

# 基于改进模拟退火的无人机自适应编队控制模型

## 摘 要

在无人机编队飞行时，需要采用纯方位无源定位的方法调整无人机位置，以尽可能减少向外辐射的电磁波信号，保持电磁静默。基于上述背景，本文利用小量近似、蒙特卡拉模拟的思想建立并验证无人机的定位模型，运用改进的模拟退火算法实现圆形编队无人机的自适应调整，并将编队调整策略推广至所有编队的情形，为无人机的编队控制提供科学的决策支持。

针对问题一 (1)，为了实现对无人机的定位，首先建立直角坐标系，运用向量夹角公式对接收到的方位信息进行表示，得到一个求解较为复杂的方程组。考虑到无人机的位置仅略有偏差，采用小量近似的方法对方程组进行进一步简化。最终，仅需对一个方程进行求解，即可确定无人机的位置。

针对问题一 (2)，在上一问的基础上，本文实现了在发射信号部分编号未知情况下的无人机定位。利用代数几何进行求解时，增加一个独立方程的同时也多引入了一个未知数，无法确定无人机的架数，因此本文从离散的角度进行分析。考虑到只增加一架无人机时，由于发射信号无人机的位置有多种可能，待测无人机的位置也相应地存在有限个多种可能。但在引入两架编号未知的无人机情况下，对二者的解集取交集，即可确定待测无人机的位置。本文利用 Matlab 软件进行蒙特卡洛模拟验证，最终得出只需两架编号未知的无人机即可进行精确定位。

针对问题一 (3)，无人机编队调整的问题实质上是一个最优化问题，优化目标为实际坐标尽可能接近理想坐标。首先运用蒙特卡洛的思想，得出定位无人机时应选择距离最近的两个无人机发射信号。由于无人机无法确定自身的精确坐标，假设发射信号的无人机位置无偏差，利用前文的定位模型反推得到发射信号无人机的伪坐标，优化目标即转化为伪坐标尽可能接近实际坐标。本文采用模拟退火算法对该模型进行优化，同时引入极径的自适应调整，使得编队最终调整到至指定半径的圆上。将模型代入实际数据，通过 Matlab 软件，成功实现了均匀分布的圆形编队调整，且收敛速度较快。

针对问题二，题目要求将调整策略推广至任意队形，本文的策略是将任意编队视为若干个圆形编队的组合。首先，在编队中选择符合问题一的圆形编队，利用上一问的最优化模型进行位置调整。接着，通过对已有的无人机定位定理进行推广，利用位置已经精确的圆形编队，调整任意一个不在圆上的无人机位置。该策略对于不规则的复杂几何编队仍适用，普适性较强。

**关键词：**模拟退火   蒙特卡洛模拟   自适应   编队控制

## 一、问题重述

### 1.1 问题背景

在无人机编队飞行时，为了尽可能减少向外辐射的电磁波信号，保持电磁静默，调整无人机位置时采用纯方位无源定位的方法。无源定位，即编队中某几架无人机发射电磁信号，其余被定位的无人机接收信号，根据接收到的方向信息进行定位，来实现无人机位置的调整的方法。相较于有源定位，该种定位方式下的无人机受环境影响小、隐蔽性能好、生存能力强<sup>[1]</sup>，是目标定位手段发展的新方向，在现代无人机编队飞行的应用中发挥着越来越重要的作用。

基于上述背景，本文通过建立数学模型解决以下问题：

**问题一** 考虑飞行于同一高度上的 10 架无人机组成了圆形编队，1 架无人机均匀分布在圆心，其余无人机均匀分布在某一圆周上的情形。

- (1) 编队中位于圆心的无人机和另外 2 架无人机发射信号，其余无人机被动地接收信号，位置略有偏差。在满足发射信号的无人机位置无偏差并且编号已知的条件下，建立定位被动接收的无人机的模型。
- (2) 编队中位于圆心的无人机和编号为 FY01 的无人机发射信号，发射信号的无人机位置均无偏差，求除了上述无人机外还需要多少架编号未知的无人机，才能实现位置略有偏差的无人机的有效定位。
- (3) 已知编队圆周半径为 100m，无人机的初始位置已知，位置略有偏差。需要给出合适的无人机位置调整方案，每次选择圆心的无人机和圆周上最多 3 架无人机发射信号，调整其余无人机的位置，重复多次直至调整至理想位置。最后，将该方案带入题目所给的具体数据。

**问题二** 考虑实际飞行中无人机的多种编队方式，给出纯方位无源定位的无人机位置调整方案。

## 二、问题分析

本文要解决的是无人机在编队飞行时采用纯方位无源定位的方式实现定位和调整的问题。问题一针对保持在同一高度上的 10 架飞机的圆形编队队形，首先在发射信号的无人机编号已知时实现对接受信号的无人机的定位，接着探讨在部分发射信号的无人机编号未知的情况下无人机的定位问题。进一步地，考虑无人机在飞行过程中的位置调整问题，并在问题二中将调整方案推广至其他编队队形。

2.1 问题一的分析

第一小问要求我们根据 3 架位置已知的无人机方位信息，定位被动接受信号的无人机。由于发射信号的无人机无偏差且编号已知，根据已知信息，可以建立极坐标系对所有无人机的位置进行描述。为了方便计算，可将极坐标系转为直角坐标系，并采用向量夹角公式对无人机接收到的方位信息进行描述。为了进一步简化方程，观察到接受信号的无人机位置仅略有偏差，考虑采用小量近似的方法对方程进一步简化。然而，由于非线性方程难以求得解析解，实际计算中需要采用数值分析方法求得数值解。

第二小问是在第一小问的基础上，探究在两架无人机编号已知的情况下需要增加的编号未知的无人机的数量，实现对接收信号的无人机的成功定位。由于增加的无人机编号未知，结合第一小问的分析，考虑到多一个飞机在多一个方程的同时也多引入了一个未知数，因此无法采用几何关系推导的方式确定无人机的架数。考虑到这种几何的方式是建立在角度连续基础之上的，可以尝试从离散的角度对该问题进行分析。首先探究增加一架无人机的情况，枚举所有可能的无人机情况，通过第一小问的模型求解，可以发现发射信号的无人机的每个位置都可以对应一个待测无人机位置的预测值，因而其无法确定。为了缩小解的范围，可以引入两架未知标签的无人机，并对二者解集取交集。为了验证该假设的正确性，考虑利用蒙特卡洛模拟的方法，通过均匀分布采样验证。

前两小问为对无人机的定位问题，全过程流程图描述如下：

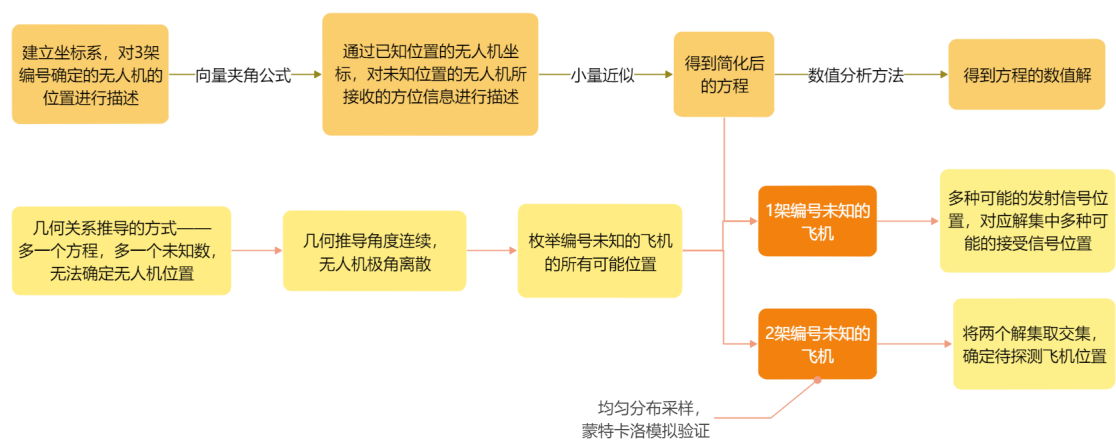


图 2-1 无人机定位的流程图表示

第三小问探究的是无人机位置调整问题，要求使得初始位置略有偏差的无人机最终均匀分布在某个圆周上。该问题可以转化为最优化问题求解，优化参数为无人机的坐标，优化目标为无人机实际坐标尽可能接近理想坐标。由于无人机无法知道当前自身的精确位置，仅可以根据接收到的方向信息，利用第一小问的定位模型反推得到当前的伪坐标，

优化目标因而可以转化为无人机的伪坐标尽可能接近实际坐标，相应地，约束条件即伪坐标的定位方式。

本小问的重点在于如何求解该最优化问题，考虑到目标函数求导、分析过程较为复杂，可以采用智能优化算法进行求解。由于无人机的调整为连续优化问题，考虑采用模拟退火算法。其中，发射信号无人机的选取是关键之一，关系到新解的生成。考虑制定相应决策方式，选取合适的无人机，使得待探测无人机通过信号的方向信息得到的伪坐标与实际偏差较小。根据相应的方向信息，对无人机的伪坐标、实际坐标进行相应更新。是否接受该更新的标准采用算法对应的 Metropolis 准则，倘若更新后的目标函数较更新前的函数更优，则接受，否则以一定概率接受。同时，引入极径的自适应调整，使得编队最终调整至指定半径的圆上。当温度降到温度下限，则停止算法，得到最终的调整结果。

## 2.2 问题二的分析

问题二本质上是对问题一第三小问的推广，将圆形编队的位置调整推广至任意编队情况的调整。考虑到如果对于每一个不同几何形状的编队进行新的建模和分析，繁琐且不具有普适性。因此，可以将任意编队拆分成若干个圆编队的组合，利用前文建立的圆形编队模型，在编队中找到合适的圆以精确调整该圆上的无人机。接着，考虑将问题一第一小问的模型进行拓展，通过位置调整完毕的圆形编队上的无人机，精确调整任意一个不在圆上的无人机位置，至此，编队中的所有无人机位置都已精确调整。利用上述策略，不仅可以解决问题中所列举的具有规则且有规律的简单几何编队——锥形编队情况，也可以解决不规则的复杂几何编队情况。无人机编队的调整全过程，可以用如图2-2所示的流程图表示。

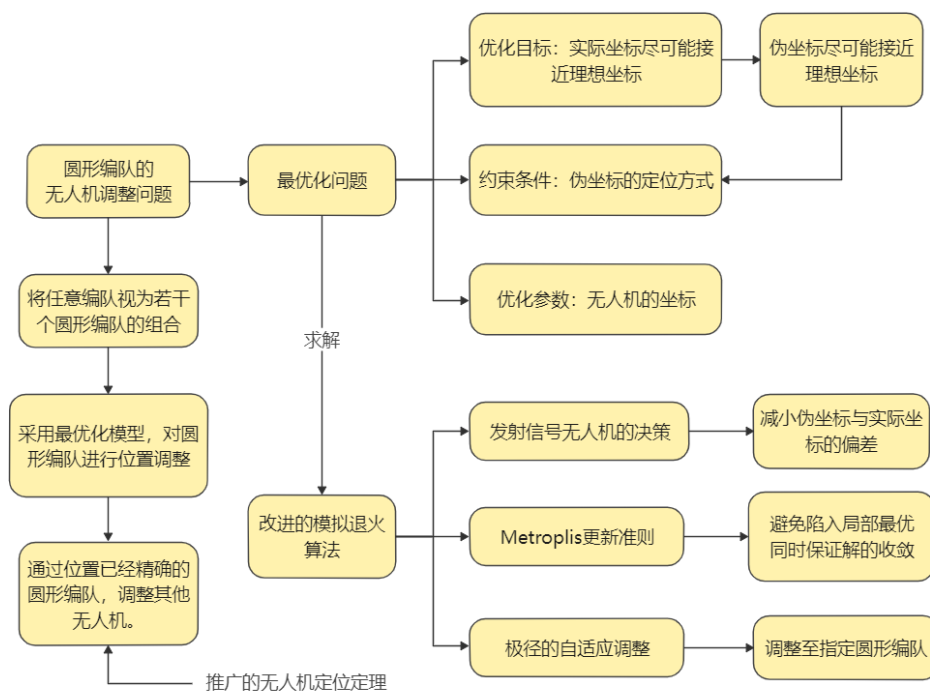


图 2-2 无人机调整的流程图表示

### 三、模型假设

- (1) 假设接收信号的无人机得到的角度数据是非常精确且真实的。
- (2) 假设无人机在得到角度数据时，可以知道该角度所对应的无人机。
- (3) 假设无人机在发射信号时，所发射的信号彼此之间不会产生干涉或其他影响精度的物理效应。
- (4) 假设无人机在编队调整时，不会收到外界因素如障碍物等的干扰。

### 四、符号说明

本文为了便于模型的数学语言表示，定义了如下符号，每个符号在使用时都会相应段落中进行详细说明。

符号	含义
$R$	预定圆周半径 (题目中为 100)
$(r_i, \theta_i)$	编号为 $i$ 的无人机的真实位置
$(r_i^D, \theta_i^D)$	编号为 $i$ 的无人机期望调整到的位置
$(\bar{r}_i, \bar{\theta}_i^D)$	通过期望位置计算的编号为 $i$ 的无人机的伪坐标
$m, i, j$	$m$ 表示接收信号无人机编号, $i, j$ 表示发送信号无人机编号
$(r_m, \theta_m) = H(\alpha_1, \alpha_2, \theta_1, \theta_2)$	$H$ 表示通过 3 架确定位置的无人机求解未知无人机位置函数
$(\alpha_1, \alpha_2) = G(r_1, \theta_1, r_2, \theta_2)$	$G$ 表示通过真实坐标产生的精确纯方向信号函数
$L(r_m, \theta_m, r_m^D, \theta_m^D)$	根据期望位置确定当前位置的的偏差损失
$X$	一组坐标集合 $\{(r_1, \theta_1), \dots, (r_n, \theta_n)\}$

## 五、问题一的建模与求解

### 5.1 无人机定位模型的建立——基于小量近似的思想

为了使模型具有普适性,采用坐标系对无人机的位置进行描述。以位于中心的无人飞机为极点,从极点出发到发射信号的无人机  $A$  的射线为  $x$  轴,建立极坐标系,如图5-1所示。设  $A$  的坐标表示为  $(R, 0)$ , 另一架发射信号的无人机  $B$  的极坐标为  $(R, \theta_1)$ , 接收信号的无人机  $M$  为  $(x, \theta_M)$ 。

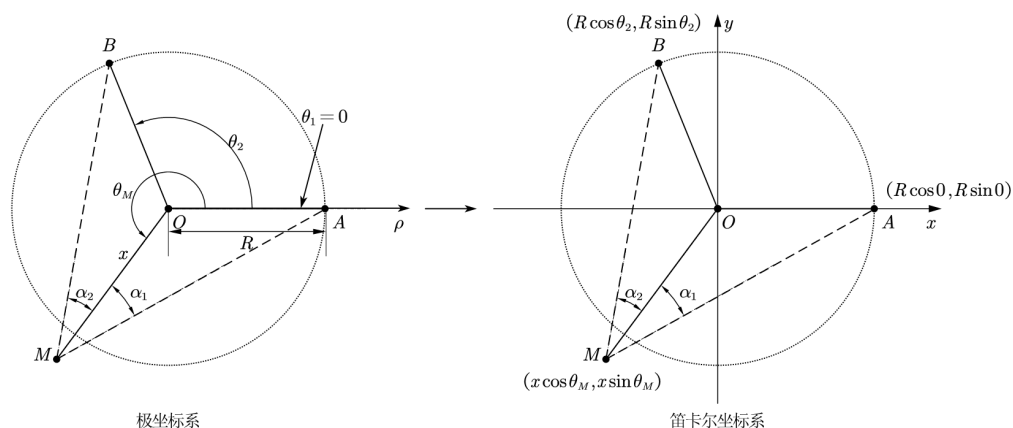


图 5-1 无人机位置表示

为了便于计算,将上述极坐标系在直角坐标系中表示,即  $A(R, 0), B(R \cos \theta_2, R \sin \theta_2)$ ,

$M(x \cos \theta_M, x \sin \theta_M)$ 。因此,  $\overrightarrow{MB}$ 、 $\overrightarrow{MO}$ 、 $\overrightarrow{MA}$  可以采用如下方式表示:

$$\begin{cases} \overrightarrow{MB} = (R \cos \theta_2 - x \cos \theta_M, R \sin \theta_2 - x \sin \theta_M) \\ \overrightarrow{MO} = (-x \cos \theta_M, -x \sin \theta_M) \\ \overrightarrow{MA} = (R - x \cos \theta_M, -x \sin \theta_M) \end{cases}$$

由平面向量夹角公式, 可以用  $\overrightarrow{MB}$  和  $\overrightarrow{MO}$  表示已知的  $\alpha_2$ :

$$\cos \alpha_2 = \cos \langle \overrightarrow{MB}, \overrightarrow{MO} \rangle = \frac{\overrightarrow{MB} \cdot \overrightarrow{MO}}{|\overrightarrow{MB}| |\overrightarrow{MO}|} = \frac{x/R - \cos(\theta_2 - \theta_M)}{\sqrt{1 + (x/R)^2 - 2x \cos(\theta_2 - \theta_M)/R}} \quad (5.1)$$

同理,  $\alpha_1$  也可以用  $\overrightarrow{MA}$  和  $\overrightarrow{MO}$  表示, 即

$$\cos \alpha_1 = \cos \langle \overrightarrow{MA}, \overrightarrow{MO} \rangle = \frac{\overrightarrow{MA} \cdot \overrightarrow{MO}}{|\overrightarrow{MA}| |\overrightarrow{MO}|} = \frac{x/R - \cos(0 - \theta_M)}{\sqrt{1 + (x/R)^2 - 2x \cos(0 - \theta_M)/R}} \quad (5.2)$$

联立式5.1和式5.2, 求解如下方程组即可确定  $\theta_M$ 、 $x$  的值, 得到接收信号的无人机的定位。

$$\begin{cases} \cos \alpha_1 = \frac{x/R - \cos(0 - \theta_M)}{\sqrt{1 + (x/R)^2 - 2x \cos(0 - \theta_M)/R}} \\ \cos \alpha_2 = \frac{x/R - \cos(\theta_2 - \theta_M)}{\sqrt{1 + (x/R)^2 - 2x \cos(\theta_2 - \theta_M)/R}} \end{cases} \quad (5.3)$$

由于该方程无法求出解析解, 在实际计算中, 需要利用已知的  $\theta_1$ 、 $\alpha_1$ 、 $\alpha_2$  已知, 采用双变元牛顿迭代法等数值分析方法求解数值解  $\theta_M$ 、 $x$ , 即可确定待测无人机的位置  $(x, \theta_M)$ 。

为了方便之后使用, 将第二个式子中 0 所代表的第一个方位角替换为  $\theta_1$ , 统一采用函数的方式显示地表达方程5.3的解;

$$\begin{cases} x = H_1(\alpha_1, \alpha_2, \theta_1, \theta_2) \\ \theta_M = H_2(\alpha_1, \alpha_2, \theta_1, \theta_2) \end{cases} \quad (5.4)$$

同时, 为了进一步简化无人机定位模型, 考虑到被定位的无人机位置仅略有偏差, 该微小偏差是小量,  $x/R$  相较于 1 偏差较小, 可以用  $1 + \delta$  表示, 因此, 式5.3可以转换为

$$\begin{cases} \cos \alpha_1 = \frac{1 + \delta - \cos(\theta_1 - \theta_M)}{\sqrt{1 + (1 + \delta)^2 - 2(1 + \delta) \cos(\theta_1 - \theta_M)}} \\ \cos \alpha_2 = \frac{1 + \delta - \cos(0 - \theta_M)}{\sqrt{1 + (1 + \delta)^2 - 2(1 + \delta) \cos(0 - \theta_M)}} \end{cases} \quad (5.5)$$

化简可得,

$$\begin{cases} \cos \alpha_1 = \frac{1}{\sqrt{2(1 - \cos(\theta_1 - \theta_M))}} \cdot \frac{1 + \delta - \cos(\theta_1 - \theta_M)}{\sqrt{1 + \delta}} \\ \cos \alpha_2 = \frac{1}{\sqrt{2(1 - \cos(0 - \theta_M))}} \cdot \frac{1 + \delta - \cos(0 - \theta_M)}{\sqrt{1 + \delta}} \end{cases} \quad (5.6)$$

利用等价无穷小的代换,  $(1 + \delta)^{-\frac{1}{2}} \sim -\frac{1}{2}\delta$ , 式5.6可以转换为

$$\begin{cases} \cos \alpha_1 = \frac{1}{\sqrt{2(1 - \cos(\theta_1 - \theta_M))}} \cdot [\frac{1}{2}\delta(1 + \cos(\theta_1 - \theta_M)) + (1 - \cos(\theta_1 - \theta_M))] \\ \cos \alpha_2 = \frac{1}{\sqrt{2(1 - \cos(0 - \theta_M))}} \cdot [\frac{1}{2}\delta(1 + \cos(0 - \theta_M)) + (1 - \cos(0 - \theta_M))] \end{cases} \quad (5.7)$$

联立式5.7, 可以得到

$$\begin{aligned} \delta &= \frac{2[\sqrt{2(1 - \cos(\theta_1 - \theta_M))} \cos \alpha_1 - (1 - \cos(\theta_1 - \theta_M))]}{1 + \cos(\theta_1 - \theta_M)} \\ &= \frac{2[\sqrt{2(1 - \cos(0 - \theta_M))} \cos \alpha_2 - (1 - \cos(0 - \theta_M))]}{1 + \cos(0 - \theta_M)} \end{aligned} \quad (5.8)$$

尽管小量近似后的方程依然不能接触精确解析解, 但是可以给出仅含有  $\theta_M$  的方程, 采用更为简单的牛顿迭代法就可以快速求解数值解  $\theta_M$ 、 $\delta$ , 即可确定待测无人机的位置  $(1 + \delta, \theta_M)$ 。

## 5.2 无人机有效定位的实现——基于蒙特卡洛的思想

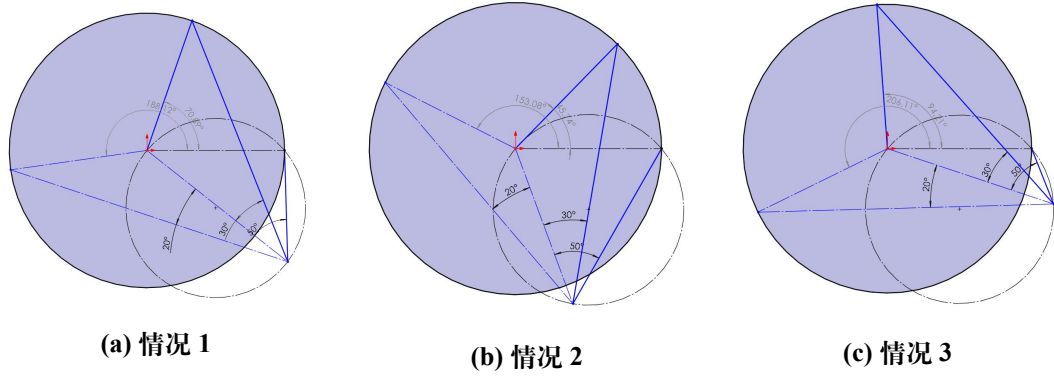
首先考虑采用类似第一小问的解决方案, 即完全通过严格的几何约束建立坐标方程求解参数。然而, 由于本问中接收信号的无人机无法知道发送信号无人机的编号, 即无法确定方程5.3中  $\theta_1$  的具体大小, 因而 2 个独立方程中有 3 个未知数, 无法对无人机进行定位。

当我们把这种情形推广到更多的未知编号的无人机时, 假设除了编号为 0 和 1 飞机外, 选择了  $n$  架编号未知的无人机发送信号, 可以得到形式对称的一组方程。记编号为  $1, \dots, n$  的位置  $\theta_1 = 0, \theta_2, \dots, \theta_n$ , 发送的信号与 0 发送信号的夹角分别为  $\alpha_1, \alpha_2, \dots, \alpha_n$ 。

$$\begin{cases} \cos \alpha_1 = \frac{x/R - \cos(\theta_1 - \theta_M)}{\sqrt{1 + (x/R)^2 - 2x \cos(\theta_1 - \theta_M)/R}} \\ \cos \alpha_2 = \frac{x/R - \cos(\theta_2 - \theta_M)}{\sqrt{1 + (x/R)^2 - 2x \cos(\theta_2 - \theta_M)/R}} \\ \vdots \\ \cos \alpha_n = \frac{x/R - \cos(\theta_n - \theta_M)}{\sqrt{1 + (x/R)^2 - 2x \cos(\theta_n - \theta_M)/R}} \end{cases} \quad (5.9)$$



类似地，该方程组具有  $n$  个独立的方程和  $n + 1$  个未知变量，不论  $n$  取为多少，仅仅通过严格几何关系建立的方程组无法得到无人机位置  $(x, \theta_m)$ 。



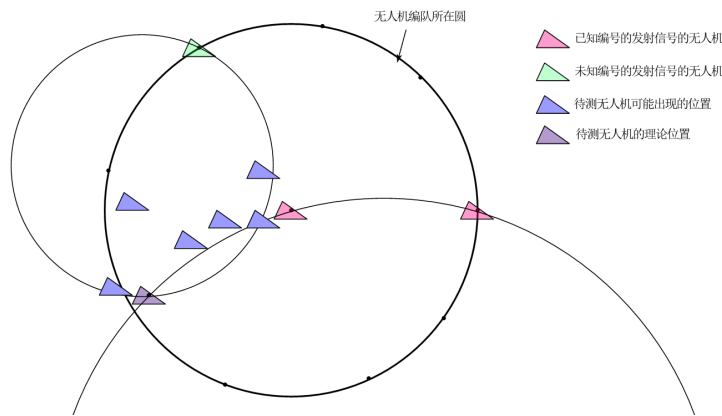
**图 5-2 几何约束不完全示意图：**增加两个编号未知的无人机时，固定  $\alpha_1, \alpha_2, \alpha_3$ ，可以发现，在待探测无人机角度连续改变的过程中，发射信号无人机的位置也可以连续改变，角度约束仍能满足，因此仅通过几何约束无法求解本问题。

考虑到发送信号的无人机位置虽然无法精确确定，但是其位置是没有偏差的，即  $\theta_1, \theta_2, \dots, \theta_n$  一定落在一组离散的坐标点上：

$$\theta_1, \theta_2, \dots, \theta_n \in \{0^\circ, 40^\circ, \dots, 320^\circ\} \quad (5.10)$$

### 5.2.1 情况 1：一架编号未知的无人机

由于该无人机编号未知，发射信号无人机可以在除取 FY01 和待测位置无人机外的任意等分点，因此可行的位置共有  $9-2=7$  种情况，，如图5-3所示。



**图 5-3 情况 1 的无人机位置表示**

根据式5.8，对于 7 种可能的  $\theta_1$ ，已知  $R, \alpha_1, \alpha_2$ ，所对应的无人机位置的解  $\theta_M, x$  有 7 个。然而这 7 中可能的解都是完全符合纯方向信息所提供的几何约束，因此无法确定该无人机的具体位置。

### 5.2.2 情况 2：两架编号未知的无人机

对于每一架无人机，根据情况 1 的讨论，其解空间中各包含 7 个待探测无人机的可能位置，则该无人机的位置一定在两个解空间的交集中。该过程如图5-4。

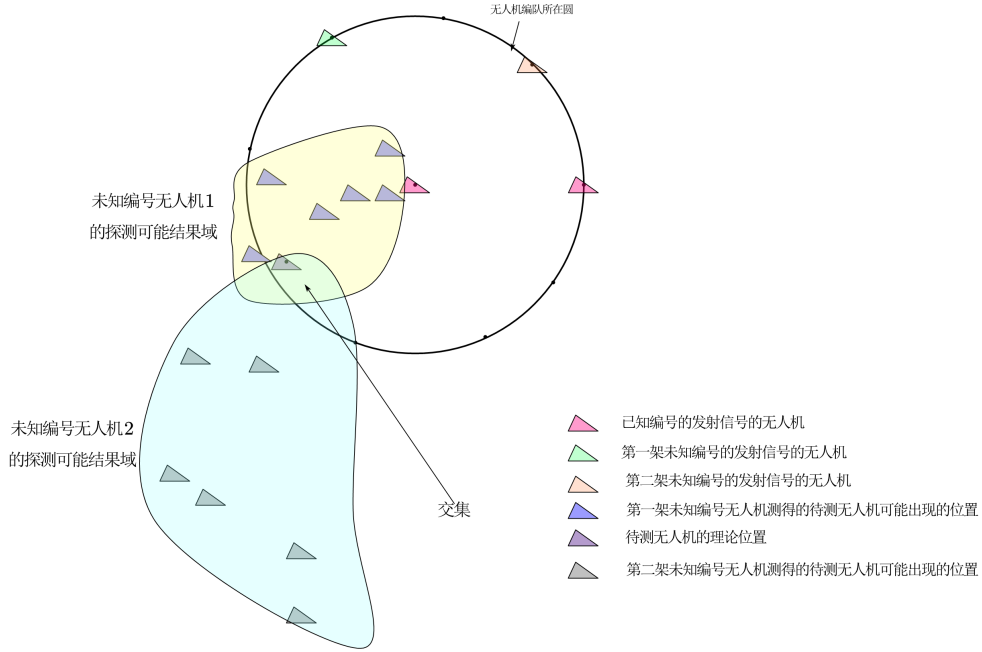


图 5-4 情况 2 的无人机位置表示

令待测位置无人机编号为  $FY0m$ , 发信号无人机编号分别为  $FY00, FY01, FY0i, FY0j$ , 测得的与  $FY00$  相邻的三个独立方位角为  $\alpha_1, \alpha_i, \alpha_j$ , 遍历两架未知编号无人机的全部 7 个可能位置:

$$\theta_i, \theta_j \in \Theta_7 = \{1 \cdot 40^\circ, \dots, (k-1) \cdot 40^\circ, (k+1) \cdot 40^\circ, \dots, 8 \cdot 40^\circ\}$$

对于每个位置，使用相同的对应的纯方向信息组  $(\alpha_1, \alpha_i, 0, \theta_i)$  和  $(\alpha_1, \alpha_j, 0, \theta_j)$ , 利用方程5.4可以得到两组可能的位置解:

$$\begin{cases} X_i = \{(r, \theta) | r = H_1(\alpha_1, \alpha_i, 0, \theta_i), \theta = H_2(\alpha_1, \alpha_i, 0, \theta_i), \theta_i \in \Theta_7\} \\ X_j = \{(r, \theta) | r = H_1(\alpha_1, \alpha_j, 0, \theta_j), \theta = H_2(\alpha_1, \alpha_j, 0, \theta_j), \theta_j \in \Theta_7\} \end{cases} \quad (5.11)$$

则待测无人机  $FY0k$  的位置  $(x, \theta_m) \in X_i \cap X_j$ 。该算法的伪代码如下:

---

**Algorithm 1** Location algorithm for 2 unknown number UAV signal

---

**Input:** 待测无人机编号  $fy_m$ , FY01 及 2 架未知编号飞机与 FY00 产生的 3 个纯向量信息  $\alpha_0, \alpha_1, \alpha_2$ , 编队圆周半径  $R$

**Output:** 待测飞机坐标  $(x_m, \theta_m)$ , 未知编号飞机坐标及编号  $fy1 : (R, \theta_1), fy2 : (R, \theta_2)$

**Variables:** 未知编号飞机各自导出的 7 个可能位置  $(X_1)_{7 \times 1}, (X_2)_{7 \times 1}$

```
1: for  $fy = 1$  to 9 do
2:   // 对未知编号飞机的每一个可能位置进行估计
3:   if 该位置与 FY00 或  $fy_m$  编号期望位置重合 then
4:     continue
5:   end if
6:   计算未知编号在  $fy$  处时对应的可能  $fy_m$  坐标并计入  $X_1, X_2$ :
7:      $X_1 \leftarrow X_1 \cup \{(H_1(\alpha_0, \alpha_1, R, 0, fy \cdot \frac{40\pi}{180}), H_2(\alpha_0, \alpha_1, R, 0, fy \cdot \frac{40\pi}{180}))\}$ 
8:      $X_2 \leftarrow X_2 \cup \{(H_1(\alpha_0, \alpha_2, R, 0, fy \cdot \frac{40\pi}{180}), H_2(\alpha_0, \alpha_2, R, 0, fy \cdot \frac{40\pi}{180}))\}$ 
9: end for
10: 取两个集合的交集即为待求坐标  $(x_m, \theta_m) \leftarrow A = X_1 \cap X_2$ , A 的元素在  $X_1$  的索引为  $fy1$ , A 的元素在  $X_2$  的索引为  $fy2$ 
11: return  $(x_m, \theta_m), fy1 : (R, fy1 \cdot \frac{40\pi}{180}), fy2 : (R, fy2 \cdot \frac{40\pi}{180})$ 
```

---

然而，算法并不能保证两组可行解的交集只有一个元素，同时由于计算机求解精度以及方位角的特殊性，交集的解可能并不是待测无人机位置的正确解。为了验证上述模型和算法的有效性，使用蒙特卡洛模拟，模拟流程图如图5-5所示。

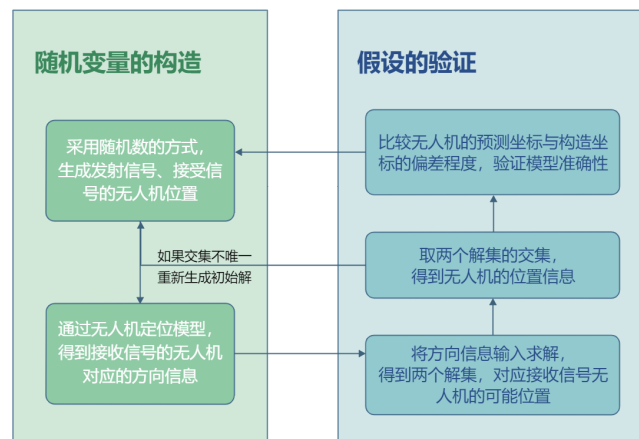


图 5-5 蒙特卡洛模拟流程图

该算法分为两个步骤，随机变量的构造与假设的验证。

### (1) 步骤一：随机变量的构造

- 首先，采用均匀分布随机生成互不相同且均为  $40^\circ$  倍数的  $\theta_1$ 、 $\theta_2$ ，对应两架未知位置的无人机所在位置的极角；随机生成位置与上述 2 架无人机不同且较理想位置略有偏差的  $\theta_M$ 、 $x$ ，对应待定位无人机的极坐标。
- 由式5.3，得到待测无人机  $(x, \theta_m)$  所接收到的方向信息： $\alpha_1$ （FY00 与无人机 1 夹角）、 $\alpha_2$ （FY00 与无人机 2 夹角）以及固定的  $\alpha_0$ 。

$$\begin{cases} \alpha_1 = G_1(r_1, \theta_1, r_2, \theta_2) = \arccos\left(\frac{x/r_1 - \cos(\theta_1 - \theta_m)}{\sqrt{1 + (x/r_1)^2 - 2x \cos(\theta_1 - \theta_m)/r_1}}\right) \\ \alpha_2 = G_2(r_1, \theta_1, r_2, \theta_2) = \arccos\left(\frac{x/r_2 - \cos(\theta_2 - \theta_m)}{\sqrt{1 + (x/r_2)^2 - 2x \cos(\theta_2 - \theta_m)/r_2}}\right) \end{cases} \quad (5.12)$$

### (2) 步骤二：假设的验证

- 将该方向信息输入求解，根据情况 1 的分析， $\alpha_1$ 、 $\alpha_2$  和  $\alpha'_1$ 、 $\alpha'_2$  各对应 7 种可能的待探测无人机的解。
- 取两个解集的交集，得到带探测无人机的极坐标信息。如果交集为不唯一或为空，由于绝大部分情况下交集仅有一个，故重新生成初始解。
- 验证无人机的预测坐标与构造坐标的偏差程度，验证模型的准确性。如果  $\theta_M$  的预测值较实际差距小于  $1 \times 10^{-5}$  且  $x$  的预测值偏差小于  $1 \times 10^{-10}$ ，则可视作预测结果正确。

为了评估结果的准确程度，引入三个指标——成功搜索概率、总搜索正确率、预测准确率，其定义如下：

- 模拟成功概率 = 交集唯一的次数/总模拟次数
- 模拟总正确率 = 预测结果正确的次数/总模拟次数
- 相对预测准确率 = 预测结果正确的次数/交集唯一的次数

通过 matlab 程序模拟多次取平均值，得到各指标见表5-1。

表 5-1 两架无人机的蒙特卡洛模拟结果

次数	100	1000	10000	100000
模拟成功概率	100%	99.80%	99.68%	99.68%
模拟总正确率	100%	99.80%	99.67%	99.66%
相对预测准确率	100%	100%	99.99%	99.98%

分析表5-1可以发现，各项指标准确率均较高。首先，模拟总正确率较高，验证了模型整体的准确性。同时，模拟成功概率较高，说明取交集的操作较为合理；相对预测准

确率极高，验证了在交集唯一的情况下预测结果的正确性。综合上述分析，可以验证假设的正确性，即只需要 2 架编号未知的无人机即可对接收信号的无人机进行定位。

### 5.3 无人机位置的调整方案——基于改进的模拟退火算法

无人机的调整问题，本质上为无人机坐标的最优化问题。设发射信号的无人机的编号为  $i, j$ ,  $i \neq j$  且  $i, j \in \{1, 2, \dots, 9\}$ 。设第  $t$  次调整时编号为  $m$  的无人机的极坐标为  $(r_m^t, \theta_m^t)$ ,  $m = 1, 2, \dots, 9$  且  $m \neq i, j$ 。优化目标为无人机的坐标尽可能接近理想坐标，对应均匀分布的圆形编队，即

$$\min \sum_{m=1}^9 \sqrt{\left| \frac{r_m^t - r_m^D}{r_m^t} \right|^2 + \left| \frac{\theta_m^t - \theta_s^t}{\theta_s^t} \right|^2} \quad (5.13)$$

其中,  $s = (m + 1) \bmod 9$ ,  $r_m^D$  为编号为  $m$  的无人机的理想极径。式子中的值越小，无人机的当前位置  $r_m^t$  越接近于目标位置  $r_m^D$ ，相邻无人机的极角之差越来越接近，无人机越来越接近于均匀圆形编队。

由于编队中的无人机无法知道自身的实际坐标  $(r_m^t, \theta_m^t)$ ，仅能通过第一小问中的模型反推得到当前的伪坐标，也就是假设发射信号的无人机位置无偏差，按照发射信号的无人机  $i$  坐标为  $(R, \theta_i^D)$ 、无人机  $j(R, \theta_j^D)$  推测，其中  $\theta_i^D = 40(i - 1)^\circ$ ,  $\theta_j^D = 40(j - 1)^\circ$ 。将第一小问中无人机 1 固定发射信号的定位模型推广至除圆心外任意 2 架无人机发射信号的情况，建立无人机定位模型

$$\begin{cases} \cos \alpha_{mi}^t = \frac{\overline{r_m^t}/r_m^D - \cos(\theta_i^D - \overline{\theta_m^t})}{\sqrt{1 + (\overline{r_m^t}/r_m^D)^2 - 2\overline{r_m^t} \cos(\theta_i^D - \overline{\theta_m^t})/r_m^D}} \\ \cos \alpha_{mj}^t = \frac{\overline{r_m^t}/r_m^D - \cos(\theta_j^D - \overline{\theta_m^t})}{\sqrt{1 + (\overline{r_m^t}/r_m^D)^2 - 2\overline{r_m^t} \cos(\theta_j^D - \overline{\theta_m^t})/r_m^D}} \end{cases} \quad (5.14)$$

其中,  $(\overline{r_m^t}, \overline{\theta_m^t})$  对应  $t$  时刻无人机  $m$  的伪坐标,  $\alpha_{mi}^t$  对应  $t$  时刻无人机  $m$  接收的来自  $i$  无人机的位置信号,  $\alpha_{mj}^t$  同理。

求解方程 5.14，得到  $t$  时刻无人机  $m$  的伪坐标，对应着约束条件

$$\begin{cases} \overline{r_m^t} = H_1(\alpha_{mi}^t, \alpha_{mj}^t, r_i^D, \theta_i^D, r_j^D, \theta_j^D) \\ \overline{\theta_m^t} = H_2(\alpha_{mi}^t, \alpha_{mj}^t, r_i^D, \theta_i^D, r_j^D, \theta_j^D) \end{cases} \quad (5.15)$$

其中,  $H_1, H_2$  为方程 5.15 确定的隐函数。

因此，优化目标可以转化为

$$\min L(\overline{r_m^t}, \overline{\theta_m^t}) = \sum_{m=1}^9 \sqrt{\left| \frac{\overline{r_m^t} - r_m^D}{\overline{r_m^t}} \right|^2 + \left| \frac{\overline{\theta_m^t} - \theta_s^t}{\overline{\theta_m^t}} \right|^2}$$

其中,  $s = (m + 1) \bmod 9$ ,  $L$  为目标函数。

可以得到该问题的形式化描述

$$\begin{cases} \min L(\bar{r}_m^t, \bar{\theta}_m^t) = \sum_{m=1}^9 \sqrt{\left| \frac{\bar{r}_m^t - r_m^D}{r_m^t} \right|^2 + \left| \frac{\bar{\theta}_m^t - \theta_s^t}{\theta_m^t} \right|^2} \\ s.t. \quad \bar{r}_m^t = H_1(\alpha_{mi}^t, \alpha_{mj}^t, r_i^D, \theta_i^D, r_j^D, \theta_j^D) \\ \quad \bar{\theta}_m^t = H_2(\alpha_{mi}^t, \alpha_{mj}^t, r_i^D, \theta_i^D, r_j^D, \theta_j^D) \end{cases} \quad (5.16)$$

为了求解该最优化问题, 考虑到目标函数的求导、分析过程较为复杂, 采用智能优化算法进行求解。由于无人机的调整为连续优化问题, 可以采用模拟退火算法。模拟退火的思想来源于固体退火原理, 是基于蒙特卡洛迭代求解策略的一种随机寻优算法。它模拟了固体退火原理, 从一个较高的温度出发, 随着温度的下降, 根据概率突跳特性在解空间中搜索目标函数的全局最优解, 使得温度趋于稳定时的解趋近于全局最优。本文对算法稍加改进, 基于其思想建立了自适应调整的无人机编队模型, 算法的核心部分包括三个模块: 发射信号无人机的决策模块、新解的更新模块、极径的自适应调整模块。

### 5.3.1 发射信号无人机的决策模块

在前面的模型中, 发射信号的无人机位置是精确的, 因此计算得到的待测无人机位置也是精确的。但是对于这一问, 发射信号的无人机位置未知, 那么用前文模型计算得到的待测无人机位置与当前精确位置必然会存在偏差。因此, 我们先研究发射信号的无人机的位置偏差  $\Delta\theta_i^-$  对计算得到的待测无人机的位置偏差  $\Delta\theta_m^-$  的影响。 $\Delta\theta_i^-$  与  $\Delta\theta_m^-$  计算公式如下:

$$\begin{aligned} \Delta\theta_i^- &= \theta_i^- - \theta_i \\ \Delta\theta_m^- &= \theta_m^- - \theta_m \end{aligned} \quad (5.17)$$

其中,  $\theta_i^-$  为发射信号的无人机实际位置,  $\theta_i$  为发射信号的无人机的精确位置,  $\theta_m^-$  为上文模型计算得到的待测无人机位置,  $\theta_m$  为待测无人机的当前精确位置 (非期望位置)。注意  $\Delta\theta_m^-$  与前文的  $\Delta\theta_m$  含义不同,  $\Delta\theta_m$  是指通过精确的定位无人机计算得到的待测无人机的与期望位置的偏差。

我们关心的是: 当定位一个无人机时, 应该用哪些无人机发射信号? 由于发射信号的无人机位置本身不精确, 因此不同的发射信号无人机之间存在的显著区别仅为: 发射信号无人机与接收信号无人机的距离不同, 即  $|\theta_i - \theta_m|$  不同。

为了直观描述  $|\theta_i - \theta_m|$  不同时,  $\Delta\theta_i^-$  对  $\Delta\theta_m^-$  的影响的不同, 我们仍利用前文的蒙特卡洛模拟, 得到的结果如下图:

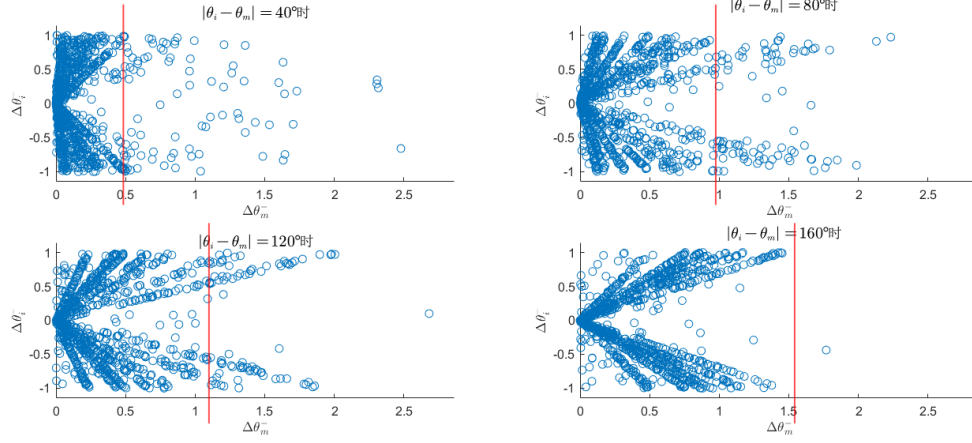


图 5-6 不同  $|\theta_i - \theta_m|$  下  $\Delta\theta_i^-$  对  $\Delta\theta_m^-$  的影响图

从图中可以看到, 对于任意一个  $|\theta_i - \theta_m|$ ,  $\Delta\theta_i^-$  越大,  $\Delta\theta_m^-$  越大, 这验证了发射信号无人机偏差越大, 计算得到的偏差就越大。但是对于不同的  $|\theta_i - \theta_m|$ , 当  $|\theta_i - \theta_m| = 40^\circ$  时,  $\Delta\theta_m^-$  基本都小于 0.5 度, 而随着  $|\theta_i - \theta_m|$  的增大,  $\Delta\theta_m^-$  的边界也增大, 因此采用  $|\theta_i - \theta_m| = 40^\circ$  的无人机进行发射信号定位时, 得到的偏差较小。

据此我们得到结论: 当定位一个无人机时, 应该选择距离它最近的无人机进行定位, 这样可以减小误差, 如图5-7所示。

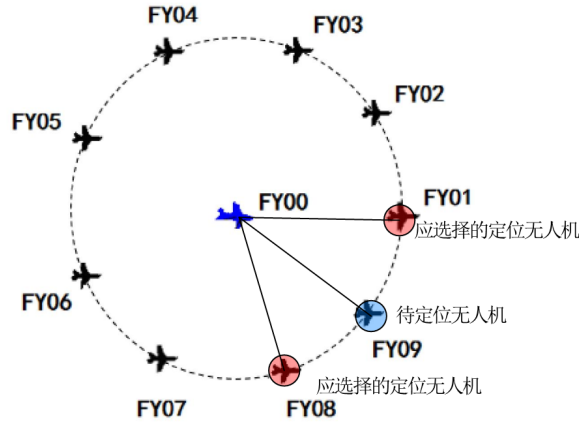


图 5-7 发信号无人机的选择策略示意图

### 5.3.2 新解的更新模块

对于第  $m$  个无人机, 第  $t$  次生成新解时, 规定控制量为

$$\begin{cases} u_r^t = lr \cdot (r_m^D - \overline{r_m^t}) \\ u_\theta^t = lr \cdot (\theta_m^D - \overline{\theta_m^t}) \end{cases} \quad (5.18)$$

更新无人机位置时，计算当前伪坐标与预期坐标的损失值

$$Loss^t = L(\overline{r_m^t}, \overline{\theta_m^t}, r_m^D) \quad (5.19)$$

依据 Metropolis 准则对解进行更新，更新解的概率表示为

$$p = \begin{cases} 1 & , Loss^t \leq Loss^{t-1} \\ \exp(-\frac{Loss^t - Loss^{t-1}}{T}) & , Loss^t > Loss^{t-1} \end{cases} \quad (5.20)$$

其中， $T$  表示为模拟退火算法中的当前温度。当损失值较上一时刻有所下降，则更新解；反之，以一定概率更新解，且温度越高，上升的损失值越小，概率越大。模拟退火允许当前解在一定概率下向更劣值更新，且随着时间的推移，更新的概率逐渐变小，既避免了陷入局部最优，同时保证了解的收敛。

如果接受新解，则对无人机的坐标进行更新

$$\begin{cases} r_m^t \leftarrow r_m^t + u_r^t \\ \theta_i^t \leftarrow r_i^t + u_\theta^t \end{cases} \quad (5.21)$$

其中， $T$  表示为模拟退火算法中的当前温度。当损失值较上一时刻有所下降，则更新解；反之，以一定概率更新解，且温度越高，上升的损失值越小，概率越大。模拟退火允许当前解在一定概率下向更劣值更新，且随着时间的推移，更新的概率逐渐变小，既避免了陷入局部最优，同时保证了解的收敛。

### 5.3.3 极径的自适应调整模块

仅采用上述两个调整更新模块，迭代后会发现所有无人机都会逼近于初始位置的平均半径所在的圆，而非算法需要的固定半径  $R$  的圆。从迭代过程的角度考虑半径的每一次更新：

$$r_m^{t+1} = r_m^t + R - H(\alpha_1, \alpha_2)$$

其中  $H(\alpha_1, \alpha_2)$  函数是使用5.12中  $G$  函数生成信号  $\alpha_1, \alpha_2$  并用5.4中的  $H$  函数反解伪坐标的综合函数。将相应涉及的自变量展开后为如下形式：

$$\begin{aligned} r_m^{t+1} &= r_m^t + R - H(G_1(r_m^t, \theta_m^t, r_i^t, \theta_i^t, r_j^t, \theta_j^t), G_2(r_m^t, \theta_m^t, r_i^t, \theta_i^t, r_j^t, \theta_j^t)) \\ &= r_m^t + R - H(r_m^t, \theta_m^t, r_i^t, \theta_i^t, r_j^t, \theta_j^t) \end{aligned}$$

迭代达到稳定时，上式左右两边对  $t$  取极限，有：

$$R = \lim_{t \rightarrow \infty} H(r_m^t, \theta_m^t, r_i^t, \theta_i^t, r_j^t, \theta_j^t)$$



理想情形下有  $H(R, \theta_m^t, R, \theta_i^t, R, \theta_j^t) = R$ ，这种情况对应着5节中问题一求解一架圆周上飞机位置的情形，因此该等式此时一定成立。然而实际情形中迭代的结果会趋向于稳定在另一个满足该狮子的坐标，即：

$$H(R^*, \theta_m^t, R^*, \theta_i^t, R^*, \theta_j^t) = R$$

$$R^* = \bar{r} = \frac{1}{9} \sum_{i=1}^9 r_i$$

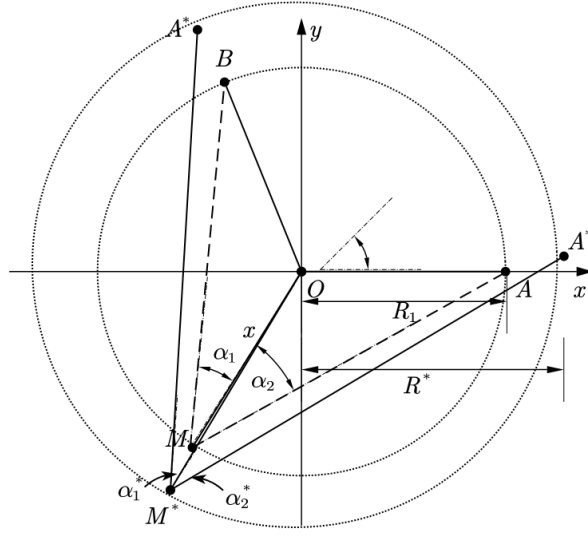


图 5-8 稳定点示意图

为了解决稳定解不在  $R$  的问题，在每次更新迭代过程中增加多次调整坐标极径均值的更新，由于迭代中无法知晓无人机的真实坐标，因此使用可计算的无人机伪坐标进行均值调整，即：

$$\bar{avg}^t = \sum_{i=1}^9 \bar{r}_i^t$$

$$r_i^{t+1} = r_i^t + R - \bar{avg}^t$$

更新后新的无人机坐标均值会更接近于  $R$ ，5.3.5中的结果证明这样的自适应调整方法能够十分有效地使结果收敛在指定的半径  $R$  的圆上。

### 5.3.4 模拟算法整体流程

综合上述所有模块，给出整体的基于模拟退火的自适应编队控制算法如下。

---

**Algorithm 2** Adaptive formation control algorithm based on SA

---

**Input:** UAV 初始位置坐标集合  $r_i, \theta_i$ , 编队预期坐标  $r_i^D, \theta_i^D, i = 1, 2, \dots, 9$

**Hyper Parameters:** 步长  $lr$ , SA 初始温度  $T$ , SA 衰减系数  $T_k$ , 迭代终止损失  $minLoss$ , 半径修正周期  $rec$ , 迭代周期上限  $iter_{max}$

**Output:** 每次位置调整控制量集合  $U_i$ , 每次调整后 UAV 坐标集合  $X_i, i = 1, \dots, 9$

**Variables:** 当前次数  $t$ , 损失值  $Loss^t$ , 伪坐标  $\overline{r}_i^t, \overline{\theta}_i^t$ , 半径均值  $avg$

```
1: for  $t = 1$  to  $iter_{max}$  do
2:   for  $m = 1$  to 9 do
3:     // 每次按顺序更新一架飞机 m
4:     选择 k 左右两侧的飞机作为圆周发射源:  $i \leftarrow m - 1, j \leftarrow m + 1$ 
5:     发送纯方向信号  $\alpha_1, \alpha_2$ :  $\alpha_k \leftarrow G_k(r_i^t, \theta_i, r_j^t, \theta_j)$ 
6:     根据预期位置计算伪坐标  $\overline{r}_m^t, \overline{\theta}_m^t$ :
7:        $\overline{r}_m^t \leftarrow H_1(\alpha_1, \alpha_2, r_i^D, \theta_i^D, r_j^D, \theta_j^D)$ 
8:        $\overline{\theta}_m^t \leftarrow H_2(\alpha_1, \alpha_2, r_i^D, \theta_i^D, r_j^D, \theta_j^D)$ 
9:     计算伪坐标与预期坐标的损失值:  $Loss^t \leftarrow L(\overline{r}_m^t, \overline{\theta}_m^t, r_m^D, \theta_m^D)$ 
10:    计算控制量:  $u^t = (u_r^t, u_\theta^t)$ :
11:       $u_r^t \leftarrow lr \cdot (r_m^D - \overline{r}_m^t)$ 
12:       $u_\theta^t \leftarrow lr \cdot (\theta_m^D - \overline{\theta}_m^t)$ 
13:    //按 Metropolis 法则接收新解
14:    if  $Loss^t - Loss^{t-1} \leq 0$  或  $rand() \leq \exp(\frac{-(Loss^t - Loss^{t-1})}{T})$  then
15:      UAV 按  $u_t$  移动, 更新真实坐标:  $(r_i^t, \theta_i^t) \leftarrow (r_i^t, \theta_i^t) + u^t$ 
16:    end if
17:    记录本次更新过程:  $U_m \leftarrow U_m \cup \{(u_r^t, u_\theta^t)\}; X_i \leftarrow X_i \cup \{(r_i^t, \theta_i^t)\}$ 
18:    更新退火温度:  $T \leftarrow T \cdot T_k$ 
19:  end for
20:  if  $mod(t, rec) == 0$  then
21:    计算极径平均值:  $avg = \frac{1}{9} \sum_{n=1}^9 \overline{r}_n^t$ 
22:    利用  $avg$  进行半径补偿:  $r_j^t \leftarrow r_j^t - avg, j = 1, \dots, 9$ 
23:  end if
24:  if  $Loss_t \leq minLoss$  then
25:    break
26:  end if
27: end for
28: return  $U_i, X_i$ 
```

---

整体的编队控制算法使用模拟退火作为迭代主体，适当改进了算法的生成新界和终止条件使得算法更易于求解圆周编队。同时，将上文中叙述的 3 个优化模块嵌入在循环不同位置。在初始化生成新的信号时，始终选择距离待更新无人机最近的两架作为信号源定位，其余飞机静默，在循环中首先计算待更新无人机的伪坐标，用这个坐标代表他的真是坐标计算距离损失作为更新控制量，同时根据损失的变化判断是否应该进行本次更新，最后无人机根据控制量进行移动，将控制量更新到无人机的实际位置中模拟下一次迭代更新。最终算法能够给出整体编队调整流程，实验结果表明算法能够有效地使无人机逼近到预定位置。

### 5.3.5 题目所给数据的调整方案

将该算法代入题目所给数据，迭代求解。算法各项超参数如下：

表 5-2 超参数表

迭代次数	步长	初始温度	衰减系数	极径自适应周期	收敛极径大小
200	0.005	1.00E+10	0.99	5	100

经过 200 次调整，无人机编队调整至均匀分布的圆形编队。下图中展示了编队迭代调整过程中 Loss 的变化。

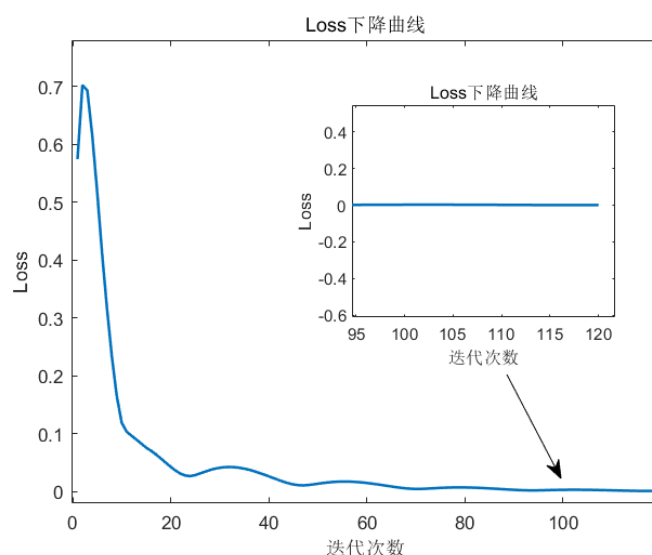


图 5-9 Loss 下降曲线

从 Loss 图中可以看出，在迭代前期，Loss 下降非常快，在迭代次数为 60 次之后，目标函数已经接近与 0，这说明我们的模型效果较好，迭代次数也较小，计算速度较快。调整过程及最终见图5-10。

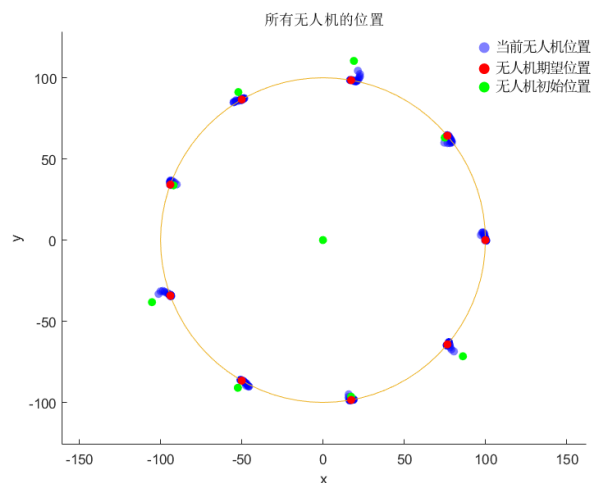


图 5-10 无人机编队调整过程

可以发现，随着无人机编队的不断调整，无人机逐渐由略有偏差的初始位置调整至理想位置，形成半径为 100 米、相邻极角之差为  $40^\circ$  的圆形编队，调整方案较为科学。为了更加清晰地描述收敛过程，将每个无人机的调整细节呈现在图5-11中。

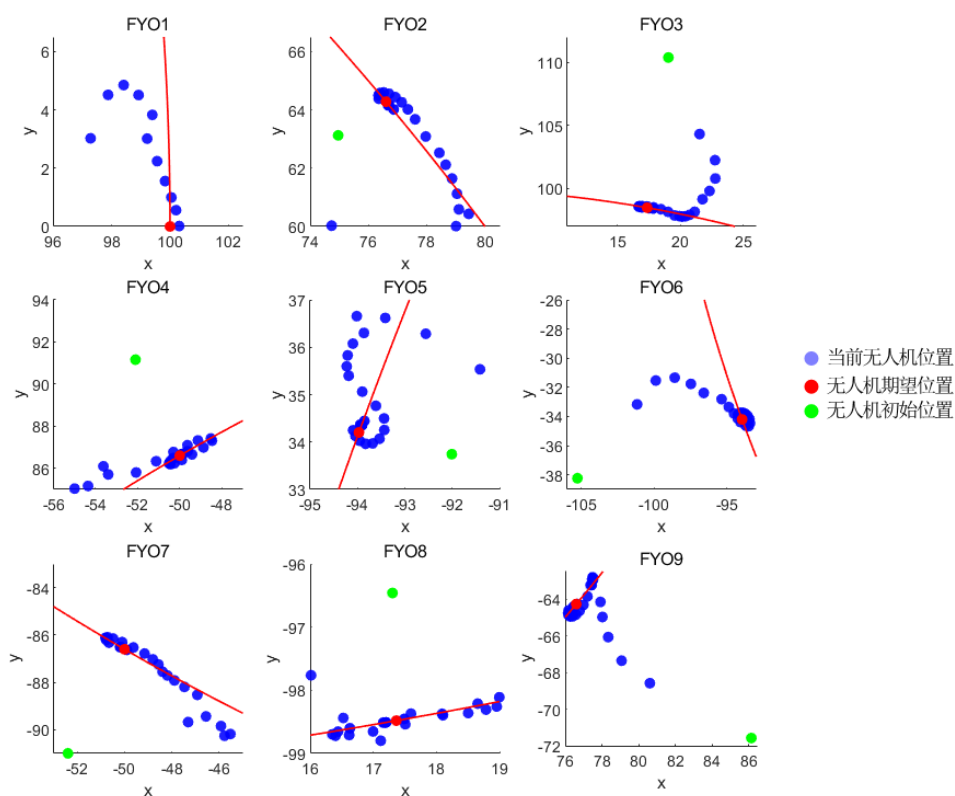


图 5-11 无人机编队调整细节

从图5-11中可以观察到，随着迭代的进行，无人机整体上趋近于期望位置，期间位

置偶有偏移：位置偏移是因为模拟退火算法会按照一定概率向更劣的策略转移，逐渐下降是因为随着温度的下降，向更劣策略转移的概率不断降低。该过程与模拟退火的原理相符合。

## 六、问题二的建模与求解

首先，对问题一第一小问中的模型进行拓展，得到以下定理：

**定理 6.1 (无人机定位)** 给定一个参考无人机以及两个任意位置的可发射信号的无人机，若可发射信号的无人机编号已知且位置精确，则待定位的无人机位置可以精确定位。

**证明** 通过几何法进行证明。若可发射信号的无人机  $A, B$  编号已知且位置精确，待定位无人机  $M$  接收到角度信号为  $\alpha_A, \alpha_B$ ，那么可根据点  $A$ 、点  $O$ 、角  $\alpha_A$  唯一确定一个圆。同理，点  $B$ 、点  $O$ 、角  $\alpha_B$  唯一确定一个圆，这两个圆确定一个交点  $M$ ，该交点即为待测无人机  $M$  的精确位置，如图6-1所示。

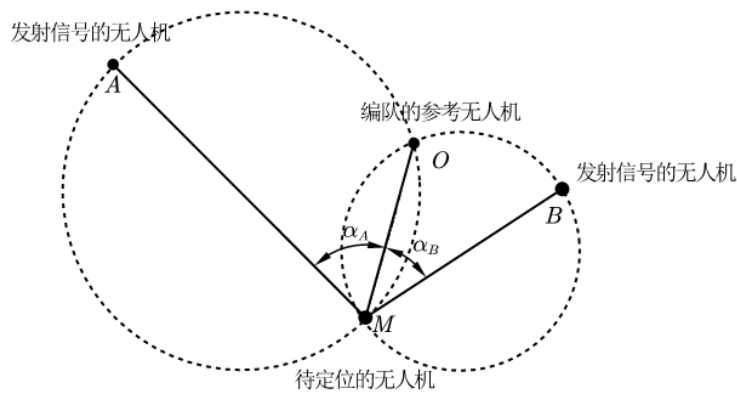


图 6-1 无人机定位示意图

该定理将问题 1 的模型 1 从圆上的无人机拓展到任意位置的无人机。

在实际飞行中，无人机集群也可以是其他编队队形。问题一中的模型都是基于圆形编队，为了解决其他编队的队形保持问题，可以将其他编队在几何上进行再分，将其他形状的编队看成圆形编队的组合或拆分。基于这个思想，可以给出一般形状编队的队形保持策略，即无人机位置调整方案。

给定任意形状的编队队形，其无人机总数为  $N$ ，其调整步骤如下：

- (1) 选取一个参考无人机，选取的方案为：除参考无人机，其他无人机尽可能多的在以参考无人机为圆心的某几个圆周上。
- (2) 假设以参考无人机为圆心的某几个圆周上的无人机数目为  $M (M > 2)$ ，那么根据问题 1 的第 3 小问，我们可以一共调整  $M + 1$  个无人机的位置（包括参考无人机），

此后  $M + 1$  个无人机已经处于理想位置。

- (3) 对于剩下的  $N - M - 1$  个无人机，先选取步骤 2 中已经定位好的其中 2 架无人机，利用这 2 架无人机和参考无人机，根据定理，即可确定其余的  $N - M - 1$  个全部无人机。

现在将该策略应用于题目中所给的锥形无人机编队，如图6-2所示。

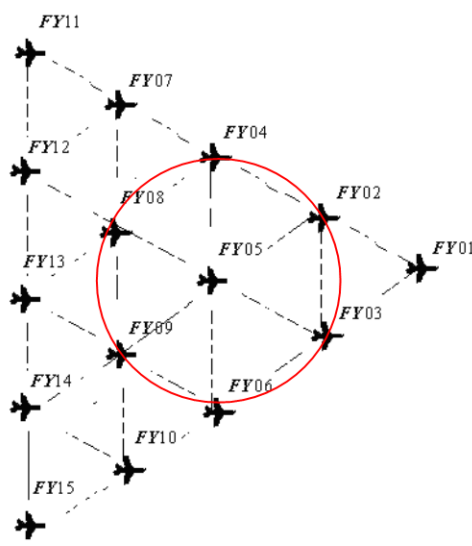


图 6-2 锥形无人机编队示意图

该编队无人机数目为 15，其位置调整策略如下：

- 选取 FY05 为参考无人机。原因是以 FY05 为圆心时，有 6 架无人机在同一个圆周上，分别为 FY02、FY03、FY04、FY06、FY08、FY09。
- 在圆周上的无人机数目为 6，根据问题 1 的第 3 小问，可以一共调整 7 个无人机的位置到达理想位置（包括参考无人机）。
- 选取步骤 2 中已经定位好的其中 2 架无人机，例如 FY02 和 FY03，利用这 2 架无人机和参考无人机，根据定理，可以确定其余所有位置上的参考无人机。

此时位置调整完毕。

除了锥形编队外，也可以将模型应用于没有固定几何形状的无人机编队。国庆来临，若某学校拟使用无人机编队来构造出五星红旗进行庆祝，如图6-3所示。

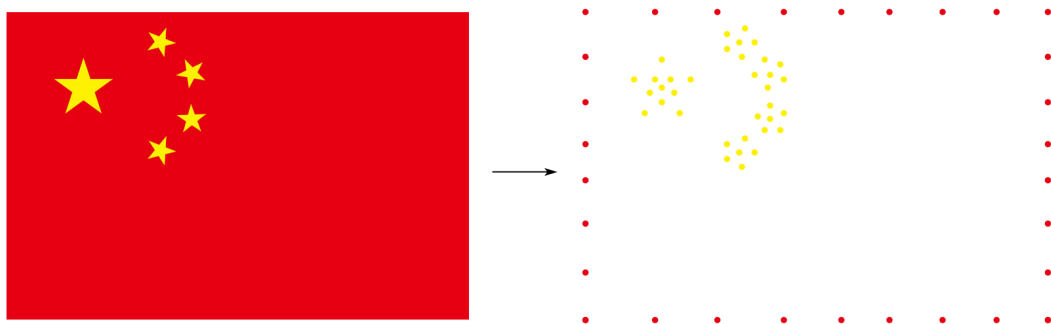


图 6-3 红旗无人机编队示意图

应用我们的策略进行位置调整和队形保持，策略如下：

- 选取无人机  $O$  为参考无人机。可以精确定位并调整粉色圈中的所有无人机。
- 在粉色圈中任选两个无人机，与参考无人机一起可精确定位并调整  $ABCD$  四个无人机。
- 以无人机  $ABCD$  为圆心，可精确定位并调整所有蓝色圈中的无人机。
- 利用定理，可以精确定位并调整其余的所有无人机。

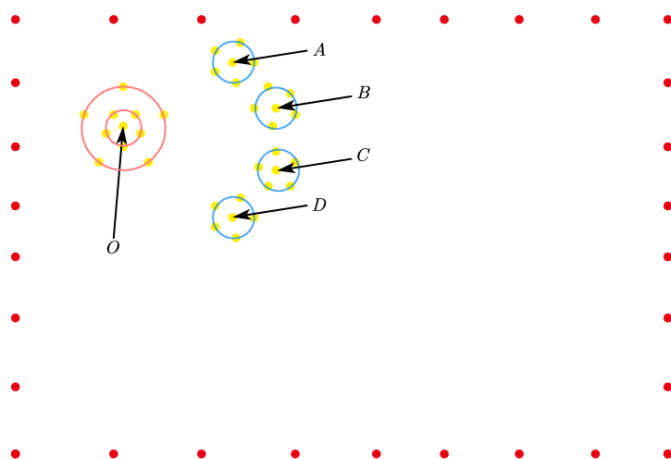


图 6-4 红旗无人机编队调整示意图

此时位置调整完毕。可以发现，本问的模型具有较强的可行性与普适性。

## 七、模型的评价

### 7.1 模型的优点

- (1) **具有普适性。**本文从代数的角度去建立和求解模型，而非几何分类讨论的角度，从代数角度建立的模型可用于任何几何情况，模型更具有普适性。

- (2) **具有说服力。**本文模型采用蒙特卡洛模拟，可以得到较为直观的表格或图片，如果仅从数学公式角度分析，难以给出直观结果让人信服。
- (3) **利用已有模型求解未知模型，更方便简洁。**问题二中对于任意编队问题，如果重新建立坐标系计算，则会十分麻烦，且无普适性。直接采用问题一的模型，将问题二拆分为多个问题一，直接省去了不必要的计算。
- (4) **考虑全面。**在问题二中，题目只给出简单几何编队即锥形编队，我们又另外考虑了非简单几何且无规律的编队情况，并给出相应策略。模型缺点

## 7.2 模型的缺点

- (1) 在进行小量近似后，求解方程组问题化简为求解单一方程问题，但未给出单一方程的解析解，略有遗憾。
- (2) 在最后的编队策略模型，如果某一编队没有一个局部圆形，则模型不适用。
- (3) 采用蒙特卡洛模拟，数据量大，程序运行耗时较长

## 参考文献

- [1] 王本才, 王国宏, 何友. 多站纯方位无源定位算法研究进展 [J]. 电光与控制, 2012, 19(05): 56-62.
- [2] 《运筹学》教材编写组. 运筹学. 第 4 版 [M]. 清华大学出版社, 2012.
- [3] 姜启源, 谢金星, 叶俊. 数学模型. 第 4 版 [M]. 高等教育出版社, 2011.
- [4] 韩中庚. 数学建模方法及其应用-第 2 版 [M]. 高等教育出版社, 2009.



## 附 录

### 1) 附录一：支撑材料清单

- 问题一 (2)/solve2.m: 问题一 (2) 的蒙特卡洛模拟，验证只需要两个未知编号的无人机
- 问题一 (2)/positionsolve2.m: 在 solve2.m 中调用的一个函数，用来求解方程数值解
- 问题一 (3)/Loss3.m: 方位调整优化策略时的目标函数
- 问题一 (3)/positionsolve2.m: 同上，文件内容一样
- 问题一 (3)/solve3\_1.m: 问题一 (2) 的定位无人机选择模拟
- 问题一 (3)/solve3.m: 问题一 (2) 的位置调整优化算法
- 问题一 (3)/positionsolve3.m: 在 solve3.m 中调用的一个函数，用来求解方程数值解

### 2) 附录二：主要代码

```
function F = Loss3(x,theta,R)
F=0;
for i=1:8
    F=F+sqrt(((x(i)-R)/R)^2+(((theta(i)-theta(i+1))*180/pi+40)/40)^2);
    %fprintf("r:%f\n",((x(i)-R)/R)^2);
    %fprintf("theta:%f\n",(((theta(i)-theta(i+1))*180/pi+40)/40)^2);
end
end
```

```
function F = positionsolve3(x,a1,a2,theta1,theta2)
F(1) =
    cos(a1)-(x(1)-cos(theta1-x(2)))/(sqrt(1+x(1)*x(1)-2*x(1)*cos(theta1-x(2))));
F(2) =
    cos(a2)-(x(1)-cos(theta2-x(2)))/(sqrt(1+x(1)*x(1)-2*x(1)*cos(theta2-x(2))));
end
```

```
clc;clear;
iter_max=1000; %迭代上限次数
loss_min=1e-5; %迭代终止误差

lr=0.2;      %步伐
T=1e10;     %模拟退火初始温度
```

```

tk=0.99; %模拟退火衰减温度
rec_t=5; %半径修正周期数
R=100; %想要收敛到的半径大小

draw_loss=1; %loss图的绘制
draw_1=0; %绘图1的绘制开关（绘图1和2的开关每次只能开1个）
draw_2=0; %绘图1的绘制开关（绘图1和2的开关每次只能开1个）
draw_itv=1; %绘图步长
theta_con=0;
loss_fake=[];
loss_true=[];
loss_record=[];
status=[0, 0;
        100, 0;
        98, 40.10;
        112, 80.21;
        105, 119.75;
        98, 159.86;
        112, 199.96;
        105, 240.07;
        98, 280.17;
        112, 320.28;];
x=zeros(9,1);
theta=zeros(9,1); %rad
target_theta=zeros(9,1);%rad
real_iter_num=0; %真实迭代次数

for i=1:9
    target_theta(i)=((i-1)*40+theta_con)*pi/180;
end
for i=1:9
    x(i)=status(i+1,1);
    theta(i)=status(i+1,2)*pi/180;
end

```

```

count_jump=0;
for i =1:iter_max
    T=T*tk;
    send=randperm(9);
    x_cal=zeros(9,1);
    theta_cal=zeros(9,1);
    x_cal(send(1))=R;
    theta_cal(send(1))=target_theta(send(1));
    x_cal(send(2))=R;
    theta_cal(send(2))=target_theta(send(2));
    loss_f=0;
    loss_t=0;
    for j=1:9 %对除了发信号的两架飞机调整位置
        %初始贪心模块
        send(1)=mod(j,9)+1;
        send(2)=j-1;
        if(j==1)
            send(2)=9;
        end
        if(j==send(1)||j==send(2))
            continue;
        end
        %fprintf("j=%d,send(1)=%d,send(2)=%d\n",j,send(1),send(2));
        %数据生成
        theta0=theta(send(1));
        theta1=theta(send(2));
        thetaM=theta(j);
        xM=x(j);
        R1=x(send(1));
        R2=x(send(2));
        alpha0=acos((xM-R1*cos(theta0-thetaM))/
            (sqrt(R1^2+xM^2-2*R1*xM*cos(theta0-thetaM))));
        alpha1=acos((xM-R2*cos(theta1-thetaM))/
            (sqrt(R2^2+xM^2-2*R2*xM*cos(theta1-thetaM))));
    end
end

```

```

%飞机求解
f = @(x) positionsolve3(x,alpha0,alpha1,
(send(1)-1)*40*pi/180,(send(2)-1)*40*pi/180);

x0 = [1,(j-1)*40*pi/180];
[ansx,fval,exitflag] = fsolve(f,x0,optimset('MaxFunEvals',1e30));
xm=ansx(1)*R;
thetam=ansx(2);

x_cal(j)=xm;
theta_cal(j)=thetam;
end
%如果到达了rec_t, 进行半径修正
if(mod(i,rec_t)==1)
    avg_r=0;
    %avg_theta=0;
    for k=1:9
        avg_r=avg_r+x(k);
        %avg_theta=avg_theta+(theta(k)-(k-1)*40*pi/180);
    end
    avg_r=avg_r/9.0;
    %avg_theta=avg_theta/9.0;
    fprintf("-----avg_r: %f-----\n",avg_r);
    for k=1:9
        x_cal(k)=x_cal(k)+(R-avg_r);
        x(k)=x(k)+(R-avg_r);
        %theta_cal(k)=theta_cal(k)+(target_theta(k)-avg_theta);
        %theta(k)=theta(k)+(target_theta(k)-avg_theta);
    end
end
end
loss_f=Loss3(x_cal,theta_cal,R);
if(i~=1)

```

```

        delta_loss_f=loss_f-loss_fake(i-1);
end
if(i==1||delta_loss_f<0||rand(1)<exp(-delta_loss_f/T))
    x=x+lr*(R-x_cal);
    theta=theta+lr*(target_theta-theta_cal);
    if(i~=1&&delta_loss_f>=0)
        count_jump=count_jump+1;
    end
    fprintf("退火成功\n");
end
fprintf("退火失败\n");
fprintf("====新的位置: iter %d====\n",i);
%记录loss
loss_t=Loss3(x,theta,R);
loss_fake(i)=loss_f;
loss_true(i)=loss_t;
loss_record(i)=loss_t;

%绘图步长的调整
if (10 < i && i <= 40)
    draw_itv = 3;
elseif (40 < i && i <= 200)
    draw_itv = 30;
elseif (200 < i)
    draw_itv = 1;
end
%图1的绘制
if (draw_1==1 && mod(i,draw_itv)==0)
    plot_figure1(x,theta,i);
end
%图2的绘制
if (draw_2==1 && mod(i,draw_itv)==0)
    plot_figure2(x,theta,i);
end
end

```

```

    real_iter_num=real_iter_num+1;
    if(loss_f<loss_min)
        break;
    end
end

fprintf("*****迭代结果*****\n");
fprintf("总迭代次数: %d\n",real_iter_num);
fprintf("退火概率: %f\n",count_jump/iter_max);
for k=1:9
    fprintf("(%f, %f)\n",x(k),theta(k)*180/pi);
end
if(draw_1==1)
subplot(3,3,1);xlabel('x');ylabel('y');title('FY01');
xlim([96,102.5]);ylim([0,6.5]);
subplot(3,3,2);xlabel('x');ylabel('y');title('FY02');
xlim([74,80.5]);ylim([60,66.5]);
subplot(3,3,3);xlabel('x');ylabel('y');title('FY03');
xlim([11,26]);ylim([97,112]);
subplot(3,3,4);xlabel('x');ylabel('y');title('FY04');
xlim([-56,-47]);ylim([85,94]);
subplot(3,3,5);xlabel('x');ylabel('y');title('FY05');
xlim([-95,-91]);ylim([33,37]);
subplot(3,3,6);xlabel('x');ylabel('y');title('FY06');
xlim([-106,-93]);ylim([-39,-26]);
subplot(3,3,7);xlabel('x');ylabel('y');title('FY07');
xlim([-53,-45]);ylim([-91,-83]);
subplot(3,3,8);xlabel('x');ylabel('y');title('FY08');
xlim([16,19]);ylim([-99,-96]);
subplot(3,3,9);xlabel('x');ylabel('y');title('FY09');
xlim([76,86.5]);ylim([-72,-62.5]);
end
if(draw_2==1)

```

```

    title('所有无人机的位置')
    xlabel('x');ylabel('y')
end
%loss图的绘制
if(draw_loss ==1)
    plot(1:120,loss_record(1:120),'LineWidth',1.5);
    title('Loss下降曲线');
    xlabel('迭代次数');
    ylabel('Loss')
end

```

```

clear;clc
iter=10000;
deta_m=zeros(100,1);
deta_t1=zeros(100,1);
figure(1)
for k = 1 : 4
    for i=1:iter
        theta=randperm(8);
        theta0 = 0;
        theta1 = pi / 180 * 40 * theta(1);
        thetaM = theta1 + k * 40 * pi / 180 + (-1 + 2 * rand(1)) * pi / 180;
        xm = 1 + 0.1 * (-1 + 2 * rand(1));
        R=1;
        alpha0=acos((xm-R*cos(theta0-thetaM))/
            (sqrt(R^2+xm^2-2*R*xm*cos(theta0-thetaM))));
        alpha1=acos((xm-R*cos(theta1-thetaM))/
            (sqrt(R^2+xm^2-2*R*xm*cos(theta1-thetaM))));

        y = (-1 + 2 * rand(1)) * pi / 180 ; % theta1 的偏移量
        f1 = @(x) positionsolve2(x,alpha1,alpha0,theta1+y);
        x0 = [1,thetaM];
        [x] = fsolve(f1,x0,optimset('MaxFunEvals',1e30));
        deta_m(i) = abs(x(2)-thetaM);
    end
end

```

```

        deta_t1(i) = y;
    end
    subplot(2,2,k);
    scatter(deta_m * 180 / pi , deta_t1 * 180 / pi);
    xlim([0 .05] * 180 / pi);
    ylim([-0.02 0.02] * 180 / pi);
    txt = xlabel('$\Delta \theta_m^-$');
    set(txt, 'Interpreter', 'latex');
    txt = ylabel('$\Delta \theta_i^-$');
    set(txt, 'Interpreter', 'latex');
end

```

为避免赘述，其余代码详见支撑材料。