Investigation of Target Occlusion Duration Based on Euler Iteration and Monte Carlo Methods

Qiyao Chen *

School of Science, Zhejiang Sci-Tech University, Hangzhou, China

* Corresponding Author Email: 2625285442@qq.com

Abstract. This paper investigates the calculation and optimization of target masking duration based on the Euler iteration method and Monte Carlo algorithm. Through numerical simulation and algorithmic optimization, it aims to explore the variation patterns of masking duration under different parameters and determine the optimal parameter configuration. First, a coordinate update model based on Euler iteration is constructed. Initial coordinates and motion parameters of moving bodies are set, and their three-dimensional coordinates are iteratively updated with infinitesimal time steps. The coordinate calculation logic for different motion phases is clarified, providing foundational data support for occlusion duration assessment. Second, an occlusion effectiveness assessment model is established. Key distances are computed using spatial geometry methods, and occlusion criteria are defined through sign functions. The cumulative time meeting occlusion conditions enables quantitative calculation of occlusion duration under specified parameters, while analyzing the impact of resistance factors on results. Finally, a single-objective optimization model is designed to maximize the duration of concealment. Based on the Monte Carlo algorithm, random sampling is performed within parameter constraints. Through multiple numerical simulations, the optimal parameter combination is selected to achieve the maximum concealment duration, validating the algorithm's effectiveness in parameter optimization and duration enhancement.

Keywords: Euler iteration method, monte Carlo, single-objective optimization model.

1. Introduction

In dynamic interactive scenarios within three-dimensional space, calculating occlusion durations between moving objects and optimizing related parameters are central to achieving precise dynamic control. These calculations directly impact the performance evaluation and strategy design of relevant systems [1]. Current traditional computational methods for such scenarios often face challenges such as insufficient iterative accuracy in dynamic coordinate updates and low optimization efficiency under multiple parameter constraints, making it difficult to balance computational accuracy with flexibility in parameter adjustment.

To address these issues, this paper focuses on the quantitative analysis and optimization of dynamic occlusion duration for moving objects. First, considering the dynamic characteristics of moving objects in 3D space, the Euler iteration method is introduced to establish a coordinate update mechanism [2]. By setting a small-time step, it achieves high-precision iterative updates of real-time object coordinates, providing a reliable dynamic data foundation for subsequent occlusion determination. Second, integrating spatial geometry principles [3] and sign functions [4], we establish scientific rules for determining occlusion validity, defining quantitative standards for occlusion conditions to achieve precise calculation of occlusion duration in specific scenarios. Finally, addressing parameter tuning requirements, we design an optimization framework based on the Monte Carlo algorithm [5]. Within predefined parameter constraints, this framework explores optimal parameter combinations through random sampling and multiple numerical simulations to maximize occlusion duration. The research process and methodology presented herein provide an algorithmic model and parameter optimization approach applicable to occlusion scenarios in three-dimensional dynamic environments, offering both theoretical and practical value [6].

2. Analysis of Masking Duration Under Specified Parameters

The initial positions of the missile and UAV are shown in Fig. 1 below.

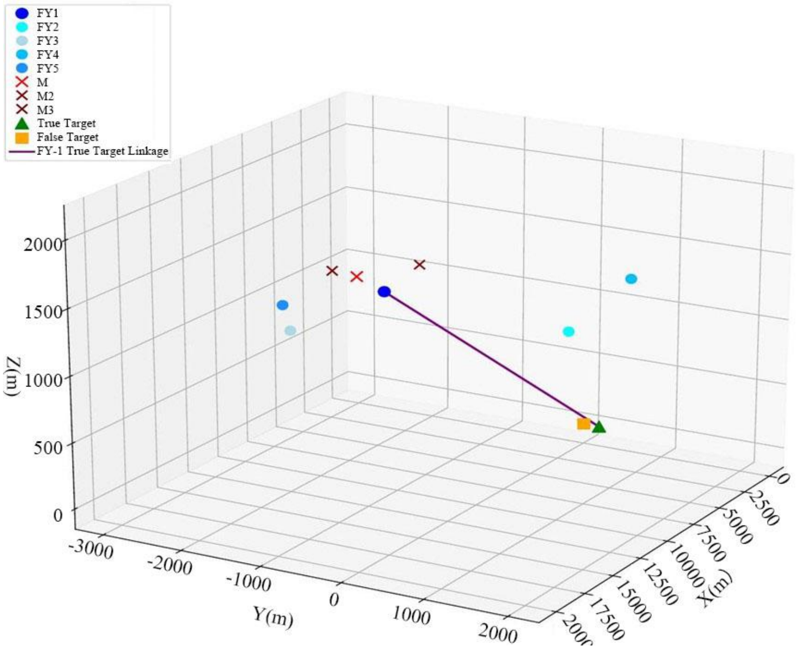

Fig. 1 Schematic Diagram of Initial Positions for Missile and Drone

Let the real-time coordinates of the drone or smoke grenade be $A(x_{11}, y_{11}, z_{11})$, and the real-time coordinates of the missile be $B(x_{21}, y_{21}, z_{21})$; The coordinates of the real target are $C(x_3, y_3, z_3)$, and those of the decoy target are $D(x_4, y_4, z_4)$. The drone's operational velocity is $v_{11} = 120$ m/s, and the missile's operational velocity is $v_{21} = 300$ m/s.

The drone is currently flying toward the real target, assuming its position along the z-axis remains constant. Let the unit direction vector of the drone's motion be $(\cos\,\alpha_{11}, \sin\,\alpha_{11}, 0)$, where $(\cos\,\alpha_{11})^2 + (\sin\,\alpha_{11})^2 = 1$.

The missile flies toward the decoy target. Let the unit direction vector of the missile's motion be $(\cos\,\alpha_2, \cos\,\beta_2, \cos\,\gamma_2)$, where: $(\cos\,\alpha_{21})^2 + (\cos\,\beta_{21})^2 + (\cos\,\gamma_{21})^2 = 1$.

It follows readily that:

$$\begin{cases} \cos\,\alpha_{11} = \frac{x_3 - x_{11}}{\sqrt{(x_3-x_{11})^2+(y_3-y_{11})^2}} \\ \sin\,\alpha_{11} = \frac{y_3 - y_{11}}{\sqrt{(x_3-x_{11})^2+(y_3-y_{11})^2}} \\ \cos\,\alpha_{21} = \frac{x_4 - x_{21}}{\sqrt{(x_4-x_{21})^2+(y_4-y_{21})^2+(z_4-z_{21})^2}} \\ \cos\,\beta_{21} = \frac{y_4 - y_{21}}{\sqrt{(x_4-x_{21})^2+(y_4-y_{21})^2+(z_4-z_{21})^2}} \\ \cos\,\gamma_{21} = \frac{z_4 - z_{21}}{\sqrt{(x_4-x_{21})^2+(y_4-y_{21})^2+(z_4-z_{21})^2}} \end{cases} \tag{1}$$

2.1. Coordinate Update Based on Euler Iteration Method

The coordinate update principle follows the Euler iteration method. Let the function before update be $g(t^-)$, with its time derivative h. Since the time interval Δt is sufficiently small, h can be considered constant within this interval. The update formula is:

$$g(t) = g(t^-) + h\Delta t \tag{2}$$

Coordinate updates are performed. Taking a drone as an example, the pre-update coordinates are $(x_1{}^-, y_1{}^-, z_1{}^-)$, and the post-update coordinates are (x_1, y_1, z_1). Each update interval is $\Delta t = t - t^- = 0.001s$. The coordinate update formulas for the drone and missile are as follows:

$$\begin{cases} x_{11}(t) = x_{11}(t^-) + v_{11,x}\Delta t = x_{11}(t^-) + v_{11}\cos\,\alpha_{11}\Delta t \\ y_{11}(t) = y_{11}(t^-) + v_{11,y}\Delta t = y_1(t^-) + v_{11}\sin\,\alpha_{11}\Delta t \\ z_{11}(t) = z_{11}(t^-) \\ x_{21}(t) = x_{21}(t^-) + v_{21,x}\Delta t = x_{21}(t^-) + v_{21}\cos\,\alpha_{21}\Delta t \\ y_{21}(t) = y_{21}(t^-) + v_{21,y}\Delta t = y_{21}(t^-) + v_{21}\cos\,\beta_{21}\Delta t \\ z_{21}(t) = z_{21}(t^-) + v_{21,z}\Delta t = z_{21}(t^-) + v_{21}\cos\,\gamma_{21}\Delta t \end{cases} \tag{3}$$

For smoke decoy deployment by the drone, analysis is divided into two phases: deployment and detonation.

2.2. Phase 1: Smoke Decoy Deployment by Drone

At this moment, the drone releases the smoke decoy. Upon release, the decoy's velocity matches the drone's velocity. The drone releases the smoke decoy at time t_{11}. The coordinates at the release point are:

$$(x_{11}^1(0), y_{11}^1(0), z_{11}^1(0)) = \left(x_{11}(t_{11}), y_{11}(t_{11}), z_{11}(t_{11})\right) \tag{4}$$

The smoke grenade moves uniformly along the x and y axes. Since the drone's initial velocity along the z-axis is zero, the smoke grenade's motion along the z-axis can be modeled as accelerated motion with initial velocity zero and acceleration g. The coordinate update at this point is as follows:

$$\begin{cases} x_{11}^1(t) = x_{11}^1(t^-) + v_{11,x}\Delta t = x_{11}^1(t^-) + v_{11}\cos\,\alpha_{11}\Delta t \\ y_{11}^1(t) = y_{11}^1(t^-) + v_{11,y}\Delta t = y_{11}^1(t^-) + v_{11}\cos\,\beta_{11}\Delta t \\ z_{11}^1(t) = z_{11}^1(t^-) + v_{11,z}\Delta t = z_{11}^1(t^-) + \left(v_{11,z}(t^-) - g\Delta t\right)\Delta t \end{cases} \tag{5}$$

2.3. Phase Two: Smoke Jammer Detonation

After deployment, the smoke jammer detonates after t_{21} seconds. The cloud cluster's x, y, and z-axis velocities are 0, 0, and $-3\ m/s$, respectively. Coordinate updates at this stage are as follows:

$$\begin{cases} x_{11}(t) = x_{11}(t^-) \\ y_{11}(t) = y_{11}(t^-) \\ z_{11}(t) = z_{11}(t^-) - 3\Delta t \end{cases} \tag{6}$$

2.4. Determining Effective Missile Obstruction

The following evaluates whether the cloud generated by the smoke grenade can block the missile.

First, the cloud must remain within its effective duration. For the cloud, if more than twenty seconds elapse after detonation, it cannot provide effective obstruction.

Next, assuming the cloud remains within its effective duration: At a given time t', the center of the smoke point $A(x_1, y_1, z_1)$, the missile point $B(x_2, y_2, z_2)$, and the true target center point (x_3, y_3, z_3) form a triangle as shown in Fig. 2 below.

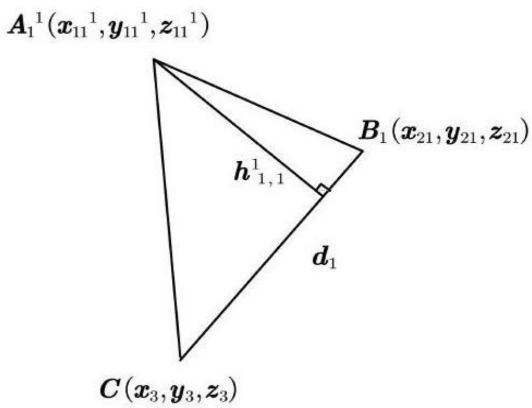

Fig. 2 Image formed by $A_{1,1}^1, B_1, C$

First, the cloud cluster's z-coordinate must be less than the missile's z-coordinate for it to potentially obscure the true target C. According to the requirement, if the distance between the cloud cluster center and the missile does not exceed $10m$, i.e., $h_{1,1}^{1} \leq 10m$, it is deemed a valid obstruction. The calculation formula for $h_{1,1}^{1}$ is as follows:

$$\begin{cases} S_{\Delta A_1^1 B_1 C} = \frac{1}{2}\left\|\overrightarrow{A_1^1 B_1} \times \overrightarrow{A_1^1 C}\right\| \\ \overrightarrow{A_1^1 B_1} \times \overrightarrow{A_1^1 C} = \begin{vmatrix} \vec{i} & \vec{j} & \vec{k} \\ x_{21} - x_{11}^1 & y_{21} - y_{11}^1 & z_{21} - z_{11}^1 \\ x_3 - x_{11}^1 & y_3 - y_{11}^1 & z_3 - z_{11}^1 \end{vmatrix}, \\ d_1 = \sqrt{(x_3 - x_{21})^2 + (y_3 - y_{21})^2 + (z_3 - z_{21})^2} \\ h_{1,1}^1 = \frac{S_{\Delta A_1^1 B_1 C}}{\frac{1}{2} d_1}. \end{cases} \tag{7}$$

Thus, the distance $h_{1,1}^{1}$ between the missile and cloud cluster at different times can be determined. At this point, the cumulative masking time is defined as function U. The formula is as follows:

$$U = \sum_{m=1}^{\frac{20}{\Delta t}} \Delta t I_1(t_m) \tag{8}$$

Where:

$$I_1(t_m) = \begin{cases} 1, \text{ Missile 1 was effectively obscured during the m th time interval} \\ 0, \text{ Other} \end{cases} \tag{9}$$

$I_1(x_1(m)) = 1$ To satisfy this, both conditions must be met simultaneously:
(1) The drone's coordinates are less than the missile's x-axis coordinates.
(2) The calculated $h_{1,1}^{1} \leq 10$ according to the formula.
Combining these two conditions yields the expression for $I_1(t_m)$ as follows:

$$I_1(t_m) = \begin{cases} 1, & \text{if } \frac{1+\text{sgn}\left(x_{11}^1(m\Delta t) - x_{21}(t_{11}+t_{21}+m\Delta t)\right)}{2} + \frac{1+\text{sgn}\left(h_{1,1}^1(m\Delta t) - 10 - 10^{-8}\right)}{2} = 0, \\ 0, & \text{otherwise.} \end{cases} \tag{10}$$

Where:

$$\text{sgn}(x) = \begin{cases} 1, x > 0 \\ 0, x = 0 \\ -1, x < 0 \end{cases} \tag{11}$$

2.5. Calculation of Effective Obstruction Duration

This section reveals that:

$$\begin{cases} v_{11} = 120 \text{ m/s}, v_{21} = 300 \text{ m/s}; \\ (x_3, y_3, z_3) = (0,200,5), (x_4, y_4, z_4) = (0,0,0); \\ t_{11}^1 = 1.5 \text{ s}, t_{21}^1 = 3.6 \text{ s}; \\ (x_{11}(0), y_{11}(0), z_{11}(0)) = (17800,0,1800) \\ (x_{21}(0), y_{21}(0), z_{21}(0)) = (20000,0,2000) \end{cases} \tag{12}$$

Based on the determined information above, the effective obstruction duration can be simply calculated via the formula as 1.39s.

2.6. Solving the Model Considering Air Resistance

In the initial analysis of the smoke grenade's first phase, it was assumed to move uniformly along the x and y axes. If air resistance acts on the x and y axes, we examine how the effective concealment time changes. The essence of air resistance is to impede relative motion.

The air resistance formula is as follows:

$$F = -\frac{1}{2}kv^2 \tag{13}$$

The magnitude of air resistance along the x-axis is $-\frac{1}{2}kv^2{}_{11,x}$; the magnitude along the y-axis is $-\frac{1}{2}kv^2{}_{11,y}$.

Applying Newton's Second Law yields:

$$\frac{dv}{dt} = -\frac{F}{m} = -\frac{\frac{1}{2}kv^2}{m} = -\frac{kv^2}{2m} \tag{14}$$

With air resistance incorporated, the coordinate updates for the smoke interference grenade during the first phase are as follows:

$$\begin{cases} x_{11}^1(t) = x_{11}^1(t^-) + v_{11,x}\Delta t = x_{11}^1(t^-) + \left(v_{11,x}(t^-) + \frac{kv_{11,x}(t^-)^2}{2m}\Delta t\right)\Delta t \\ y_{11}^1(t) = y_{11}^1(t^-) + v_{11,y}\Delta t = y_{11}^1(t^-) + \left(v_{11,y}(t^-) - \frac{kv_{11,y}(t^-)^2}{2m}\Delta t\right)\Delta t \\ z_{11}^1(t) = z_{11}^1(t^-) + v_{11,z}\Delta t = z_{11}^1(t^-) + \left(v_{11,z}(t^-) - g\Delta t\right)\Delta t \end{cases} \tag{15}$$

After accounting for air resistance, the calculated effective masking time is 0.171s. This differs significantly from the 1.39s obtained without considering air resistance. Therefore, when analyzing the motion of the smoke-screening decoy, the effect of air resistance is disregarded.

3. Investigation of Masking Duration Through Parameter Adjustment

3.1. Establishment of a Single-Objective Optimization Model

Building upon the parameter exploration conducted earlier, this section requires adjusting certain fundamental data to maximize masking duration. For this single-objective optimization model, the optimal solution must be identified within the specified parameter range. The optimization objective is to maximize masking duration, defined as the objective function U. The formula is as follows:

$$U = \sum_{m=1}^{\frac{20}{\Delta t}} \Delta t I_1(t_m) \tag{16}$$

Where:

$$I_1(t_m) = \begin{cases} 1, & \frac{1+\mathrm{sgn}\left(x_{11}^1(m\Delta t) - x_{21}(t_{11}+t_{21}+m\Delta t)\right)}{2} + \frac{1+\mathrm{sgn}\left(h_{1,1}^1(m\Delta t) - 10 - 10^{-8}\right)}{2} = 0, \\ 0, & \text{Other.} \end{cases} \tag{17}$$

The constraints are as follows:

$$\begin{cases} v_{11} \in (70,140)\mathrm{m/s} \\ \alpha_{11} \in (0,\pi) \\ t_{11}^1 \in (1,5)\mathrm{s} \\ t_{21}^1 \in (1,5)\mathrm{s} \end{cases} \tag{18}$$

3.2. Solving the Single-Objective Optimization Model Using the Monte Carlo Cloaking Optimization Algorithm

Based on the determined parameters and the above constraints, the Monte Carlo screening optimization algorithm is employed for optimization. The core of this algorithm involves: randomly sampling the deployment time and detonation time of smoke-screening decoys, numerically simulating the three-dimensional motion of the missile and cloud cluster through iterative calculations, determining missile screening based on distance calculations derived from spatial geometry, and selecting the longest screening duration from multiple simulations.

The algorithm's operational steps are as follows:

1. Initialize drone flight speed, direction, and other parameters, while defining the range of constraint data.

2. Random sampling: Generate initial values for drone speed, smoke grenade deployment time, and detonation time within the constraint ranges.

3. Using the sampled data, iterate the motion of the missile, drone, smoke decoy, and its generated cloud cluster with a time step Δt.

4. Record and accumulate the duration of missile concealment based on the distance determination method derived from spatial geometry calculations.

5. Through multiple random samples, derive various durations of missile concealment. Ultimately save the longest concealment time and its corresponding parameters, then output the results.

The longest concealment time without considering air resistance is 4.4520 s. The longest concealment time with air resistance is 0.176 s.

4. Conclusion

This paper investigates the calculation and parameter optimization of occlusion duration in dynamic three-dimensional scenes, integrating Euler iteration and Monte Carlo algorithms.

First, the coordinate update model based on Euler iteration enables precise iteration of real-time coordinates for dynamic objects. By setting a small-time step and designing coordinate calculation logic tailored to different motion phases, it reliably outputs 3D position data. Comparing results with and without resistance (occlusion duration reduced from 1.39s to 0.171s) validates the rationality of simplifying the model by ignoring resistance in this scenario. Secondly, the occlusion detection model, combining spatial geometry principles with sign functions, enables quantitative calculation of occlusion duration. By deriving key distances through vector cross-product operations and defining evaluation metrics with sign functions, the model accurately identifies occlusion time intervals and accumulates durations, ensuring computational precision. Finally, the Monte Carlo single-objective optimization framework effectively solves the problem of maximizing occlusion duration under multiple parameter constraints. Within the preset parameter range, random sampling and simulation yielded the optimal occlusion duration (4.4520 seconds when resistance is disregarded).

References

[1] Yu Zhou, Shan Ganlin, Duan Xiusheng, et al. Optimal Multi-Sensor Station Deployment in 3D Space Under Obstacle Occlusion [J]. Journal of Detection and Control, 2019, 41 (02): 110-114+119. DOI: CNKI: SUN: XDYX.0.2019-02-020.

[2] Li Ronghui, Chen Zhijuan, Li Zongxuan, et al. Underactuated Surface Vessel Path Tracking Control Based on Euler Iterative Model Prediction [J]. Journal of Guangdong Ocean University, 2020, 40 (01):104-110.

[3] He, D. H., Sun, F. P., Xiao, K., et al. Modeling and Validation of BeiDou Tri-frequency IF-PPP Based on Spatial Geometric Principles [J]. Geodesy and Geodynamics, 2020, 40 (12):1290-1293+1320. DOI: 10.14075/j.jgg.2020.12.016.

[4] Ge Xinguang, Lu Jiakang, Zhang Lirong, et al. Efficient Numerical Solution for a Class of Infinite Integrals Containing Sign Functions [J]. Journal of Guangxi University of Science and Technology, 2024, 35 (03): 48-55. DOI: 10.16375/j.cnki.cn45-1395/t.2024.03.007.

[5] Deng Ziyi. Quasi-Monte Carlo Estimation for Eigenvalue Problems with Random Coefficients [D]. Hangzhou Dianzi University, 2025. DOI: 10.27075/d.cnki.ghzdc.2025.000395.

[6] Wang Anlan, Li Fangchun, Zhang Yanci. Anisotropic Screen-Space Ambient Occlusion Algorithm for 3D Flow Visualization [J]. Research on Computer Applications, 2024, 41 (10): 3166-3172. DOI: 10.19734/j.issn.1001-3695.2024.01.0012.