BRANCHING IMPROVED DEEP Q NETWORKS FOR SOLVING PURSUIT-EVASION STRATEGY SOLUTION OF SPACECRAFT

Bingyan Liu∗, Xiongbing Ye and Xianzhou Dong
Academy of Military Science of the People’s Liberation Army, Beijing, China
Lei Ni
Space Engineering University, Beijing, China

(Communicated by Bin Li)

Abstract. With the continuous development of space rendezvous technology, more and more attention has been paid to the study of spacecraft orbital pursuit-evasion differential game. Therefore, we propose a pursuit-evasion game algorithm based on branching improved Deep Q Networks to obtain a space rendezvous strategy with non-cooperative target. Firstly, we transform the optimal control of space rendezvous between spacecraft and non-cooperative target into a survivable differential game problem. Next, in order to solve this game problem, we construct Nash equilibrium strategy and test its existence and uniqueness. Then, in order to avoid the dimensional disaster of Deep Q Networks in the continuous behavior space, we construct a TSK fuzzy inference model to represent the continuous space. Finally, in order to solve the complex and timeconsuming self-learning problem of discrete action sets, we improve Deep Q Networks algorithm, and propose a branching architecture with multiple groups of parallel neural Networks and shared decision modules. The simulation results show that the algorithm achieves the combination of optimal control and game theory, and further improves the learning ability of discrete behaviors. The algorithm has the comparative advantage of continuous space behavior decision, can effectively deal with the continuous space chase game problem, and provides a new idea for the solution of spacecraft orbit pursuit-evasion strategy.

1. Introduction. If non-cooperative target has maneuvering capability, spacecraft may face the problem of target escaping during rendezvous with the non-cooperative target, which may develop into a spacecraft orbital pursuit-evasion game[15]. The pursuit-evasion game between spacecraft and non-cooperative target is a typical sequential decision-making process oriented to incomplete information, and its essence is a continuous dynamic confrontation problem with bilateral control. The two sides have conflicting behavioral goals, spacecraft aims to approach non-cooperative target, and non-cooperative target changes original orbit to try to get rid of spacecraft approaching.

Recently, the differential game theory is the dominant theory in the research of pursuit-evasion game. Classical guidance control theory[44], modern nonlinear

2020 Mathematics Subject Classification. Primary: 58F15, 58F17; Secondary: 53C35.
Key words and phrases. Orbital pursuit-evasion, differential game, orbital maneuver, Deep Q Networks.

∗ Corresponding author: Bingyan Liu, Academy of Military Science of the People’s Liberation Army, Beijing 100091, PR China.
guidance theory\cite{30} and optimal guidance theory\cite{18} also can be applied to pursuit-evasion game, but these theories are limited to the situation that escaping spacecraft has strong maneuvering ability and changeable maneuvering law\cite{25}.

Compared with the general differential game, the pursuit-evasion problem has more maneuvering constraints and more complicated dynamics model, so it is a special kind of differential game. Therefore, Anderson and Grazier\cite{1} studied and proposed an analytic solution of pursuit-evasion boundary grid under linear dynamics model, but the optimal thrust direction obtained by this method is along the direction of solid line. In view of this deficiency, Zhang\cite{45} studied the chase boundary grid in the case of variable thrust magnitude near circular orbit, but this method was also affected by the bias rate greatly and only suitable for the case of small thrust. Hafer et al.\cite{11} studied this problem from a quantitative perspective, transformed the thrust problem to avoid the sensitivity defect, and proposed to use numerical solution to determine the solution of the pursuit-evasion boundary barrier. But this method was difficult to meet the requirement of timeliness due to its large computational cost.

On the other hand, Pontani and Conway\cite{29} proposed a semidirect collocation with nonlinear programming algorithm based linear quadratic differential game. Carr et al.\cite{33} improved the Conway method, solved the game scenario of proportional guidance approaching the optimal tracking law by using unilateral optimal control, and obtained the optimal solution through semi-direct collocation point nonlinear programming on this basis. However, semi-direct collocation point method is only a necessary condition for the optimal solution and cannot guarantee the existence of saddle points. Sun et al.\cite{37} then combined the semidirect point-matching method with the multiple target shooting method to improve the calculation accuracy of the two-point boundary value problem. Li et al.\cite{20} transformed the original two-point boundary value problem by dimension reduction, and used the near-circular deviation formula to describe the pursuit-evasion game motion, but problems such as complex transformation relations and low solving efficiency still exist.

Generally speaking, the two kinds of common differential game methods listed above can solve the pursuit-evasion game problem. However, for the spacecraft orbital pursuit-evasion differential game, the solution process is more complex and computationally intensive, so a more efficient and intelligent algorithm is needed. In addition, the above methods are all carried out under the assumption of complete information condition, that is, both actors know each other’s payment function and coefficient exactly. However, in the actual rendezvous between spacecraft and non-cooperative target, it is difficult for the two sides to know each other’s behavior exactly, and the pursuit-evasion game presents the characteristics of incomplete information differential game.

Differential game solution of the pursuit-evasion is always a difficult and thorny problem because the complex constraint conditions of differential formula involve many nonlinear state variables\cite{19, 4}. Therefore, on the one hand, the solution method of differential game needs to be further innovated to ease the computation, reduce the algorithm complexity and improve the problem solving efficiency. On the other hand, it needs to approach the pursuit-evasion reality between spacecraft and non-cooperative target, and focus on the game problem between spacecraft and non-cooperative target under the condition of incomplete information.
DQN FOR SOLVING PURSUIT-EVASION STRATEGY SOLUTION OF SPACECRAFT

With the rapid development of the new generation of artificial intelligence methods represented by Deep Q Networks[27], the processing of decision control problem is not limited by task mode according to its advantages in self-learning and self-optimization. It has been widely used in computer transportation and other fields, achieved remarkable results[23]. Although these studies of Yin C[5], Liu B Y[22], Wu X G[42], et al make Deep Q Networks be widely used in the field of control decision. Deep Q Networks still faces a similar problem to table Q-learning in continuous space applications, that is, the number of operations that need to be explicitly expressed increases exponentially with the increase of operational dimensions.

In view of the potential of Deep Q Networks in control decision and the limitations of its current application in continuous space, we improve Deep Q Networks and propose a pursuit-evasion game algorithm based on branching Deep Q Networks to obtain the optimal rendezvous strategy. This algorithm can realize the combination of optimal control and game theory, further enhance the learning ability of Deep Q Networks on discrete behaviors, and effectively solve the problem that the differential game model is highly nonlinear and difficult to solve by using the classical optimal control theory.

2. The game between spacecraft and non-cooperative target. In order to describe the relative motion between spacecraft and non-cooperative target better, we construct a two-body model with the central object as the reference point. We take a reference star in the same orbital plane as origin of coordinates, take direction of the line between the reference star and the central celestial body as the X-axis, take the direction of the orbital velocity in the orbital plane as the Y-axis, and take Z-axis perpendicular to the transfer orbital plane. Fig.1 shows the relative motion relationship between spacecraft and non-cooperative target, which P represents spacecraft and E represents non-cooperative target.

The relative distance between spacecraft and non-cooperative target is far less than the orbit radius of non-cooperative target, and its dynamic model can be described as follows[12]

Figure 1. Coordinate diagram of spacecraft and non-cooperative target
\[
\begin{align*}
\ddot{x}_i(t) &= 2 \frac{\mu}{r^3(t)} x_i(t) + 2\omega(t) y_i(t) + \dot{\omega}(t) y_i(t) + \omega^2(t) x_i(t) + \frac{F_i}{m_i} u_1^x(t) \\
\ddot{y}_i(t) &= -\frac{\mu}{r^3(t)} y_i(t) + 2\omega(t) \dot{x}_i(t) + \dot{\omega}(t) x_i(t) + \omega^2(t) y_i(t) + \frac{F_i}{m_i} u_1^y(t) \\
\ddot{z}_i(t) &= -\omega^2(t) z_i(t) + \frac{F_i}{m_i} u_1^z(t)
\end{align*}
\]

Where \(x_i(t), y_i(t)\) and \(z_i(t)\) \((i = P, E)\) respectively represent the components of the spacecraft and the non-cooperative target in the X-axis, Y-axis, and Z-axis directions. \(\dot{x}_i(t), \dot{y}_i(t)\) and \(\dot{z}_i(t)\) respectively represent the first derivatives of the coordinate components with respect to time \(t\). \(\ddot{x}_i(t), \ddot{y}_i(t), \text{ and } \ddot{z}_i(t)\) respectively represent the second derivative of the coordinate components with respect to time \(t\). \(r(t)\) represents the orbital altitude of the reference star. \(\mu\) represents the gravitational constant of the earth. \(\omega\) represents the angular velocity of the reference star. \(F_i\) represents the continuous thrust. \(\delta_i\) represents the pitch angle between the thrust and the orbital plane. \(m_i\) represents the quality. \(\theta_i\) represents the thrust angle in the orbital plane.

For survivable differential game problem\(^4\), both spacecraft and non-cooperative target will take the maximum thrust. Thus, the actual behavior control quantity taken by both sides is the thrust direction angle, which is represented by \(u_p = [\theta_p, \delta_p]\) and \(u_e = [\theta_e, \delta_e]\). Where, the spatial diagram of the pitch angle \(\delta\) and thrust angle \(\theta\) is shown in Fig.2.
Accordingly, we construct the dynamic model of spacecraft orbital pursuit-evasion under survivable differential game

\begin{equation}
\begin{cases}
\ddot{x}_i(t) = \frac{2\mu}{r_3(t)}x_i(t) + 2\omega(t)\dot{y}_i(t) + \omega^2(t)x_i(t) + \frac{F_i}{m_i}\cos\delta_i\cos\theta_i \\
\ddot{y}_i(t) = -\frac{\mu}{r_3(t)}y_i(t) + 2\omega(t)\dot{x}_i(t) + \omega^2(t)y_i(t) + \frac{F_i}{m_i}\cos\delta_i\sin\theta_i \\
\ddot{z}_i(t) = -\omega^2(t)z_i(t) + \frac{F_i}{m_i}\sin\delta_i
\end{cases}
\end{equation}

(2)

Where $\delta_i$ represents the pitch angle between the thrust and the orbital plane. $\theta_i$ represents the thrust angle in the orbital plane, where subscripts $i = P$ and $i = E$ represent spacecraft and space targets, respectively.

The following four basic elements are required to constitute the pursuit-evasion game between spacecraft and non-cooperative target.

**Actor.** Also known as the player, which refers to the decision-making subject who can choose certain behaviors in the game confrontation so as to maximize their own interests in the pursuit-evasion game. The game consisting of spacecraft and non-cooperative target is called a two-person game, which is represented by $N = P, E$.

**Rational.** In the pursuit-evasion game between spacecraft and non-cooperative target, it should be assumed that both players are rational and will select acts under certain constraints to maximize their respective interests. In other words, when faced with two act choices that cannot coexist, each participant will choose the act that makes the objective function optimal.

**Act.** The act of spacecraft orbital pursuit-evasion differential game mainly refers to the decision variables of spacecraft and target at a specific time point, and the acts of both sides at the same time constitutes the act binary group $[u_p, u_e]$.

**Objective function.** Spacecraft and non-cooperative target have different targets and different preferences, so they can optimize their respective expectations through strategy or act selection. Since expected optimization is the goal of each participant, it is called the objective function of actor.

For the objective function of the pursuit-evasion game, we firstly consider the Euclidian distance between the two actors

\[ j(u_p, u_e) = ||[x_p - x_e, y_p - y_e, z_p - z_e]_T^{T = t_f} ||_2^2 \]

(3)

Where $|| \cdot ||_2$ represents the Euclidean norm, $t_f$ represents the time at which the thrust ends.

For a continuous thrust, the fuel consumption is proportional to the thrust action time which means that the longer the thrust action time, the more fuel consumption. Therefore, we take the time interval of thrust action as a part of the objective function and construct the objective function of the comprehensive optimal time-distance control

\[ J(u_p, u_e) = (1 - k) \int_{t_0}^{t_f} j(u_p, u_e)dt + k \int_{t_0}^{t_f} dt \]

(4)

Where $k$ represents proportional weight and $k \in [0, 1]$.

In the pursuit-evasion game, spacecraft and non-cooperative target respectively take acts by independently optimizing the objective function $J$ according to the current state. In this process, spacecraft will strive to obtain the behavior strategy to minimize the objective function $J$, while non-cooperative target will expect to
obtain the behavior strategy to maximize the objective function $J$. According to the Nash equilibrium theory[9, 28] in game theory, the behavior of two parties can reach the Nash equilibrium if and only if the following inequalities are met

$$J(u_p, u_e^*) \geq J(u_p^*, u_e^*) \geq J(u_p^*, u_e)$$  \hspace{1cm} (5)

Where $J(\cdot)$ represents objective function. $u_p$ represents spacecraft’s act strategy. $u_e$ represents non-cooperative target’s act strategy. $u_p^*$ represents spacecraft’s Nash equilibrium act strategy. $u_e^*$ represents non-cooperative target’s Nash equilibrium act strategy.

When spacecraft chooses the Nash equilibrium act strategy $u_p^*$, non-cooperative target takes any act $u_e$ other than the Nash equilibrium. Non-cooperative target will not be able to obtain the optimal objective value.

Therefore, the purpose of solving the pursuit-evasion game problem is to seek a group of behavioral strategies to satisfy the Nash equilibrium

$$J^* = \max_{u_e^*} \min_{u_p^*} J = \min_{u_p^*} \max_{u_e^*} J$$  \hspace{1cm} (6)

By solving the above optimization problem, spacecraft can obtain the Nash equilibrium act of the pursuit-evasion game problem, so as to achieve the optimal intersection with non-cooperative target.

3. **Existence and uniqueness.** Space rendezvous between spacecraft and non-cooperative object is a zero-sum differential game. The competitive goal of both sides is that $P$ wants to minimize the benefit and $E$ wants to maximize the benefit. Because of these conflicting goals, the existence and uniqueness of the pursuit-evasion game problem should be considered carefully.

3.1. **The existence of Nash equilibrium strategy.** In the pursuit-evasion game problem, the two sides have opposite profit goals and try to adopt the most advantageous behavior strategy. Since the Nash equilibrium strategy does not necessarily exist, it is necessary to verify the existence of the Nash equilibrium strategy between spacecraft and non-cooperative target space. In other words, we need to know under what conditions does a Nash equilibrium strategy exist.

**Assumption 3.1.** The admissible strategy sets $u_p$ and $u_e$ are compact sets in some metric space, and $J : u_p \times u_e \rightarrow \mathbb{R}$ is a continuous function from $u_p \times u_e$.

**Definition 3.2.** For the pursuit-evasion game problem, If strategy $u_e \in U_e$ and strategy $u_p \in U_p$ are fixed respectively, the optimal behavior strategy set between spacecraft and non-cooperative target is defined as

$$S^p(u_e^*) = \{u_p \in U_p : J(u_p, u_e^*) = \min_{u_p} J(u_p, u_e)\}$$ \hspace{1cm} (7)

$$S^e(u_p^*) = \{u_e \in U_e : J(u_p^*, u_e) = \min_{u_e} J(u_p, u_e)\}$$ \hspace{1cm} (8)

**Definition 3.3.** The upper value $V^+$ and the lower value $V^-$ of the pursuit-evasion game problem are defined as

$$V^+ = \min_{u_p} \max_{u_e} J(u_p, u_e) = \min_{u_p} J(u_p, u_e^*(u_p))$$ \hspace{1cm} (9)

$$V^- = \max_{u_e} \min_{u_p} J(u_p, u_e) = \max_{u_e} J(u_p^*(u_e), u_e)$$ \hspace{1cm} (10)
For any $n > 0$ there will be a corresponding strategy $u^n_e \in U_e$ that makes the following formula true:

$$ V^- - n \leq J(u^n_p, u^n_e) $$

(11)

So we can get the conclusion that the upper value is greater than or equal to the lower value

$$ V^+ = \min_{u_p} \max_{u_e} J(u_p, u_e) \geq \min_{u_p} J(u^n_p, u_e) = J(u^n_p, u_e) \geq V^- $$

(12)

Similarly, for any $n > 0$ there is a corresponding strategy $u^n_p \in U_p$ such that the lower value is greater than or equal to the upper value

$$ V^- = \max_{u_e} \min_{u_p} J(u_p, u_e) \geq \max_{u_e} (u^n_p, u_e) = J(u^n_p, u_e) \geq V^+ $$

(13)

Thus, there are strategies $u^n_p \in U_p$ and $u^n_e \in U_e$ that satisfy the following formula

$$ V^+ = J(u^n_p, u^n_e) = J(u^n_p, u_e) = V^- $$

(14)

When $u^n_e(n \in [1, N])$ is a series of behavior strategies of the spacecraft, we make $u_e(u^n_p)$ as a non-cooperative target corresponding to $u^n_p$. For any $n \geq 1$, there is a behavior strategy $u^n_e \in U_e$ that satisfies

$$ J(u^n_p, u^n_e) \leq J(u^n_p, u_e(u^n_p)) = V^+ $$

(15)

According to the continuity of the objective function $J$, we know that $u^n_e$ is the Nash equilibrium strategy corresponding to $u^n_p$

$$ J(u^n_p, u^n_e) = \max_{u_e} J(u^n_p, u_e) = J(u^n_p, u_e(u^n_p)) $$

(16)

Similarly, for the behavioral strategy $u^n_p(n \in [1, N])$ adopted by non-cooperative target, $u_p(u^n_e)$ is the corresponding behavioral strategy of spacecraft. For any $n \geq 1$, the behavior strategy $u^n_p \in U_p$ is the Nash equilibrium strategy corresponding to $u^n_e$.

**Theorem 3.4.** Under Assumption 3.1, the Nash equilibrium for the pursuit-evasion game problem exists when $V^+ = V^-$ is present.

3.2. The uniqueness of Nash equilibrium strategy. In process of planning the optimal strategy, the Nash equilibrium strategy is not the only one to be determined, and there may be many equilibrium solutions. Therefore, we need to test the uniqueness of Nash equilibrium, that is to say, we need to clarify the relationship between game values corresponding to different Nash equilibria in the pursuit-evasion game.

In order to better illustrate the characteristics of the pursuit-evasion game, we firstly give assumption and theorem.

**Assumption 3.5.** $f(\cdot)$ is a bounded function that satisfies Lipschitz continuity, that is, the existence of $L > 0$ satisfies the following formula

$$ |f(t, x_1, u_p, u_e) - f(t, x_2, u_p, u_e)| \leq L|x_1 - x_2| $$

(17)

Where $t$ represents time, $x$ represents position, $L$ represents real number and $L \in R^+$. And there are positive real number $K \in R^+$ that satisfy the following formula

$$ |f(t, x, u_p, u_e)| \leq K(1 + |x|) $$

(18)
Theorem 3.6. [36] If \( V^+(t, x) \) is the upper value at time \( t \) in state \( x \), then for any \( h \in [0, t_f - t_0] \)

\[
V^+(t_0, x_0) = \min_{u^p_0 \in U_p} \max_{u^e_0 \in U_e} \left\{ \int_{t_0}^{t_0+h} \ell(t, x_t, u_p, u_e) dt + V^+(t_0 + h, x(t_0 + h)) \right\} \tag{19}
\]

Where \( x(t_0, x_0, u_p, u_e) \) is the X-axis component of formula (2) with \( (t_0, x_0) \) as the initial condition; \( \ell(t, x_t, u_p, u_e) \) is the process component of the objective function formula (4), and constitutes the objective function \( J(u_p, u_e) \) with the terminal component \( \phi(x(t_0)) \)

\[
J(u_p, u_e) = \phi(x(t_0)) + \int_{t_0}^{t_0+h} \ell(t, x_t, u_p, u_e) dt \tag{20}
\]

Let’s set formula (19) equal to 0, that is

\[
\min_{u^p_0 \in U_p} \max_{u^e_0 \in U_e} \left\{ \frac{1}{h} \int_{t_0}^{t_0+h} \ell(t_0, x(t_0), u_p, u_e) dt + \frac{V^+(t_0 + h, x(t_0 + h)) - V^-(t_0, x(t_0))}{h} \right\} = 0 \tag{21}
\]

If \( V^+ \) is a first order differentiable function, when \( h \to 0 \) has

\[
\min_{u^p_0 \in U_p} \max_{u^e_0 \in U_e} \left\{ \ell(t_0, x(t_0), u_p, u_e) + \partial_t V^+ + DV^+ \cdot f(t_0, x(t_0), u_p, u_e) \right\} = 0 \tag{22}
\]

Where \( \partial_t (\cdot) \) is the first partial derivative of the function at time \( t \); \( D \) is the first partial derivative of the function in the state variable \( x \). In addition, formula (21) is called hamilton-bellman-isaacs formula in differential game[3, 24].

We define \( H^+ \) to be

\[
H^+(t, x, \zeta) = \min_{u^p_0 \in U_p} \max_{u^e_0 \in U_e} \left\{ \ell(t_0, x(t_0), u_p, u_e) + \zeta f(t_0, x(t_0), u_p, u_e) \right\} \tag{23}
\]

Then, the hamilton-bellman-isaacs formula can be expressed as

\[
\partial_t V^+ + H^+(t, x, DV^+) = 0 \tag{24}
\]

According to Assumption 3.1, we know that \( V^+(t, x) \) is a continuous function, so \( V^+(t, x) - \psi(t, x) \) has a local maximum at \( (t_0, x_0) \)

\[
V^+(t_0, x_0) - \psi(t_0, x_0) \geq V^+(t, x) - \psi(t, x) \tag{25}
\]

According to Theorem 3.4, there has \( h_0 \) for any \( h \in (0, h_0) \)

\[
\min_{u^p_0 \in U_p} \max_{u^e_0 \in U_e} \left\{ \int_{t_0}^{t_0+h} \ell(t, x_t, u_p, u_e) dt + \varphi(t_0 + h, x(t_0 + h)) - \varphi(t_0, x_0) \right\} \geq 0 \tag{26}
\]

Where, for any given behavior \( u_p \) has

\[
\max_{u^e_0 \in U_e} \left\{ \int_{t_0}^{t_0+h} \ell(t, x_t, u_p, u^e_0) dt + \varphi(t_0 + h, x(t_0 + h)) - \varphi(t_0, x_0) \right\} \geq 0 \tag{27}
\]

So for any \( \varepsilon \) and \( h \), there is a \( u^e_0 \) that satisfies

\[
\int_{t_0}^{t_0+h} \ell(t, x_t, u_p, u^e_0) dt + \varphi(t_0 + h, x(t_0 + h)) - \varphi(t_0, x_0) \geq \varepsilon h \tag{28}
\]
From Assumption 3.5, we can know that $f$ is bounded and $\ell$ is a uniformly continuous function, then we can get

$$
\left| \int_{t_0}^{t_0+h} \ell(t, x_t, u_p, u_e^*) dt - \int_{t_0}^{t_0+h} \ell(t_0, x_0, u_p, u_e^*) dt \right| \\
\leq \int_{t_0}^{t_0+h} |\ell(t, x_t, u_p, u_e^*) - \ell(t_0, x_0, u_p, u_e^*)| dt \\
\leq \int_{t_0}^{t_0+h} lip(\ell) (|t - t_0| + |x - x_0|) dt \\
\leq \int_{t_0}^{t_0+h} lip(\ell) (|t - t_0| + ||f||_\infty |x - x_0|) dt = o(h) \tag{29}
$$

Since $\varphi$ is a function of formula (18), then has

$$
\varphi(t_0 + h, x(t_0 + h)) - \varphi(t_0, x(t_0)) = \int_{t_0}^{t_0+h} \{\partial_t \varphi(t, x_t) + D\varphi(t, x_t) \cdot f(t_t, x_t, u_p, u_e^*)\} dt \tag{30}
$$

According to the Lipschitz continuity of $f(t_t, x_t, u_p, u_e^*)$ and the continuity of $\partial_t \varphi(t, x_t)$ and $D\varphi(t, x_t)$, it can be known

$$
\left| \int_{t_0}^{t_0+h} \partial_t \varphi(t, x_t) dt - h\partial_t \varphi(t, x_t) \right| \leq o(h) \tag{31}
$$

$$
\left| \int_{t_0}^{t_0+h} D\varphi(t, x_t) \cdot f(t_t, x_t, u_p, u_e^*) dt - \int_{t_0}^{t_0+h} D\varphi(t_0, x_0) \cdot f(t_0, x_0, u_p, u_e^*) dt \right| \leq o(h) \tag{32}
$$

We substitute formulas (29), (31) and (32) into formula (28), then has

$$
\int_{t_0}^{t_0+h} \{f(t_0, x_0, u_p, u_e^*) + D\varphi(t_0, x_0) \cdot f(t_0, x_0, u_p, u_e^*)\} dt + h\partial_t \varphi(t_0, x_0) \geq o(h) - \varepsilon h \tag{33}
$$

We generalize this to behavioral strategies in general

$$
\int_{t_0}^{t_0+h} \max_{u_e^* \in U_e} \{f(t_0, x_0, u_p, u_e^*) + D\varphi(t_0, x_0) \cdot f(t_0, x_0, u_p, u_e^*)\} dt + h\partial_t \varphi(t_0, x_0) \geq o(h) - \varepsilon h \tag{34}
$$

Divide both sides of formula (34) by $h(h > 0)$ to get

$$
\frac{1}{h} \int_{t_0}^{t_0+h} \max_{u_e^* \in U_e} \{f(t_0, x_0, u_p, u_e^*) + D\varphi(t_0, x_0) \cdot f(t_0, x_0, u_p, u_e^*)\} dt + \partial_t \varphi(t_0, x_0) \geq \frac{o(h)}{h} - \varepsilon \tag{35}
$$

When $\varepsilon \to 0, h \to 0$, then has

$$
\max_{u_e^* \in U_e} \{f(t_0, x_0, u_p, u_e^*) + D\varphi(t_0, x_0) \cdot f(t_0, x_0, u_p, u_e^*)\} \geq 0 \tag{36}
$$

Because of the arbitrariness of $u_p$, there is

$$
\min_{u_p \in U_p} \max_{u_e^* \in U_e} \{f(t_0, x_0, u_p, u_e^*) + D\varphi(t_0, x_0) \cdot f(t_0, x_0, u_p, u_e^*)\} \geq 0 \tag{37}
$$

According to the weak solution[6] of hamilton-bellman-isaacs formula for any test function $\varphi(t, x) \in ([0, T] \times \mathbb{R}^n)$, there is a local maximum value of $V^+(t, x) - \varphi(t, x)$
at the point \((t_0, x_0)\), and it satisfies formula (37), then \(V^+\) is a valid solution of formula (23).

Similarly, for the lower value \(V^-(t, x)\) at time \(t\) in state \(x\), the hamilton-bellman-isaacs formula can be expressed as

\[
\frac{\partial}{\partial t} V^- + H^-(t, x, DV^-) = 0
\]  

(38)

Where \(H^-(t, x, \zeta) = \max_{u_p \in U_p} \min_{u_e \in U_e} \{\ell(t_0, x_0, u_p, u_e) + \zeta f(t_0, x_0, u_p, u_e)\}\)  

(39)

For any test function \(\phi(t, x) \in (0, T] \times R^n\), there is a local minimum value for \(V^-(t, x) - \phi(t, x)\) at \((t_1, x_1)\), and it satisfies formula (40), then \(V^-\) is a valid solution to formula (39).

\[
\max_{u_p \in U_p} \min_{u_e \in U_e} \{\ell(t_1, x_1, u_p, u_e) + \partial_t \phi(t_1, x_1) + D\phi(t_1, x_1) f(t_1, x_1, u_p, u_e)\} \leq 0
\]  

(40)

Under the conditions of Assumption 3.1 and Assumption 3.5 in this paper, according to the theoretical derivation and proof of [10], formula (41) is valid

\[
H^+(t, x, \zeta) = H^-(t, x, \zeta), \forall (t, x, \zeta) \in [t_0, t_f] \times R^{2n} \times R^{2n}
\]  

(41)

According to formulas (12) and (13), \(V^+ \geq V^-\) and \(V^+ \leq V^-\) are always true. Therefore, \(V = V^+ = V^-\) is satisfied for any effective game.

**Theorem 3.7.** Under the conditions of Assumption 3.1 and Assumption 3.5, if there are multiple Nash equilibrium strategies in the pursuit-evasion game problem, different Nash equilibrium strategies will correspond to the same value.

4. TSK fuzzy inference model of pursuit-evasion. Spacecraft rendezvous with non-cooperative target is carried out in the continuous state space, but the traditional Deep Q Networks algorithm may cause the dimension disaster due to its difficulty in dealing with the continuous state space and the large continuous state space and behavior space[34]. To avoid this problem, we construct a fuzzy inference model of spatial behavior. In order to realize the mapping transformation from continuous state through fuzzy reasoning to continuous behavior output, it is beneficial to give play to the advantage of discrete behavior algorithm of Deep Q Networks.

Zero-order takagi-sugeno-kang (TSK) [39] is usually used as the most commonly fuzzy inference model. After representing the continuous behavior through the Membership Function (MF)[16], TSK can obtain the mapping relationship between fuzzy set and output linear function by using IF-THEN[7] fuzzy rule.

\[R_l: \text{IF } x_1 \text{ is } A^l_1 \text{ AND } x_2 \text{ is } A^l_2 \text{ AND } \cdots \text{ AND } x_i \text{ is } A^l_i \text{ THEN } u_l = c_l\]

(42)

Where \(R_l\) represents the rule \(l(l = 1, \cdots, L)\) in the fuzzy inference model. \(x_i(i = 1, \cdots, n)\) represents the input variable passed to the fuzzy model. \(A^l_i\) represents the fuzzy set corresponding to the input variable \(x_i\). \(u_l\) represents the output function of rule \(R_l\). \(c_l\) represents a constant that describes fuzzy concentration[8].

The Fig.3 shows the spatial behavior fuzzy inference model when the input is \(n = 2\) and the membership function \(y = 3\). The model is a 5-layers network structure, in which small circles represent variable nodes and small boxes represent operation nodes.
In general, it is assumed that there are \( n \) continuous spatial variables \( x_i (i = 1, \cdots, n) \) as input. After \( y \) membership functions are applied to each variable \( x_i \), the precise output \( u \) can be obtained through the process of fuzzification and defuzzification. The functions of each layer are described below.

In the first layer of the network, there are \((n \cdot y)\) adaptive output nodes after the input variables are processed by the fuzzy function. According to formula (42), the output of each node is the membership degree \( \mu_{A^l_i} \) of its input variable \( x_i \). When the spatial behavior fuzzy inference model is mapped by if-then fuzzy rules, we replace \( c_l \) in formula (42) with \( a_l \).

\[
R_l: \text{IF } x_1 \text{ is } A_1^l \text{ AND } x_2 \text{ is } A_2^l \text{ AND } \cdots \text{ AND } x_i \text{ is } A_i^l \text{ THEN } u_l = a^l_l
\]  

(43)

Where \( a^l_l \) represents act corresponding to rule \( l \) in the discrete acts \( a = \{a^1, a^2, \cdots, a^L\} \).

In the second layer of the network, we adopt direct product reasoning[16] for fuzzy sets, that is, crossmultiply each membership degree at \( L(L = y^n) \) operation nodes.

\[
L^2_l = \prod_{i=1}^{n} \mu_{A^l_i}(x_i)
\]  

(44)

Where \( \mu_{A^l_i} \) represents the membership degree of \( A^l_i \), and its function represents usually described graphically.

Gauss membership function is widely used in fuzzy inference model because of its simple formula and high computational efficiency. Gauss membership function can be expressed as

\[
\mu_{A^l_i}(x_i) = \exp \left( -\left( \frac{x_i - m^l_i}{\sigma^l_i} \right)^2 \right)
\]  

(45)

Where \( m^l_i \) represents the mean of the Gaussian membership. \( \sigma^l_i \) represents the variance of the Gaussian membership.
In the third layer of the network, in order to realize the weighted average de-fuzzification, we carry on the normalization processing to the membership degree.

\[ L_3^l = \Psi^l = \frac{L_2^l}{\sum_{l=1}^L L_2^l} = \frac{\prod_{i=1}^n \mu_{A_i}(x_i)}{\sum_{l=1}^L \left( \prod_{i=1}^n \mu_{A_i}(x_i) \right)} \]  

(46)

In the fourth layer of the network, we introduce act \( a^l \) to dot each node.

\[ L_4^l = L_3^l \cdot a^l \]  

(47)

In the fifth layer of the network, we carry on the cumulative processing to the node, then we can convert the fuzzy quantity into the exact quantity\[32\].

\[ L_5^l = u = \sum_{l=1}^L L_4^l = \frac{\sum_{l=1}^L \left( \prod_{i=1}^N \mu_{A_i}(x_i) \cdot a^l \right)}{\sum_{l=1}^L \left( \prod_{i=1}^N \mu_{A_i}(x_i) \right)} = \sum_{l=1}^L \left( \Psi^l \cdot a^l \right) \]  

(48)

5. Pursuit-evasion game based on branching improved Deep Q Networks.

When Deep Q Networks are directly applied to the fuzzy inference model of pursuit-evasion game, they will face the combinatorial growth problem of behavior quantity and mapping rules, which will greatly weaken the behavior control decision-making ability after discretization. In addition, the naive distribution of value functions and the policy representation across multiple independent function approximators also encounter many difficulties, resulting in stability problems\[26\].

Therefore, we improve the original single neural network estimation method into double Q network estimation to alleviate the overestimation problem, thus achieving the stability improvement of Deep Q Networks algorithm. We distribute the representation of state behavior value function on multiple network branches. We realize independent training and fast processing of discrete behavior through multiple groups of parallel neural networks. While sharing a behavior decision module, we decompose the state behavior value function into state function and advantage function to realize an implicit centralized coordination. We construct the game interaction between spacecraft and non-cooperative target, so the stability of the algorithm and the convergence of good strategies can be realized by proper training.

5.1. Multiple groups of parallel Deep Q Networks. Based on the idea of improving stability and convergence of Deep Q Networks, we construct L groups of parallel neural Networks in Deep Q Networks. Based on the spatial behavior fuzzy inference model, the representation of state behavior value function is distributed on several network branches. According to the spatial behavior fuzzy inference model, representation of the state behavior value function is distributed on several network branches. Multi-group parallel neural networks are the addition of multi-group parallel neural networks on the basis of single neural networks. Similar to single neural network\[17\], parallel neural networks can train and make independent decision in continuous interaction with the environment. Different from single-group neural network, multi-group parallel neural networks which combined with game and feedback mechanism can realize distributed training and decision-making with different network parameters at the same time. It has stronger autonomy, flexibility
and coordination, which will greatly enhance the independent learning ability of discrete behaviors and enhance the overall exploration ability of the environment.

Figure 4. Branching Deep Q Networks architecture

The multiple groups of parallel neural Networks in the architecture of branching Deep Q Networks are shown in Fig.4. In figure, actor first obtain current state through environmental perception, next obtain behavioral value function through multiple groups of parallel network branches training, and then make behavioral decisions and determine next act through shared behavior decision model. Finally, act is applied to environment to generate new state and effect, which is then fed back to decision model.

In multiple groups of parallel network branches, each neural network is composed of input layer, hidden layer and output layer. When the state information is respectively input into the L groups parallel neural networks, the forward transmission and the reverse training of gradient descent are carried out independently through the excitation function, and behavioral value function of discrete behavior (referred to as q-function) can be obtained by output.

5.2. Sharing behavior decision based on improved Deep Q Networks. For the fuzzy inference model with $n$ inputs and $y$ membership functions, $y^n$ possible q-functions need to be considered simultaneously if the traditional Deep Q Networks algorithm is used directly. This makes the algorithm of Deep Q Networks difficult and even hard to explore effectively in the application of multiple discrete behaviors[21].

In addition, Q learning in Deep Q Networks maximize on the basis of estimated values, which can be seen as an implicit estimation of the maximum value, and this processing will produce a significant reward bias. In particular, when the model is unstable in the training process, the bias will lead to the bias of the model in judging behavior, thus affecting convergence of model. Moreover, since Deep Q Networks itself may have prediction errors, using its maximum approximation behavior value function each time will make behavior value function correct in the direction of maximum error. In this way, after repeated iteration, error will be magnified, resulting in the final convergence of behavior value function value is much higher.
than real value. Thus, the problem of overestimation will affect stability of the network.

![Diagram](image)

**Figure 5.** Sharing behavior decision diagram based on improved Deep Q Networks

Therefore, we build a sharing behavior decision model based on improved Deep Q Networks, which is shown in Fig. 5. The main idea is described below. Firstly, we change the original single neural network estimation to double Q network estimation. Secondly, we decompose the output q-function of multiple parallel neural networks into state value and advantage value. Thirdly, state value and behavioral advantage of individual branches are evaluated separately. Finally, through a special aggregation layer, we combine state value and advantage value to output continuous space behavior strategy. The detailed algorithm is described below.

At the state input, in order to effectively solve the overestimation problem in Deep Q Networks, we divide the behavioral value function into two parts and calculate the behavioral value separately, so as to obtain network $q_1$ and network $q_2$. Among them, we use one of the networks to get the maximum behavior

$$A^* = \arg\max_a q_1(a)$$  \hspace{1cm} (49)

We use another network to determine its estimate of value

$$q_2(A^*) = q_2(\arg\max_a q_1(a))$$  \hspace{1cm} (50)

Since $E[q_2(A^*)] = q(A^*)[31]$, $q_2(A^*)$ represents an unbiased estimate.

We swap the processes of formula (49) and formula (50), that is, swap the roles of $q_1$ and $q_2$, then another unbiased estimator can be obtained

$$q_1(A^*) = q_1(\arg\max_a q_2(a))$$  \hspace{1cm} (51)

In the behavior decision stage, the state value can measure the behavior state in a particular state, and the q-function can measure the value of choosing a particular behavior in this state. Based on this, the difference between q-function and state value is defined as advantage value

$$o_t(S,a^t) = q_t(S,a^t) - V_t(S)$$  \hspace{1cm} (52)

In theory, the dominance function can obtain a relative measure of the importance of each action and satisfy $E_{a \sim \epsilon\text{-greedy}}[o_t(S,a^t)] = 0$. However, since q-function is only an estimate of state value, it is impossible to specify state value and advantage value.
Therefore, q-function is decomposed into a state value and an advantage value by using the property that expected value of advantage function is 0.

\[ q_t(S, a^l) = v_t(S) + \left( \alpha_t(S, a^l) - \max_{a' \in a} q_t(S, a') \right) \]  

(53)

On the behavior output side, state value independent of behavior selection can be separated. Only after optimizing each dominant value, formula (47) can be output through the full connection layer to get formula (54). This processing not only alleviates the computation of q-function, but also avoids the combinatorial growth of behavior quantity and mapping rule, thus ensuring the rapid convergence of model and high output efficiency.

\[ u^*(\bar{x}_t) = v_t(S) + \sum_{l=1}^{L} \left( \Psi_l \cdot q_t(S, a^l) - \Psi_l \cdot \max_{a^l \in a} q_t(S, a^l) \right) \]  

(54)

Where \( u^*(\bar{x}_t) \) represents the global behavior with optimal q-function in rules.

In the algorithm of Deep Q Networks, after best behavior \( a^* \) of state \( S_{t+1} \) is selected according to update of time-series differential target, same parameter \( \theta_t \) is usually used to select and evaluate behavior[2]. However, in this way, maximum value of estimated value will be regarded as the maximum estimate of real value, resulting in overestimation phenomenon. Although the goal of Deep Q Networks is to find optimal strategy, non-uniform occurrence of overestimation phenomenon will cause value function to overestimate and affect decision, thus leading to final decision is not optimal but suboptimal[27].

In order to reduce the influence of maximum error, we introduce another neural network based on double Q learning model, select and update behaviors with different value functions respectively.

Therefore, we use parameter \( \theta_t \) to conduct behavior selection through formula. After selecting the best behavior \( u^* \), we use parameter \( \theta_t^- \) of another neural network to conduct behavior update[13]

\[ Y_t^{DQ} = R_{t+1} + \gamma q(S_{t+1}, u^*; \theta_t^-) \]  

(55)

According to formulas (49) and (50), a new time-series difference (TD) target formula can be obtained by modification

\[ Y_t^{DQ} = R_{t+1} + \gamma q(S_{t+1}, \max_a q(S_{t+1}, u; \theta_t); \theta_t^-) \]  

(56)

Where: \( \gamma \in [0, 1] \) represents the discount factor. \( R_{t+1} \) represents the reward and punishment value available at time \( t + 1 \), and define \( R_{t+1} = 2e^{-u^2} - 1 \).

The double Q learning model adopted by us can improve the way of temporal difference and reduce the overestimation of behavioral value function. This method can help the method to select a better allocation scheme, so as to achieve a better execution effect and further improve stability of the algorithm.

5.3. The game process of the pursuit-evasion between spacecraft and non-cooperative target. We describe the problem of space intersection with non-cooperative target as a game problem by differential strategy. We obtain the continuous behavior output by using the pursuit-evasion game algorithm based on branching improved Deep Q Networks. Taking the spacecraft as an example, as shown in Fig.6, we show the dynamic game interaction
Process 1. According to the current state $S$ of the spacecraft, the input quantity $n$ is set, and the membership function $y$ is set. According to the number of fuzzy rules, the neural network $L(L = y^n)$ is defined, and the q-function of each network is randomly initialized.

Process 2. Take the spacecraft current state $\bar{x} = (x_1, x_2, \cdots, x_n)$ as input and mapping it to rule $L$ through IF-THEN.

Process 3. Respectively calculate q-function in the neural network, and select discrete behavior $a^l(l = 1, 2, \cdots, L)$ according to formula (49).

Process 4. Substitute $c_l$ for $a^l$ in formula (43) according to formula (42). According to the fuzzy inference model of Formulas (44) to (48), combined with the extraction of the dominance function, spacecraft will obtain the behavior in the current state $u(\sqrt{x_t})$. Spacecraft takes act to move to a new position $S + 1$.

Process 5. Calculate the Euclidean distance between spacecraft and non-cooperative target. Judge whether the rendezvous condition is satisfied. If so, let variable $Done = 1$ and go to process 10. If not, go to process 6.

---

**Figure 6.** The interactive flow of pursuit-evasion game
Process 6. The variable $Done = 0$. The non-cooperative target takes the most advantageous behavior according to the escape strategy and moves to the new position $P + 1$.

Process 7. Calculate reward based on the behavior $u$ and change of position. In each branching network, Combine current state $S$, discrete behavior $a'$, reward $R$ and the next state $S + 1$ into a matrix $[S, a', R, S + 1]$ and store it in the memory bank[31, 14].

Process 8. Conduct autonomous reinforcement learning in shared behavior decision module. According to formulas (52) to (56), take error $p_t$ as traction, take a certain learning rate $\eta$, and update $q$-function.

Process 9. Judge whether the number of steps reaches maximum action step $M$. If so, go to process 10. Otherwise, add 1 to number of steps and go to process 2.

Process 10. End interaction process of this round of pursuit-evasion game.

6. Example analysis. We assume that both spacecraft and non-cooperative target are in low earth orbit, the mass of spacecraft is 2500kg, the constant thrust is 0.05N, and the space steering angle range is $\delta_p \in [-0.4, 0.4]$ and $\theta_p \in [-0.5, 0.5]$. We assume that admissible strategy sets $u_p$ and $u_e$ are compact sets in some metric space, and $J : u_p \times u_e \rightarrow \mathbb{R}$ is a continuous function from $u_p \times u_e$. We assume that $f$ is a bounded function that satisfies Lipschitz continuity. Considering that fuel consumption is small relative to spacecraft and non-cooperative target mass, we assume that the spacecraft’s non-cooperative target mass remains constant throughout the maneuver process. We assume that initial orbital altitude at origin of the coordinates is 500km. Since pursuit-evasion game takes place in low earth orbit, we approximate orbital angular velocity of the reference star to a constant $\omega = \sqrt{\mu/r^3}$. We set that orbital altitude is $r = 6871$km and the gravitational constant is $\mu = 3.986 \times 10^5$km$^3$/s$^2$ and the maximum number of action steps in game process is $M = 3600$. The initial state parameters of spacecraft and non-cooperative target are shown in Tab.1.

| Table 1. The initial state of the spacecraft and the non-cooperative target |
|---|---|---|---|---|---|
| | $x$ | $y$ | $z$ | $\dot{x}$ | $\dot{y}$ | $\dot{z}$ |
| P | 0 | 0 | 0 | -0.0563 | 0.0418 | 0 |
| E | 70 | 70 | 0 | -0.0425 | 0.0314 | 0 |

The space angle difference between spacecraft and non-cooperative target is composed of $\Delta \delta$ and $\Delta \theta$, that is $\varphi = [\Delta \delta, \Delta \theta]$

$$
\begin{align*}
\Delta \delta &= \delta_e - \delta_p \\
\Delta \theta &= \tan^{-1}(\frac{y_e - y_p}{x_e - x_p}) - \theta_p
\end{align*}
$$

The change rate of the angle difference is

$$
\dot{\varphi} = \frac{\varphi - \varphi'}{T}
$$
Where $\varphi'$ represents angle difference from the previous state. $T$ represents sampling time.

When spacecraft is approaching non-cooperative target, the non-cooperative target will take escape behavior. In order to better reflect the whole game interaction process, we use angle difference and its rate of change as state quantity, that is $S = (\varphi, \dot{\varphi})$ and $P = (\varphi, \dot{\varphi})$. To avoid the dimension disaster, we set the fuzzy inference model with input $n = 2$ and membership function $y = 3$. The fuzzy set of degree difference and its rate of change is denoted by \{negative(N), zero(Z), positive(P)\}.

Our experimental environment mainly includes computing platform and environment configuration. The detailed parameters of its environment configuration are shown in Tab.2. In the branching improved deep reinforcement learning, the number of neural network layers is 3, the number of neurons in the hidden layer is 10, and the activation function is sigmoid, and the explore rate is $\varepsilon = 0.3$, and the discount factor is $\gamma = 0.9$, the learning rate is $\eta = 0.3$, the sampling time is $T = 1s$.

**Table 2.** Experimental environment parameters

| computing platform | environment configuration |
|--------------------|-------------------------|
| CPU                | Intel Core i5-7300H QCPU @2.50GHz |
| RAM                | 8 GB                     |
| System             | Windows 10               |
| Programming language | Python 3.6               |
| Compiling environment | Pycharm 2018            |
| Deep Learning framework | TensorFlow 0.12.0       |

Through simulation comparison, our algorithm has the comparative advantage of continuous spatial behavior decision application. We adopt the same $\varepsilon$-greedy strategy, respectively using the improved algorithm and the traditional Deep Q Networks algorithm for comparison.

**Figure 7.** The error function value comparison of the two algorithms
The comparison of error function values of the two algorithms is shown in Fig. 7. In figure, the improved algorithm adopts the method of distinguishing the full connection layer, enabling the training effect of error 0.1 to be achieved after only 200 times of learning. The error function value of the whole training process also decreases twice as fast, and the improvement effect in the convergence is obvious.

The reward value comparison of the two algorithms is shown in Fig. 8. The improved algorithm introduce another neural network in the behavior estimation to ensure that the reward value rise rapidly while the fluctuation is less. After only 800 times of autonomous learning, the improved algorithm can remain near the optimal reward value of 197.8, which fully reflect the improvement advantage in stability.

![Fig. 8.](image1.png)

**Figure 8.** The reward value comparison of the two algorithms

Therefore, according to the error value of the algorithm and the change of reward value with the number of training times, we can well grasp the relationship between training efficiency and training time.

![Fig. 9.](image2.png)

**Figure 9.** The pursuit-evasion trajectory after learning 0 times
Simulation result show that our algorithm can effectively solve the pursuit-evasion game between spacecraft and non-cooperative target. In the algorithm training process, the pursuit-evasion trajectory after learning 0 times is shown in Fig.9. In figure, although the spacecraft is driven by the objective function, it’s q-function is randomly generated and there is no prior knowledge, resulting in uncertain behavior and floating back and forth. Non-cooperative target will continue on original orbit because of no threat. Finally, spacecraft will be farther and farther away from non-cooperative target, unable to complete the mission.

After 400 times of autonomous learning, the pursuit-evasion trajectory is shown in Fig.10. In figure, spacecraft can approach non-cooperative target, and non-cooperative target take evasive action to change its orbit. After the two sides playing pursuit-evasion game 1798s, spacecraft finally realizes the space rendezvous with non-cooperative target.

Figure 10. The pursuit-evasion trajectory after learning 400 times

Figure 11. Probability distribution of pursuit-evasion behavior
After 800 times of autonomous learning, the probability distribution of the two actors’ pursuit-evasion behavior is shown in Fig.11. In figure, spacecraft can better deal with the escape behavior of non-cooperative target, and after playing with the non-cooperative target for a period of time, the mutual behavior tends to be stable soon.

Finally, driven by the equilibrium strategy, spacecraft is able to choose the best trajectory and achieve space rendezvous with non-cooperative target after the shortest time of 1236s. The pursuit-evasion trajectory is shown in Fig.12. It can be seen from the figure that the trajectories of the two actors in the Z direction did not change significantly, which conforms to the conclusion that the optimal pursuit strategy for spacecraft and non-cooperative target should occur in the coplanar orbit during the pursuit-evasion process[35].

7. Conclusion. In this paper, in order to solve the space rendezvous problem between spacecraft and non-cooperative target and alleviate the application limitation of Deep Q Networks in continuous space, we propose a pursuit-evasion strategy solution algorithm based on branching improved Deep Q Networks. First of all, we construct the tracking motion model of spacecraft in low Earth orbit and transfer the strategy of space rendezvous for non-cooperative target to a differential game problem. Next, we give the Nash equilibrium strategy of pursuit-evasion game, and test its existence and uniqueness, so as to satisfy solving conditions of the zero sum differential game. Then, we construct a fuzzy inference model of pursuit-evasion behavior, and realize mapping transformation of continuous state through fuzzy inference to continuous behavior output, which effectively avoid the dimensional disaster of Deep Q Networks in dealing with continuous space. Finally, we improve Deep Q Networks and propose a new branching architecture of Deep Q Networks, which realize branching training and shared decision-making of behavioral strategies, and effectively solves the problem of combination growth of behavioral number and mapping rules.

All in all, this paper realizes the combination of optimal control and game theory, further improves Deep Q Networks’s learning ability of discrete behaviors, and effectively solves the problem that differential game models are highly non-linear
and difficult to solve by using classical optimal control theory. At the same time, it has a strong reference significance for solving the problem of pursuit-evasion game in other fields.

REFERENCES

[1] G. M. Anderson and V. W. Grazier, Barrier in pursuit-evasion problems between two low-thrust orbital spacecraft, *AIAA J.*, 14 (1976), 158–163.

[2] J. Ba, V. Mnih and K. Kavukcuoglu, Multiple Object Recognition with Visual Attention, In ICLR, 2015, arXiv:1412.7755.

[3] E. N. Barron, L. C. Evans and R. Jensen, Viscosity solutions of Isaacs’ formulas and differential games with Lipschitz controls, *J. Differential Equations*, 53 (1984), 213–233.

[4] Y. L. Chen, Research on Differential Games-Based Finite-Time Adaptive Dynamic Programming Guidance Law, Nanjing University of Aeronautics and Astronautics, 2019.

[5] Y. Cheng, Z. Sun, Y. Huang and W. Zhang, Fuzzy categorical deep reinforcement learning of a defensive game for an unmanned surface vessel, *International Journal of Fuzzy Systems*, 21 (2019), 592–606.

[6] M. G. Crandall, L. C. Evans and P.-L. Lions, Some properties of viscosity solutions of Hamilton-Jacobi equations, *Trans. Amer. Math. Soc.*, 282 (1984), 487–502.

[7] X. Dai, C. K. Li and A. B. Rad, An approach to tune fuzzy controllers based on reinforcement learning for autonomous vehicle control, *IEEE Transactions on Intelligent Transportation Systems*, 6 (2005), 285–293.

[8] S. F. Desouky and H. M. Schwartz, Q(λ)-learning fuzzy logic controller for a multi-robot system, in *IEEE International Conference on Systems, Man and Cybernetics*, 10 (2010), 4075–4080.

[9] J. Engwerda, Algorithms for computing Nash equilibria indeterministic LQ games, *Comput. Manag. Sci.*, 4 (2007), 113–140.

[10] A. Friedman, *Differential Games*, Rhode Island: American Mathematical Society, 1974.

[11] W. T. Hafer, H. L. Reed, J. D. Turner and K. Pham, Sensitivity methods applied to orbital pursuit evasion, *J. Guid Control Dyn.*, 38 (2015), 1118–1126.

[12] Z. W. Hao, S. T. Sun, Q. H. Zhang and Y. Chen, Application of Semi-Direct Collocation method for solving pursuit-evasion problems of spacecraft, *Journal of Astronautics*, 40 (2019), 628–635.

[13] H. V. Hasselt, A. Guez and D. Silver, Deep reinforcement learning with double q-learning, in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 02 (2016), 2094–2100, arXiv:1509.06461.

[14] M. Hessel, J. Modayil, H. H. Van et al, Rainbow: Combining improvements in deep reinforcement learning, *Association for the Advancement of Artificial Intelli-gence*, 10 (2017), 3215–3222, arXiv:1710.02298v1.

[15] R. Isaacs, *Differential Games*, New York: Wiley, 1965.

[16] J. S. R. Jang, C. T. Sun and E. Mizutani, Neuro-fuzzy and soft computing: A computational approach to learning and machine intelligence, in *IEEE Transactions on Automatic Control*, 42 (1997), 1482–1484.

[17] F. Jürgen, W. K. Härdle and C. M. Hafer, *Neural Networks and Deep Learning*, Statistics of Financial Markets, 2019.

[18] C. Y. Li, *Study on Guidance and Control Problems for Tactical Ballistic Missile Interceptor*, Dissertation for Doctoral Degree Harbin: Harbin Institute of Technology, 2008.

[19] L. Li, F. Liu, X. Shi and J. Wang, Differential game model and solving method for missile-pursuit-evasion, *Systems Engineering-Theory & Practice*, 36 (2016), 2161–2168.

[20] Z.-Y. Li, H. Zhu, Z. Yang and Y.-Z. Luo, A dimension-reduction solution of free-time differential games for spacecraft pursuit-evasion, *Acta Astronautica*, 163 (2019), 201–210.

[21] T. P. Lillicrap, J. J. Hunt, A. Pritzel et al, *Continuous Control with Deep Reinforcement Learn-Ing*, In International Conference on Learning Representa-tions, 2016.

[22] B. Liu, X. Ye, Y. Gao, X. Dong, X. Wang and B. Liu, Forward-looking imaginative planning framework combined with prioritized replay double DQN, *International Conference on Control, Automation and Robotics*, 4 (2019), 336–341.

[23] B. Liu, X. Ye, C. Zhou and B. Liu, Composite mode on-orbit service resource allocation based on improved DQN, *Acta Aeronautica et Astronautica Sinica*, 41 (2020), 323630–323630.
DQN FOR SOLVING PURSUIT-EVASION STRATEGY SOLUTION OF SPACECRAFT

[24] R. C. Loxton, K. L. Teo, V. Rehbock and K. F. C. Yiu, Optimal control problems with a continuous inequality constraint on the state and the control, Automatica J. IFAC, 45 (2009), 2250–2257.
[25] Y. Z. Luo, Z. Y. Li and H. Zhu, Survey on spacecraft orbital pursuit-evasion differential games, Scientia Sinica Technologica, 50 (2020), 1533–1545.
[26] L. Matignon, G. J. Laurent and N. Le Fort-Piat, Independent reinforcement learners in cooperative markov games: A survey regarding coordination problems, The Knowledge Engineering Review, 27 (2012), 1–31.
[27] V. Mnih, K. Kavukcuoglu, D. Silver et al, Human-level control through deep reinforcement learning, Nature, 518 (2015), 529–533.
[28] J. F. Nash, Classic in Game Theory, Beijing: China Renmin University Press, 2013.
[29] M. Pontani and B. A. Conway, Numerical solution of the three-dimensional orbital pursuit-evasion game, J. Guid Control Dyn., 32 (2009), 474–487.
[30] X. F. Qian, R. X. Lin and Y. N. Zhao, Flight Mechanics of Guided Missile, Beijing: Beijing Institute of Technology Press, 2008.
[31] S. S. Richard and G. B. Andrew, Reinforcement Learning: An Introduction (second edition), London, England: The MIT Press, 2018.
[32] T. J. Ross, Fuzzy Logic with Engineering Applications, the United States of America: John Wiley & Sons, Ltd, 2010.
[33] W. C. Ryan, G. C. Richard, P. Meir and P. Scott, Solution of a pursuit-evasion game using a near-optimal strategy, J. Guid Control Dyn, 41 (2018), 841–850.
[34] H. M. Schwartz, Multi-Agent Machine Learning: A Reinforcement Approach, Canada: John Wiley & Sons, Inc, 2014.
[35] F. Su, J. Liu, Y. Zhang et al, Analysis of optimal impulse for in-plane collision avoidance maneuver, Systems Engineering and Electronics, 40 (2018), 2782–2789.
[36] S. T. Sun, Two Spacecraft Pursuit-Evasion Strategies on Low Earth Orbit and Numerical Solution, Harbin Institute of Technology, 2015.
[37] S. Sun, Q. Zhang, R. Loxton and B. Li, Numerical solution of a pursuit-evasion differential game involving two spacecraft in low earth orbit, J. Ind. Manag. Optim., 11 (2015), 1127–1147.
[38] S. Sun, Q. Zhang, R. Loxton and B. Li, Numerical solution of a pursuit-evasion differential game involving two spacecraft in low earth orbit, J. Ind. Manag. Optim., 11 (2015), 1127–1147.
[39] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modelling ad control, IEEE Transactions on Systems, Man and Cybernetics, 15 (1985), 116–132.
[40] L. X. Wang, A Course in Fuzzy Systems and Control, New Jersey: Prentice-Hall, Inc., 1997.
[41] Z. Wang, T. Schaul, M. Hessel et al, Dueling network architectures for deep reinforcement learning, preprint, arXiv:1511.06581, 2015, 10.
[42] X. Wu, S. Liu, L. Yang and Z. Jia, A gait control method for biped robot on slope based on deep reinforcement learning, ACTA Automatica Sinica, 1-13 (2020-02-28).
[43] Q. Wu and H. Zhang, Spacecraft pursuit strategy and numerical solution based on survival differential strategy, Control and Information Technology, 04 (2019), 39–43.
[44] D. T. Yu, H. Wang and W. M. Zhou, Anti-rendezvous evasive maneuver method considering space geometrical relationship, J. Natl. Univ. Def. Technol., 38 (2016), 89–94.
[45] Q. H. Zhang, Y. Sun, M. M. Huang et al, Pursuit-evasion barrier of two spacecrafts under minute continuous radial thrust in coplanar orbit, Control Decision, 22 (2007), 530–534.

Received June 2020; revised October 2020.

E-mail address: bingyan1@outlook.com
E-mail address: 592964097@qq.com
E-mail address: ansers@163.com
E-mail address: 11196926@qq.com