aerospace

MDPI

Article

Two-Stage Pursuit Strategy for Incomplete-Information Impulsive Space Pursuit-Evasion Mission Using Reinforcement Learning

Bin Yang [1], Pengxuan Liu [1], Jinglang Feng [2] and Shuang Li [1,*]

[1] College of Astronautics, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China; binyang@nuaa.edu.cn (B.Y.); liupx2015@nuaa.edu.cn (P.L.)
[2] Department of Mechanical and Aerospace Engineering, University of Strathclyde, Glasgow G1 1XJ, UK; jinglang.feng@strath.ac.uk
* Correspondence: lishuang@nuaa.edu.cn; Tel.: +86-25-8489-6039

Abstract: This paper presents a novel and robust two-stage pursuit strategy for the incomplete-information impulsive space pursuit-evasion missions considering the J2 perturbation. The strategy firstly models the impulsive pursuit-evasion game problem into a far-distance rendezvous stage and a close-distance game stage according to the perception range of the evader. For the far-distance rendezvous stage, it is transformed into a rendezvous trajectory optimization problem and a new objective function is proposed to obtain the pursuit trajectory with the optimal terminal pursuit capability. For the close-distance game stage, a closed-loop pursuit approach is proposed using one of the reinforcement learning algorithms, i.e., the deep deterministic policy gradient algorithm, to solve and update the pursuit trajectory for the incomplete-information impulsive pursuit-evasion missions. The feasibility of this novel strategy and its robustness to different initial states of the pursuer and evader and to the evasion strategies are demonstrated for the sun-synchronous orbit pursuit-evasion game scenarios. The results of the Monte Carlo tests show that the successful pursuit ratio of the proposed method is over 91% for all the given scenarios.

Keywords: space pursuit-evasion mission; incomplete-information game; reinforcement learning; impulsive propulsion; J2 perturbation

Citation: Yang, B.; Liu, P.; Feng, J.; Li, S. Two-Stage Pursuit Strategy for Incomplete-Information Impulsive Space Pursuit-Evasion Mission Using Reinforcement Learning. *Aerospace* **2021**, *8*, 299. https://doi.org/10.3390/aerospace8100299

Academic Editor: Fanghua Jiang

Received: 29 August 2021
Accepted: 7 October 2021
Published: 14 October 2021

1. Introduction

The space pursuit-evasion (PE) game is a typical zero-sum game [1,2], where the goals of both confrontation sides are completely opposite and irreconcilable. With the development of space technology, it is one of the focuses of space security and has been investigated extensively by many scholars. Differential game theory was firstly proposed by Isaacs [3] in 1965 and is an effective approach to address the zero-sum game problem [4,5]. In differential game, the PE game is transformed into a two-point boundary value problem (TPBVP) using Hamilton–Jacobi–Bellman equation [6]. However, for the space PE game, it is a challenge to solve the transformed TPBVP due to its high dimensionality and strong nonlinearity.

Some approaches were proposed to address these two challenges and improve the performance of differential theory on space PE game problems. Anderson et al. [7] linearized the equations of the spacecraft motion and approximated the thrust angle control using polynomial to obtain a simplified spacecraft planar PE analytical expression. Li et al. [8] modeled the relative states of two spacecraft in near-circular orbits with the circular-orbit variational equations to reduce the dimensionality. Jagat and Sinclair [9] applied the state-dependent Riccati equation method to obtain nonlinear control law for two space-

craft PE game in the Hill coordinate system. Blasch et al. [10] degenerated the three-dimensional (3D) PE game into a two-dimensional coplanar PE game by firstly assuming that the pursuer matches the orbital plane of the escaper, which is easy to address. However, both the linear simplified approaches and the planarity assumptions are inconsistent with the actual situation as the space PE game is a typical 3D nonlinear PE game. Furthermore, the perturbations in realistic dynamics of the actual space PE game missions make it more challenging to solve the game. The major perturbation is from the J2 spherical harmonic term of the Earth gravity field.

Li et al. [11] developed the combined shooting and collocation method to address the accurate saddle point of the 3D PE game using the J2-perturbed dynamics. Based on the state-dependent Riccati equation method, Jagat et al. [12] used a state-dependent coefficient matrix to derive a nonlinear control law from the linear quadratic differential game theory. Pontani and Conway [13] proposed a semi-direct collocation with nonlinear programming (SDCNLP) method, which obtains the solution for one side with the analytical necessary conditions of another side, and the initial guesses of the nonlinear programming method are generated using the genetic algorithm (GA). Carr et al. [14] developed a fast method to obtain initial guesses of the co-states needed in the SDCNLP method and a penalty-functions technique to deal with state inequality constraints in the indirect player's objective. Because SDCNLP only uses the analytic optimal necessary condition for the evader, the obtained saddle point is not accurate. Therefore, Sun et al. [15] proposed a hybrid method combining the new SDCNLP that introduces two optimal control problems corresponding to the differential game and the multiple shooting method to improve the convergence and accuracy of solving the TPBVP of the space PE game. Hafer et al. [16] employed the sensitivity method to address space PE game problems and utilized a homotopy strategy to improve the efficiency of the algorithms. Shen et al. [17] applied an indirect optimization method to the 3D space PE game and found the local optimal solutions, which satisfy the analytical necessary conditions for optimality. Further, the constraints of the minimum altitude and mass variation were considered for making the saddle-point solution more accurate.

For the above studies, the information of both players is completely disclosed and both two players in space PE game are assumed to be sane enough. Actually, due to the communication delay and the non-cooperation of the players, there are large uncertainties during the PE game. Cavalieri et al. [18] applied a two-step dynamic inversion to allow behavior learning methods to estimate the opponent behavior for incomplete-information PE games with uncertain relative dynamics. Shen et al. [19] considered the uncertainty of the J2-perturbed dynamical model and used quantitative indicators of uncertainty as the game payoff function to solve the incomplete-information space PE problem. Li et al. [20] developed a currently optimal evasive control method using a modified strong tracking unscented Kalman filter to modify the guess and to update the strategy during the game.

The closed-loop control method, which can update the trajectory based on the real-time feedback, is a valid approach to deal with uncertainties and emergencies and is widely used in space missions, especially for the realistic space PE game that is a dynamical process [20]. However, the approaches based on the differential game theory are mainly used for continuous-thrust cases and inapplicable for impulse cases. In addition, the computational time cost of solving the saddle point is expensive. Therefore, it is challenging to develop a feedback closed-loop control method with high efficiency for the impulsive space PE game missions considering the perturbations of the dynamics. The development of artificial intelligence provides alternative ways to address this challenge. Reinforcement learning (RL) as the representative of intelligent algorithms can interact with the environment in real time and obtains the optimal control of the maximum reward through data training [21,22]. RL has been widely employed to solve PE problems in the field of unmanned aerial vehicle (UAV) [23–25]. Different from the UAV PE game, the space PE game has a long mission duration and complex dynamics. In the field of space PE game, Liu [26] and Wang [27] developed the improved branching deep Q networks

and the fuzzy actor-critic learning algorithm, respectively. These previous researches usually restricted the initial distance between the two spacecraft to reduce the PE game duration and used a simplified dynamical model to improve the computational efficiency. To remove this limitation and consider realistic space PE game problems, in this paper, a novel two-stage pursuit strategy is developed to find a robust solution for incomplete-information impulsive space pursuit-evasion missions considering J2 perturbation. For the far-distance rendezvous stage (FRS), a new game capability index of the pursuer is proposed as the objective function of multi-impulses transfer trajectory optimization with the J2-perturbed dynamical model. For the close-distance game stage (CGS), a novel closed-loop approach using the deep deterministic policy gradient (DDPG) algorithm is developed to solve the impulsive maneuver strategy according to the incomplete feedback information. The proposed method is applied to the scenarios of spacecraft games in the sun-synchronous orbit, which demonstrates outstanding advantages in robustness to various initial states of the pursuer and the evader and to the different evasion strategies.

2. Problem Formulation

This section introduces the dynamical model considering the J2 non-spherical term of the Earth and the formulations of the space pursuit-evasion game problem.

2.1. Dynamical Model with J2 Perturbation

Motion of the spacecraft during impulsive PE game is described in the J2000 Earth-centered inertial frame, and both the pursuer and evader use impulse maneuvers to perform orbital transfer. J2 perturbation is considered in the dynamical model, and the corresponding equations of the spacecraft's motion are given as follows:

$$\begin{cases} \dot{x} = v_x \\ \dot{y} = v_y \\ \dot{z} = v_z \\ \dot{v}_x = -\dfrac{\mu x}{r^3}\left(1+\dfrac{3}{2}J_2\left(\dfrac{R_0}{r}\right)^2\left(1-5\dfrac{z^2}{r^2}\right)\right) \\ \dot{v}_y = -\dfrac{\mu y}{r^3}\left(1+\dfrac{3}{2}J_2\left(\dfrac{R_0}{r}\right)^2\left(1-5\dfrac{z^2}{r^2}\right)\right) \\ \dot{v}_z = -\dfrac{\mu z}{r^3}\left(1+\dfrac{3}{2}J_2\left(\dfrac{R_0}{r}\right)^2\left(3-5\dfrac{z^2}{r^2}\right)\right) \end{cases} \tag{1}$$

where $[x, y, z]^T$ and $[v_x, v_y, v_z]^T$ denote the position and velocity vectors of the spacecraft in the J2000 Earth-centered inertial frame. $r = \sqrt{x^2 + y^2 + z^2}$ is the magnitude of the position. J_2 is the J2 zonal harmonic coefficient representing the effect of the Earth's oblateness, and R_0 represents the mean equatorial radius of the Earth. $[\Delta v_x, \Delta v_y, \Delta v_z,]^T$ denotes the impulse maneuver of the spacecraft.

2.2. Formulation of Non-Cooperation Target Pursuit Problem

Actually, the pursuit and evasion spacecraft move in different orbits at a safe distance before the space PE game mission starts. Considering the perception range constraint of the evasion spacecraft (e.g., 200 km), the practical space PE game mission usually breaks down into two phases: the far-distance rendezvous stage and the close-distance game stage, as shown in Figure 1. The evader and pursuer spacecraft have different game strategies at different stages of missions.

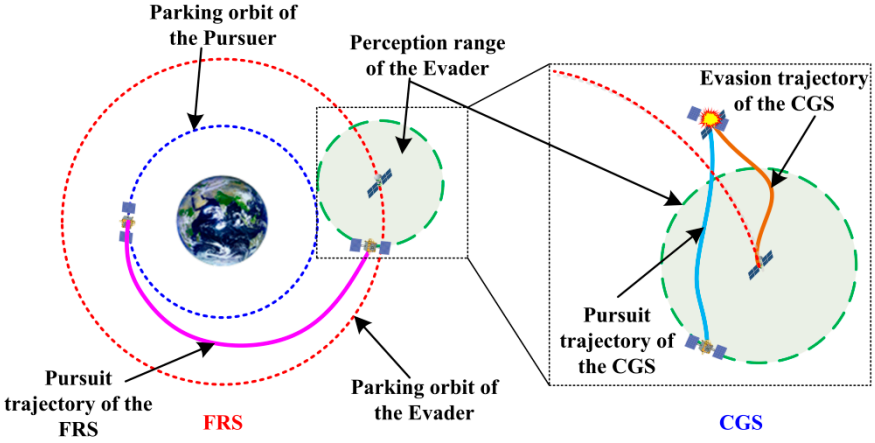

Figure 1. Illustration of FRS and CGS for a practical space PE game mission.

In the FRS, the pursuit spacecraft implements impulse maneuvers to be injected into the rendezvous trajectory of the evader and approaches the evader until reaching its perceived boundary. During this stage, the evader stays in its initial orbit with no response to the pursuer's action, since the pursuit spacecraft is out of its perception range. Thus, the PE game problem is transformed to a multi-impulse rendezvous trajectory optimization problem of the pursuit spacecraft. The formula of the objective function is given as follows,

$$J_F = f\left(t_1, \Delta \boldsymbol{v}_1, \cdots, t_{n-2}, \Delta \boldsymbol{v}_{n-2}, t_{n-1}, t_n\right) \tag{2}$$

where t_i is the epoch of the i-th impulse maneuver, and Δv_i denotes the i-th velocity increments. n is the total number of impulse maneuvers. The last two impulse maneuvers are calculated by solving the Lambert problem.

The CGS starts when the pursuit spacecraft moves within the evader's perception range. At this stage, the evader performs impulsive maneuvers to evade the pursuer. Meanwhile, the pursuit spacecraft also try to rendezvous the evader by impulsive maneuver operations. For complete-information games, the pursuer and evader know each other's objective function and game strategy. However, for a more general and realistic space PE mission, the players only know their own game strategies and the delayed information of their opponent's actions, which is defined as the incomplete-information game.

Without loss of generality, the state of space PE game is defined as $\boldsymbol{s} = \boldsymbol{s}_{\mathrm{P}}\text{-}\boldsymbol{s}_{\mathrm{E}}$, where $\boldsymbol{s}_i = [x_i, y_i, z_i, v_{xi}, v_{yi}, v_{zi}]^T$ denotes the state vector of the spacecraft in the J2000 Earth-centered inertial frame, where the subscript i = P or E indicates the pursuer and evader respectively. Therefore, the general objective functions of the pursuit and evasion spacecraft are defined in Equations (3) and (4) [20]. The objective function consists of two parts: the process state and the terminal state. The former includes the relative states of the pursuer and evader and their control consumption during the mission. The game strategy of the spacecraft is determined by the weight matrix of each item.

$$J_{\mathrm{P}} = 0.5\int_{t_0}^{t_f}\left[\boldsymbol{s}^T\mathbf{Q}_{\mathrm{P}}\boldsymbol{s} + \boldsymbol{u}_{\mathrm{P}}^T\mathbf{W}_{\mathrm{P\text{-}P}}\boldsymbol{u}_{\mathrm{P}} - \boldsymbol{u}_{\mathrm{E}}^T\mathbf{W}_{\mathrm{P\text{-}E}}\boldsymbol{u}_{\mathrm{E}}\right]\mathrm{d}t + 0.5\boldsymbol{s}_{\mathrm{f}}{}^T\mathbf{Q}_{\mathrm{fP}}\boldsymbol{s}_{\mathrm{f}} \tag{3}$$

$$J_{\mathrm{E}} = 0.5\int_{t_0}^{t_f}\left[\boldsymbol{s}^T\mathbf{Q}_{\mathrm{E}}\boldsymbol{s} + \boldsymbol{u}_{\mathrm{P}}^T\mathbf{W}_{\mathrm{E\text{-}P}}\boldsymbol{u}_{\mathrm{P}} - \boldsymbol{u}_{\mathrm{E}}^T\mathbf{W}_{\mathrm{E\text{-}E}}\boldsymbol{u}_{\mathrm{E}}\right]\mathrm{d}t + 0.5\boldsymbol{s}_{\mathrm{f}}{}^T\mathbf{Q}_{\mathrm{fE}}\boldsymbol{s}_{\mathrm{f}} \tag{4}$$

where t_0 and t_f are the initial and final epoch of the mission, respectively. $\boldsymbol{s}_f$ denotes the final state of the game mission. $\boldsymbol{u}_i = [\Delta v_{xi}, \Delta v_{yi}, \Delta v_{zi}]^T$ represents the impulsive maneuver of spacecraft, where the subscript i = P or E indicates the pursuer and evader respectively. $\boldsymbol{Q}_i$ and $\boldsymbol{Q}_{fi}$ present the weight coefficient matrices of the process and terminal states, respectively. $\boldsymbol{W}_{\mathrm{P\text{-}P}}$ and $\boldsymbol{W}_{\mathrm{E\text{-}E}}$ denote the self-control weight coefficient matrices of pursuer

and evader. $\boldsymbol{W}_{\text{P-E}}$ and $\boldsymbol{W}_{\text{E-P}}$ are the weight coefficient matrices of opponent's control strategy of pursuer and evader, respectively. These weight coefficient matrices are defined as follows [20]

$$\begin{cases} \mathbf{Q}_i = q_i \mathbf{I}^{6\times 6}, & \mathbf{Q}_{\text{f}i} = q_{\text{f}i} \mathbf{I}^{6\times 6} \\ \mathbf{W}_{\text{P}-i} = w_{\text{P}i} \mathbf{I}^{3\times 3}, & \mathbf{W}_{\text{E}-i} = w_{\text{E}i} \mathbf{I}^{3\times 3} \end{cases}, i = \text{P or E} \tag{5}$$

where q_i, $q_{\text{f}i}$, $w_{\text{P}i}$, and $w_{\text{E}i}$ are the preference parameters of the player i between the relative distance and consumed energy in the game, which are the private information of player i.

For traditional zero-sum game problems, the values of the weight matrices in Equations (3) and (4) are the same, and the signs are opposite. However, the game strategies of players have different preferences and the information obtained by both players is also incomplete in realistic missions. Therefore, the weight coefficient matrices in Equations (3) and (4) have different values.

The game mission ends when the states of the spacecraft firstly meet the successful pursuit conditions in Equation (6) or any other terminal constraints in Equation (7).

$$\begin{cases} \left\| \boldsymbol{r}_{\text{P}} - \boldsymbol{r}_{\text{E}} \right\| \le r_{\max} \\ \left\| \boldsymbol{v}_{\text{P}} - \boldsymbol{v}_{\text{E}} \right\| \le v_{\max} \end{cases} \tag{6}$$

$$\begin{cases} t_f = t_{\max} \\ \Delta v_{\text{Pres}} \le 0 \text{ km/s} \end{cases} \tag{7}$$

where $r_{\max}$ and $v_{\max}$ are the maximum distance and velocity tolerances for a successful pursuit, respectively. $t_{\max}$ is the maximum mission duration. These mission parameters are set according to realistic mission requirements. Δv_{Pres} denotes the residual velocity increment of the pursuit spacecraft.

Both players in the space PE game aim to minimize their own objectives. However, the incomplete-information game is a non-zero-sum game due to the different preferences of the players. It is a challenge to address the robust pursuit solution due to the lack of the information of the evader's game strategy. A novel method using the RL technique is proposed to obtain a robust pursuit solution efficiently, which will be introduced in detail in Section 3.2.

3. Two-Stage Pursuit Strategy Using Reinforcement Learning

A two-stage pursuit strategy that consists of an FRS and a CGS is proposed in this section for incomplete-information impulse pursuit-evasion missions. Firstly, a GA is employed to solve the multi-impulse rendezvous trajectory with the optimal terminal game capability. Then, a closed-loop pursuit method using the DDPG algorithm is developed to address a robust impulsive pursuit trajectory for the incomplete-information PE game.

3.1. Multi-Impulse Pursuit Trajectory Optimization for FRS

During the FRS, the pursuit trajectory solving is a typical transfer trajectory optimization problem because the evader cannot perceive the pursuer. The process of multi-impulse rendezvous is shown in Figure 2. In order to ensure successful rendezvous with the evader at the terminal time epoch, the last two impulse maneuvers are obtained by solving the Lambert problem. Therefore, the independent variables to be optimized are the maneuver time t_i and the first $n-2$ velocity increments $\Delta \boldsymbol{v}_i$, i.e., $\boldsymbol{X} = \{t_1, t_2, \ldots, t_n, \Delta \boldsymbol{v}_1, \Delta \boldsymbol{v}_2, \ldots, \Delta \boldsymbol{v}_{n-2},\}$.

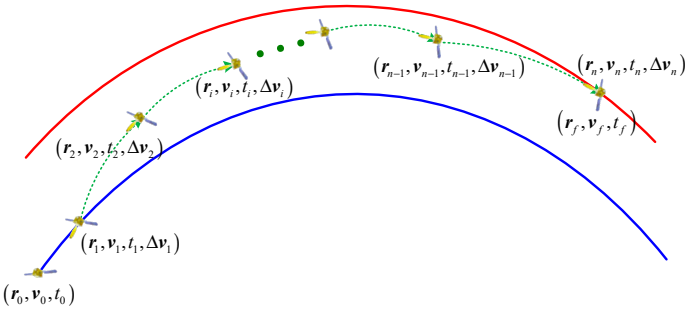

Figure 2. The multi-impulse pursuit trajectory for the far-distance rendezvous stage.

Different from the traditional rendezvous mission trajectory, the pursuit trajectory of the FRS terminates when it reaches the perception range of the evader. Therefore, the maneuvers planned in the perception range of the evader are not actually implemented. The pursuit trajectory aims to achieve the optimal terminal game capability for the FRS. Firstly, the pursuer has a stronger pursuit potential when the terminal residual velocity increment is large. Secondly, when the pursuer's terminal state is closer to that of the evader, it is easier for the subsequent operations in the CGS. The required velocity increment for the close-distance PE game is the minimum if the evader does not perform any evasive maneuvers, which is equal to the sum of the Δv that were planned in FRS but not executed because they are within the perception range. Therefore, the terminal game capability of the pursuer is defined as the ratio of the minimum velocity increments required for the close-distance PE game to the terminal residual velocity increments of the pursuer in the FRS. The corresponding formula is defined as follows,

$$J_F = \min\left(\frac{\sum_{j=k+1}^{n}\left\|\Delta \boldsymbol{v}_j\right\|}{\Delta v_{\text{tol}} - \sum_{j=1}^{k}\left\|\Delta \boldsymbol{v}_j\right\|}\right) = f\left(t_1, \Delta \boldsymbol{v}_1, \cdots, t_{n-2}, \Delta \boldsymbol{v}_{n-2}, t_{n-1}, t_n\right) \tag{8}$$

where n is the number of the planned impulse maneuvers, and k is the number of impulse maneuvers actually performed in the FRS. $\Delta \boldsymbol{v}_j$ denotes the j-th velocity increment vector. Δv_{tol} is the total velocity increment of the pursuit spacecraft.

The equations of motion of the spacecraft with the J2-perturbed dynamics are given as Equation (1), and it is assumed that the impulsive maneuver is performed instantaneously. Therefore, the constraints on the states before and after impulsive maneuver are listed as follows.

$$\begin{cases} \boldsymbol{r}_i^+ = \boldsymbol{r}_i^- \\ t_i^+ = t_i^- \\ \boldsymbol{v}_i^+ = \boldsymbol{v}_i^- + \Delta \boldsymbol{v}_i \end{cases} \tag{9}$$

where the superscripts "+" and "−" indicate before and after the i-th impulse maneuver, respectively.

Finally, the GA is used to search the optimal pursuit trajectory for the FRS. The velocity increment vector is described by the spherical coordinate to improve the optimization performance of algorithm, i.e., $\Delta \boldsymbol{v}_j = [\Delta v, \alpha, \beta]^T$, where Δv, α, and β are the magnitude, azimuthal angle, and polar angle of the velocity increment vector.

3.2. DDPG-Based Pursuit Method for CGS

After completing the FRS, the pursuer moves within the evader's perception range and is discovered by the evader. Then, the evader will perform evasive maneuvers in response to the threat of the pursuer during the CGS. As mentioned in Section 2.2, the close-

distance PE game is actually an incomplete-information game, where the pursuer does not know the game strategy of the evader. Therefore, the pursuer must have the capability to continuously update its pursuit strategy based on the feedback information, to improve the robustness of the pursuit during the CGS. Reinforcement learning, as an important methodology of machine learning, is mainly used to describe and solve the problem of maximizing returns or achieving specific goals through learning strategies in the process of interaction between the agent and the environment. Therefore, a closed-loop pursuit method using a deep deterministic policy gradient algorithm, which is one of the earliest deep RL algorithms, is proposed in this section to solve the robust pursuit strategy for the incomplete-information PE game.

3.2.1. Deep Deterministic Policy Gradient Algorithm

The DDPG algorithm is designed to operate on the large potential state and action spaces with a deterministic policy, which combines both Q-learning and Policy gradients and uses the deep neural networks to approximate the action and the Q-value [25]. DDPG adopts the actor and critic (AC) architecture, as shown in Figure 3. The actor is a policy network that takes the state as the input and outputs the exact action, rather than a probability distribution over actions. The critic is a Q-value network to evaluate the value of the action, which takes state and action as the inputs and outputs the Q-value. Both actor and critic have two networks: the online network and the target network. The roles of these four networks in DDPG are briefly introduced as follows.

- Online actor network $\boldsymbol{a} = A(\boldsymbol{s}, \boldsymbol{\theta}^A)$: it takes state $\boldsymbol{s}$ and returns the corresponding action $\boldsymbol{a}$ that maximizes the long-term reward R.
- Target actor network $\boldsymbol{a}' = A'(\boldsymbol{s}', \boldsymbol{\theta}^{A'})$: it outputs the next action $\boldsymbol{a}'$ using the next state $\boldsymbol{s}'$ sampled in the experience replay memory. Its parameters $\boldsymbol{\theta}^{A'}$ are regularly updated according to the parameters of the online actor network $\boldsymbol{\theta}^A$.
- Online critic network $q = Q(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}^Q)$: it takes state $\boldsymbol{s}$ and action $\boldsymbol{a}$ as inputs and returns the corresponding expectation of Q-value q.
- Target critic network $q' = Q'(\boldsymbol{s}', \boldsymbol{a}', \boldsymbol{\theta}^{Q'})$: it outputs the next expectation of Q-value q' using the next action $\boldsymbol{a}'$ and the next state $\boldsymbol{s}'$ sampled in the empirical playback pool. Its parameters $\boldsymbol{\theta}^{Q'}$ are regularly updated according to the parameters of the online critic network $\boldsymbol{\theta}^Q$.

$\boldsymbol{\theta}^Q$ and $\boldsymbol{\theta}^{Q'}$ are the weights of the online critic network and the target critic network, respectively. $\boldsymbol{\theta}^A$ and $\boldsymbol{\theta}^{A'}$ denote the weights of the online actor network and the target actor network, respectively.

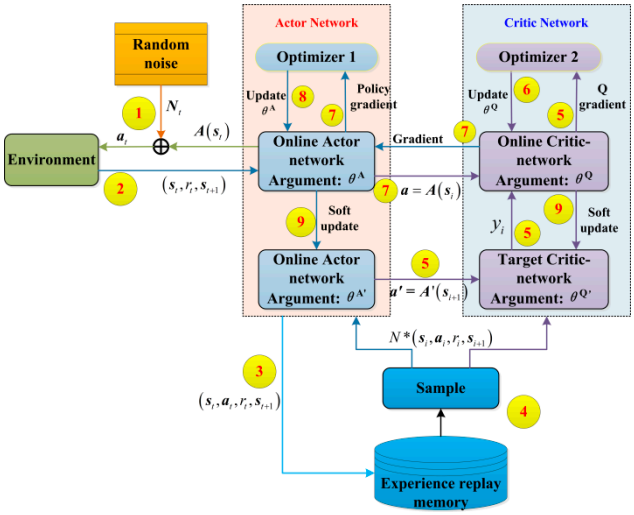

Figure 3. The actor and critic architecture of DDPG.

The soft update technique and the target network are applied to improve the convergence and robustness of the training. The parameters of the online networks are firstly updated through the optimizers (e.g., stochastic gradient descent algorithm), and then the parameters of the target networks are updated through the soft update algorithm, where only a fraction of the weight parameters is transferred in the following manner.

$$\begin{cases} \boldsymbol{\theta}^{Q'} \leftarrow \tau\boldsymbol{\theta}^{Q} + (1-\tau)\boldsymbol{\theta}^{Q'} \\ \boldsymbol{\theta}^{A'} \leftarrow \tau\boldsymbol{\theta}^{A} + (1-\tau)\boldsymbol{\theta}^{A'} \end{cases} \tag{10}$$

where $\tau \in [0,1]$ is the parameter of soft update algorithm.

The loss function of the online critic network is formatted as follows

$$\begin{aligned} L_i &= \frac{1}{N}\sum\nolimits_i \left(y_i - Q\left(\boldsymbol{s}_i, \boldsymbol{a}_i, \boldsymbol{\theta}^{Q}\right)\right)^2 \\ y_i &= R_i + \gamma Q'\left(\boldsymbol{s}_{i+1}, A'\left(\boldsymbol{s}_{i+1}, \boldsymbol{\theta}^{A'}\right), \boldsymbol{\theta}^{Q'}\right) \end{aligned} \tag{11}$$

where N is the number of samples from the replay memory buffer. R_i is the reward of the i-th action. γ is the discount factor of the future reward.

The policy gradient of the online actor network was formulated according to the deterministic policy gradient method as,

$$\nabla_{\boldsymbol{\theta}^A} J \approx \frac{1}{N}\sum\nolimits_i \nabla_{\boldsymbol{a}} Q\left(\boldsymbol{s}, \boldsymbol{a}, \boldsymbol{\theta}^{Q}\right)\Big|_{\boldsymbol{s}=\boldsymbol{s}_i, \boldsymbol{a}=A(\boldsymbol{s}_i)} \nabla_{\boldsymbol{\theta}^A} A\left(\boldsymbol{s}, \boldsymbol{\theta}^{A}\right)\Big|_{\boldsymbol{s}_i} \tag{12}$$

According to the deterministic policy gradient method, the actor-network only outputs the action with the highest probability. This effectively improves the computational efficiency of the algorithm, while its exploration capability was significantly insufficient. Therefore, the off-policy method, which chooses the action $\boldsymbol{a}_t$ based on the current policy and the exploration noise $\boldsymbol{N}_t$, was employed to improve the exploratory capability of the algorithm.

$$\boldsymbol{a}_t = A\left(\boldsymbol{s}_t, \boldsymbol{\theta}^{A}\right) + \boldsymbol{N}_t \tag{13}$$

3.2.2. Closed-Loop Pursuit Method Using DDPG

This section presents a closed-loop pursuit method using DDPG, which enables the pursuer to interact with the environment, to address the incomplete-information PE game problem, as shown in Figure 4. Markov Decision Process (MDP) [28], which is a common model for RL, was used to model the space PE game problem. According to the MDP theory, the agent (here, it is the pursuit spacecraft) takes action after interacting with the environment to change its state for obtaining a reward.

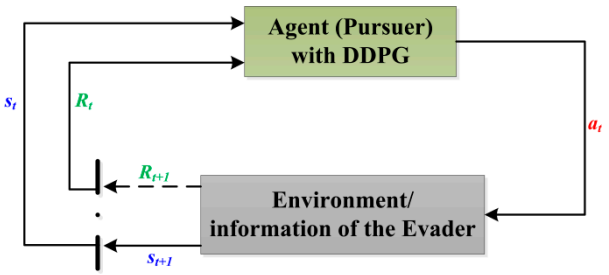

Figure 4. The agent-environment interaction in MDP.

The state and action spaces of the PE game are defined as follows

$$\begin{cases} \boldsymbol{S}_{\mathrm{PE}} = \{\boldsymbol{r}_{\mathrm{P}}, \boldsymbol{v}_{\mathrm{P}}, \boldsymbol{r}_{\mathrm{E}}, \boldsymbol{v}_{\mathrm{E}}\} \\ \boldsymbol{A}_{\mathrm{PE}} = \{\Delta \boldsymbol{v}_{\mathrm{P}}\} \end{cases} \tag{14}$$

The states of the pursuer and the evader are propagated using Equation (1). The return and reward functions are defined as follows

$$R_i = 0.5\int_{t_i}^{t_{i+1}} \left[\boldsymbol{s}^T \mathbf{Q}_{\mathrm{P}} \boldsymbol{s} + \boldsymbol{u}_{\mathrm{P}}^T \mathbf{W}_{\mathrm{P\text{-}P}} \boldsymbol{u}_{\mathrm{P}} - \boldsymbol{u}_{\mathrm{E}}^T \mathbf{W}_{\mathrm{P\text{-}E}} \boldsymbol{u}_{\mathrm{E}} \right] \mathrm{d}t + 0.5 \boldsymbol{s}_{i+1}{}^T \mathbf{Q}_{\mathrm{fP}} \boldsymbol{s}_{i+1} \tag{15}$$

$$G = \varepsilon_{\mathrm{f}} - \kappa \sum_{i=0}^{N_{\mathrm{f}}-1} \frac{R_i}{i} \tag{16}$$

where the variables in Equation (15) have the same definition as those in Equation (3). $\mathbf{W}_{\mathrm{P\text{-}E}}$ is equal to $\mathbf{0}^{6\times 6}$ because the strategy of the evader was unknown. N_{f} denotes the number of steps when the successful pursuit condition in Equation (6) or terminal constraints in Equation (7) are met. κ is the scale coefficient, whose default value is 0.0001. ε_{f} is the reward of the mission completion and is defined as

$$\varepsilon_{\mathrm{f}} = \begin{cases} 10 & \text{, if it satisfies successful pursuit conditions in Eq.(6)} \\ -10 & \text{, else} \end{cases} \tag{17}$$

If the pursuer has a successful rendezvous with the evader, it receives a positive constant reward. Otherwise, it was punished with a negative constant reward.

It is assumed that the evader will perform an impulse maneuver to evade the pursuer when the evasive condition was activated. The evasive condition is defined as

$$\min\left(\left\|\boldsymbol{r}_{t\mathrm{P}} - \boldsymbol{r}_{t\mathrm{E}}\right\|\right) < r_{\mathrm{ec}}, t \in [t_{\mathrm{c}}, t_{\mathrm{f}}] \tag{18}$$

where $\boldsymbol{r}_{t\mathrm{P}}$ and $\boldsymbol{r}_{t\mathrm{E}}$ are the position vectors of the pursuer and evader at time t. t_{c} and t_{f} are the current and terminal time of the mission. r_{ec} denotes the warning distance of the evader.

The maneuver time t_{m} and delta-v $\Delta \boldsymbol{v}$ are optimized using the sequential quadratic programming (SQP) with the following objective function

$$J_{\mathrm{E}} = 0.5\int_{t_{\mathrm{c}}}^{t_{\mathrm{f}}} \left[\boldsymbol{s}^T \mathbf{Q}_{\mathrm{E}} \boldsymbol{s} - \boldsymbol{u}_{\mathrm{E}}^T \mathbf{W}_{\mathrm{E\text{-}E}} \boldsymbol{u}_{\mathrm{E}} \right] \mathrm{d}t + 0.5 \boldsymbol{s}_{\mathrm{f}}{}^T \mathbf{Q}_{\mathrm{fE}} \boldsymbol{s}_{\mathrm{f}} = f\left(t_{\mathrm{m}}, \Delta \boldsymbol{v}_{\mathrm{E}}\right) \tag{19}$$

where the variables in Equation (19) have the same definition as those in Equation (4). Therefore, the evader's strategy was adjusted by changing the weight matrix $\mathbf{Q}_{\mathrm{E}}$, $\mathbf{W}_{\mathrm{E\text{-}E}}$ and $\mathbf{Q}_{\mathrm{fE}}$ during the training.

In order to improve the robustness and generalization capability of the training agent, the initial states of the pursuer and evader and the evasive strategy of the evader are randomly initialized before each episode. The initial states of the pursuer and evader for the CGS are their terminal states of the FRS that are solved using the multi-impulse pursuit trajectory optimization for FRS in Section 3.2.1.

4. Simulations and Analysis

A series of PE games in the sun-synchronous orbit (SSO), whose right ascension of ascending node drifts with a fixed precession rate under the effect of J2 perturbation, are studied to verify the feasibility and performance of the proposed two-stage pursuit strategy. The pursuit and evasion spacecraft park on a sun-synchronous circular orbit and a sun-synchronous elliptical orbit, respectively. Both the pursuer and the evader use the impulse to implement orbital maneuver. According to the realistic space PE mission requirements, the mission constraints are listed in Table 1. The mission duration was limited to 4 h for the consideration of the timeliness of the space PE mission. In addition, considering the difference between the initial orbital planes of the pursuer and evader, the total delta-V of the pursuer was set to be 3 times of that of the evader.

Table 1. The mission constraints of the sun-synchronous orbit PE game case.

Constraints	Value
Mission duration	0~4 h
Total delta-v of the pursuer	1.5 km/s
Total delta-v of the evader	0.5 km/s
Perceived distance of the evader	200 km
Warning distance of the evader	20 km
Maximum distance tolerances of successful pursuit	1 km
Maximum velocity tolerances of successful pursuit	0.1 km/s

The initial orbital ranges of the pursuit and evasion spacecraft are given in Table 2. We always generate the initial conditions starting from an initial set of orbital elements. Values of the orbital elements for each sample are randomly generated with the *rand* function in MATLAB using the intervals defined in Table 2.

Table 2. The initial orbital ranges of the pursuit and evasion spacecraft.

Orbital Elements	Pursuer	Evader
Semi-major axis, a_0 (km)	[6678, 7178]	[6678, 7178]
eccentricity, e_0	0	[0, 0.02]
inclination, Inc_0 (deg)	[96.67, 98.6]	[96.67, 98.6]
Right ascension of the ascending node, $RAAN_0$ (deg)	[56.25, 60]	[56.25, 60]
Argument of perigee, w_0 (deg)	[0, 180]	[0, 180]
Mean anomaly, M_0 (deg)	[0, 180]	[0, 180]

A pair of pursuer and evader forms one PE game sample scenario and 50,000 game sample scenarios are randomly generated. The disparity of the initial parking orbits of the pursuer and evader of all sample scenarios are given in Figure 5. It is seen that the difference of the semi-major axis and the orbital plane are limited to 500 km and 5 deg respectively. In addition, the initial relative distances between the pursuers and evaders are all over 200 km, making sure that the pursuer was out of the perceived range of the evader. Therefore, the proposed two-stage pursuit strategy in Section 3 can be applied to generate the pursuit trajectory.

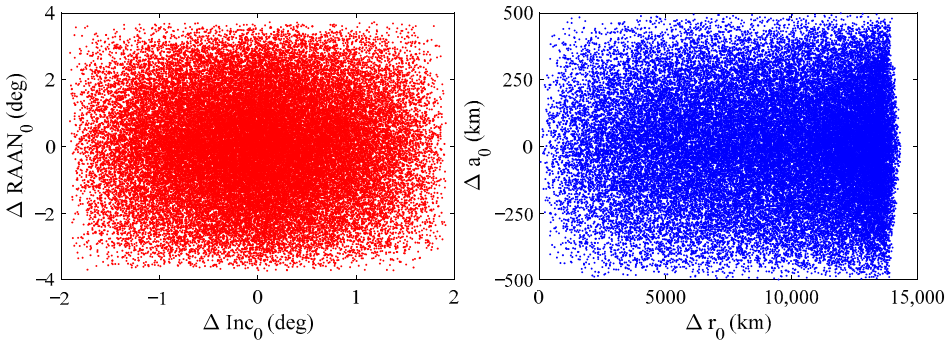

Figure 5. The disparity of the initial parking orbits of the pursuer and evader of all sample scenarios

4.1. Far-Distance Rendezvous

The number of the impulse maneuvers was set to three for the FRS because the mission duration was limited to 4 h. GA was used as the optimizer to find the optimal transfer

trajectory, and the fitness function is defined as Equation (8). For the training, the population was 200 and the maximal generation was 300. The rates of reproduction, crossover and mutation are 0.9, 0.75 and 0.05, respectively.

A specific scenario (denoted as case A) with initial states given in Table 3 was implemented to verify the performance of the proposed method. The variation of the pursuer's terminal game capability J_F with the generations is given in Figure 6. After 216 generations, the J_F finally converges to 0.00789, which indicates that the pursuer retains a strong pursuit potential when reaching the evader's perception boundary.

Table 3. The initial orbits of the pursuer and evader in case A.

Player	a_0 (km)	e_0	Inc_0 (deg)	$RAAN_0$ (deg)	w_0 (deg)	M_0 (deg)
Pursuer	7045.317	0	98.054	56.542	116.152	249.070
Evader	6688.282	0.002	96.707	57.635	225.359	63.076

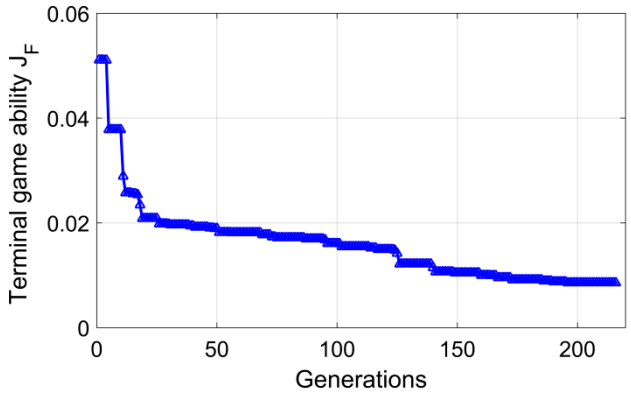

Figure 6. The variation of the pursuer's terminal game ability J_F with generations for GA optimization.

The pursuit trajectory of the FRS for case A is shown in Figure 7. The pursuer performed the first impulse maneuver at 25.11 min to be injected into the pursuit trajectory. Then, the second impulse maneuver was performed at 2 h 51 min to rendezvous with the evader. With these two maneuvers, the trajectory of the pursuer until it reached the evader's perception boundary is given as the red solid line in Figure 7. The trajectory represented by the pink dotted line is the planned pursuit trajectory but not executed because it is within the evader's perception range. The third maneuver was planned at the rendezvous position with the evader, which was also not executed. The obtained pursuit trajectory in the FRS allowed the pursuer to retain the pursuit potential and obtain more advantage in the subsequent close-distance PE game.

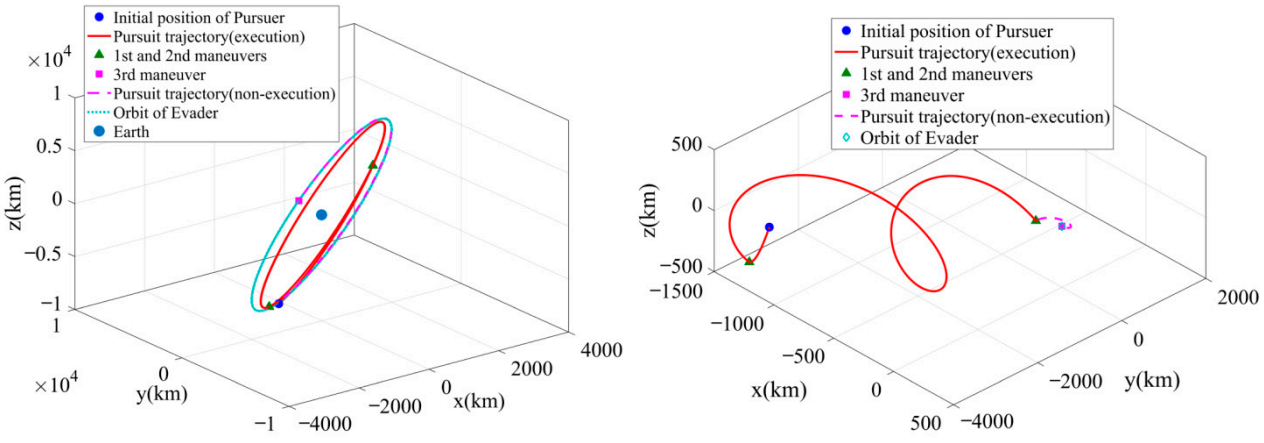

Figure 7. The pursuit trajectory of the FRS for case A (**left** is the pursuit trajectory in J2000 Earth-centered inertial frame; **right** is the pursuit trajectory in the orbital coordinate system of the evader).

Similarly, the pursuit trajectory of the FRS is optimized using GA for all 50,000 sample scenarios to obtain the initial state of the close-distance game, which generates the initial state database for the DDPG training. The optimization results of 50,000 sample scenarios are given in Figure 8.

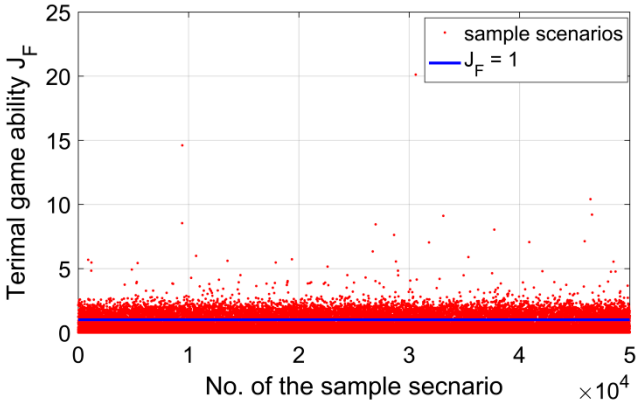

Figure 8. The terminal game capability of the pursuer J_F in the FRS for 50,000 sample scenarios.

The terminal game ability of the pursuer J_F represents the pursuit potential of the pursuer. If J_F is greater than 1, it means the pursuer does not have enough delta-v to reach the evader. A smaller J_F indicates the greater pursuit potential of the pursuer. There are 41,926 sample scenarios, whose J_F are all less than 1, have enough delta-v to continue the subsequent close-distance game. The J_F distribution of these 41,926 sample scenarios is given in Figure 9.

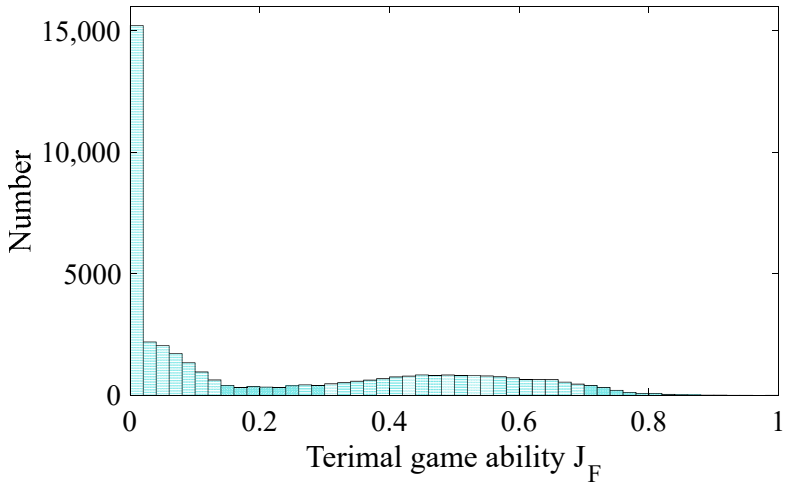

Figure 9. The J_F distribution of the feasible sample scenarios for FRS.

4.2. Close-Distance Pursuit-Evasion Game

DDPG includes four deep neural networks that are fully connected, and the specific parameters of these neural networks are given in Table 4. All critic neural networks have five hidden layers and actor neural networks have three hidden layers. Based on the experience from the multiple tests and the test results, the number of neurons per hidden layer was 100 for all neural networks. The activation functions of all deep neural networks

used a combination of the linear function "relu" and the hyperbolic tangent function "tanh".

Table 4. The specific parameters of the neural networks in DDPG.

Type	Online Critic	Target Critic	Online Actor	Target Actor
Hidden layers	5	5	3	3
Neurons per hidden layer	[100; 100; 100; 100; 100]	[100; 100; 100; 100; 100]	[100; 100; 100]	[100; 100; 100]
Activation function	[relu; relu; relu; relu; tanh]	[relu; relu; relu; relu; tanh]	[relu; relu; tanh]	[relu; relu; tanh]

The maximum time duration of the CGS was set to 3600 s, and the time-step of the training was set to 10 s. Therefore, the maximal steps of each game were 360. The learning rates of online actor network and online critic network are 0.001 and 0.0001, respectively. The discount factor of future reward was 0.95. In order to increase the agent's exploration ability, the action interference factor was introduced with the initial value of 0.1, and it decayed at a rate of 0.8 per episode. The capacity of the experience library was set to 30,000. When the experience library was full, the network training was carried out. In the follow-up training, the experience library was gradually updated. In order to obtain independent samples as many as possible, each episode extracted a small batch of samples from the experience library for training. The number of the batch samples set was 256.

For each episode, the sample scenario was randomly selected from the initial state database that is obtained in Section 4.1, and the evasion strategy of the evader was updated as well. According to Equations (5) and (19), the weight matrices $\boldsymbol{Q}_{\mathrm{E}}$, $\boldsymbol{W}_{\mathrm{E\text{-}E}}$ and $\boldsymbol{Q}_{\mathrm{fE}}$ in Equation (19) are updated by the random parameters q_{E}, $w_{\mathrm{E\text{-}E}}$, and q_{fE}, whose value ranges are listed in Table 5. Similarly, the weight matrices in Equation (15), which are defined in Equation (5), and the values of weight parameters are also given in Table 5.

Table 5. The values and ranges of weight parameters of pursuer and evader.

Players	q_{i}	q_{fi}	w_{Pi}	w_{Ei}
Pursuer	2	4	2	0
Evader	$[-5, -1]$	$[-5, -1]$	[1, 5]	[1, 5]

The return value in the DDPG training process is obtained and given in Figure 10. After more than 5000 episodes of random exploration, the return value of the agent and the probability of successful rendezvous gradually increased. Finally, after 30,000 episodes, the pursuit success percentage (PSP) per 100 episodes reached above 95%, as shown in Figure 11.

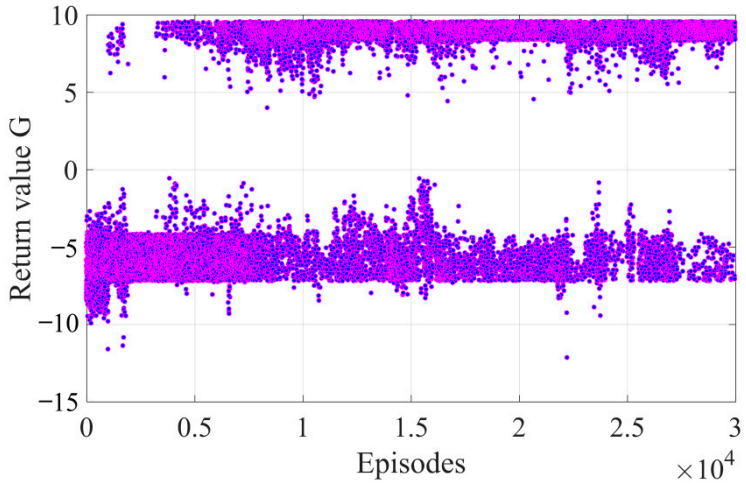

Figure 10. The return value of the DDPG during training.

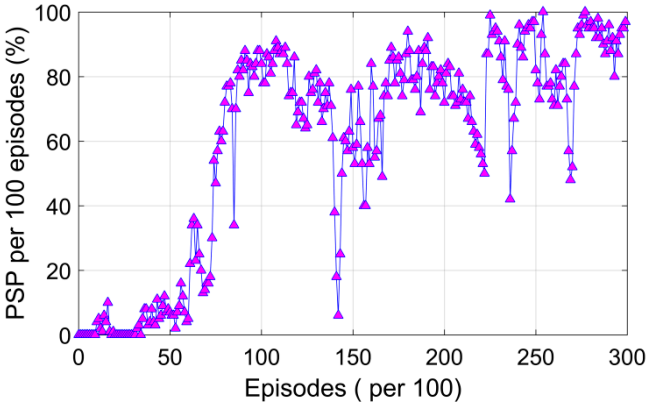

Figure 11. The pursuit success percentage per 100 episodes during the training.

A well-trained agent was applied to solve the pursuit trajectory in the CGS for case A. Without loss of generality, the evasion strategy parameters q_E, $w_{E\text{-}E}$, and q_{fE} are 1, 2 and 1, respectively. The relative distance and velocity between the pursuer and evader during the CGS are given in Figure 12. The evader performed five impulse maneuvers to evade the pursuer. However, the pursuer with DDPG always updated the pursuit strategy and implemented corresponding maneuvers in time to maintain the tendency of approaching the evader. The trajectory of the pursuer in the orbital coordinate system of the evader is given in Figure 13.

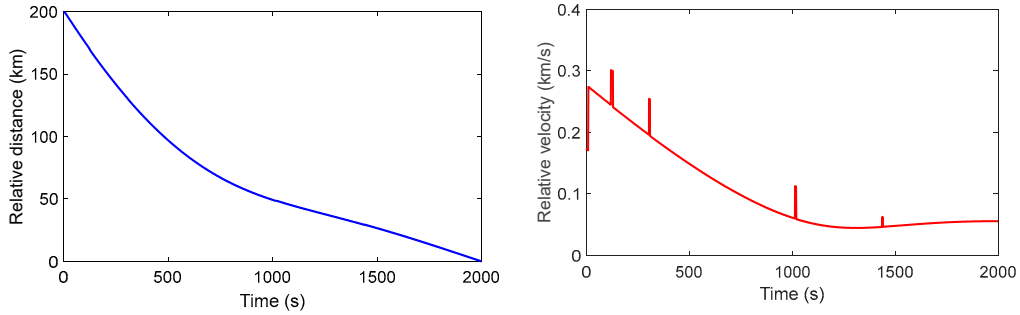

Figure 12. The relative distance (**left**) and velocity (**right**) between the pursuer and evader during the CGS.

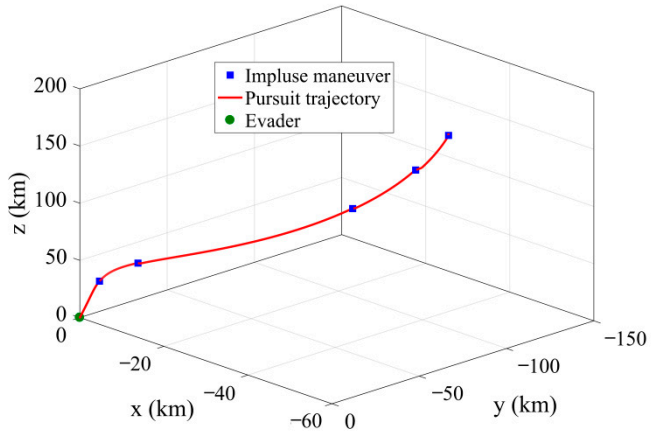

Figure 13. The trajectory of the pursuer in the orbital coordinate system of the evader.

4.3. Monte Carlo Analysis

In order to verify the robustness of the DDPG-based pursuit approach to the initial states of the pursuer and the evader and to the evasion strategies for the CGS, four sets of Monte Carlo simulations were performed. The J_F value ranges of the four sets are [0, 0.25], [0.25, 0.5], [0.5, 0.75] and [0.75, 1], respectively. Each set contained 100 samples and the evasion strategy of the evader for each sample was obtained by randomly generating parameters q_E, $w_{E\text{-}E}$, and q_{fE}. The number of successful pursuits for each set is given in Figure 14. When J_F is less than 0.5, all pursuers successfully rendezvous with the corresponding evaders. The successful pursuit rate decreased with the increase of J_F, because a large J_F indicates poor pursuit capability. The total successful pursuit rate was 99.5% and the successful pursuit rate also achieved 91% even for the worst set with JF of [0.75, 1]. This indicates that the proposed method has good robust performance for incomplete-information impulsive pursuit-evasion missions.

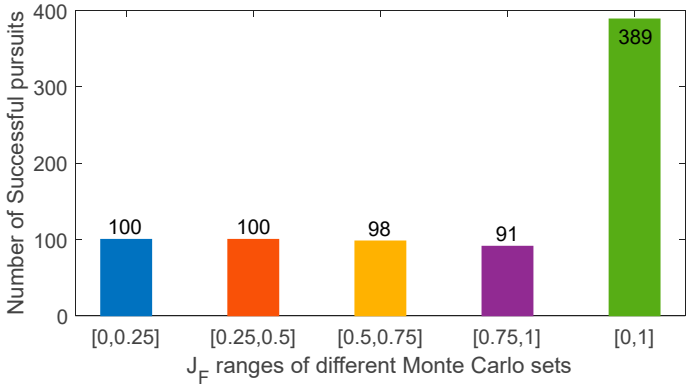

Figure 14. Number of successful pursuits for scenarios with different J_F using DDPG.

5. Conclusions

A novel two-stage pursuit strategy is proposed to find the pursuit trajectory for the incomplete-information impulsive pursuit-evasion missions with the J2-perturbed dynamics using reinforcement learning. The major contributions of this method include the following aspects. The spacecraft PE game problem is modeled into two stages, i.e., FRS and CGS, for the first time. For the FRS, a new objective function defining the terminal pursuit capability of FRS is proposed to optimize the pursuit trajectory for FRS with GA. For the CGS, a closed-loop pursuit approach using the DDPG algorithm is developed to solve the robust pursuit trajectory based on the real-time feedback information of the evader. The consideration of the J2 perturbation significantly improves the feasibility and reliability of the solutions for realistic missions. In addition, the well-trained agent with DDPG directly outputs the impulsive maneuver information based on the real-time conditions of the dynamical environment, which is very efficient because it does not require complicated calculation operations of solving the nonlinear equations and integration. The application to the sun-synchronous orbital PE game scenario demonstrates the feasibility and validity of the proposed method. The Monte Carlo tests show that the proposed method is very robust to the initial states of the pursuer and the evader and to the evasion strategies. The successful pursuit ratio achieves 91% even for the worst test scenarios. Therefore, it is concluded that the proposed two-stage pursuit strategy is an efficient and promising method to obtain robust pursuit trajectories for the realistic incomplete-information impulsive pursuit-evasion missions.

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

Funding: This work was supported by the National Natural Science Foundation of China (No. 11972182), sponsored by Qing Lan Project, Science and Technology on Space Intelligent Control Laboratory (Grant No. 6142208200203, HTKJ2020KL502019), Innovation Fund of CAST (Grant No. CAST-2021-01-02), Funding for Outstanding Doctoral Dissertation in NUAA (Grant No. BCXJ19-12), State Scholarship from China Scholarship Council (Grant No. 201906830066). The authors fully appreciate their financial supports.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The authors would like to thank the reviewers for their constructive comments and suggestions that may help improve this paper.

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

Reference

1. Myerson, R.B. *Game Theory: Analysis of Conflict*, 2nd ed.; Harvard University Press: Cambridge, MA, USA, 2013; pp: 122–127.
2. Nash, J.F. Equilibrium points in n-person games. *Proc. Natl. Acad. Sci. USA* **1950**, *36*, 48–49, doi:10.1073/pnas.36.1.48.
3. Isaacs, R. *Differential Games: A Mathematical Theory with Applications to Warfare and Pursuit, Control and Optimization*, 2nd ed.; Courier Corporation: New York, NY, USA, 1999; pp: 25–44.
4. Ho, Y.; Bryson, A.; Baron, S. Differential games and optimal pursuit-evasion strategies. *IEEE Trans. Autom. Control* **1965**, *10*, 385–389, doi:10.1109/TAC.1965.1098197.
5. Berkovitz, L.D. Differential games of generalized pursuit and evasion. *SIAM J. Control Optim.* **1986**, *24*, 361–373, doi:10.1137/0324021.
6. Bellman, R. Dynamic programming and a new formalism in the calculus of variations. *Proc. Natl. Acad. Sci. USA.* **1954**, *40*, 231–235, doi:10.1073/pnas.40.4.231.
7. Anderson, G.M.; Grazier, V.W. Barrier in pursuit-evasion problems between two low-thrust orbital spacecraft. *AIAA J.* **1976**, *14*, 158–163, doi:10.2514/3.61350.
8. Li, Z.; Zhu, H.; Yang, Z.; Luo, Y.Z. A dimension-reduction solution of free-time differential games for spacecraft pursuit-evasion. *Acta Astronaut* **2019**, *163*, 201–210, doi:10.1016/j.actaastro.2019.01.011.
9. Jagat, A.; Sinclair, A.J. Optimization of spacecraft pursuit-evasion game trajectories in the Euler-hill reference frame. In Proceedings of the AIAA/AAS Astrodynamics Specialist Conference, San Diego, CA, USA, 4–7 August 2014; doi:10.2514/6.2014-4131.
10. Blasch, E.P.; Pham, K.; Shen, D. Orbital satellite pursuit-evasion game-theoretical control. In Proceedings of the 2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA), IEEE, Montreal, QC, Canada, 2–5 July 2012; pp. 1007-1012, doi:10.1109/ISSPA.2012.6310436.
11. Li, Z.; Zhu, H.; Yang, Z.; Luo, Y.Z. Saddle point of orbital pursuit-evasion game under J 2-perturbed dynamics. *J. Guid. Control Dyn.* **2020**, *43*, 1733–1739, doi:10.2514/1.G004459.
12. Jagat, A.; Sinclair, A.J. Nonlinear control for spacecraft pursuit-evasion game using the state-dependent Riccati equation method. *IEEE Trans. Aerosp. Electron. Syst.* **2017**, *53*, 3032–3042, doi:10.1109/TAES.2017.2725498.
13. Pontani, M.; Conway, B.A. Numerical solution of the three-dimensional orbital pursuit-evasion game. *J. Guid. Control Dyn.* **2009**, *32*, 474–487, doi:org/10.2514/1.37962.
14. Carr, R.W.; Cobb, R.G.; Pachter, M.; Pierce, S. Solution of a pursuit–evasion game using a near-optimal strategy. *J. Guid. Control Dyn.* **2018**, *41*, 841–850, doi:10.2514/1.G002911.
15. Sun, S.; Zhang, Q.; Loxton, R.; Li, B. Numerical solution of a pursuit-evasion differential game involving two spacecraft in low earth orbit. *J. Ind. Manag. Optim.* **2015**, *11*, 1127–1147, doi:10.3934/jimo.2015.11.1127.
16. Hafer, W.T.; Reed, H.L.; Turner, J.D.; Pham, K. Sensitivity methods applied to orbital pursuit evasion. *J. Guid. Control Dyn.* **2015**, *38*, 1118–1126, doi:10.2514/1.G000832.
17. Shen, H.X.; Casalino, L. Revisit of the three-dimensional orbital pursuit-evasion game. *J. Guid. Control Dyn.* **2018**, *41*, 1823–1831, doi:10.2514/1.G003127.

18. Cavalieri, K.A.; Satak, N.; Hurtado, J.E. Incomplete information pursuit-evasion games with uncertain relative dynamics. In Proceedings of the AIAA Guidance, Navigation, and Control Conference, National Harbor, MD, USA, 13–17 January 2014; doi:10.2514/6.2014-0971.
19. Shen, D.; Jia, B.; Chen, G.; Blasch, E.; Pham, K. Pursuit-evasion games with information uncertainties for elusive orbital maneuver and space object tracking. In *Sensors and Systems for Space Applications VIII*; International Society for Optics and Photonics: Baltimore, MD, USA, 22 May 2015; doi:10.1117/12.2181160.
20. Li, Z.Y.; Zhu, H.; Luo, Y.Z. An escape strategy in orbital pursuit-evasion games with incomplete information. *Sci. China Technol. Sci.* **2021**, *64*, 559–570, doi:10.1007/s11431-020-1662-0.
21. Izzo, D.; Märtens, M.; Pan, B. A survey on artificial intelligence trends in spacecraft guidance dynamics and control. *Astrodynamics* **2019**, *3*, 287–299, doi:org/10.1007/s42064-018-0053-6.
22. Hasanzade, M.; Koyuncu, E. A dynamically feasible fast replanning strategy with deep reinforcement learning. *J. Intell. Robot. Syst*. **2021**, *101*, 1–17, doi:10.1007/s10846-020-01274-1.
23. Liu, P.; Ma, Y. A deep reinforcement learning based intelligent decision method for UCAV air combat. In Proceedings of the Asian Simulation Conference, Melaka, Malaysia, 27–29 August 2017; pp. 274–286, doi:10.1007/978-981-10-6463-0_24.
24. De Souza, C.; Newbury, R.; Cosgun, A.; Castillo, P.; Vidolov, B.; Kulić, D. Decentralized multi-agent pursuit using deep reinforcement learning. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4552–4559, doi:10.1109/LRA.2021.3068952.
25. Yang, Q.; Zhu, Y.; Zhang, J.; Liu, J. UAV air combat autonomous maneuver decision based on DDPG algorithm. In Proceedings of the 2019 IEEE 15th International Conference on Control and Automation IEEE, Edinburgh, UK, 16–19 July 2019; pp. 37–42, doi:10.1109/ICCA.2019.8899703.
26. Liu, B.; Ye, X.; Dong, X.; Ni, L. Branching improved Deep Q Networks for solving pursuit-evasion strategy solution of spacecraft. *J. Ind. Manag. Optim.* **2020**, doi:10.3934/jimo.2021016.
27. Wang, X.; Shi, P.; Zhao, Y.; Sun, Y. A pre-trained fuzzy reinforcement learning method for the pursuing satellite in a one-to-one game in space. *Sensors* **2020**, *20*, 2253, doi:10.3390/s20082253.
28. Andrade, P.; Silva, C.; Ribeiro, B.; Santos, B.F. Aircraft maintenance check scheduling using reinforcement learning. *Aerospace* **2021**, *8*, 113, doi:10.3390/aerospace8040113.