Journal of Physics:
Conference Series



PAPER • OPEN ACCESS

# Low-orbit Space Debris Warning and Autonomous Collision Avoidance for Space Environment Governance

To cite this article: Liwen Yang *et al* 2025 *J. Phys.: Conf. Ser.* **3015** 012005

# Low-orbit Space Debris Warning and Autonomous Collision Avoidance for Space Environment Governance

**Liwen Yang**[1][0009-0004-1921-3695], **Jikai Wang**[2][0009-0005-2782-408X], **Jun Jiang**[3], **Xue Bai**[1,2*] [0000-0002-0776-1107], **Ming Xu**[1,2][0000-0001-6996-0577]

[1] Shenyuan Honors College, Beihang University, Beijing 100191, China

[2] School of Astronautics, Beihang University, Beijing 100191, China

[3] Chinese Society of Astronautics, Beijing 100048, China

[*]Corresponding author's e-mail: baixue1002@buaa.edu.cn

**Abstract**. With the increasing frequency of human space activities, the issue of space debris has become progressively more severe, posing a significant threat to spacecraft in orbit and highlighting the need for effective space environmental governance. This study addresses the challenges of debris warning and autonomous collision avoidance in low earth orbit by analyzing the limitations of current technologies and developing a more comprehensive problem model. In the context of space environmental governance, we design multiple optimization scenarios tailored to various avoidance situations, employing a range of reinforcement learning methods to solve and optimize these issues. These methods include the Monte Carlo Tree Search Algorithm, the Cross Entropy Method, and Proximal Policy Optimization in continuous action spaces. Our research not only provides an effective debris avoidance strategy for spacecraft but also contributes to the advancement of technological solutions for the sustainable management and protection of the space environment.

## 1. Introduction

In recent years, with the increase in the number of commercial satellite launches, the amount of low earth orbits debris has shown an explosive growth. The presence of such debris not only poses a direct threat to the safety of spacecraft in orbit, but also may trigger a chain reaction that exacerbates the deterioration of the space environment. Therefore, the need for early warning and avoidance of space debris is increasingly urgent. However, the traditional method often needs a lot of computing resources and time when calculating collision probability and risk assessment, and it is difficult to realize real-time decision making. In order to effectively deal with the threat posed by space debris, spacecraft need to have more autonomous decision-making capabilities. The development of artificial intelligence and machine learning technology provides new possibilities to solve the above problems.

In order to reduce unnecessary evasive maneuvers, methods based on the collision probability are widely used. Chan proposed a calculation method of approximate analytical expression in the 20th century[1]. Bai studied the error propagation characteristics of space targets on the basis of relative motion theory and derived the explicit expression of collision probability in circular orbit and general orbit on the basis of geometric analysis in 2013 [2]. Scholars have done a lot of research on maneuvering avoidance strategies of space debris. Alfano proposed a collision avoidance strategy using the instantaneous maneuver method[3]. Mueller constructs the optimal control standard form of the avoidance orbit control scheme and uses the discretization method of nonlinear programming to solve the problem[4][5]. With the development of artificial intelligence, more and more intelligent methods have been applied to this problem. Kim proposed a new method of applying genetic algorithm to the optimal collision avoidance strategy[6], Seong used heuristic algorithms to compare and analyze the avoidance strategy under multi-target threat[7], and Yuan studied the rendezvous avoidance of spacecraft under multiple constraints and uncertainties using reinforcement learning by constructing a discrete Markov decision model[8], Qu proposes a collision avoidance control strategy based on Deep Deterministic Policy Gradient algorithm[9], and Liu proposes a space debris avoidance strategy based on reinforcement learning proximal strategy optimization Proximal Policy Optimization algorithm and penalized proximal policy optimization algorithm[10].

This paper will discuss how to use intelligent optimization methods, especially reinforcement learning technology, to improve the autonomous response ability of spacecraft in the face of space debris threat, so as to ensure the safety of spacecraft and the sustainable development of space environment.

## 2. Warning and avoidance model

### *2.1. Physical model*

In intelligent space navigation, the motion of spacecraft and space debris can be described by Kepler motion model. In this model, six orbital elements are used to represent the orbital state of the spacecraft, including orbital semi-major axis $a$, eccentricity $e$, right ascension of ascending node $\Omega$, orbital inclination $i$, argument of perigee $\omega$, and the Mean Anomaly $M(t_0)$ of epoch time $t_0$. These parameters can fully describe the motion characteristics of the spacecraft in orbit.

For a large number of debris, the preliminary screening of high-risk collision events is a very important part in the process of satellite collision warning, which can avoid the consumption of computing power caused by high-precision early warning of a large number of targets. Perigee-apogee screening method is a screening method based on the characteristics of orbit altitude. Perigee and apogee are used as screening parameters, which represent the lowest and highest points of orbit altitude. Assuming that the orbital heights of perigee and apogee of a certain satellite are respectively ($h_p$, $h_a$), it is considered that orbits with perigee orbital heights greater than $h_a$ and apogee orbital heights less than $h_p$ are unlikely to collide with satellite orbits. The remaining orbital area after screening is shown in Figure 1 Through this method, the computational efficiency can be greatly improved. The practice has proved that with the different scale of the problem, the speed of the algorithm can be increased to 2-3 times of the original in the training process, and the effect is shown as in Figure 2.

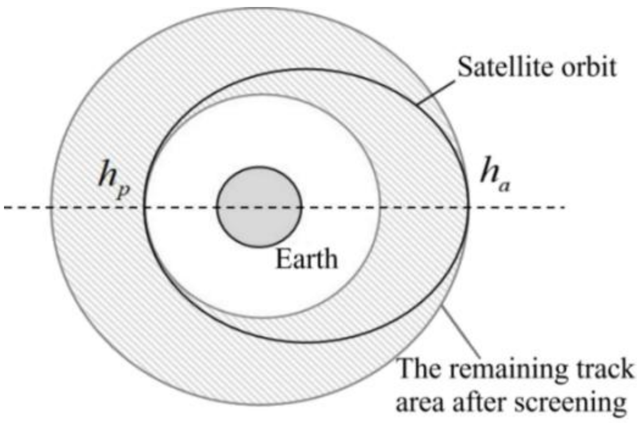
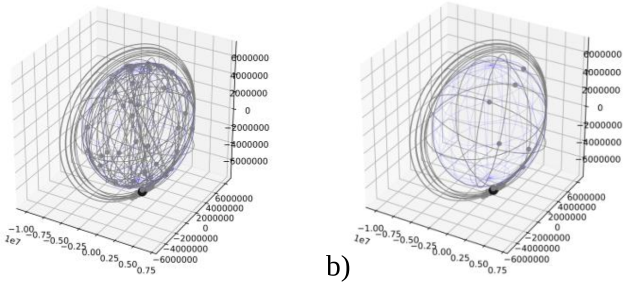



Figure 1. Schematic diagram of perigee apogee screening method



a) b)

Figure 2. Effect diagram: a) Before perigee-apogee screening; b) After perigee-apogee screening

After that, the collision probability of the remaining dangerous target is calculated in the object-centered RSW coordinate system. RSW coordinate system is established based on spacecraft, the following expression for collision probability can be obtained according to Bai's research[2]:

$$P_c = \exp\left[-\frac{1}{2}\left(\frac{R_o^2}{\sigma_R^2} + \frac{S_o^2+W_o^2}{\sigma_{SW}^2}\right)\right] \cdot \left[1 - \exp\left(-\frac{r_A^2}{2\sigma_R\sigma_{SW}}\right)\right] \tag{1}$$

It is expressed by the joint error variance of the relative distance and position at the moment of TCA closest approach in the RSW coordinate system. Where $r_A$ represents the joint radius of the spacecraft and the collision object, $R_o$, $S_o$ and $W_o$ are the components of the R, S and W directions in the RSW coordinate system, and the joint variance of the R, S and W directions is $\sigma_R$, $\sigma_S$ and $\sigma_W$ , respectively. The errors in the direction of S and W are coupled through the Angle $\varphi$ of the orbital plane, and the joint error variance is $\sigma_{SW}$.Their relationship is expressed as follows:

$$\sigma_{SW}^2 = \sigma_S^2\cos^2\frac{\varphi}{2} + \sigma_W^2\sin^2\frac{\varphi}{2} \tag{2}$$

### 2.2. Optimization model

*2.2.1. LOSS function.* Collision probability p, fuel consumption f, orbit offset r are directly related to the safety of the spacecraft. On the basis of establishing the physical model, the optimization model is further constructed, and the three are taken as optimization objectives, so as to minimize the fuel consumption and orbit offset while minimizing the collision risk. To this end, the following LOSS function is constructed as the objective function of optimization. In the optimization process, the absolute value of the loss function should be as small as possible. The expression is as follows (where $k_1$, $k_2$, and $k_3$ are weight coefficients used to balance the importance of different optimization objectives):

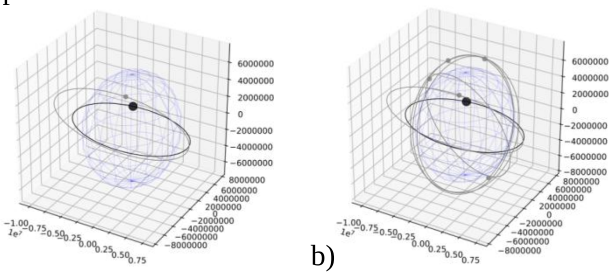
$$\mathrm{LOSS} = k_1 \cdot p + k_2 \cdot f + k_3 \cdot r \tag{3}$$

*2.2.2. Optimization scenario construction*. To better train and comprehensively verify the actual efficacy of the Collision Avoidance model constructed above, to guarantee that the model can accurately handle various complex and mutable space, the author designed a series of simulation optimization scenarios. These scenarios not only simulate a variety of typical situations that spacecraft might encounter during mission execution but also integrate orbital data of some actual debris in space, striving to approximate the complexity of the real space environment to the greatest extent. The specific optimization scenarios are as given in Table 1.

Table 1. Summary of optimization scenario

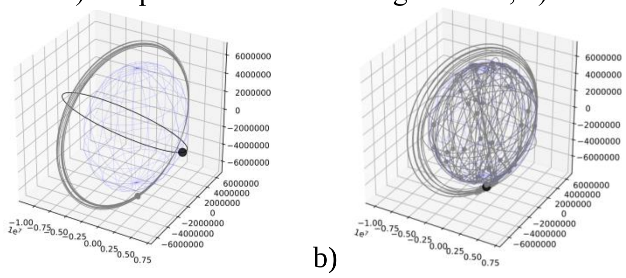
| | simple intersection of single debris | multi-level collision | debris cloud collision | reality-fused collision |
|---|---|---|---|---|
| Amount of debris | 1 | 5 | 10 | 45 |
| Amount of dangerous debris | 1 | 5 | 3-7 | 7-15 |

The first type of scenario is the simple intersection scenario of single debris, which takes into account the intersection and collision between the spacecraft and a single piece of debris. The second type of scenario is the complex collision scenario of multiple debris. We designed three kinds of multi-debris collision scenarios, including multi-level collision, debris cloud collision based on probability distribution, and reality-fused collision, as shown in Figure 3 and Figure 4. The debris will successively collide with the spacecraft at different positions in the multi-level collision scenario. The debris cloud collision scenario is designed to simulate the movement of the spacecraft in the actual debris cloud. We generate a cluster of debris with random initial velocities at specific locations in space based on a normal distribution. In the reality-fused collision scenario, the real data of the FY-1C satellite impact debris group and the above simulated debris cloud collision scene are combined.



a) b)

Figure 3. Diagram of: a) Simple intersection of single debris; b) Multi-level collision



a) b)

Figure 4. Diagram of: a) Debris cloud collision; b) Reality-fused collision

## 3. Reinforcement learning methods

### *3.1. Monte Carlo Tree Search method for continuous action space*

Monte Carlo Tree Search (MCTS) is a traditional machine learning algorithm used for decision planning, which employs Monte Carlo simulation to perform the search. Here, it is applied in reinforcement learning scenarios. In the field of artificial intelligence, especially in complex decision problems and game theory, the MCTS has proved its powerful power. In the early warning and avoidance scenario of fragments, the action space of the model is continuous, and each decision point does not select the action from a finite set. Therefore, the traditional MTCS algorithm is improved to sample the action from a probability distribution. Evaluate the potential value of different actions by building a search tree where each node represents a state and an edge represents an action that can be performed from that state. Specifically, an iteration in MCTS consists of four main steps:

A. selection: Start from the root node, select actions according to the probability distribution, and move down the tree. The selection process can be combined with UCB strategies to balance exploration and utilization.

B. Expansion: Under the selected action, expand new nodes to represent the new state of the spacecraft.

C. Simulation: Monte Carlo simulation starting from the new node, simulating a series of state transitions of the spacecraft after performing the action until a termination state is reached.

D. Backpropagation: The simulation results (such as rewards or costs) are backpropagated to nodes in the tree, updating statistics for those nodes, such as average rewards or number of visits.

### *3.2. Cross Entropy Method for continuous action space*

The Cross-Entropy Method (CEM) is a Monte Carlo method for optimization and importance sampling. Similar to evolutionary algorithms, evolutionary algorithms scatter points in space according to certain rules, obtain the error of each point, and then decide the rule of the next round of scattering points based on this error information. The cross-entropy method is so named because the goal (theoretically) is to minimize the cross-entropy (equivalent to minimizing the KL distance) between the distribution of the data obtained from random scattered points and the actual distribution of the data, so that the sample distribution (scattered points) is as distributed as possible. In the early warning and avoidance scenario of space debris, CEM algorithm finds the optimal strategy by iteratively improving the probability distribution of action, which is especially suitable for solving reinforcement learning problems with continuous action space. The algorithm process is as follows:

A. Initialization: To begin, define an initial probability distribution, such as a normal distribution, whose mean and variance are based on prior knowledge of the problem or random initialization.

B. Sample collection: Drawing a large number of samples from the current probability distribution that represent possible actions or strategies.

C. Performance evaluation: The performance of each sample is evaluated by simulation or actual execution, usually measured by calculating cumulative rewards.

D. Sample selection: Select the first small number of samples with the best performance, and the corresponding actions or strategies of these samples are considered to be approximations to the current optimal solution.

E. Update parameters: Update the parameters of the probability distribution (such as the mean and variance of the Gaussian distribution) according to the selected sample, so that the distribution is more concentrated on these good samples.

### *3.3. Proximal Policy Optimization method*

Proximal Policy Optimization (PPO) is a deep reinforcement learning algorithm based on Actor-Critic framework. In the A-C framework, two neural networks, Actor and Critic, are set up. In the space debris early warning avoidance scenario, the Actor network of A-C framework is used to determine the best avoidance action generated in A given state. The Critic network is used to estimate the value of the current policy action. The PPO algorithm is based on the strategy gradient method, its working principle block diagram is shown as Figure 5.

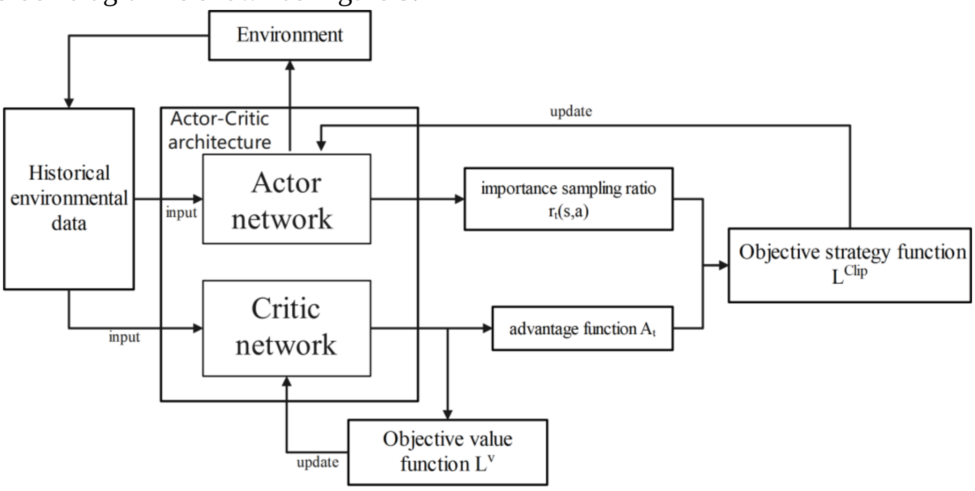



Figure 5. PPO algorithm training schematic

The algorithm is implemented in the following five steps:

A. Initialize the network: Define two neural networks, one as Actor and the other as Critic. The Actor network outputs the probability of taking various possible actions in a particular state, while the Critic network outputs an estimate of the value of the state under the current policy.

B. Interaction and Acquisition: Interact with the environment by performing actions generated by the Actor network to collect a series of experiential data, including states, actions, rewards, and the next state.

C. Compute the advantage function $A_t$: Use the Critic network to compute the advantage function $A_t$, which measures the additional value of taking an action in a particular state compared to the average strategy. In PPO, the advantage function is a key component in calculating the gradient of the strategy and is used to guide the updating of the strategy so that it is optimized in the direction of improving cumulative returns.

D. Calculation of objective function: PPO uses importance sampling ratio $r_t(s,a)$ and Clip clipping methods to assist calculation of objective strategy function $L^{Clip}$ and objective value function $L^v$ to optimize update strategy and value network. The objective value function $L^v$ is (Where $V(s_t)$ is the value function and $A_t$ is the advantage function mentioned above.):

$$L^v = (V(s_t) - A_t)^2 \tag{4}$$

Then, the objective strategy function $L^{Clip}$ is obtained using the Clip clipping method:

$$L^{CLIP} = \mathbb{E}_t[\min(r_t(s,a)A_t, \mathrm{clip}(r_t(s,a), 1-\epsilon, 1+\epsilon)A_t))] \tag{5}$$

Where the importance sampling ratio $r_t(s,a)$ is ($\pi_\theta(a|s)$ represents the probability of performing action a in state s under the new policy, and $\pi_\theta old(a|s)$ represents the probability of performing action a in state s under the old policy):

$$r_t(s,a) = \frac{\pi_\theta(a|s)}{\pi_{\theta old}(a|s)} \tag{6}$$

E. Optimize the network: compute $\nabla L^{Clip}$ and $\nabla L^v$ to optimize the target policy function $L^{Clip}$ and the target value function $L^v$, and update the parameters of the Actor network and the Critic network by the gradient ascent method.

## 4. Results

### *4.1. Simulation*

Taking the use of CEM algorithm to optimize multi-level collision of 5 debris scenarios as an example, the training time of is 127.4 seconds, and the final optimization value of loss function is -0.6581347118872334. The collision risk is successfully avoided by 7 pulse maneuvers. The seven maneuvers and maneuver-moments are shown in Table 2, Figure 6 and 7.

Table 2. Maneuver vector table of CEM

| $dV_x$/(m/s) | $dV_y$/(m/s) | $dV_z$/(m/s) | t/(s) |
|---|---|---|---|
| -0.159517 | 1.311503 | -0.728971 | 0.050073 |
| -0.353771 | 0.558027 | 0.021606 | 0.009990 |
| -0.416164 | -0.432246 | 0.507010 | 0.017773 |
| -0.804603 | -0.858479 | -0.318332 | 0.028309 |
| 0.268582 | 0.072024 | 0.101790 | 0.066975 |
| -0.418381 | -0.440039 | -0.321017 | 0.024214 |
| -0.896662 | -0.328175 | 0.320508 | 0.033512 |

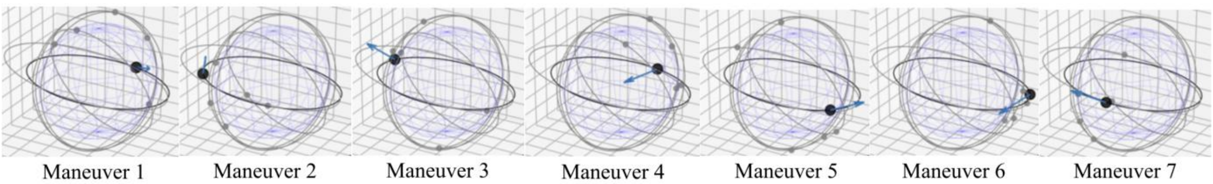



Figure 6. Maneuvers diagram of CEM optimizing multi-level collision with 5 debris

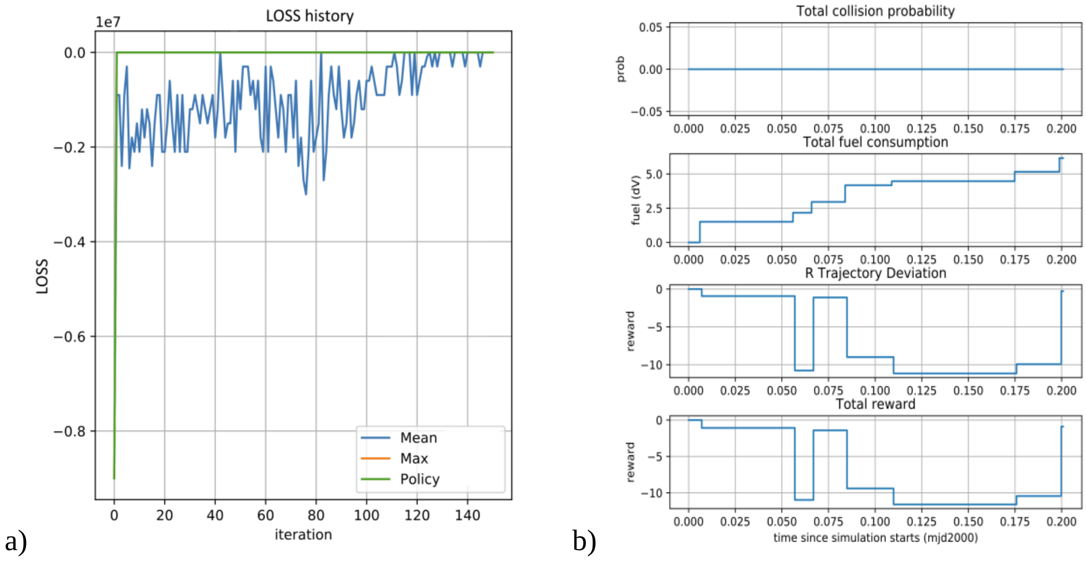



Figure 7. Diagram of: a) LOSS-value chart during training process; b) collision probability, fuel consumption, trajectory deviation and total optimization value graph

### *4.2. Comparison of the reinforcement learning methods*

As shown in Table 3, it can be seen that each algorithm has its advantages and disadvantages. MCTS, CEM and PPO methods can all complete the optimal solution of the simple intersection and the multi-level collision scenario. The remaining blank parts do not get good results and need further improvement and debugging. MCTS method can find the right balance point between exploration and utilization in continuous space, but needs a lot of simulation to accurately evaluate the action. CEM can explore the action space more quickly and carefully. However, it is sensitive to the selection of initial probability distribution. PPO significantly enhances generalization ability and transplantation ability compared with MCTS and CEM, and responds quickly after training, but it is sensitive to the choice of network structure and hyper-parameters and needs to be carefully adjusted for optimal performance.

Table 3. Summary table of the solution results of three reinforcement learning methods

| Scenario | Method | MCTS | CEM | PPO |
|---|---|---|---|---|
| simple intersection of single debris | Training time(s) | 5.46 | 17.29 | 15.16 |
| | Maneuver-times | 2 | 1 | 1 |
| | LOSS value | -0.23 | -0.02 | -0.86 |
| multi-level collision | Training time(s) | 17.76 | 127.4 | 52.582 |
| | Maneuver-times | 4 | 7 | 1 |
| | LOSS value | -0.22 | -0.66 | -0.93 |
| debris cloud collision | Training time(s) | —— | —— | 20.282 |
| | Maneuver-times | —— | —— | 1 |
| | LOSS value | —— | —— | -1.09 |
| reality-fused collision | Training time(s) | 14.04 | 326.82 | —— |
| | Maneuver-times | 1 | 1 | —— |
| | LOSS value | -0.04 | -0.03 | —— |

## 5. Conclusions

In the context of the problem of growing space debris and the limitations of existing technologies, this paper builds a model framework for this problem. Reinforcement learning algorithms, including

MCTS, CEM and PPO, are used to solve and optimize different types of fragmentation avoidance scenarios. The comparative analysis shows that PPO algorithm is a better choice in the new scenario because of its high efficiency and good portability. The study not only solves the avoidance problem of spacecraft in typical scenarios such as simple rendezvous of single debris, complex collision of multiple debris and simulated debris cloud collision, but also provides effective strategies for autonomous decision-making of spacecraft in real space environment, significantly improving the efficiency and accuracy of early warning and avoidance. Through research in these directions, we anticipate significantly enhancing the autonomous warning and avoidance capabilities of spacecraft in complex space environments and providing more reliable guarantees for the utilization of space resources. Future research will focus on optimizing the LOSS function, verifying the generalization and transplantation ability, and ensure the safety and sustainable development of human space activities.

## References

[1] Chan, K. (1997). Collision probability analysis for earth orbiting satellites. Space cooperation into the 21 st century, 1033-1048.
[2] Xian-Zong Bai,Lei Chen & Guo-Jin Tang.(2013).Explicit expression of collision probability in terms of RSW and NTW components of relative position.Advances in Space Research(6),1078-1096.
[3] Alfano, S. (2006). Collision Avoidance Maneuver Planning Tool (AAS 05-308). ADVANCES IN THE ASTRONAUTICAL SCIENCES, 123(1), 873.
[4] Mueller, J. B., Griesemer, P. R., & Thomas, S. J. (2013). Avoidance maneuver planning incorporating station-keeping constraints and automatic relaxation. Journal of Aerospace Information Systems, 10(6), 306-322.
[5] Mueller, J. (2009). Onboard planning of collision avoidance maneuvers using robust optimization. In AIAA Infotech@ Aerospace Conference and AIAA Unmanned... Unlimited Conference (p. 2051).
[6] Kim, E. H., Kim, H. D., & Kim, H. J. (2012). A study on the collision avoidance maneuver optimization with multiple space debris. Journal of Astronomy and Space Sciences, 29(1), 11-21.
[7] Seong, J. D., & Kim, H. D. (2015). Collision avoidance maneuvers for multiple threatening objects using heuristic algorithms. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering, 229(2), 256-268.
[8] Yuan, H., & Li, D. (2022). Deep reinforcement learning for rendezvous guidance with enhanced angles-only observability. Aerospace Science and Technology, 129, 107812.
[9] Qu, Q., Liu, K., Wang, W., & Lü, J. (2022). Spacecraft proximity maneuvering and rendezvous with collision avoidance based on reinforcement learning. IEEE Transactions on Aerospace and Electronic Systems, 58(6), 5823-5834.
[10] Mu, C., Liu, S., Lu, M., Liu, Z., Cui, L., & Wang, K. (2024). Autonomous spacecraft collision avoidance with a variable number of space debris based on safe reinforcement learning. Aerospace Science and Technology, 149, 109131.