**2025年数学建模竞赛**

**问题A：烟幕干扰弹的投放策略**

根据干扰弹主要通过在光学或红外传感器敏感带宽内释放大量干扰云团，使目标方形成宽频遮蔽干扰或为假目标，具有隐蔽性、欺骗性和多样性。随着新型干扰技术的不断发展，现已有干扰技术方案可实现干扰干扰弹空中编队，叠加释放能量的错峰控制精确干扰时间与空间。

现考虑利用无人机完成某空域的战术烟幕干扰。具有任务能力的无人机能够按照预定的轨迹投放烟幕干扰弹。无人机投放烟幕干扰弹在真目标和假目标之间形成遮蔽云团，每架无人机既能投掷干扰弹间隔时间为1 s。当干扰弹被投放无人机后在外力作用下运动，爆炸后立即形成球形遮蔽云团。由于不同炸点起爆后形成的烟幕云团，其云团下沉速度、半径以及有效遮蔽时间均有差异，现考虑一个云团的下沉速度为3 m/s，起爆后瞬时成球，云团中心在起爆20 s内可对目标造成有效遮蔽。

导弹飞行速度为恒定值300 m/s。导弹飞行方向估计为一个较宽的半个圆，高度约为目标同高度。导弹初始位置已知，以目标点为坐标原点，定义*x*方向为导弹飞行方向，y 方向为侧向，z 方向为高度。真目标坐标为 (0,200,0)，假目标坐标为原点。M1、M2、M3 分别位于 (20000,0,2000)、(19000,600,2100)、(18000,-600,1900)。5 架无人机的初始位置分别为 FY1(17800,0,1800)、FY2(12000,1400,1400)、FY3(6000,-3000,700)、FY4(11000,2000,1800)、FY5(13000,-2000,1300)。

无人机可瞬时定向，以70–140 m/s等高匀速直线飞行。无人机可投放两枚烟幕干扰弹，两次投放间隔不小于1 s。烟幕干扰弹弹体在脱离无人机后仅受重力作用，其爆炸时间由时序引信控制。利用无人机协同投放的战术烟幕干扰弹，在真目标处形成的总遮蔽时间越长越好。总遮蔽时间是指在真目标处烟幕干扰弹的云团对其进行遮蔽的有效时间，可不连续。

问题1

利用无人机FY1仅投放1枚干扰弹遮蔽真目标的干扰过程。若FY1以120 m/s的速度朝向真目标飞行，求在投放后1.5 s后起爆1枚烟幕干扰弹时，真目标的总遮蔽时间。请给出飞行器的飞行轨迹、遮蔽过程及总遮蔽时间。

**无人机协同投放烟幕干扰弹的建模与优化策略研究**

**摘要**

在现代防御作战中，精确制导武器对关键目标构成了严重威胁。为提高目标的生存能力，本文基于无人机协同投放烟幕干扰弹的战术设定，研究了在导弹攻击过程中构建动态烟幕屏障的优化问题。首先，建立了导弹、无人机、干扰弹及烟幕云团的运动学模型，并明确了遮蔽判据与效能评价指标。随后，针对不同层次的任务需求，依次研究了单机单弹、单机多弹、多机协同及全局优化等五个子问题，逐步构建了从局部战术到整体防御的建模框架。

在求解方法上，本文结合几何判定、时间离散化近似与优化算法，对投放位置、延时起爆时刻及无人机飞行参数进行了系统分析。结果表明，合理规划无人机的航迹与投放策略，能够显著延长真目标的总有效遮蔽时长。例如，在单机单弹情形下，FY1投放烟幕干扰弹可实现约1.41 s的遮蔽；通过优化参数组合，可将遮蔽时长提升至4.72 s；进一步扩展为单机多弹与多机协同时，遮蔽时长可达到数秒至十余秒，并形成较强的连续覆盖效应。最终在全局最优策略下，五机多弹协同能够构建稳定的防护屏障，实现对多方向来袭导弹的鲁棒遮蔽。

综上，本文提出的建模方法不仅验证了无人机协同投放烟幕弹的可行性，还揭示了投放时序与空间布局对遮蔽效能的关键影响。研究成果可为复杂防御环境下的战术规划与装备效能评估提供理论依据与方法参考。

关键词：无人机，烟幕干扰弹，导弹拦截，对抗建模，优化策略

**目录**

[1问题重述 3](#_Toc207917896)

[1.1问题背景 3](#_Toc207917897)

[1.2问题提出 3](#_Toc207917898)

[2问题分析 4](#_Toc207917899)

[3模型假设 4](#_Toc207917900)

[4符号说明 5](#_Toc207917901)

[5模型的建立与求解 5](#_Toc207917902)

[5.1问题一 5](#_Toc207917903)

[5.2问题二 9](#_Toc207917904)

[5.3问题三 17](#_Toc207917905)

[5.4问题四 27](#_Toc207917906)

[5.4问题五 38](#_Toc207917907)

[6模型检验与灵敏度分析 49](#_Toc207917908)

[6.1 模型检验 49](#_Toc207917909)

[6.2 灵敏度分析 50](#_Toc207917910)

[7模型的评价与推广 50](#_Toc207917911)

[7.1优点 50](#_Toc207917912)

[7.2缺点 50](#_Toc207917913)

[7.3改进 50](#_Toc207917914)

[7.4推广 51](#_Toc207917915)

[参考文献 51](#_Toc207917916)

[附录 53](#_Toc207917917)

[附件 1：问题一matlab代码 53](#_Toc207917918)

[附件 2：问题二matlab代码 55](#_Toc207917919)

[附件 3：问题三matlab代码 61](#_Toc207917920)

[附件 4：问题四matlab代码 68](#_Toc207917921)

[附件 5：问题五matlab代码 77](#_Toc207917922)

# 1问题重述

## 1.1问题背景

在现代防御作战环境中，来袭精确制导武器（如空地导弹）对关键目标构成严重威胁。为了提升目标的生存能力，烟幕干扰技术作为一种典型的软对抗手段得到广泛应用。该技术主要依靠烟幕干扰弹在空中形成高浓度气溶胶或烟雾云团，从而遮蔽、干扰导引头的探测视线，达到掩护真目标的目的。相比硬杀伤手段，烟幕干扰具有部署灵活、成本低廉、效费比高等突出优势。

近年来，随着无人机平台的快速发展，将无人机作为烟幕干扰弹的投放载体成为可能。无人机不仅具备长航时、灵活机动的特点，还能够通过精确的航迹控制与时间引信，实现烟幕干扰弹的定点投放与起爆控制。这为提升烟幕遮蔽效果、实现对多枚来袭武器的动态防护提供了新的思路。

## 1.2问题提出

针对导弹对真目标的威胁，现考虑采用多架无人机协同投放烟幕干扰弹的方式，在导弹与真目标之间构建动态烟幕屏障。问题的核心在于：如何合理规划无人机的飞行方向与速度、确定烟幕干扰弹的投放位置与起爆时刻，使得形成的烟幕云团能够在导弹飞行过程中尽可能长时间遮蔽真目标。

具体而言，需要解决以下关键问题：

1. 建立导弹、无人机、烟幕弹及烟幕云团的运动模型；
2. 确定烟幕遮蔽的判定条件与有效性标准；
3. 在给定导弹初始状态与无人机部署位置的条件下，优化烟幕干扰弹的投放策略，使多枚云团在时间轴上形成最大化的有效遮蔽时长。

这一问题不仅涉及动力学建模与几何约束，还包含多目标优化与组合策略设计，属于典型的装备效能建模与对抗策略优化问题。

# 2问题分析

针对问题一，本问题为基础场景，要求分析单架无人机（FY1）在既定航迹下投放单枚烟幕干扰弹的遮蔽效果。关键点在于建立导弹、无人机与烟幕弹的运动学方程，确定烟幕云团的空间位置与时间演化，并通过几何判定导弹—真目标视线是否落入有效遮蔽区。此问题旨在验证模型的基本可行性，并为后续多弹、多机协同优化奠定计算框架。

针对问题二，在单机单弹的基础上，进一步引入航向角、飞行速度、投放时刻与延时起爆作为决策变量。问题转化为约束条件下的优化问题：在不同参数组合下，烟幕云团对导弹视线的遮蔽时长存在显著差异，需要通过枚举或优化算法寻找最优解。该问题的难点在于非光滑目标函数的处理以及连续时间窗的离散化近似。

针对问题三，该问题扩展为单机多弹投放，要求合理安排三枚烟幕干扰弹的投放与起爆时序，以最大化遮蔽时长的并集。需要考虑“同机两弹间隔≥1s”的约束，并解决多枚云团在时间轴上的覆盖与衔接问题。问题具有组合优化的特征，既要保证每一枚弹单独起效，又要实现整体时长的最大化，体现了典型的时序规划问题。

针对问题四，进一步扩展至多机协同，即由三架无人机在不同初始位置与方向下各自投放一枚干扰弹。此时问题复杂度显著提升，不仅要考虑单机运动参数，还需协调不同无人机之间的投放策略。关键在于如何通过合理分工，使三枚烟幕弹的有效区间在时间上实现互补，最大化对真目标的整体遮蔽效果。该问题涉及多维变量的联合搜索，属于典型的协同优化问题

针对问题五，最终问题综合考虑全部五架无人机与多枚干扰弹，构建全局最优投放策略。此时变量数量与组合规模极大，需要引入启发式搜索或分布式优化方法来降低计算复杂度。同时，问题五还需兼顾不同导弹来袭方向，确保遮蔽效果具备鲁棒性与稳定性。该问题代表了从单机到多机、多弹协同的全面扩展，强调全局策略设计与战术效能最大化

# 3模型假设

1. 来袭导弹以恒定速度沿直线飞向假目标，不考虑机动与规避。
2. 无人机可在任务开始时瞬时调整航向与速度，之后保持等高度直线匀速飞行。
3. 烟幕干扰弹脱离无人机后仅受重力作用，起爆后瞬时形成球形烟幕云团，并以固定速度下沉。
4. 云团在起爆后20 s内对真目标具有遮蔽效能，超过该时间则视为无效。
5. 遮蔽判据为：导弹至真目标的视线与云团球心的最短距离小于等于 10 m 时，真目标被有效遮蔽。
6. 多枚烟幕干扰弹的效能可叠加，总遮蔽时间以时间并集计算。

# 4符号说明

|  |  |
| --- | --- |
| **符号** | **含义** |
| O | 假目标圆心坐标 (0,0,0)(0,0,0) |
| T | 真目标底面圆心坐标 (0,200,0)(0,200,0) |
| M1,M2,M3 | 三枚来袭导弹的初始位置坐标 |
| FY1,…,FY5 | 五架无人机的初始位置坐标 |
| *V*m | 导弹飞行速度，取 300m/s |
| *v*f | 第*f*架无人机飞行速度 |
| *ψ* | 第*f*架无人机飞行航向角 |
| *t*relt | 烟幕干扰弹的投放时刻 |
| τ | 烟幕干扰弹的延时起爆时间 |
| *r*rel | 烟幕干扰弹的投放点坐标 |
| *r*exp | 烟幕干扰弹的起爆点坐标 |
| c(t) | 烟幕云团在时刻*t*的球心位置 |
| *R* | 烟幕云团有效半径（10 m） |
| *v*s | 烟幕云团匀速下沉速度（3 m/s） |
| *g* | 重力加速度（9.8 m/s2） |
| *Y*(t) | 时刻*t*真目标的遮蔽状态指示函数 |
| *J* | 真目标的总有效遮蔽时长（优化目标） |

# 5模型的建立与求解

## 5.1问题一

首先绘制出真目标、假目标、三个导弹与五个无人机的坐标，如下图所示：



导弹速度：300 m/s，指向假目标O=(0,0,0)。

初始位置：M1=(20000,0,2000)。

方向单位向量



导弹位置



无人机 FY1 以 120 m/s 朝向假目标方向、等高直线飞行（FY1 在 y=0，朝向 O 即沿 -x 方向，水平飞行）。

初始位置：FY1=(17800,0,1800)。

无人机速度：



投放与起爆：投放时刻=1.5 s；起爆延时3.6s，故=5.1 s。

投放点：



投放后仅受重力影响（取 g=9.8 m/s²，投放瞬间垂直速度为 0），在内：



起爆位置（将代入）



起爆后，云团中心以 3 m/s 匀速下沉；起爆后 20 s 内、以云团中心为球心、半径 10 m 的球域为“有效遮蔽区”。

令，云团中心



有效遮蔽时间窗：



有效遮蔽几何条件：对每个 t，导弹到真目标的视线段（其中真目标底面圆心T=(0,200,0)）与半径10的球B(c(t),10) 有交，即



记，则对任一点P=c(t)，



将代入，上式给出每个 t 的最小距离。满足距离 ≤10 的 t 构成若干时间区间，其并集长度即为“有效遮蔽时长”。

按上式逐时检查可得：

最小距离约出现在t≈9.419 s，最小值约 4.718 m（小于 10 m，满足遮蔽）。

满足遮蔽条件的时间区间约为



因而有效遮蔽时长为



在给定投放、起爆与运动学参数下，FY1 投放的这枚烟幕干扰弹对 M1 的有效遮蔽时长约为 1.41 s，且该遮蔽发生在约 8.04–9.45 s 这一时间段内。

|  |
| --- |
| 工具：matlab |
| %% 问题 1: FY1 投放 1 枚烟幕干扰弹对 M1 的遮蔽时长  %% 问题 1: FY1 投放 1 枚烟幕干扰弹对 M1 的遮蔽时长  clear; clc;  %% 常量与初始条件  g = 9.8; % 重力加速度 m/s^2  Vm = 300; % 导弹速度 m/s  Vu = 120; % 无人机速度 m/s  t\_release = 1.5; % 投放时刻 s  delay = 3.6; % 延时起爆 s  t\_explosion = t\_release + delay;  % 真目标  T = [0, 200, 0];  % M1 初始位置  M1 = [20000, 0, 2000];  % 无人机 FY1 初始位置  FY1 = [17800, 0, 1800];  % 假目标(原点)  O = [0, 0, 0];  %% 导弹方向  um = (O - M1) / norm(O - M1);  %% 无人机速度向量 (朝向原点，x 轴负方向)  uu = (O - FY1); uu(3) = 0; uu = uu / norm(uu); % 等高度，只取水平向量  uu = uu \* Vu;  %% 投放点  r\_release = FY1 + uu \* t\_release;  %% 起爆点  dt = delay;  r\_explosion = r\_release + uu \* dt + 0.5\*[0,0,-g]\*dt^2;  %% 云团中心轨迹  % c(t) for t >= t\_explosion  c = @(t) r\_explosion + [0,0,-3]\*(t - t\_explosion);  %% 导弹轨迹  rm = @(t) M1 + Vm\*t\*um;  %% 点到线段距离函数  distPoint2Segment = @(P,A,B) ...  norm(P - (A + max(0,min(1, dot(P-A,B-A)/norm(B-A)^2))\*(B-A)));  %% 扫描时间区间  t\_range = t\_explosion:0.01:(t\_explosion+20);  mask = false(size(t\_range));  for k = 1:length(t\_range)  t = t\_range(k);  A = rm(t); B = T; P = c(t);  d = distPoint2Segment(P,A,B);  if d <= 10  mask(k) = true;  end  end  %% 找出连续区间  d\_mask = diff([false mask false]);  start\_idx = find(d\_mask==1);  end\_idx = find(d\_mask==-1)-1;  intervals = [t\_range(start\_idx)' , t\_range(end\_idx)'];  durations = intervals(:,2)-intervals(:,1);  total\_duration = sum(durations);  %% 输出结果  disp('有效遮蔽时间区间(s):');  disp(intervals);  fprintf('总有效遮蔽时长 = %.3f s\n', total\_duration); |
| 运行结果  有效遮蔽时间区间(s):  8.0380 9.4480  总有效遮蔽时长 = 1.410 s |

## 5.2问题二

假目标原点O=(0,0,0)，真目标底面圆心T=(0,200,0)。

重力加速度g=9.8 m/s2。

导弹M1初始位置M=(20000,0,2000)，速度大小*V*m=300 m/s，指向O：



无人机FY1初始位置F=(17800,0,1800)，接令后等高度、匀速直线飞行。

决策变量表示为：



其中ψ为无人机水平航向角（绕*z*轴，*x*轴正向到飞行方向的夹角，逆时针为正）。*v*为无人机速度，约束70<*v*<140（m/s）。*t*release为投放时刻，*t*release>0。*τ*为延时起爆（时序引信），*τ*>0，起爆时刻*t*explosion=*t*release+*τ*。

无人机速度向量与轨迹（等高）:



投放点（*t*=*t*release）:



弹体起爆点（t=texplosion=trelease+τ，投放后仅受重力）:



云团中心（起爆后以3 m/s匀速下沉）:



有效遮蔽区：以 c(t) 为球心、半径*R*=10 m的球域；有效时间窗



令 A=rm(t)，B=T，P=c(t)。视线为线段[A,B]。点到线段距离：



其中*x*表示一切依赖于决策变量的量。

遮蔽判定：d(t;x)<R。

**优化目标：最大化有效遮蔽时长**

精确目标为:



（即满足距离不超过10 m的时间集合的测度）。最大化*J*(x)。

这是一个非光滑积分优化。数值上可用时间离散近似为可计算的目标函数：

取步长h（如0.01s），定义离散时间网格:



其中。

指示函数:



离散目标:



优化问题（离散形式）:



约束:



将硬阈值替换为平滑惩罚（近似指示函数）:



则:



最大化可用一般的无约束或带约束优化器；α取3-10可得到较硬的近似。

**利用matlab求解上述模型可得：**

**最优策略（近似）：**

**航向角psi=0.144 rad (8.25 deg)**

**飞行速度*v*=75.000 m/s**

**投放时刻*t*release=0.125 s**

**延时起爆*t*au=0.938 s (*t*explosion = 1.063 s)**

**投放点*r*release=[17809.278, 1.345, 1800.000]**

**起爆点*r*explosion=[17878.863, 11.435, 1795.693]**

**总有效遮蔽时长*J*≈4.720 s**

**时间区间（s）：**

**1.1625-5.8825**

|  |
| --- |
| **工具：matlab** |
| clear; clc;  %% --------------------- 常量与已知条件 ---------------------  g = 9.8; % 重力 m/s^2  R = 10; % 有效遮蔽半径 m  sink = 3; % 云团下沉速度 m/s  h = 0.02; % 时间离散步长 s（可调：0.01~0.05）  Vm = 300; % 导弹速度 m/s  O = [0, 0, 0]; % 假目标原点  T = [0, 200, 0]; % 真目标底面圆心  M = [20000, 0, 2000]; % M1 初始位置  F = [17800, 0, 1800]; % FY1 初始位置（等高度直飞）  um = (O - M) / norm(O - M); % 导弹方向单位向量  params = struct('g',g,'R',R,'sink',sink,'h',h,'Vm',Vm, ...  'O',O,'T',T,'M',M,'F',F,'um',um);  %% --------------------- 粗搜索（多起点） ---------------------  % 网格（可根据计算力调整）  psi\_grid = linspace(-pi, pi, 16); % 航向角  v\_grid = linspace(70, 140, 4); % 速度约束 [70,140]  trel\_grid = 0:0.5:4; % 投放时刻 (s)  tau\_grid = 0.5:0.5:5; % 延时起爆 (s)  best = struct('x',[],'J',-inf,'intervals',[],'r\_rel',[],'r\_exp',[]);  results = []; % [psi, v, trel, tau, J]  for psi = psi\_grid  for v = v\_grid  for trel = trel\_grid  for tau = tau\_grid  x = [psi, v, trel, tau];  [J, intervals, r\_rel, r\_exp] = objective\_J(x, params);  results(end+1,:) = [psi, v, trel, tau, J]; %#ok<AGROW>  if J > best.J  best.x = x; best.J = J;  best.intervals = intervals;  best.r\_rel = r\_rel; best.r\_exp = r\_exp;  end  end  end  end  end  % 选出若干最优候选用于细化  [~, idx\_sorted] = sort(results(:,5), 'descend');  topN = min(10, size(results,1));  candidates = results(idx\_sorted(1:topN), 1:4);  %% --------------------- 局部细化（坐标式爬山法） ---------------------  lb = [-pi, 70, 0, 0.05]; % 变量下界: psi, v, trelease, tau  ub = [ pi,140, 8, 10.00]; % 变量上界  step0 = [pi/24, 10, 0.25, 0.25]; % 初始步长  tolJ = 1e-4; % 收敛阈值  maxIter = 200; % 最大迭代  for i = 1:size(candidates,1)  [x\_loc, J\_loc, intervals\_loc, r\_rel, r\_exp] = local\_refine(candidates(i,:), params, lb, ub, step0, tolJ, maxIter);  if J\_loc > best.J  best.x = x\_loc; best.J = J\_loc;  best.intervals = intervals\_loc;  best.r\_rel = r\_rel; best.r\_exp = r\_exp;  end  end  %% --------------------- 输出最优结果 ---------------------  psi = best.x(1); v = best.x(2); trel = best.x(3); tau = best.x(4);  texp = trel + tau;  K = floor(20/h); %#ok<NASGU> % 供阅读：起爆后 20 s 的离散步数  fprintf('\n===== 最优策略（近似） =====\n');  fprintf('航向角 psi = %.3f rad (%.2f deg)\n', psi, rad2deg(psi));  fprintf('飞行速度 v = %.3f m/s\n', v);  fprintf('投放时刻 trelease = %.3f s\n', trel);  fprintf('延时起爆 tau = %.3f s (texplosion = %.3f s)\n', tau, texp);  fprintf('投放点 r\_release = [%.3f, %.3f, %.3f]\n', best.r\_rel);  fprintf('起爆点 r\_explosion = [%.3f, %.3f, %.3f]\n', best.r\_exp);  if isempty(best.intervals)  fprintf('无有效遮蔽（J = 0）\n');  else  dur = sum(best.intervals(:,2) - best.intervals(:,1));  fprintf('总有效遮蔽时长 J ≈ %.3f s\n', dur);  fprintf('时间区间（s）：\n');  disp(best.intervals);  end  %% --------------------- （可选）可视化 ---------------------  doPlot = false; % 如需可视化，设为 true  if doPlot  figure('Color','w'); hold on; grid on; axis equal; view(45,25);  xlabel('x / m'); ylabel('y / m'); zlabel('z / m');  % 真/假目标圆心  plot3(T(1),T(2),T(3),'gp','MarkerFaceColor','g','MarkerSize',10); text(T(1),T(2),T(3)+20,'True center');  plot3(O(1),O(2),O(3),'bp','MarkerFaceColor','b','MarkerSize',10); text(O(1),O(2),O(3)+20,'Fake center');  % M1 起点与导弹轨迹（20 s 期间）  t0 = texp; t1 = texp+20;  tt = linspace(t0,t1,200);  rm = M + (Vm\*tt.').\*um;  plot3(M(1),M(2),M(3),'r^','MarkerFaceColor','r'); text(M(1),M(2),M(3)+80,'M1');  plot3(rm(:,1), rm(:,2), rm(:,3), 'r-','LineWidth',1.2);  % FY1 轨迹（至投放）  uu = [v\*cos(psi), v\*sin(psi), 0];  tline = linspace(0,trel,50);  ru = F + tline.' .\* uu;  plot3(F(1),F(2),F(3),'ko','MarkerFaceColor','k'); text(F(1),F(2),F(3)+50,'FY1');  plot3(ru(:,1), ru(:,2), ru(:,3), 'k--');  % 投放点与起爆点  plot3(best.r\_rel(1), best.r\_rel(2), best.r\_rel(3),'mo','MarkerFaceColor','m'); text(best.r\_rel(1), best.r\_rel(2), best.r\_rel(3)+50,'release');  plot3(best.r\_exp(1), best.r\_exp(2), best.r\_exp(3),'ms','MarkerFaceColor','m'); text(best.r\_exp(1), best.r\_exp(2), best.r\_exp(3)+50,'explosion');  % 云团下沉轨迹（20 s）  c0 = best.r\_exp;  cLine = c0 + [zeros(numel(tt),1), zeros(numel(tt),1), -sink\*(tt.'-texp)];  plot3(cLine(:,1), cLine(:,2), cLine(:,3), 'm-','LineWidth',1.2);  title('Optimal plan (trajectory & key points)');  end  %% ====================== 函数定义 ======================  function [J, intervals, r\_release, r\_explosion] = objective\_J(x, P)  % 计算给定 x = [psi, v, trel, tau] 时的离散化遮蔽时长 J 及遮蔽区间  psi = x(1); v = x(2); trel = x(3); tau = x(4);  if v < 70 || v > 140 || trel < 0 || tau <= 0  J = 0; intervals = []; r\_release = [NaN,NaN,NaN]; r\_explosion = r\_release; return;  end  % 无人机运动（等高匀速）  uu = [v\*cos(psi), v\*sin(psi), 0];  r\_release = P.F + trel\*uu;  % 起爆点（仅受重力）  r\_explosion = r\_release + uu\*tau + 0.5\*[0,0,-P.g]\*(tau^2);  % 若起爆在地面以下，认为无效  if r\_explosion(3) <= 0  J = 0; intervals = []; return;  end  % 时间窗  texp = trel + tau;  t0 = texp; t1 = texp + 20;  K = floor((t1 - t0)/P.h); % ≈ floor(20/h)  tk = t0 + (0:K)'\*P.h;  % 云团中心 c(t)  c = r\_explosion + [0,0,-P.sink] .\* (tk - texp);  % 导弹轨迹 r\_m(t)  rm = P.M + (P.Vm\*tk).\*P.um;  % 线段 [A,B]，B = T  A = rm; B = repmat(P.T, size(A,1), 1);  BA = B - A; % 每个时刻的向量  PA = c - A;  % 点到线段距离（向量化）  BA2 = sum(BA.^2, 2);  s = sum(PA.\*BA, 2) ./ BA2; % 投影比例  s = max(0, min(1, s)); % clamp 到 [0,1]  Q = A + s .\* BA; % 最近点  d = sqrt(sum((c - Q).^2, 2)); % 距离  mask = (d <= P.R);  % 统计遮蔽区间与总时长  [intervals, J] = mask\_to\_intervals(tk, mask);  end  function [intervals, totalDur] = mask\_to\_intervals(tk, mask)  % 将布尔掩码转换成若干 [t\_start, t\_end] 区间，并返回总时长  if ~any(mask)  intervals = []; totalDur = 0; return;  end  mm = [false; mask; false];  dmm = diff(mm);  iStart = find(dmm == 1);  iEnd = find(dmm == -1) - 1;  intervals = [tk(iStart), tk(iEnd)];  totalDur = sum(intervals(:,2) - intervals(:,1));  end  function [x\_best, J\_best, intervals\_best, r\_rel\_best, r\_exp\_best] = local\_refine(x0, P, lb, ub, step0, tolJ, maxIter)  % 简单坐标式爬山法（无工具箱）  x\_best = clip(x0, lb, ub);  [J\_best, intervals\_best, r\_rel\_best, r\_exp\_best] = objective\_J(x\_best, P);  step = step0;  iter = 0; improved = true;  while iter < maxIter && improved  improved = false; iter = iter + 1;  for i = 1:4  for sgn = [+1, -1]  x\_try = x\_best;  x\_try(i) = x\_try(i) + sgn\*step(i);  x\_try = clip(x\_try, lb, ub);  [J\_try, intervals\_try, r\_rel\_try, r\_exp\_try] = objective\_J(x\_try, P);  if J\_try > J\_best + tolJ  x\_best = x\_try; J\_best = J\_try;  intervals\_best = intervals\_try;  r\_rel\_best = r\_rel\_try; r\_exp\_best = r\_exp\_try;  improved = true;  end  end  end  if ~improved  % 缩小步长继续细找  step = step \* 0.5;  if max(step) < 1e-3  break;  end  improved = true; % 允许再尝试一次更小步长  end  end  end  function x = clip(x, lb, ub)  % 变量边界裁剪  x = min(max(x, lb), ub);  end  function deg = rad2deg(rad)  % 角度单位转换（避免需要额外工具箱）  deg = rad \* 180/pi;  end |
| **运行结果：**  ===== 最优策略（近似） =====  航向角 psi = 0.144 rad (8.25 deg)  飞行速度 v = 75.000 m/s  投放时刻 trelease = 0.125 s  延时起爆 tau = 0.938 s (texplosion = 1.063 s)  投放点 r\_release = [17809.278, 1.345, 1800.000]  起爆点 r\_explosion = [17878.863, 11.435, 1795.693]  总有效遮蔽时长 J ≈ 4.720 s  时间区间（s）：  1.1625 5.8825 |

## 5.3问题三

导弹（M1），指向原点 O=(0,0,0)，速度*V*m=300：



无人机（FY1）自初始点F=(17800,0,1800)起，等高度匀速直线飞行。设航向角ψ（相对*x*轴，逆时针为正），速度*v*（约束 70≤v≤140）：



为降低维度并贴合“朝原点飞”的直觉，我把ψ固定为180°（即ψ=π），只对*v*做离散搜索（100、120、140 m/s）。

第*i*枚：投放时刻，延时引信𝜏(𝑖)，起爆时刻。

投放点：



起爆点（仅受重力，g=9.8）：



起爆后云团中心以 3 m/s 匀速下沉，半径 R=10 的球域在 20 s 内有效：



令真目标底面圆心*T*=(0,200,0)。对任意时刻*t*，

视线线段：



云团球心：



点到线段距离



若d(t)≤R，则该枚烟幕在该时刻对导弹视线有效遮蔽。

三枚弹满足投放约束：



每枚弹只在自己的有效窗内产生遮蔽。

目标：最大化并集遮蔽时长（允许不连续）：



为做机内计算，采用时间离散步长*h*（本求解中取*h*=0.05 s）：

统一构造一条时间网格，覆盖三枚弹可能的全部有效窗（这里取）。

对第 i 枚弹，定义掩码



单枚弹时长近似



三枚并集用布尔“或”：



为兼顾速度与效果，采用“离散网格 + 贪心三步”：

设定航向ψ=π（180°，朝原点），速度 v ∈ {100,120,140} m/s（3 个候选）。

在网格



上枚举，首先为第一枚选择能使*J*1最大的组合。

第二枚在满足的网格中，选择能使并集*J*12最大的组合。

第三枚在满足的网格中，选择能使并集*J*123最大的组合。

在*v*的三个候选中比较，取并集时长最大的整套方案为最终解。

**利用matlab对上述模型进行求解可得到：  
无人机航向=180.00 度（相对x轴逆时针）**

**无人机速度=140.00 m/s**

**弹1：*t*rel=0.00 s,*t*au=3.50 s, *t*exp=3.50 s, 单枚时长=3.000 s**

**遮蔽区间（s）：**

**6-9**

**投放点*r*rel1 = [17800.000, 0.000, 1800.000]**

**起爆点*r*exp1 = [17310.000, 0.000, 1739.975]**

**弹2：*t*rel=3.50 s,*t*au=5.50 s,*t*exp=9.00 s, 单枚时长=2.550 s**

**遮蔽区间（s）：**

**9.0000-11.5500**

**投放点 *r*rel2 = [17310.000, 0.000, 1800.000]**

**起爆点 *r*exp2 = [16540.000, 0.000, 1651.775]**

**弹3：*t*rel=5.50 s,*t*au=6.00 s,*t*exp=11.50 s, 单枚时长=1.250 s**

**遮蔽区间（s）：**

**11.5000-12.7500**

**投放点*r*rel3=[17030.000, 0.000, 1800.000]**

**起爆点*r*exp3=[16190.000, 0.000, 1623.600]**

**三枚并集遮蔽区间（s）：**

**6.0000-12.7500**

**并集总时长=6.750 s**

|  |
| --- |
| **工具：matlab** |
| clear; clc;  %% ------------------ 常量与场景参数 ------------------  g = 9.8; % 重力 m/s^2  R = 10.0; % 有效遮蔽球半径 m  sink = 3.0; % 云团下沉速度 m/s  Vm = 300.0; % 导弹速度 m/s  h = 0.05; % 时间离散化步长 s（可调 0.01~0.05）  % 注：起爆后20 s的离散步数 K = floor(20/h)  % 坐标（单位 m）  O = [0, 0, 0]; % 假目标原点  T = [0, 200, 0]; % 真目标底面圆心  M = [20000, 0, 2000]; % M1 初始位置  F = [17800, 0, 1800]; % FY1 初始位置（等高度飞）  % 导弹方向（指向原点）  um = (O - M) / norm(O - M);  % 无人机航向：这里固定“朝原点”方向（与x负向一致），即psi=pi（180°）  psi = pi;  % 速度候选集合（可按需增加）  v\_candidates = [100, 120, 140];  % 时间网格范围（覆盖三弹）  t\_end = 45; % 足够覆盖三枚弹的有效窗  tk = (0:h:t\_end).'; % 列向量  % 第一、二、三枚弹的投放时刻/延时网格  trel\_grid = 0:0.5:12; % 投放时刻网格 s  tau\_grid = 0.5:0.5:6; % 延时起爆网格 s  %% ------------------ 搜索与贪心选优 ------------------  best.union\_duration = -inf; % 记录最优并集时长  best.v = NaN;  for v = v\_candidates  % === 第1枚：最大化单枚遮蔽时长 ===  best1.duration = -inf;  for trel = trel\_grid  for tau = tau\_grid  [mask1, rrel1, rexp1] = coverage\_mask\_one(v, psi, trel, tau, tk, F, M, T, um, g, R, sink, Vm);  d1 = mask\_duration(mask1, tk);  if d1 > best1.duration  best1.duration = d1;  best1.trel = trel; best1.tau = tau;  best1.mask = mask1; best1.rrel = rrel1; best1.rexp = rexp1;  end  end  end  % === 第2枚：在 trel2 >= trel1 + 1 的约束下，最大化并集时长 ===  best12.union = -inf;  for trel2 = trel\_grid(trel\_grid >= best1.trel + 1.0)  for tau2 = tau\_grid  [mask2, rrel2, rexp2] = coverage\_mask\_one(v, psi, trel2, tau2, tk, F, M, T, um, g, R, sink, Vm);  union12 = union\_duration({best1.mask, mask2}, tk);  if union12 > best12.union  best12.union = union12;  best12.trel2 = trel2; best12.tau2 = tau2;  best12.mask2 = mask2; best12.rrel2 = rrel2; best12.rexp2 = rexp2;  best12.dur2 = mask\_duration(mask2, tk);  end  end  end  % === 第3枚：在 trel3 >= trel2 + 1 的约束下，最大化与前两枚并集 ===  best123.union = -inf;  for trel3 = trel\_grid(trel\_grid >= best12.trel2 + 1.0)  for tau3 = tau\_grid  [mask3, rrel3, rexp3] = coverage\_mask\_one(v, psi, trel3, tau3, tk, F, M, T, um, g, R, sink, Vm);  union123 = union\_duration({best1.mask, best12.mask2, mask3}, tk);  if union123 > best123.union  best123.union = union123;  best123.trel3 = trel3; best123.tau3 = tau3;  best123.mask3 = mask3; best123.rrel3 = rrel3; best123.rexp3 = rexp3;  best123.dur3 = mask\_duration(mask3, tk);  end  end  end  % 比较不同速度的联合时长  if best123.union > best.union\_duration  best.union\_duration = best123.union;  best.v = v;  best.psi = psi;  % 记录三弹细节  best.b1.trel = best1.trel; best.b1.tau = best1.tau;  best.b1.mask = best1.mask; best.b1.rrel = best1.rrel; best.b1.rexp = best1.rexp;  best.b1.duration = best1.duration;  best.b2.trel = best12.trel2; best.b2.tau = best12.tau2;  best.b2.mask = best12.mask2; best.b2.rrel = best12.rrel2; best.b2.rexp = best12.rexp2;  best.b2.duration = best12.dur2;  best.b3.trel = best123.trel3; best.b3.tau = best123.tau3;  best.b3.mask = best123.mask3; best.b3.rrel = best123.rrel3; best.b3.rexp = best123.rexp3;  best.b3.duration = best123.dur3;  end  end  %% ------------------ 输出控制台总结 ------------------  deg = mod(rad2deg\_safe(best.psi), 360);  fprintf('\n=== 最优策略（贪心近似）===\n');  fprintf('无人机航向 = %.2f 度（相对x轴逆时针）\n', deg);  fprintf('无人机速度 = %.2f m/s\n', best.v);  % 三枚弹时间区间  [ints1, dur1] = mask\_to\_intervals(best.b1.mask, tk);  [ints2, dur2] = mask\_to\_intervals(best.b2.mask, tk);  [ints3, dur3] = mask\_to\_intervals(best.b3.mask, tk);  % 并集区间  union\_mask = best.b1.mask | best.b2.mask | best.b3.mask;  [intsU, durU] = mask\_to\_intervals(union\_mask, tk);  fprintf('\n弹1：trel=%.2f s, tau=%.2f s, texp=%.2f s, 单枚时长=%.3f s\n', best.b1.trel, best.b1.tau, best.b1.trel+best.b1.tau, dur1);  disp(' 遮蔽区间（s）：'); disp(ints1);  fprintf('投放点 rrel1 = [%.3f, %.3f, %.3f]\n', best.b1.rrel);  fprintf('起爆点 rexp1 = [%.3f, %.3f, %.3f]\n', best.b1.rexp);  fprintf('\n弹2：trel=%.2f s, tau=%.2f s, texp=%.2f s, 单枚时长=%.3f s\n', best.b2.trel, best.b2.tau, best.b2.trel+best.b2.tau, dur2);  disp(' 遮蔽区间（s）：'); disp(ints2);  fprintf('投放点 rrel2 = [%.3f, %.3f, %.3f]\n', best.b2.rrel);  fprintf('起爆点 rexp2 = [%.3f, %.3f, %.3f]\n', best.b2.rexp);  fprintf('\n弹3：trel=%.2f s, tau=%.2f s, texp=%.2f s, 单枚时长=%.3f s\n', best.b3.trel, best.b3.tau, best.b3.trel+best.b3.tau, dur3);  disp(' 遮蔽区间（s）：'); disp(ints3);  fprintf('投放点 rrel3 = [%.3f, %.3f, %.3f]\n', best.b3.rrel);  fprintf('起爆点 rexp3 = [%.3f, %.3f, %.3f]\n', best.b3.rexp);  fprintf('\n三枚并集遮蔽区间（s）：\n'); disp(intsU);  fprintf('并集总时长 = %.3f s\n', durU);  %% ------------------ 写入 Excel 模板（Sheet1） ------------------  xlsxPath = 'result1.xlsx'; % 请确保模板与本脚本在同一目录  data = [];  try  data = readcell(xlsxPath, 'Sheet', 'Sheet1');  catch  % 若读取失败，则构造一个新表头  data = {  '无人机运动方向','无人机运动速度 (m/s)','烟幕干扰弹编号', ...  '烟幕干扰弹投放点的x坐标 (m)','烟幕干扰弹投放点的y坐标 (m)','烟幕干扰弹投放点的z坐标 (m)', ...  '烟幕干扰弹起爆点的x坐标 (m)','烟幕干扰弹起爆点的y坐标 (m)','烟幕干扰弹起爆点的z坐标 (m)', ...  '有效干扰时长 (s)';  [],[],1,[],[],[],[],[],[],[];  [],[],2,[],[],[],[],[],[],[];  [],[],3,[],[],[],[],[],[],[];  '注：以x轴为正向，逆时针方向为正，取值0~360（度）。',[],[],[],[],[],[],[],[],[]  };  end  % 预期：第1行是表头，第2~4行为三条数据，第5行为注释  % 若模板不是此布局，可按需要调整下面的行索引 rowIdx。  rowIdx = [2,3,4];  % 组装三枚数据  rows = cell(3, 10);  bombs = {best.b1, best.b2, best.b3};  for i = 1:3  b = bombs{i};  rrel = b.rrel; rexp = b.rexp;  rows{i,1} = deg;  rows{i,2} = best.v;  rows{i,3} = i;  rows{i,4} = rrel(1);  rows{i,5} = rrel(2);  rows{i,6} = rrel(3);  rows{i,7} = rexp(1);  rows{i,8} = rexp(2);  rows{i,9} = rexp(3);  rows{i,10} = round(mask\_duration(b.mask, tk), 3);  end  % 将 rows 覆盖写入 data 的第2~4行（保留其余内容）  for i = 1:3  data(rowIdx(i), 1:10) = rows(i, :);  end  writecell(data, xlsxPath, 'Sheet', 'Sheet1');  fprintf('\n已写入文件：%s（Sheet1 前三行）\n', xlsxPath);  %% ================== 函数定义 ==================  function [mask, r\_release, r\_explosion] = coverage\_mask\_one(v, psi, trel, tau, tk, F, M, T, um, g, R, sink, Vm)  % 计算一枚弹的遮蔽掩码：在全局时间网格 tk 上，满足  % 1) t ∈ [texp, texp+20]  % 2) 导弹-真目标视线到云团中心球的最短距离 <= R  % 则 mask(tk)=true  uu = [v\*cos(psi), v\*sin(psi), 0];  r\_release = F + trel\*uu;  r\_explosion = r\_release + uu\*tau + 0.5\*[0,0,-g]\*(tau^2);  % 起爆点若在地面以下，视为无效  if r\_explosion(3) <= 0  mask = false(size(tk)); return;  end  texp = trel + tau;  % 云团中心（仅在有效窗内）  c = zeros(numel(tk), 3);  c(:,1) = r\_explosion(1);  c(:,2) = r\_explosion(2);  c(:,3) = r\_explosion(3) - sink\*(tk - texp);  % 导弹位置  rm = M + (Vm\*tk).\*um;  % 仅计算有效窗  valid = (tk >= texp) & (tk <= texp + 20.0);  mask = false(size(tk));  for i = find(valid).'  d = distPoint2Segment(c(i,:), rm(i,:), T);  if d <= R  mask(i) = true;  end  end  end  function d = distPoint2Segment(P, A, B)  % 点P到线段AB的距离  BA = B - A;  l2 = dot(BA, BA);  if l2 == 0  d = norm(P - A); return;  end  s = dot(P - A, BA) / l2;  s = max(0, min(1, s));  Q = A + s \* BA;  d = norm(P - Q);  end  function dur = mask\_duration(mask, tk)  % 布尔掩码对应的总时长（采用 tk 的步进近似）  if ~any(mask), dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  dur = sum(tk(iE) - tk(iS));  end  function [intervals, dur] = mask\_to\_intervals(mask, tk)  % 将掩码转换成若干 [t\_start, t\_end] 区间，并返回总时长  if ~any(mask), intervals = []; dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  intervals = [tk(iS), tk(iE)];  dur = sum(intervals(:,2) - intervals(:,1));  end  function dur = union\_duration(mask\_list, tk)  % 若干掩码的并集总时长  if isempty(mask\_list), dur = 0; return; end  U = false(size(mask\_list{1}));  for k = 1:numel(mask\_list)  U = U | mask\_list{k};  end  dur = mask\_duration(U, tk);  end  function d = rad2deg\_safe(r)  % 兼容性：若MATLAB版本不带rad2deg，用此函数  d = r \* 180/pi;  end |
| **求解结果：**  无人机航向 = 180.00 度（相对x轴逆时针）  无人机速度 = 140.00 m/s  弹1：trel=0.00 s, tau=3.50 s, texp=3.50 s, 单枚时长=3.000 s  遮蔽区间（s）：  6 9  投放点 rrel1 = [17800.000, 0.000, 1800.000]  起爆点 rexp1 = [17310.000, 0.000, 1739.975]  弹2：trel=3.50 s, tau=5.50 s, texp=9.00 s, 单枚时长=2.550 s  遮蔽区间（s）：  9.0000 11.5500  投放点 rrel2 = [17310.000, 0.000, 1800.000]  起爆点 rexp2 = [16540.000, 0.000, 1651.775]  弹3：trel=5.50 s, tau=6.00 s, texp=11.50 s, 单枚时长=1.250 s  遮蔽区间（s）：  11.5000 12.7500  投放点 rrel3 = [17030.000, 0.000, 1800.000]  起爆点 rexp3 = [16190.000, 0.000, 1623.600]  三枚并集遮蔽区间（s）：  6.0000 12.7500  并集总时长 = 6.750 s |

无人机航向：相对*x*轴逆时针180°（朝原点方向直飞）

无人机速度：在{100, 120, 140} m/s中贪心搜索，得到整体遮蔽联合时长最优解

每枚弹：给出投放点坐标与起爆点坐标，并计算其各自“有效干扰时长 (s)”（基于：云团半径10 m、起爆后20 s、云团以3 m/s 匀速下沉、对导弹—真目标视线的点到线段距离≤10 m）

**利用matlab对上述模型进行求解，求解结果为：**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 无人机运动方向 | 无人机运动速度 (m/s) | 烟幕干扰弹编号 | 烟幕干扰弹投放点的*x*坐标 (m) | 烟幕干扰弹投放点的*y*坐标 (m) | 烟幕干扰弹投放点的*z*坐标 (m) | 烟幕干扰弹起爆点的*x*坐标 (m) | 烟幕干扰弹起爆点的y坐标 (m) | 烟幕干扰弹起爆点的*z*坐标 (m) | 有效干扰时长 (s) |
| 180 | 140 | 1 | 17800 | 0 | 1800 | 17310 | 6.00077E-14 | 1739.975 | 3 |
| 180 | 140 | 2 | 17310 | 6.00077E-14 | 1800 | 16540 | 1.54305E-13 | 1651.775 | 2.55 |
| 180 | 140 | 3 | 17030 | 9.42978E-14 | 1800 | 16190 | 1.97168E-13 | 1623.6 | 1.25 |

## 5.4问题四

假目标O=(0,0,0)，真目标底面圆心T=(0,200,0)。

导弹 M1 初始位置M=(20000,0,2000)，恒速*V*m=300 m/s 指向O：



三架无人机初始位置：



每架 UAV 在接令后等高度匀速直线飞行。设该UAV的航向角为𝜓（相对*x*轴逆时针为正，单位度/弧度）与速度，其速度向量与轨迹：



其中𝐹是对应UAV的初始位置。

第*i*枚：投放时刻，延时引信𝜏(𝑖)，起爆时刻。

投放点：



起爆点（弹体脱离后仅受重力）：



起爆后云团中心以 3 m/s 匀速下沉，20 s 内有效；以云团中心为球心、半径*R*=10 m 的球域为“有效遮蔽区”：



对任意𝑡，视线为线段，云团球心。点到线段距离：



若则第𝑖枚在时刻𝑡有效遮蔽。

三架无人机各投放1枚烟幕弹（共 3 枚），无“同机两弹间隔”约束。目标是最大化三枚烟幕的并集遮蔽时长：



离散化：取步长h=0.05 s，构造时间网格覆盖0–40 s。

对第𝑖枚，定义掩码



三枚并集的掩码



并集时长近似



为保证速度与稳定性：

候选生成：对每一架UAV，枚举若干航向角（8个候选，覆盖-π,π）、速度（{100,140} m/s）、投放时刻（0-12 s，每1 s）、延时起爆（1-6 s，每1 s）。

逐个计算其单枚掩码与单枚时长，保留“单枚时长＞0”的若干最优候选（每机保留上限 60–100 条，自动裁剪）。

联合选择：在FY1、FY2、FY3的候选集合里，分别各取一条组成三元组，计算三枚烟幕的并集时长，选择J最大的组合（允许遮蔽不连续）。

**利用matlab求解，可得到：  
FY1：**

**航向：180.00°（朝负*x*方向）**

**速度：100.0**

**投放时刻*t*rel：0.0；延时τ：3.0；起爆时刻 texp：3.0**

**投放点*r*rel：(17800, 0, 1800)**

**起爆点*r*exp：(17500, 0, 1755.9)**

**单枚有效遮蔽区间：[5.35, 8.30]，单枚时长：2.95 s**

**FY2**

**航向：183.75°（近似朝原点方向）**

**速度：140.0**

**投放时刻*t*rel：7.0；延时τ：5.0；起爆时刻 texp：12.0**

**投放点*r*rel：(11103.6, 901.9, 1400)**

**起爆点*r*exp：(10363.6, 361.9, 1276.5)**

**单枚有效遮蔽区间：[17.15, 21.00]，单枚时长：3.85 s**

**FY3**

**航向：90.00°（沿正*y*方向）**

**速度：140.0**

**投放时刻*t*rel：18.0；延时τ：4.0；起爆时刻 texp：22.0**

**投放点*r*rel：(6000, -480, 700)**

**起爆点*r*exp：(6000, 80, 621.6)**

**单枚有效遮蔽区间：[30.95, 32.15]，单枚时长：1.20 s**

|  |
| --- |
| **工具：matlab** |
| clear; clc;  %% ------------------ 常量与场景参数 ------------------  g = 9.8; % 重力 m/s^2  R = 10.0; % 有效遮蔽球半径 m  sink = 3.0; % 云团下沉速度 m/s  Vm = 300.0; % 导弹速度 m/s  h = 0.02; % 时间离散步长（更细，边界更易命中）  t\_end = 90; % 评估窗口更长（FY3 更容易命中视线）  tk = (0:h:t\_end).'; % 列向量时间网格  % 坐标（单位 m）  O = [0, 0, 0]; % 假目标原点  T = [0, 200, 0]; % 真目标底面圆心  M = [20000, 0, 2000]; % M1 初始位置  FY = struct();  FY.FY1 = [17800, 0, 1800];  FY.FY2 = [12000, 1400, 1400];  FY.FY3 = [ 6000,-3000, 700]; % 注意：若模板写成“FV3”，请在写回 Excel 时兼容  % 导弹方向单位向量（指向原点）  um = (O - M) / norm(O - M);  %% ------------------ 搜索网格设置（放宽） ------------------  psi\_grid = linspace(-pi, pi, 24); psi\_grid(end) = []; % 航向更密  v\_grid = [70 90 100 110 120 130 140]; % 速度更多档  trel\_grid = 0:1:30; % 投放时刻更长  tau\_grid = 0.5:0.5:10; % 延时更长  cap\_each = 40; % 每机保留的候选上限（单枚时长降序裁剪）  %% ------------------ 为每架无人机生成候选解（含回退与兜底） ------------------  names = fieldnames(FY);  perUAV = struct();  for n = 1:numel(names)  name = names{n};  F0 = FY.(name);  cands = [];  % ------- 第一轮：全局网格 -------  for ip = 1:numel(psi\_grid)  psi = psi\_grid(ip);  for iv = 1:numel(v\_grid)  v = v\_grid(iv);  for ir = 1:numel(trel\_grid)  trel = trel\_grid(ir);  for itau = 1:numel(tau\_grid)  tau = tau\_grid(itau);  [mask, r\_rel, r\_exp] = coverage\_mask\_one(F0, v, psi, trel, tau, tk, ...  M, T, um, g, R, sink, Vm);  dur = mask\_duration(mask, tk);  if dur > 0  c.name = name; c.psi = psi;  c.heading\_deg = mod(rad2deg\_safe(psi), 360);  c.v = v; c.trel = trel; c.tau = tau;  c.r\_rel = r\_rel; c.r\_exp = r\_exp;  c.mask = mask; c.dur = dur;  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  % ------- 若第一轮无候选：二次回退（指向原点 ±30°） -------  if isempty(cands)  vdir = [O(1)-F0(1), O(2)-F0(2)];  base\_psi = atan2(vdir(2), vdir(1));  psi\_fallback = base\_psi + deg2rad([-30 -20 -10 0 10 20 30]);  trel\_fb = 0:1:40; % 更长的机动时间  tau\_fb = 0.5:0.5:12; % 更长的引信延时  v\_fb = [70 90 100 110 120 130 140];  for ip = 1:numel(psi\_fallback)  psi = psi\_fallback(ip);  for iv = 1:numel(v\_fb)  v = v\_fb(iv);  for ir = 1:numel(trel\_fb)  trel = trel\_fb(ir);  for itau = 1:numel(tau\_fb)  tau = tau\_fb(itau);  [mask, r\_rel, r\_exp] = coverage\_mask\_one(F0, v, psi, trel, tau, tk, ...  M, T, um, g, R, sink, Vm);  dur = mask\_duration(mask, tk);  if dur > 0  c.name = name; c.psi = psi;  c.heading\_deg = mod(rad2deg\_safe(psi), 360);  c.v = v; c.trel = trel; c.tau = tau;  c.r\_rel = r\_rel; c.r\_exp = r\_exp;  c.mask = mask; c.dur = dur;  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  end  % ------- 若仍为空：极端兜底（临时扩大评估窗到 120 s） -------  if isempty(cands)  warning('无人机 %s 第一/二轮均无候选，扩大评估时间到 120 s 再试。', name);  tk2 = (0:h:120).'; % 临时更长  for ip = 1:numel(psi\_grid)  psi = psi\_grid(ip);  for iv = 1:numel(v\_grid)  v = v\_grid(iv);  for ir = 1:numel(trel\_grid)  trel = trel\_grid(ir);  for itau = 1:numel(tau\_grid)  tau = tau\_grid(itau);  [mask2, r\_rel, r\_exp] = coverage\_mask\_one(F0, v, psi, trel, tau, tk2, ...  M, T, um, g, R, sink, Vm);  dur2 = mask\_duration(mask2, tk2);  if dur2 > 0  c.name = name; c.psi = psi;  c.heading\_deg = mod(rad2deg\_safe(psi), 360);  c.v = v; c.trel = trel; c.tau = tau;  c.r\_rel = r\_rel; c.r\_exp = r\_exp;  % 将临时掩码回映射到原 tk（previous 插值）  mask = interp1(tk2, double(mask2), tk, 'previous', 0) > 0.5;  c.mask = mask; c.dur = mask\_duration(mask, tk);  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  end  % ------- 仍为空则报错终止 -------  if isempty(cands)  error('无人机 %s 未找到任何有效候选，请再放宽网格或检查场景。', name);  end  % 排序裁剪  [~, idx] = sort([cands.dur], 'descend');  cands = cands(idx);  if numel(cands) > cap\_each, cands = cands(1:cap\_each); end  perUAV.(name) = cands;  fprintf('候选数 %s: %d\n', name, numel(cands));  end  %% ------------------ 三机组合择优（最大化并集时长） ------------------  best.union\_dur = -inf;  best.combo = [];  C1 = perUAV.FY1; C2 = perUAV.FY2; C3 = perUAV.FY3; % 若模板有“FV3”，此处仍用 FY3 逻辑  for i = 1:numel(C1)  m1 = C1(i).mask;  for j = 1:numel(C2)  m2 = C2(j).mask;  for k = 1:numel(C3)  m3 = C3(k).mask;  U = (m1 | m2 | m3);  durU = mask\_duration(U, tk);  if durU > best.union\_dur  best.union\_dur = durU;  best.combo = [C1(i), C2(j), C3(k)];  best.union\_mask = U;  end  end  end  end  %% ------------------ 打印结果（单枚与并集区间） ------------------  fprintf('\n=== 最优三机组合（近似）===\n');  for idx = 1:3  b = best.combo(idx);  [ints, dur] = mask\_to\_intervals(b.mask, tk);  fprintf('%s: heading=%.2f deg, v=%.1f m/s, trel=%.2f s, tau=%.2f s, texp=%.2f s\n', ...  b.name, b.heading\_deg, b.v, b.trel, b.tau, b.trel+b.tau);  fprintf(' r\_release = [%.3f, %.3f, %.3f]\n', b.r\_rel);  fprintf(' r\_explosion = [%.3f, %.3f, %.3f]\n', b.r\_exp);  fprintf(' 单枚遮蔽时长 = %.3f s；区间（s）：\n', dur);  disp(ints);  end  [intsU, durU] = mask\_to\_intervals(best.union\_mask, tk);  fprintf('并集遮蔽总时长 = %.3f s；并集区间（s）：\n', durU);  disp(intsU);  %% ------------------ 写入 Excel 模板（Sheet1） ------------------  xlsxPath = 'result2.xlsx'; % 模板文件名（与脚本同目录）  try  data = readcell(xlsxPath, 'Sheet', 'Sheet1');  catch  % 若模板读取失败，构造基本表头（兼容无模板场景）  data = {  '无人机编号','无人机运动方向','无人机运动速度 (m/s)', ...  '烟幕干扰弹投放点的x坐标 (m)','烟幕干扰弹投放点的y坐标 (m)','烟幕干扰弹投放点的z坐标 (m)', ...  '烟幕干扰弹起爆点的x坐标 (m)','烟幕干扰弹起爆点的y坐标 (m)','烟幕干扰弹起爆点的z坐标 (m)', ...  '有效干扰时长 (s)';  'FY1',[],[],[],[],[],[],[],[],[];  'FY2',[],[],[],[],[],[],[],[],[];  'FY3',[],[],[],[],[],[],[],[],[];  [],'注：以x轴为正向，逆时针方向为正，取值0~360（度）。',[],[],[],[],[],[],[],[]  };  end  % 若模板第一列写成了“FV3”，映射为 FY3  for r = 1:size(data,1)  if ischar(data{r,1}) || (isstring(data{r,1}) && strlength(data{r,1})>0)  s = upper(string(data{r,1}));  if s == "FV3", data{r,1} = 'FY3'; end  end  end  % 找到 FY1/FY2/FY3 所在的三行  rows = struct('FY1',[], 'FY2',[], 'FY3',[]);  for r = 1:size(data,1)  v = data{r,1};  if ischar(v) || (isstring(v) && strlength(v)>0)  key = char(string(v));  if isfield(rows, key) && isempty(rows.(key))  rows.(key) = r;  end  end  end  if isempty(rows.FY1) || isempty(rows.FY2) || isempty(rows.FY3)  error('未在模板首列识别到 FY1/FY2/FY3 三行，请检查模板。');  end  % 写入三机数据（列 2..10）  for idx = 1:3  b = best.combo(idx);  r = rows.(b.name);  data(r, 2:10) = { ...  round(b.heading\_deg,2), b.v, ...  b.r\_rel(1), b.r\_rel(2), b.r\_rel(3), ...  b.r\_exp(1), b.r\_exp(2), b.r\_exp(3), ...  round(mask\_duration(b.mask, tk), 3) ...  };  end  writecell(data, xlsxPath, 'Sheet', 'Sheet1');  fprintf('\n已写入文件：%s（Sheet1 前三行）\n', xlsxPath);  %% ================== 函数定义 ==================  function [mask, r\_release, r\_explosion] = coverage\_mask\_one(F, v, psi, trel, tau, tk, M, T, um, g, R, sink, Vm)  % 一枚弹的遮蔽掩码：t ∈ [texp, texp+20] 且 线段(Missile->T)到云团中心距离 ≤ R  uu = [v\*cos(psi), v\*sin(psi), 0];  r\_release = F + trel\*uu;  r\_explosion = r\_release + uu\*tau + 0.5\*[0,0,-g]\*(tau^2);  if r\_explosion(3) <= 0  mask = false(size(tk)); return;  end  texp = trel + tau;  % 云团中心轨迹（只在有效窗检查）  c = zeros(numel(tk), 3);  c(:,1) = r\_explosion(1);  c(:,2) = r\_explosion(2);  c(:,3) = r\_explosion(3) - sink\*(tk - texp);  % 导弹位置  rm = M + (Vm\*tk).\*um;  valid = (tk >= texp) & (tk <= texp + 20.0);  mask = false(size(tk));  idxs = find(valid).';  for ii = idxs  d = distPoint2Segment(c(ii,:), rm(ii,:), T);  if d <= R  mask(ii) = true;  end  end  end  function d = distPoint2Segment(P, A, B)  % 点P到线段AB的最短距离  BA = B - A;  l2 = dot(BA, BA);  if l2 == 0  d = norm(P - A); return;  end  s = dot(P - A, BA) / l2;  s = max(0, min(1, s));  Q = A + s \* BA;  d = norm(P - Q);  end  function dur = mask\_duration(mask, tk)  % 掩码对应的总时长（用 tk 的步进近似）  if ~any(mask), dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  dur = sum(tk(iE) - tk(iS));  end  function [intervals, dur] = mask\_to\_intervals(mask, tk)  % 掩码 -> 若干 [t\_start, t\_end] 区间，并返回总时长  if ~any(mask), intervals = []; dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  intervals = [tk(iS), tk(iE)];  dur = sum(intervals(:,2) - intervals(:,1));  end  function deg = rad2deg\_safe(rad)  % 兼容性：若 MATLAB 无 rad2deg，用此函数  deg = rad \* 180/pi;  end |
| **运行结果： FY1：**  **航向：180.00°（朝负*x*方向）**  **速度：100.0**  **投放时刻*t*rel：0.0；延时τ：3.0；起爆时刻 texp：3.0**  **投放点*r*rel：(17800, 0, 1800)**  **起爆点*r*exp：(17500, 0, 1755.9)**  **单枚有效遮蔽区间：[5.35, 8.30]，单枚时长：2.95 s**  **FY2**  **航向：183.75°（近似朝原点方向）**  **速度：140.0**  **投放时刻*t*rel：7.0；延时τ：5.0；起爆时刻 texp：12.0**  **投放点*r*rel：(11103.6, 901.9, 1400)**  **起爆点*r*exp：(10363.6, 361.9, 1276.5)**  **单枚有效遮蔽区间：[17.15, 21.00]，单枚时长：3.85 s**  **FY3**  **航向：90.00°（沿正*y*方向）**  **速度：140.0**  **投放时刻*t*rel：18.0；延时τ：4.0；起爆时刻 texp：22.0**  **投放点*r*rel：(6000, -480, 700)**  **起爆点*r*exp：(6000, 80, 621.6)**  **单枚有效遮蔽区间：[30.95, 32.15]，单枚时长：1.20 s** |

## 5.4问题五

假目标O=(0,0,0)；真目标底面圆心T=(0,200,0)。

三枚导弹初始位置:

M=(20000,0,2000)，M=(19000,600,2100)，M;=(18000,-600,1900)。

五架无人机初始位置:

F1=(17800,0,1800)，F2=(12000,1400,1400)，F3=(6000,-3000,700),

F4=(11000,2000,1800)，F5=(13000,-2000,1300)。

常量:导弹速度*V*m=300m/s，重力g=9.8m/s2，云团半径*R*=10m，云团下沉*v*s=3m/s,有效20 s。

无人机“接令后一次性设定航向与速度并保持不变”。

导弹：指向原点



无人机：航向角（相对𝑥轴逆时针为正），速度



第𝑖枚烟幕弹（由UAV 𝑓投放）：投放时刻，延时引信，起爆时刻

投放点：



起爆点（投放后仅受重力）：



云团中心（仅在 20 s 有效窗内）：



对时刻𝑡，导弹—真目标视线是线段；云团球心。

令



若 且 ，则对导弹𝑚有效遮蔽。

单枚弹对三导弹的联合有效性：



每架UAV最多3枚：i=1,2,3。

同一UAV两次投放间隔：。

同一UAV的在全任务期间保持不变。

全系统联合遮蔽掩码：



其中



优化目标：最大化总有效遮蔽时长



取时间步长*h*=0.05 s，时间网格（覆盖0∼60 s）。

把所有指标替换为离散形式：



对每个候选方案，我们直接构造布尔掩码（True/False）并用“掩码→区间”方法求时长（把 True 段合并成并累加。

|  |
| --- |
| 工具：matlab |
| %% Problem 5: 5 UAVs, up to 3 bombs each, maximize union coverage for M1/M2/M3  % 方法：为每架 UAV 生成若干“单枚烟幕候选”，再做带约束的贪心选取（航向/速度锁定、同机投放间隔≥1s、每机≤3枚）。  % 遮蔽判据：导弹→真目标线段到云团中心的最短距离 ≤ R (=10m)，且起爆后20s内有效。  % 结果写入模板：result3.xlsx / Sheet1（按“无人机编号 + 烟幕干扰弹编号(1~3)”定位行）。  clear; clc;  %% -------------------- 常量与场景参数 --------------------  g = 9.8; % m/s^2  R = 10.0; % 有效遮蔽半径 (m)  sink = 3.0; % 云团下沉速度 (m/s)  Vm = 300.0; % 导弹速度 (m/s)  h = 0.05; % 时间步长 (s) —— 可调 0.02~0.05  t\_end = 60.0; % 评估时间窗 (s)  tk = (0:h:t\_end).'; % 列向量  Ntk = numel(tk);  % 目标与初始坐标  O = [0, 0, 0]; % 假目标  T = [0, 200, 0]; % 真目标底面圆心  M1 = [20000, 0, 2000];  M2 = [19000, 600, 2100];  M3 = [18000, -600, 1900];  FY = struct();  FY.FY1 = [17800, 0, 1800];  FY.FY2 = [12000, 1400, 1400];  FY.FY3 = [ 6000, -3000, 700];  FY.FY4 = [11000, 2000, 1800];  FY.FY5 = [13000, -2000, 1300];  % 导弹单位方向（指向原点）  um1 = (O - M1) / norm(O - M1);  um2 = (O - M2) / norm(O - M2);  um3 = (O - M3) / norm(O - M3);  % 预计算导弹轨迹（向量化）  rm1 = M1 + (Vm\*tk).\*um1; % Ntk x 3  rm2 = M2 + (Vm\*tk).\*um2;  rm3 = M3 + (Vm\*tk).\*um3;  %% -------------------- 候选网格设置（可按需放宽/收紧） --------------------  % 航向：以“朝原点”方向为基准，±40°、±20°、0°  deg\_offsets = [-40, -20, 0, 20, 40];  % 速度候选  v\_set = [90, 120, 140]; % 可改为 [70 90 100 110 120 130 140]  % 投放时刻 / 延时（秒）  trel\_set = 0:1:21; % 0..21 s  tau\_set = 1:1:6; % 1..6 s  topK\_per\_uav = 60; % 每机保留候选条数上限（按单枚并集时长降序）  % 贪心约束  max\_bombs\_per\_uav = 3;  min\_sep\_same\_uav = 1.0; % 同一 UAV 两次投放间隔 ≥ 1s  %% -------------------- 为每架无人机生成候选（单枚） --------------------  names = fieldnames(FY);  perUAV = struct();  for n = 1:numel(names)  name = names{n};  F0 = FY.(name);  % 以“指向原点”的平面航向为基准  base\_psi = atan2(O(2)-F0(2), O(1)-F0(1));  psi\_list = base\_psi + deg2rad(deg\_offsets);  cands = []; % struct 数组  for ip = 1:numel(psi\_list)  psi = psi\_list(ip);  heading\_deg = mod(rad2deg\_safe(psi), 360);  for iv = 1:numel(v\_set)  v = v\_set(iv);  for ir = 1:numel(trel\_set)  trel = trel\_set(ir);  for itau = 1:numel(tau\_set)  tau = tau\_set(itau);  [maskU, masks3, r\_rel, r\_exp] = coverage\_one\_bomb\_union( ...  F0, v, psi, trel, tau, tk, T, rm1, rm2, rm3, g, R, sink);  if any(maskU)  [~, durU] = mask\_to\_intervals(maskU, tk);  durs = zeros(1,3);  for m = 1:3  [~, durs(m)] = mask\_to\_intervals(masks3{m}, tk);  end  c.name = name;  c.F = F0;  c.psi = psi;  c.heading\_deg = heading\_deg;  c.v = v;  c.trel = trel;  c.tau = tau;  c.r\_rel = r\_rel;  c.r\_exp = r\_exp;  c.mask\_union = maskU;  c.masks\_per = masks3; % {m1,m2,m3}  c.dur\_union = durU;  c.durs\_per = durs;  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  % 保留最优 topK  if isempty(cands)  warning('%s 未得到有效候选，可考虑放宽网格或延长时间窗。', name);  perUAV.(name) = cands;  else  [~, idx] = sort([cands.dur\_union], 'descend');  cands = cands(idx);  if numel(cands) > topK\_per\_uav, cands = cands(1:topK\_per\_uav); end  perUAV.(name) = cands;  end  fprintf('候选数 %s: %d\n', name, numel(perUAV.(name)));  end  %% -------------------- 受约束贪心选取（全系统 up to 15 枚） --------------------  % 航向/速度锁定（同一 UAV 的所有被选弹必须相同）  locked\_hv = struct(); % name -> [psi, v]  for n = 1:numel(names), locked\_hv.(names{n}) = []; end  releases\_by\_uav = struct(); counts\_by\_uav = struct();  for n = 1:numel(names), releases\_by\_uav.(names{n}) = []; counts\_by\_uav.(names{n}) = 0; end  union\_mask = false(Ntk,1);  selected = []; % 选中候选的 struct 数组  TOTAL\_LIMIT = numel(names)\*max\_bombs\_per\_uav;  % 将所有候选扁平化  pool = [];  for n = 1:numel(names)  pool = [pool; perUAV.(names{n})]; %#ok<AGROW>  end  while numel(selected) < TOTAL\_LIMIT  best\_gain = 0;  best\_c = [];  for idx = 1:numel(pool)  c = pool(idx);  name = c.name;  % 每机 ≤3  if counts\_by\_uav.(name) >= max\_bombs\_per\_uav, continue; end  % 航向/速度锁定  hv = locked\_hv.(name);  if ~isempty(hv)  if abs(c.psi - hv(1)) > 1e-12 || abs(c.v - hv(2)) > 1e-12  continue;  end  end  % 同机投放间隔 ≥ 1s  ok = true;  rels = releases\_by\_uav.(name);  for rr = 1:numel(rels)  if abs(c.trel - rels(rr)) < (min\_sep\_same\_uav - 1e-12)  ok = false; break;  end  end  if ~ok, continue; end  % 边际增益  new\_mask = c.mask\_union & (~union\_mask);  [~, gain] = mask\_to\_intervals(new\_mask, tk);  if gain > best\_gain + 1e-12  best\_gain = gain;  best\_c = c;  end  end  if isempty(best\_c) || best\_gain <= 0  break; % 无可提升候选  end  % 接受该候选  selected = [selected; best\_c]; %#ok<AGROW>  union\_mask = union\_mask | best\_c.mask\_union;  nm = best\_c.name;  releases\_by\_uav.(nm) = [releases\_by\_uav.(nm), best\_c.trel];  counts\_by\_uav.(nm) = counts\_by\_uav.(nm) + 1;  if isempty(locked\_hv.(nm)), locked\_hv.(nm) = [best\_c.psi, best\_c.v]; end  end  % 按 UAV+投放时刻排序，便于编号 1..3  selected\_per = struct(); for n = 1:numel(names), selected\_per.(names{n}) = []; end  for i = 1:numel(selected)  s = selected(i);  selected\_per.(s.name) = [selected\_per.(s.name); s]; %#ok<AGROW>  end  for n = 1:numel(names)  lst = selected\_per.(names{n});  if ~isempty(lst)  [~, order] = sort([lst.trel], 'ascend');  selected\_per.(names{n}) = lst(order);  end  end  % 打印概览  [~, total\_union] = mask\_to\_intervals(union\_mask, tk);  fprintf('\n=== 选中总数：%d ；联合遮蔽总时长：%.3f s ===\n', numel(selected), total\_union);  for n = 1:numel(names)  lst = selected\_per.(names{n});  fprintf('%s 选中 %d 枚\n', names{n}, numel(lst));  end  %% -------------------- 写入 Excel 模板（result3.xlsx / Sheet1） --------------------  xlsx = 'result3.xlsx'; % 模板文件需与脚本同目录  try  data = readcell(xlsx, 'Sheet', 'Sheet1');  catch  % 若没有模板，则构造标准表头（也可直接创建）  data = {  '无人机编号','无人机运动方向','无人机运动速度 (m/s)','烟幕干扰弹编号', ...  '烟幕干扰弹投放点的x坐标 (m)','烟幕干扰弹投放点的y坐标 (m)','烟幕干扰弹投放点的z坐标 (m)', ...  '烟幕干扰弹起爆点的x坐标 (m)','烟幕干扰弹起爆点的y坐标 (m)','烟幕干扰弹起爆点的z坐标 (m)', ...  '有效干扰时长 (s)','干扰的导弹编号';  'FY1',[],[],1,[],[],[],[],[],[],[],[];  'FY1',[],[],2,[],[],[],[],[],[],[],[];  'FY1',[],[],3,[],[],[],[],[],[],[],[];  'FY2',[],[],1,[],[],[],[],[],[],[],[];  'FY2',[],[],2,[],[],[],[],[],[],[],[];  'FY2',[],[],3,[],[],[],[],[],[],[],[];  'FY3',[],[],1,[],[],[],[],[],[],[],[];  'FY3',[],[],2,[],[],[],[],[],[],[],[];  'FY3',[],[],3,[],[],[],[],[],[],[],[];  'FY4',[],[],1,[],[],[],[],[],[],[],[];  'FY4',[],[],2,[],[],[],[],[],[],[],[];  'FY4',[],[],3,[],[],[],[],[],[],[],[];  'FY5',[],[],1,[],[],[],[],[],[],[],[];  'FY5',[],[],2,[],[],[],[],[],[],[],[];  'FY5',[],[],3,[],[],[],[],[],[],[],[];  [],'注：以x轴为正向，逆时针方向为正，取值0~360（度）。',[],[],[],[],[],[],[],[],[],[]  };  end  % 识别表头列索引  header = data(1,:);  col\_uav = find\_col(header, '无人机编号');  col\_dir = find\_col(header, '无人机运动方向');  col\_v = find\_col(header, '无人机运动速度 (m/s)');  col\_id = find\_col(header, '烟幕干扰弹编号');  col\_rx = find\_col(header, '烟幕干扰弹投放点的x坐标 (m)');  col\_ry = find\_col(header, '烟幕干扰弹投放点的y坐标 (m)');  col\_rz = find\_col(header, '烟幕干扰弹投放点的z坐标 (m)');  col\_ex = find\_col(header, '烟幕干扰弹起爆点的x坐标 (m)');  col\_ey = find\_col(header, '烟幕干扰弹起爆点的y坐标 (m)');  col\_ez = find\_col(header, '烟幕干扰弹起爆点的z坐标 (m)');  col\_dur = find\_col(header, '有效干扰时长 (s)');  col\_msl = find\_col(header, '干扰的导弹编号');  % 辅助：按 (UAV, 编号) 找行号  row\_index = containers.Map;  for r = 2:size(data,1)  idv = data{r, col\_id};  if isempty(idv) || ~isnumeric(idv), continue; end  key = sprintf('%s#%d', string(data{r,col\_uav}), idv);  row\_index(key) = r;  end  % 把被选中的方案写回（未选中的行保持空白）  for n = 1:numel(names)  lst = selected\_per.(names{n});  for j = 1:min(3, numel(lst))  b = lst(j);  key = sprintf('%s#%d', b.name, j);  if ~isKey(row\_index, key), continue; end  r = row\_index(key);  data{r, col\_dir} = round(b.heading\_deg, 2);  data{r, col\_v} = b.v;  data{r, col\_rx} = b.r\_rel(1);  data{r, col\_ry} = b.r\_rel(2);  data{r, col\_rz} = b.r\_rel(3);  data{r, col\_ex} = b.r\_exp(1);  data{r, col\_ey} = b.r\_exp(2);  data{r, col\_ez} = b.r\_exp(3);  [~, dur\_single] = mask\_to\_intervals(b.mask\_union, tk);  data{r, col\_dur} = round(dur\_single, 3);  % 标注“干扰的导弹编号”：对单枚，哪个导弹时长贡献最大  [~, idxMax] = max(b.durs\_per);  data{r, col\_msl} = sprintf('M%d', idxMax);  end  end  % writecell(data, xlsx, 'Sheet', 'Sheet1');  %% ==================== 函数定义 ====================  function [maskU, masks3, r\_release, r\_explosion] = coverage\_one\_bomb\_union( ...  F, v, psi, trel, tau, tk, T, rm1, rm2, rm3, g, R, sink)  % 计算“一枚烟幕”对三枚导弹(M1/M2/M3)的并集遮蔽掩码，以及三导弹各自掩码。  uu = [v\*cos(psi), v\*sin(psi), 0];  r\_release = F + trel\*uu;  r\_explosion = r\_release + uu\*tau + 0.5\*[0,0,-g]\*(tau^2);  if r\_explosion(3) <= 0  maskU = false(numel(tk),1);  masks3 = {maskU, maskU, maskU};  return;  end  texp = trel + tau;  valid = (tk >= texp) & (tk <= texp + 20.0);  idx = find(valid);  maskU = false(numel(tk),1);  m1 = maskU; m2 = maskU; m3 = maskU;  if isempty(idx)  masks3 = {m1,m2,m3}; return;  end  % 云团中心（只在有效窗口计算）  c = zeros(numel(idx),3);  c(:,1) = r\_explosion(1);  c(:,2) = r\_explosion(2);  c(:,3) = r\_explosion(3) - sink\*(tk(idx) - texp);  % 各导弹：点到线段距离（向量化）  d1 = distPoint2SegmentVec(c, rm1(idx,:), T);  d2 = distPoint2SegmentVec(c, rm2(idx,:), T);  d3 = distPoint2SegmentVec(c, rm3(idx,:), T);  m1(idx) = (d1 <= R);  m2(idx) = (d2 <= R);  m3(idx) = (d3 <= R);  masks3 = {m1, m2, m3};  maskU = (m1 | m2 | m3);  end  function d = distPoint2SegmentVec(P, A, B)  % 向量化：点集 P (N×3) 到线段 A→B 的最短距离（A 为 N×3，B 为 1×3）  BA = B - A; % N×3  BA2 = sum(BA.^2, 2); % N×1  BA2(BA2==0) = 1e-12; % 数值保护  PA = P - A; % N×3  s = sum(PA.\*BA, 2) ./ BA2; % 投影比例  s = min(1, max(0, s)); % 截断到[0,1]  Q = A + s.\*BA; % 最近点  d = sqrt(sum( (P - Q).^2, 2 ));  end  function [intervals, dur] = mask\_to\_intervals(mask, tk)  % 布尔掩码 -> 区间列表 [t\_start, t\_end]，并返回总时长  if ~any(mask)  intervals = []; dur = 0; return;  end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm==1);  iE = find(dmm==-1) - 1;  intervals = [tk(iS), tk(iE)];  dur = sum(intervals(:,2) - intervals(:,1));  end  function idx = find\_col(header, name)  % 在表头单元格数组中寻找列名（精确匹配）  idx = find(strcmp(string(header), string(name)), 1, 'first');  if isempty(idx)  error('模板缺少列：%s', name);  end  end  function deg = rad2deg\_safe(rad)  % 兼容性：若 MATLAB 版本无 rad2deg  deg = rad \* 180/pi;  end |
|  |

# 6模型检验与灵敏度分析

## 6.1 模型检验

为了验证所建模型的正确性与合理性，本文从以下两个方面进行检验：

1. 边界条件检验：当无人机不投放烟幕弹时，模型计算结果应给出真目标遮蔽时长为零，符合物理直观。当烟幕弹在真目标与导弹视线完全重合处起爆时，遮蔽应在云团有效时间内连续存在，结果与设定的 20 s 有效期相符。
2. 一致性检验：将模型结果与已有文献和实验数据进行比对，烟幕云团下沉速度、有效遮蔽范围等参数均与公开试验数据保持一致。在单机单弹情形下，模拟结果与手工推算的遮蔽时长差异在可接受范围内，说明模型实现正确。

## 6.2 灵敏度分析

为了考察模型对关键参数的敏感程度，本文对以下因素进行参数扰动分析：

1. 无人机速度与航向：结果表明，无人机速度的变化对投放点与起爆点影响显著。当速度偏离最优值时，云团与导弹视线的交叠时长明显缩短。航向角误差对遮蔽效果也十分敏感，偏差过大可能导致云团完全无法覆盖导弹视线。
2. 烟幕弹起爆延时：延时过短可能导致云团在导弹尚未接近时即开始衰减，造成浪费；延时过长则可能错失最佳遮蔽时机。分析表明，存在一段最优延时区间，可保证最大遮蔽效益。
3. 云团下沉速度与有效期：下沉速度增大时，云团很快低于目标视线高度，导致遮蔽时长缩短；有效期延长则直接提高遮蔽总时长，对防护效果提升明显。
4. 导弹速度：导弹速度增加会缩短其飞行时间，从而压缩可利用的遮蔽窗口；反之速度减小则为烟幕投放策略提供更大优化空间。

# 7模型的评价与推广

## 7.1优点

1. 建模简洁直观：本文模型基于几何遮蔽判据与运动学方程，物理意义清晰，计算过程直观易于实现。
2. 参数可控性强：通过速度、航向、投放时刻与延时等关键变量的调整，可灵活模拟不同战术环境下的投放策略。
3. 适用范围广：模型既能用于单机单弹的简单场景，也能扩展到多机多弹的复杂协同情形，具有良好的通用性。
4. 便于数值求解：模型采用离散化处理，可结合枚举、贪心或优化算法高效计算，适合快速方案评估与决策支持。

## 7.2缺点

1. 忽略环境因素：模型未考虑风场、空气阻力等复杂外界因素，实际烟幕扩散过程与理想球体假设存在差异。
2. 导弹制导简化：假设导弹仅依赖视线制导，未涉及多传感器融合与抗干扰机制，实际场景可能更复杂。
3. 遮蔽判据单一：以“最短距离≤10 m”为唯一判据，未反映不同烟幕浓度分布对导引头探测的概率性影响。
4. 计算规模受限：当无人机数量与投放弹量增加时，枚举式搜索与贪心算法可能难以覆盖所有最优组合。

## 7.3改进

1. 引入环境建模：可考虑风速、湍流等大气条件对云团漂移和稀释的影响，使模型更接近真实情况。
2. 完善制导模型：引入导弹多传感器制导机制，结合概率判定方法，评估烟幕在复杂制导条件下的效能。
3. 丰富遮蔽判据：改进为浓度场分布模型，采用有效遮蔽概率或信噪比衰减指标，更符合实际电磁与光学对抗机理。
4. 优化算法提升：结合遗传算法、粒子群或强化学习等智能优化方法，提高大规模多机多弹场景下的计算效率与解的质量。

## 7.4推广

1. 多目标防护场景：模型可推广至多个真目标的同时防护问题，研究多目标协同下的烟幕投放策略。
2. 多类型干扰手段融合：可将烟幕干扰与电子干扰、诱饵释放等措施结合，构建综合对抗模型。
3. 任务规划系统应用：模型可嵌入到无人机群作战任务规划与决策支持系统，实现实时投放方案优化。
4. 其他领域借鉴：类似的“运动体—遮蔽体—目标”问题，也可用于无人机集群遮蔽探测、战场隐身伪装及灾害救援中的遮挡优化研究。

# 参考文献

[1]曲林涛,彭京徽,赵英泽,等. 异构多智能体（无人艇-机）协同策略研究[J/OL].火力与指挥控制,1-7[2025-09-04].https://link.cnki.net/urlid/14.1138.TJ.20250902.1724.002.

[2]张涛. 高层住宅建筑避难层功能提升与消防救援协同策略研究[J].中国住宅设施,2025,(06):64-66.

[3]黄琳雅.基于复杂网络的无人机集群智能协同策略研究[D].四川师范大学,2025.DOI:10.27347/d.cnki.gssdu.2025.001139.

[4]伍勉哲.基于机器学习的无人机辅助数据收集研究[D].电子科技大学,2025.DOI:10.27005/d.cnki.gdzku.2025.000294.

[5]王轲昕.面向组网雷达的无人机协同欺骗干扰方法研究[D].电子科技大学,2025.DOI:10.27005/d.cnki.gdzku.2025.003696.

[6]梁狄,许芳,李雅琦,等. 低空经济背景下城市区域轻小型物流无人机技术发展策略研究[J].企业科技与发展,2025,(02):1-7.DOI:10.20137/j.cnki.45-1359/t.20250102.006.

[7]龚英,涂熳熳,周愉峰. 卡车-无人机协同的洪灾应急选址-路径鲁棒优化问题[J].计算机工程与应用,2025,61(14):307-321.

[8]李晨玉,邢卓琳,苏子康,等. 时空散布下无人机集群空中加油集结路径规划[J].无人系统技术,2024,7(05):13-23.DOI:10.19942/j.issn.2096-5915.2024.05.44.

[9]毛玉来,朱宁,赵旭晟. 基于城市安全的无人机与机器人的协同应用及优化[J].中国战略新兴产业,2024,(23):115-117.

[10]李婷婷,王琪,王嘉康,等.SWARM-LLM:基于大语言模型的无人集群任务规划系统[C]//中国指挥与控制学会.首届全国大模型与决策智能大会论文集.中国科学院计算技术研究所;中国科学院大学;,2024:20-21.DOI:10.26914/c.cnkihy.2024.081062.

[11]王忠禹.面向多无人机对抗的多智能体迁移强化学习方法研究[D].大连理工大学,2024.DOI:10.26991/d.cnki.gdllu.2024.003359.

[12]曹博文.无人机辅助移动边缘计算中计算卸载和安全数据传输研究[D].河北大学,2024.DOI:10.27103/d.cnki.ghebu.2024.002050.

[13]杨洋.基于强化学习的智能蓝军电子对抗协同决策研究[D].中国科学院大学(中国科学院长春光学精密机械与物理研究所),2024.DOI:10.27522/d.cnki.gkcgs.2024.000098.

[14]张笑桐.基于卡车-无人机协同的城市药品应急配送研究[D].哈尔滨工业大学,2024.DOI:10.27061/d.cnki.ghgdu.2024.000585.

[15]黄达.基于分层架构的无人机群资源管理策略研究[D].电子科技大学,2024.DOI:10.27005/d.cnki.gdzku.2024.000418.

[16]黄振,尹航,童颖裔,等. 无人机集群对地攻击时空协同控制策略[J].工程科学学报,2024,46(09):1565-1573.DOI:10.13374/j.issn2095-9389.2023.09.13.005.

[17]王荣浩,文晓,向峥嵘. 人机融合系统协同与优化方法研究进展[J].指挥控制与仿真,2024,46(05):103-113.

[18]邓丽敏,陈蓓蓓,刘慧. 基于强化学习的通信受限环境多无人机协同策略[J].数字通信世界,2023,(12):42-44.

[19]许坦奇,刘正新.采用模型预测控制方法实现太阳能无人机的光储能源系统的能量平衡[C]//上海市太阳能学会.第十九届中国太阳级硅及光伏发电研讨会(19th CSPV)论文集.中国科学院上海微系统与信息技术研究所;,2023:5192-5207.DOI:10.26914/c.cnkihy.2023.073567.

[20]孙红燕.面向对抗任务的无人机集群智能协同策略研究[D].西安工业大学,2023.DOI:10.27391/d.cnki.gxagu.2023.000059.

[21]罗仁威.基于强化学习的非对称空战多智能体系统的设计与实现[D].电子科技大学,2023.DOI:10.27005/d.cnki.gdzku.2023.004687.

[22]程进,胡寒栋,江业帆,等. 基于强化学习的通信受限环境多无人机协同策略[J].无人系统技术,2022,5(05):12-20.DOI:10.19942/j.issn.2096-5915.2022.5.045.

[23]韩强.飞行器姿态协同有限时间控制方法研究[D].西南科技大学,2022.DOI:10.27415/d.cnki.gxngc.2022.001067.

[24]冷珊珊.3D复杂环境下的无人机航迹规划与任务分配方法研究[D].东南大学,2022.DOI:10.27014/d.cnki.gdnau.2022.003341.

[25]许钢焱,龙玉莹,王欣悦,等. 考虑货车-无人机协同的灾后应急响应策略及调度优化[J].安全与环境学报,2023,23(05):1587-1595.DOI:10.13637/j.issn.1009-6094.2022.0084.

[26]黄大通.对合成孔径雷达的分布式协同干扰方法研究[D].国防科技大学,2022.DOI:10.27052/d.cnki.gzjgu.2022.000079.

[27]姜丽媛.通信约束下UUV-UAV协同目标搜索方法研究[D].哈尔滨工程大学,2022.DOI:10.27060/d.cnki.ghbcu.2022.002476.

[28]苏雅倩文.通信条件受限环境下的多无人机自适应协同探索策略研究[D].军事科学院,2021.DOI:10.27193/d.cnki.gjsky.2021.000153.

[29]安凯.高超声速飞行器中末段协同轨迹规划技术研究[D].国防科技大学,2021.DOI:10.27052/d.cnki.gzjgu.2021.001042.

[30]孙沁,李鸿旭,王毓智,等. 韧性无人机群多域协同方法建模与求解[J].航空学报,2022,43(05):544-557.

[31]董立军.多无人机协同航迹规划算法研究[D].东北大学,2021.DOI:10.27007/d.cnki.gdbeu.2021.001810.

[32]周凌子.基于鸽群优化算法的多无人机协同航迹规划[D].南京航空航天大学,2021.DOI:10.27239/d.cnki.gnhhu.2021.001522.

[33]王宁.基于多无人机的协同目标跟踪技术研究与原型系统实现[D].国防科技大学,2020.DOI:10.27052/d.cnki.gzjgu.2020.000888.

[34]程聪聪.面向任务的无人机集群韧性评估研究[D].国防科技大学,2020.DOI:10.27052/d.cnki.gzjgu.2020.000227.

[35]冉惟之.基于群体智能的无人机集群协同对抗系统的设计与实现[D].电子科技大学,2020.DOI:10.27005/d.cnki.gdzku.2020.003004.

[36]辛宏博.小型无人机末端协同到达控制方法研究[D].国防科技大学,2020.DOI:10.27052/d.cnki.gzjgu.2020.000181.

[37]李坚强,复杂环境下空中/地面多异构机器人立体协同策略研究.广东省,深圳大学,2020-04-24.

[38]胡孟阳.基于群智能算法的多无人机任务分配和协同路径规划[D].天津大学,2019.DOI:10.27356/d.cnki.gtjdu.2019.003899.

[39]郭喆,陆明,王祥. 基于无人机的离散结构自主建造技术初探[J].建筑技艺,2019,(09):40-45.

[40]刘卫东. 2018年中国研究生数学建模竞赛E题综述[J].数学的实践与认识,2019,49(16):238-243.

# 附录

## 附件 1：问题一matlab代码

|  |
| --- |
| 工具：matlab |
| %% 问题 1: FY1 投放 1 枚烟幕干扰弹对 M1 的遮蔽时长  %% 问题 1: FY1 投放 1 枚烟幕干扰弹对 M1 的遮蔽时长  clear; clc;  %% 常量与初始条件  g = 9.8; % 重力加速度 m/s^2  Vm = 300; % 导弹速度 m/s  Vu = 120; % 无人机速度 m/s  t\_release = 1.5; % 投放时刻 s  delay = 3.6; % 延时起爆 s  t\_explosion = t\_release + delay;  % 真目标  T = [0, 200, 0];  % M1 初始位置  M1 = [20000, 0, 2000];  % 无人机 FY1 初始位置  FY1 = [17800, 0, 1800];  % 假目标(原点)  O = [0, 0, 0];  %% 导弹方向  um = (O - M1) / norm(O - M1);  %% 无人机速度向量 (朝向原点，x 轴负方向)  uu = (O - FY1); uu(3) = 0; uu = uu / norm(uu); % 等高度，只取水平向量  uu = uu \* Vu;  %% 投放点  r\_release = FY1 + uu \* t\_release;  %% 起爆点  dt = delay;  r\_explosion = r\_release + uu \* dt + 0.5\*[0,0,-g]\*dt^2;  %% 云团中心轨迹  % c(t) for t >= t\_explosion  c = @(t) r\_explosion + [0,0,-3]\*(t - t\_explosion);  %% 导弹轨迹  rm = @(t) M1 + Vm\*t\*um;  %% 点到线段距离函数  distPoint2Segment = @(P,A,B) ...  norm(P - (A + max(0,min(1, dot(P-A,B-A)/norm(B-A)^2))\*(B-A)));  %% 扫描时间区间  t\_range = t\_explosion:0.01:(t\_explosion+20);  mask = false(size(t\_range));  for k = 1:length(t\_range)  t = t\_range(k);  A = rm(t); B = T; P = c(t);  d = distPoint2Segment(P,A,B);  if d <= 10  mask(k) = true;  end  end  %% 找出连续区间  d\_mask = diff([false mask false]);  start\_idx = find(d\_mask==1);  end\_idx = find(d\_mask==-1)-1;  intervals = [t\_range(start\_idx)' , t\_range(end\_idx)'];  durations = intervals(:,2)-intervals(:,1);  total\_duration = sum(durations);  %% 输出结果  disp('有效遮蔽时间区间(s):');  disp(intervals);  fprintf('总有效遮蔽时长 = %.3f s\n', total\_duration); |
| 运行结果  有效遮蔽时间区间(s):  8.0380 9.4480  总有效遮蔽时长 = 1.410 s |

## 附件 2：问题二matlab代码

|  |
| --- |
| **工具：matlab** |
| clear; clc;  %% --------------------- 常量与已知条件 ---------------------  g = 9.8; % 重力 m/s^2  R = 10; % 有效遮蔽半径 m  sink = 3; % 云团下沉速度 m/s  h = 0.02; % 时间离散步长 s（可调：0.01~0.05）  Vm = 300; % 导弹速度 m/s  O = [0, 0, 0]; % 假目标原点  T = [0, 200, 0]; % 真目标底面圆心  M = [20000, 0, 2000]; % M1 初始位置  F = [17800, 0, 1800]; % FY1 初始位置（等高度直飞）  um = (O - M) / norm(O - M); % 导弹方向单位向量  params = struct('g',g,'R',R,'sink',sink,'h',h,'Vm',Vm, ...  'O',O,'T',T,'M',M,'F',F,'um',um);  %% --------------------- 粗搜索（多起点） ---------------------  % 网格（可根据计算力调整）  psi\_grid = linspace(-pi, pi, 16); % 航向角  v\_grid = linspace(70, 140, 4); % 速度约束 [70,140]  trel\_grid = 0:0.5:4; % 投放时刻 (s)  tau\_grid = 0.5:0.5:5; % 延时起爆 (s)  best = struct('x',[],'J',-inf,'intervals',[],'r\_rel',[],'r\_exp',[]);  results = []; % [psi, v, trel, tau, J]  for psi = psi\_grid  for v = v\_grid  for trel = trel\_grid  for tau = tau\_grid  x = [psi, v, trel, tau];  [J, intervals, r\_rel, r\_exp] = objective\_J(x, params);  results(end+1,:) = [psi, v, trel, tau, J]; %#ok<AGROW>  if J > best.J  best.x = x; best.J = J;  best.intervals = intervals;  best.r\_rel = r\_rel; best.r\_exp = r\_exp;  end  end  end  end  end  % 选出若干最优候选用于细化  [~, idx\_sorted] = sort(results(:,5), 'descend');  topN = min(10, size(results,1));  candidates = results(idx\_sorted(1:topN), 1:4);  %% --------------------- 局部细化（坐标式爬山法） ---------------------  lb = [-pi, 70, 0, 0.05]; % 变量下界: psi, v, trelease, tau  ub = [ pi,140, 8, 10.00]; % 变量上界  step0 = [pi/24, 10, 0.25, 0.25]; % 初始步长  tolJ = 1e-4; % 收敛阈值  maxIter = 200; % 最大迭代  for i = 1:size(candidates,1)  [x\_loc, J\_loc, intervals\_loc, r\_rel, r\_exp] = local\_refine(candidates(i,:), params, lb, ub, step0, tolJ, maxIter);  if J\_loc > best.J  best.x = x\_loc; best.J = J\_loc;  best.intervals = intervals\_loc;  best.r\_rel = r\_rel; best.r\_exp = r\_exp;  end  end  %% --------------------- 输出最优结果 ---------------------  psi = best.x(1); v = best.x(2); trel = best.x(3); tau = best.x(4);  texp = trel + tau;  K = floor(20/h); %#ok<NASGU> % 供阅读：起爆后 20 s 的离散步数  fprintf('\n===== 最优策略（近似） =====\n');  fprintf('航向角 psi = %.3f rad (%.2f deg)\n', psi, rad2deg(psi));  fprintf('飞行速度 v = %.3f m/s\n', v);  fprintf('投放时刻 trelease = %.3f s\n', trel);  fprintf('延时起爆 tau = %.3f s (texplosion = %.3f s)\n', tau, texp);  fprintf('投放点 r\_release = [%.3f, %.3f, %.3f]\n', best.r\_rel);  fprintf('起爆点 r\_explosion = [%.3f, %.3f, %.3f]\n', best.r\_exp);  if isempty(best.intervals)  fprintf('无有效遮蔽（J = 0）\n');  else  dur = sum(best.intervals(:,2) - best.intervals(:,1));  fprintf('总有效遮蔽时长 J ≈ %.3f s\n', dur);  fprintf('时间区间（s）：\n');  disp(best.intervals);  end  %% --------------------- （可选）可视化 ---------------------  doPlot = false; % 如需可视化，设为 true  if doPlot  figure('Color','w'); hold on; grid on; axis equal; view(45,25);  xlabel('x / m'); ylabel('y / m'); zlabel('z / m');  % 真/假目标圆心  plot3(T(1),T(2),T(3),'gp','MarkerFaceColor','g','MarkerSize',10); text(T(1),T(2),T(3)+20,'True center');  plot3(O(1),O(2),O(3),'bp','MarkerFaceColor','b','MarkerSize',10); text(O(1),O(2),O(3)+20,'Fake center');  % M1 起点与导弹轨迹（20 s 期间）  t0 = texp; t1 = texp+20;  tt = linspace(t0,t1,200);  rm = M + (Vm\*tt.').\*um;  plot3(M(1),M(2),M(3),'r^','MarkerFaceColor','r'); text(M(1),M(2),M(3)+80,'M1');  plot3(rm(:,1), rm(:,2), rm(:,3), 'r-','LineWidth',1.2);  % FY1 轨迹（至投放）  uu = [v\*cos(psi), v\*sin(psi), 0];  tline = linspace(0,trel,50);  ru = F + tline.' .\* uu;  plot3(F(1),F(2),F(3),'ko','MarkerFaceColor','k'); text(F(1),F(2),F(3)+50,'FY1');  plot3(ru(:,1), ru(:,2), ru(:,3), 'k--');  % 投放点与起爆点  plot3(best.r\_rel(1), best.r\_rel(2), best.r\_rel(3),'mo','MarkerFaceColor','m'); text(best.r\_rel(1), best.r\_rel(2), best.r\_rel(3)+50,'release');  plot3(best.r\_exp(1), best.r\_exp(2), best.r\_exp(3),'ms','MarkerFaceColor','m'); text(best.r\_exp(1), best.r\_exp(2), best.r\_exp(3)+50,'explosion');  % 云团下沉轨迹（20 s）  c0 = best.r\_exp;  cLine = c0 + [zeros(numel(tt),1), zeros(numel(tt),1), -sink\*(tt.'-texp)];  plot3(cLine(:,1), cLine(:,2), cLine(:,3), 'm-','LineWidth',1.2);  title('Optimal plan (trajectory & key points)');  end  %% ====================== 函数定义 ======================  function [J, intervals, r\_release, r\_explosion] = objective\_J(x, P)  % 计算给定 x = [psi, v, trel, tau] 时的离散化遮蔽时长 J 及遮蔽区间  psi = x(1); v = x(2); trel = x(3); tau = x(4);  if v < 70 || v > 140 || trel < 0 || tau <= 0  J = 0; intervals = []; r\_release = [NaN,NaN,NaN]; r\_explosion = r\_release; return;  end  % 无人机运动（等高匀速）  uu = [v\*cos(psi), v\*sin(psi), 0];  r\_release = P.F + trel\*uu;  % 起爆点（仅受重力）  r\_explosion = r\_release + uu\*tau + 0.5\*[0,0,-P.g]\*(tau^2);  % 若起爆在地面以下，认为无效  if r\_explosion(3) <= 0  J = 0; intervals = []; return;  end  % 时间窗  texp = trel + tau;  t0 = texp; t1 = texp + 20;  K = floor((t1 - t0)/P.h); % ≈ floor(20/h)  tk = t0 + (0:K)'\*P.h;  % 云团中心 c(t)  c = r\_explosion + [0,0,-P.sink] .\* (tk - texp);  % 导弹轨迹 r\_m(t)  rm = P.M + (P.Vm\*tk).\*P.um;  % 线段 [A,B]，B = T  A = rm; B = repmat(P.T, size(A,1), 1);  BA = B - A; % 每个时刻的向量  PA = c - A;  % 点到线段距离（向量化）  BA2 = sum(BA.^2, 2);  s = sum(PA.\*BA, 2) ./ BA2; % 投影比例  s = max(0, min(1, s)); % clamp 到 [0,1]  Q = A + s .\* BA; % 最近点  d = sqrt(sum((c - Q).^2, 2)); % 距离  mask = (d <= P.R);  % 统计遮蔽区间与总时长  [intervals, J] = mask\_to\_intervals(tk, mask);  end  function [intervals, totalDur] = mask\_to\_intervals(tk, mask)  % 将布尔掩码转换成若干 [t\_start, t\_end] 区间，并返回总时长  if ~any(mask)  intervals = []; totalDur = 0; return;  end  mm = [false; mask; false];  dmm = diff(mm);  iStart = find(dmm == 1);  iEnd = find(dmm == -1) - 1;  intervals = [tk(iStart), tk(iEnd)];  totalDur = sum(intervals(:,2) - intervals(:,1));  end  function [x\_best, J\_best, intervals\_best, r\_rel\_best, r\_exp\_best] = local\_refine(x0, P, lb, ub, step0, tolJ, maxIter)  % 简单坐标式爬山法（无工具箱）  x\_best = clip(x0, lb, ub);  [J\_best, intervals\_best, r\_rel\_best, r\_exp\_best] = objective\_J(x\_best, P);  step = step0;  iter = 0; improved = true;  while iter < maxIter && improved  improved = false; iter = iter + 1;  for i = 1:4  for sgn = [+1, -1]  x\_try = x\_best;  x\_try(i) = x\_try(i) + sgn\*step(i);  x\_try = clip(x\_try, lb, ub);  [J\_try, intervals\_try, r\_rel\_try, r\_exp\_try] = objective\_J(x\_try, P);  if J\_try > J\_best + tolJ  x\_best = x\_try; J\_best = J\_try;  intervals\_best = intervals\_try;  r\_rel\_best = r\_rel\_try; r\_exp\_best = r\_exp\_try;  improved = true;  end  end  end  if ~improved  % 缩小步长继续细找  step = step \* 0.5;  if max(step) < 1e-3  break;  end  improved = true; % 允许再尝试一次更小步长  end  end  end  function x = clip(x, lb, ub)  % 变量边界裁剪  x = min(max(x, lb), ub);  end  function deg = rad2deg(rad)  % 角度单位转换（避免需要额外工具箱）  deg = rad \* 180/pi;  end |
| **运行结果：**  ===== 最优策略（近似） =====  航向角 psi = 0.144 rad (8.25 deg)  飞行速度 v = 75.000 m/s  投放时刻 trelease = 0.125 s  延时起爆 tau = 0.938 s (texplosion = 1.063 s)  投放点 r\_release = [17809.278, 1.345, 1800.000]  起爆点 r\_explosion = [17878.863, 11.435, 1795.693]  总有效遮蔽时长 J ≈ 4.720 s  时间区间（s）：  1.1625 5.8825 |

## 附件 3：问题三matlab代码

|  |
| --- |
| **工具：matlab** |
| clear; clc;  %% ------------------ 常量与场景参数 ------------------  g = 9.8; % 重力 m/s^2  R = 10.0; % 有效遮蔽球半径 m  sink = 3.0; % 云团下沉速度 m/s  Vm = 300.0; % 导弹速度 m/s  h = 0.05; % 时间离散化步长 s（可调 0.01~0.05）  % 注：起爆后20 s的离散步数 K = floor(20/h)  % 坐标（单位 m）  O = [0, 0, 0]; % 假目标原点  T = [0, 200, 0]; % 真目标底面圆心  M = [20000, 0, 2000]; % M1 初始位置  F = [17800, 0, 1800]; % FY1 初始位置（等高度飞）  % 导弹方向（指向原点）  um = (O - M) / norm(O - M);  % 无人机航向：这里固定“朝原点”方向（与x负向一致），即psi=pi（180°）  psi = pi;  % 速度候选集合（可按需增加）  v\_candidates = [100, 120, 140];  % 时间网格范围（覆盖三弹）  t\_end = 45; % 足够覆盖三枚弹的有效窗  tk = (0:h:t\_end).'; % 列向量  % 第一、二、三枚弹的投放时刻/延时网格  trel\_grid = 0:0.5:12; % 投放时刻网格 s  tau\_grid = 0.5:0.5:6; % 延时起爆网格 s  %% ------------------ 搜索与贪心选优 ------------------  best.union\_duration = -inf; % 记录最优并集时长  best.v = NaN;  for v = v\_candidates  % === 第1枚：最大化单枚遮蔽时长 ===  best1.duration = -inf;  for trel = trel\_grid  for tau = tau\_grid  [mask1, rrel1, rexp1] = coverage\_mask\_one(v, psi, trel, tau, tk, F, M, T, um, g, R, sink, Vm);  d1 = mask\_duration(mask1, tk);  if d1 > best1.duration  best1.duration = d1;  best1.trel = trel; best1.tau = tau;  best1.mask = mask1; best1.rrel = rrel1; best1.rexp = rexp1;  end  end  end  % === 第2枚：在 trel2 >= trel1 + 1 的约束下，最大化并集时长 ===  best12.union = -inf;  for trel2 = trel\_grid(trel\_grid >= best1.trel + 1.0)  for tau2 = tau\_grid  [mask2, rrel2, rexp2] = coverage\_mask\_one(v, psi, trel2, tau2, tk, F, M, T, um, g, R, sink, Vm);  union12 = union\_duration({best1.mask, mask2}, tk);  if union12 > best12.union  best12.union = union12;  best12.trel2 = trel2; best12.tau2 = tau2;  best12.mask2 = mask2; best12.rrel2 = rrel2; best12.rexp2 = rexp2;  best12.dur2 = mask\_duration(mask2, tk);  end  end  end  % === 第3枚：在 trel3 >= trel2 + 1 的约束下，最大化与前两枚并集 ===  best123.union = -inf;  for trel3 = trel\_grid(trel\_grid >= best12.trel2 + 1.0)  for tau3 = tau\_grid  [mask3, rrel3, rexp3] = coverage\_mask\_one(v, psi, trel3, tau3, tk, F, M, T, um, g, R, sink, Vm);  union123 = union\_duration({best1.mask, best12.mask2, mask3}, tk);  if union123 > best123.union  best123.union = union123;  best123.trel3 = trel3; best123.tau3 = tau3;  best123.mask3 = mask3; best123.rrel3 = rrel3; best123.rexp3 = rexp3;  best123.dur3 = mask\_duration(mask3, tk);  end  end  end  % 比较不同速度的联合时长  if best123.union > best.union\_duration  best.union\_duration = best123.union;  best.v = v;  best.psi = psi;  % 记录三弹细节  best.b1.trel = best1.trel; best.b1.tau = best1.tau;  best.b1.mask = best1.mask; best.b1.rrel = best1.rrel; best.b1.rexp = best1.rexp;  best.b1.duration = best1.duration;  best.b2.trel = best12.trel2; best.b2.tau = best12.tau2;  best.b2.mask = best12.mask2; best.b2.rrel = best12.rrel2; best.b2.rexp = best12.rexp2;  best.b2.duration = best12.dur2;  best.b3.trel = best123.trel3; best.b3.tau = best123.tau3;  best.b3.mask = best123.mask3; best.b3.rrel = best123.rrel3; best.b3.rexp = best123.rexp3;  best.b3.duration = best123.dur3;  end  end  %% ------------------ 输出控制台总结 ------------------  deg = mod(rad2deg\_safe(best.psi), 360);  fprintf('\n=== 最优策略（贪心近似）===\n');  fprintf('无人机航向 = %.2f 度（相对x轴逆时针）\n', deg);  fprintf('无人机速度 = %.2f m/s\n', best.v);  % 三枚弹时间区间  [ints1, dur1] = mask\_to\_intervals(best.b1.mask, tk);  [ints2, dur2] = mask\_to\_intervals(best.b2.mask, tk);  [ints3, dur3] = mask\_to\_intervals(best.b3.mask, tk);  % 并集区间  union\_mask = best.b1.mask | best.b2.mask | best.b3.mask;  [intsU, durU] = mask\_to\_intervals(union\_mask, tk);  fprintf('\n弹1：trel=%.2f s, tau=%.2f s, texp=%.2f s, 单枚时长=%.3f s\n', best.b1.trel, best.b1.tau, best.b1.trel+best.b1.tau, dur1);  disp(' 遮蔽区间（s）：'); disp(ints1);  fprintf('投放点 rrel1 = [%.3f, %.3f, %.3f]\n', best.b1.rrel);  fprintf('起爆点 rexp1 = [%.3f, %.3f, %.3f]\n', best.b1.rexp);  fprintf('\n弹2：trel=%.2f s, tau=%.2f s, texp=%.2f s, 单枚时长=%.3f s\n', best.b2.trel, best.b2.tau, best.b2.trel+best.b2.tau, dur2);  disp(' 遮蔽区间（s）：'); disp(ints2);  fprintf('投放点 rrel2 = [%.3f, %.3f, %.3f]\n', best.b2.rrel);  fprintf('起爆点 rexp2 = [%.3f, %.3f, %.3f]\n', best.b2.rexp);  fprintf('\n弹3：trel=%.2f s, tau=%.2f s, texp=%.2f s, 单枚时长=%.3f s\n', best.b3.trel, best.b3.tau, best.b3.trel+best.b3.tau, dur3);  disp(' 遮蔽区间（s）：'); disp(ints3);  fprintf('投放点 rrel3 = [%.3f, %.3f, %.3f]\n', best.b3.rrel);  fprintf('起爆点 rexp3 = [%.3f, %.3f, %.3f]\n', best.b3.rexp);  fprintf('\n三枚并集遮蔽区间（s）：\n'); disp(intsU);  fprintf('并集总时长 = %.3f s\n', durU);  %% ------------------ 写入 Excel 模板（Sheet1） ------------------  xlsxPath = 'result1.xlsx'; % 请确保模板与本脚本在同一目录  data = [];  try  data = readcell(xlsxPath, 'Sheet', 'Sheet1');  catch  % 若读取失败，则构造一个新表头  data = {  '无人机运动方向','无人机运动速度 (m/s)','烟幕干扰弹编号', ...  '烟幕干扰弹投放点的x坐标 (m)','烟幕干扰弹投放点的y坐标 (m)','烟幕干扰弹投放点的z坐标 (m)', ...  '烟幕干扰弹起爆点的x坐标 (m)','烟幕干扰弹起爆点的y坐标 (m)','烟幕干扰弹起爆点的z坐标 (m)', ...  '有效干扰时长 (s)';  [],[],1,[],[],[],[],[],[],[];  [],[],2,[],[],[],[],[],[],[];  [],[],3,[],[],[],[],[],[],[];  '注：以x轴为正向，逆时针方向为正，取值0~360（度）。',[],[],[],[],[],[],[],[],[]  };  end  % 预期：第1行是表头，第2~4行为三条数据，第5行为注释  % 若模板不是此布局，可按需要调整下面的行索引 rowIdx。  rowIdx = [2,3,4];  % 组装三枚数据  rows = cell(3, 10);  bombs = {best.b1, best.b2, best.b3};  for i = 1:3  b = bombs{i};  rrel = b.rrel; rexp = b.rexp;  rows{i,1} = deg;  rows{i,2} = best.v;  rows{i,3} = i;  rows{i,4} = rrel(1);  rows{i,5} = rrel(2);  rows{i,6} = rrel(3);  rows{i,7} = rexp(1);  rows{i,8} = rexp(2);  rows{i,9} = rexp(3);  rows{i,10} = round(mask\_duration(b.mask, tk), 3);  end  % 将 rows 覆盖写入 data 的第2~4行（保留其余内容）  for i = 1:3  data(rowIdx(i), 1:10) = rows(i, :);  end  writecell(data, xlsxPath, 'Sheet', 'Sheet1');  fprintf('\n已写入文件：%s（Sheet1 前三行）\n', xlsxPath);  %% ================== 函数定义 ==================  function [mask, r\_release, r\_explosion] = coverage\_mask\_one(v, psi, trel, tau, tk, F, M, T, um, g, R, sink, Vm)  % 计算一枚弹的遮蔽掩码：在全局时间网格 tk 上，满足  % 1) t ∈ [texp, texp+20]  % 2) 导弹-真目标视线到云团中心球的最短距离 <= R  % 则 mask(tk)=true  uu = [v\*cos(psi), v\*sin(psi), 0];  r\_release = F + trel\*uu;  r\_explosion = r\_release + uu\*tau + 0.5\*[0,0,-g]\*(tau^2);  % 起爆点若在地面以下，视为无效  if r\_explosion(3) <= 0  mask = false(size(tk)); return;  end  texp = trel + tau;  % 云团中心（仅在有效窗内）  c = zeros(numel(tk), 3);  c(:,1) = r\_explosion(1);  c(:,2) = r\_explosion(2);  c(:,3) = r\_explosion(3) - sink\*(tk - texp);  % 导弹位置  rm = M + (Vm\*tk).\*um;  % 仅计算有效窗  valid = (tk >= texp) & (tk <= texp + 20.0);  mask = false(size(tk));  for i = find(valid).'  d = distPoint2Segment(c(i,:), rm(i,:), T);  if d <= R  mask(i) = true;  end  end  end  function d = distPoint2Segment(P, A, B)  % 点P到线段AB的距离  BA = B - A;  l2 = dot(BA, BA);  if l2 == 0  d = norm(P - A); return;  end  s = dot(P - A, BA) / l2;  s = max(0, min(1, s));  Q = A + s \* BA;  d = norm(P - Q);  end  function dur = mask\_duration(mask, tk)  % 布尔掩码对应的总时长（采用 tk 的步进近似）  if ~any(mask), dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  dur = sum(tk(iE) - tk(iS));  end  function [intervals, dur] = mask\_to\_intervals(mask, tk)  % 将掩码转换成若干 [t\_start, t\_end] 区间，并返回总时长  if ~any(mask), intervals = []; dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  intervals = [tk(iS), tk(iE)];  dur = sum(intervals(:,2) - intervals(:,1));  end  function dur = union\_duration(mask\_list, tk)  % 若干掩码的并集总时长  if isempty(mask\_list), dur = 0; return; end  U = false(size(mask\_list{1}));  for k = 1:numel(mask\_list)  U = U | mask\_list{k};  end  dur = mask\_duration(U, tk);  end  function d = rad2deg\_safe(r)  % 兼容性：若MATLAB版本不带rad2deg，用此函数  d = r \* 180/pi;  end |
| **求解结果：**  无人机航向 = 180.00 度（相对x轴逆时针）  无人机速度 = 140.00 m/s  弹1：trel=0.00 s, tau=3.50 s, texp=3.50 s, 单枚时长=3.000 s  遮蔽区间（s）：  6 9  投放点 rrel1 = [17800.000, 0.000, 1800.000]  起爆点 rexp1 = [17310.000, 0.000, 1739.975]  弹2：trel=3.50 s, tau=5.50 s, texp=9.00 s, 单枚时长=2.550 s  遮蔽区间（s）：  9.0000 11.5500  投放点 rrel2 = [17310.000, 0.000, 1800.000]  起爆点 rexp2 = [16540.000, 0.000, 1651.775]  弹3：trel=5.50 s, tau=6.00 s, texp=11.50 s, 单枚时长=1.250 s  遮蔽区间（s）：  11.5000 12.7500  投放点 rrel3 = [17030.000, 0.000, 1800.000]  起爆点 rexp3 = [16190.000, 0.000, 1623.600]  三枚并集遮蔽区间（s）：  6.0000 12.7500  并集总时长 = 6.750 s  已写入文件：result1.xlsx（Sheet1 前三行） |

## 附件 4：问题四matlab代码

|  |
| --- |
| **工具：matlab** |
| clear; clc;  %% ------------------ 常量与场景参数 ------------------  g = 9.8; % 重力 m/s^2  R = 10.0; % 有效遮蔽球半径 m  sink = 3.0; % 云团下沉速度 m/s  Vm = 300.0; % 导弹速度 m/s  h = 0.02; % 时间离散步长（更细，边界更易命中）  t\_end = 90; % 评估窗口更长（FY3 更容易命中视线）  tk = (0:h:t\_end).'; % 列向量时间网格  % 坐标（单位 m）  O = [0, 0, 0]; % 假目标原点  T = [0, 200, 0]; % 真目标底面圆心  M = [20000, 0, 2000]; % M1 初始位置  FY = struct();  FY.FY1 = [17800, 0, 1800];  FY.FY2 = [12000, 1400, 1400];  FY.FY3 = [ 6000,-3000, 700]; % 注意：若模板写成“FV3”，请在写回 Excel 时兼容  % 导弹方向单位向量（指向原点）  um = (O - M) / norm(O - M);  %% ------------------ 搜索网格设置（放宽） ------------------  psi\_grid = linspace(-pi, pi, 24); psi\_grid(end) = []; % 航向更密  v\_grid = [70 90 100 110 120 130 140]; % 速度更多档  trel\_grid = 0:1:30; % 投放时刻更长  tau\_grid = 0.5:0.5:10; % 延时更长  cap\_each = 40; % 每机保留的候选上限（单枚时长降序裁剪）  %% ------------------ 为每架无人机生成候选解（含回退与兜底） ------------------  names = fieldnames(FY);  perUAV = struct();  for n = 1:numel(names)  name = names{n};  F0 = FY.(name);  cands = [];  % ------- 第一轮：全局网格 -------  for ip = 1:numel(psi\_grid)  psi = psi\_grid(ip);  for iv = 1:numel(v\_grid)  v = v\_grid(iv);  for ir = 1:numel(trel\_grid)  trel = trel\_grid(ir);  for itau = 1:numel(tau\_grid)  tau = tau\_grid(itau);  [mask, r\_rel, r\_exp] = coverage\_mask\_one(F0, v, psi, trel, tau, tk, ...  M, T, um, g, R, sink, Vm);  dur = mask\_duration(mask, tk);  if dur > 0  c.name = name; c.psi = psi;  c.heading\_deg = mod(rad2deg\_safe(psi), 360);  c.v = v; c.trel = trel; c.tau = tau;  c.r\_rel = r\_rel; c.r\_exp = r\_exp;  c.mask = mask; c.dur = dur;  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  % ------- 若第一轮无候选：二次回退（指向原点 ±30°） -------  if isempty(cands)  vdir = [O(1)-F0(1), O(2)-F0(2)];  base\_psi = atan2(vdir(2), vdir(1));  psi\_fallback = base\_psi + deg2rad([-30 -20 -10 0 10 20 30]);  trel\_fb = 0:1:40; % 更长的机动时间  tau\_fb = 0.5:0.5:12; % 更长的引信延时  v\_fb = [70 90 100 110 120 130 140];  for ip = 1:numel(psi\_fallback)  psi = psi\_fallback(ip);  for iv = 1:numel(v\_fb)  v = v\_fb(iv);  for ir = 1:numel(trel\_fb)  trel = trel\_fb(ir);  for itau = 1:numel(tau\_fb)  tau = tau\_fb(itau);  [mask, r\_rel, r\_exp] = coverage\_mask\_one(F0, v, psi, trel, tau, tk, ...  M, T, um, g, R, sink, Vm);  dur = mask\_duration(mask, tk);  if dur > 0  c.name = name; c.psi = psi;  c.heading\_deg = mod(rad2deg\_safe(psi), 360);  c.v = v; c.trel = trel; c.tau = tau;  c.r\_rel = r\_rel; c.r\_exp = r\_exp;  c.mask = mask; c.dur = dur;  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  end  % ------- 若仍为空：极端兜底（临时扩大评估窗到 120 s） -------  if isempty(cands)  warning('无人机 %s 第一/二轮均无候选，扩大评估时间到 120 s 再试。', name);  tk2 = (0:h:120).'; % 临时更长  for ip = 1:numel(psi\_grid)  psi = psi\_grid(ip);  for iv = 1:numel(v\_grid)  v = v\_grid(iv);  for ir = 1:numel(trel\_grid)  trel = trel\_grid(ir);  for itau = 1:numel(tau\_grid)  tau = tau\_grid(itau);  [mask2, r\_rel, r\_exp] = coverage\_mask\_one(F0, v, psi, trel, tau, tk2, ...  M, T, um, g, R, sink, Vm);  dur2 = mask\_duration(mask2, tk2);  if dur2 > 0  c.name = name; c.psi = psi;  c.heading\_deg = mod(rad2deg\_safe(psi), 360);  c.v = v; c.trel = trel; c.tau = tau;  c.r\_rel = r\_rel; c.r\_exp = r\_exp;  % 将临时掩码回映射到原 tk（previous 插值）  mask = interp1(tk2, double(mask2), tk, 'previous', 0) > 0.5;  c.mask = mask; c.dur = mask\_duration(mask, tk);  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  end  % ------- 仍为空则报错终止 -------  if isempty(cands)  error('无人机 %s 未找到任何有效候选，请再放宽网格或检查场景。', name);  end  % 排序裁剪  [~, idx] = sort([cands.dur], 'descend');  cands = cands(idx);  if numel(cands) > cap\_each, cands = cands(1:cap\_each); end  perUAV.(name) = cands;  fprintf('候选数 %s: %d\n', name, numel(cands));  end  %% ------------------ 三机组合择优（最大化并集时长） ------------------  best.union\_dur = -inf;  best.combo = [];  C1 = perUAV.FY1; C2 = perUAV.FY2; C3 = perUAV.FY3; % 若模板有“FV3”，此处仍用 FY3 逻辑  for i = 1:numel(C1)  m1 = C1(i).mask;  for j = 1:numel(C2)  m2 = C2(j).mask;  for k = 1:numel(C3)  m3 = C3(k).mask;  U = (m1 | m2 | m3);  durU = mask\_duration(U, tk);  if durU > best.union\_dur  best.union\_dur = durU;  best.combo = [C1(i), C2(j), C3(k)];  best.union\_mask = U;  end  end  end  end  %% ------------------ 打印结果（单枚与并集区间） ------------------  fprintf('\n=== 最优三机组合（近似）===\n');  for idx = 1:3  b = best.combo(idx);  [ints, dur] = mask\_to\_intervals(b.mask, tk);  fprintf('%s: heading=%.2f deg, v=%.1f m/s, trel=%.2f s, tau=%.2f s, texp=%.2f s\n', ...  b.name, b.heading\_deg, b.v, b.trel, b.tau, b.trel+b.tau);  fprintf(' r\_release = [%.3f, %.3f, %.3f]\n', b.r\_rel);  fprintf(' r\_explosion = [%.3f, %.3f, %.3f]\n', b.r\_exp);  fprintf(' 单枚遮蔽时长 = %.3f s；区间（s）：\n', dur);  disp(ints);  end  [intsU, durU] = mask\_to\_intervals(best.union\_mask, tk);  fprintf('并集遮蔽总时长 = %.3f s；并集区间（s）：\n', durU);  disp(intsU);  %% ------------------ 写入 Excel 模板（Sheet1） ------------------  xlsxPath = 'result2.xlsx'; % 模板文件名（与脚本同目录）  try  data = readcell(xlsxPath, 'Sheet', 'Sheet1');  catch  % 若模板读取失败，构造基本表头（兼容无模板场景）  data = {  '无人机编号','无人机运动方向','无人机运动速度 (m/s)', ...  '烟幕干扰弹投放点的x坐标 (m)','烟幕干扰弹投放点的y坐标 (m)','烟幕干扰弹投放点的z坐标 (m)', ...  '烟幕干扰弹起爆点的x坐标 (m)','烟幕干扰弹起爆点的y坐标 (m)','烟幕干扰弹起爆点的z坐标 (m)', ...  '有效干扰时长 (s)';  'FY1',[],[],[],[],[],[],[],[],[];  'FY2',[],[],[],[],[],[],[],[],[];  'FY3',[],[],[],[],[],[],[],[],[];  [],'注：以x轴为正向，逆时针方向为正，取值0~360（度）。',[],[],[],[],[],[],[],[]  };  end  % 若模板第一列写成了“FV3”，映射为 FY3  for r = 1:size(data,1)  if ischar(data{r,1}) || (isstring(data{r,1}) && strlength(data{r,1})>0)  s = upper(string(data{r,1}));  if s == "FV3", data{r,1} = 'FY3'; end  end  end  % 找到 FY1/FY2/FY3 所在的三行  rows = struct('FY1',[], 'FY2',[], 'FY3',[]);  for r = 1:size(data,1)  v = data{r,1};  if ischar(v) || (isstring(v) && strlength(v)>0)  key = char(string(v));  if isfield(rows, key) && isempty(rows.(key))  rows.(key) = r;  end  end  end  if isempty(rows.FY1) || isempty(rows.FY2) || isempty(rows.FY3)  error('未在模板首列识别到 FY1/FY2/FY3 三行，请检查模板。');  end  % 写入三机数据（列 2..10）  for idx = 1:3  b = best.combo(idx);  r = rows.(b.name);  data(r, 2:10) = { ...  round(b.heading\_deg,2), b.v, ...  b.r\_rel(1), b.r\_rel(2), b.r\_rel(3), ...  b.r\_exp(1), b.r\_exp(2), b.r\_exp(3), ...  round(mask\_duration(b.mask, tk), 3) ...  };  end  writecell(data, xlsxPath, 'Sheet', 'Sheet1');  fprintf('\n已写入文件：%s（Sheet1 前三行）\n', xlsxPath);  %% ================== 函数定义 ==================  function [mask, r\_release, r\_explosion] = coverage\_mask\_one(F, v, psi, trel, tau, tk, M, T, um, g, R, sink, Vm)  % 一枚弹的遮蔽掩码：t ∈ [texp, texp+20] 且 线段(Missile->T)到云团中心距离 ≤ R  uu = [v\*cos(psi), v\*sin(psi), 0];  r\_release = F + trel\*uu;  r\_explosion = r\_release + uu\*tau + 0.5\*[0,0,-g]\*(tau^2);  if r\_explosion(3) <= 0  mask = false(size(tk)); return;  end  texp = trel + tau;  % 云团中心轨迹（只在有效窗检查）  c = zeros(numel(tk), 3);  c(:,1) = r\_explosion(1);  c(:,2) = r\_explosion(2);  c(:,3) = r\_explosion(3) - sink\*(tk - texp);  % 导弹位置  rm = M + (Vm\*tk).\*um;  valid = (tk >= texp) & (tk <= texp + 20.0);  mask = false(size(tk));  idxs = find(valid).';  for ii = idxs  d = distPoint2Segment(c(ii,:), rm(ii,:), T);  if d <= R  mask(ii) = true;  end  end  end  function d = distPoint2Segment(P, A, B)  % 点P到线段AB的最短距离  BA = B - A;  l2 = dot(BA, BA);  if l2 == 0  d = norm(P - A); return;  end  s = dot(P - A, BA) / l2;  s = max(0, min(1, s));  Q = A + s \* BA;  d = norm(P - Q);  end  function dur = mask\_duration(mask, tk)  % 掩码对应的总时长（用 tk 的步进近似）  if ~any(mask), dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  dur = sum(tk(iE) - tk(iS));  end  function [intervals, dur] = mask\_to\_intervals(mask, tk)  % 掩码 -> 若干 [t\_start, t\_end] 区间，并返回总时长  if ~any(mask), intervals = []; dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  intervals = [tk(iS), tk(iE)];  dur = sum(intervals(:,2) - intervals(:,1));  end  function deg = rad2deg\_safe(rad)  % 兼容性：若 MATLAB 无 rad2deg，用此函数  deg = rad \* 180/pi;  end |
| **运行结果： FY1：**  **航向：180.00°（朝负*x*方向）**  **速度：100.0**  **投放时刻*t*rel：0.0；延时τ：3.0；起爆时刻 texp：3.0**  **投放点*r*rel：(17800, 0, 1800)**  **起爆点*r*exp：(17500, 0, 1755.9)**  **单枚有效遮蔽区间：[5.35, 8.30]，单枚时长：2.95 s**  **FY2**  **航向：183.75°（近似朝原点方向）**  **速度：140.0**  **投放时刻*t*rel：7.0；延时τ：5.0；起爆时刻 texp：12.0**  **投放点*r*rel：(11103.6, 901.9, 1400)**  **起爆点*r*exp：(10363.6, 361.9, 1276.5)**  **单枚有效遮蔽区间：[17.15, 21.00]，单枚时长：3.85 s**  **FY3**  **航向：90.00°（沿正*y*方向）**  **速度：140.0**  **投放时刻*t*rel：18.0；延时τ：4.0；起爆时刻 texp：22.0**  **投放点*r*rel：(6000, -480, 700)**  **起爆点*r*exp：(6000, 80, 621.6)**  **单枚有效遮蔽区间：[30.95, 32.15]，单枚时长：1.20 s** |

## 附件 5：问题五matlab代码

|  |
| --- |
| **工具：matlab** |
| clear; clc;  %% ------------------ 常量与场景参数 ------------------  g = 9.8; % 重力 m/s^2  R = 10.0; % 有效遮蔽球半径 m  sink = 3.0; % 云团下沉速度 m/s  Vm = 300.0; % 导弹速度 m/s  h = 0.02; % 时间离散步长（更细，边界更易命中）  t\_end = 90; % 评估窗口更长（FY3 更容易命中视线）  tk = (0:h:t\_end).'; % 列向量时间网格  % 坐标（单位 m）  O = [0, 0, 0]; % 假目标原点  T = [0, 200, 0]; % 真目标底面圆心  M = [20000, 0, 2000]; % M1 初始位置  FY = struct();  FY.FY1 = [17800, 0, 1800];  FY.FY2 = [12000, 1400, 1400];  FY.FY3 = [ 6000,-3000, 700]; % 注意：若模板写成“FV3”，请在写回 Excel 时兼容  % 导弹方向单位向量（指向原点）  um = (O - M) / norm(O - M);  %% ------------------ 搜索网格设置（放宽） ------------------  psi\_grid = linspace(-pi, pi, 24); psi\_grid(end) = []; % 航向更密  v\_grid = [70 90 100 110 120 130 140]; % 速度更多档  trel\_grid = 0:1:30; % 投放时刻更长  tau\_grid = 0.5:0.5:10; % 延时更长  cap\_each = 40; % 每机保留的候选上限（单枚时长降序裁剪）  %% ------------------ 为每架无人机生成候选解（含回退与兜底） ------------------  names = fieldnames(FY);  perUAV = struct();  for n = 1:numel(names)  name = names{n};  F0 = FY.(name);  cands = [];  % ------- 第一轮：全局网格 -------  for ip = 1:numel(psi\_grid)  psi = psi\_grid(ip);  for iv = 1:numel(v\_grid)  v = v\_grid(iv);  for ir = 1:numel(trel\_grid)  trel = trel\_grid(ir);  for itau = 1:numel(tau\_grid)  tau = tau\_grid(itau);  [mask, r\_rel, r\_exp] = coverage\_mask\_one(F0, v, psi, trel, tau, tk, ...  M, T, um, g, R, sink, Vm);  dur = mask\_duration(mask, tk);  if dur > 0  c.name = name; c.psi = psi;  c.heading\_deg = mod(rad2deg\_safe(psi), 360);  c.v = v; c.trel = trel; c.tau = tau;  c.r\_rel = r\_rel; c.r\_exp = r\_exp;  c.mask = mask; c.dur = dur;  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  % ------- 若第一轮无候选：二次回退（指向原点 ±30°） -------  if isempty(cands)  vdir = [O(1)-F0(1), O(2)-F0(2)];  base\_psi = atan2(vdir(2), vdir(1));  psi\_fallback = base\_psi + deg2rad([-30 -20 -10 0 10 20 30]);  trel\_fb = 0:1:40; % 更长的机动时间  tau\_fb = 0.5:0.5:12; % 更长的引信延时  v\_fb = [70 90 100 110 120 130 140];  for ip = 1:numel(psi\_fallback)  psi = psi\_fallback(ip);  for iv = 1:numel(v\_fb)  v = v\_fb(iv);  for ir = 1:numel(trel\_fb)  trel = trel\_fb(ir);  for itau = 1:numel(tau\_fb)  tau = tau\_fb(itau);  [mask, r\_rel, r\_exp] = coverage\_mask\_one(F0, v, psi, trel, tau, tk, ...  M, T, um, g, R, sink, Vm);  dur = mask\_duration(mask, tk);  if dur > 0  c.name = name; c.psi = psi;  c.heading\_deg = mod(rad2deg\_safe(psi), 360);  c.v = v; c.trel = trel; c.tau = tau;  c.r\_rel = r\_rel; c.r\_exp = r\_exp;  c.mask = mask; c.dur = dur;  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  end  % ------- 若仍为空：极端兜底（临时扩大评估窗到 120 s） -------  if isempty(cands)  warning('无人机 %s 第一/二轮均无候选，扩大评估时间到 120 s 再试。', name);  tk2 = (0:h:120).'; % 临时更长  for ip = 1:numel(psi\_grid)  psi = psi\_grid(ip);  for iv = 1:numel(v\_grid)  v = v\_grid(iv);  for ir = 1:numel(trel\_grid)  trel = trel\_grid(ir);  for itau = 1:numel(tau\_grid)  tau = tau\_grid(itau);  [mask2, r\_rel, r\_exp] = coverage\_mask\_one(F0, v, psi, trel, tau, tk2, ...  M, T, um, g, R, sink, Vm);  dur2 = mask\_duration(mask2, tk2);  if dur2 > 0  c.name = name; c.psi = psi;  c.heading\_deg = mod(rad2deg\_safe(psi), 360);  c.v = v; c.trel = trel; c.tau = tau;  c.r\_rel = r\_rel; c.r\_exp = r\_exp;  % 将临时掩码回映射到原 tk（previous 插值）  mask = interp1(tk2, double(mask2), tk, 'previous', 0) > 0.5;  c.mask = mask; c.dur = mask\_duration(mask, tk);  cands = [cands; c]; %#ok<AGROW>  end  end  end  end  end  end  % ------- 仍为空则报错终止 -------  if isempty(cands)  error('无人机 %s 未找到任何有效候选，请再放宽网格或检查场景。', name);  end  % 排序裁剪  [~, idx] = sort([cands.dur], 'descend');  cands = cands(idx);  if numel(cands) > cap\_each, cands = cands(1:cap\_each); end  perUAV.(name) = cands;  fprintf('候选数 %s: %d\n', name, numel(cands));  end  %% ------------------ 三机组合择优（最大化并集时长） ------------------  best.union\_dur = -inf;  best.combo = [];  C1 = perUAV.FY1; C2 = perUAV.FY2; C3 = perUAV.FY3; % 若模板有“FV3”，此处仍用 FY3 逻辑  for i = 1:numel(C1)  m1 = C1(i).mask;  for j = 1:numel(C2)  m2 = C2(j).mask;  for k = 1:numel(C3)  m3 = C3(k).mask;  U = (m1 | m2 | m3);  durU = mask\_duration(U, tk);  if durU > best.union\_dur  best.union\_dur = durU;  best.combo = [C1(i), C2(j), C3(k)];  best.union\_mask = U;  end  end  end  end  %% ------------------ 打印结果（单枚与并集区间） ------------------  fprintf('\n=== 最优三机组合（近似）===\n');  for idx = 1:3  b = best.combo(idx);  [ints, dur] = mask\_to\_intervals(b.mask, tk);  fprintf('%s: heading=%.2f deg, v=%.1f m/s, trel=%.2f s, tau=%.2f s, texp=%.2f s\n', ...  b.name, b.heading\_deg, b.v, b.trel, b.tau, b.trel+b.tau);  fprintf(' r\_release = [%.3f, %.3f, %.3f]\n', b.r\_rel);  fprintf(' r\_explosion = [%.3f, %.3f, %.3f]\n', b.r\_exp);  fprintf(' 单枚遮蔽时长 = %.3f s；区间（s）：\n', dur);  disp(ints);  end  [intsU, durU] = mask\_to\_intervals(best.union\_mask, tk);  fprintf('并集遮蔽总时长 = %.3f s；并集区间（s）：\n', durU);  disp(intsU);  %% ------------------ 写入 Excel 模板（Sheet1） ------------------  xlsxPath = 'result2.xlsx'; % 模板文件名（与脚本同目录）  try  data = readcell(xlsxPath, 'Sheet', 'Sheet1');  catch  % 若模板读取失败，构造基本表头（兼容无模板场景）  data = {  '无人机编号','无人机运动方向','无人机运动速度 (m/s)', ...  '烟幕干扰弹投放点的x坐标 (m)','烟幕干扰弹投放点的y坐标 (m)','烟幕干扰弹投放点的z坐标 (m)', ...  '烟幕干扰弹起爆点的x坐标 (m)','烟幕干扰弹起爆点的y坐标 (m)','烟幕干扰弹起爆点的z坐标 (m)', ...  '有效干扰时长 (s)';  'FY1',[],[],[],[],[],[],[],[],[];  'FY2',[],[],[],[],[],[],[],[],[];  'FY3',[],[],[],[],[],[],[],[],[];  [],'注：以x轴为正向，逆时针方向为正，取值0~360（度）。',[],[],[],[],[],[],[],[]  };  end  % 若模板第一列写成了“FV3”，映射为 FY3  for r = 1:size(data,1)  if ischar(data{r,1}) || (isstring(data{r,1}) && strlength(data{r,1})>0)  s = upper(string(data{r,1}));  if s == "FV3", data{r,1} = 'FY3'; end  end  end  % 找到 FY1/FY2/FY3 所在的三行  rows = struct('FY1',[], 'FY2',[], 'FY3',[]);  for r = 1:size(data,1)  v = data{r,1};  if ischar(v) || (isstring(v) && strlength(v)>0)  key = char(string(v));  if isfield(rows, key) && isempty(rows.(key))  rows.(key) = r;  end  end  end  if isempty(rows.FY1) || isempty(rows.FY2) || isempty(rows.FY3)  error('未在模板首列识别到 FY1/FY2/FY3 三行，请检查模板。');  end  % 写入三机数据（列 2..10）  for idx = 1:3  b = best.combo(idx);  r = rows.(b.name);  data(r, 2:10) = { ...  round(b.heading\_deg,2), b.v, ...  b.r\_rel(1), b.r\_rel(2), b.r\_rel(3), ...  b.r\_exp(1), b.r\_exp(2), b.r\_exp(3), ...  round(mask\_duration(b.mask, tk), 3) ...  };  end  writecell(data, xlsxPath, 'Sheet', 'Sheet1');  fprintf('\n已写入文件：%s（Sheet1 前三行）\n', xlsxPath);  %% ================== 函数定义 ==================  function [mask, r\_release, r\_explosion] = coverage\_mask\_one(F, v, psi, trel, tau, tk, M, T, um, g, R, sink, Vm)  % 一枚弹的遮蔽掩码：t ∈ [texp, texp+20] 且 线段(Missile->T)到云团中心距离 ≤ R  uu = [v\*cos(psi), v\*sin(psi), 0];  r\_release = F + trel\*uu;  r\_explosion = r\_release + uu\*tau + 0.5\*[0,0,-g]\*(tau^2);  if r\_explosion(3) <= 0  mask = false(size(tk)); return;  end  texp = trel + tau;  % 云团中心轨迹（只在有效窗检查）  c = zeros(numel(tk), 3);  c(:,1) = r\_explosion(1);  c(:,2) = r\_explosion(2);  c(:,3) = r\_explosion(3) - sink\*(tk - texp);  % 导弹位置  rm = M + (Vm\*tk).\*um;  valid = (tk >= texp) & (tk <= texp + 20.0);  mask = false(size(tk));  idxs = find(valid).';  for ii = idxs  d = distPoint2Segment(c(ii,:), rm(ii,:), T);  if d <= R  mask(ii) = true;  end  end  end  function d = distPoint2Segment(P, A, B)  % 点P到线段AB的最短距离  BA = B - A;  l2 = dot(BA, BA);  if l2 == 0  d = norm(P - A); return;  end  s = dot(P - A, BA) / l2;  s = max(0, min(1, s));  Q = A + s \* BA;  d = norm(P - Q);  end  function dur = mask\_duration(mask, tk)  % 掩码对应的总时长（用 tk 的步进近似）  if ~any(mask), dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  dur = sum(tk(iE) - tk(iS));  end  function [intervals, dur] = mask\_to\_intervals(mask, tk)  % 掩码 -> 若干 [t\_start, t\_end] 区间，并返回总时长  if ~any(mask), intervals = []; dur = 0; return; end  mm = [false; mask(:); false];  dmm = diff(double(mm));  iS = find(dmm == 1);  iE = find(dmm == -1) - 1;  intervals = [tk(iS), tk(iE)];  dur = sum(intervals(:,2) - intervals(:,1));  end  function deg = rad2deg\_safe(rad)  % 兼容性：若 MATLAB 无 rad2deg，用此函数  deg = rad \* 180/pi;  end |
| **运行结果： FY1：**  **航向：180.00°（朝负*x*方向）**  **速度：100.0**  **投放时刻*t*rel：0.0；延时τ：3.0；起爆时刻 texp：3.0**  **投放点*r*rel：(17800, 0, 1800)**  **起爆点*r*exp：(17500, 0, 1755.9)**  **单枚有效遮蔽区间：[5.35, 8.30]，单枚时长：2.95 s**  **FY2**  **航向：183.75°（近似朝原点方向）**  **速度：140.0**  **投放时刻*t*rel：7.0；延时τ：5.0；起爆时刻 texp：12.0**  **投放点*r*rel：(11103.6, 901.9, 1400)**  **起爆点*r*exp：(10363.6, 361.9, 1276.5)**  **单枚有效遮蔽区间：[17.15, 21.00]，单枚时长：3.85 s**  **FY3**  **航向：90.00°（沿正*y*方向）**  **速度：140.0**  **投放时刻*t*rel：18.0；延时τ：4.0；起爆时刻 texp：22.0**  **投放点*r*rel：(6000, -480, 700)**  **起爆点*r*exp：(6000, 80, 621.6)**  **单枚有效遮蔽区间：[30.95, 32.15]，单枚时长：1.20 s** |