aerospace

MDPI

Article

Optimal Time-to-Entry Pursuit-Evasion Games Under Sun-Angle Constraints with Non-Smooth Terminal Regions

Xingchen Li, Xiao Zhou *, Xiaodong Yu, Guangyu Zhao and Yidan Liu

Aerospace Information Research Institute, Chinese Academy of Sciences, Beijing 100094, China; lixingchen@aircas.ac.cn (X.L.); yuxd@aircas.ac.cn (X.Y.); zhaogy@aircas.ac.cn (G.Z.); liuyd@aircas.ac.cn (Y.L.)
* Correspondence: zhouxiao@aircas.ac.cn

Abstract

Recent advancements in satellite optical reconnaissance have elevated the sun angle to a critical factor in orbital pursuit-evasion games. The stringent imaging constraints imposed by sun angle and relative distance induce non-smoothness in the terminal region of such differential games, significantly complicating equilibrium-solution derivation. To address this challenge, we formulated a novel differential game model where the pursuer minimizes the time-to-entry into the evader's effective imaging region. We first constructed a sequence of low-dimensional manifolds that collectively cover the terminal region, solving the differential game with this sequence to yield the Nash equilibrium. Subsequently, we approximated the terminal region using a smooth manifold of identical dimensions, enabling a computationally efficient approximate solution. Both methodologies demonstrate broad applicability to orbital differential games featuring non-smooth terminal regions. Simulation results confirm that the approximation error becomes pronounced only under extreme initial sun angles, though this error remains acceptable for practical space reconnaissance applications.

Keywords: orbital pursuit-evasion game; differential game; sun angle

1. Introduction

Spacecraft orbital games simulate non-cooperative strategic interactions between spacecraft under orbital dynamics and operational constraints. Pioneering work by Anderson and Grazier [1] in the 1970s applied Isaacs' differential game theory [2] to low Earth orbit pursuit-evasion problems, establishing the first framework for continuous low-thrust orbital pursuit-evasion differential games. In such scenarios, the pursuer (hereinafter referred to as P) aims to intercept or rendezvous with the evader (hereinafter referred to as E), while E seeks to avoid capture. As space-domain operations gain strategic significance, orbital pursuit-evasion differential games have become a focal research area [3,4], with contemporary studies optimizing control strategies for objectives including relative distance, fuel efficiency, and game duration.

Recent advances in satellite optical imaging—exemplified by systems like the Geosynchronous Space Situational Awareness Program (GSSAP)—enable high-fidelity imaging of distant targets under precise geometric conditions [5]. Crucially, sun angle (defined as the angle between the target-sun vector and target-imager vector) critically governs image quality: smaller sun angles (where the sun is nearly behind the target relative to the imager) cause severe backscatter and glare, degrading image quality, while larger sun angles (sun illumination on the target's imaged face) yield superior resolution [6]. This technological

Check for updates

Academic Editors: Mauro Pontani and Stefano Carletta

Received: 6 March 2026
Revised: 3 April 2026
Accepted: 9 April 2026
Published: 11 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

shift elevates sun-angle-constrained pursuit-evasion scenarios from theoretical interest to operational necessity.

Consequently, research on orbital pursuit-evasion games incorporating sun-angle constraints is both timely and critical. Central questions include minimizing imaging reconnaissance time for non-cooperative targets and developing evasion strategies against adversarial imaging efforts. Addressing these challenges is essential for constellation design, space situational awareness, and mission planning, and is the core contribution of this paper.

Current research on orbital pursuit-evasion games predominantly centers on interception and rendezvous games. In interception games, P typically aims to achieve spatial coincidence with E or to enter a defined spherical region around E, a problem that has garnered significant scholarly attention. For instance, foundational work by Pontani and Conway [3] investigated remote interception between two constant-thrust spacecraft, a model later extended by Shen and Casalino [7] to include constraints on minimum orbital altitude and spacecraft mass variation. Further developments include studies by Stupik et al. [8] under Clohessy–Wiltshire dynamics, with Zhang et al. [9] advancing this framework through the application of deep neural networks. The complexity of multi-pursuer scenarios has been addressed through reachable set [10] and interception domain [11] analyses. Meanwhile, the challenge of imperfect information has been tackled using diverse methods, including compensation-based control [12], mode-matched smooth variable structure filters [13], and receding-horizon parameter optimization [14]. Model fidelity and solution methods continue to be refined, with Li et al. [15] incorporating J2 perturbations, Zeng et al. [16] proposing improved algorithms for both long- and short-range interception, and Li et al. [17] developing a multi-impulse dynamic game model with corresponding Stackelberg equilibrium solutions. Rendezvous games, a specialized subclass requiring P to match both position and velocity with E [18], have been similarly explored. Jagat and Sinclair examined these games under both linear [19] and nonlinear [20] dynamics, focusing on optimizing relative distance and fuel consumption. Moreover, reachable set methods have proven effective in optimizing rendezvous time [21] and fuel expenditure [22], providing deeper insights into strategy and performance.

To our knowledge, three prior studies address pursuit-evasion games with sun-angle constraints. Prince et al. [18] pioneered this domain by modeling sun-angle effects, defining the effective imaging region as the line extending from the evader (E) through the Sun center. However, this model lacks operational relevance: real-world imaging requires finite sun-angle ranges and imposes distance constraints—both critical for practical reconnaissance. Our work directly addresses these limitations by incorporating simultaneous sun-angle and distance constraints, significantly elevating problem complexity while enhancing realism. Li et al. [4] investigated an orbital game optimizing time spent within E's imaging region under sun-angle ranges, distance limits, and velocity constraints. Their analysis assumes P begins near or within this region and seeks to maximize dwell time—a scenario inconsistent with reconnaissance objectives where rapid target acquisition (not dwell time) is paramount. In contrast, our paper minimizes the time-to-entry into E's imaging region, directly addressing the core challenge of timely reconnaissance. Wang et al. [23] investigated a competitive imaging game regarding sun angles with discrete impulses as maneuvering methods. Their scenario involves both spacecrafts targeting each other for imaging, differing from the scenario in this paper, where one party is solely evading. In actual space reconnaissance, it is uncommon for the imaged party to also have imaging capabilities [4]. Additionally, unlike their idealized assumption of discrete impulsive maneuvers with fixed time intervals, both spacecraft in this paper utilize continuous low-thrust maneuvers. Collectively, these distinctions establish our model's operational relevance:

dual-constraint sun-angle dynamics, time-to-entry optimization, and continuous-thrust maneuvering form the foundation for practical space situational awareness applications.

This paper addresses a critical theoretical gap in orbital pursuit-evasion games under sun-angle constraints, with specific focus on time-sensitive reconnaissance scenarios. We introduce a novel differential game model where P seeks to minimize the time required to enter the evader's (E) effective imaging region. This represents the first rigorous analysis of such a model in the literature. To solve the game, we present a primary algorithm for computing the Nash equilibrium and a computationally efficient approximation algorithm for practical implementation. The equilibrium characteristics and comparative performance of both algorithms are systematically characterized.

In contrast to conventional pursuit-evasion games involving interception or rendezvous (including Prince et al.'s work [18]), where terminal regions typically form low-dimensional manifolds enabling direct transformation into two-point boundary-value problems (TPBVPs), this study confronts a bounded, non-smooth terminal region. This structural complexity significantly impedes standard solution approaches. We overcome this challenge by developing a novel solution framework for the Nash equilibrium and a corresponding approximation algorithm. Both methodological approaches demonstrate broad applicability to orbital differential games featuring non-smooth terminal regions, offering a generalizable solution strategy for similar constrained optimization problems.

In summary, this paper makes three key contributions: (1) a more realistic imaging region model with simultaneous sun-angle and distance constraints; (2) a novel time-to-entry optimization objective for the pursuer; and (3) two complementary algorithms (exact and approximate) for solving the differential game with non-smooth terminal region.

The remainder of this paper is organized as follows: Section 2 introduces the orbital pursuit-evasion differential game model with sun-angle constraints and discusses the solution problem. Section 3 presents the algorithm for solving the Nash equilibrium of the game. Section 4 describes the fast algorithm for approximate solutions. Section 5 shows simulation examples of the algorithm. Section 6 concludes the paper.

2. Differential Game Model

Consider a game scenario where P attempts to perform optical imaging of E, while E tries to avoid being imaged by P. P and E adopt a maneuvering method with continuous low thrust. Both P and E can obtain real-time information about each other's true states (including position, velocity vectors, and the magnitude of maneuvering accelerations) and can make corresponding continuous maneuvers based on their own optimization objectives until P reaches E's optical imaging region, at which point the game ends. During this process, P aims to minimize the time it takes to reach E's optical imaging region, while E seeks to maximize this time.

Let the sun center coordinates in a geocentric inertial coordinate system be denoted as $\mathrm{S} = (s_x, s_y, s_z)$. $\boldsymbol{ES}$ donates the vector from E to S. The imaging conditions required for P's optical imaging equipment are that the relative distance must be less than r_0 and the sun angle must be less than j_0. Therefore, E's imaging region is defined as a spherical cone centered at E, with a radius r_0, an axis of symmetry $\boldsymbol{ES}$, and a latitude angle range of $\left[\frac{\pi}{2} - j_0, \frac{\pi}{2}\right]$ (as shown in Figure 1). The outer surface of this conical region is denoted as ϕ. In this paper, we consistently consider the initial position of P to be outside E's imaging region to avoid trivial scenarios. Therefore, P can meet the imaging conditions of E if and only if it reaches ϕ, and the pursuit-evasion differential game terminates at this point. Let Δt represent the time taken from the initial state until P arrives ϕ.

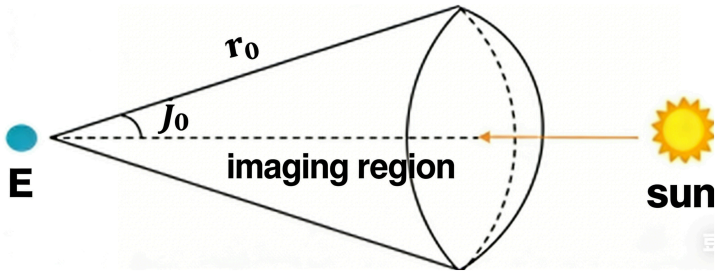


Figure 1. Imaging region ϕ.

Let P and E's position and velocity vectors in the geocentric coordinate system be represented by

$$x_p = \left[r_{px}, r_{py}, r_{pz}, v_{px}, v_{py}, v_{pz}\right]^T,\ x_e = \left[r_{ex}, r_{ey}, r_{ez}, v_{ex}, v_{ey}, v_{ez}\right]^T, \tag{1}$$

These vectors serve as the differential game's state variables. The maneuvering accelerations of P and E are denoted by $T_p,\ T_e$, with directions $\left(\cos\beta_p\cos\gamma_p, \sin\beta_p\cos\gamma_p, \sin\beta_p\right)^T$, $\left(\cos\beta_e\cos\gamma_e, \sin\beta_e\cos\gamma_e, \sin\beta_e\right)^T$.

The distances from P and E to the Earth's center are represented as r_p and r_e, leading to the following dynamic equations of two body gravitational field of the state variables:

$$\begin{aligned}
\dot{r}_{ix} &= v_{ix},\\
\dot{r}_{iy} &= v_{iy},\\
\dot{r}_{iz} &= v_{iz},\\
\dot{v}_{ix} &= -\tfrac{\mu}{r_i^3} r_{ix} + T_i\cos\beta_i\cos\gamma_i,\\
\dot{v}_{iy} &= -\tfrac{\mu}{r_i^3} r_{iy} + T_i\cos\beta_i\sin\gamma_i,\\
\dot{v}_{iz} &= -\tfrac{\mu}{r_i^3} r_{iz} + T_i\sin\beta_i.
\end{aligned} \tag{2}$$

In (2), $i = \{p, e\}$. Denote the right side of the 12 equations in (2) as $f_j, j = \{1, 2, \ldots, 12\}$. Compared to studies that use relative coordinate systems [8,9] with Clohessy–Wiltshire linearization, this formulation inherently accommodates game scenarios with larger initial distances, and any perturbation terms—such as J2 effects, solar radiation pressure, or Earth–Moon gravitational influences—can be directly added to the right-hand side of (2) without affecting the subsequent solution framework. However, to focus on the core theoretical challenge posed by the non-smooth terminal region, the present study adopts an idealized two-body gravitational model. The decision variables for this game are the maneuvering acceleration direction angles of both players (γ_i, β_i), $i = \{p, e\}$, where (γ_i, β_i) is a continuous function of time over the interval $[0, 2\pi)$. The assumption of continuous direction angles throughout the entire space originates from satellite propulsion systems that possess the capability for controlling the thrust vector direction [3,8]. To meet the Nash equilibrium conditions of the game, these decision variables satisfy the following conditions:

$$\begin{aligned}
\left(\gamma_p^*, \beta_p^*\right) &= \mathrm{argmin}_{\gamma_p, \beta_p}\left[\Delta t\left(\gamma_p, \beta_p, \gamma_e^*, \beta_e^*\right)\right],\\
\left(\gamma_e^*, \beta_e^*\right) &= \mathrm{argmax}_{\gamma_e, \beta_e}\left[\Delta t\left(\gamma_p^*, \beta_p^*, \gamma_e, \beta_e\right)\right].
\end{aligned} \tag{3}$$

In the optimization theory, to solve the optimization problem of the state variables satisfying the dynamic equations, it is necessary to introduce the costate variables with the

same dimensions, which are denoted as $\lambda_{i1}, \lambda_{i2}, \lambda_{i3}, \lambda_{i4}, \lambda_{i5}, \lambda_{i6}$, $i = \{p, e\}$. Then, according to main equation theorem [2], the decision variables satisfy the following conditions:

$$\begin{aligned}(\gamma_p^*, \beta_p^*) &= \text{argmin}_{\gamma_p, \beta_p}\left[\sum_{j=1}^{6}(\lambda_{pj}f_j + \lambda_{ej}f_{j+6}) + 1\right],\\ (\gamma_e^*, \beta_e^*) &= \text{argmax}_{\gamma_e, \beta_e}\left[\sum_{j=1}^{6}(\lambda_{pj}f_j + \lambda_{ej}f_{j+6}) + 1\right].\end{aligned} \tag{4}$$

The above formulas are equivalent to:

$$\begin{aligned}\left(\cos\beta_p^*\cos\gamma_p^*, \cos\beta_p^*\sin\gamma_p^*, \sin\beta_p^*\right) &= -\frac{(\lambda_{p4}, \lambda_{p5}, \lambda_{p6})}{\sqrt{\lambda_{p4}^2 + \lambda_{p5}^2 + \lambda_{p6}^2}},\\ (\cos\beta_e^*\cos\gamma_e^*, \cos\beta_e^*\sin\gamma_e^*, \sin\beta_e^*) &= \frac{(\lambda_{e4}, \lambda_{e5}, \lambda_{e6})}{\sqrt{\lambda_{e4}^2 + \lambda_{e5}^2 + \lambda_{e6}^2}}.\end{aligned} \tag{5}$$

After substituting (5) with (2), we can get differential equations only related to the costate variables. On the other hand, according to path equation theorem [2], the dynamic equations of the costate variables under the Nash equilibrium are:

$$\begin{aligned}\dot{\lambda}_{i1} &= \lambda_{i4}\left(\frac{\mu\left(2r_{ix}^2 - r_{iy}^2 - r_{iz}^2\right)}{r_i^5}\right) + \lambda_{i5}\left(\frac{3\mu r_{ix}r_{iy}}{r_i^5}\right) + \lambda_{i6}\left(\frac{3\mu r_{ix}r_{iz}}{r_i^5}\right),\\ \dot{\lambda}_{i2} &= \lambda_{i4}\left(\frac{3\mu r_{ix}r_{iy}}{r_i^5}\right) + \lambda_{i5}\left(\frac{\mu\left(2r_{iy}^2 - r_{ix}^2 - r_{iz}^2\right)}{r_i^5}\right) + \lambda_{i6}\left(\frac{3\mu r_{iy}r_{iz}}{r_i^5}\right),\\ \dot{\lambda}_{i3} &= \lambda_{i4}\left(\frac{3\mu r_{ix}r_{iz}}{r_i^5}\right) + \lambda_{i5}\left(\frac{3\mu r_{iy}r_{iz}}{r_i^5}\right) + \lambda_{i6}\left(\frac{\mu\left(2r_{iz}^2 - r_{ix}^2 - r_{iy}^2\right)}{r_i^5}\right),\\ \dot{\lambda}_{i4} &= \lambda_{i1},\\ \dot{\lambda}_{i5} &= \lambda_{i2},\\ \dot{\lambda}_{i6} &= \lambda_{i3}.\end{aligned} \tag{6}$$

Now, a 24-dimensional differential equation $\dot{\vec{X}} = \vec{F}$ for the state and costate variables has been derived by coupling (2) and (6). The differential game problem is, thus, transformed into a TPBVP for this system, applying the necessary conditions [24]: the initial state variables (the position and velocity vectors of both players at the start of the game) are known, while the terminal conditions for the state and costate variables are determined by the game's terminal region.

Denote the terminal region of the game in this paper as Φ. Then, by definition,

$$\begin{aligned}\Phi = \{(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12})\},\\ s.t.\ (x_1 - x_7, x_2 - x_8, x_3 - x_9)\epsilon\phi.\end{aligned} \tag{7}$$

It is noted that with ϕ as the outer surface of a spherical cone, both the intersection circle between the sphere and the cone and the apex of the cone are non-smooth. Therefore, the terminal region Φ is also non-smooth. When applying differential game theory to the terminal region Φ, a low-dimensional hyperplane called an equal-value strategy surface (referred to as EVS hereafter, as shown in Figure 2), which passes through the non-smooth part, must be considered. When the initial state is not on the same side of the EVS as Φ, the state trajectory under the Nash equilibrium of the game will first reach the EVS and then Φ. The costate variables along these two paths have different definitions and need to be solved separately. Moreover, the EVS may also be bounded, leading to the generation of a new layer of EVS. In this model, the EVS is a 10-dimensional hyperplane in a 12-dimensional space, making the solution of its parametric equations and the determination of whether the initial state needs to pass through the EVS relatively complex. Therefore, this paper does not choose to solve problems directly using Φ as the terminal region.

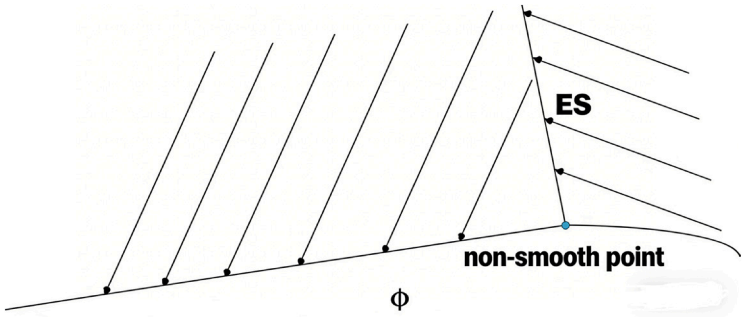


Figure 2. Sketch of EVS passing through non-smooth points.

In Sections 3 and 4, we present two algorithms for this problem. The algorithms in this paper can be extended to other problems involving terminal regions characterized by parametric equations with rotational symmetry axes or with measure-zero discontinuities.

3. Nash Equilibrium Solution

The method in this section is derived from the geometric properties of ϕ, noting that the outer surface of the spherical cone can be traversed by countable circles perpendicular to the axis of symmetry, originating from the apex of the cone (as shown in Figure 3). Therefore, although it is not a two-dimensional manifold, it can be traversed by countable one-dimensional manifolds. Let the circle with its center at a distance h from the apex of the cone be denoted as R_h, then $\phi = \cup_{h\in[0,r_0]} R_h$.

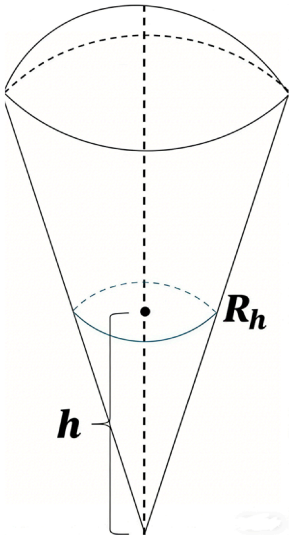


Figure 3. Circles traversing ϕ.

According to geometric relationships, the radius r_h of circle R_h is:

$$r_h = \begin{cases} htanj_0, \ if \ h \le r_0 cosj_0 \\ \sqrt{r_0^2 - h^2}, \ if \ r_0 cosj_0 < h \le r_0 \end{cases}. \tag{8}$$

Define

$$\begin{gathered} \Phi^h = \{(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12})\}, \\ s.t. \ (x_1 - x_7, x_1 - x_7, x_1 - x_7) \epsilon R_h. \end{gathered} \tag{9}$$

Thus, Φ^h is a 10-dimensional manifold in 12-dimensional space, $\Phi = \cup_{h\in[0,r_0]} \Phi^h$. We will construct a smooth mapping from R^{10} to Φ^h here. First, we establish a rectangular coordinate system with the ***ES*** direction as the z axis, where the unit vector along the z axis

is $\boldsymbol{E_z} = \frac{\boldsymbol{ES}}{\|\boldsymbol{ES}\|}$. We can assume the unit vector along the x axis is $\boldsymbol{E_x} = (0,0,1) \times \boldsymbol{E_z}$ and the unit vector along the y axis is $\boldsymbol{E_y} = \boldsymbol{E_x} \times \boldsymbol{E_z}$. The coordinate transformation matrix from this coordinate system to the geocentric coordinate system is given by

$$M = (\boldsymbol{E_x}, \boldsymbol{E_y}, \boldsymbol{E_z})^T. \tag{10}$$

Define a mapping from R^{10} to Φ^h:

$$\begin{gathered} g^h(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, \alpha_{10}) = (\alpha_4 + b_1, \alpha_5 + b_2, \alpha_6 + b_3, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9), \\ s.t.\ (b_1, b_2, b_3) = (r_h \cos\alpha_{10}, r_h \sin\alpha_{10}, h) \cdot M. \end{gathered} \tag{11}$$

It is easy to verify that g^h is a smooth mapping from R^{10} to Φ^h. Based on Φ^h and g^h, the following TPBVP relating to the differential equations $\dot{\vec{\boldsymbol{X}}} = \vec{\boldsymbol{F}}$ can be established: given the known initial state variables for the first 12 dimensions, after a time interval of Δt, there exists $\vec{\alpha} \in \mathrm{R}^{10}$ such that the state variables are $g^h\left(\vec{\alpha}\right)$, and the costate variables satisfy the following conditions:

$$\begin{gathered} \sum_{j=1}^{6} \left(\lambda_{pj} \frac{\partial g_j^h}{\partial \alpha_i} + \lambda_{ej} \frac{\partial g_{j+6}^h}{\partial \alpha_i} \right) = 0,\ for\ i = \{1, 2, \ldots, 10\}, \\ \sum_{j=1}^{6} \left(\lambda_{pj} f_j + \lambda_{ej} f_{j+6} \right) + 1 = 0. \end{gathered} \tag{12}$$

There has been extensive research on the solution methods for TPBVPs. This paper employs the homotopy method [25,26] to solve the examples. The detailed steps, robustness, and convergence of this algorithm have been thoroughly discussed by Hafer et al. [26] and, therefore, are not elaborated further here. After solving this TPBVP, we obtain the Nash equilibrium solution $\boldsymbol{s_h}$ for the pursuit-evasion differential game with the terminal region Φ^h under the given initial state variables. $\boldsymbol{s_h}$ can be expressed by the initial costate variables and the duration of the game as follows:

$$\boldsymbol{s_h} = \left(\lambda_{p1}^h, \lambda_{p2}^h, \lambda_{p3}^h, \lambda_{p4}^h, \lambda_{p5}^h, \lambda_{p6}^h, \lambda_{e1}^h, \lambda_{e2}^h, \lambda_{e3}^h, \lambda_{e4}^h, \lambda_{e5}^h, \lambda_{e6}^h, \Delta t_h \right). \tag{13}$$

Consider the Nash equilibrium solution $\boldsymbol{s}$ of the original game problem with the terminal region Φ. Denote the duration of the game at this equilibrium as Δt^*. Then, (1) for any $h \in [0, r_0]$, $\Delta t^* \leq \Delta t_h$ since $\Phi^h \subset \Phi$ and (2) there exists a $h_f \in [0, r_0]$ such that the state variables at the termination time under $\boldsymbol{s}$ are within Φ^{h_f}. $\boldsymbol{s}$ also satisfies the boundary conditions of the TPBVP defined with the terminal region Φ^h, $\boldsymbol{s} = \boldsymbol{s_{h_f}}$, and $\Delta t^* = \Delta t_{h_f}$ due to the uniqueness of the Nash equilibrium solution. In summary, $\boldsymbol{s}$ is the game solution with the shortest duration time among $\boldsymbol{s_h}$.

On the other hand, it is observed that on the closed intervals $[0, r_0 cosj_0]$ and $[r_0 cosj_0, r_0]$, the mappings from h to r_h, R_h, Φ^h are all smooth. Based on the smoothness of the solutions of bounded differential equations with respect to boundary conditions, $\boldsymbol{s_h}$ is smooth with respect to h on these two closed intervals. Consequently, Δt_h is uniformly continuous with respect to h on these intervals. Therefore, for any allowable time error ϵ, there exists a natural number n_E such that when $\left| h - h_f \right| < \frac{r_0}{n_E}$, we have $|\Delta t_h - \Delta t^*| < \epsilon$. Hence, choosing n_E as the traversal step size in the algorithm would satisfy the accuracy condition.

The following algorithm can be designed to solve for $\boldsymbol{s}$:

1. Choose a sufficiently large natural number n_E such that $\frac{r_0}{n_E}$ is less than the allowable error tolerance.
2. For $h = \frac{r_0}{n_E} i,\ i = \{0, 1, 2, \ldots, n\}$, solve $\boldsymbol{s_h}$ by the corresponding TPBVP.
3. Return the $\boldsymbol{s_h}$ with the shortest duration time.

This algorithm will be referred to as the ER algorithm in the following Section. The ER algorithm requires the solving of 24-dimensional TPBVPs for n_E times. The homotopy method used in this paper proceeds by first providing an initial solution when the homotopy parameter is zero, then using this solution as the starting point for a local search to find the solution after incrementing the homotopy parameter by the iteration step and repeating this process until the homotopy parameter reaches one. Let m_t denote the number of homotopy iterations, m_j the maximum number of iterations for the local search algorithm, and n the number of nodes in the numerical integration of the differential equations. Then, the computational complexity for solving one TPBVP is $O(m_t m_j n)$, and the overall computational complexity of the ER algorithm is $O(n_E m_t m_j n)$. In the simulations presented in this paper, the local search is implemented using an interior-point method, specifically the fmincon routine in MATLAB 2022a with a step-size tolerance of 1×10^{-6} and a constraint tolerance of 1×10^{-8}. With $m_t = 300$, $m_j = 3000$, $n_E = 100$ and $n = 100$, the total computation time on a system with 8 GB RAM and an i5-6500 3.2 GHz processor typically ranges between 50 and 100 s.

Although the ER algorithm can ensure that the game terminates in the region Φ, its computation time is not negligible compared to the game time.

4. Approximation Algorithm

The ER algorithm may not address cases where decision-making needs to be immediate. Therefore, in this section, we construct a fast algorithm for approximating solutions. The idea of this algorithm is to "polish" the outer surface of the spherical cone ϕ. Specifically, we first construct a sequence of 2-dimensional manifolds that are sufficiently close to ϕ, thereby obtaining an 11-dimensional manifold that approximates Φ. By solving the TPBVP with this manifold as the terminal region, we can obtain an approximate solution.

First, take the center of the intersection circle between the sphere and the cone as the origin, and use the polar coordinates of the x, y, z axis, which are along direction $\boldsymbol{E_x}, \boldsymbol{E_y}, \boldsymbol{E_z}$, to make a two-dimensional parametric representation of ϕ.

As shown in Figure 4, denoting the latitude angle is θ_1 and the longitude angle is θ_2, then

$$\phi = \{(r(\theta_2)\cos\theta_1\cos\theta_2, r(\theta_2)\sin\theta_1\cos\theta_2, r(\theta_2)\sin\theta_2)\}.s.t.\ (\theta_1, \theta_2)\epsilon[-\pi, \pi], \tag{14}$$

According to geometric relationships,

$$r(\theta_2) = \begin{cases} r_1(\theta_2) = -\sin\theta_2\cos j_0 r_0 + \sqrt{\sin^2\theta_2\cos^2 j_0 r_0^2 + \sin^2 j_0 r_0^2},\ if\ \sin\theta_2 \geq 0. \\ r_2(\theta_2) = \frac{r_0 cos j_0}{cos\theta_2 - sin\theta_2 cot j_0},\ if\ \sin\theta_2 < 0,\ \cos\theta_2 \geq 0. \\ r_3(\theta_2) = \frac{r_0 cos j_0}{-cos\theta_2 - sin\theta_2 cot j_0},\ if\ sin\theta_2 < 0,\ cos\theta_2 < 0. \end{cases} \tag{15}$$

The three classifications in (15) correspond to the two non-smooth regions of ϕ ($\sin\theta_2 = 0$ refers to the intersection circle between the sphere and the cone and $\sin\theta_2 < 0, \cos\theta_2 = 0$ refers to the apex of the cone). The judgments regarding the signs of $\sin\theta_2$ and $\cos\theta_2$ can be represented using switch functions $\mathrm{sgn}(\sin\theta_2)$ and $\mathrm{sgn}(cos\theta_2)$ ($\mathrm{sgn}(x) = 1$, if $x \geq 0$; $\mathrm{sgn}(x) = 0$, if $x < 0$). The switch function is discontinuous, but it can be approximated by a smooth function known as a Sigmoid function, which refers to a class of functions of the form $s_b(x) = \frac{1}{1+b^{-x}}$, converging to sgn(x) in measure with the increase in b. By replacing the switch function with the Sigmoid function, we can obtain the following approximate function of $r(\theta_2)$:

$$r_b^1(\theta_2) = (1 - s_b(\sin\theta_2))(1 - s_b(\cos\theta_2))r_3(\theta_2) + (1 - s_b(\sin\theta_2))s_b(\cos\theta_2)r_2(\theta_2) + s_b(\sin\theta_2)r_1(\theta_2). \tag{16}$$

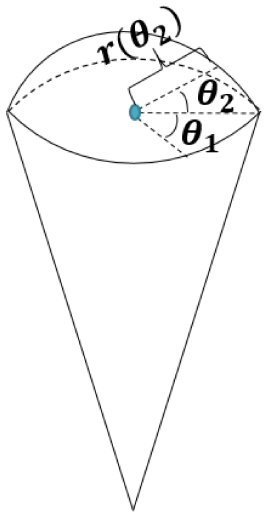


Figure 4. The two-dimensional parametric representation of ϕ.

It is noted that $r_2(\theta_2)$, $r_3(\theta_2)$ is smooth within its domain but has singularities outside of it. We further use

$$\begin{aligned} r_{2b}(\theta_2) &= \frac{r_0 \cos j_0}{\cos\theta_2 - \sin\theta_2 \cot j_0 + b^{-(\theta_2 + j_0)}} \\ r_{3b}(\theta_2) &= \frac{r_0 \cos j_0}{-\cos\theta_2 - \sin\theta_2 \cot j_0 + b^{-(-\theta_2 - (\pi - j_0))}} \end{aligned} \tag{17}$$

to replace $r_2(\theta_2)$, $r_3(\theta_2)$, respectively. Note that within the corresponding domains, $r_{2b}(\theta_2)$ and $r_{3b}(\theta_2)$ converge to $r_2(\theta_2)$ and $r_3(\theta_2)$ everywhere; outside the domains, they are uniformly bounded and can be controlled by the switch function approximated by the Sigmoid function. Therefore, the resulting approximate function

$$r_b^2(\theta_2) = (1 - s_b(\sin\theta_2))(1 - s_b(\cos\theta_2))r_{3b}(\theta_2) + (1 - s_b(\sin\theta_2))s_b(\cos\theta_2)r_{2b}(\theta_2) + s_b(\sin\theta_2)r_1(\theta_2). \tag{18}$$

is a smooth function that converges to $r(\theta_2)$ in measure. Theorem 1 provides an estimate of the convergence rate of $r_b^2(\theta_2)$ to $r(\theta_2)$, allowing parameter b to be selected based on the desired convergence accuracy.

Theorem 1.

(1) *When* $\sin\theta_2 > 0$, $\left|r_b^2(\theta_2) - r(\theta_2)\right| < \frac{3r_0}{b^{\sin\theta_2}}$;

(2) *When* $\sin\theta_2 < 0$, $\cos\theta_2 > 0$, $\left|r_b^2(\theta_2) - r(\theta_2)\right| < \frac{r_0}{b^{\cos\theta_2}} + \frac{r_0}{b^{-\sin\theta_2}} + \frac{r_0}{\cos j_0 b^{j_0}}$;

(3) *When* $\sin\theta_2 < 0$, $\cos\theta_2 < 0$, $\left|r_b^2(\theta_2) - r(\theta_2)\right| < \frac{r_0}{b^{-\cos\theta_2}} + \frac{r_0}{b^{-\sin\theta_2}} + \frac{r_0}{\cos j_0 b^{j_0}}$.

Proof.

(1) When $\sin\theta_2 > 0$,

$$\begin{aligned} &\left|r_b^2(\theta_2) - r(\theta_2)\right| = \left|r_b^2(\theta_2) - r_1(\theta_2)\right| \\ =&\left|(1 - s_b(\sin\theta_2))[(1 - s_b(\cos\theta_2))r_{3b}(\theta_2) + s_b(\cos\theta_2)r_{2b}(\theta_2) - r_1(\theta_2)]\right| \\ =&\left|1 - \frac{1}{1 + b^{-\sin\theta_2}}\right|\left|[(1 - s_b(\cos\theta_2))r_{3b}(\theta_2) + s_b(\cos\theta_2)r_{2b}(\theta_2) - r_1(\theta_2)]\right| \\ <&\left|1 - \left(1 - b^{-\sin\theta_2}\right)\right|(|r_{3b}(\theta_2)| + |r_{2b}(\theta_2)| + |r_1(\theta_2)|) < \frac{3r_0}{b^{\sin\theta_2}}. \end{aligned} \tag{19}$$

(2) When $\sin\theta_2 < 0,\ \cos\theta_2 > 0$,

$$\begin{aligned} &\left|r_b^2(\theta_2) - r(\theta_2)\right| = \left|r_b^2(\theta_2) - r_2(\theta_2)\right| \\ &\le \left|(1 - s_b(\cos\theta_2))(1 - s_b(\sin\theta_2))r_{3b}(\theta_2)\right| + \left|s_b(\sin\theta_2)r_1(\theta_2)\right| \\ &\quad + \left|(1 - s_b(\sin\theta_2))s_b(\cos\theta_2)(r_{2b}(\theta_2) - r_2(\theta_2)\right| \\ &< \frac{r_0}{b^{\cos\theta_2}} + \frac{r_0}{b^{-\sin\theta_2}} + \left|\frac{1}{\frac{1}{r_2(\theta_2)} + \frac{b^{-(\theta_2+j_0)}}{r_0\cos j_0}} - r_2(\theta_2)\right| \\ &< \frac{r_0}{b^{\cos\theta_2}} + \frac{r_0}{b^{-\sin\theta_2}} + \left|\frac{b^{-(\theta_2+j_0)}}{r_0\cos j_0} {r_2}^2(\theta_2)\right| \\ &< \frac{r_0}{b^{\cos\theta_2}} + \frac{r_0}{b^{-\sin\theta_2}} + \frac{r_0}{\cos j_0 b^{j_0}}. \end{aligned} \tag{20}$$

(3) When $\sin\theta_2 < 0,\ \cos\theta_2 < 0$,

$$\begin{aligned} &\left|r_b^2(\theta_2) - r(\theta_2)\right| = \left|r_b^2(\theta_2) - r_3(\theta_2)\right| \\ &\le \left|(1 - s_b(\sin\theta_2))s_b(\cos\theta_2)r_{2b}(\theta_2)\right| + \left|s_b(\sin\theta_2)r_1(\theta_2)\right| \\ &\quad + \left|(1 - s_b(\sin\theta_2))(1 - s_b(\cos\theta_2))(r_{3b}(\theta_2) - r_3(\theta_2)\right| \\ &< \frac{r_0}{b^{-\cos\theta_2}} + \frac{r_0}{b^{-\sin\theta_2}} + \left|\frac{1}{\frac{1}{r_3(\theta_2)} + \frac{b^{\theta_2+(\pi-j_0)}}{r_0\cos j_0}} - r_3(\theta_2)\right| \\ &< \frac{r_0}{b^{-\cos\theta_2}} + \frac{r_0}{b^{-\sin\theta_2}} + \left|\frac{b^{\theta_2+(\pi-j_0)}}{r_0\cos j_0} {r_3}^2(\theta_2)\right| \\ &< \frac{r_0}{b^{-\cos\theta_2}} + \frac{r_0}{b^{-\sin\theta_2}} + \frac{r_0}{\cos j_0 b^{j_0}} \end{aligned} \tag{21}$$

□

Define

$$\begin{aligned} \phi_b = \left\{\left(r_b^2(\theta_2)\cos\theta_1\cos\theta_2, r_b^2(\theta_2)\sin\theta_1\cos\theta_2, r_b^2(\theta_2)\sin\theta_2\right)\right\}, \\ s.t.\ (\theta_1, \theta_2)\epsilon \mathrm{R}^2. \end{aligned} \tag{22}$$

Then, ϕ_b (as shown in Figure 5) is a 2-dimensional manifold approximate to ϕ.

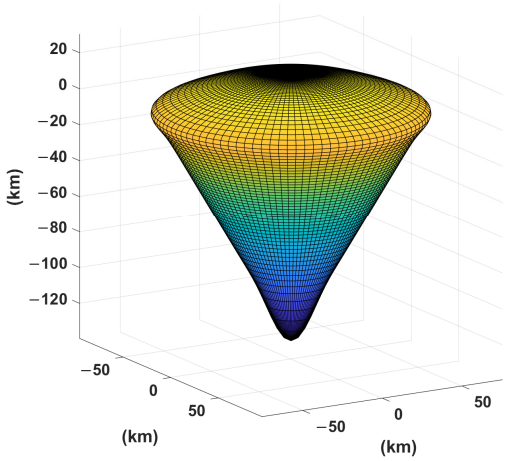
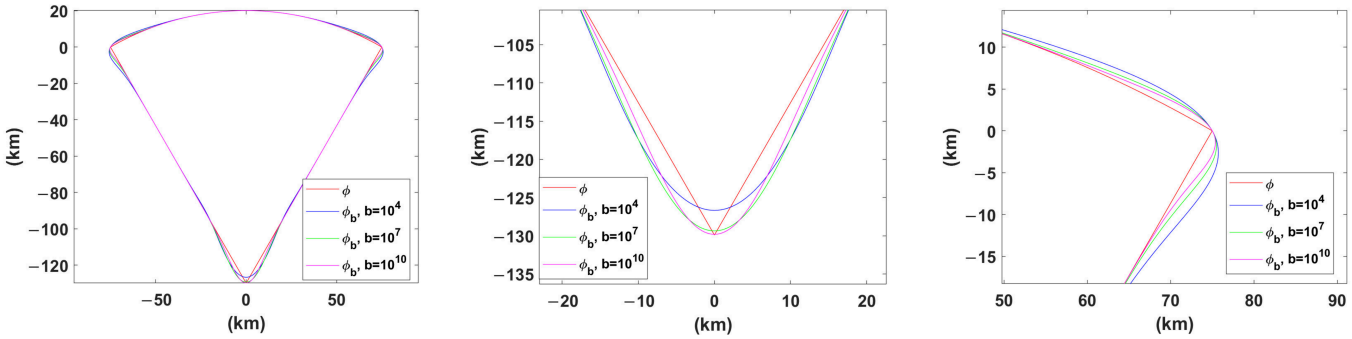


Figure 5. 3D view of ϕ_b ($r_0 = 150$ km, $j_0 = 30°$, $b = 10^{10}$).

Figure 6 further shows the approximate degree between ϕ and ϕ_b through their cross section when $r_0 = 150$ km and $j_0 = 30°$. It can be found that the approximate degree increases in b. In the vast majority of situations for ϕ, the approximate degree is very high. Only in the regions near to the non-smooth point of ϕ, do some difference exists.



(**a**) Cross section of ϕ and ϕ_b. (**b**) Apex of cross section. (**c**) Intersection circle of cross section.

Figure 6. Examples of cross section.

Define

$$\begin{gathered}\Phi_b = \{(x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12})\}, \\ s.t.\ (x_1 - x_7, x_2 - x_8, x_3 - x_9)\epsilon\phi_b.\end{gathered} \tag{23}$$

Then, Φ_b is an 11-dimensional manifold approximate to Φ. Define a mapping from R^{11} to R^{12}:

$$\begin{gathered}k^b(\alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9, \alpha_{10}, a_{11}) = (\alpha_4 + b_1, \alpha_5 + b_2, \alpha_6 + b_3, \alpha_1, \alpha_2, \alpha_3, \alpha_4, \alpha_5, \alpha_6, \alpha_7, \alpha_8, \alpha_9), \\ s.t.\ (b_1, b_2, b_3) = (r(a_{11})\cos\alpha_{10}\cos a_{11}, r(a_{11})\sin\alpha_{10}\cos a_{11}, r_0\cos j_0 + r(a_{11})\sin a_{11})\cdot M.\end{gathered} \tag{24}$$

It is easy to verify that k^b is a smooth mapping from R^{11} to Φ_b. Similarly, based on Φ_b and k^b, the following TPBVP relating to the differential equations $\dot{\vec{X}} = \vec{F}$ can be established: given the known initial state variables for the first 12 dimensions, after a time interval of Δt, there exists $\vec{\alpha}\epsilon R^{11}$ such that at that time the state variables are $k^b\left(\vec{\alpha}\right)$, and the costate variables satisfy the following conditions:

$$\begin{gathered}\sum_{j=1}^{6}\left(\lambda_{pj}\frac{\partial k_j^b}{\partial\alpha_i} + \lambda_{ej}\frac{\partial k_{j+6}^b}{\partial\alpha_i}\right) = 0,\ for\ i = \{1, 2, \ldots, 11\}. \\ \sum_{j=1}^{6}\left(\lambda_{pj}f_j + \lambda_{ej}f_{j+6}\right) + 1 = 0.\end{gathered} \tag{25}$$

Unlike the algorithm in Section 3, here we only need to determine a suitable b, and can then solve the corresponding TPBVP with Φ_b as the terminal region to obtain the approximate solution. According to Theorem 1, the larger b is, the closer to $\boldsymbol{s}$ the solution will be. However, the computation time will also increase in b, and the marginal effect of increasing b is decreasing. Typically, b can be determined by the distance threshold of the imaging region, the specific computational conditions and the required accuracy.

This algorithm will be referred to as the AP algorithm in the following Section. Since it only requires our solving a 24-dimension TPBVP once, the overall computational complexity is $O(\log_2 bm_t n)$. When $b = 10^{10}$ $m_t = 300$ and $n = 100$, on a system with 8 GB RAM and an i5-6500 3.2 GHz processor, the total computation time typically ranges between 1 and 2 s.

The AP algorithm significantly improves computational speed compared to the ER algorithm. However, its solutions may fall outside the actual imaging region of E, rendering it less suitable for imaging devices with limited tolerance for deviation.

5. Simulation Examples

Since the duration of most orbital games does not exceed a few hours, for the sake of program simplicity, we ignore the changes in the sun's coordinates during the game and assume the initial coordinates of the sun as $\boldsymbol{s}_0 = (s_1, s_2, s_3)$. The following cases are

based on the assumption from Table 1 and primarily examine the performance of the two algorithms across varying initial sun angles, in both high and low orbits, and within identical and disparate plane configurations.

Table 1. Basic simulation assumption.

Maneuvering accelerations of P(T_p)	0.002 km/s^2
Maneuvering accelerations of E(T_e)	0.001 km/s^2
Distance threshold of imaging region(r_0)	150 km
Sun-angle threshold of imaging region(s_0)	15°
Sigmoid function parameter of AP algorithm(b)	10^{10}

The maneuvering accelerations of both players T_p, T_e follow the assumptions used in previous studies [8,12] to maintain consistency with the established literature. The distance threshold r_0 is set slightly higher than the collision warning distance for high-orbit satellites, reflecting the enhanced imaging capability of modern reconnaissance systems. The sun-angle threshold s_0 is chosen based on typical observation angles used in Earth-observation satellites. The parameter b is determined by balancing computational performance and approximation accuracy on our computing platform. According to Theorem 1, the approximate termination region of the AP algorithm has an error not exceeding 1 km when the pitch angle differs from 0 or $-90°$ by more than 15°.

Details of their specific parameter settings and the resultant outcomes for each algorithm are presented below.

5.1. Example Parameters

Case 1 (medium-low orbit; large initial sun angle; and same plane)

The initial orbital elements are as follows:

$$\begin{aligned}(a_{p0}, e_{p0}, i_{p0}, \Omega_{p0}, \omega_{p0}, E_{p0}) &= (10{,}000 \text{ km}, 0.05, 0, 0, 0, 0),\\ (a_{e0}, e_{e0}, i_{e0}, \Omega_{e0}, \omega_{e0}, E_{e0}) &= (10{,}000 \text{ km}, 0, 0, 0, 0, 1.17°).\end{aligned} \tag{26}$$

Assuming $\boldsymbol{s}_0 = (149{,}600{,}000 \text{ km}, 0, 0)$, then the initial distance between both parties is 1513 km, and the initial sun angle is 172 degrees.

Case 2 (medium-low orbit; large initial sun angle; and same plane)

The initial orbital elements are as follows:

$$\begin{aligned}(a_{p0}, e_{p0}, i_{p0}, \Omega_{p0}, \omega_{p0}, E_{p0}) &= (11{,}000 \text{ km}, 0, 0, 0, 0, 1.17°),\\ (a_{e0}, e_{e0}, i_{e0}, \Omega_{e0}, \omega_{e0}, E_{e0}) &= (10{,}000 \text{ km}, 0.05, 0, 0, 0, 0).\end{aligned} \tag{27}$$

Assuming $\boldsymbol{s}_0 = (149{,}600{,}000 \text{ km}, 0, 0)$, then the initial distance between both parties is 1513 km, and the initial sun angle is 8.36 degrees.

Case 3 (high orbit; large initial sun angle; and same plane)

The initial orbital elements are as follows:

$$\begin{aligned}(a_{p0}, e_{p0}, i_{p0}, \Omega_{p0}, \omega_{p0}, E_{p0}) &= (40{,}000 \text{ km}, 0.05, 0, 0, 0, 0),\\ (a_{e0}, e_{e0}, i_{e0}, \Omega_{e0}, \omega_{e0}, E_{e0}) &= (41{,}000 \text{ km}, 0, 0, 0, 0, 11.7°).\end{aligned} \tag{28}$$

Assuming $\boldsymbol{s}_0 = (149{,}600{,}000 \text{ km}, 0, 0)$, then the initial distance between both parties is 8433 km, and the initial sun angle is 105 degrees.

Case 4 (medium-low orbit; middle initial sun angle; and different plane)

The initial orbital elements are as follows:

$$\begin{aligned}(a_{p0}, e_{p0}, i_{p0}, \Omega_{p0}, \omega_{p0}, E_{p0}) &= (10{,}800 \text{ km}, 0, 0, 0, 0, 0),\\ (a_{e0}, e_{e0}, i_{e0}, \Omega_{e0}, \omega_{e0}, E_{e0}) &= (10{,}000 \text{ km}, 0.05, 17.2^\circ, 0, 0, 11.7^\circ).\end{aligned} \tag{29}$$

Assuming $\boldsymbol{s}_0 = (0, 149{,}600{,}000 \text{ km}, 0)$, then the initial distance between both parties is 1300 km, and the initial sun angle is 93 degrees.

Note that in Case 1 and Case 2, the initial positions of both spacecraft are set to be nearly collinear with the sun direction to examine the scenarios of large and small initial sun angles, respectively. In Case 3 and Case 4, we further investigate more complex configurations by introducing a large orbital phase difference and a large inclination difference between the two spacecraft.

5.2. Results of the Two Algorithms

The computational speed advantage of the AP algorithm is evident. In this subsection, we present simulation examples for both algorithms to verify their effectiveness and further explore the accuracy trade-offs of the AP algorithm by comparing the results obtained from both algorithms.

The duration time, the distance and the sun angle at the termination moment in four cases are shown in Table 2.

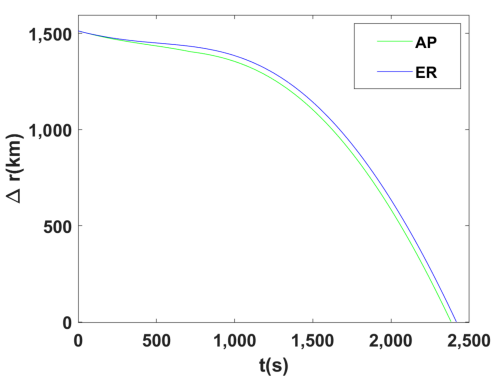
Table 2. Results of the two algorithms.

	Case 1	**Case 2**	**Case 3**	**Case 4**
Duration time Δt(ER)	2414 s	2260 s	3730 s	2990 s
Duration time Δt(AP)	2383 s	2260 s	3730 s	2970 s
Terminal distance Δr^f(ER)	0.25 km	150 km	11 km	131 km
Terminal distance Δr^f(AP)	1.14 km	150.4 km	14 km	129 km
Terminal sun angle s^f(ER)	15°	14.7°	15°	15°
Terminal sun angle s^f(AP)	77°	13.7°	33°	15°

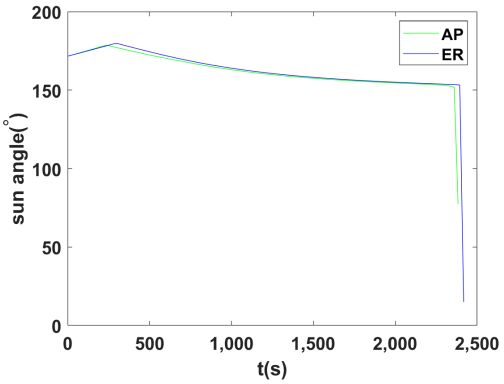
The distance between both parties, sun angle, and trajectories over time in Case 1–4 are shown in Figures 7–10, respectively.

First, it can be observed that the ER(AP) algorithm is effective, as the game terminates on $\Phi(\Phi_b)$ in all four cases. The results of the two algorithms are very close in terms of the distance and the duration time. Through these cases, we summarize the following characteristics from the Nash equilibrium of the game:

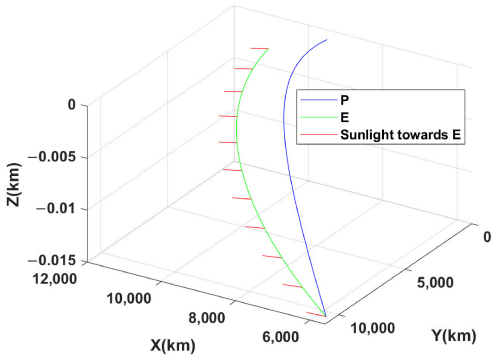
- The duration time increases in terms of both the initial distance and the sun angle;
- The sun angle decreases dramatically in the terminal region. When the initial sun angle is not too small, P will always prioritize shortening the distance. Only after reducing the distance to a certain extent will it decrease the sun angle. This is because, at that point, the distance is sufficiently short, allowing for a more rapid rate of change in the sun angle. Consequently, as shown in Figures 7 and 9, although P has an imaging distance of 150 km, it will approach near the apex of the spherical cone, which is very close to E in the terminal region;
- The larger the initial sun angle, the closer P's position on the outer surface of the spherical cone is to the apex at the termination moment; conversely, if the initial sun angle is small, P's position approaches the sphere.



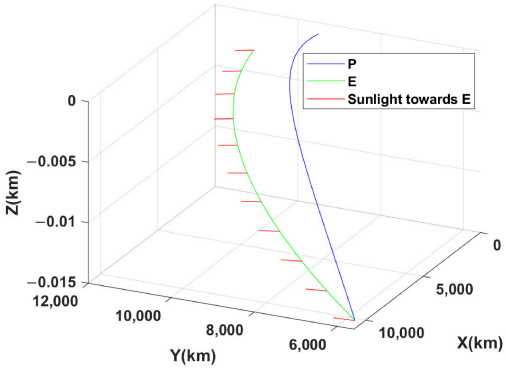
(**a**) The distance in Case 1.



(**b**) The sun angle in Case 1.

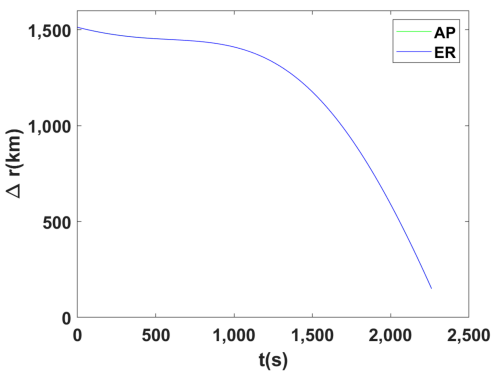


(**c**) The trajectories of the ER algorithm in Case 1.

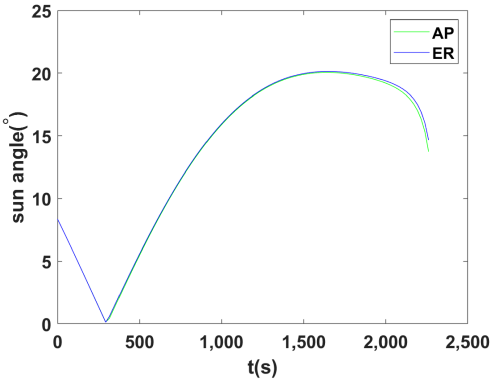


(**d**) The trajectories of the AP algorithm in Case 1.

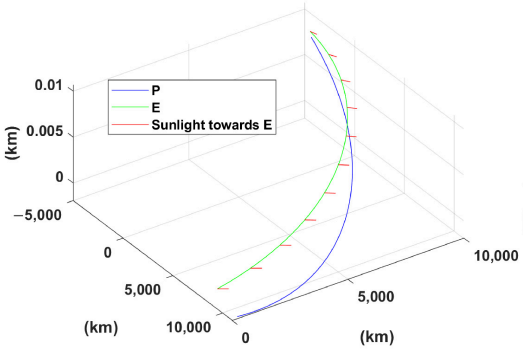
Figure 7. Results of the two algorithms in Case 1.



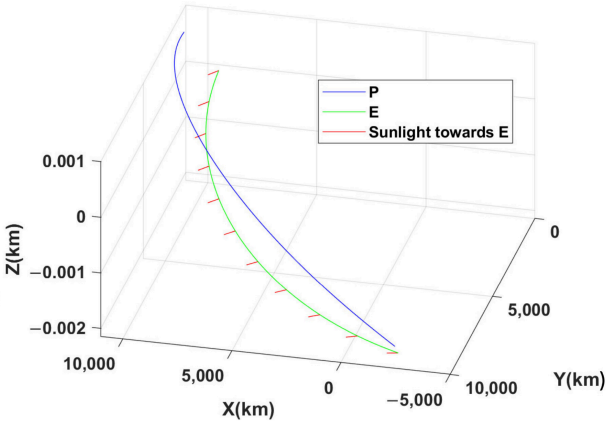
(**a**) The distance in Case 2.



(**b**) The sun angle in Case 2.

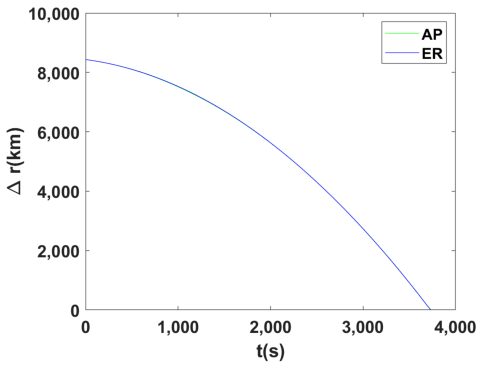


(**c**) The trajectories of the ER algorithm in Case 2.

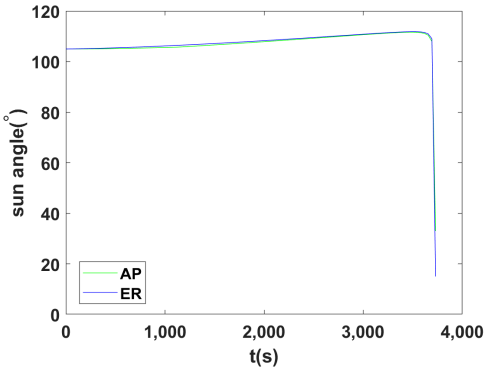


(**d**) The trajectories of the AP algorithm in Case 2.

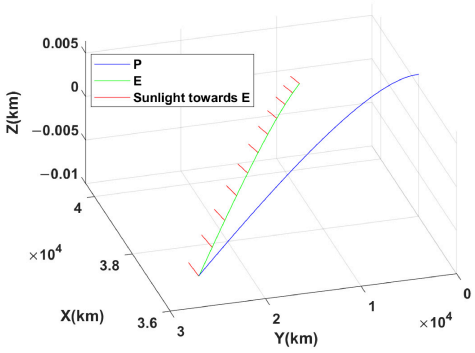
Figure 8. Results of the two algorithms in Case 2.



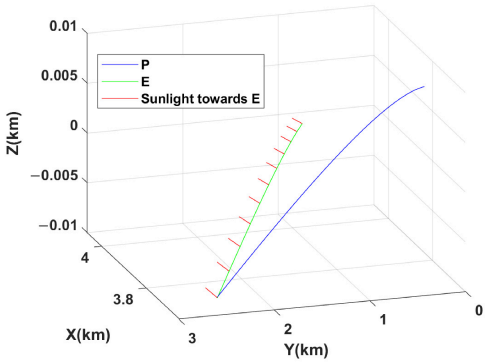
(**a**) The distance in Case 3.



(**b**) The sun angle in Case 3.

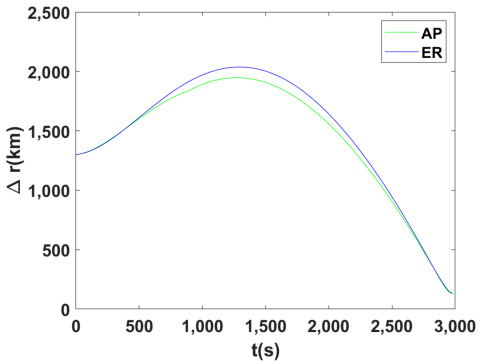


(**c**) The trajectories of the ER algorithm in Case 3.

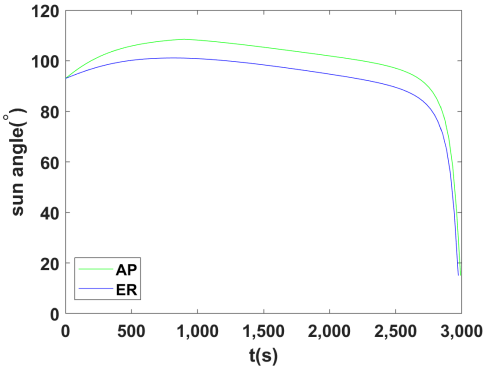


(**d**) The trajectories of the AP algorithm in Case 3.

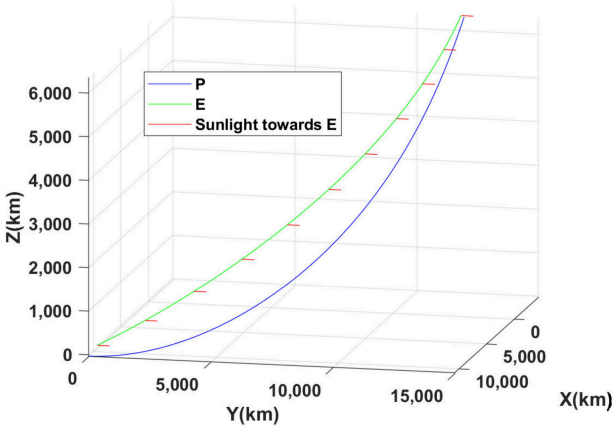
Figure 9. Results of the two algorithms in Case 3.



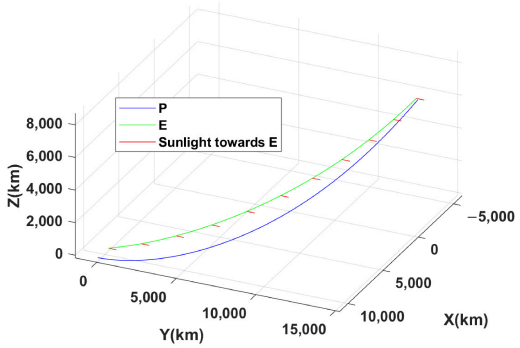
(**a**) The distance in Case 4.



(**b**) The sun angle in Case 4.



(**c**) The trajectories of the ER algorithm in Case 4.



(**d**) The trajectories of the AP algorithm in Case 4.

Figure 10. Results of the two algorithms in Case 4.

Obviously, due to different model settings, our simulation trajectory significantly differs from that in similar studies. In the results of Prince et al. [18], the relative distance at the termination time is often greater than that at the initial time, and the thrust direction of the two spacecraft remains almost unchanged in the relative coordinate system. This contrasts with our result, where P first reduces the distance and then adjusts the sun angle. This difference arises because their model only considers the sun angle as an imaging constraint. In the results of Li et al. [4], P always stays near the imaging region of E to maximize imaging quality, which is different from the process we show, where P approaches the imaging region of E from a farther distance.

However, due to the difference between Φ_b and Φ, the following difference in the results of the two algorithms exists:

- When the initial sun angle is large, the termination sun angle of the AP algorithm is larger, placing P outside the defined imaging range. This is due to the AP algorithm's terminal region smoothing the apex of the cone into a small surface, which encompasses a larger sun angle. This result can be optimized by further increasing b at the cost of computation time;
- As shown in Figures 5 and 6, ϕ_b has a larger enclosed area than ϕ, especially in the region near the two non-smooth parts. Thus, the duration time of the AP algorithm is longer, especially when the termination distance is small.

While the AP algorithm offers computational efficiency, it occasionally produces equilibrium solutions where the sun angle exceeds the imaging threshold under large initial sun angles. However, this error manifests only at very short imaging distances, where modern optical imaging systems exhibit relaxed sun-angle constraints due to reduced geometric sensitivity. This physical characteristic ensures that the AP algorithm's error remains operationally irrelevant for most reconnaissance scenarios.

In summary, the ER algorithm is applicable for high-precision requirements regardless of initial sun angle. The AP algorithm is applicable for all scenarios except those demanding high precision at large initial sun angles. Computational constraints determine algorithm selection based on these operational conditions.

6. Conclusions

This paper investigates the zero-sum orbital pursuit-evasion game under sun-angle constraints within the context of spacecraft optical imaging reconnaissance. The primary contributions are: (1) the first rigorous formulation and analysis of a differential game where P minimizes the time-to-entry into E's effective imaging region; (2) a novel algorithm for resolving the Nash equilibrium despite the terminal region's bounded and non-smooth geometry; (3) an efficient approximation algorithm derived through smooth manifold approximation of the terminal region; and (4) a comprehensive characterization of equilibrium properties and comparative algorithmic performance.

Based on this work, there remain several directions worth investigating regarding the orbital pursuit-evasion game problem with sun-angle constraints. Firstly, solving the differential game problem with the original imaging region as the terminal region directly through the equivalent strategy surface can achieve both efficiency and accuracy algorithmically. However, as described in Section 3, the theoretical difficulty of this method in high-dimensional space is considerable, and we hope to address this in future research. Secondly, this paper assumes that the game involves complete information, but in practical scenarios, there may be uncertainties regarding the performance of imaging devices. Thirdly, the background of this problem could also be extended to scenarios involving many-to-one or many-to-many interactions. Lastly, the impact of orbital debris, which can render certain areas unpassable, is also worth studying.

https://doi.org/10.3390/aerospace13040356

Author Contributions: Conceptualization, X.L.; formal analysis, G.Z.; investigation, X.L.; methodology, X.L.; resources, Y.L.; software, X.Y.; supervision, X.Z.; validation, X.Z.; and writing—original draft, X.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author upon reasonable request.

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

Abbreviations

The following abbreviations are used in this manuscript:

TPBVP	Two-point boundary-value problem
EVS	Equal-value strategy surface
ER	Algorithm in Section 3
AP	Algorithm in Section 4

References

1. Anderson, G.M.; Grazier, V.W. Barrier in Pursuit-Evasion Problems Between Two Low-Thrust Orbital Spacecraft. *AIAA J.* **1976**, *14*, 158–163. [CrossRef]
2. Vajda, S.; Isaacs, R. *Differential Games*; Wiley: New York, NY, USA, 1965.
3. Pontani, M.; Conway, B.A. Numerical Solution of the Three-Dimensional Orbital Pursuit-Evasion Game. *J. Guid. Control Dyn.* **2009**, *32*, 474–487. [CrossRef]
4. Li, Z.Y.; Zhu, H.; Luo, Y.Z. Orbital Inspection Game Formulation and Epsilon-Nash Equilibrium Solution. *J. Spacecr. Rockets* **2024**, *61*, 157–172. [CrossRef]
5. Nakka, Y.K.; Hönig, W.; Choi, C.; Harvard, A.; Rahmani, A.; Chung, S.J. Information-Based Guidance and Control Architecture for Multi-Spacecraft On-Orbit Inspection. *J. Guid. Control Dyn.* **2022**, *45*, 1184–1201. [CrossRef]
6. Sharma, J. Space-Based Visible Space Surveillance Performance. *J. Guid. Control Dyn.* **2000**, *23*, 153–158. [CrossRef]
7. Shen, H.X.; Casalino, L. Revisit of the Three-Dimensional Orbital Pursuit-Evasion Game. *J. Guid. Control Dyn.* **2018**, *41*, 1823–1831. [CrossRef]
8. Stupik, J.; Pontani, M.; Conway, B. Optimal Pursuit/Evasion Spacecraft Trajectories in the Hill Reference Frame. In *Proceedings of the AIAA/AAS Astrodynamics Specialist Conference*; AIAA: Reston, VA, USA, 2012.
9. Zhang, C.G.; Zhu, Y.W.; Yang, L.P. An Optimal Guidance Method for Free-Time Orbital Pursuit-Evasion Game. *J. Syst. Eng. Electron.* **2022**, *33*, 1294–1308.
10. Jansson, O.; Harris, M.W. A Geometrical, Reachable Set Approach for Constrained Pursuit–Evasion Games with Multiple Pursuers and Evaders. *Aerospace* **2023**, *10*, 477. [CrossRef]
11. Li, X.; Zhou, X.; Yu, X.; Zhao, G.; Liu, Y. Interception Domain Approach to Orbital Multi-Player "Encirclement-Capture" Games: Theoretical Foundations and Solutions. *Aerospace* **2025**, *12*, 875. [CrossRef]
12. Zhou, J.F.; Zhao, L.; Li, H.; Cheng, J.H.; Wang, S. Compensation Control Strategy for Orbital Pursuit-Evasion Problem with Imperfect Information. *Appl. Sci.* **2020**, *11*, 1400. [CrossRef]
13. Tang, X.; Ye, D.; Huang, L.; Sun, Z.W.; Sun, J.Y. Pursuit-evasion game switching strategies for spacecraft with incomplete-information. *Aerosp. Sci. Technol.* **2021**, *117*, 107–112. [CrossRef]
14. Liu, P.; Yang, B.; Li, S.; Xin, M. Parameter-Optimized Pursuit Strategy for Orbital Games with Incomplete Information. *J. Guid. Control Dyn.* **2025**, *48*, 1730–1742. [CrossRef]
15. Li, Z.Y.; Zhu, H.; Yang, Z. Saddle Point of Orbital Pursuit-Evasion Game Under J2-Perturbed Dynamics. *J. Guid. Control Dyn.* **2020**, *43*, 1733–1739. [CrossRef]
16. Zeng, X.; Yang, L.; Zhu, Y.; Yang, F. Comparison of Two Optimal Guidance Methods for the Long-Distance Orbital Pursuit-Evasion Game. *IEEE Trans. Aerosp. Electron. Syst.* **2021**, *57*, 521–539. [CrossRef]
17. Li, Z.Y.; Luo, Y.Z. Orbital Impulsive Pursuit-Evasion Game Formulation and Stackelberg Equilibrium Solutions. *J. Spacecr. Rockets* **2025**, *62*, 349–364. [CrossRef]
18. Prince, E.R.; Hess, J.A.; Cobb, R.G. Elliptical Orbit Proximity Operations Differential Games. *J. Guid. Control Dyn.* **2019**, *42*, 1458–1472. [CrossRef]
19. 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*; AIAA: Reston, VA, USA, 2014.

20. 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. [CrossRef]
21. Zhang, S.; Yang, Z.; Luo, Y.Z. Time-Dependent Reachable Domain and Its Application to Impulsive Orbital Pursuit–Evasion Analysis. *J. Spacecr. Rockets* **2024**, *61*, 631–642. [CrossRef]
22. Venigalla, C.; Scheeres, D.J. Delta-V-Based Analysis of Spacecraft Pursuit-Evasion Games. *J. Guid. Control Dyn.* **2021**, *44*, 1961–1971. [CrossRef]
23. Wang, C.; Chen, D.; Liao, W. Research on Maneuver Strategy in Satellite Observation and Counter-Observation Game. *Adv. Space Res.* **2024**, *74*, 3170–3185. [CrossRef]
24. Breitner, M.H.; Pesch, H.J.; Grimm, W. Complex Differential Games of Pursuit-Evasion type with State Constraints, Part 1: Necessary Conditions for Optimal Open-Loop Strategies. *J. Optim. Theory Appl.* **1993**, *78*, 419–441. [CrossRef]
25. Pang, B.; Wen, C.X.; Han, H.W.; Qiao, D. Solving Pursuit/Evasion Game Along Elliptical Orbit by Providing Precise Gradient. *J. Guid. Control Dyn.* **2024**, *47*, 797–807. [CrossRef]
26. 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. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.