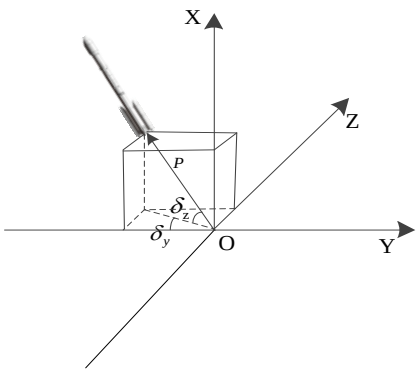

Figure 3. Thrust and its angle diagram.

3.2. State Transition Probability

The state transition probability represents the probability of transferring to the state set S_{t+1} of $t+1$ the next time when the launch vehicle is in the state set S_t and action set A_t of the current time t, and the deterministic state transition probability of the launch vehicle landing guidance MDP is established as follows:

$$P(S_{t+1}|S_t, A_t) = 1 \tag{8}$$

3.3. Reward Function Design

In order to improve the adaptive ability of the rocket in landing, the design of the reward function not only considers the return of the terminal constraint of the launch vehicle but also combines the cumulative return of the process in flight.

In order to control the landing trajectory of the rocket from deviating from the target direction, the target acceleration is set. When the rocket acceleration is closer to the target acceleration, the reward value is greater. In the course of the flight, the thrust should be as small as possible to reduce fuel consumption, and the greater the thrust, the smaller the corresponding reward value; as the attitude control error decreases, the reward value will gradually increase. Therefore, the process cumulative return R_1 is designed as follows.

According to Equation (1), the carrier rocket undergoes engine thrust, aerodynamic drag, and gravity during landing. So, the optimal target acceleration can be expressed by combining it with kinematic formulas:

$$\boldsymbol{a_{targ}} = \boldsymbol{a}' + \boldsymbol{g} = -\left(2\boldsymbol{r}/{t_{go}}^2 + 2\boldsymbol{v}_0/t_{go} + \boldsymbol{g}\right) \tag{9}$$

where $\boldsymbol{a}'$ represents the acceleration generated by the thrust vector and aerodynamic drag vector; $\boldsymbol{v}_0$ is the initial velocity of the carrier rocket. t_{go} is the remaining flight time, which can be expressed as $t_{go} = \|\boldsymbol{r}\|/\|\boldsymbol{v}\|$.

Then, the reward function R_1 is

$$\begin{cases} R_1 = -0.01 \times \left\|\boldsymbol{a} - \boldsymbol{a_{targ}}\right\| - 10^{-8} \times \|\boldsymbol{P}\| - 10 \times |\varphi| + 2 \\ \boldsymbol{a} = \boldsymbol{P}/m \\ \boldsymbol{a_{targ}} = -2\boldsymbol{r}/{t_{go}}^2 - 2\boldsymbol{v}_0/t_{go} - \boldsymbol{g} \\ t_{go} = \|\boldsymbol{r}\|/\|\boldsymbol{v}\| \end{cases} \tag{10}$$

where $\boldsymbol{a}$ is the acceleration, $\boldsymbol{a_{targ}}$ is the target acceleration, and φ is the attitude angle.

In order to make the terminal velocity of the rocket landing and the terminal height accuracy of the terminal position 0 and to limit the terminal angular velocity and attitude angle of the rocket, the terminal reward returns R_2 is designed as follows:

$$\begin{cases} R_2 = R + R_r + R_v \\ R = 10 \times (x + v + r_{targ} + \omega + \varphi) \\ R_r = 100 - r_{targ} \\ R_v = 5 \times (100 - v) \end{cases} \tag{11}$$

where R is the terminal state reward, R_r is the terminal position reward, R_v is the terminal speed reward, x is the rocket flight height, and r_{targ} is the rocket landing radius.

In order to reduce useless training, the training is terminated, and then a penalty is given as the following equation when the rocket attitude angle exceeds the limit of the attitude angle during flight:

$$R_3 = \begin{cases} -50, & |\varphi| > \delta \\ 0, & other \end{cases} \tag{12}$$

where δ represents the attitude angle limitation during the landing.

Then, the total reward is expressed as

$$reward = R_1 + R_2 + R_3 \tag{13}$$

4. Deep Reinforcement Learning Algorithm

Deep reinforcement learning [25] is a decision algorithm based on a deep learning model. It combines the perception ability of deep learning and the decision ability of reinforcement learning to realize an end-to-end perception and control system, which has strong universality.

4.1. PPO Algorithm

PPO algorithm [26] is a new deep reinforcement learning algorithm proposed by scholar Schulman, which can be applied in a continuous state and action space. PPO uses the importance sampling principle to update the strategy, combining importance sampling with the actor-critic framework. Its agent consists of two parts; one is the actor, responsible for interacting with the environment to collect samples, and the other is the critic, responsible for judging the actor's actions. The PPO gradient update formula can be used to update the actor:

$$J_{ppo}^{\theta'}(\theta) = \hat{E}_t[min(\frac{p_\theta(a_t|s_t)}{p_{\theta'}(a_t|s_t)}A^\theta(s_t,a_t), clip(\frac{p_\theta(a_t|s_t)}{p(a_t|s_t)}, 1-\varepsilon, 1+\varepsilon))A^{\theta'}(s_t,a_t)] \tag{14}$$

where θ is the policy parameter; $\hat{E}_t$ refers to the empirical expectation of the time step; $p_\theta(a_t|s_t)$ refers to the state transition probability of the actor network that needs training; $p_{\theta'}(a_t|s_t)$ refers to the state transition probability of the old actor network; ε is a hyperparameter, usually with a value of 0.1 or 0.2; and A_t is the estimated advantage of the time step t. The advantage function is calculated as follows:

$$A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \cdots + (\gamma\lambda)^{T-t+1}\delta_{T-1} \tag{15}$$

where $\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$, $V(s_t)$ is the state value function calculated by the critic network at the time t, and r_t is the reward value at the time t. The loss of the critic is calculated as follows:

$$L_c = \left(r + \gamma\left(max\left(Q\left(s',a'\right)\right)\right) - Q(s,a)\right) \tag{16}$$

where $\gamma(max(Q(s',a')))$ is the target value function, $Q(s,a)$ is the predicted value, and s and a are the state and action, respectively.

4.2. Improve PPO Algorithm

4.2.1. LTSM Network

LSTM is an improved recurrent neural network [27], which uses a memory unit called LSTM to determine what information should be retained and to control the transmission of information from one moment to the next. It is the most widely used network with a memory function at present.

Since the relationship between the carrier rocket and the environment during flight has the characteristics of a time series, its landing process is not only related to the state of the current moment but also related to historical motion information. On the premise of fully recognizing the environment, the rocket needs to learn enough motion before and after relations to improve its generalization ability and prediction ability. Therefore, using the fully connected neural network in the PPO algorithm to approximate the policy function and the value function can not meet the needs of complexity. In this paper, the strategy network and value network use an LSTM network architecture. First, the LSTM network is introduced to extract features from complex flight states, output useful perceptual information and enhance the learning ability of sequence sample data. Then, the policy function and value function are approximated by a fully connected neural network.

The LSTM-actor network structure design is shown in Figure 4. In the input layer, 14 nodes are set, corresponding to 14 state quantities of the launch vehicle; the hidden layer is set with an LSTM network layer and full connection layer; the LSTM network layer is set with 3 network units; and the full connection layer is designed with 3 layers. The output layer has three nodes, corresponding to the three motion quantities of the launch vehicle, and Relu is used as the activation function.

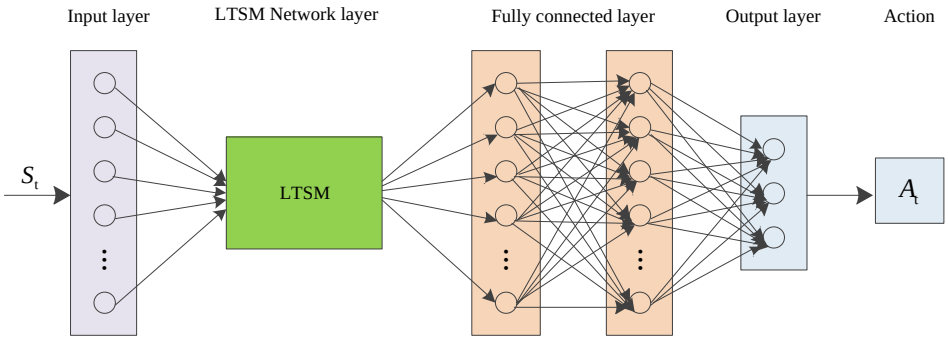

Figure 4. LSTM-actor network structure.

The network structure design of LSTM-critic is shown in Figure 5, in which 17 nodes are set in the input layer, corresponding to 14 state quantities of the launch vehicle and 3 action quantities generated by the current strategy network, 3 network units are set in the LSTM network layer in the hidden layer, 3 layers are set in the fully connected layer, and 1 node is set in the output layer, corresponding to the state value function. Relu is used as the activation function.

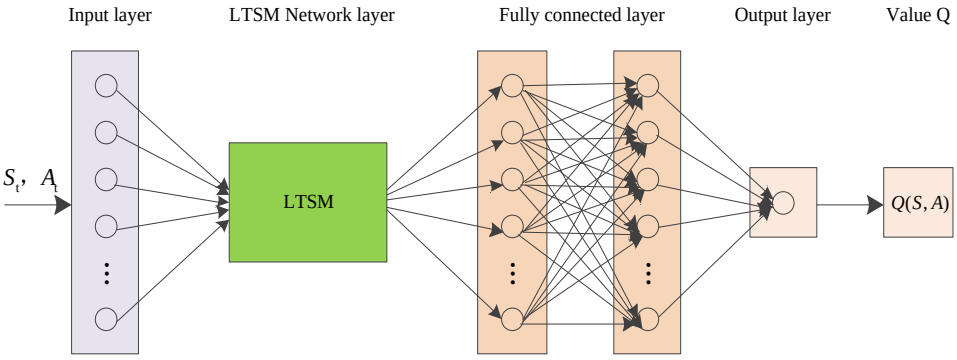

Figure 5. Network structure of LSTM-critic.

4.2.2. Imitation Learning

In this paper, the commonly used behavioral cloning (BC) is chosen for the imitative learning algorithm. Behavior cloning is an algorithm that directly learns agent strategies by applying supervised learning to demonstration sets [28]. The expert data set in this paper obtains the motion control instructions and corresponding state information of launch vehicle landing guidance through the traditional convex optimization algorithm and sets a demonstration set

$$D = \{(s_1, a_1), (s_2, a_2), \ldots (s_n, a_n)\} \tag{17}$$

where s_i is the state and a_i is the action demonstrated by the expert. Let the agent's policy be π, and the agent's policy can be obtained by minimizing the following objective function:

$$min \sum_{i}^{n} (\pi(s_i) - a_i)^2 \tag{18}$$

4.2.3. Overall Algorithm Framework

In order to improve the training speed of the launch vehicle landing process and the generalization ability of the model, the landing guidance strategy network in this paper is in the form of an LSTM network combined with a full connection layer, PPO is used as the reinforcement learning network parameter training algorithm, BC is used as the reinforcement learning pre-training imitation learning algorithm, and the strategic neural network model is trained. Figure 6 shows the overall framework of the rocket landing process based on imitation learning combined with reinforcement algorithms.

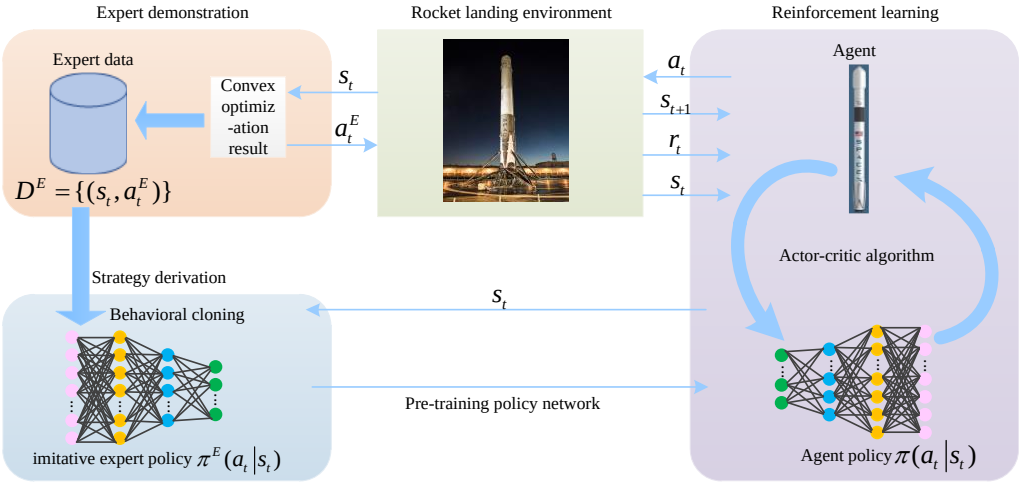

Figure 6. Overall framework.

According to the overall framework shown in Figure 6, the main flow of the algorithm designed in this paper is shown in Figure 7. Through this flow, the landing control of the launch vehicle is trained, and the control strategy network is finally obtained to guide the landing flight of the launch vehicle.

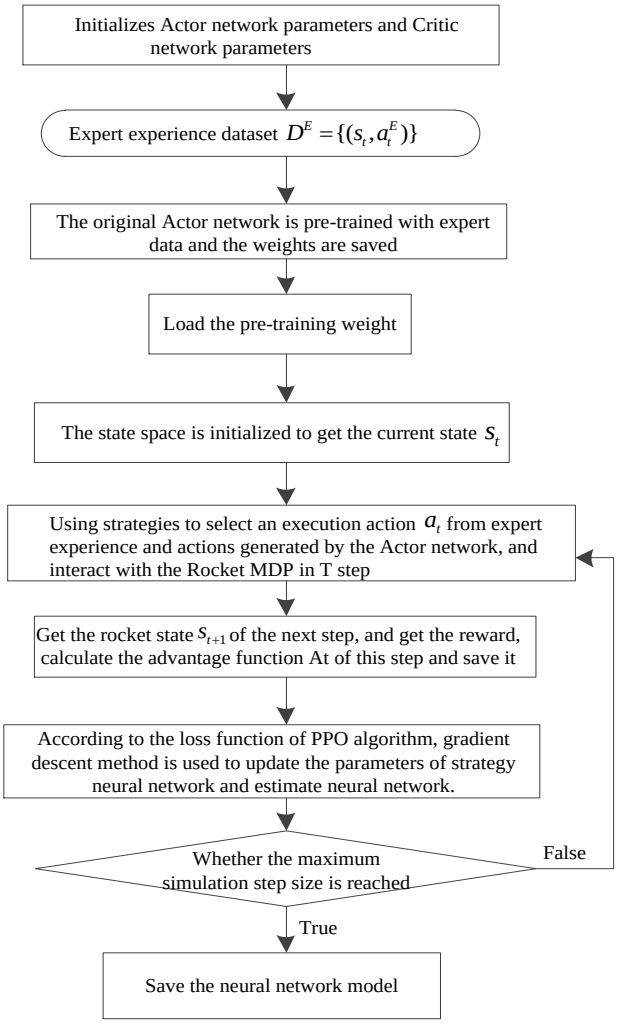

Figure 7. Algorithm flow chart.

In the calculation process of this algorithm, based on the landing characteristics of the launch vehicle, 14 important motion parameters were selected as the state space of the algorithm and 3 control parameters were selected as the action space. In each interaction between the launch vehicle and the environment, actions were taken according to a certain strategy based on the current state of the environment, and corresponding returns were obtained. In the training process of carrier rocket landing, for n state variables and m action variables, if all possible future states of each state variable are considered, it is n^2, and then it is mn^2 in each iteration process. Therefore, the time complexity of the algorithm in this paper is:

$$T(n) = O(mn^2) = 14 \times 3^2 = 126 \tag{19}$$

5. Simulation Verification and Analysis

5.1. Simulation Scenario Settings

In this paper, a simulation scenario is set to verify the algorithm. In the simulation scenario, the initial state of the rocket, the position of the landing point and the main parameters of the rocket are set, as shown in Table 1. During the simulation process, the stroke field is set as 1 times the standard wind field. The time step is 0.1 s, and the maximum simulation step per turn is 150 s. The maximum thrust of the rocket is 981 kN. In order to verify the generalization of the model, the initial state parameters $\boldsymbol{r}$, $\boldsymbol{v}$, m and C_D of the rocket were simulated again with 10% deviation.

Table 1. Initial rocket parameters table.

Rocket Parameter	Value
Length	40 m
Diameter	3.66 m
Quality	50 t
Specific impulse I_{sp}	360 s
Atmospheric density ρ	1.225 kg/m^3
Acceleration of gravity g_0	9.8 m/s^2
Aerodynamic drag coefficient C_D	0.82
Initial position (x, y, z)	(2000 m, −1600 m, 50 m)
Initial velocity (vx, vy, vz)	(−90 m/s, 180 m/s, 0)
Landing site location (x, y, z)	(0, 0, 0)
Attitude angle limitation during landing (ψ, φ, γ)	$[85^\circ, 85^\circ, 360^\circ]$
Terminal position constraint	$r \leq 20.0$ m
Terminal velocity constraint	$v \leq 10$ m/s
Terminal attitude deviation constraint (ψ, φ, γ)	$[10^\circ, 10^\circ, 360^\circ]$

5.2. Simulation Parameter Settings

In this paper, the algorithm program is programmed in the Python language, the agent environment is built by Gym, and the deep reinforcement learning training environment is built based on pytorch. The test host is based on a Windows 10 operating system, and the hardware configuration is Intel Core I7 9700K, RTX 3080Ti and 16 GB RAM. The algorithm hyperparameter settings are shown in Table 2. In addition, the environment was transplanted to the Nvidia Jetson TX2 embedded development board, and the test was carried out on a missile-borne platform under the condition that other parameters were the same, and the test results were compared with those of the Windows system.

Table 2. Algorithm hyperparameter settings.

Parameter	Value
PPO algorithm learning rate	0.0003
BC algorithm learning rate	0.0003
Maximum number of training steps	2048
Training lot size	64
Reward discount rate	0.99
Estimate the clipping coefficient of the advantage function	0.2
Generalized dominance estimation parameters	0.95

5.3. Analysis of Simulation Results

For the training of the launch vehicle landing strategy network, we first used the imitation learning algorithm of behavior cloning to train it. Then, the PPO algorithm was used to train in the same environment. Then, the PPO algorithm was improved, the LTSM network was added to its neural network, and the PPO_LSTM algorithm was used to train it. Finally, the algorithm was further improved, imitation learning was introduced on the basis of the PPO_LTSM algorithm, and the Imitation + LTSM_PPO algorithm in this paper was obtained to train the launch vehicle landing strategy network. The training results of the four algorithms were compared, and their respective reward function convergence curves are shown in Figure 8.

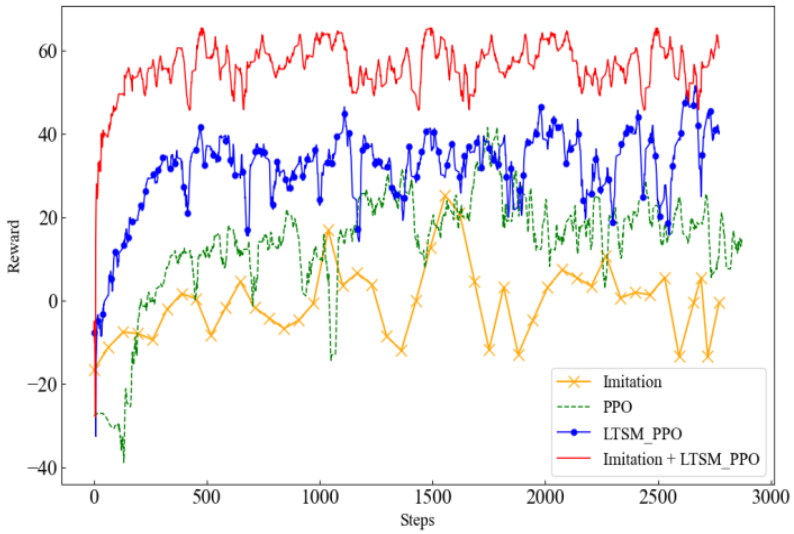

Figure 8. Change of reward function.

As can be seen from Figure 8, for the same simulation environment, when the number of training steps is the same, the reward function of the BC algorithm is not converging, and the maximum reward learned is 25. The convergence steps of the PPO algorithm, PPO_LSTM algorithm and Imitation + LTSM_PPO algorithm are, respectively, 1142, 323 and 84, and the maximum rewards are 41, 51 and 66. It can be seen from the results that the convergence speed of the Imitation + LTSM_PPO algorithm is improved by 92.6% and 74% compared with the original PPO algorithm, and the maximum reward explored by the proposed algorithm is also greater than that of the other three algorithms. Therefore, using the Imitation + LTSM_PPO algorithm can achieve the maximum reward explored by the launch vehicle, the algorithm convergence speed is also significantly accelerated, and the convergence curve of the reward function is relatively stable.

5.4. Simulation Verification

After the training of the Imitation + LTSM_PPO algorithm, the strategy network parameters are fixed, and the simulation and verification are carried out by interaction with the rocket landing simulation environment. Meanwhile, in order to verify the terminal position, velocity accuracy, attitude change and fuel consumption of the landing strategy proposed in this paper, a convex optimization method is adopted to conduct a simulation under the same initial conditions. In order to verify the generalization of the model, the simulation is compared under the condition of 10% deviation in the initial condition. In addition, in order to test its real-time performance on the missile-borne platform, the simulation environment built in this paper is transplanted to the Nvidia Jetson TX2 embedded platform for verification. The verification results of the above four cases are compared, as shown in the figure below.

Figures 9–14 show the state change curve of the rocket during landing, and Figures 15–17 show the action change curve of the rocket during landing. Among them, Figure 9 is the trajectory curve of the rocket, from which it can be seen that the rocket successfully reached the landing target point, and the convergence of the terminal position, velocity and attitude was basically achieved, as shown in Figures 10–14. Figure 15 shows the magnitude change of the rocket thrust. Figures 16 and 17 show the direction change of the rocket thrust.

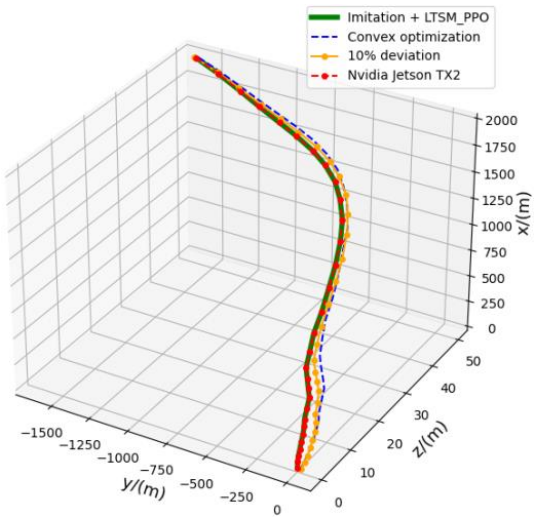

Figure 9. Trajectory curve of rocket motion.

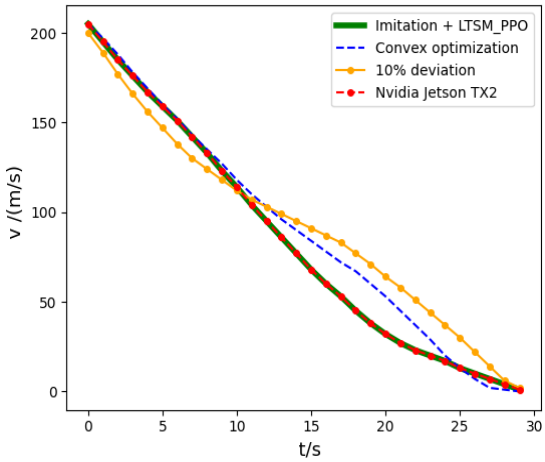

Figure 10. The velocity change of the rocket.

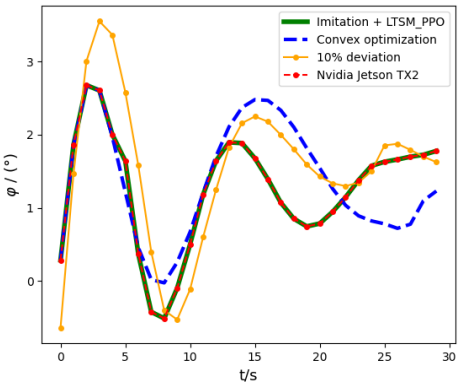

Figure 11. Variation of pitch angle deviation of rocket.

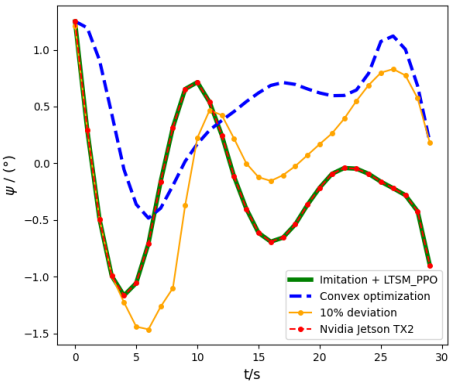

Figure 12. Variation of yaw angle deviation of rocket.

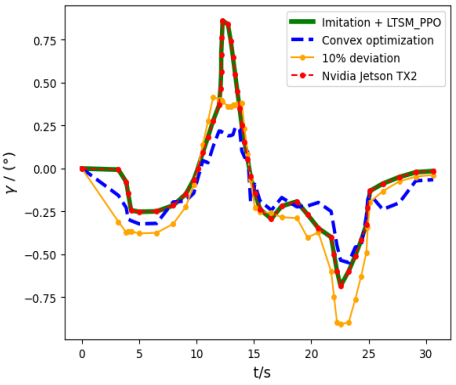

Figure 13. Variation of roll angle deviation of rocket.

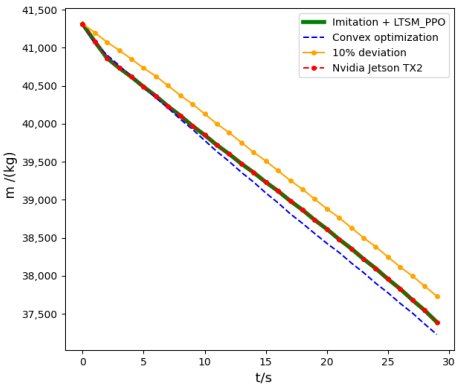

Figure 14. Mass change of the rocket.

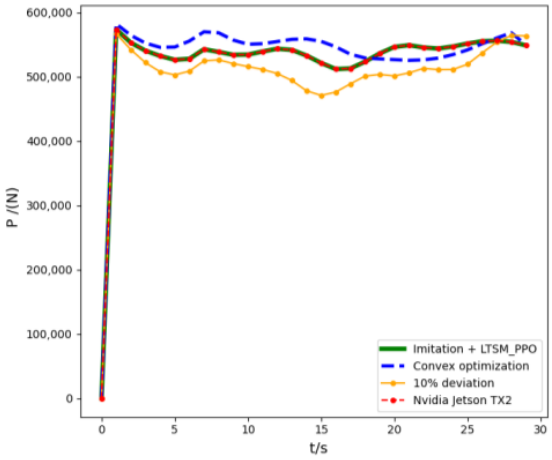

Figure 15. The thrust magnitude of the rocket changes.

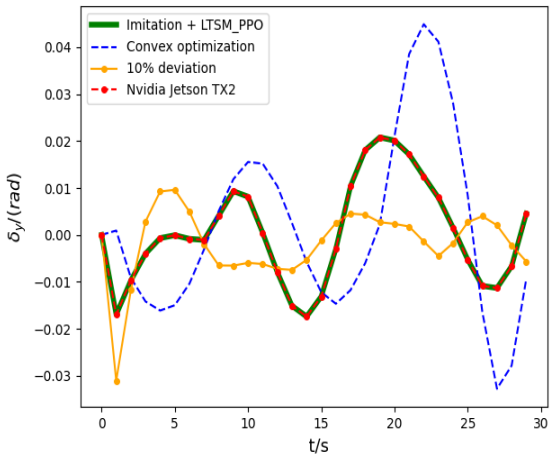

Figure 16. Thrust angle change of the rocket.

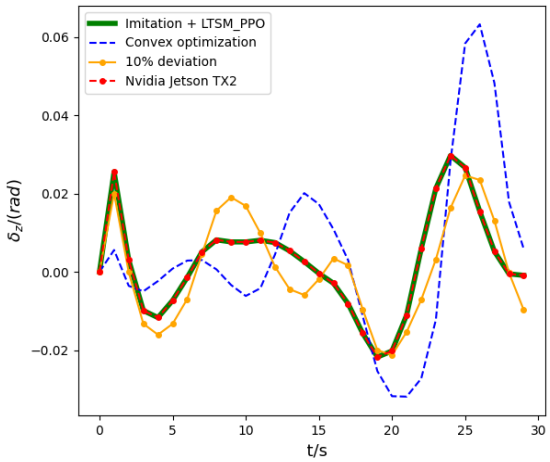

Figure 17. Thrust angle change of the rocket.

According to the curves in Figures 9–17, simulation results and calculation speed in the above four cases were compared, as shown in Table 3.

Table 3. Comparison of results.

Name	Terminal Position Accuracy/(m)	Terminal Speed Accuracy/(m/s)	Terminal Attitude Deviation $(\varphi,\psi,\gamma)/(^\circ)$	Fuel Consumption/(kg)	Calculation Speed/(ms)
Imitation + LTSM_PPO	8.4	4.9	(−1.6, −0.9, 0.009)	3820.0	2.5
Convex optimization	11.9	12.9	(−1, 0.25, 0.003)	3970.4	243
10% deviation	10.0	7.5	(−1.5, 0.25, 0.008)	3827.1	7.4
Nvidia Jetson TX2	8.4	4.9	(−1.6, −0.9, 0.009)	3820.0	9.0

As can be seen from Table 3, under the same simulation conditions, the terminal landing accuracy of the intelligent control method used in this paper to guide the rocket landing flight is higher than that of the traditional convex optimization method, with the average accuracy increased by 3%, the calculation speed increased by 98% and the fuel consumption deviation of the two methods being 4%, meeting the requirements of the launch vehicle landing flight. At the same time, by comparing the thrust amplitude curve and thrust angle curve given in Figures 15–17, it can be seen that the thrust amplitude and angle of the intelligent control strategy conform to the changing law.

In the case of a 10% deviation of the model using the method presented in this paper, the terminal accuracy is as follows: at 10.0 m/s and 7.5 m/s, the terminal attitude angle deviation is small, the fuel consumption is 3827.1 kg, and the thrust size and angle changes are consistent with the law. The results show that the intelligent control method based on deep reinforcement learning does not rely on accurate modeling and has a certain generalization ability when the model is biased.

Comparing the simulation curve on the Nvidia Jetson TX2 embedded platform with the proposed method, it can be seen that the changes in the state parameter curve and action parameter curve are basically consistent, the landing accuracy and fuel consumption are exactly the same, and the calculation speed of control instructions is 9 ms, which can meet the real-time requirements. It is verified that the proposed method can be deployed on the missile-borne platform and has online autonomous learning ability.

6. Conclusions

In order to solve the problem that a traditional guidance and control algorithm is not able to deal with multi-factor uncertainty during launch vehicle landing, an intelligent control method based on deep reinforcement learning is designed in this paper. By designing the rocket landing Markov decision process and using an improved deep reinforcement learning algorithm, the LTSM network and imitation learning are combined with a reinforcement learning algorithm to improve the algorithm's speed and robustness. Through training, a neural network model with generalization ability is obtained. After setting up the simulation environment of the landing section of the launch vehicle, the intelligent control algorithm proposed in this paper is verified, the results show that the intelligent control method can meet the accuracy requirements of the rocket landing terminal, and the time of generating guidance and control instructions is fast, meeting the real-time requirements. In order to test its performance on the missile-borne platform, this paper sets up its environment on the Nvidia Jetson TX2 embedded platform for simulation under the same conditions. The results show that the guidance and control command output of the algorithm on the missile-borne platform can meet the requirements of rapidness, has online control capability and realizes fast autonomous decision making. It enhances the autonomy and adaptability of rockets in typical scenarios such as uncertain flight environments, uncertain fault handling, uncertain disturbance handling and uncertainty in one's own model. For example, when a rocket encounters strong winds during landing, it will generate huge disturbances. At this time, intelligent methods can be controlled through adaptive control without relying on precise modeling. This paper provides a model and methodological foundation for the intelligence of launch vehicles, improves their ability

to handle uncertain factors, reduces the dependence on precise modeling and has certain engineering application value.

Author Contributions: Conceptualization, S.X.; methodology, S.X. and H.B.; software, S.X. and D.Z.; validation, J.Z.; formal analysis, H.B.; investigation, H.B.; resources, H.B.; data curation, H.B.; writing—original draft preparation, S.X.; writing—review and editing, S.X.; visualization, S.X.; supervision, H.B.; project administration, S.X. and H.B.; funding acquisition, H.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the National Natural Science Foundation of China through Grant No. U21B2028.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy.

Acknowledgments: Shuai Xue thanks Hongyang Bai for the helpful guide.

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

References

1. Wu, X.F.; Peng, Q.B.; Zhang, H.L. Analysis and reflection on the development history of manned launch vehicles at Home and abroad. *Manned Spacefl.* **2019**, *26*, 783–793.
2. Jo, B.U.; Ahn, J. Optimal staging of reusable launch vehicles for minimum life cycle cost. *Aerosp. Sci. Technol.* **2022**, *127*, 107703. [CrossRef]
3. Jones, H.W. The recent large reduction in space launch cost. In Proceedings of the 48th International Conference on Environmental Systems, Albuquerque, NM, USA, 8–12 July 2018.
4. Mukundan, V.; Maity, A.; Shashi Ranjan Kumar, S.R.; Rajeev, U.P. Terminal Phase Descent Trajectory Optimization of Reusable Launch Vehicle. *IFAC-PapersOnLine* **2022**, *55*, 37–42. [CrossRef]
5. Song, Z.; Pan, H.; Wang, C.; Gong, Q. Development of flight control technology for Long March launch vehicle. *J. Astronaut.* **2020**, *41*, 868–879.
6. Wei, C.; Ju, X.; He, F.; Pan, H.; Xu, S. Adaptive augmented control of active segment of launch vehicle. *J. Astronaut.* **2019**, *40*, 918–927.
7. Ma, W.; Yu, C.; Lu, K.; Liu, J.; Si, W.; Li, W. Guidance and Control Technology of "Learning" launch vehicle. *Aerosp. Control* **2019**, *38*, 3–8.
8. Zhang, H.P.; Lu, K.F.; Cao, Y.T. Application status and development Prospect of Artificial Intelligence Technology in "Learning" launch vehicle. *China Aerosp.* **2021**, 8–13. [CrossRef]
9. Hwang, J.; Ahn, J. Integrated Optimal Guidance for Reentry and Landing of a Rocket Using Multi-Phase Pseudo-Spectral Convex Optimization. *Int. J. Aeronaut. Space Sci.* **2022**, *23*, 766–774. [CrossRef]
10. Botelho, A.; Martinez, M.; Recupero, C.; Fabrizi, A.; De Zaiacomo, G. Design of the landing guidance for the retro-propulsive vertical landing of a reusable rocket stage. *CEAS Space J.* **2022**, *14*, 551–564. [CrossRef]
11. Silver, D.; Huang, A.; Maddison, C.J.; Guez, A.; Sifre, L.; van den Driessche, G.; Schrittwieser, J.; Antonoglou, I.; Panneershelvam, V.; Lanctot, M.; et al. Mastering the game of Go with deep neural networks and tree search. *Nature* **2016**, *529*, 484–489. [CrossRef]
12. Vasquez-Jalpa, C.; Nakano-Miyatake, M.; Escamilla-Hernandez, E. A deep reinforcement learning algorithm based on modified Twin delay DDPG method for robotic applications. In Proceedings of the 2021 21st International Conference on Control, Automation and Systems (ICCAS), Jeju, Republic of Korea, 12–15 October 2021; pp. 743–748. [CrossRef]
13. Duan, J.; Eben Li, S.; Guan, Y.; Sun, Q.; Cheng, B. Hierarchical reinforcement learning for self-driving decision-making without reliance on labelled driving data. *IET Intell. Transp. Syst.* **2020**, *14*, 297–305. [CrossRef]
14. Liu, J.; Chen, Z.-X.; Dong, W.-H.; Wang, X.; Shi, J.; Teng, H.-L.; Dai, X.-W.; Yau, S.S.-T.; Liang, C.-H.; Feng, P.-F.; et al. Microwave integrated circuits design with relational induction neural network. *arXiv* **2019**, arXiv:1901.02069.
15. He, L.K.; Zhang, R.; Gong, Q.H. Returnable launch vehicle landing guidance based on reinforcement learning. *Aerosp. Def.* **2019**, *4*, 33–40. [CrossRef]
16. Li, B.H.; Wu, Y.J.; Li, G.F. Hierarchical reinforcement learning guidance with threat avoidance. *Syst. Eng. Electron. Technol.* **2022**, *33*, 1173–1185. [CrossRef]
17. Blackmore, L.; Acikmese, B.; Scharf, D.P. Minimum-Landing-Error Powered-Descent Guidance for Mars Landing Using Convex Optimization. *J. Guid. Control Dyn.* **2010**, *33*, 1161–1171. [CrossRef]
18. Guo, J.; Xiang, Y.; Wang, X.; Shi, P.; Tang, S. An online trajectory planning method for rocket vertical recovery based on HP pseudospectral homotopy convex optimization. *J. Astronaut.* **2022**, *43*, 603–614.
19. Wang, X.Y.; Li, Y.F.; Quan, Z.Y.; Wu, J.B. Optimal trajectory-tracking guidance for reusable launch vehicle based on adaptive dynamic programming. *Eng. Appl. Artif. Intell.* **2023**, *117*, 105497. [CrossRef]
20. Ignatyev, D.I.; Shin, H.S.; Tsourdos, A. Sparse online Gaussian process adaptation for incremental backstepping flight control. *Aerosp. Sci. Technol.* **2023**, *136*, 108157. [CrossRef]

21. Simplicio, P.; Marcos, A.; Bennani, S. Reusable Launchers: Development of a Coupled Flight Mechanics, Guidance, and Control Benchmark. *J. Spacecr. Rocket.* **2019**, *57*, 74–89. [CrossRef]
22. Song, Y.; Zhang, W.; Miao, X.Y.; Zhang, Z.; Gong, S. Online guidance algorithm for the landing phase of recoverable rocket power. *J. Tsinghua Univ.* **2021**, *61*, 230–239.
23. Zhang, Y.; Huang, C.; Li, X.F. Online Attitude Adjustment Planning Method for Long March 5 Launch Vehicle. *Missile Space Launch Technol.* **2021**, *3*, 22–25.
24. Howard, M. *Multi-Agent Machine Learning: A Reinforcement Approach*; China Machine Press: Beijing, China, 2017.
25. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [CrossRef] [PubMed]
26. Gallego, V.; Naveiro, R.; Insua, D.R. Reinforcement learning under threats. In Proceedings of the AAAI Conference on Artificial Intelligence, Honolulu, HI, USA, 27 January–1 February 2019; pp. 9939–9940.
27. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [CrossRef] [PubMed]
28. Huang, W.Y.; Huang, S.J. Behavioral cloning method based on demonstrative active sampling. *J. Nanjing Univ. Aeronaut. Astronaut.* **2021**, *53*, 766–771. [CrossRef]