



中国研究生创新实践系列大赛

“华为杯”第十七届中国研究生数学建模竞赛

学　　校	空军工程大学
参赛队号	20910240043
队员姓名	1.吴傲
	2.侯岳奇
	3.王玉冰

中国研究生创新实践系列大赛

“华为杯”第十七届中国研究生

数学建模竞赛

题 目　　　无人机集群协同对抗

摘　　要：

本文针对红、蓝双方无人机集群的协同对抗问题，分析了红蓝双方无人机的运动学约束、突防与拦截的区域约束、突防和拦截成功判定规则；建立了无人机运动模型、红方无人机集群布设模型和基于截击三角形的单机空中拦截模型，**针对蓝方的突防问题，**以最大化突防成功概率和最小化突防时间为目标，提出了垂直突防、斜向突防和机动突防三种蓝方突防模型，理论推导证明了突防成功的充分必要条件，求解出对应的垂直突防区和斜向突防区，给出了蓝方无人机的最优突防策略；**针对红方的拦截问题，**以拦截概率最大化为目标，将拦截概率最大化问题转化为突防区最小化及红方集群的最优布设策略。理论推导并证明了拦截成功的充分必要条件，求解了圆周构型下的红方拦截区域和拦截距离，基于转弯截击法提出了红方无人机集群的最优拦截策略。**针对红、蓝双方的动态博弈问题，**基于 Stackelberg 博弈提出蓝方最优突防策略，以及红方无人机集群的防守型策略和围捕型策略，分别体现了无人机集群基于规模效应和基于结构效应的能力涌现机理，从而得出红方最优拦截策略和蓝方突防策略。具体建模求解过程及结果如下：

针对问题一：首先，分析了无人机集群协同对抗的区域约束、运动学约束、突防成功规则和拦截成功规则；其次，建立了基于截击三角形的空中拦截模型，求解了圆周构型下的红方拦截区域和拦截距离；然后，按照蓝方机动策略从特殊到一般的思路，从垂直突防推广到斜向突防再到机动突防，严格推导并证明了垂直突防和斜向突防情况下“无论红方如何拦截，蓝方都能突防成功”的充分必要条件，最后依据该条件求解出蓝方能够突防成功的垂直突防区和斜向突防区，边界如式 3.36 所示，范围如图 3.14 所示。蓝方自由突防可以综合利用垂直突防推广到斜向突防，具有一般性。

针对问题二：在问题一的突防区解析解基础上，首先以蓝方突防区面积最小化为准则，严格证明得到红方集群最优布设策略为 $DG_1 = CG_2 = M/4$，$G_1G_2 = M/2$，在通道带宽一定时可求解出红方集群最优布设位置；然后通过改变通道带宽，计算出在红方集群最优布设情况下不同通道带宽对应的突防区范围，直至该突防区包含蓝方无人机初始位置点，此时的通道带宽即为 M 的下限 $M_{\min}$ =161.428km，此时蓝方的时间最短突防策略为垂直突防。

针对问题三：首先，针对协同对抗过程决策空间大、不易用解析法表征的问题，搭建了基于 MATLAB 的多智能体协同对抗仿真系统，基于数量可扩展、策略可替换的仿真思想，建立了集群类和多智能体类，构建了红蓝双方机动策略库，通过调用不同的策略库实现面向任务需求的策略推演；其次，从双方博弈过程出发，建立了红方拦截线模型和蓝方突防机动区间模型，提出了红方转弯截击法；然后，基于 Stackelberg 博弈思想，针对红方无人机集群拦截问题设计了防守型策略和围捕型策略；针对防守型策略，从规模效应出发给出了“|”字构型和群集构型，运用仿真系统推演得出最优防守构型为“|”字构型，并给出了不同通道带宽下的突防成功概率，防守型策略下求得 $M_{\max} = 5km$；针对围捕型策略，

提出了基于截击三角形的转弯截击法，建立了追踪拦截控制模型，从结构效应出发设计了多集群多角度动态围捕策略；最后运用仿真系统基于拦截概率最大准则推演得出红方集群最优布设方案，通过 20000 次蒙特卡洛仿真逼近突防成功概率为 0 的情况，求得围捕型策略下 $M_{\max}=21.04km$ 。

针对问题四：面向红蓝双方的 Stackelberg 博弈协同对抗需求，分析了红蓝双方各自的优势力量，分别设计了红方拦截的**自适应围捕策略**和蓝方突防的**“直线接近—临界机动”策略**。针对红方拦截特点和需求，设计了基于运载机速度优势的第二波次最优布设方案，提出了基于动态联盟机制的自适应任务分配方法，采用基于转弯截击法的追踪拦截控制，实现红方动态联盟对蓝方无人机的围捕拦截。针对蓝方突防特点和需求，设计了基于蓝方速度优势的“直线接近—临界机动”突防策略，将蓝方速度优势转化为空间优势。

关键词：截击三角形；最优布设；转弯截击法；Stackelberg 博弈；动态联盟组建；能力涌现；协同对抗

目录

1 问题重述

1.1 问题背景

新一代人工智能技术和自主技术快速走向战场，将催生新型作战力量，颠覆传统战争模式，未来战争必将是智能化战争。无人机集群作战作为智能作战的重要形式，正在崭露头角。通过多架无人机协同侦察、协同探测、协同跟踪、协同攻击、协同拦截等，共同完成较复杂的作战任务。

现考虑红、蓝双方的无人机集群在平面区域内的协同对抗问题。蓝方作为进攻方，希望突破红方无人机的拦截，成功抵达目的地遂行军事行动；红方则希望在给定的区域内完成对蓝方无人机的拦截，阻止蓝方的突防。本赛题讨论的对抗区域约定为图 1.1 所示的矩形区域 $ABCD$，攻击纵深即 BC 之间的距离为 $L=50km$，蓝方无人机的飞行轨迹不能越过 AD、BC 两边，即考虑的是攻击通道（突防走廊）带宽有一个限定约束的情形，通道带宽即 AB 之间的距离记为 M。蓝方无人机的速度为 $V_E=250m/s$，最小转弯半径为 $R_E=500m$；红方无人机的速度为 $V_P=200m/s$，最小转弯半径为 $R_P=350m$；红蓝双方无人机的速度保持不变，运动的方向可根据机动策略的需要随时改变，但受转弯半径的限制。

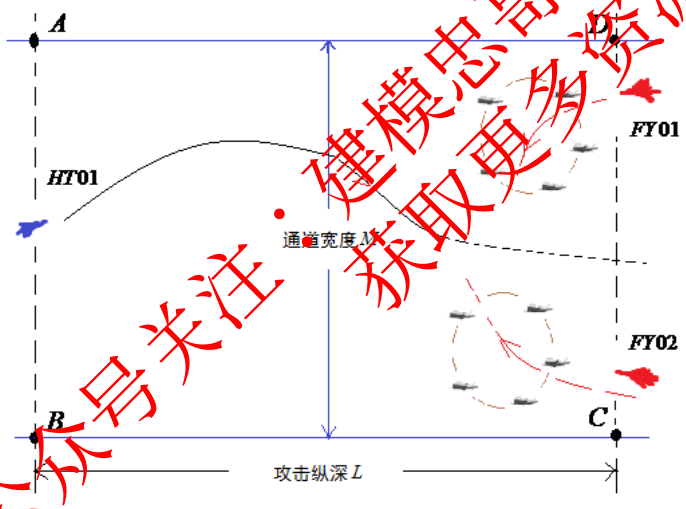


图 1.1 红方两个无人机集群拦截一架蓝方突防无人机示意图

1.2 需要解决的问题

问题 1：对抗伊始红方 2 个无人机集群的圆周中心分别位于 $G1$ 和 $G2$，圆周半径为 $100m$，其中 $DG1=20km$，$G1G2=30km$，$CG2=20km$。要求分析蓝方无人机处于矩形区域 $ABCD$ 内哪些位置时，无论红方无人机采用什么样的追击策略，蓝方无人机总能采用合适的策略以躲避红方的拦截，实现成功突防；讨论蓝方无人机相应的最优突防策略。

问题 2：对抗伊始蓝方突防无人机位于边界 AB 的中心点，红方 2 个无人机集群的圆周中心分别位于 $G1$ 和 $G2$，圆周半径为 $100m$，其中 $G1$ 和 $G2$ 位于边界 CD 上，具体位置根据需要确定。要求分析是否存在一个通道带宽 M 的下限 $M_{\min}$，当实际通道带宽 M 比 $M_{\min}$ 大时，蓝方无人机一定能突破红方无人机集群的拦截；要求给出此种情形下蓝方无人机时间最短的突防策略。

问题 3：红方每架运载机可分两个波次共发射 10 架无人机，组成两个无人机集群遂行拦截任务，每个无人机集群的无人机数量不少于 3 架。每一波次发射时无人机集群初始队形如为圆周构型，运载机与圆周中心的距离为 $2km$，随后无人机集群的队形可根据需要调整，但要求满足相应的间距约束。对抗伊始，蓝方无人机位于边界 AB 的中心，通道带宽 $M=70km$；红方两架运载机分别位于边界 CD 上 $G1$ 点和 $G2$ 点，并开始发射第一波次的无

人机集群，运载机和无人机集群中心具体位置根据需要确定。运载机第二波次发射无人机集群时，必须保证运载机与第一波次发射的无人机集群满足间距上的约束。要求讨论红方两架运载机两个波次发射的无人机数量、每架运载机第二波次发射的时刻和位置以及第二波次发射的无人机集群的中心位置，以实现最优的拦截效果；进一步具体建模分析是否存在一个通道带宽 M 的上限 $M_{\max}$，当实际通道带宽 M 小于 $M_{\max}$ 时，无论蓝方无人机采用什么样的突防策略，红方无人机集群均存在相应的拦截策略，在区域 $ABCD$ 内成功阻止蓝方无人机的突防。

问题 4：通道带宽 $M=100km$，蓝方 3 架突防无人机组成突防集群从矩形边界 AB 一侧开始突防（任 2 架突防无人机的间距需大于 $30m$），红方 5 架运载机各携带 10 架无人机，从边界 CD 一侧同时开始遂行协同拦截任务。红方每架运载机分两个波次发射无人机，分别组成两个无人机集群，每个集群的无人机数量不少于 3 架；每架运载机第一波次发射无人机的时刻为初始对抗时刻，与所属无人机集群几何构型圆周中心的距离为 $2km$。红方运载机初始位置、红方运载机发射的第一个波次的无人机集群中心位置、红方运载机发射第二波次无人机集群的时刻和位置、第二波次发射的无人机集群中心位置、两个波次无人机数量以及蓝方突防无人机初始位置根据需要确定。蓝方希望尽可能多的无人机突防成功，红方则希望成功拦截尽可能多的蓝方无人机。要求讨论红方最优拦截策略和蓝方最优突防策略。

2 合理假设与符号系统

2.1 模型假设

1）无人机在飞行过程中当作质点；

2）忽略运载机布设无人机集群的时间，视为运载机瞬间布设好无人机集群，且满足相关约束条件；

3）假设红蓝双方的无人机均在同一高度上机动飞行，本题只讨论平面上的红蓝双方对抗问题。

*注：以上为全文的通用假设，解题过程中还涉及一些对应各个问题的具体假设，将在各章中予以说明。

2.2 符号说明

符号	符号说明
V_E	蓝方飞机速度
V_P	红方无人机速度
L	两点间的距离
D	拦截距离
$d_{\min}$	每架无人机与其他无人机的最小距离
$d_{\max}$	每架无人机与其他无人机的最大距离

3 问题一的建模与求解

3.1 问题分析

题目中指出“无论红方无人机采用什么样的追逃策略，蓝方无人机总能……，实现成功突防”。在理论研究中，我们无法通过穷举红方的追逃策略来判断蓝方是否能够成功突防。因此，在解决本问题时，我们将不从追逃策略入手来研究，而是从拦截点存在性入手进行研究。

拦截点存在性指的是：给定初始时刻和红蓝双方无人机的初始位置，在矩形 ABCD 中存在一个拦截点使得红方无人机、蓝方无人机、拦截点三者构成截击三角形。如果在矩形区域 ABCD 中，对于红方无人机无可行拦截点，那么无论采取何种追逃策略都无法实现拦截，即蓝方必定成功突防；反之，若存在一个可达的拦截点，则在红方策略足够“智能”的情况下，必定能够实现拦截。

考虑一种极限情况：蓝方无人机飞行航线固定，且航线信息对于红方无人机完全已知。在上述情况下，红方无人机无需采用复杂的追逃策略，只需根据蓝方无人机的航线信息，采用直线拦截的方式前出拦截即可。若通过直线拦截方法，无法构成截击三角形或截击点位于矩形 ABCD 外，则红方无论采取何种追逃策略都无法实现拦截。

针对问题一，本节基于如下假设进行求解：完成初始布设后，无人机集群在圆上均匀分布的位置和初始航向可以根据需要指定。

3.2 模型建立

3.2.1 任务场景描述

如图 3.1 所示，是无人机集群协同对抗的任务场景。作战区域为 ABCD 围成的矩形区域，红方无人机以圆周形式布设，两个集群的中心分别位于点 G1，G2。

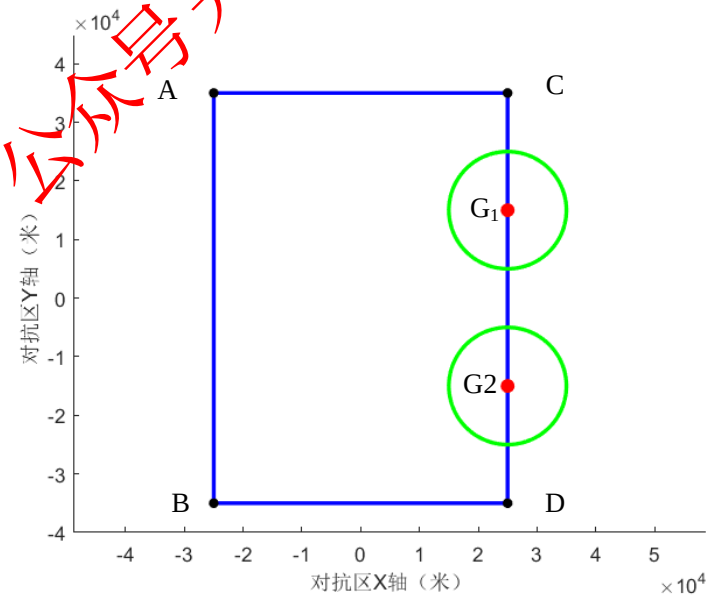


图 3.1 无人机集群协同对抗场景描述

3.2.2 约束条件

根据题目要求，在对抗过程中蓝方和红方无人机集群都要遵循一定的约束条件，包括任务区域约束、无人机运动学约束和集群布设约束，这些约束成为策略制定和航迹规划的基本前提。具体如下：

3.2.2.1 蓝方无人机集群的约束条件

（1）蓝方任务区域约束为：

$$-\frac{L}{2} \le x_F \le \frac{L}{2} \tag{3.1}$$

$$-\frac{M}{2} \le y_F \le \frac{M}{2} \tag{3.2}$$

（2）蓝方运动学约束为：

$$V_E = 250m/s \tag{3.3}$$

$$R_E = 500m \tag{3.4}$$

（3）蓝方的突防成功规则为：

$$t_E \le 360s \tag{3.5}$$

$$d_{EP_iP_j} \ge 300m \tag{3.6}$$

3.2.2.2 红方无人机集群的约束条件

（1）红方运动学约束为：

$$V_P = 200m/s \tag{3.7}$$

$$R_P = 350m \tag{3.8}$$

（2）红方集群布设约束为：要求红方任何两架无人机的间距需大于$30m$，即：

$$\forall P_i, P_j, d_{P_iP_j} > 30m \tag{3.9}$$

每一架无人机与本集群中至少两架无人机的距离不超过$200m$：

$$\forall P_i, d_{P_iP_j} \cap d_{P_iP_k} \le 200 \quad i \ne j \ne k \tag{3.10}$$

红方运载机与所属无人机集群中至少一架无人机的距离不超过$10km$：

$$\exists P_i, d_{FYP_i} < 10km \tag{3.11}$$

与任何一架无人机的距离需大于$100m$：

$$\forall P_i, d_{FYP_i} > 100m \tag{3.12}$$

红方运载机与蓝方突防无人机的距离需大于$5km$：

$$d_{EFY} > 5km \tag{3.13}$$

（3）红方的拦截成功规则为：

$$d_{EP_iP_j} < 300m \tag{3.14}$$

3.2.3 基于截击三角形的空中拦截模型

空中拦截过程是一个动态、不断变化的过程，因此，需要研究拦截过程中敌机和我机动态变化运动规律。由于敌我双方动态博弈的场景较为复杂，需要先从静态场景入手，研究敌机匀速直线飞行情况下的拦截策略，再由静到动，进一步研究敌机机动情况下的拦截

策略。首先，考虑敌机匀速直线运动的情况，拦截的方式主要分为追踪拦截和直线拦截两大类。

追踪拦截起源于动物间的追杀过程，要求每一瞬间攻击机的速度矢量始终对准目标进行机动飞行，又称为“狗追兔”攻击。追踪拦截需要实时机动，且在某些极端情况下进行大过载机动，对攻击机的转弯半径具有强约束。

直线拦截是一种直线攻击方式，即以一定的规则与目标相遇，从而避免攻击机的大过载机动。直线拦截是攻击机以恒定航向直线运动方式指向与目标的相遇点，可以达到接敌距离最短、所需接敌时间最少的目的。直线拦截攻击方式对攻击机机动性能要求远比追踪拦截要低，且时间短、距离短，因此采用直线拦截攻击方式作为本文拦截方式。

在拦截攻击中，达成的攻击条件可以用速度矢量三角形来描述，即：在任意时刻，我机速度矢量、敌机速度矢量、拦截点三者均保持一种空间三角形态势，称这个速度矢量三角形为截击三角形。

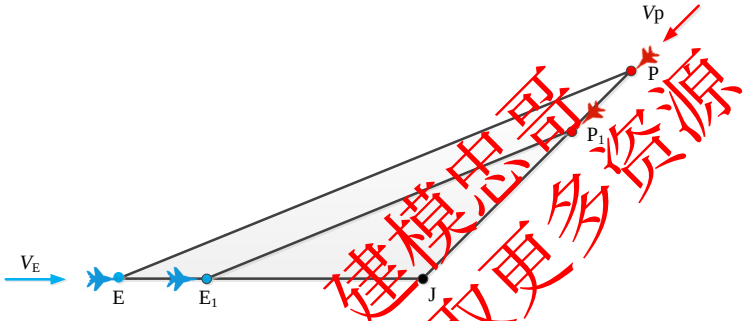


图 3.2 截击三角形示意图

截击三角形如图 3.2 所示，E 点为敌机，沿 EJ 方向匀速飞行，P 点为我机，沿 PJ 方向匀速飞行，最终我机与敌机在拦截点 J 相遇。在上述情况下，截击三角形具有以下性质：

（1）我机与敌机飞行距离比等于速度比

在截击三角形中，敌机从 E 点飞到拦截点 J 的飞行距离为：

$$L_{JE}=V_E t \tag{3.15}$$

其中，t 为截击飞行时间。我机从 P 点飞到拦截点 J 的飞行距离为：

$$L_{JP}=V_P t \tag{3.16}$$

因此，有：

$$\frac{L_{JP}}{L_{JE}}=\frac{V_P}{V_E}=m \tag{3.17}$$

其中，m 为我机与敌机的速度比。

（2）拦截过程中我机与敌机始终保持截击三角形

飞行一段时间 t_1 后，观察△E_1P_1J，可以得到：

$$\frac{L_{JP_1}}{L_{JE_1}}=\frac{L_{JP}-L_{PP_1}}{L_{JE}-L_{EE_1}}=\frac{V_P t-V_P t_1}{V_E t-V_E t_1}=\frac{V_P}{V_E}=m \tag{3.18}$$

可以看出，拦截过程中我机与敌机剩余飞行距离比始终等于速度比，始终保持截击三角形态势。

上述截击三角形是将我机视为一个恒速弹丸与目标在截击点相遇，是我机不使用武器

的特殊情况。但是在实际应用中，通常采用武器进行攻击，存在一个攻击距离，即本题中给出的“当蓝方突防无人机与红方至少 2 架无人机的距离均小于 300*m* 时，就认为红方成功拦截了蓝方突防无人机”。因此，这里需要引入拦截距离 *D*，图 3.3 给出了考虑拦截距离的截击三角形示意图。

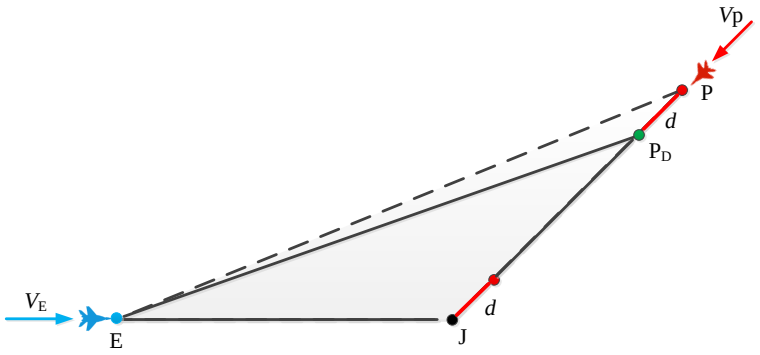


图 3.3 考虑拦截距离的截击三角形示意图

在考虑拦截距离的截击三角形中，我机与敌机相比始终落后一个拦截距离。当敌机到达截击点时，我机在敌机后方，距离恰好等于拦截距离，即：

$$\frac{L_{JP}-d}{L_{JE}}=\frac{V_P}{V_E}=m \tag{3.19}$$

可以看到，考虑拦截距离的截击三角形与普通的截击三角形原理相同，只需将攻击无人机的位置前推一个拦截距离即可转化为普通截击三角形。因此，只需将我机所在位置 P 等效为 P_D 即可，$\triangle EP_DJ$ 仍然构成截击三角形。

截击三角形是本文开展研究的重要支撑，后文提出的拦截策略都是基于截击三角形推广得到的。

3.2.4 基于圆周构型的拦截区域

当无人机集群按照 5 机编队形成圆周构型时，5 架无人机均匀分布在圆周半径 r 为 100m 的圆上，根据几何关系，任意相邻两架无人机之间的夹角 θ 为 $2\pi/5$，每架无人机与其他无人机的最短距离 d_{min} 为 117.6 米，最长距离 d_{max} 为 190.2 米，因此该构型满足无人机编队飞行的构型约束条件。

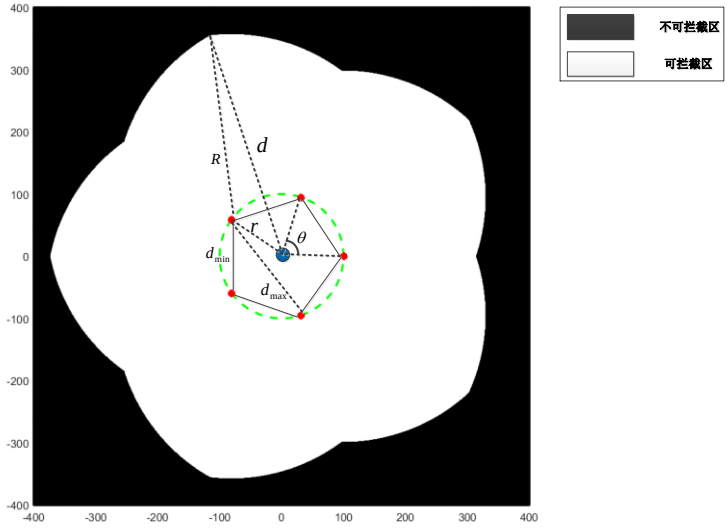


图 3.4 圆周构型拦截区域示意图

此构型下，拦截区域示意图如图 3.4 所示。以圆周构型的圆心为坐标原点构建坐标系，在边长为 400m 的正方形区域内，白色区域为可拦截区，黑色区域为不可拦截区，处于拦截区内的点与红方至少两架无人机的距离小于 $R=300m$。定义拦截距离为可拦截区到原点的最远距离，用 d 表示，d 的计算公式为

$$d = r \times cos(\theta/2) + \sqrt{R^2 - (r \times sin(\theta/2))^2} \tag{3.20}$$

经计算得拦截距离 $d=375m$。拦截距离代表了红方无人机集群处于固定构型下所能拦截的蓝方无人机的最大范围。

3.2.5 突防区模型

基于拦截点可达性的判断思想，借助截击三角形方法，我们通过理论推导得到了蓝方无人机必定实现突防的初始布设区域及其突防策略。通过分析，可以将矩形 ABCD 划分为垂直突防区、斜向突防区和策略突防区，下面给出几种突防区的定义。

（一）垂直突防区

当蓝方无人机的突防航线为一条与 CD 垂直的线段时，蓝方无人机的突防路径和突防时间最短，最容易造成红方无人机无可行拦截点。垂直突防区用 Sv 表示，如图 3.5 所示。其中紫色区域代表垂直突防区，蓝色三角形代表截击三角形。

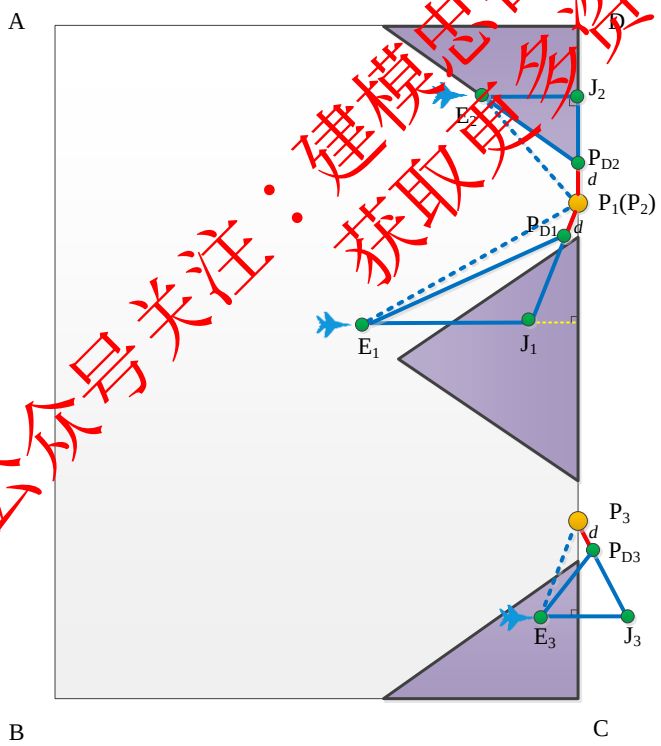


图 3.5 垂直突防区示意图

由于红方无人机集群可获取全局态势信息和蓝方无人机的突防轨迹，因此在初始时刻即可调整初始航向，期望在 ABCD 区域内达成截击三角形，并实现截击。当蓝方无人机从垂直突防区选择垂直航线突防时，由于飞行平台和几何关系的限制，必定导致蓝方无人机从某些区域出发无法形成截击三角形或截击点在区域外，此时无论红方无人机采用什么样的追击策略，都不能成功拦截，蓝方无人机成功突防。

图 3.5 给出了蓝方无人机选择垂直航线突防的三种情况，起始位置分别为 E1、E2、E3。

情况一：起始位置位于垂直突防区外

当从 E_1 突防时，起始位置位于垂直突防区外，红方无人机集群从 P_1 点出发，形成的

截击三角形为△$P_{D1}J_1E_1$，截击点为 J_1，此时截击点位于 ABCD 区域内，意味着蓝方无人机突防失败。

情况二：起始位置位于垂直突防区边界

当从 E_2 突防时，起始位置位于垂直突防区边界上，红方无人机集群从 P_2 点出发，形成的截击三角形为△$P_{D2}J_2E_2$，截击点为 J_2，此时截击点恰好落在边界 CD 上，意味着蓝方无人机突防失败。

情况三：起始位置位于垂直突防区内

当从 E_3 突防时，起始位置位于垂直突防区内，红方无人机集群从 P_3 点出发，形成的截击三角形为△$P_{D3}J_3E_3$，截击点为 J_3，此时截击点位于 ABCD 区域外，意味着蓝方无人机突防成功。

（二）斜向突防区

斜向突防指的是蓝方无人机的突防航线为一条与 CD 成一定夹角的线段。与垂直突防相比，斜向突防虽然不是时间和距离最优的突防路径，但是也能够造成红方无人机无法拦截。若蓝方无人机初始位置位于斜向突防区中，在某些特定航向情况下，可以实现成功突防。可以简单推论，前文提出的**垂直突防区是斜向突防区的一种特殊情况，且垂直突防区是完全包含于斜向突防区之内的**，后文的结果同样也印证了这一点。

在矩形区域 ABCD 中任取一点 E 作为蓝方无人机初始位置，以 E 点为航线起点，可以得到无穷多条斜向突防航线。我们难以穷举所有初始位置下所有斜向突防航线，因此，我们从斜向突防的充分和必要条件入手，逐步缩小斜向突防区范围，最终得到实现斜向突防的斜向突防区。参考垂直突防区的建模和求解方法，对斜向突防区进行建模和求解的思路如下：

（1）通过截击三角形方法，给出蓝方无人机能够实现斜向突防的必要条件，并给出必要条件下的斜向突防区，记为 S_{tb}。

（2）通过拦截点存在性判断方法，给出蓝方无人机能够实现斜向突防的充分条件。

（3）对必要条件下的斜向突防区 S_{tb} 进行分割，根据斜向突防的充分条件，求解得到最终的斜向突防区 S_t。

通过简单分析可以知道，矩形 ABCD 中的部分区域显然无法实现斜向突防。若将矩形 ABCD 作为求解空间，则会大大增加工作量和推导难度。因此，我们首先给出蓝方无人机能够实现斜向突防的必要条件，进而大大缩小了下一步求解的空间范围。

3.3 模型解算及结果分析

下面借助截击三角形方法分别对垂直突防区和斜向突防区进行求解。

3.3.1 垂直突防区

（1）垂直突防区

定理 1（垂直突防充分必要条件） 给定垂直突防区 S_v 的约束条件：

$$
\begin{aligned}
&S_v = S_{v1} \cup S_{v2} \cup S_{v3} \cup S_{v4} \\
&S_{v1} = \left\{ (x,y) \middle| x \le 25, 0 \le y < \frac{4}{5}x - (d+5) \right\} \\
&S_{v2} = \left\{ (x,y) \middle| x \le 25, -\frac{4}{5}x + (d+5) < y \le 0 \right\} \\
&S_{v3} = \left\{ (x,y) \middle| x \le 25, -\frac{4}{5}x + (d+35) < y \le 35 \right\} \\
&S_{v4} = \left\{ (x,y) \middle| x \le 25, -35 < y < \frac{4}{5}x - (d+35) \right\}
\end{aligned}
\tag{3.21}
$$

当蓝方无人机初始位置位于垂直突防区 S_v 中，且其航线为一条与 X 轴平行的线段时，无论红方无人机采用什么样的追击策略，蓝方无人机总能采用合适的策略以躲避红方的拦截，实现成功突防。

证明：为便于证明过程的描述，给出垂直突防区 S_v 的示意图，如图 3.6 所示。

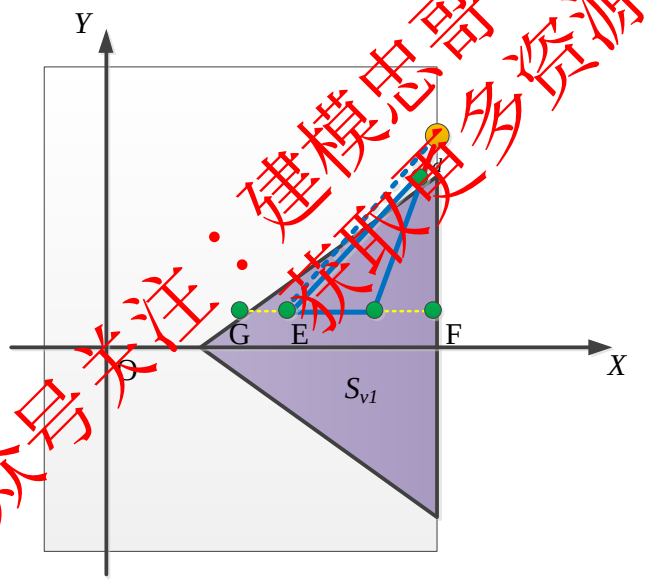


图 3.6 垂直突防区示意图

记蓝方无人机初始位置坐标为 $E(x_E, y_E)$，突防航线末端点为 $F(25, y_E)$。红方无人机集群所在位置为 $P(25,15)$。采用反证法给出证明：

对于 $\forall E \in S_{v1}$，假设在突防航线 EF 上存在一点 J 为拦截点，则△EPJ 构成截击三角形。根据考虑拦截距离的截击三角形的性质，有：

$$
\frac{L_{JP} - d}{L_{JE}} = \frac{V_P}{V_E} = \frac{4}{5} \tag{3.22}
$$

延长 FE，与垂直突防区 S_{v1} 的边界交点记为 G。根据几何关系，易得：

$$
\begin{cases}
L_{JP} - d \ge L_{FP} - d \\
L_{JE} < L_{FG}
\end{cases} \tag{3.23}
$$

通过上式可得：

$$\frac{L_{JP}-d}{L_{JE}}>\frac{L_{FP}-d}{L_{FG}} \tag{3.24}$$

由于 LFP 与 LFG 的比值恰好等于直线 GP 的斜率，即：

$$\frac{L_{FP}-d}{L_{FG}}=\frac{4}{5} \tag{3.25}$$

结合式（3.24）和式（3.25），可得：

$$\frac{L_{JP}-d}{L_{JE}}>\frac{L_{FP}-d}{L_{FG}}=\frac{4}{5} \tag{3.26}$$

显然，式（3.22）和式（3.26）矛盾，则在突防航线 EF 上不存在拦截点 J，即矩形区域 ABCD 中不存在拦截点 J。那么，在拦截点存在性不满足的前提下，无论红方无人机采用什么样的追击策略，都无法拦截蓝方，则蓝方能够实现成功突防。

由于垂直突防区 S_{v1} 和 S_{v2} 关于 X 轴对称、S_{v1} 和 S_{v3} 关于直线 $y=15$ 对称、S_{v3} 和 S_{v4} 关于 X 轴对称，因此垂直突防区 S_{v2}、S_{v3}、S_{v4} 的证明过程与 S_{v1} 类似，不再赘述。

综上所述，当蓝方无人机初始位置位于垂直突防区 S_{v1}、S_{v2}、S_{v3}、S_{v4} 中，且其航线为一条与 X 轴平行的线段时，矩形区域 ABCD 中不存在拦截点 J 构成截击三角形，则蓝方无人机必定能实现成功突防，结论得证。

证毕。

根据约束条件，得到垂直突防区的仿真结果如图 3.7 所示。

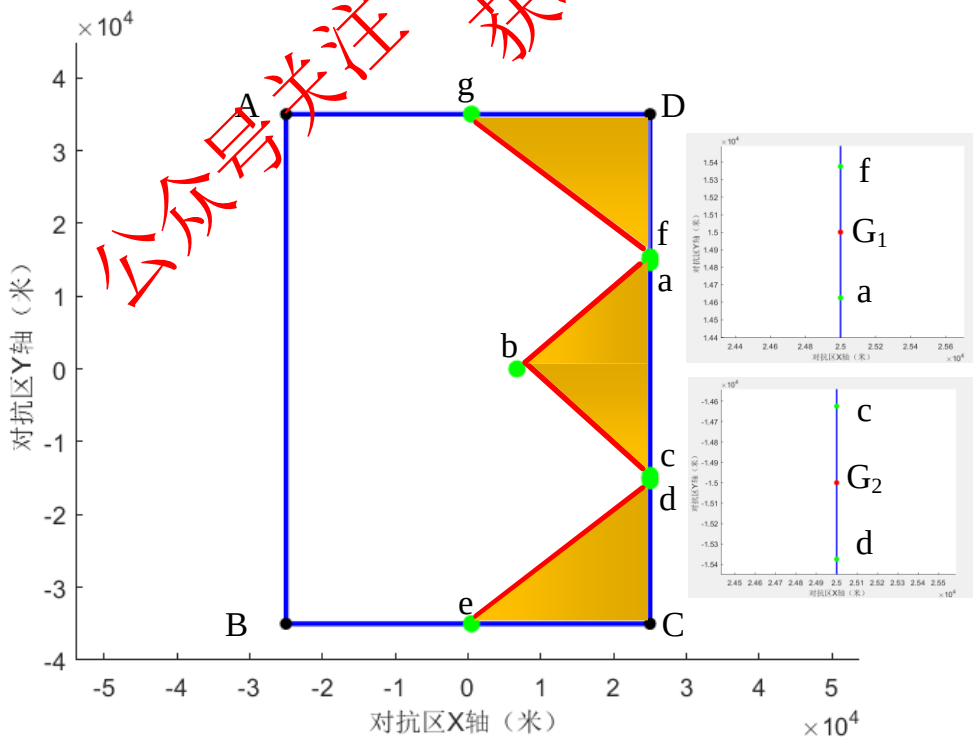


图 3.7 垂直突防区仿真结果

其中突防区各顶点坐标分别为：a(25000,14625)、b(6719,0)、c(-25000,14625)、d(25000,-15375)、e(469,-35000)、f(25000,15375)、g(469,35000)。图 3.7 还展示了 a、c、d、a 四个顶点和 G1、G2 的局部关系。

3.3.2 斜向突防区

定理 2（斜向突防必要条件） 若蓝方无人机能够实现斜向突防，则蓝方无人机的初始位置必定在区域 S_{tb} 内，其约束条件为：

$$
\begin{aligned}
&S_{tb}=S_{tb1}\cup S_{tb2}\cup S_{tb3}\\
&S_{tb1}=\left\{(x,y)\middle| x\le\frac{L}{2},\left((x-\frac{L}{2})^2+y^2\right)^{\frac{1}{2}}<\frac{V_E}{V_P}\left(\frac{G_1G_2}{2}-d\right)\right\}\\
&S_{tb2}=\left\{(x,y)\middle| x\le\frac{L}{2},y\le\frac{M}{2},\left((x-\frac{L}{2})^2+(y-\frac{M}{2})^2\right)^{\frac{1}{2}}<\frac{V_E}{V_P}\left(DG_1-d\right)\right\}\\
&S_{tb3}=\left\{(x,y)\middle| x\le\frac{L}{2},y\ge-\frac{M}{2},\left((x-\frac{L}{2})^2+(y+\frac{M}{2})^2\right)^{\frac{1}{2}}<\frac{V_E}{V_P}\left(CG_2-d\right)\right\}
\end{aligned}
\tag{3.27}
$$

证明：为便于证明过程的描述，给出区域 S_{tb} 的示意图，如图 3.8 所示。

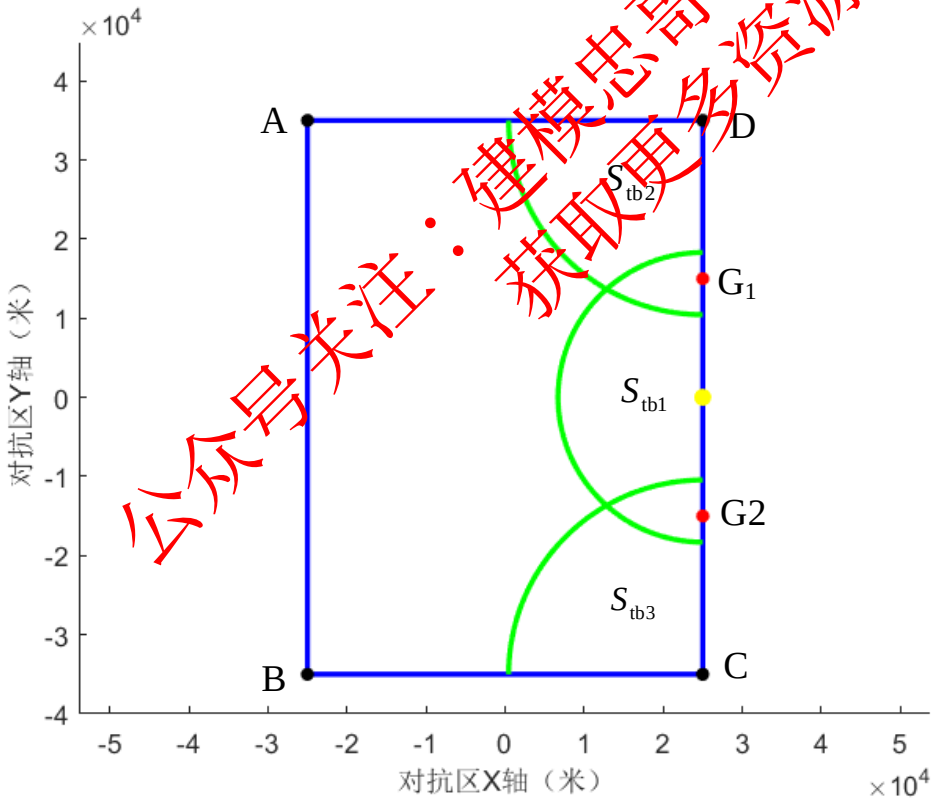


图 3.8 区域 S_{tb} 示意图

将线段 CD 分为 DG_1、G_1G_2、CG_2 三段，首先对 G_1G_2 段进行分析。在 G_1G_2 上任取一点 $F(x_F,y_F)$ 作为突防点，即蓝方无人机成功突防时在 CD 上的位置。在矩形 ABCD 中任取一点 E，为蓝方无人机的初始位置，则蓝方无人机的斜向突防航线为 EF，其长度记为 L_{EF}，如图 3.9 所示。

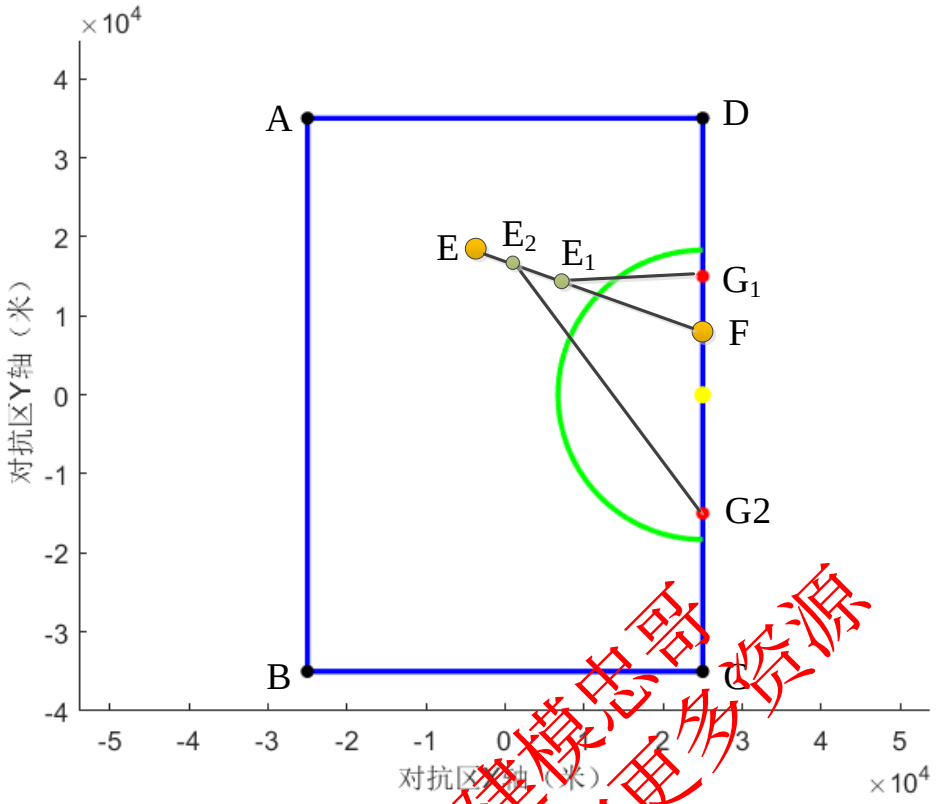


图 3.9 任取一条突防航线

考虑初始位置位于 G1 的红方无人机集群对其进行拦截。根据截击三角形可知，当：

$$\frac{L_{G_1F}-d}{L_{EF}}=\frac{V_P}{V_E} \tag{3.28}$$

红方恰好能够在边界 CD 上对蓝方进行拦截，记满足上述条件的初始点 $E=E_1$。即：

$$L_{E_1F}=\frac{V_E}{V_P}\left(L_{G_1F}-d\right) \tag{3.29}$$

当

$$L_{EF}>L_{E_1F} \tag{3.30}$$

时，由截击三角形的性质易得，在 EF 上必定存在拦截点 J，使△EJG_1 构成截击三角形，即蓝方无人机一定无法实现斜向突防。在给定突防点 F 的情况下，若蓝方无人机能够实现斜向突防，必须满足 $L_{EF}<L_{E_1F}$。因此，可行的初始位置 E 是一个以突防点 $F(x_F,y_F)$ 为圆心，以 L_{E_1F} 为半径的半圆，即图 3.10 中的 C_1。

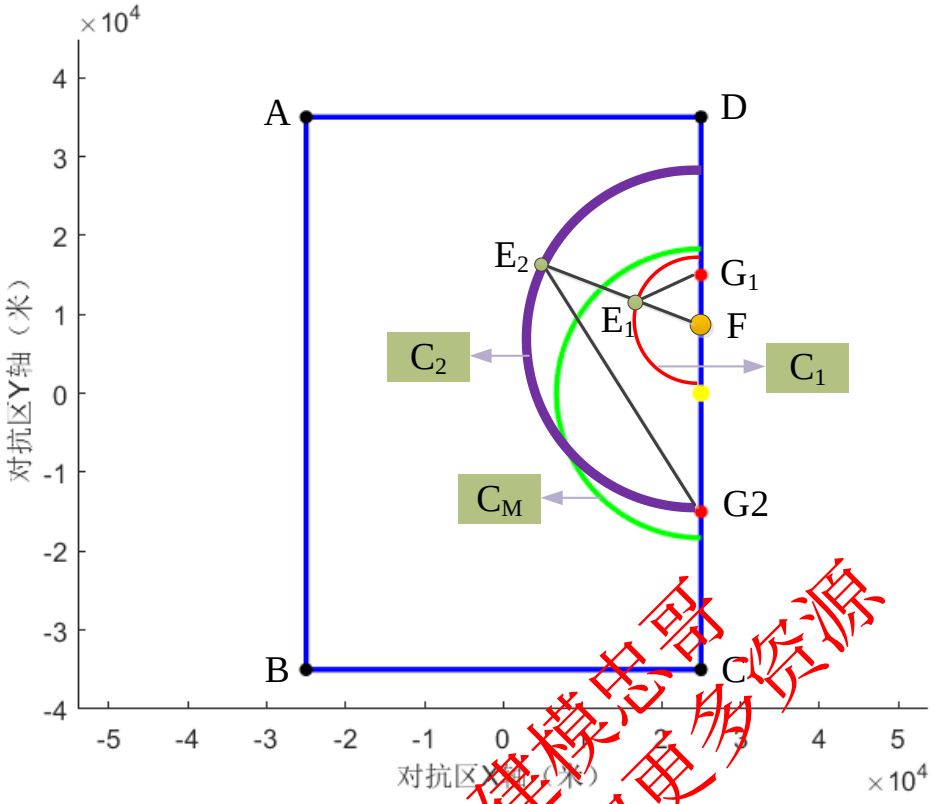


图 3.10 可行的初始位置 E

同理，考虑初始位置位于 G_2 的红方无人机集群对其进行拦截，得到相应的临界点 $E=E_2$，使得：

$$L_{E_2F}=\frac{V_E}{V_P}\left(L_{G_2F}-d\right) \tag{3.31}$$

以突防点 F 为圆心，以 L_{E_2F} 为半径的半圆，即图 3.10 中的 C_2。显然，C_1 和 C_2 为半径不同的同心半圆，其半径大小与突防点 F 的位置有关。在同时考虑 G_1 和 G_2 的情况下，形成的区域为 $C_1\cap C_2$。

综合上述分析，在给定突防点 F 的情况下，考虑位于 G_1 和 G_2 红方无人机集群协同对其进行拦截，若蓝方无人机能够实现斜向突防，必须满足：

$$L_{EF}<\min\left\{L_{E_1F},L_{E_2F}\right\}=\frac{V_E}{V_P}\left(\min\left\{L_{G_1F},L_{G_2F}\right\}-d\right) \tag{3.32}$$

可以看出，在 G_1G_2 上选取不同的突防点 F 时，斜向突防区的半径随变化。考虑选取 G_1G_2 中点为突防点的情况（记为 $F_M\left(\frac{L}{2},0\right)$），按照上述推导过程，我们容易得到其半圆区域 C_M 满足：

$$L_{EF_M}<\frac{V_E}{V_P}\left(\min\left\{L_{G_1F_M},L_{G_2F_M}\right\}-d\right)=\frac{V_E}{V_P}\left(\frac{G_1G_2}{2}-d\right) \tag{3.33}$$

根据 $C_1\cap C_2$ 和 C_M 的圆心位置及半径，可以得到：

$$C_1\cap C_2\subset C_M \tag{3.34}$$

即：C_M 能够将 $C_1 \cap C_2$ 完全覆盖。根据前文突防点的任意性，对于 $\forall F(x_F, y_F) \in G_1G_2$，产生的斜向突防区 $C_p \cap C_q$，均能被 C_M 完全覆盖。下面给出 C_M 的解析表达式：

$$C_M = \left\{ (x,y) \middle| x \le \frac{L}{2}, \left((x-\frac{L}{2})^2 + y^2 \right) < \frac{V_E}{V_P} \left(\frac{G_1G_2}{2} - d \right) \right\} \tag{3.35}$$

综上分析，对于选取 G_1G_2 上任意一点作为突防点的情况，若要使蓝方无人机能够实现斜向突防，蓝方无人机的初始位置必须位于 C_M 内；反之，当初始位置在 C_M 外时，一定能够构成截击三角形使蓝方无人机突防失败。为方便统一符号，这里将 C_M 记为 S_{tb1}。对于突防点在 DG_1、CG_2 上的情况，推导过程与 G_1G_2 的情况类似，最终得到的斜向突防区为 S_{tb2} 和 S_{tb3}。

证毕

上述定理给出了蓝方无人机能够实现斜向突防的必要条件，即蓝方无人机初始位置必须位于区域 S_{tb} 内；反之，若初始位置位于区域 S_{tb} 外，则一定不能实现斜向突防。但是，这并非一个充分条件，并不是区域 S_{tb} 内的所有点都能够实现斜向突防。**蓝方无人机能够实现斜向突防的充分条件**是：给定蓝方无人机初始位置 E 和斜向突防航线 EF，在 EF 上不存在使得△EJG_1 或△EJG_2 构成截击三角形的拦截点 J。

上述给出的两个条件是后续求解斜向突防区的重要基础，后续求解过程中将对区域 S_{tb} 进行分割，剔除不满足充分条件的区域，找出最终的斜向突防区。

定理 3（斜向突防充分条件） 给定斜向突防区 S_t 的约束条件：

$$S_t = S_{t1} \cup S_{t2} \cup S_{t3}$$

$$S_{t1} = \left\{ (x,y) \middle| x \le \frac{L}{2}, -\frac{G_1G_2}{2} + d \le y \le \frac{G_1G_2}{2} - d, \left((x-\frac{L}{2})^2 + y^2 \right)^{\frac{1}{2}} < \frac{V_E}{V_P} \left(\frac{G_1G_2}{2} - d \right) \right\}$$

$$S_{t2} = \left\{ (x,y) \middle| x \le \frac{L}{2}, \frac{G_1G_2}{2} + d \le y \le \frac{M}{2}, \left((x-\frac{L}{2})^2 + (y-\frac{M}{2})^2 \right)^{\frac{1}{2}} < \frac{V_E}{V_P} (DG_1 - d) \right\} \tag{3.36}$$

$$S_{t3} = \left\{ (x,y) \middle| x \le \frac{L}{2}, -\frac{M}{2} \le y \le -\frac{G_1G_2}{2} - d, \left((x-\frac{L}{2})^2 + (y+\frac{M}{2})^2 \right)^{\frac{1}{2}} < \frac{V_E}{V_P} (CG_2 - d) \right\}$$

当蓝方无人机初始位置位于斜向突防区 S_t 中，且斜向突防航线指向每个子区域圆的圆心时，无论红方无人机采用什么样的追击策略，蓝方无人机总能实现成功突防。

证明：首先，对区域 S_{tb} 进行分割。以区域 S_{tb1} 为例，采用两条直线

$$y = \frac{G_1G_2}{2} - d, y = -\frac{G_1G_2}{2} + d \tag{3.37}$$

对 S_{tb1} 区域进行分割，得到分割后的区域分别记为 S_{t1}、S_{t1a}、S_{t1b}，如图 3.11 所示。下面分别证明区域 S_{t1} 中的所有点都满足“蓝方无人机能够实现斜向突防的充分条件”，而区域 S_{t1a}、S_{t1b} 中的所有点都不满足该充分条件。

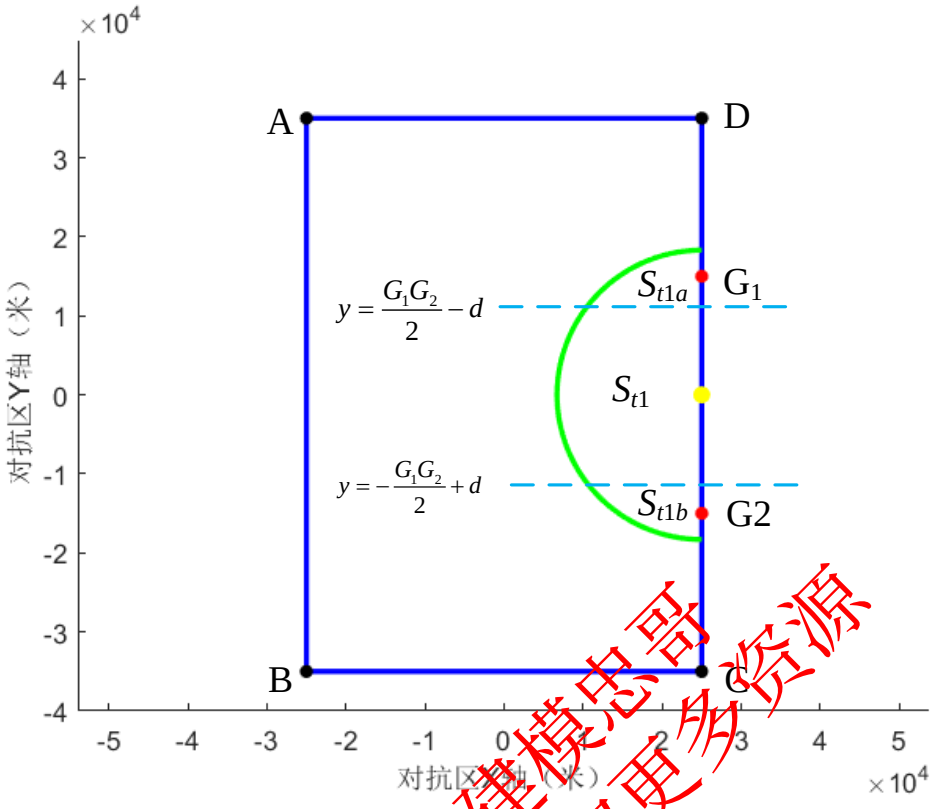


图 3.11 区域 S_{t1} 分割示意图

首先，证明区域 S_{t1} 中的所有点都满足“蓝方无人机能够实现斜向突防的充分条件”。在区域 S_{t1} 中任取一点作为蓝方无人机的初始位置（记为 E），记 G_1G_2 的中点为 F，连接 FE 并延长，与区域 S_{t1} 边界的交点记为 E'。作 $FQ\perp EF$，交 $E'G_1$ 的延长线于 Q 点，如图 3.12 所示。

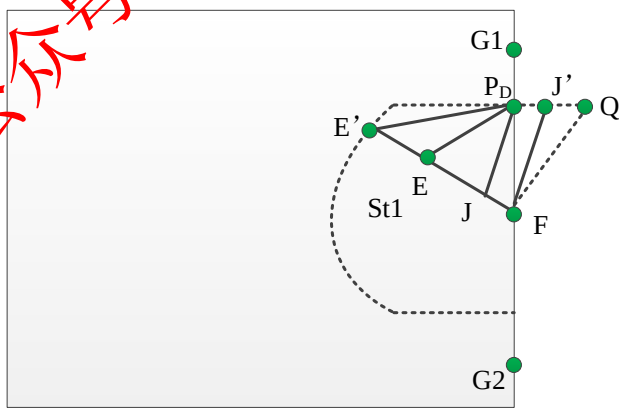


图 3.12 区域 S_{t1} 示意图

圆弧的半径为：

$$L_{E'F}=\frac{V_E}{V_P}\left(\frac{G_1G_2}{2}-d\right)=\frac{V_E}{V_P}\left(L_{FG_1}-d\right) \tag{3.38}$$

通过变化得到：

$$\frac{L_{FG_1}-d}{L_{E'F}}=\frac{L_{FP_D}}{L_{E'F}}=\frac{V_E}{V_P} \tag{3.39}$$

因此，$\triangle E'FP_D$ 满足截击三角形的性质，F 恰好为拦截点。假设 EF 上存在一个拦截点 J，使得 $\triangle E'JP_D$ 为截击三角形。从 F 点作一条与 JP_D 平行线，交线段 $E'Q$ 于 J' 点。因为直线 $y=\frac{G_1G_2}{2}-d$ 与 CD 垂直，因此 $\angle FP_DQ$ 必为钝角，则有 $L_{FJ'}>L_{FP_D}$，则：

$$\frac{L_{FJ'}}{L_{E'F}}>\frac{L_{FP_D}}{L_{E'F}}=\frac{V_E}{V_P} \tag{3.40}$$

根据相似三角形原理，$\triangle E'FJ'\sim\triangle E'JP_D$，则：

$$\frac{L_{JP_D}}{L_{E'J}}=\frac{L_{FJ'}}{L_{E'F}}>\frac{V_E}{V_P} \tag{3.41}$$

可以看出，$\triangle E'JP_D$ 不满足截击三角形的性质，则 J 点一定不是拦截点。这一结论与前文假设矛盾，因此 EF 上不存在拦截点。根据“蓝方无人机能够实现斜向突防的充分条件”，即初始位置为 E 点的蓝方无人机一定能够实现斜向突防。根据 E 点的任意性可知，区域 S_{t1} 中的所有点都满足“蓝方无人机能够实现斜向突防的充分条件”。

其次，以区域 S_{t1a} 为例，证明区域 S_{t1a}、S_{t1b} 中的所有点都不满足“蓝方无人机能够实现斜向突防的充分条件”。在区域 S_{t1a} 中任取一点作为蓝方无人机的初始位置（记为 E），记 G_1G_2 的中点为 F，连接 FE 并延长，与区域 S_{t1} 边界的交点记为 E'。作 $FQ\perp E'P_D$，交 $E'P_D$ 的延长线于 Q 点。延长 P_DQ，使得 $P_DQ=QK$，如图 3.13 所示。

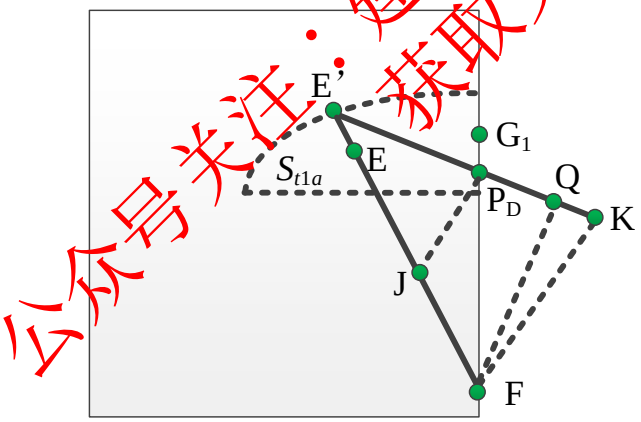


图 3.13 区域 S_{t1} 分割示意图

与前文相同，$\triangle E'FP_D$ 为截击三角形，满足性质：

$$\frac{L_{FG_1}-d}{L_{E'F}}=\frac{L_{FP_D}}{L_{E'F}}=\frac{V_E}{V_P} \tag{3.42}$$

根据全等三角形原理，$\triangle P_DFQ\cong\triangle KFQ$，则有 $L_{FP_D}=L_{FK}$，则有：

$$\frac{L_{FK}}{L_{E'F}}=\frac{L_{FP_D}}{L_{E'F}}=\frac{V_E}{V_P} \tag{3.43}$$

从 P_D 作 FK 的平行线，交 EF 于 J 点，有 $\triangle KFE'\sim\triangle P_DJE'$，则：

$$\frac{L_{JP_D}}{L_{E'J}}=\frac{L_{FK}}{L_{E'F}}=\frac{V_E}{V_P} \tag{3.44}$$

可以看出，$\triangle E'JP_D$ 满足截击三角形的性质，则 J 点是拦截点，即 EF 上存在拦截点 J。根据“蓝方无人机能够实现斜向突防的充分条件”，即初始位置为 E 点时，一定存在截击点能够拦截蓝方无人机。根据 E 点的任意性可知，区域 S_{t1} 中的所有点都不满足“蓝方无人机能够实现斜向突防的充分条件”。

综上所述，将 S_{tb1} 区域进行分割为 S_{t1}、S_{t1a}、S_{t1b} 后，只有区域 S_{t1} 中的所有点同时满足“蓝方无人机能够实现斜向突防的充分必要条件”，因此区域 S_{t1} 为斜向突防区的一部分。同样地，对 S_{tb2} 和 S_{tb3} 区域进行分割，推导过程类似，不再赘述。最终得到的区域 S_{t1}、S_{t2}、S_{t3} 的并集为斜向突防区 S_t。当蓝方无人机的初始位置位于斜向突防区 S_t 中，且突防航线为指向圆心的线段时，突防航线上一定不存在拦截点构成截击三角形，即蓝方无人机能够实现成功突防。

证毕

最终斜向突防区的仿真结果如图 3.14 所示。

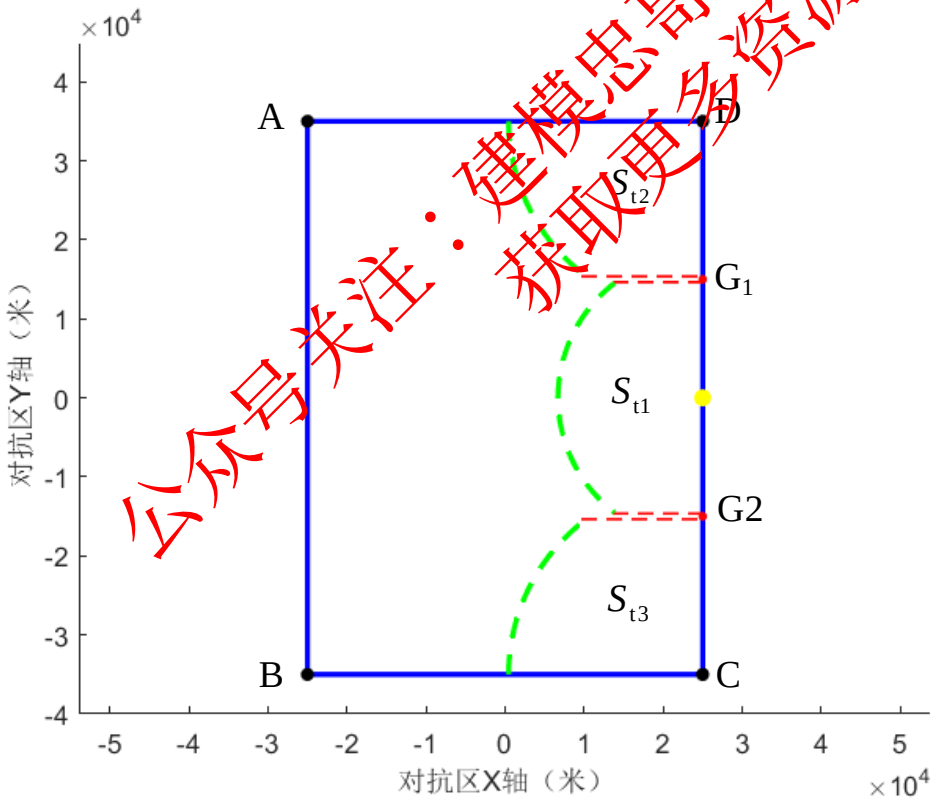


图 3.14 斜向突防区仿真结果

4 问题二的建模与求解

4.1 问题分析

问题二需要求解使得蓝方无人机一定能突破拦截的通道带宽下限，并且红方集群圆周中心的具体位置可以根据需要确定。从任务需求出发分析红蓝双方的博弈特点，红方为了拦截蓝方，首先会选择最优的初始布设位置，在通道带宽一定的情况下使得基于截击三角形的蓝方突防区最小，从而最大化蓝方突防难度，即红方集群的布设准则；蓝方则为了在

尽量短的时间内实现突防，优先选择距离短的突防路线，尽量减少机动，即蓝方突防准则。

基于问题一的分析和结论，当给定通道带宽和 G1，G2 位置时，可以得到蓝方突防区范围，只要蓝方位于突防区内则一定能突破红方的拦截。但是突防区的范围是由红方集群布设位置决定的，红方的布设方式要配合其战术意图。

因此，问题二可以转化为：在红方以最小化突防区为准则进行最优布设的情况下，求解其突防区能够包含蓝方初始位置的最小通道带宽。

4.2 模型建立

4.2.1 任务场景描述

如图 4.1 所示，对抗伊始蓝方突防无人机位于边界 AB 的中心点，红方 2 个无人机集群的圆周中心分别位于 $G1$ 和 $G2$，圆周半径为 $100m$，其中 $G1$ 和 $G2$ 位于边界 CD 上，具体位置根据需要确定。

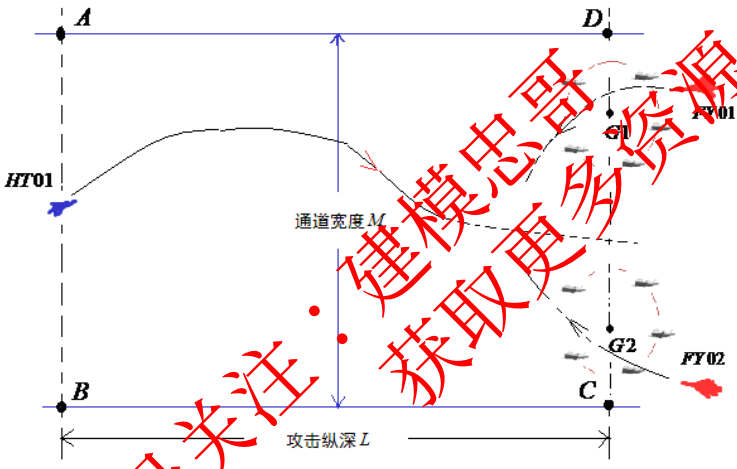


图 4.1 红蓝双方无人机对抗初始位置示意图

4.2.2 约束条件

根据题目要求，在对抗过程中蓝方和红方无人机集群都要遵循一定的约束条件，包括任务区域约束、无人机运动学约束和集群布设约束，这些约束成为策略制定和航迹规划的基本前提。具体如下：

4.2.2.1 蓝方无人机集群的约束条件

（1）蓝方任务区域约束：

$$-\frac{L}{2} \le x_F \le \frac{L}{2} \tag{4.1}$$

$$-\frac{M}{2} \le y_F \le \frac{M}{2} \tag{4.2}$$

（2）蓝方运动学约束：

$$V_E = 250m/s \tag{4.3}$$

$$R_E = 500m \tag{4.4}$$

（3）蓝方的突防成功规则：

$$t_E \le 360s \tag{4.5}$$

$$d_{EP_iP_j} \geq 300m \tag{4.6}$$

4.2.2.2 红方无人机集群的约束条件

（1）红方运动学约束：

$$V_P = 200m/s \tag{4.7}$$

$$R_P = 350m \tag{4.8}$$

（2）红方集群布设约束：要求红方任何两架无人机的间距需大于30m，即：

$$\forall P_i, P_j, d_{P_iP_j} > 30m \tag{4.9}$$

每一架无人机与本集群中至少两架无人机的距离不超过200m：

$$\forall P_i, d_{P_iP_j} \cap d_{P_iP_k} \leq 200 \quad i \neq j \neq k \tag{4.10}$$

红方运载机与所属无人机集群中至少一架无人机的距离不超过10km：

$$\exists P_i, d_{FYP_i} < 10km \tag{4.11}$$

与任何一架无人机的距离需大于100m：

$$\forall P_i, d_{FYP_i} > 100m \tag{4.12}$$

红方运载机与蓝方突防无人机的距离需大于5km：

$$d_{EFY} > 5km \tag{4.13}$$

（3）从红方视角出发，拦截成功规则：

$$d_{EP_iP_j} < 300m \tag{4.14}$$

4.2.3 突防区面积求解

通过问题分析，得到红方最优的初始布设位置是：最小化突防区面积。因此，本节根据问题一求得的突防区解析式，建立突防区面积求解模型，为优化红方最优部分提供支撑。由问题一可得在给定 G1、G2 位置的情况下，蓝方突防区的解析表达式为：

$$S_t = S_{t1} \cup S_{t2} \cup S_{t3}$$

$$S_{t1} = \left\{ (x,y) \middle| x \leq \frac{L}{2}, -\frac{G_1G_2}{2} + d \leq y \leq \frac{G_1G_2}{2} - d, \left[\left(x - \frac{L}{2} \right)^2 + y^2 \right]^{\frac{1}{2}} < \frac{V_E}{V_P} \left(\frac{G_1G_2}{2} - d \right) \right\}$$

$$S_{t2} = \left\{ (x,y) \middle| x \leq \frac{L}{2}, \frac{G_1G_2}{2} + d \leq y \leq \frac{M}{2}, \left[\left(x - \frac{L}{2} \right)^2 + \left(y - \frac{M}{2} \right)^2 \right]^{\frac{1}{2}} < \frac{V_E}{V_P} \left(DG_1 - d \right) \right\} \tag{4.15}$$

$$S_{t3} = \left\{ (x,y) \middle| x \leq \frac{L}{2}, y \geq -\frac{M}{2}, \left[\left(x - \frac{L}{2} \right)^2 + \left(y + \frac{M}{2} \right)^2 \right]^{\frac{1}{2}} < \frac{V_E}{V_P} \left(CG_2 - d \right) \right\}$$

突防区由S_{t1}、S_{t2}、S_{t3}三部分组成，这三个区域的圆弧段半径分别为：

$$r_1=\frac{V_E}{V_P}\left(\frac{G_1G_2}{2}-d\right),r_2=\frac{V_E}{V_P}\left(DG_1-d\right),r_3=\frac{V_E}{V_P}\left(CG_2-d\right) \tag{4.16}$$

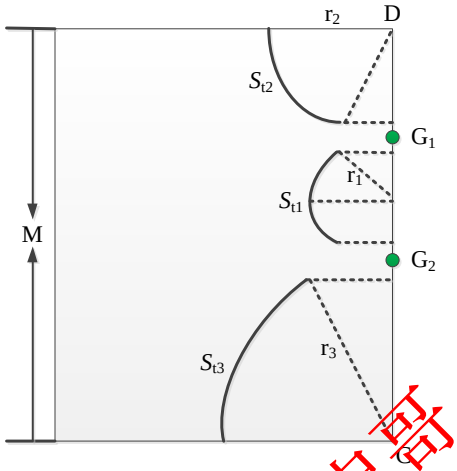


图 4.2 突防区面积求解示意图

从图 4.2 中可以看到，每个部分都可以拆分为一个扇形和一个三角形的组合，扇形和三角形的面积之和记为一个部分的面积，最后突防区总面积为三部分面积之和。三个部分形状相似，进行面积求解的方法是一致的，以区域S_{t2}为例，首先将扇形和三角形区域分别记为$S_{t2}^{(1)}$和$S_{t2}^{(2)}$，如图 4.3 所示：

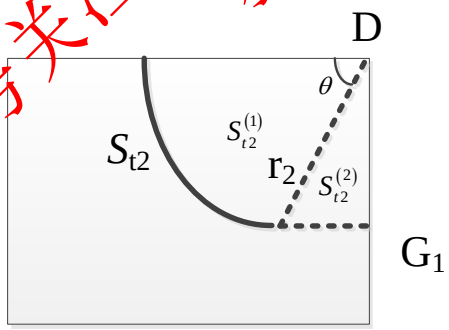


图 4.3 S_{t2}区域示意图

扇形$S_{t2}^{(1)}$的面积计算公式为：

$$S_{t2}^{(1)}=\frac{1}{2}r_2^2\theta=\frac{1}{2}DG_1^2\frac{\theta}{\sin^2\theta} \tag{4.17}$$

三角形$S_{t2}^{(1)}$的面积计算公式为：

$$S_{t2}^{(2)}=\frac{1}{2}DG_1r_2\cos\theta=\frac{1}{2}DG_1^2\frac{\cos\theta}{\sin\theta} \tag{4.18}$$

区域S_{t2}的面积为两部分之和：

$$S_{t2}=S_{t2}^{(1)}+S_{t2}^{(2)}=\frac{1}{2}DG_1^2\left(\frac{\theta}{\sin^2\theta}+\frac{\cos\theta}{\sin\theta}\right) \tag{4.19}$$

可以看到，区域S_{t2}的面积与DG_1的长度和角度θ有关，对角度θ进行分析：

$$\sin\theta=\frac{DG_1}{r_2}=\frac{DG_1}{\frac{V_E}{V_P}(DG_1-d)}=\frac{1}{\frac{V_E}{V_P}\left(1-\frac{d}{DG_1}\right)} \tag{4.20}$$

因为$d\ll M$，则根据上式可以近似认为：

$$\sin\theta\approx\frac{V_P}{V_E} \tag{4.21}$$

因此，这里我们可以认为θ近似为一定值。同样地，可以求得区域S_{t3}的面积为

$$S_{t3}=\frac{1}{2}CG_2^2\left(\frac{\theta}{\sin^2\theta}+\frac{\cos\theta}{\sin\theta}\right) \tag{4.22}$$

将S_{t1}分为上下两部分，两部分面积相等，且求解方法与区域S_{t2}相同：

$$S_{t1}=2\times\frac{1}{2}\times\left(\frac{G_1G_2}{2}\right)^2\times\left(\frac{\theta}{\sin^2\theta}+\frac{\cos\theta}{\sin\theta}\right)=\frac{(G_1G_2)^2}{4}\left(\frac{\theta}{\sin^2\theta}+\frac{\cos\theta}{\sin\theta}\right) \tag{4.23}$$

将三部分面积相加，我们得到突防区的总面积为：

$$S_t=S_{t1}+S_{t2}+S_{t3}=\frac{1}{4}\left((G_1G_2)^2+2(DG_1)^2+2(CG_2)^2\right)\left(\frac{\theta}{\sin^2\theta}+\frac{\cos\theta}{\sin\theta}\right) \tag{4.24}$$

其中，角度θ近似为一个定值。

4.3 模型解算及结果分析

4.3.1 红方无人机集群最优布设策略

红方无人机集群最优布设策略是使得突防区总面积最小，下面给出突防区面积最小的优化求解过程。给定 M 的情况下，不同的 G1、G2 初始布设方式，对应了不同的突防区及其面积大小。确定最优布设方案就是要找出DG_1，G_1G_2，CG_2之间的关系，使得突防区S_t的面积最小。

在模型建立中给出了突防区面积计算公式为：

$$S_t=S_{t1}+S_{t2}+S_{t3}=\frac{1}{4}\left((G_1G_2)^2+2(DG_1)^2+2(CG_2)^2\right)\left(\frac{\theta}{\sin^2\theta}+\frac{\cos\theta}{\sin\theta}\right) \tag{4.25}$$

由于θ近似为定值，要使S_t取得最小，只需使 $f=(G_1G_2)^2+2(DG_1)^2+2(CG_2)^2$取最小值。

代入$G_1G_2=M-DG_1-CG_2$，得到：

$$\begin{aligned} f &= (G_1G_2)^2 + 2(DG_1)^2 + 2(CG_2)^2 \\ &= \left(M - DG_1 - CG_2\right)^2 + 2(DG_1)^2 + 2(CG_2)^2 \\ &= 3(DG_1)^2 + 3(CG_2)^2 + 2DG_1 \cdot CG_2 + M^2 - 2M \cdot DG_1 - 2M \cdot CG_2 \end{aligned} \tag{4.26}$$

优化函数 f 对 DG_1、CG_2 分别求偏导：

$$\frac{\partial f}{\partial DG_1} = 6DG_1 + 2CG_2 - 2M \tag{4.27}$$

$$\frac{\partial f}{\partial CG_2} = 6CG_2 + 2DG_1 - 2M \tag{4.28}$$

若要使 f 取得最小值，则有：

$$\frac{\partial f}{\partial DG_1} = \frac{\partial f}{\partial CG_2} = 0 \tag{4.29}$$

联立求解得到：

$$\begin{cases} 6DG_1 + 2CG_2 - 2M = 0 \\ 6CG_2 + 2DG_1 - 2M = 0 \end{cases} \Rightarrow DG_1 = CG_2 = \frac{1}{4}M \tag{4.30}$$

因此有：

$$G_1G_2 = M - DG_1 - CG_2 = \frac{1}{2}M \tag{4.31}$$

综上，给定 M 时，当 $DG_1 = CG_2 = \frac{1}{4}M$，$G_1G_2 = \frac{1}{2}M$ 时，突防区的面积最小，即为红方无人机集群的最优布设策略。

4.3.2 通道带宽下限求解

当 M 不断增大时，突防区会不断增大，直到 M 达到一个临界值时，突防区恰好能够覆盖蓝方无人机的初始位置 E。蓝方无人机的初始位置为 AB 中点，记为 $E\left(-\frac{L}{2},0\right)$，CD 的中点记为 $F\left(\frac{L}{2},0\right)$。当 $r_1 > L_{EF}$ 时，突防区能够覆盖 E 点，即：

$$r_1 = \frac{V_E}{V_P}\left(\frac{G_1G_2}{2} - d\right) > L \tag{4.32}$$

代入 $G_1G_2 = \frac{M}{2}$，得：

$$\frac{V_E}{V_P}\left(\frac{M}{4} - d\right) > L \tag{4.33}$$

求解不等式，可得：

$$M > 4\left(\frac{V_P}{V_E}L + d\right) = 161.428km \tag{4.34}$$

因此存在一个 M 的下限 $M_{\min}=161.428\text{km}$，当 $M>M_{\min}$ 时，蓝方无人机位于突防区内，采取 EF 为突防航线时，一定能够实现突防。

5 问题三的建模与求解

5.1 问题分析

本题根据题意，红方兵力由两架运载机组成，每架运载机可分两个波次共发射 10 架无人机组成 4 个无人机集群对蓝方一架无人机进行拦截。本题可视为多个智能体的博弈问题，1 个蓝方智能和 4 个红方智能体进行攻防对抗。智能体反复的在执行 OODA 循环，首先观察敌我态势信息，接着对敌我态势做出判断、并对敌方意图进行预测，然后根据当前态势和预测信息根据智能体攻防策略做出最佳决策，最后执行决策开启下一轮 OODA 循环，如图 5.1 所示。其中蓝方执行突防策略，红方执行拦截策略[1,2]。

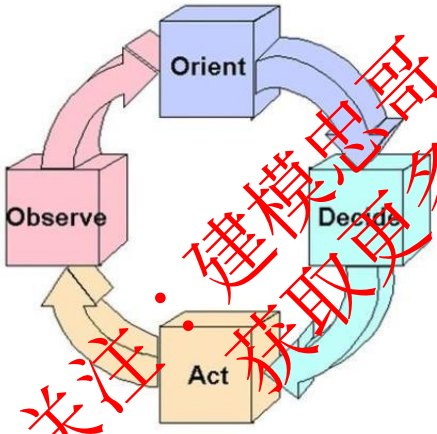


图 5.1 OODA 循环

本题需要为红方设定拦截策略，根据策略确定每个子群的构型、无人机数量、运载机的发射时间、发射位置、第二波无人机集群的中心位置等。本题分为两问，第一问是为红方设定拦截策略以实现对蓝方的最优拦截效果，第二问在第一问最优策略的基础上缩小通道带宽 M，寻找 M 的上限 $M_{\max}$，当实际通道带宽 M 小于 $M_{\max}$ 时，无论蓝方无人机采用什么样的突防策略，红方无人机集群都可以在区域 ABCD 内对其实现拦截。

本题红方无人机的优势在于数量优势，通过集群间的协同可以弥补飞行能力上的劣势，因此本题红方拦截策略的研究重点为发挥集群的协同优势。我们从两个角度来进行协同，分别是数量协同和能力协同。**数量协同**是不考虑各个子群间的战术配合，将所有子群视为一个整体，单纯靠数量优势的实现“以多胜少”；**能力协同**是指将集群拆分成若干个子群，为子群设定一定的战术，子群在空间是时间上实现战术的配合，从而达到能力的涌现，实现“1+1>2”的效果。后文将会对两种协同策略进行详细阐述。

本问题基于如下合理假设进行求解：

（1）红方无人机集群切换构型时，由于受到运动约束，因此构型切换时间不能忽略，但该时间的计算不是本文的重点，因此本文对构型切换时间做简化处理，假设构型切换时间为一固定值 10 秒。

（2）由于本题场景具有强烈的对抗博弈，因此将红蓝无人机及运载机的转弯半径视为一定值，均等于其各自的最小转弯半径，这样的假设也是在对抗场景中飞机平台充分发挥其机动性能做出的，具有一定的合理性。

5.2 仿真系统搭建

鉴于本文的决策变量空间维度较大，敌我双方博弈决策不确定程度高，我们难以通过解析或优化的方法直接求解。为此，我们利用 Matlab 软件搭建了一套简单仿真系统。一方面，我们希望借助仿真系统，在众多决策变量难以确定的情况下，通过仿真摸索敌我双方博弈对策中的一些潜在规律。另一方面，我们能够基于仿真系统进行大规模蒙特卡洛仿真，从而不断逼近客观的普适结果。搭建的仿真系统框架如图 5.2 所示。

图 5.2 仿真系统框架

仿真系统由运行入口、集群类、智能体类、蓝方机动策略库、红方机动策略库和终止条件等模块构成，图 5.2 中展示了仿真系统中的核心模块。下面对各个模块及其交联关系进行简要介绍，并叙述我们是如何基于仿真系统完成本文工作的。

本文中涉及到的实体主要有蓝方无人机、红方运载机和红方无人机集群，将上述实体的属性抽象出来，建立集群类 ClassSwarm。集群由若干个智能体构成，集群类中只描述了集群的参数属性和功能，而对于集群中的每一架无人机，我们建立了智能体类 ClassAgent 对其进行建模。

集群类的初始化。首先，采用集群类对蓝方无人机和红方无人机集群进行实体化，建立集群对象。针对蓝方无人机对象，其中只包含一个智能体 Agent。对于红方无人机集群对象，其中则包含了数量可设置的智能体个数。其次，对集群进行参数初始化，输入 Agent 个数、各 Agent 初始位置和航向、飞行速度和最小转弯半径。最后，通过类的初始化函数，计算转弯次数、转弯角速度等必要参数，存储到对象中以便后续调用。

集群类的功能。集群类的主要功能有红蓝双方无人机的飞行运动和态势显示。给定初始参数和属性后，若不对集群对象施加控制指令，则集群会沿着当前航线向前飞行，需要不断更新位置和航向信息。我们综合考虑无人机的各项机动转弯约束，将无人机状态方程写入类函数 UpdateState 中，用于驱动无人机按照当前状态飞行。若从外部对集群施加一个

控制指令，则集群需要执行转弯机动动作。为此，我们建立了类函数 SwarmTurn，严格遵守转弯半径约束执行转弯动作。为进一步观察集群对抗飞行过程，我们建立了类函数 plotPath 和 scatterPoint，用于实时绘制集群航线和散点图。以 Matlab 高级绘图功能做辅助，我们可以动态绘制实时态势图，为我们分析对抗过程和挖掘内在规律提供了有力支撑。

机动策略库是一个动态开放的函数库。当我们为红蓝双方设计不同的机动对抗策略时，只需要调用不同的机动策略库即可，具有较强的可扩展性。机动策略库的功能主要是根据当前红蓝双方无人机的态势信息，在相应的策略指导下，为红蓝双方产生相应的控制指令。例如，我们后面设计了基于转弯截击法的追踪拦截策略，截击点的计算和追踪控制都写入到策略库中，进而生成输出控制指令。控制指令作用于集群类的实体对象，基于集群类的功能，就能实现控制指令的执行。

运行入口 Main.m 主要用于驱动整个仿真系统的运行。我们设置了离散时间步长，利用离散时间计数器来驱动仿真环境不断向前推演。在达到终止条件后，完成一次仿真，计数器清零，重新开始新一轮仿真，这是我们进行大规模蒙特卡洛仿真的重要支撑功能。

对于问题三，我们首先设计了多种红方拦截策略，并形成红方机动策略库。在蓝方无人机随机飞行的情况下，进行大规模蒙特卡洛仿真，记录突防成功率和突防成功的航线。进一步分析仿真结果，我们可以挖掘出双方策略博弈的内在机理，从而指导我们进行策略改进。完成策略的优化后，我们在不断缩小通道带宽 M 的情况下，进行多组蒙特卡洛仿真，最后寻找蓝方突防概率为 0 的宽度上限。需要特别指出的是，搭建的仿真系统在集群类的支撑下，能够适用于集群数量和规模较大的情况，具有较强的扩展性，尤其是对于问题四一类无人机集群数量较多的情况。

5.3 模型建立

5.3.1 Stackelberg 博弈论模型

Stackelberg 博弈模型是以德国经济学家 Heinrich Freiherr von Stackelberg 的名字命名的著名博弈论模型,他在 1934 发表的论文《Market Structure and Equilibrium》中首次提出了这种博弈模型。该模型原是用于描述经济领域中的双寡头竞争模型，市场中有两个相互竞争的厂商,每个厂商要顾忌对方的产量。在存在市场进入壁垒和两个厂商都拥有市场力量的前提下,这个博弈的两个参与者分别是领导者和跟随者。领导者先行选择产量,跟随者观察到跟随者的选择后再作选择。

在博弈达到均衡的过程中,领导者知道跟随者会观察它的选择,还知道跟随者将采取最优的跟随策略同时,跟随者知道领导者知道自己会观察它,也知道领导者知道自己会采取的跟随策略。这样,领导者的最优策略是选择对应于跟随者的策略对自己最优的策略,而跟随者的最优策略是使自己最优化。

从模型描述中可以看出，Stackelberg 博弈是一种博弈双方在非完全信息条件下的动态博弈，并且双方都会根据对方的行动来调整自己的行动策略，以求得在双方都进行动态调整的情况下对己方最有利的结果。

这种场景非常适用于无人机集群协同对抗问题。在本题中，红方无人机集群对应于 Stackelberg 博弈模型中的领导者，试图突防的蓝方无人机集群对应于 Stackelberg 博弈模型中的跟随者。红方无人机集群执行拦截任务，能够发现敌机，并且具有选择己方战术、飞行区域以及机动策略的自由，但同时蓝方无人机集群也能够观察到红方无人机集群的拦截策略，并基于红方的集群布设方式和机动情况选择对蓝方而言最有利的相应突防策略。在这个动态博弈中，红方作为领导者具有先行发现和先行机动的优势，但同时也存在机动速度慢、不知道敌机应变策略的劣势；蓝方作为跟随者具有机动速度快、能够观察红方拦截策略和布设方式从而及时调整突防策略的优势，但同时存在数量上的劣势。

5.3.2 无人机运动模型

为便于研究，假设无人机为质量恒定的刚体，且认为地球是一个平面。在上述假设下，本节选取经典的三自由度模型来描述无人机的运动，视无人机为带有一定运动约束的质点，无人机运动学模型如下：

$$\begin{aligned}\dot{x}_i(t) &= V_i(t)\cos\gamma_i(t)\cos\psi_i(t)\\ \dot{y}_i(t) &= V_i(t)\cos\gamma_i(t)\sin\psi_i(t)\\ \dot{z}_i(t) &= V_i(t)\sin\gamma_i(t)\end{aligned} \tag{5.1}$$

其中，$i=1,\cdots,n$ 表示无人机的编号；n 为无人机数量。$x_i(t), y_i(t)$ 分别表示 UAV_i 在惯性坐标系中的东向和北向坐标，$z_i(t)$ 表示高度。$V_i(t)$ 为地速，$\gamma_i(t)$ 为爬升角，$\psi_i(t)$ 为偏航角。

根据题意，无人机在任务区域上空定高飞行，且飞行速度为定值，即爬升角 $\gamma_i(t)=0$，飞行速度 $V_i(t)$ 为常数，等于各飞行平台的运动速度。在（5.2）式无人机模型的基础上，对其进行线性化和离散化处理，得到如下简化的无人机运动模型[3]：

$$\begin{aligned}x_i(k+1) &= x_i(k)+v_0\triangle t\cos\psi_i(k)\\ y_i(k+1) &= y_i(k)+v_0\triangle t\sin\psi_i(k)\end{aligned} \tag{5.2}$$

其中，$[x_i(k), y_i(k)]$ 为 UAV_i 的位置；v_0 为平飞速度；$\triangle t$ 为时间步长。无人机运动过程中的控制输入为航向偏转角 $\triangle\psi_i(k)$，单位时间步长内无人机航向偏转存在一定的约束限制：$\triangle\psi_i\in[-\psi_{\max},\psi_{\max}]$，$\psi_{\max}$ 为受机动性能限制下的最大转弯角。偏航角的更新规则为：$\psi_i(k)=\psi_i(k-1)+\triangle\psi_i(k)$。

由于飞行轨迹可离散化为一系列航点，一系列航点构成了一段可飞航线，为了进一步简化分析，将各飞机平台的机动动作设定为直行、左转、右转三种，飞行轨迹如图 5.3 所示。

(a)直行　　(b) 右转　　(c) 左转

图 5.3 飞机平台机动动作示意图

5.3.3 拦截线模型

拦截线是指红方的兵力部署线，它是一条与 CD 边平行的线段。红方无人机集群在拦截线上完成部署形成一定构型后，编队中心在拦截线上移动，意味着当蓝方无人机突破拦截线失败被红方拦截时，拦截点必定位于拦截线上，因此拦截线也是一条由一系列拦截点的连线组成的线段。拦截线示意图如图 5.4 所示。

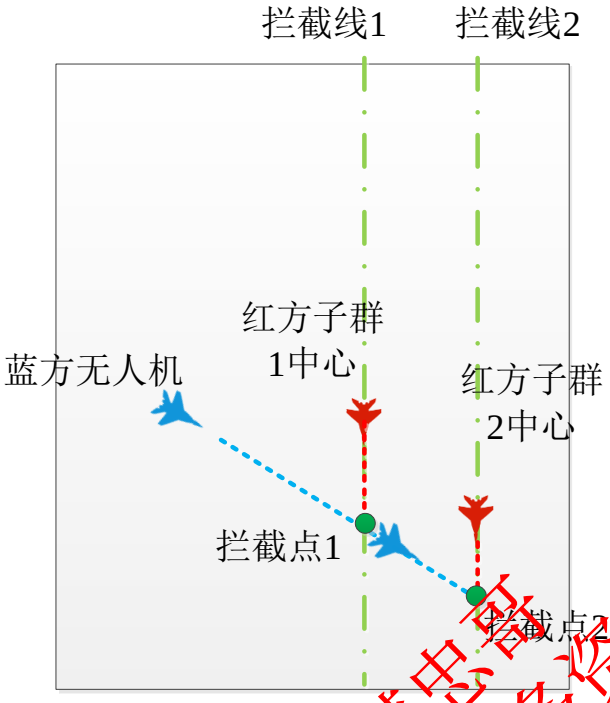


图 5.4 拦截线示意图

5.3.4 突防机动区间模型

蓝方无人机无论采用何种突防方式，当突防越过拦截线时，蓝方无人机的最后一次突防机动一定是在当前航向下采用一次转弯机动加一次直线机动越过拦截线，我们称这样的机动动作为突防机动。因此，根据蓝方无人机的运动约束，当给定一航向角时，蓝方无人机在距离拦截线某一距离之内，采用一次突防机动必定将越过拦截线，那么蓝方无人机的航迹与拦截线的交点就是突防点。我们称所有可能的突防点相连组成的线段为**突防机动区间**。突防机动区间可为红方无人机在拦截线上位置部署和运动提供参考，详细模型将在问题求解中具体阐述。

下面给出任意航向角下突防机动区间的计算方法。

首先定义偏航角 θ，令场景模型中沿 Y 轴方向顺时针转向飞机速度方向的角度为航向角，顺时针偏转为正，逆时针偏转为负。偏航角示意图如图 5.5 所示。

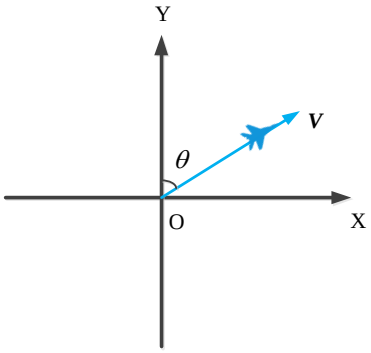


图 5.5 偏航角示意图

设蓝方无人机在某一时刻偏航角为 θ，距拦截线直线距离为 l 时开始进行突防机动。突防机动的极限情况是仅通过一次转弯机动便可以突防，此时进行转弯机动的轨迹应恰好与拦截线相切，定义此种情况下蓝方飞机离拦截线的直线距离 l 为转弯机动距离。由于无人

机可选择左转和右转两种转弯机动，因此存在两个转弯机动距离，分别为左转弯机动距离 l_l 和右转弯机动距离 l_r。突防机动区间示意图如图 5.6 所示。

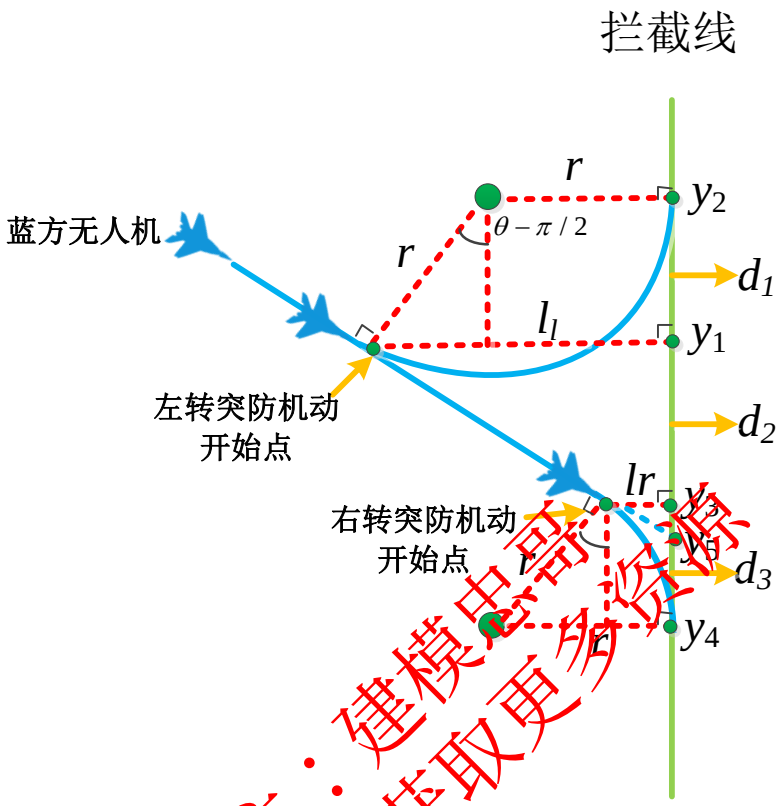


图 5.6 突防机动区间示意图

图中 y_2 为仅实施左转机动时的突防点，y_5 为仅实施直线机动时的突防点，y_4 为仅实施右转机动时的突防点。因此当蓝方飞机从离拦截线 l_l 距离左转一定偏航角进行突防时，突防范围必定在拦截点 y_2 y_5 之间，当蓝方飞机从离拦截线 l_r 距离右转一定偏航角进行突防时，突防范围必定在拦截点 y_5 y_4 之间。令 d_a 为突防机动区间，通过计算可得

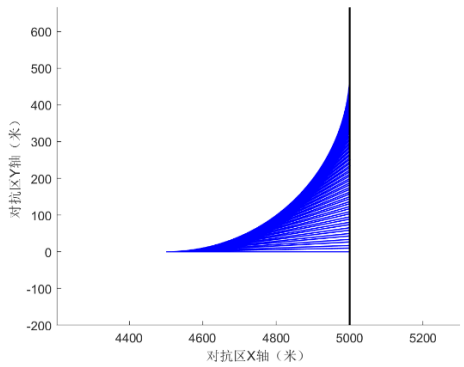
$$l_l = r(1-\cos\theta) \tag{5.3}$$

$$l_r = r(1+\cos\theta) \tag{5.4}$$

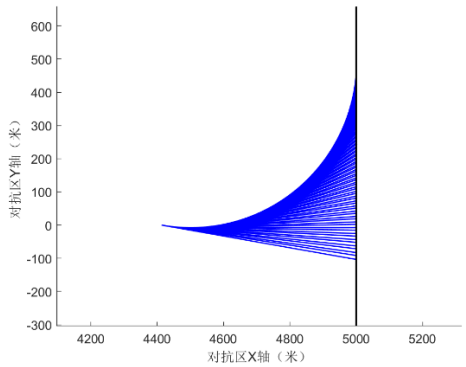
$$d_a = d_1 + d_2 + d_3 = \frac{2r}{\sin\theta} \tag{5.5}$$

同时，还可以得到当无人机在左转机动距离实施左转机动，到达突防点时偏转角的变化区间为$[0,\theta]$，当无人机在左转机动距离实施右转机动，到达突防点时偏转角的变化区间为$[0,\pi-\theta]$。

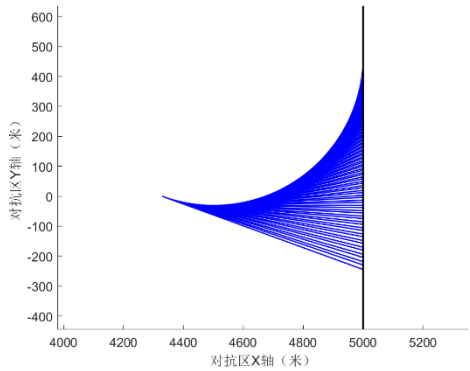
突防机动区间的物理意义是进入机动距离后的突防范围。因此本文通过仿真，给出了在$\theta\in[\frac{\pi}{2},\pi]$时，无人机在其左转机动距离通过转弯机动加直线机动，转过$[0,\theta]$区间内任意的偏航角后越过拦截线时的拦截点位置和突防时间（见附件）。不同偏航角下的左转机动突防区间如图 5.7 所示。



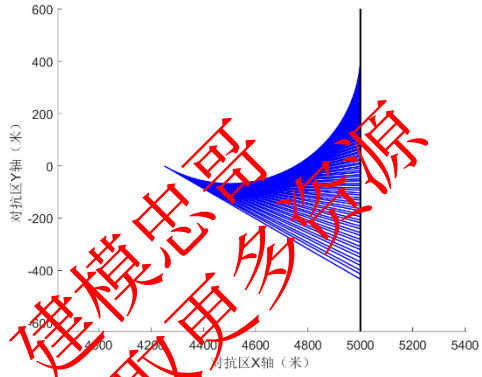
（a） $\theta=90^{\circ}$



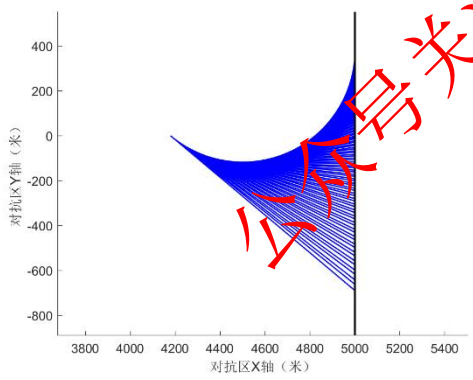
（b） $\theta=100^{\circ}$



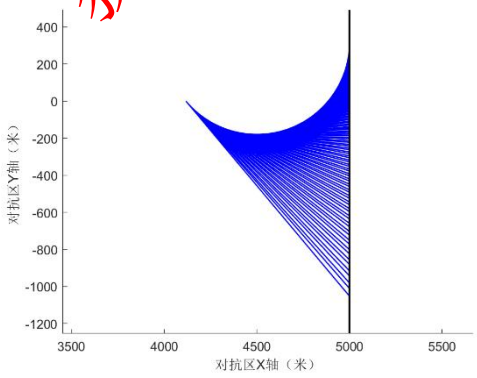
（c） $\theta=110^{\circ}$



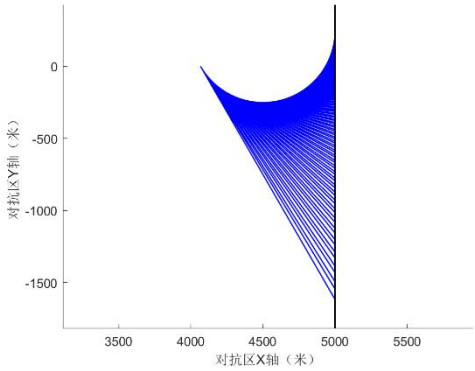
（d） $\theta=120^{\circ}$



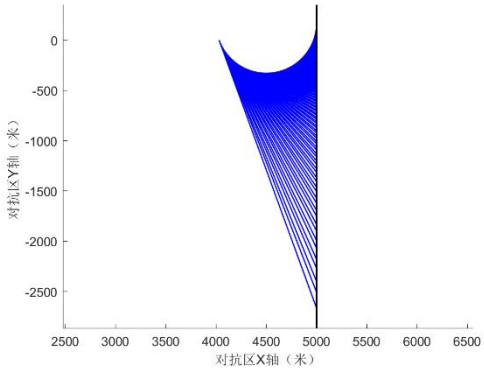
（e） $\theta=130^{\circ}$



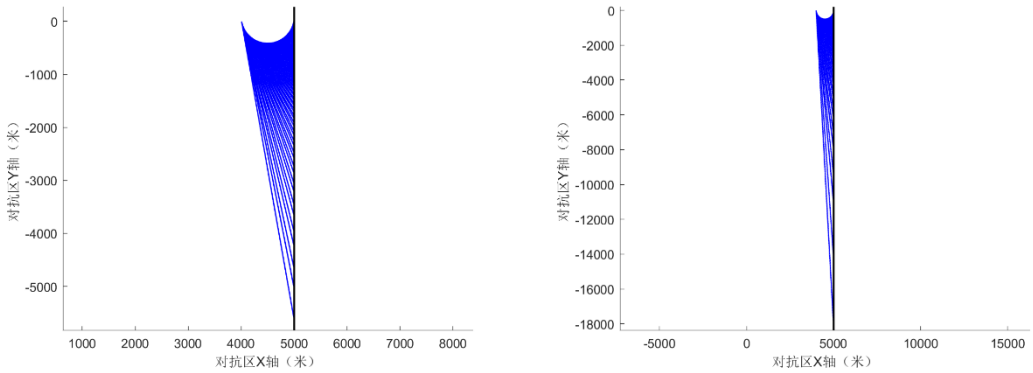
（f） $\theta=140^{\circ}$



（g） $\theta=150^{\circ}$



（h） $\theta=160^{\circ}$



（i） $\theta=170^{\circ}$　　（j） $\theta=180^{\circ}$

图 5.7 不同偏航角下的左转机动突防区间

5.3.5 转弯截击法

基于截击三角形方法，当我机位置、航向与敌机位置、航向构成一定关系时，我机恰好能在截击点与敌机相遇，实现成功拦截。但是，在敌我双方对抗博弈的过程中，由于敌机机动和转弯半径的限制，截击三角形的关系很难持续保持，如图 5.8 所示。

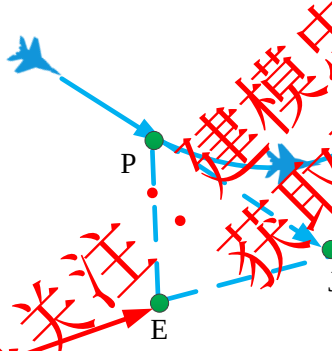


图 5.8 截击三角形丢失图

敌机位置为 E，我机位置为 P，假设在初始态势下恰好能够构成截击三角形，即截击点为 J，$\triangle EPJ$ 满足截击三角形的性质。当敌机由 E 点向 E_1 点机动转弯时，截击三角形会立刻被破坏，因此，此时我机需要立刻进行机动转弯，重新生成截击三角形。下面，给出不满足截击三角形的初始态势，我机利用一次转弯机动形成截击三角形态势。我们通过矢量加法求解转弯角度和新的截击点坐标，如图 5.9 所示。

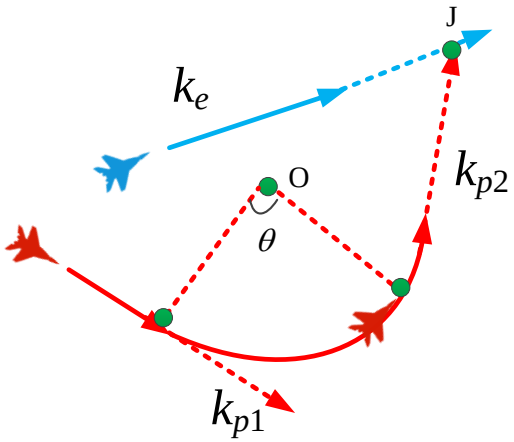


图 5.9 求解转弯角度和新的截击点坐标示意图

图 5.9 中，敌机位置为 $E(x_e,y_e)$，敌机航向为 k_e，我机位置为 $P(x_p,y_p)$，我机航向为 k_{p1}。由矢量加法法则，可得：

$$\overrightarrow{PE}+\overrightarrow{EJ}=\overrightarrow{PO}+\overrightarrow{OQ}+\overrightarrow{QJ} \tag{5.6}$$

根据欧拉公式可以将矢量表示成复数指数形式[4]：

$$\overrightarrow{PE}+V_E te^{ik_e}=r_p e^{i\left(k_{p1}+n_1\frac{\pi}{2}\right)}+r_p e^{i\left(k_{p2}-n_1\frac{\pi}{2}\right)}+L_{QJ}e^{ik_{p2}} \tag{5.7}$$

其中，矢量 $\overrightarrow{PE}$ 可通过敌我机位置解算得到。对上述公式进行实部和虚部的拆解，可以得到两个等式，利用 Matlab 软件求解，即可得到应飞航向 k_{p2} 和截击点位置 J。

与追踪拦截法相比，转弯截击法并不要求我机始终指向敌机来逼近敌机。因为敌机在速度上比我机具有较大优势，若采取类似比例导引法的追踪拦截法，可能会以后敌方的机动迅速被敌方甩在身后，在速度劣势下则永远无法追上敌机。**转弯截击法的优势在于，预测了敌机未来运动位置，避免了因为速度劣势造成被置尾的情况。**

5.4 模型解算及结果分析

本题通过设计红方智能体的拦截策略，来实现最优拦截效果。我们认为红方的策略按照战术目的可以分为**防守型策略**和**围捕型策略**，按照协同方式可以分为**数量协同策略**和**能力协同策略**。下面将对本文红方无人机策略进行介绍。

5.4.1 防守型策略

本文首先介绍防守型策略，防守型策略是指红方无人机集群仅在固定拦截线上集结和移动，不选择主动出击，编队中心始终位于拦截线上，跟随蓝方无人机集群的位置进行移动，目的是希望在拦截线上成功拦截蓝方无人机。防守型策略将介绍“|”**字构型策略**。

“|”字构型策略是指将红方 20 架无人机机为一个整体，在拦截线上“|”字排开，摈弃了子群的概念，单纯的发挥数量优势，属于数量协同策略。

(1)兵力部署及策略安排

表 5.1 给出了详细的兵力部署及策略安排。

表 5.1 一字构型兵力部署及策略安排

	运载机 1	**运载机 2**
初始时刻位置	(25000,50)	(25000,-50)
第一波次兵力/架	5	5
第二波次兵力/架	5	5
第二波次发射时刻/秒	20.944	20.944
第二波次发射位置	(24400,50)	(24400,-50)
第二波发射的无人机集群中心位置	(22400,50)	(22400,-50)

在表 5.1 的策略安排下，形成的作战场景如图 5.10 所示。

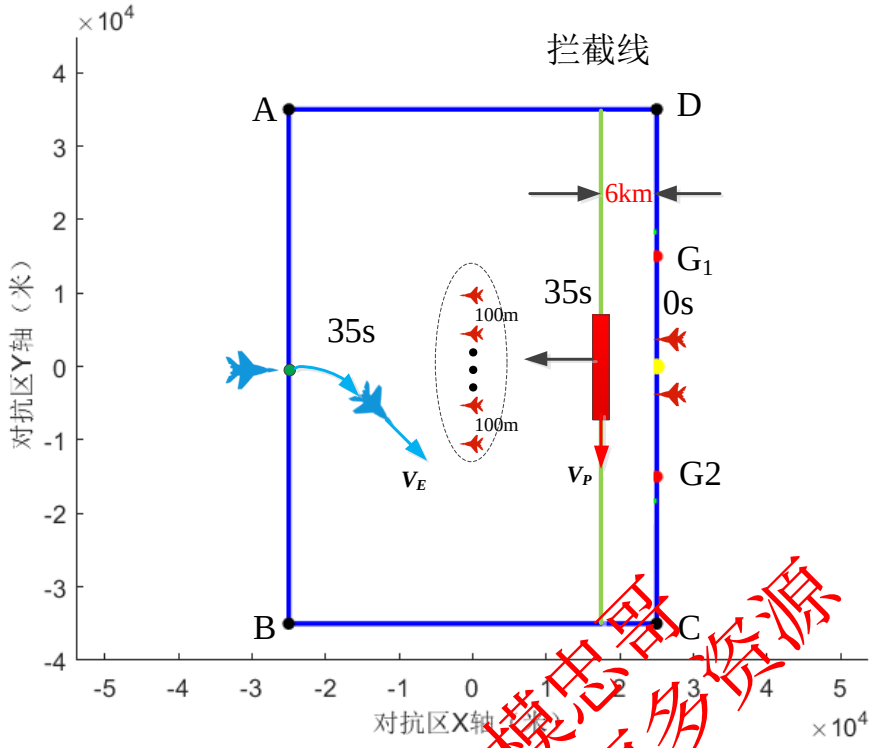


图 5.10 作战场景示意图

(2)作战时序表

表 5.2 “|”字构型作战时序表

事件	开始时间	结束时间
红方第一波次发射	0s	0s
红方第一波次前往拦截线	0s	20s
红方第一波无人机集群由圆形构形切换为一字构型	20s	30s
红方第一波沿 Y 轴负方向飞行 1km，到达待战位	30s	35s
红方运载机在起始位置按照最小飞行半径飞行 5s	0s	5s
红方第二波次发射	5s	5s
红方第一波无人机集群由圆形构形切换为一字构型	5s	15s
红方第二波向前飞行 4km 到达待战位	15s	35s

给出作战时序表 5.2 是为了证明仿真系统种初始场景设定的可实现性，仿真系统种初始场景设置为作战开始 35 秒后，红方 20 架无人机在离 CD 边 6km 的拦截线上就位，开始执行“|”字构型策略。

(3)策略细节

下面给出一字构型策略的具体细节：

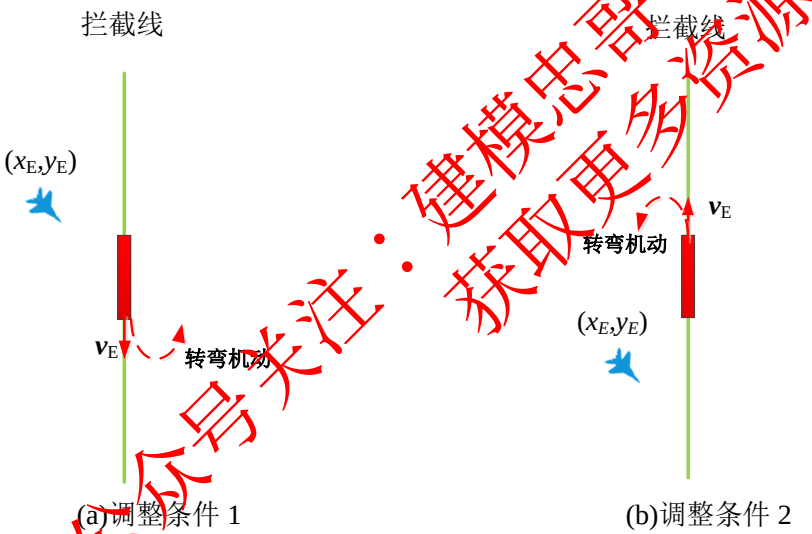
1）作战开始 35s 时，红方 20 架无人机在拦截线两侧分布，任意相邻两架无人机之间的距离为 100m，编队总长度为 1.9km，编队中心位于拦截线中点；

2）红方无人机形成“|”字构型后，每架无人机的速度方向总是沿拦截线方向，如果要调整航向，则集群内无人机通过同步转弯机动实现同步掉头，仿真环境中的无人机集群转弯细节如图 5.11 所示；

3）红方无人机的速度调整根据集群编队的位置和速度与蓝方无人机位置和速度的关系进行调整，规定速度方向与 y 轴同向为正，则调整规则为：

$$if\ \ (y_E < y_{P2}\ and\ v_P > 0)\ or\ (y_E > y_{P2}\ and\ v_P < 0) \tag{5.8}$$

调整示意图如图 5.12 所示。



(a)调整条件 1　　(b)调整条件 2

图 5.11 红方无人机集群转弯规则展示

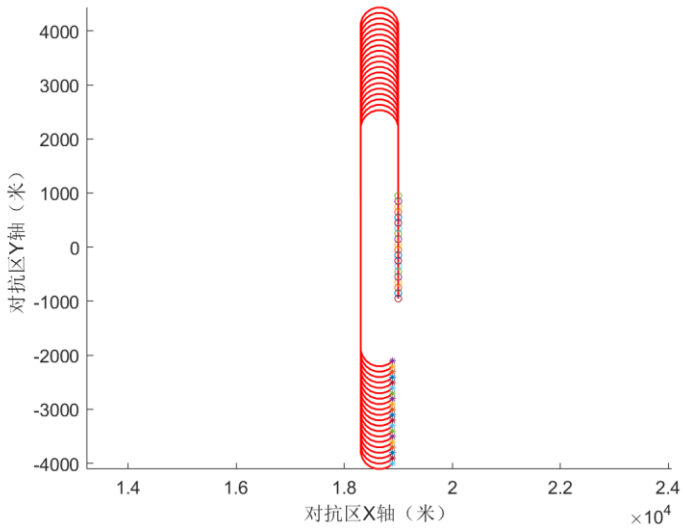


图 5.12 红方无人机集群转弯细节展示

（4）拦截效果评估

为了评估“|”字构型策略的拦截效果，在我们搭建的无人机集群协同对抗仿真系统中，进行了 2000 次仿真，通过记录成功突防的次数来计算拦截率，作为评估红方拦截效果的指标。仿真系统中，设置方蓝方策略为随机突防策略，来模拟“无论蓝方采用何种策略”。最终 2000 次仿真中，蓝方飞机有 586 次成功突防，拦截成功率为 70.7%。

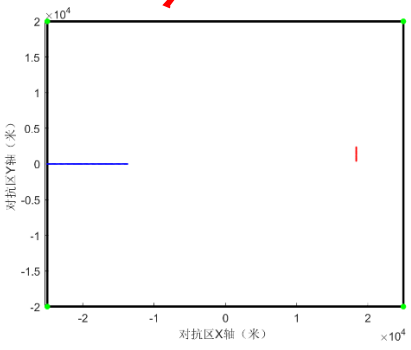
为了寻找能使拦截成功率逼近 100%的通道带宽 M 的上限 M_{max},在 5km-70km 范围内每 5km 确定一个 M 值，在仿真系统中针对每一个 M 进行 2000 次仿真，得到不同通道带宽下的拦截成功率，仿真结果如表 5.3 所示。

表 5.3 不同通道带宽下的拦截成功率

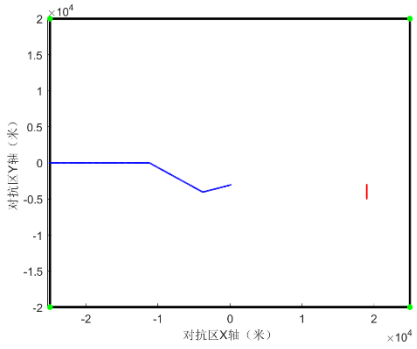
M(km)	仿真次数	突防成功次数	拦截成功概率%
5	2000	22	98.9
10	2000	121	93.95
15	2000	153	92.35
20	2000	214	89.3
25	2000	302	84.9
30	2000	352	82.4
35	2000	406	79.7
40	2000	420	79
45	2000	466	76.7
50	2000	471	76.45
55	2000	497	75.15
60	2000	551	72.45
65	2000	554	72.3
70	2000	586	70.7

通过仿真结果得到，在“|”字构型策略下，当 M=5km 时拦截成功率高达 99%，我们已经达到了的极限，因此**当红方实行防守型策略时通道带宽 M 的上限为 5km**。

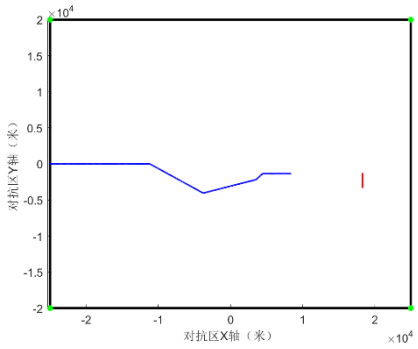
接下来进行仿真实例展示，在 M=40km 时给出两个成功突防实例和两个成功拦截实例，具体对抗过程如下图 5.13-图 5.16 所示。



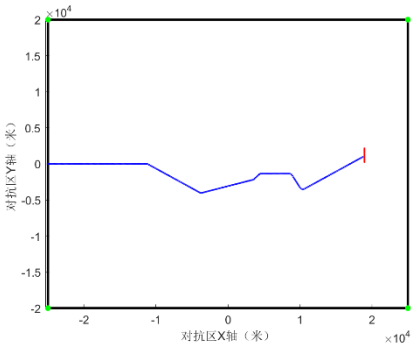
(a) 时刻 1



(b) 时刻 2

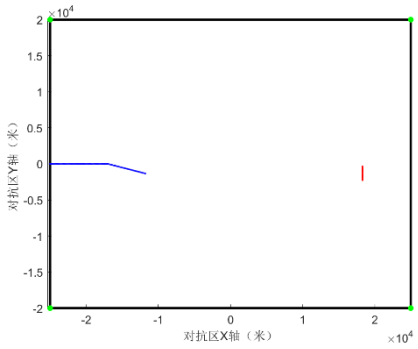


(c) 时刻 3

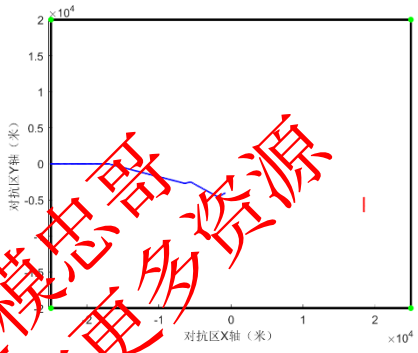


(d) 时刻 4

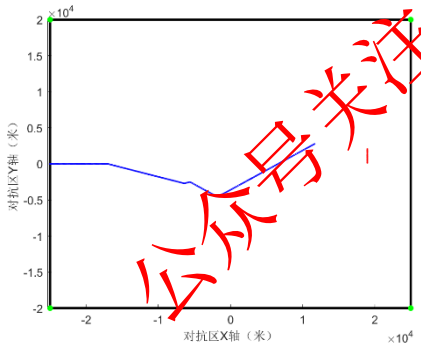
图 5.13 成功拦截实例 1 对抗过程复现



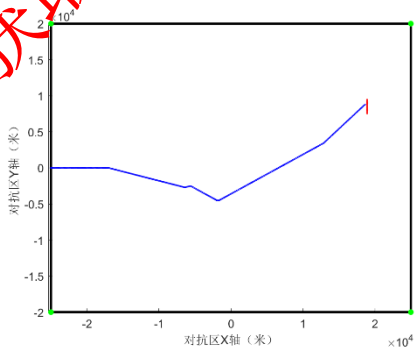
(a)时刻 1



(b) 时刻 2

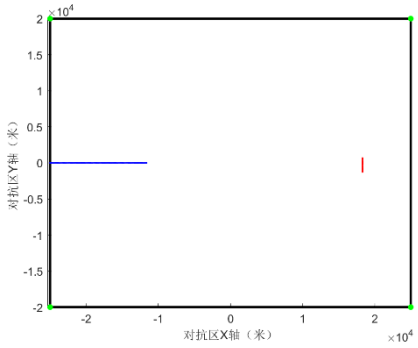


(c) 时刻 3

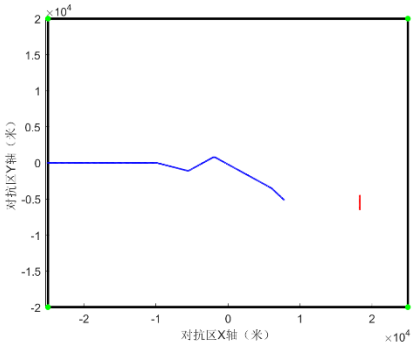


(d) 时刻 4

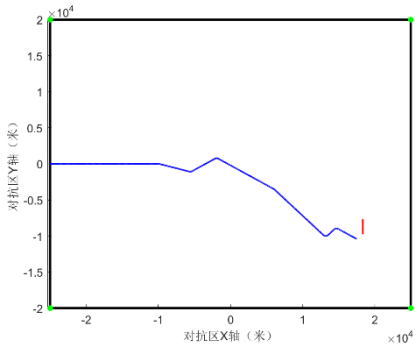
图 5.14 成功拦截实例 2 对抗过程复现



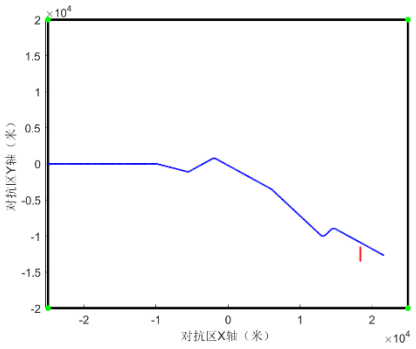
(a)时刻 1



(b) 时刻 2

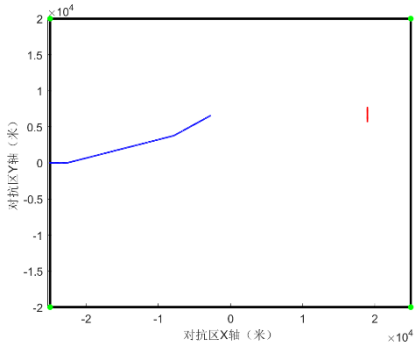


(c) 时刻 3

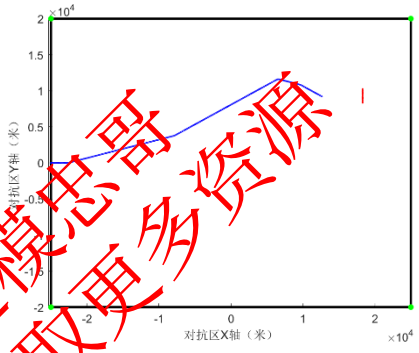


(d) 时刻 4

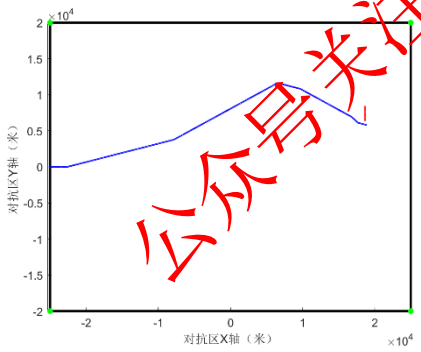
图 5.15 成功突防实例 1 对抗过程复现



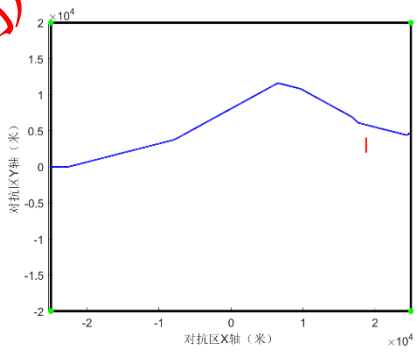
(a)时刻 1



(b) 时刻 2



(c) 时刻 3



(d) 时刻 4

图 5.16 成功突防实例 2 对抗过程复现

5.4.2 围捕型策略

由于敌机的机动变幻莫测，通过转弯截击法形成截击三角形后，敌机可能会再次进行机动，使新的截击三角形被再次破坏。基于转弯截击法，无论敌机如何机动，只要能够构造出新的截击三角形（敌机回转大于 90 度时无法形成截击三角形），我机就能够迅速通过机动指向最新的截击点。结合转弯截击法和追踪拦截法，我们提出了基于转弯截击法的追踪拦截控制，其控制流程如图 5.17 所示。

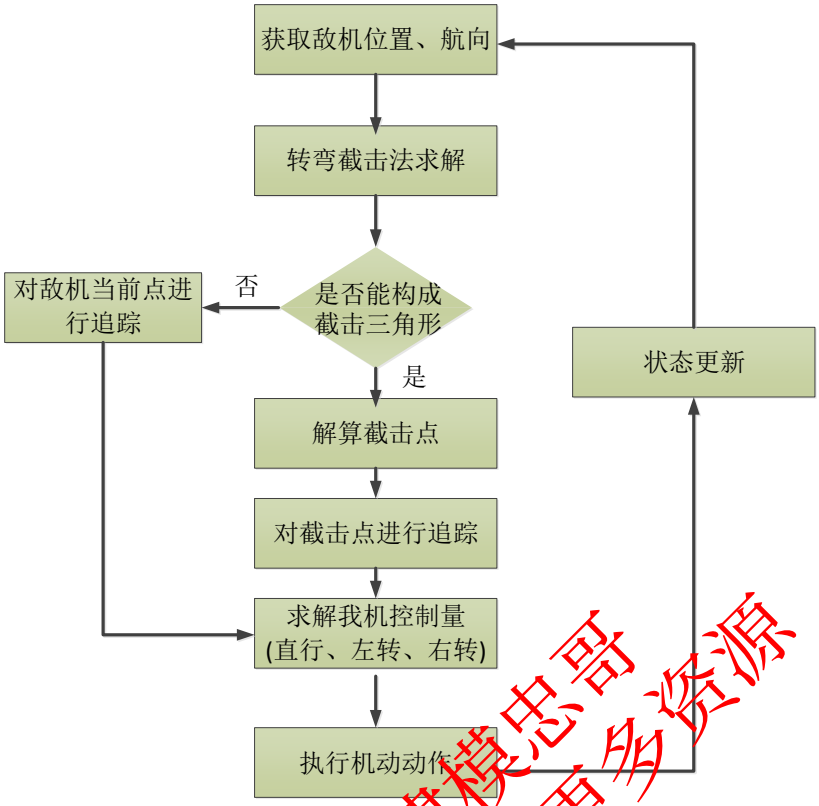


图 5.17 基于转弯截击法的追踪拦截控制流程

基于转弯截击法的追踪拦截控制流程如下：

Step1：根据传感器信息获取敌机位置和航向。

Step2：利用转弯截击法进行截击三角形解算，若不存在截击三角形，则将敌机当前位置设置为追踪点，转到 Step4；若存在截击三角形，转到 Step3。

Step3：求解拦截点坐标，将拦截点设置为追踪点。

Step4：根据我机位置、航向和追踪点坐标，求解我机控制量（直行、左转、右转）。

Step5：执行机动控制量，进行状态更新，转到 Step1。直到成功拦截敌机，算法停止。

本文提出的基于转弯截击法的追踪拦截控制是实现进攻策略中的核心方法。当多个无人机集群从不同角度对蓝方无人机形成截击三角形时，此时不仅仅是数量上的优势，而是能够**实现多个集群之间的多角度时间协同包围，从一定程度上弥补了平台性能上的差距，进而实现了无人机集群在能力上的涌现**。

基于前文的推论，可以在 t=35s 时，在拦截线 x=19km 上完成两个波次无人机集群的布设，FY01 布设的两个波次无人机集群分别记为 Swarm1 和 Swarm3，FY02 布设的两个波次无人机集群分别记为 Swarm2 和 Swarm4，布设的集群中心点坐标分别记为 $(19, y_{si})(i=1,2,3,4)$。每个集群布设的位置可以根据需要指定，但集群的布设位置对拦截效果起到了重要作用。

在围捕型策略中，我们将释放红方无人机集群的运动范围，不再将其限制在拦截线上进行垂直运动。在问题三模型中，我们建立了基于转弯截击法的追踪拦截控制模型，这一模型考虑了对敌方未来运动的预测，并避免了由于速度劣势造成被蓝方无人机置尾的情况。因此，我们在进攻策略中采取这种方法建立红方无人机集群的机动策略。借助四个红方无人机集群的协同优势，通过调整每个集群的初始位置，实现多集群多角度动态围捕策略，最大程度上封锁蓝方无人机可能的机动摆脱。

在通道宽度固定为 M=70km 的情况，我们需要寻找到一种红方集群的最优初始布设位置，使得蓝方无人机的突防概率尽可能小：

$$\left(y_{s1}^{*}, y_{s2}^{*}, y_{s3}^{*}, y_{s4}^{*}\right) = \operatorname{argmin} \eta \tag{5.9}$$

其中，η为蓝方无人机的突防概率。

由于难以穷尽蓝方无人机在不同策略下的突防策略，这里我们采用随机策略的方式，在大规模蒙特卡洛仿真下进行模拟。在每次仿真中，为蓝方无人机设定随机的转弯次数、转弯时刻和转弯角度，从而生成大规模的蓝方随机突防样本。下面，随机选取其中一次突防失败仿真进行分析，如图 5.18 所示。

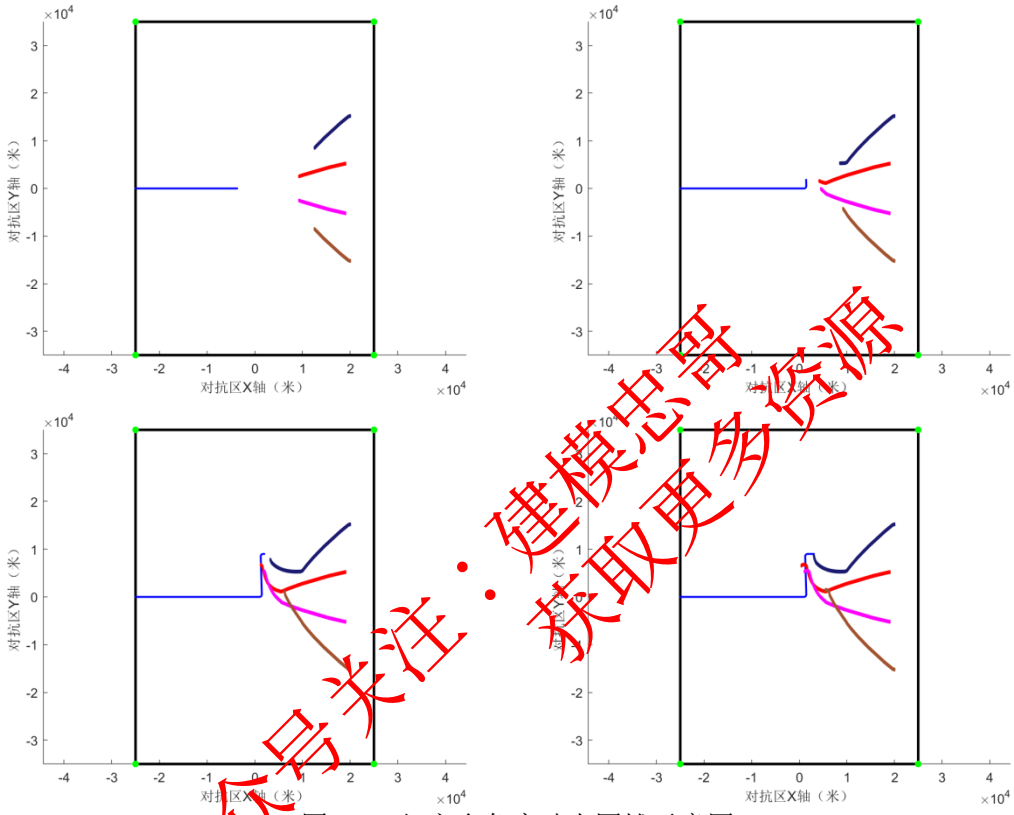


图 5.18 红方多角度动态围捕示意图

从仿真图中可以看到，在距离较远的情况下，蓝方无人机采取直线飞行方式，以尽可能缩短突防时间。红方无人机则采用多集群多角度动态围捕的策略，构成截击三角形态势向蓝方靠近。当蓝方无人机发现红方无人机集群距离较近时，立即采用一个左转 90 度机动，期望利用速度优势摆脱红方的追击。由于红方无人机布设分散，位于两侧的红方无人机集群并没有丢失蓝方目标，仍然能够可以利用转弯角速度的优势迅速回转，立刻形成新的截击三角形态势。

在蓝方策略相同的情况下，若蓝方利用速度优势摆脱的飞行时间足够长，则两侧的红方无人机也无法对其进行拦截，如图 5.19 所示。

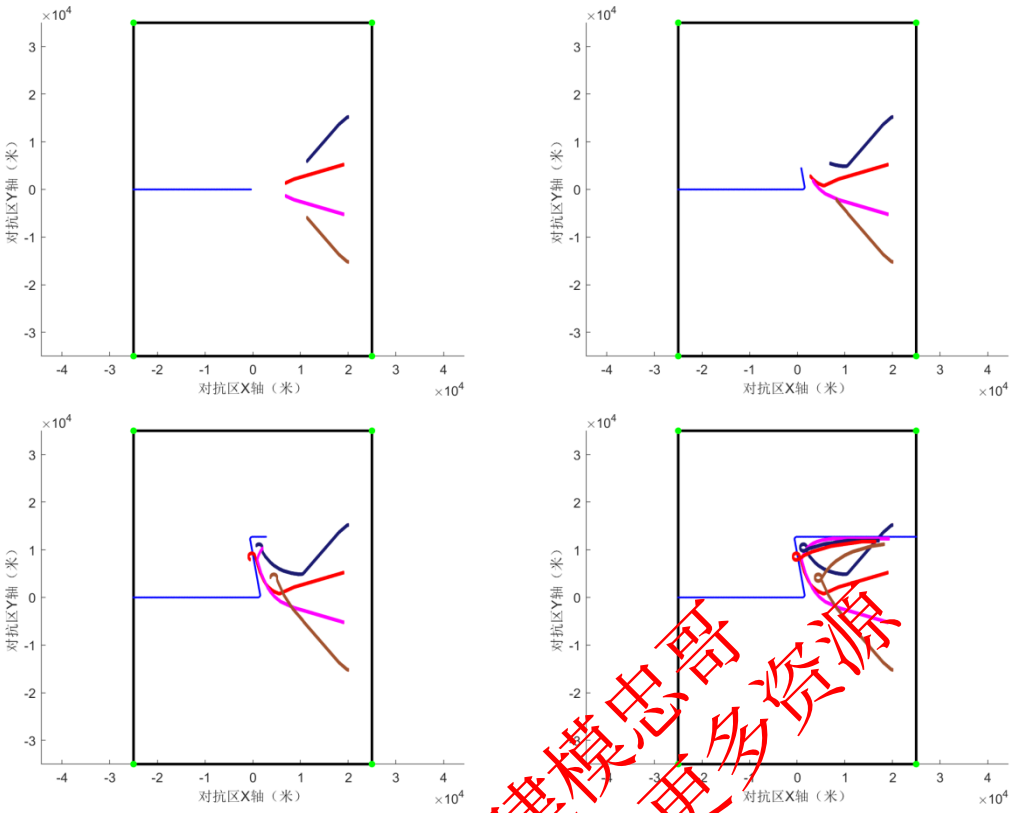


图 5.19 蓝方垂直机动摆脱示意图

对比以上两次仿真，我们可以清楚的看到，蓝方的机动摆脱依赖于通道带宽 M。经过多次仿真，我们发现，蓝方采取垂直机动策略是最有效的，而在其他策略下都会被红方无人机集群成功拦截。在通道带宽不断缩小的情况下，留给蓝方垂直机动摆脱的距离会不断缩小，直到达到通道带宽的上限 M_{max} 时，蓝方将不再能够实现突防。最后，我们通过仿真得到 M_{max}=21.04km，当 $M<M_{max}$ 时，因为预留给蓝方的摆脱距离过短，导致蓝方超出边界或被拦截。

6 问题四的建模与求解

6.1 问题分析

在问题四中，红蓝双方的无人机集群规模都有所增大，对红方而言由于在速度上处于劣势，因此更需要充分发挥数量优势以及数量优势带来的协同策略优势对蓝方进行拦截。

又由于突防通道带宽较大，红方平台中运载机相对无人机的速度更快，因此需要充分运用红方运载机的速度优势，将第二波无人机集群布设在合理的位置，形成拦截蓝方无人机的有利态势。

此外，本章中目标分配是增强拦截效能的关键问题，合理的目标分配机制可以形成灵活的围捕态势，避免盲目分配目标导致的冲突和混乱。尤其是红蓝态势是强对抗博弈时变的，传统的静态任务分配方法无法满足实时更新调整的需要，因此本章引入动态的联盟组建机制，根据红蓝态势进行自适应的联盟重组，最大化布势效率和拦截概率。

6.2 模型建立

6.2.1 自适应围捕策略

在问题四中，红蓝双方的决策变量空间维度非常大，相比问题三更加复杂。仿真系统为我们提供了有力的支撑，我们对大部分变量进行了随机设置，对部分变量进行人为取值，

通过仿真寻找其内在规律。

在初步的仿真中，考虑蓝方在 AB 上的初始位置随机生成，其突防航线的转弯次数、转弯时间和转弯角度随机。红方在 x=19km 的拦截线上均匀分布，并采用基于转弯截击法的追踪控制方法对蓝方实施拦截。在目标分配方面，初始时刻为每个无人机集群随机设置一个蓝方目标，如图 6.1 所示。

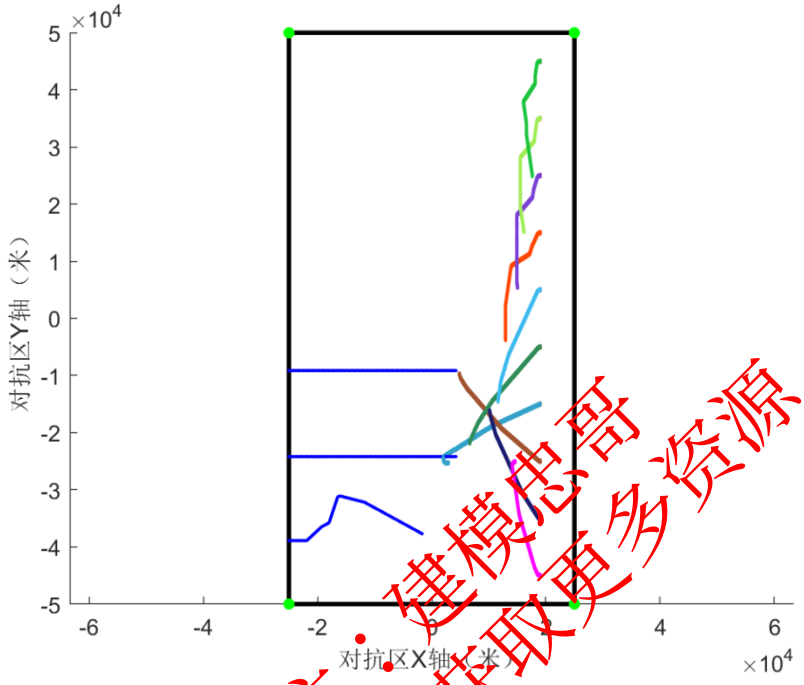


图 6.1 随机目标分配示意图

通过多次仿真发现，由于通道带宽为 100km，为蓝方突防创造了良好的条件。若不进行有效的目标分配，则容易造成初始时刻敌我双方距离较远的情况，造成拦截十分困难。因此，**必须要进行有效的动态目标分配**，而且目标分配不能根据初始状态直接固定，还需要在拦截过程中进行动态分配，确保集群战术运用的灵活性。

对于红方而言，在第一波次无人机集群布设时，由于初始蓝方无人机未知，应采取均匀分布的方式，提高平均拦截概率。而第二波次的布放时机和位置则需要根据蓝方无人机的运动来确定。最后，我们给出了一种自适应围捕策略，主要遵守以下规则：

（1）红方运载机 FY01-FY05 在 CD 上均匀分布，其释放的第一波次无人机（均为 5 架）中心位置在释放线 x=23km 上均匀分布。第一波次完成释放后，通过一定的动态分配原则将无人机集群 Swarm1-Swarm5 与蓝方无人机 HT01-HT03 对应，并采用基于转弯截击法的追踪拦截控制模型进行动态围捕。

（2）红方运载机 FY01-FY05 完成第一波释放后，立即在边界 CD 上盘旋飞行，且与通过一定的分配原则与蓝方无人机 HT01-HT03 对应。当红方运载机与其对应的蓝方无人机在水平方向上达到一致时，则立即释放第二波无人机（均为 5 架），其释放时间和释放位置取决于蓝方无人机的初始位置和运动情况。释放的第二波次无人机与第一波次相同，完成动态目标分配后立即进行动态围捕。

（3）在完成两个波次的无人机集群布设后，采取基于联盟机制的目标分配算法，能够将 10 个无人机集群划分为 3 个联盟，与蓝方无人机 HT01-HT03 对应。联盟组建完成后，在短时间或蓝方无机动的情况下是稳定的。一旦蓝方出现大机动或部分兵力损失的情况下，联盟要相应的进行重新分配。在对抗过程中，红方无人机集群必须根据态势情况进行自适应联盟组建和目标分配。

6.2.2 “直线接近—临界机动”突防策略

由于蓝方最优突防策略难以进行确定性的解析表达，因此本题给出蓝方突防的一种策略思想，即由于蓝方无人机具有飞行速度优势，因此蓝方突防的最好方法就是最大化的利用其速度优势。基于此思想，为蓝方设计了一种“直线接近—临界机动”策略，将自身速度优势转化为空间优势，从而实现突防。

具体策略为：

1.首先进行威胁评估，将离蓝方无人机最近的红方无人机作为最大威胁源；

2.当距离红方飞机达到一定遭遇距离 d_m 时实施转弯机动，转过的偏航角为 90°；

3.转弯机动结束后继续直线飞行，至少飞过 t_P 秒之后再次实行 90°转弯机动，即可成功摆脱红方无人机。

下面给出 t 的计算公式：

$$v_E t_P + R_E > v_P(t_P - \frac{\pi R_E}{2v_E}) \tag{6.1}$$

解得 t 的时间下限为 2.6 秒。

“直线接近—临界机动”策略如图 6.2 所示。

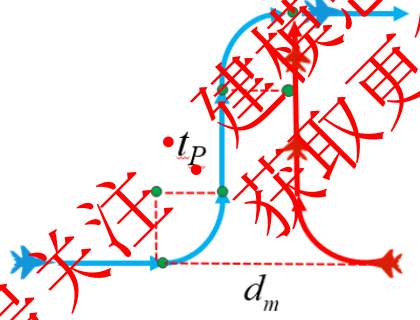


图 6.2 蓝方无人机“直线接近—临界机动”策略示意图

图 6.3 给出仿真系统中蓝方无人机实施“直线接近—临界机动”策略时的突防效果图。

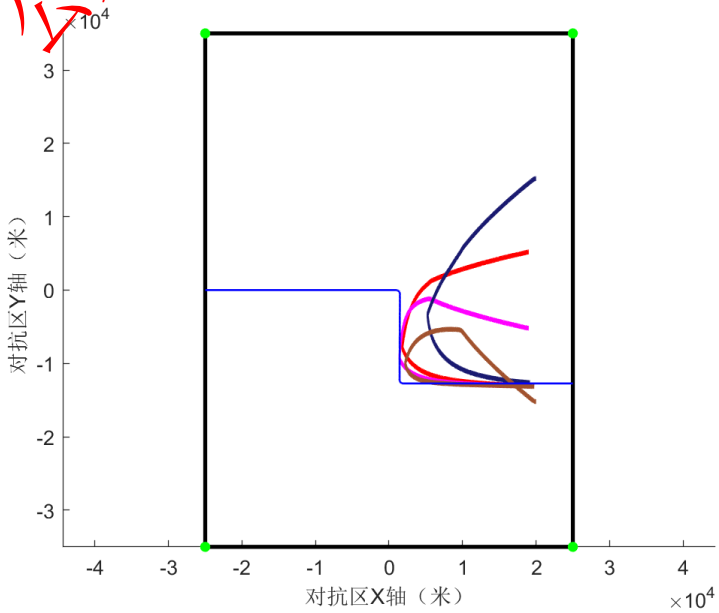


图 6.3 蓝方无人机“直线接近—临界机动”策略突防效果图

从突防效果图中可以看出，红方无人机从各个角度来袭拦截，均朝向拦截点进行飞行，此时蓝方无人机在离红方无人机一定距离时按照“直线接近—临界机动”策略实施 90°机动转弯，转弯结束后保持向前直线飞行一定时间，将飞行速度优势转化为空间优势，在空间上达到摆脱红方无人机的条件后再实施一次 90°机动转弯，成功摆脱，完成突防。图 6.3 证明了蓝方采用“直线接近—临界机动”策略进行突防时的有效性。

6.3 模型解算及结果分析

联盟任务分配问题[5]的目标时将包含 N_T 个任务的任务集 $J \triangleq \{1,...,N_T\}$ 分配给 N_a 个智能体的集合 $I \triangleq \{1,...,N_a\}$，智能体 $i \in I$ 最多可执行 L_i 个任务。任务 $j \in J$ 由四元素组 $< pos_j, tae_j, num_j, tc_j >$ 表示，其中 pos_j 为任务坐标；tae_j 为执行该任务所需载荷资源；num_j 为完成任务 j 所需智能体数量；$tc_j = \{\tau_j^{start}, \tau_j^{end}, dur_j, \lambda_j\}$ 为任务时间约束集合，其中 τ_j^{start} 和 τ_j^{end} 分别为任务 j 执行最早时最晚时间节点，dur_j 为任务持续时间，$0 \le \lambda_j < 1$ 为任务时间折扣因子。

多智能体联盟分布式任务分配问题数学模型为：

$$\begin{aligned} &\max \sum_{i=1}^{N_a}(\sum_{i=1}^{N_r} r_{ij} x_{ij}) \\ &s.t. \sum_{i=1}^{N_r} x_{ij} \le L_i \quad \sum_{i=1}^{N_a} x_{ij} \le num_i \quad x_{ij} \in \{1,0\}, \forall (i,j) \in I \times J \end{aligned} \tag{6.2}$$

式中 r_{ij} 为智能体 i 执行任务 j 的收益，$x_{ij} \in \{1,0\}$ 为任务分配决策变量 x_{ij}=1 表明智能体 i 执行任务 j，否则 x_{ij}=0。

基于联盟机制的任务分配流程如图 6.4 所示：

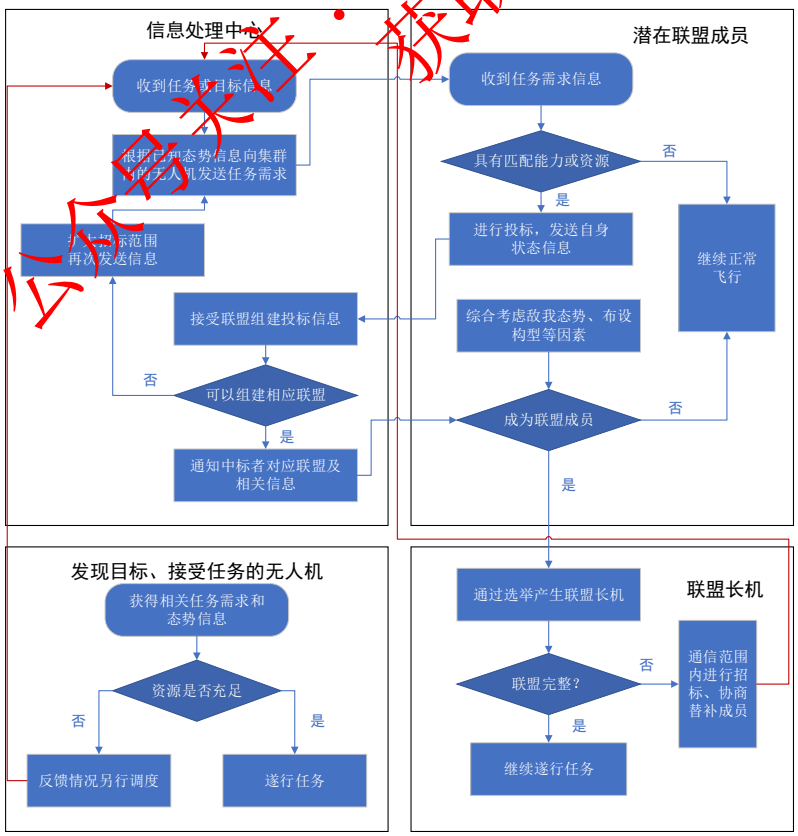
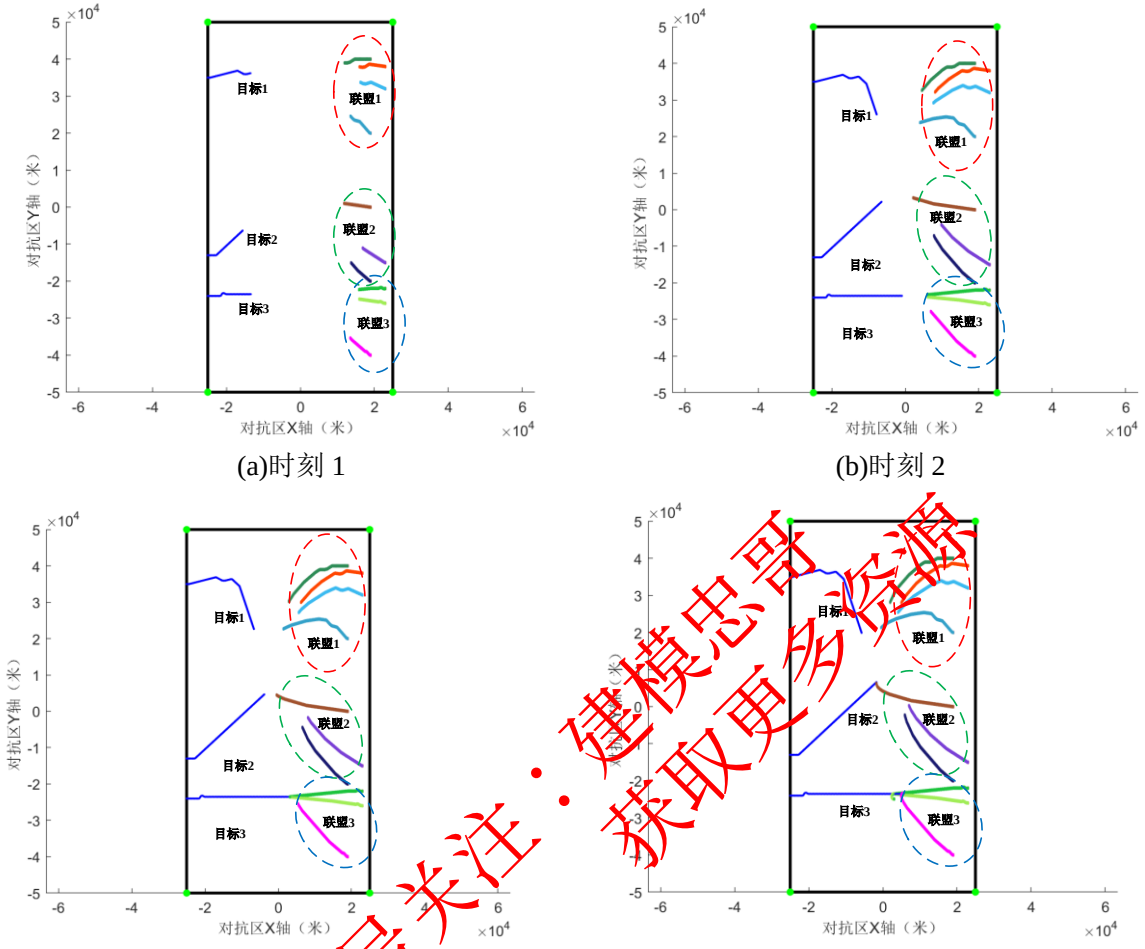


图 6.4 基于联盟机制的任务分配流程

多智能体联盟任务分配策略，针对三架蓝方无人机目标，将 10 架运载机分为 3 个联盟，其中联盟 1 包含 4 架运载机，联盟 2 和联盟 3 各包含 3 架运载机，三个联盟各分配一

个目标进行拦截，联盟 1、2、3 分配的目标分别为目标 1、2、3。图 6.5 给出了 3 个目标下基于动态联盟的任务分配对抗过程。



(a)时刻 1　　(b)时刻 2

(c)时刻 3　　(d)时刻 4

图 6.5 基于动态联盟的任务分配对抗过程

7 模型的评价与改进方向

7.1 模型的优点

（1）分析了基于战术意图的红蓝双方博弈策略，给出了红方无人机集群最优布设准则和蓝方无人机集群最优机动准则，采用从特殊到一般的思想，将突防区划分为垂直突防和斜向突防，体现了协同对抗的博弈特点。

（2）通过数学推导严格证明了实现“无论红方无人机采用什么样的追击策略，蓝方无人机总能采用合适的策略以躲避红方的拦截，实现成功突防”的充分必要条件，得到的结论有严谨的理论支撑。

（3）求解出的突防区有解析解，其边界可用公式表达，易于表述和计算，在应用上易于扩展。

7.2 模型的缺点

（1）本文的问题求解是基于一定合理假设进行的，相比实际运用而言进行了问题简化。

（2）当约束条件更为复杂、作战区域不规则或集群规模较大的情况下，运用解析法进行计算的运算量会增大。

7.3 模型的改进

模型的改进主要从两方面进行：

（1）对模型本身的改进:在约束条件上进一步向实际情况靠拢，细化模型的各项假设，在参数设置上更加逼近真实值。

（2）对模型解算方法的改进：主要从算法复杂度、算法精度上进行进一步提升。

参考文献

[1] 梁晓龙,何吕龙,张佳强等.航空集群构型控制及其演化方法[J].中国科学:技术科学,2019,49(03):277-287.

[2] 梁晓龙，孙强，尹忠海等. 大规模无人系统集群智能控制方法综述[J]. 计算机应用研究, 2015(01):17-22.

[3] 侯岳奇，梁晓龙，何吕龙,等. 未知环境下无人机集群协同区域搜索算法[J]. 北京航空航天大学学报, 2019, 45(2):347-356.

[4] 崔荣华.基于矢量加法的引导拦截解算[J].兵工自动化,2005(01):5-6.

[5] 唐嘉钰，李相民，代进进，薄宁. 复杂约束条件下异构多智能体联盟任务分配[J]. 控制理论与应用.

[6] 朱梦圆，吕娜，陈柯帆. 航空集群混合多层式联盟组建方法[J]. 兰州大学学报(自然科学版), 2019(3).

附　录

附录一：仿真系统核心代码

ClassSwarm.m 集群类

```
classdef ClassSwarm
    %    CLASSSWARM 此处显示有关此类的摘要
    %    这是一个集群类
    %    包含集群的属性：智能体数量，智能体参数，集群参数等
    %    包含集群的功能：集群状态更新，集群转弯，集群态势显示等

    properties
        N   % 集群中个体数量
        agent % 集群中个体的参数
        isTurn % 编队当前是否在转弯 -1 左转 0 直行 1 右转
        turnTime % 转弯次数
        maxTurnTime % 最大转弯次数
        r       % 转弯半径
        w       % 转弯角速度
        velo    % 速度
    end

    methods
        function obj = ClassSwarm(N,state,vel,minr)
            global dT
            obj.velo = vel;
            obj.r = minr;
            obj.w = obj.velo/obj.r;
            obj.N = N;
            obj.isTurn = 0;
            obj.turnTime = 0;
            obj.maxTurnTime = pi/(obj.w*dT);
            for i = 1:N
                obj.agent = [obj.agent,ClassAgent(i,state(:,i))];
            end
        end

        function [output1,output2,output3] = UpdateState(obj) % 更新平台状态信息
            global dT
            if obj.isTurn == -1 || obj.isTurn == 1
                if obj.turnTime <obj.maxTurnTime
                    obj.turnTime = obj.turnTime+1;
                else
                    obj.isTurn = 0;
                    obj.turnTime = 0;
                end
            end
            for i = 1:obj.N
                obj.agent(i).yaw = [obj.agent(i).yaw,
mc(obj.agent(i).yaw(end)+obj.isTurn*dT*obj.w*180/pi)];
                direct = [ sin(obj.agent(i).yaw(end)*pi/180) ;
cos(obj.agent(i).yaw(end)*pi/180) ];
                obj.agent(i).pos = [obj.agent(i).pos,
obj.agent(i).pos(:,end)+dT*obj.velo*direct];
            end
            output1 = obj.agent;
            output2 = obj.isTurn;
            output3 = obj.turnTime;
        end

        function [output1,output2,output3] = SwarmTurn(obj,isTurn,angle) % 左右转，转弯角
度 angle（度）
```

```
            global dT
            if obj.isTurn == -1 || obj.isTurn == 1 % 若集群正在转弯，则转弯指令不生效。
                output1 = obj.isTurn;
                output2 = obj.turnTime;
                output3 = obj.maxTurnTime;
            else                                                    % 若集群正在平飞，则转弯指令生效
                output1 = isTurn;
                output2 = 0; % turntime
                output3 = (angle*pi/180)/(obj.w*dT);
            end
        end


        function plotPath(obj,k)
            % 绘制轨迹图
            if k > 0
                color =[[1 0 0]
                        [0 0 1]
                        [1 0 1]
                        [0.1 0.1 0.44]
                        [0.63 0.32 0.18]
                        [0.2 0.63 0.79]
                        [0.18 0.55 0.34]
                        [0.23 0.73 0.94]
                        [1 0.27 0]
                        [0.48 0.25 0.84]
                        [0.63 0.93 0.34]
                        [0.11 0.77 0.24]];
                for i = 1:obj.N
                    x = obj.agent(i).pos(1,:);
                    y = obj.agent(i).pos(2,:);
                    plot(x,y,'Color',color(k,:),'LineWidth',1);
                end
            else
                for i = 1:obj.N
                    x = obj.agent(i).pos(1,:);
                    y = obj.agent(i).pos(2,:);
                    plot(x,y,'LineWidth',1);
                end
            end
        end

        function scatterPoint(obj,k)
            % 绘制轨迹图
            if k > 0
                color =[[1 0 0]
                        [0 0 1]
                        [1 0 1]
                        [0.1 0.1 0.44]
                        [0.63 0.32 0.18]
                        [0.2 0.63 0.79]
                        [0.18 0.55 0.34]
                        [0.23 0.73 0.94]
                        [1 0.27 0]
                        [0.48 0.25 0.84]
                        [0.63 0.93 0.34]
                        [0.11 0.77 0.24]];
                for i = 1:obj.N
                    x(i) = obj.agent(i).pos(1,end);
                    y(i) = obj.agent(i).pos(2,end);
                end
                scatter(x,y,1,color(k,:));
```

```
                else
                    for i = 1:obj.N
                        x(i) = obj.agent(i).pos(1,end);
                        y(i) = obj.agent(i).pos(2,end);
                    end
                    scatter(x,y,1);
                end
            end

        end
    end
```

ClassAgent.m 智能体类

```
classdef ClassAgent
    %   CLASSAGENT 此处显示有关此类的摘要
    %    用于储存 Agent 的当前和历史状态信息

    properties
        ID % 编号
        pos   % 位置
        yaw % 航向
    end

    methods
        function obj = ClassAgent(ID,state)
            obj.ID = ID;
            obj.pos = state(1:2);
            obj.yaw = state(3);
        end
    end

end
```

isAttack.m 终止条件

```
function is = isAttack(blue,swarm)
is = 0;
num = 0;
posb = blue.agent.pos(:,end);
for i = 1:swarm.N
    posr = swarm.agent(i).pos(:,end);
    if norm(posr-posb) < 300
        num = num + 1;
    end
end
if num >= 2
    is = 1;
end
end
```

Main.m 运行入口：驱动一次仿真（本代码为问题三一字型防守策略仿真）

```
clc
clear all
close all

%% 参数初始化
global dT
dT = 0.5; % 间隔时间
```

```
L = 50000;
M = 40000;

%% 初始化无人机集群的位置和速度
n1 = 20; % FY01 第一波无人机集群的数量
state1 = []; % x y k  位置和航向
for i = 1:20
    state1(:,i) = [19000;-950+100*(i-1);0];
end
swarm = ClassSwarm(n1,state1,200,350);

n2 = 1; %  蓝方无人机
state2 = [-25000;0;90];
blue = ClassSwarm(n2,state2,250,500);

%% 循环执行  离散计时器
simtime = 0; %  记录仿真次数
for num = 1:1
    simtime = simtime+1;
    index = 0;
    turnpara = [];
    [turntime ,turnpara] = genturn()
    hasturn = 0; %已经转弯的次数
    swarm = ClassSwarm(n1,state1,200,350);
    blue = ClassSwarm(n2,state2,250,500);
    clf
    figure(1)
    hold on;
    axis equal;
    xlabel('对抗区 X 轴（米）');
    ylabel('对抗区 Y 轴（米）');
    set(gcf, 'doublebuffer', 'on') %  避免闪烁
    plotArea(M,L);
    for i = 1:n1
        xswarm(i) = swarm.agent(i).pos(1,end);
        yswarm(i) = swarm.agent(i).pos(2,end);
    end
    plotred = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    plotblue  =  scatter(blue.agent.pos(1,end),blue.agent.pos(2,end),20,'b.');%画 blue 的初始位
置
    for i = 1:ceil(360/dT) %  一次仿真
        index = index + 1; %  计数器
        if hasturn < turntime
            if index*dT > turnpara(hasturn+1,1)
                thisangle = genangle(blue,M,L);
                [blue.isTurn , blue.turnTime, blue.maxTurnTime] =
blue.SwarmTurn(sign(thisangle),abs(thisangle));
                hasturn = hasturn +1;
            end
        end
        [blue.agent , blue.isTurn , blue.turnTime] = blue.UpdateState; %  状态更新

        if index*dT > 35
            if swarm.isTurn == 0 %  若集群此时正在平飞，则判断是否需要转弯掉头
                if isNeedTurn(swarm,blue) == 1 %  若根据策略需要转弯掉头，则执行左转
180 度
                    [swarm.isTurn , swarm.turnTime, swarm.maxTurnTime] =
swarm.SwarmTurn(-1,180);
                end
            end
            [swarm.agent , swarm.isTurn , swarm.turnTime] = swarm.UpdateState; %  状态更
新
        end
```

```
            figure(1)
            for i = 1:n1
                xswarm(i) = swarm.agent(i).pos(1,end);
                yswarm(i) = swarm.agent(i).pos(2,end);
            end
            set(plotred,'xdata',xswarm,'ydata',yswarm);%设置 agent2 的运动过程
            blue.scatterPoint(2)
            drawnow

            if index*dT > 360 || isAttack(blue,swarm) || (blue.agent.pos(2,end)>M/2 ||
blue.agent.pos(2,end)<-M/2) || (blue.agent.pos(1,end)>L/2 || blue.agent.pos(1,end)<-L/2)% 终
止条件
                break
            end

        end
end
```

附录二：红方无人机动态围捕策略

fourSwarnAttack.m

```
clc
clear all
close all

%% 参数初始化
global dT
dT = 0.5; % 间隔时间
L = 50000;
M = 70000;
oneangle = 200/350*dT;

%% 初始化无人机集群的位置和速度
n0 = 1; % 蓝方无人机
state0 = [-25000;0;90];
% blue = ClassSwarm(n0,state0,250,500);

n1 = 5; % FY01 第一波无人机集群的数量
state1 = []; % x y k 位置和航向
for i = 1:5
    state1(:,i) = [19000;5000+100*(i-1);270];
end
% swarm1 = ClassSwarm(n1,state1,200,350);

n2 = 5; % FY02 第一波无人机集群的数量
state2 = []; % x y k 位置和航向
for i = 1:5
    state2(:,i) = [19000;-5000-100*(i-1);270];
end
% swarm2 = ClassSwarm(n2,state2,200,350);

deltax = 1000;
deltay = 10000;
n3 = 5; % FY01 第二波无人机集群的数量
state3 = state1; % x y k 位置和航向
state3(1,:) = state1(1,:) + deltax;
state3(2,:) = state1(2,:) + deltay;
% swarm3 = ClassSwarm(n3,state3,200,350);

n4 = 5; % FY02 第二波无人机集群的数量
state4 = state2; % x y k 位置和航向
```

```
state4(1,:) = state2(1,:) + deltax;
state4(2,:) = state2(2,:) - deltay;
% swarm4 = ClassSwarm(n4,state4,200,350);

%% 循环执行
simtime = 0; % 记录仿真次数
for num = 1:1
    %%%%% 每次仿真开始的时候，进行集群参数初始化 %%%%%
    simtime = simtime+1;
    index = 0;
    turnpara = [];
    [turntime ,turnpara] = genturn()
    hasturn = 0; %已经转弯的次数
    blue = ClassSwarm(n0,state0,250,500);
    swarm1 = ClassSwarm(n1,state1,200,350);
    swarm2 = ClassSwarm(n2,state2,200,350);
    swarm3 = ClassSwarm(n3,state3,200,350);
    swarm4 = ClassSwarm(n4,state4,200,350);
    %%%%% 每次仿真开始的时候，进行集群参数初始化 %%%%%

    %%%%%%%%%%%%% 画图 %%%%%%%%%%%%
    clf
    figure(1)
    hold on;
    % axis([-30000 35000 -40000 45000]);
    axis equal;
    xlabel('对抗区 X 轴（米）');
    ylabel('对抗区 Y 轴（米）');
    set(gcf, 'doublebuffer', 'on') % 避免闪烁
    plotArea(M,L);
    for i = 1:n1
        xswarm(i) = swarm1.agent(i).pos(1,end);
        yswarm(i) = swarm1.agent(i).pos(2,end);
    end
    plotred1 = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    for i = 1:n2
        xswarm(i) = swarm2.agent(i).pos(1,end);
        yswarm(i) = swarm2.agent(i).pos(2,end);
    end
    plotred2 = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    for i = 1:n3
        xswarm(i) = swarm3.agent(i).pos(1,end);
        yswarm(i) = swarm3.agent(i).pos(2,end);
    end
    plotred3 = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    for i = 1:n4
        xswarm(i) = swarm4.agent(i).pos(1,end);
        yswarm(i) = swarm4.agent(i).pos(2,end);
    end
    plotred4 = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    plotblue = scatter(blue.agent.pos(1,end),blue.agent.pos(2,end),20,'b.');%画 blue 的初始位
置
    %%%%%%%%%%%%% 画图 %%%%%%%%%%%%

    for i = 1:ceil(360/dT) % 一次仿真 最大 360s
        %%%%%%%% 蓝方无人机转弯和状态更新处理 %%%%%%%%
        index = index + 1; % 计数器
        if hasturn < turntime
            if index*dT > turnpara(hasturn+1,1)
                thisangle = genangle(blue,M,L);
                [blue.isTurn , blue.turnTime, blue.maxTurnTime] =
blue.SwarmTurn(sign(thisangle),abs(thisangle));
                hasturn = hasturn +1;
```

```
            end
        end
        [blue.agent , blue.isTurn , blue.turnTime] = blue.UpdateState; % 状态更新
        %%%%%%% 蓝方无人机转弯和状态更新处理 %%%%%%%

        %%%% 红方 4 个无人机集群的转弯和状态更新处理 %%%%
        if index*dT > 35
            if swarm1.isTurn == 0 % 若集群此时正在平飞，则判断是否需要转弯掉头
                if isNeedTurn3(swarm1,blue) ~= 0 % 若根据策略需要转弯掉头，则执行左
转 180 度
                    [swarm1.isTurn , swarm1.turnTime, swarm1.maxTurnTime] =
swarm1.SwarmTurn(isNeedTurn3(swarm1,blue),oneangle);
                end
                if isNeedTurn3(swarm2,blue) ~= 0 % 若根据策略需要转弯掉头，则执行左
转 180 度
                    [swarm2.isTurn , swarm2.turnTime, swarm2.maxTurnTime] =
swarm2.SwarmTurn(isNeedTurn3(swarm2,blue),oneangle);
                end
                if isNeedTurn3(swarm3,blue) ~= 0 % 若根据策略需要转弯掉头，则执行左
转 180 度
                    [swarm3.isTurn , swarm3.turnTime, swarm3.maxTurnTime] =
swarm3.SwarmTurn(isNeedTurn3(swarm3,blue),oneangle);
                end
                if isNeedTurn3(swarm4,blue) ~= 0 % 若根据策略需要转弯掉头，则执行左
转 180 度
                    [swarm4.isTurn , swarm4.turnTime, swarm4.maxTurnTime] =
swarm4.SwarmTurn(isNeedTurn3(swarm4,blue),oneangle);
                end
            end
            [swarm1.agent , swarm1.isTurn , swarm1.turnTime] = swarm1.UpdateState; % 状
态更新
            [swarm2.agent , swarm2.isTurn , swarm2.turnTime] = swarm2.UpdateState; % 状
态更新
            [swarm3.agent , swarm3.isTurn , swarm3.turnTime] = swarm3.UpdateState; % 状
态更新
            [swarm4.agent , swarm4.isTurn , swarm4.turnTime] = swarm4.UpdateState; % 状
态更新
        end
        %%%% 红方 4 个无人机集群的转弯和状态更新处理 %%%%

        %%%%%%% 绘制实时轨迹图 %%%%%%%
        figure(1)
        for i = 1:n1
            xswarm(i) = swarm1.agent(i).pos(1,end);
            yswarm(i) = swarm1.agent(i).pos(2,end);
        end
        set(plotred1,'xdata',xswarm,'ydata',yswarm);%设置 swarm1 的运动过程
        for i = 1:n2
            xswarm(i) = swarm2.agent(i).pos(1,end);
            yswarm(i) = swarm2.agent(i).pos(2,end);
        end
        set(plotred2,'xdata',xswarm,'ydata',yswarm);%设置 swarm2 的运动过程
        for i = 1:n3
            xswarm(i) = swarm3.agent(i).pos(1,end);
            yswarm(i) = swarm3.agent(i).pos(2,end);
        end
        set(plotred3,'xdata',xswarm,'ydata',yswarm);%设置 swarm2 的运动过程
        for i = 1:n4
            xswarm(i) = swarm4.agent(i).pos(1,end);
            yswarm(i) = swarm4.agent(i).pos(2,end);
        end
        set(plotred4,'xdata',xswarm,'ydata',yswarm);%设置 swarm2 的运动过程
        blue.scatterPoint(2)
```

```
            drawnow
            %%%%%%%% 绘制实时轨迹图 %%%%%%%%

            %%%%%%%% 给定终止条件 %%%%%%%%
            if index*dT > 360 || isAttack(blue,swarm1) || isAttack(blue,swarm2) || isAttack(blue,swarm3) || isAttack(blue,swarm4) || (blue.agent.pos(2,end)>M/2 || blue.agent.pos(2,end)<-M/2) || (blue.agent.pos(1,end)>L/2 || blue.agent.pos(1,end)<-L/2)% 终止条件
                break
            end
            %%%%%%%% 给定终止条件 %%%%%%%%
        end
end
```

附录三：蓝方无人机赛跑突防策略

fourSwarmAttackbuleEscape.m

```
clc
clear all
close all

%% 参数初始化
global dT
dT = 0.5; % 间隔时间
L = 50000;
M = 70000;
oneangle = 200/350*dT;

%% 初始化无人机集群的位置和速度
n0 = 1; % 蓝方无人机
state0 = [-25000;0;90];
% blue = ClassSwarm(n0,state0,250,500);

n1 = 5; % FY01 第一波无人机集群的数量
state1 = []; % x y k 位置和航向
for i = 1:5
    state1(:,i) = [19000;5000+100*(i-1);270];
end
% swarm1 = ClassSwarm(n1,state1,200,350);

n2 = 5; % FY02 第一波无人机集群的数量
state2 = []; % x y k 位置和航向
for i = 1:5
    state2(:,i) = [19000;-5000-100*(i-1);270];
end
% swarm2 = ClassSwarm(n2,state2,200,350);

deltax = 1000;
deltay = 10000;
n3 = 5; % FY01 第二波无人机集群的数量
state3 = state1; % x y k 位置和航向
state3(1,:) = state1(1,:) + deltax;
state3(2,:) = state1(2,:) + deltay;
% swarm3 = ClassSwarm(n3,state3,200,350);

n4 = 5; % FY02 第二波无人机集群的数量
state4 = state2; % x y k 位置和航向
state4(1,:) = state2(1,:) + deltax;
state4(2,:) = state2(2,:) - deltay;
% swarm4 = ClassSwarm(n4,state4,200,350);

%% 循环执行
```

```
simtime = 0; %  记录仿真次数
for num = 1:1
    %%%%% 每次仿真开始的时候，进行集群参数初始化 %%%%%
    simtime = simtime+1;
    index = 0;
    turnpara = [];
    [turntime ,turnpara] = genturn()
    hasturn = 0; %已经转弯的次数
    blue = ClassSwarm(n0,state0,250,500);
    swarm1 = ClassSwarm(n1,state1,200,350);
    swarm2 = ClassSwarm(n2,state2,200,350);
    swarm3 = ClassSwarm(n3,state3,200,350);
    swarm4 = ClassSwarm(n4,state4,200,350);
    %%%%% 每次仿真开始的时候，进行集群参数初始化 %%%%%

    %%%%%%%%%%%% 画图 %%%%%%%%%%%
    clf
    figure(1)
    hold on;
    % axis([-30000 35000 -40000 45000]);
    axis equal;
    xlabel('对抗区 X 轴（米）');
    ylabel('对抗区 Y 轴（米）');
    set(gcf, 'doublebuffer', 'on') %  避免闪烁
    plotArea(M,L);
    for i = 1:n1
        xswarm(i) = swarm1.agent(i).pos(1,end);
        yswarm(i) = swarm1.agent(i).pos(2,end);
    end
    plotred1 = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    for i = 1:n2
        xswarm(i) = swarm2.agent(i).pos(1,end);
        yswarm(i) = swarm2.agent(i).pos(2,end);
    end
    plotred2 = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    for i = 1:n3
        xswarm(i) = swarm3.agent(i).pos(1,end);
        yswarm(i) = swarm3.agent(i).pos(2,end);
    end
    plotred3 = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    for i = 1:n4
        xswarm(i) = swarm4.agent(i).pos(1,end);
        yswarm(i) = swarm4.agent(i).pos(2,end);
    end
    plotred4 = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    plotblue  =  scatter(blue.agent.pos(1,end),blue.agent.pos(2,end),20,'b.');%画 blue 的初始位
置
    %%%%%%%%%%%% 画图 %%%%%%%%%%%

    isblueturn1 = true;
    isblueturn2 = false;
    for i = 1:ceil(360/dT) %  一次仿真  最大 360s
        %%%%%%%% 蓝方无人机转弯和状态更新处理 %%%%%%%%
        index = index + 1; %  计数器
        if  norm(swarm1.agent(3).pos(:,end)-blue.agent.pos(:,end))<5000  &&  isblueturn1  ==
true
            [blue.isTurn , blue.turnTime, blue.maxTurnTime] = blue.SwarmTurn(1,90);
            tnow = index*dT;
            isblueturn1 = false;
            isblueturn2 = true;
        end
        if isblueturn2
            if (index*dT - tnow) > 50
```

```
                [blue.isTurn , blue.turnTime, blue.maxTurnTime] = blue.SwarmTurn(-1,90);
                isblueturn2 = false;
            end
        end

        [blue.agent , blue.isTurn , blue.turnTime] = blue.UpdateState; % 状态更新
        %%%%%%% 蓝方无人机转弯和状态更新处理 %%%%%%%

        %%%% 红方 4 个无人机集群的转弯和状态更新处理 %%%%
        if index*dT > 35
            if swarm1.isTurn == 0 % 若集群此时正在平飞，则判断是否需要转弯掉头
                if isNeedTurn3(swarm1,blue) ~= 0 % 若根据策略需要转弯掉头，则执行左
转 180 度
                    [swarm1.isTurn , swarm1.turnTime, swarm1.maxTurnTime] =
swarm1.SwarmTurn(isNeedTurn3(swarm1,blue),oneangle);
                end
                if isNeedTurn3(swarm2,blue) ~= 0 % 若根据策略需要转弯掉头，则执行左
转 180 度
                    [swarm2.isTurn , swarm2.turnTime, swarm2.maxTurnTime] =
swarm2.SwarmTurn(isNeedTurn3(swarm2,blue),oneangle);
                end
                if isNeedTurn3(swarm3,blue) ~= 0 % 若根据策略需要转弯掉头，则执行左
转 180 度
                    [swarm3.isTurn , swarm3.turnTime, swarm3.maxTurnTime] =
swarm3.SwarmTurn(isNeedTurn3(swarm3,blue),oneangle);
                end
                if isNeedTurn3(swarm4,blue) ~= 0 % 若根据策略需要转弯掉头，则执行左
转 180 度
                    [swarm4.isTurn , swarm4.turnTime, swarm4.maxTurnTime] =
swarm4.SwarmTurn(isNeedTurn3(swarm4,blue),oneangle);
                end
            end
            [swarm1.agent , swarm1.isTurn , swarm1.turnTime] = swarm1.UpdateState; % 状
态更新
            [swarm2.agent , swarm2.isTurn , swarm2.turnTime] = swarm2.UpdateState; % 状
态更新
            [swarm3.agent , swarm3.isTurn , swarm3.turnTime] = swarm3.UpdateState; % 状
态更新
            [swarm4.agent , swarm4.isTurn , swarm4.turnTime] = swarm4.UpdateState; % 状
态更新
        end
        %%%% 红方 4 个无人机集群的转弯和状态更新处理 %%%%

        %%%%%%% 绘制实时轨迹图 %%%%%%%
        figure(1)
        for i = 1:n1
            xswarm(i) = swarm1.agent(i).pos(1,end);
            yswarm(i) = swarm1.agent(i).pos(2,end);
        end
        set(plotred1,'xdata',xswarm,'ydata',yswarm);%设置 swarm1 的运动过程
        for i = 1:n2
            xswarm(i) = swarm2.agent(i).pos(1,end);
            yswarm(i) = swarm2.agent(i).pos(2,end);
        end
        set(plotred2,'xdata',xswarm,'ydata',yswarm);%设置 swarm2 的运动过程
        for i = 1:n3
            xswarm(i) = swarm3.agent(i).pos(1,end);
            yswarm(i) = swarm3.agent(i).pos(2,end);
        end
        set(plotred3,'xdata',xswarm,'ydata',yswarm);%设置 swarm2 的运动过程
        for i = 1:n4
            xswarm(i) = swarm4.agent(i).pos(1,end);
            yswarm(i) = swarm4.agent(i).pos(2,end);
```

```
            end
            set(plotred4,'xdata',xswarm,'ydata',yswarm);%设置 swarm2 的运动过程
            set(plotblue,'xdata',blue.agent.pos(1,end),'ydata',blue.agent.pos(2,end));%设置 agent2
的运动过程
            blue.scatterPoint(2)
            %%%%%%% 绘制实时轨迹图 %%%%%%%

            %%%%%%% 给定终止条件 %%%%%%%
            if index*dT > 360 || isAttack(blue,swarm1) || isAttack(blue,swarm2) ||
isAttack(blue,swarm3) || isAttack(blue,swarm4) || (blue.agent.pos(2,end)>M/2 ||
blue.agent.pos(2,end)<-M/2) || (blue.agent.pos(1,end)>L/2 || blue.agent.pos(1,end)<-L/2)% 终
止条件
                break
            end
            %%%%%%% 给定终止条件 %%%%%%%
        end
end
```

附录四：红方无人机集群动态联盟组建与协同围捕仿真

DynamicAlliance.m

```
clc
clear all
close all

%% 参数初始化
global dT
dT = 0.5; % 间隔时间
L = 50000;
M = 100000;
oneangle = 200/350*dT;

%% 初始化无人机集群的位置和速度
blue = [];
n0 = 3;
bstate = [-25000 -25000 -25000
                    35000    -13000    -24000
                    90          90          90];

swarm = [];
n1 = 10; % FY01 第一波无人机集群的数量
rstate = [19000 19000 19000 19000 19000 23000 23000 23000 23000 23000
                    10000    30000 50000 70000 90000 82000 88000 35000 24000 28000
                    270      270      270      270      270      270      270      270
270      270]; % x y k 位置和航向

%% 循环执行
simtime = 0; % 记录仿真次数
for num = 1:1
    %%%%% 每次仿真开始的时候，进行集群参数初始化 %%%%%
    simtime = simtime+1;
    index = 0;
    turnpara1 = [];
    turnpara2 = [];
    turnpara3 = [];
    [turntime1 ,turnpara1] = genturn()
    [turntime2 ,turnpara2] = genturn()
    [turntime3 ,turnpara3] = genturn()
    turntime = [turntime1,turntime2,turntime3];
    hasturn = [0 0 0]; %已经转弯的次数
    blue = [];
    for i = 1:n0
```

```
        blue = [blue ClassSwarm(1,bstate(:,i),250,500)];
    end
    swarm = [];
    for i = 1:n1
        state = [rstate(:,i) rstate(:,i) rstate(:,i) rstate(:,i) rstate(:,i)];
        state(2,3) = state(2,3) + 0 -50000;
        state(2,1) = state(2,3) + 200;
        state(2,2) = state(2,3) + 100;
        state(2,4) = state(2,3) - 100;
        state(2,5) = state(2,3) - 200;
        swarm = [swarm ClassSwarm(5,state,200,350)];
    end
    %%%%% 每次仿真开始的时候，进行集群参数初始化 %%%%%

    %%%%%%%%%%%%% 画图 %%%%%%%%%%%%
    clf
    figure(1)
    hold on;
    % axis([-30000 35000 -40000 45000]);
    axis equal;
    xlabel('对抗区 X 轴（米）');
    ylabel('对抗区 Y 轴（米）');
    set(gcf, 'doublebuffer', 'on') % 避免闪烁
    plotArea(M,L);
    plotblue = [];
    plotred2 = [];
    for i = 1:n0
        plotblue(i) = scatter(blue(i).agent.pos(1,end),blue(i).agent.pos(2,end),20,'b.');%画 blue
的初始位置
    end
    for j = 1:n1
        for i = 1:5
            xswarm(i) = swarm(j).agent(i).pos(1,end);
            yswarm(i) = swarm(j).agent(i).pos(2,end);
        end
        plotred(j) = scatter(xswarm,yswarm,20,'r.');%画 red swarm 的初始位置
    end
    %%%%%%%%%%%%% 画图 %%%%%%%%%%%%

    for i = 1:ceil(360/dT),% 一次仿真 最大 360s
        %%%%%%% 蓝方无人机转弯和状态更新处理 %%%%%%%
        for i = 1:n0
            if i == 1
                turnpara = turnpara1;
            elseif i ==2
                turnpara = turnpara2;
            elseif i == 3
                turnpara = turnpara3;
            end
            index = index + 1; % 计数器
            if hasturn(i) < turntime(i)
                if index*dT > turnpara(hasturn+1,1)
                    thisangle = genangle(blue(i),M,L);
                    [blue(i).isTurn , blue(i).turnTime, blue(i).maxTurnTime] =
blue(i).SwarmTurn(sign(thisangle),abs(thisangle));
                    hasturn = hasturn +1;
                end
            end
            [blue(i).agent , blue(i).isTurn , blue(i).turnTime] = blue(i).UpdateState; % 状态更
新
        end
        %%%%%%% 蓝方无人机转弯和状态更新处理 %%%%%%%
```

```
        %%%% 红方 10 个无人机集群的转弯和状态更新处理 %%%%
        if index*dT > 35
            for i = [1,9,10]
                if isNeedTurn3(swarm(i),blue(3)) ~= 0 % 若根据策略需要转弯掉头，则执行左转 180 度
                    [swarm(i).isTurn , swarm(i).turnTime, swarm(i).maxTurnTime] = swarm(i).SwarmTurn(isNeedTurn3(swarm(i),blue(3)),oneangle);
                end
                [swarm(i).agent , swarm(i).isTurn , swarm(i).turnTime] = swarm(i).UpdateState; % 状态更新
            end
            for i = [2,3,8]
                if isNeedTurn3(swarm(i),blue(2)) ~= 0 % 若根据策略需要转弯掉头，则执行左转 180 度
                    [swarm(i).isTurn , swarm(i).turnTime, swarm(i).maxTurnTime] = swarm(i).SwarmTurn(isNeedTurn3(swarm(i),blue(2)),oneangle);
                end
                [swarm(i).agent , swarm(i).isTurn , swarm(i).turnTime] = swarm(i).UpdateState; % 状态更新
            end
            for i = 4:7
                if isNeedTurn3(swarm(i),blue(1)) ~= 0 % 若根据策略需要转弯掉头，则执行左转 180 度
                    [swarm(i).isTurn , swarm(i).turnTime, swarm(i).maxTurnTime] = swarm(i).SwarmTurn(isNeedTurn3(swarm(i),blue(1)),oneangle);
                end
                [swarm(i).agent , swarm(i).isTurn , swarm(i).turnTime] = swarm(i).UpdateState; % 状态更新
            end
        end
        %%%% 红方 10 个无人机集群的转弯和状态更新处理 %%%%

        %%%%%%% 绘制实时轨迹图 %%%%%%%
        figure(1)
        for i = 1:n0
            blue(i).scatterPoint(2)
        end
        for i = 1:n1
            swarm(i).scatterPoint(i+2)
        end
        drawnow
        %%%%%%% 绘制实时轨迹图 %%%%%%%

        %%%%%%% 给定终止条件 %%%%%%%
%           if index*dT > 360 || isAttack(blue,swarm1) || isAttack(blue,swarm2) || isAttack(blue,swarm3) || isAttack(blue,swarm4) || (blue.agent.pos(2,end)>M/2 || blue.agent.pos(2,end)<-M/2) || (blue.agent.pos(1,end)>L/2 || blue.agent.pos(1,end)<-L/2)% 终止条件

%               break
%           end
        %%%%%%% 给定终止条件 %%%%%%%

    end
end
```