

# 无人机遂行编队飞行中的纯方位无源定位

## 摘 要

在现代无人机编队飞行任务中，为最大化隐蔽性和安全性，无人机群体需尽量减少向外界发射电磁波信号，以避免被敌方探测。所以需要在保持电磁静默的同时，确保无人机能够准确维持或调整至正确的编队位置。为此，我们开发了一种纯方位无源定位方法。通过编队内部部分无人机发射信号，其余无人机被动接收信号并根据接收到的方向信息进行自我位置调整，实现了编队飞行中的精确定位与队形保持。

对于问题一第 1 小问，我们使用**两圆相交的定位方法**，在极坐标系下，给出了待定位无人机的极坐标  $(\rho, \theta)$  应满足的方程（见式4.8），并给出了  $\tan \theta$  的表达式（见式4.10）。该方程组中有两个参数  $\theta_{0N1}$  和  $\theta_{0NM}$  需要确定。为此，我们根据待定位无人机与发射信号的无人机的几何关系，分类讨论，给出了不同情况下  $\theta_{0N1}$  和  $\theta_{0NM}$  的取值（见表4.1）。最后，我们根据待定位无人机与 FY01 的几何关系，给出了  $\theta$  的表达式（见表4.2），在此基础上，我们可以给出  $\rho$  的表达式（见式4.11）。

对于问题一第 2 小问。我们提出了位置些许偏差的接收信号无人机根据接收的角度信息估计发射信号无人机序号的方法。并在此基础上得到了结论：1. 在位置偏差满足式5.4，除 FY00 和 FY01 外，再有 **2 架**序号未知的无人机提供角度信息就可以正确地判断自身位置；2. 在**待定位无人机极角正确或额外无人机关于待定位无人机与 FY01 对称**的特殊情况下，如果位置偏差满足式5.4再有 **1 架**序号未知的无人机提供角度信息就可以正确地判断自身位置。

对于问题一第 3 小问。首先，我们先提出了在角度指引下无人机移动的基本模式：在接收两个角度信号时，保持其中一个角度不变、调整自身位置使得另一个角度为目标数值。接下来，我们提出了“**半径矫正步**”和“**角度矫正步**”两种调整模式。“半径矫正步”在假设参与调整的无人机极角正确，调整极径；“角度矫正步”则通过调整接收到的角度到理想角度的方式调整角度。在此基础上，我们设计了使用“半径矫正步”进行预处理，然后使用“角度矫正步”进行迭代调整的调整策略。并通过模拟发现其可以稳定地工作，并且加入“半径矫正步”作为预处理可以加快收敛速度（见图6.5、图6.6和图6.7）。进一步地，我们提出了优化后的策略，也就是先通过类似的方式调整 FY01、FY04 和 FY07 的位置到正确位置，其余无人机的位置直接由标准角度确定。这种调整方式有单次迭代调整次数更少，迭代收敛速度更快、鲁棒性更强等优点（见图6.8）。

对于问题二。对于初始位置略有偏差的锥形无人机编队，我们沿用上一问中优化方案的策略。先将 FY01、FY07、FY10 通过与上一问相同的策略调整到以 FY05 为圆心且等分圆周的位置上。然后其余无人机的位置直接由标准角度确定。并通过模拟验证了这种调整方案的正确性与可行性（见图7.2和图7.3）。

**关键词：**几何关系、纯方向无源定位、无人机编队调整、迭代调整策略

# 1 问题重述

在现代无人机编队飞行任务中，为最大化隐蔽性和安全性，无人机群体需尽量减少向外界发射电磁波信号，以避免被敌方探测。所以需要在保持电磁静默的同时，确保无人机能够准确维持或调整至正确的编队位置。

因此题目要求我们，开发一种纯方位无源定位方法，通过编队内部部分无人机发射信号，其余无人机被动接收信号并根据接收到的方向信息进行自我位置调整，以实现编队飞行中的精确定位与队形保持。

首先，我们先关注一个由 10 架无人机组成的编队，其中 1 架位于圆心，剩下的 9 架无人机均匀分布在圆周上。其编号以及示意图如图1.1所示。

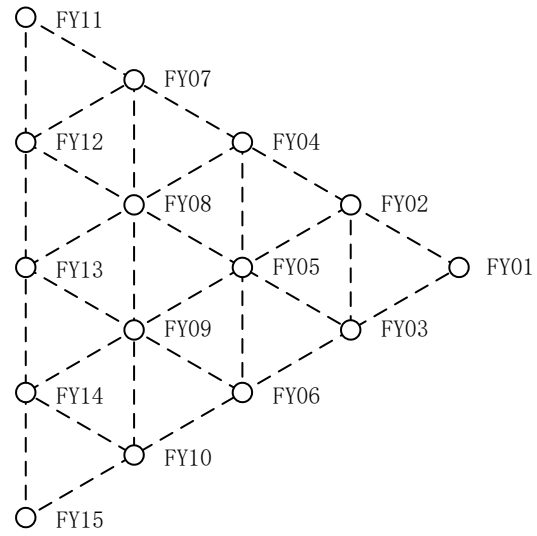
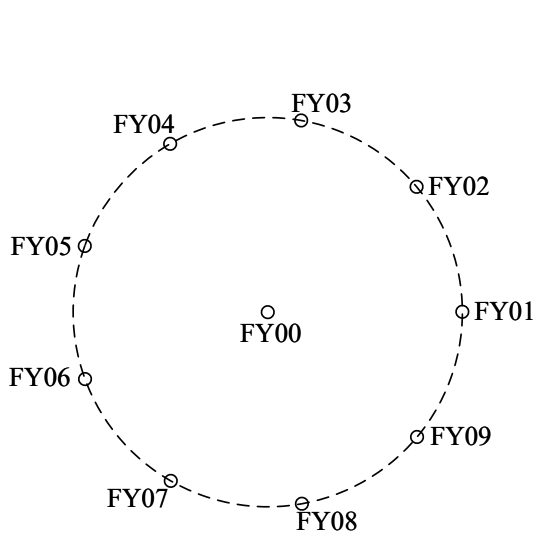


图 1.1: 圆周无人机编队以及编号示意图      图 1.2: 锥形无人机编队以及编号示意图

对此，题目要求我们完成以下任务：

1. 位于圆心的一架飞机和位于圆周的两架飞机发射信号。接受信号的无人机可以接收到该无人机和圆周上发射信号无人机的连线，与该无人机和位于圆心的无人机连线之间的夹角。假设发射信号的无人机位置精确，接受信号的无人机位置略有偏差。建立接受信号无人机的定位模型，确定无人机的位置。
2. 某位置略有偏差的无人机接收到 FY00 和 FY01 的信号，假设发射信号的无人机位置无偏差，其需要再接受几架编号位置的其他无人机发射的信号，才能实现无人机位置的准确定位。
3. 按编队要求，1 架无人机位于圆心，另 9 架无人机均匀分布在半径为 100 m 的圆周上，但初始位置略有偏差。需要我们给出调整方案，通过多次的迭代调整，使得最终外围的 9 架无人机可以均匀分布在某个圆周上。每次调整我们可以选择圆心

和圆周上的三架无人机发射信号，其余无人机接收信号，根据接收到的方向信息进行自我位置调整。

之后，我们关注一个如图1.2所示的锥形编队。题目要求我们，仍然考虑无源定位的情况，并给出调整的方案。

## 2 基本假设

我们有如下的基本假设：

- 无人机之间的信息不能共享；
- 发射信号的无人机不能同时接收信号；
- 每一架无人机都知道自己的编号；
- 每一架无人机都知道不同编号无人机的相对位置关系；
- 除特殊要求外（问题一第 2 小问），接收信号的无人机知道每一个接收的角度信息来源于哪两架无人机；
- 在以 FY00 为原点的极坐标系中，圆周上无人机的极角偏差小于  $\frac{1}{18}\pi$ ；
- 无人机只能接收角度信息，也只能依靠角度的信息来引导自己的移动，其基本移动模式为，接收两个角度信号，保持一个角度不变，调整自身位置使得另一个角度为目标数值；

## 3 符号说明

见表3.1。表中列举了全文中的常用符号，部分符号在全文中会有不同的含义，这些符号在会在使用时进行说明。

表 3.1: 本文所用符号说明

符号	符号说明	备注
$r_0$	无人机编队的圆半径	——
FY0N	序号为 $n$ 的无人机	可用其他字母替代
$\alpha_{ABC}$	FY0B 接收的来自于 FY0A 和 FY0C 的角信号	可用数字或其他字母替代
$\theta_{ABC}$	FY0A、FY0B、FY0C 三点共圆的圆心极角	可用数字或其他字母替代
$\rho_{ABC}$	FY0A、FY0B、FY0C 三点共圆的圆心极径	可用数字或其他字母替代
$d_i$	FY0I 当前位置与正确位置间的距离	可用数字或其他字母替代

## 4 问题一 (1): 接受信号无人机的定位模型

### 4.1 基于两圆相交的定位模型

由于圆的对称性, 我们不妨设其中一个发射信号的无人机为 FY01。如图4.1所示, 位于圆心的 FY00、位于圆周的 FY01 和另一架位于圆周但序数任意的无人机 FY0M 发射信号, FY0N 接受信号。

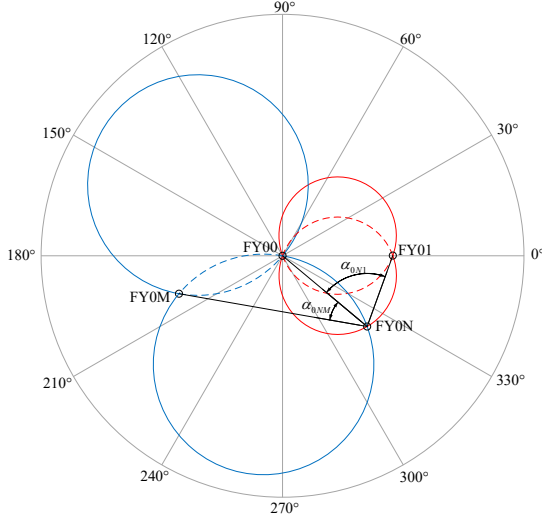


图 4.1: 接受信号无人机的定位模型

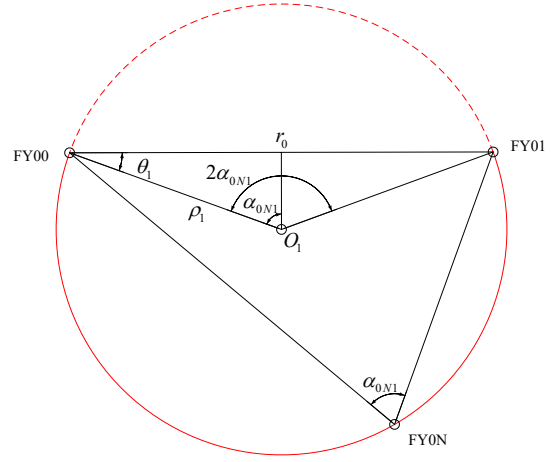


图 4.2: 单个圆的几何分析

FY0N 可以接收到两个角度信号  $\alpha_{0N1}$  和  $\alpha_{0NM}$ 。根据简单的几何知识 (圆周角相等) 我们可以知道, 满足  $\alpha_{0N1}$  为固定值的所有点都在图4.1中所示的红色实线上, 其是两段圆弧组合而成。同理, 满足  $\alpha_{0NM}$  为固定值的所有点都在图4.1中所示的蓝色实线上。

我们取 FY00 为极点, FY00 为起点指向 FY01 的射线为极轴, 构建极坐标系, 设 FY0M 的序数为  $m$ 。设 FY01 的极坐标为  $(r_0, 0)$ , 则 FY0M 的极坐标为  $(r_0, \frac{2}{9}(m-1)\pi)$ 。其中,  $r_0$  为无人机编队所在圆的半径。

接下来, 我们以红色下半部分圆弧为例, 来讨论圆心的坐标, 其局部图如图4.2所示。从中我们可以得到, 确定圆心坐标的两个重要的参数:

$$\rho_1 = \frac{r_0}{2 \sin \alpha_{0N1}} \quad (4.1)$$

$$\theta_1 = \frac{\pi}{2} - \alpha_{0N1} \quad (4.2)$$

这样, 我们可以得到  $O_1$  的极坐标为:

$$A = (\rho_1, -\theta_1) = \left( \frac{r_0}{2 \sin \alpha_{0N1}}, -\frac{\pi}{2} + \alpha_{0N1} \right) \quad (4.3)$$

同理, 我们可以得到与其对应的圆的圆心坐标, 以及蓝色圆弧的圆心坐标。设 FY00, FY01 和 FY0N 组成的圆的圆心坐标为  $A_{0N1}$ , FY00, FY0M 和 FY0N 组成的圆的圆心坐标为  $A_{0NM}$ , 则其可以表示为:

$$A_{0N1} = \left( \frac{r_0}{2 \sin \alpha_{0N1}}, 0 \pm \left( \frac{\pi}{2} - \alpha_{0N1} \right) \right) \quad (4.4)$$

$$A_{0NM} = \left( \frac{r_0}{2 \sin \alpha_{0NM}}, \frac{2}{9} (m-1) \pi \pm \left( \frac{\pi}{2} - \alpha_{0NM} \right) \right) \quad (4.5)$$

接下来, 我们求取圆在极坐标系下的方程。我们知道, 极坐标下, 圆心在  $(\rho_0, \theta_0)$ , 半径为  $r_0$  的圆的一般方程为:

$$\rho^2 + \rho_0^2 - 2\rho\rho_0 \cos(\theta - \theta_0) = r_0^2 \quad (4.6)$$

我们将两个圆的信息带入, 可以得到如下方程组:

$$\begin{cases} \rho^2 + \rho_{0N1}^2 - 2\rho\rho_{0N1} \cos(\theta - \theta_{0N1}) = r_0^2 \\ \rho^2 + \rho_{0NM}^2 - 2\rho\rho_{0NM} \cos(\theta - \theta_{0NM}) = r_0^2 \end{cases} \quad (4.7)$$

整理可以得到 (显然原点是一个解, 我们不考虑原点的情况, 所以将  $\rho$  约去):

$$\begin{cases} \rho = \frac{r_0}{\sin \alpha_{0N1}} (\cos \theta \cos \theta_{0N1} + \sin \theta \sin \theta_{0N1}) \\ \rho = \frac{r_0}{\sin \alpha_{0NM}} (\cos \theta \cos \theta_{0NM} + \sin \theta \sin \theta_{0NM}) \end{cases} \quad (4.8)$$

整理可以得到:

$$\cos \theta \frac{\cos \theta_{0N1}}{\sin \alpha_{0N1}} - \cos \theta \frac{\cos \theta_{0NM}}{\sin \alpha_{0NM}} = \sin \theta \frac{\sin \theta_{0NM}}{\sin \alpha_{0NM}} - \sin \theta \frac{\sin \theta_{0N1}}{\sin \alpha_{0N1}} \quad (4.9)$$

进一步化简可以得到:

$$\tan \theta = \frac{\frac{\cos \theta_{0N1}}{\sin \alpha_{0N1}} - \frac{\cos \theta_{0NM}}{\sin \alpha_{0NM}}}{\frac{\sin \theta_{0NM}}{\sin \alpha_{0NM}} - \frac{\sin \theta_{0N1}}{\sin \alpha_{0N1}}} \quad (4.10)$$

从图1.1中可以看出, 在无人机的位置仅略有偏差的情况下, 应该不会出现在  $\tan \theta$  的值不存在的情况 (这要求 FY03 和 FY08 的偏移不能超过 10 度), 我们这里使用  $\tan \theta$  的值来确定无人机的位置是合理的。

得到 FY0N 的在极坐标系下的角度信息后, 我们将其带回式4.8中任意一个式子, 就可以得到  $\rho$  的取值。比如:

$$\rho = \frac{r_0}{\sin \alpha_{0N1}} (\cos \theta \cos \theta_{0N1} + \sin \theta \sin \theta_{0N1}) \quad (4.11)$$

接下来, 我们所需要解决的还有两个问题:

1.  $\theta_{0N1}$  和  $\theta_{0NM}$  的取值有两个, 我们需要确定选择哪一个;
2. 在 0 到  $2\pi$  范围内, 有两个  $\theta$  的取值使得  $\tan \theta$  为某一特定值, 我们需要确定选择哪一个。

## 4.2 特定情况下的讨论

首先，我们来确定  $\theta_{0N1}$  和  $\theta_{0NM}$  的具体取值。那么我们首先要讨论  $\alpha_{0N1}$  和  $\alpha_{0NM}$  的取值范围。

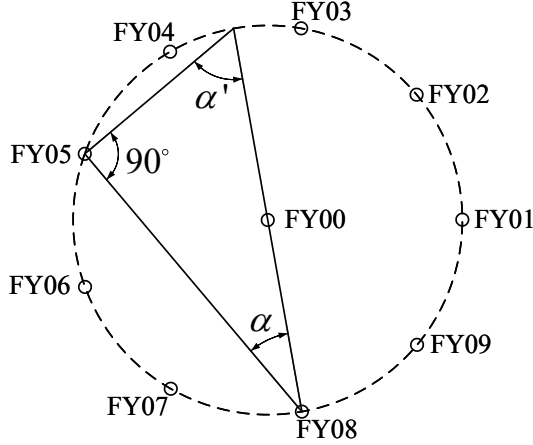


图 4.3:  $\alpha$  取值范围示意图

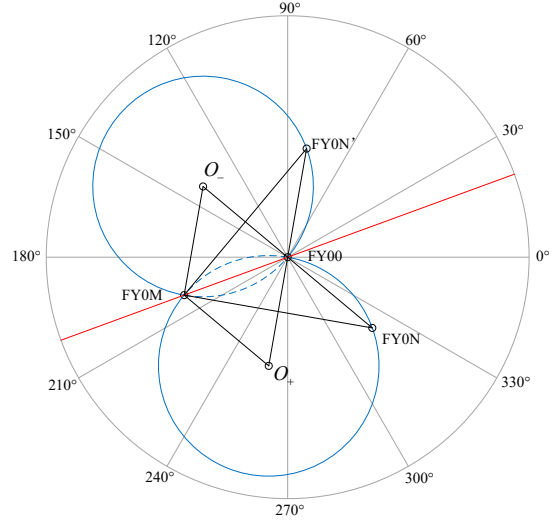


图 4.4:  $\theta$  取值示意图

如图4.3所示，对于任意的一个  $\alpha$ ，我们都会有：

$$\alpha + \alpha' = \frac{\pi}{2} \quad (4.12)$$

又显然有  $\alpha$  和  $\alpha'$  都是大于 0 的，所以

$$0 < \alpha < \frac{\pi}{2} \quad (4.13)$$

这对于，所有的圆周上无人机所接收到的角度信号都是成立的，所以：

$$0 < \alpha_{0N1} < \frac{\pi}{2} \quad (4.14)$$

$$0 < \alpha_{0NM} < \frac{\pi}{2} \quad (4.15)$$

现在我们观察图4.4。我们从中可以看到，对于  $\theta_{0NM}$  来说， $\frac{2}{9}(m-1)\pi \pm (\frac{\pi}{2} - \alpha_{0NM})$  中取 + 还是 -，取决于，FY0N 位于 FY0M 和 FY00 连成的直线（图4.4中红色直线）的上方或者是下方（当正好位于直线上时，外接圆不存在）。设 FY0N 在极坐标系下的极角为  $\theta_n$  用数学语言表达就是：

$$\theta_{0NM} = \begin{cases} \frac{2}{9}(m-1)\pi + \left(\frac{\pi}{2} - \alpha_{0NM}\right) & , 0 < \theta_n - \frac{2}{9}(m-1)\pi + 2k\pi < \pi \\ \frac{2}{9}(m-1)\pi - \left(\frac{\pi}{2} - \alpha_{0NM}\right) & , \pi < \theta_n - \frac{2}{9}(m-1)\pi + 2k\pi < 2\pi \end{cases} \quad (4.16)$$



其中,  $k \in \mathbb{Z}$ 。

如果我们认为略有偏差的位置, 不会影响 FY0N 与直线的位置关系 (这里要求相当宽松, 仅要求偏差小于  $20^\circ$ )。我们设 FY0N 无人机所对应的序数为  $n$ , 则  $\theta_n = \frac{2}{9}(n-1)\pi$ , 将其带入式4.16。可以得到:

$$\theta_{0NM} = \begin{cases} \frac{2}{9}(m-1)\pi + \left(\frac{\pi}{2} - \alpha_{0NM}\right) & , 0 < n - m + 9k < \frac{9}{2} \\ \frac{2}{9}(m-1)\pi - \left(\frac{\pi}{2} - \alpha_{0NM}\right) & , \frac{9}{2} < n - m + 9k < 9 \end{cases} \quad (4.17)$$

同理, 我们可以得到  $\theta_{0N1}$  在不同情况下的取值, 将  $m$  为 1 带入式4.17中即可。表4.1展示了对于不同的  $n$  和  $m$ ,  $\theta_{0N1}$  和  $\theta_{0NM}$  的取值。

表 4.1: $\theta_{0N1}$ 和 $\theta_{0NM}$ 的取值	
	$1 \leq n - m + 9k \leq 4$
	$5 \leq n - m + 9k \leq 8$
$2 \leq n \leq 5$	$\begin{cases} \theta_{0N1} = \frac{\pi}{2} - \alpha_{0N1} \\ \theta_{0NM} = \frac{2}{9}(m-1)\pi + \left(\frac{\pi}{2} - \alpha_{0NM}\right) \end{cases}$
$6 \leq n \leq 9$	$\begin{cases} \theta_{0N1} = -\frac{\pi}{2} + \alpha_{0N1} \\ \theta_{0NM} = \frac{2}{9}(m-1)\pi + \left(\frac{\pi}{2} - \alpha_{0NM}\right) \end{cases}$
	$\begin{cases} \theta_{0N1} = \frac{\pi}{2} - \alpha_{0N1} \\ \theta_{0NM} = \frac{2}{9}(m-1)\pi - \left(\frac{\pi}{2} - \alpha_{0NM}\right) \end{cases}$
	$\begin{cases} \theta_{0N1} = -\frac{\pi}{2} + \alpha_{0N1} \\ \theta_{0NM} = \frac{2}{9}(m-1)\pi - \left(\frac{\pi}{2} - \alpha_{0NM}\right) \end{cases}$

接下来, 我们还需要通过  $\tan \theta$  的值来确定  $\theta$  的取值。不能直接使用  $\arctan \tan \theta$  来确定  $\theta$  的值, 因为  $\arctan$  函数的值域是  $(-\frac{\pi}{2}, \frac{\pi}{2})$ , 而我们需要的  $\theta$  的值域是  $(0, 2\pi)$ 。在无人机的位置仅仅略有偏差 (这里要求偏差小于  $10^\circ$ , 否则会出现跨区域问题) 的情况下, 我们可以通过  $n$  的取值, 来判断  $\theta$  的取值范围, 然后再通过  $\tan \theta$  的值来确定  $\theta$  的具体取值。如表4.2所示。

表 4.2: $\theta$ 的取值	
$\theta$	
$n = 2, 3$	$\arctan \tan \theta$
$n = 4, 5, 6, 7$	$\arctan \tan \theta + \pi$
$n = 8, 9$	$\arctan \tan \theta + 2\pi$

综上所述, 在无人机位置的偏差角度小于  $10^\circ$  的情况下, 我们完成了题目要求的无人机的定位模型。

## 5 问题一 (2): 无人机有效定位条件

### 5.1 通过角度信息估算发射信号无人机序数

对于任意两个无人机, 假设其编号分别为  $n$  和  $m$ , 则如图5.1所示。

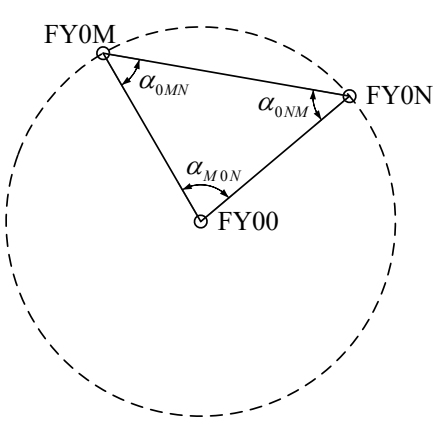


图 5.1: 两圆周无人机角度关系

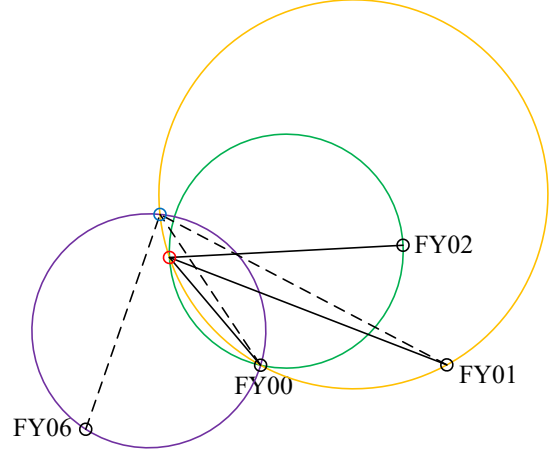


图 5.2: 仅有额外 1 架飞机的情况

如果, FY0M 和 FY0N 的位置是准确的, 因为 9 架无人机均匀分布在圆周上, 我们可以得到:

$$\alpha_{M0N} = \frac{2}{9}\pi |n - m| \quad (5.1)$$

根据等腰三角形的性质, 我们很容易可以得到:

$$\alpha_{0MN} = \alpha_{0NM} = \frac{1}{2}\pi - \frac{1}{9}\pi |n - m| \quad (5.2)$$

通过上面的式子, 即使在 FY0M 和 FY0N 的位置略有偏差的情况下, 我们也可以通过  $\alpha_{0MN}$  和  $\alpha_{0NM}$  的值, 来估计  $|n - m|$  的值。我们以  $\alpha_{0MN}$  为例, 我们可以得到:

$$|n - m| \approx 9 \left( \frac{1}{2} - \frac{\alpha_{0MN}}{\pi} \right) \quad (5.3)$$

为了使得对  $|n - m|$  的估计不出现错误,  $\alpha_{0MN}$  的误差应当小于  $\frac{1}{18}\pi$ , 也就是:

$$\Delta\alpha_{0MN} < \frac{1}{18}\pi \quad (5.4)$$

接下来的讨论都将以此为基础。

显然, 对于一个  $m$ , 有两个  $n$  的可能取值满足条件, 分别记为  $n_1$  和  $n_2$ , 为了简化后续叙述, 我们称 FY0N1 与 FY0N2 关于 FY0M 对称。



## 5.2 额外一架无人机的情况

现在我们来讨论，如图5.2所示的情况。FY04 的位置略有偏差，其实际位置位于图中红色圆圈标识的位置。其接收到了 FY00 和 FY01 所提供的角度信息，结合自身的序号，知道了自身位于橙色的圆周上。同时其接收到了一个未知来源的（实际来源于 FY02），角度约为  $\frac{7}{18}\pi$  的角度信息。其通过式5.3，估计出，信号源的序数与自己相差 2。则信号来源可能为 FY02 或者 FY06，则 FY04 可以确定自己的位置在图中绿色圆周或者紫色圆周上。但是 FY04 无法确定自身位置，有两个点符合上述的条件，一个是红色圆圈标识的位置，一个是蓝色圆圈标识的位置。所以只有额外一架无人机提供的角度信息是不够的。

为简便后续讨论，我们称因为对发射信号的无人机序数的误判而推导出的错误位置称为“幻觉”。

这种位置的不确定性起始来自于发射信号的无人机的位置不准确性。所以在以下两种情况下，接收信号的无人机可以确定自身的位置：

- 发射信号的无人机的位置确定。设 FY0M 接收信号，FY0N 为 FY0M 不知道序号的信号发射者。如果 FY0N 和 FY01 关于 FY0M 对称，则 FY0M 可以确定自己的位置，因为 FY01 不可能向它发射两个信号。
- 发射信号的无人机的位置不确定性不影响定位。“幻觉”位置和真实位置相同，也就是两个可能点的位置重合。这要求在以 FY00 为原点的极坐标中，待确定位置的无人机的极角是正确的。

我们现在来证明第二点，如图5.3。图中，O 为圆心无人机，N 为发射信号的无人机，N' 为可能被误认为是发射信号无人机的无人机。M 为任意满足  $\angle OMN$  为接收到信号角度的点，OS 为  $\angle NON'$  的角平分线，M' 为 M 关于 OS 的对称点。

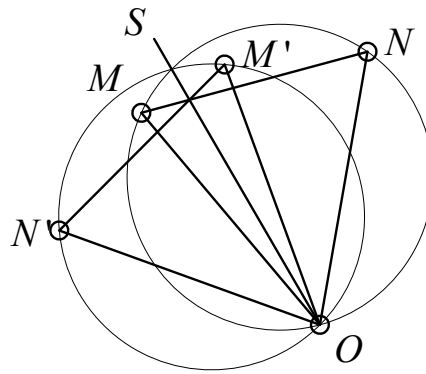


图 5.3: 幻觉的可能位置

由已有条件，我们可以得到： $OM = OM'$ ， $\angle MOS = \angle M'OS$ ， $\angle NOS = \angle N'OS$ 。我们可以得到：

$$\angle MON = \angle M'ON' = \angle MOS + \angle NOS = \angle M'OS + \angle N'OS \quad (5.5)$$

则可以根据，全等三角形的边角边定理，得到  $\triangle MON \cong \triangle M'ON'$ 。则可以得到：

$$\angle N'M'O = \angle NMO \quad (5.6)$$

这样构建起了一个到一一对应关系,对于一个点  $M$ ,我们可以找到一个点  $M'$ ,  $\angle N'M'O = \angle NMO$ 。我们知道，使得  $\angle NMO$  为固定值的点，在经过  $N$  和  $O$  且半径固定的圆上。则使得  $\angle N'M'O$  为相同固定值的点，在其关于  $OS$  对称的圆上。

如图所示，这两个圆交与 2 点，一个是点  $O$ ，另一个点在  $OS$  上。也就是说，只有点  $M$  在  $OS$  上时，才能满足  $\angle OMN = \angle OMN'$ 。

回到各个的问题，想要“幻觉”与真实位置相同，则需要满足  $\angle OMN = \angle OMN'$ 。所以，其必须在  $OS$  上，如果  $N$  和  $N'$  的位置是准确的，那么等价于  $M$  在极坐标系的极角是与正确位置相同的。

### 5.3 额外两架无人机的情况

接下来，我们讨论有额外两架发射信号的无人机的情况，即除了 FY00 和 FY01 外，还有序号分别为  $m$  和  $n$  的无人机 FY0M 和 FY0N 发射信号，序号为  $p$  的无人机 FY0P 接收信号，但是 FY0P 不知道除了 FY00 和 FY01 外的发射信号无人机的序号。序号为  $p$  的无人机接收到三个角度信息，分别为  $\alpha_{0PM}$ ， $\alpha_{0PN}$  和  $\alpha_{0P1}$ 。和上面相同，除了正确位置外， $\alpha_{0PM}$ ， $\alpha_{0PN}$  各会误导出一个错误的位置，我们称之为“幻觉”。只要两个“幻觉”的位置不同，FY0P 就可以通过投票的方式得到自己的真实位置。

下面，我们来讨论，两个幻觉的位置在什么时候会重合。

如果，待确定位置无人机的位置的在极坐标系下的极角是正确的，则根据上面的讨论，两个幻觉都和正确位置重合，FY0P 可以确定自己的位置。

如果，两个额外发射信号无人机中的任何一架，关于 FY0P 于与 FY01 对称，则 FY0P 可以立即确定自身位置。

我们现在来讨论不同于上面两个特殊情况的一般情况。假设两个“幻觉”重合于真实位置外的任意一点。则如图5.4所示，图中  $D$  是 FY01 位置， $P$  是真实位置， $M$  和  $N$  为实际发射信号无人机位置， $M'$  和  $N'$  为被误认为是发射信号无人机位置， $P'$  为两个“幻觉重合的位置”。

那么真实位置是红色两个圆的交点，两个“幻觉”重合于蓝色两个圆相交的点。因为两个蓝色的圆是两个红色的圆关于  $OS$  堆成得到的，则  $P$  和  $P'$  也关于  $OS$  对称。则可以得到

$$OP = OP' \quad (5.7)$$

因为无论是幻觉还是真实位置还需要满足：

$$\angle OPD = \angle OP'D \quad (5.8)$$

我们不讨论  $P'$  和  $P$  在  $OD$  两侧的情况，因为这至少要求 FY05 或 FY06 的极角偏差  $\frac{1}{9}\pi$ 。那么则  $P'$  和  $P$  在  $OD$  同侧， $O$ 、 $D$ 、 $P$ 、 $P'$  四点共圆。无论  $D$  在哪一侧，我们都



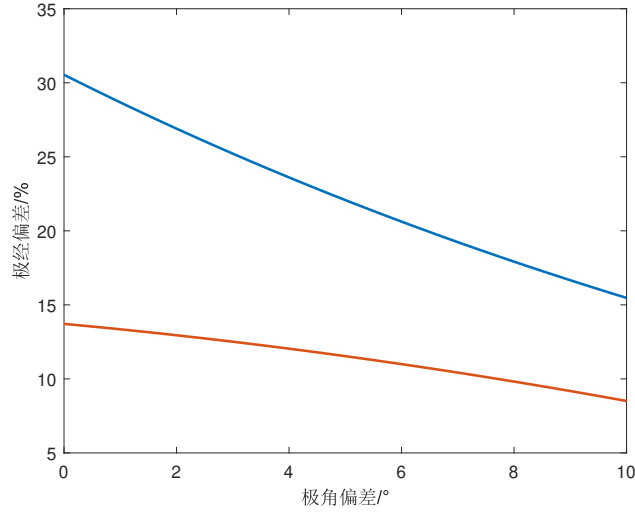


图 5.5: FY09 临界状态下极坐标偏差关系

## 6 问题一 (3): 无人机编队调整

### 6.1 在角度指引下的无人机移动

对于无人机位置的调整问题，我们首先需要处理的就是，如何根据接收到的角度信息来调整自己的位置。如图6.1所示。 $O$  为在圆心的发射信号的无人机， $L_1$  和  $L_2$  为在圆周上的两架发射信号的无人机， $R$  为需要通过接收的信号调整位置的无人机。在这个例子中， $R$  需要从  $R_1$  的位置移动到  $R_2$  的位置。

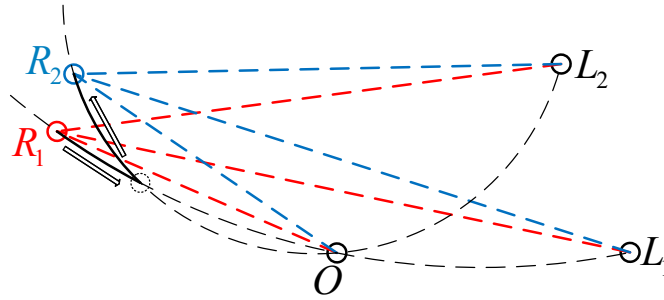


图 6.1: 角度指引下无人机调整示意图

我们给出如下的调整策略：

1. 保持  $\angle ORL_1$  不变为  $\angle OR_1L_1$ ，顺着梯度方向或梯度下降方向，将  $\angle ORL_2$  从  $\angle OR_1L_2$  调整至  $\angle OR_2L_2$ ；
2. 保持  $\angle ORL_2$  不变为  $\angle OR_2L_2$ ，顺着梯度方向或梯度下降方向，将  $\angle ORL_1$  从  $\angle OR_1L_1$  调整至  $\angle OR_2L_1$ 。

调整的路线如图6.1中黑色实线和箭头所示。通过这种基于角度指引的无人机移动方案，通过一架位于圆心和两架位于圆周上的无人机发射信号，可以将一架接收信号的无人机在小范围内进行调整，使得其接收到的两个角度信息为指定值。

接下来的无人机编队调整方案中，所有无人机位置的调整都将是基于这种在角度指引下的无人机移动的。

## 6.2 无人机编队调整方案

### 6.2.1 调整方案的层次结构

我们的无人机调整有如下层次结构：

- 针对单架无人机进行单一目的的调整称为“步”；
- 所有无人机（除“半径标定机”外）都进行了一“步”调整则称为“轮”。

我们的无人机调整方案共有两种“步”的调整方案：“角度矫正步”、“半径矫正步”。其原则如下：

- 角度矫正步：无人机接收两侧无人机和圆心无人机发出的信号。调整自身位置，使得自身接收到的角度信号为理想角度；
- 半径矫正步：无人机接收来关于“半径标定机”对称的无人机、“半径标定机”还有圆心无人机的信号，假设发射信号和接收信号的无人机角度信息是正确的。维持自身、FY00与“半径标定机”形成的夹角不变，调整自身位置，使自身距离FY00的距离与“半径标定机”距离FY00的距离相等。

### 6.2.2 角度矫正步

观察图6.2，其中， $O$  为圆心， $L_1$  和  $L_2$  为在圆周上的两架发射信号的无人机， $R$  为圆周上接收信号的无人机， $R'$  为我们希望  $R$  移动到的位置。

我们知道，在正确位置时，有

$$\angle R'OL_1 = \frac{2}{9}\pi \quad (6.1)$$

由于  $OL_1 = OR'$ ，所以我们可以得到：

$$\angle R'L_1O = \angle L_1R'O \quad (6.2)$$

则：

$$\angle R'L_1O = \angle L_1R'O = \frac{\pi - \frac{2}{9}\pi}{2} = \frac{7}{18}\pi \quad (6.3)$$

这样，我们得到了角度矫正步的调整方案：设任意一个无人机 FY0N，其顺时针方向的下一个无人机为 FY0N-1，逆时针方向的下一个无人机为 FY0N+1。FY0N 接收到的角度信息为  $\alpha_{0NN-1}$  和  $\alpha_{0NN+1}$ ，我们需要调整其至  $\alpha'_{0NN-1}$  和  $\alpha'_{0NN+1}$  满足如下条件：

$$\alpha'_{0NN-1} = \alpha'_{0NN+1} = \frac{7}{18}\pi \quad (6.4)$$

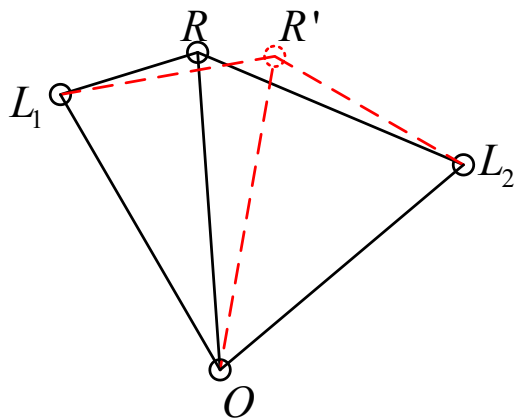


图 6.2: 角度矫正步示意图

### 6.2.3 半径矫正步

图6.3和图6.4展示了半径矫正步的两种不同情况。假设“半径标定机”为 FY01，则情况 1 对应待调整无人机序数为 2、3、8 和 9，情况 2 对应待调整无人机序数为 4、5、6 和 7。

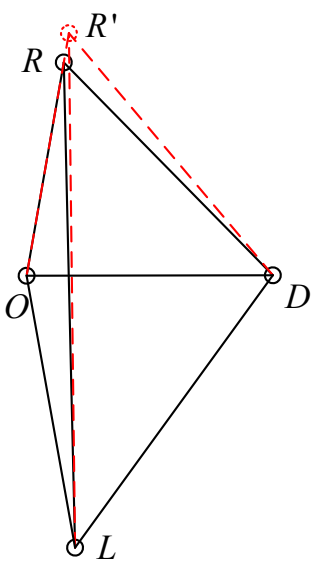


图 6.3: 半径矫正步情况 1

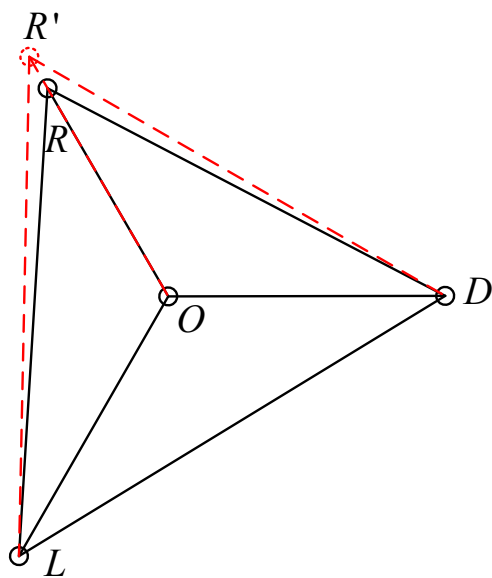


图 6.4: 半径矫正步情况 2

我们以情况 1 为例，我们假设  $R$  和  $L$  的角度信息都是对的。如果无人机序数  $n$  为 2 或者 3，我们可以得到：

$$\angle LOD = \angle ROD = \frac{2}{9}(n-1)\pi \quad (6.5)$$

如果无人机序数  $n$  为 8 或者 9，我们可以得到：

$$\angle LOD = \angle ROD = \frac{2}{9}(10 - n)\pi \quad (6.6)$$

由于  $OR' = OD$ ，所以我们可以得到：

$$\angle OR'D = \angle ODR' = \frac{\pi - \angle R'OD}{2} \quad (6.7)$$

通过对  $\triangle ORD$  使用正弦定理，可以得到：

$$OR = \frac{OD \sin(\pi - \angle ROD - \angle ORD)}{\sin \angle ORD} \quad (6.8)$$

然后通过对  $\triangle ORL$  使用正弦定理，可以得到：

$$OL = \frac{OR \sin \angle ORL}{\sin(\pi - \angle ROL - \angle ORL)} \quad (6.9)$$

其中， $\angle ROL = 2\angle ROD$  最后通过对  $\triangle OR'L$  使用正弦定理，可以得到方程：

$$\frac{OR'}{\sin(\pi - \angle ROL - \angle OR'L)} = \frac{OL}{\sin \angle OR'L} \quad (6.10)$$

整理可以得到：

$$\frac{\sin \angle ORL \sin(\angle ROD + \angle ORD) \sin(\angle ROL + \angle OR'L)}{\sin \angle ORD \sin(\angle ROL + \angle ORL) \sin \angle OR'L} = 1 \quad (6.11)$$

我们定义  $k$  为：

$$k = \frac{\sin \angle ORL \sin(\angle ROD + \angle ORD)}{\sin \angle ORD \sin(\angle ROL + \angle ORL)} \quad (6.12)$$

则式6.11可以变为：

$$\frac{\sin(\angle ROL + \angle OR'L)}{\sin \angle OR'L} = \frac{1}{k} \quad (6.13)$$

化简后整理可以得到：

$$\angle OR'L = \arctan \frac{\sin \angle ROL}{\frac{1}{k} - \cos \angle ROL} \quad (6.14)$$

我们依据  $\angle OR'L$  与  $\angle OR'D$  的指引，将接收信号的无人机移动到指定位置即可。

情况 2 的情况基本类似， $\angle ROD$  的计算方式当无人机序数为 4 和 5 时同式6.5，当计算无人机序数为 6 和 7 时同式6.6，唯一的区别在于  $\angle ROL = 2\pi - 2\angle ROD$ 。

这样，我们得到了半径校正步的调整方案：对于任意一个无人机，通过接收“半径标定机”和其关于“半径标定机”对称的无人机的信号，通过接收的信号，通过上述过程计算出计算出目标角度后，按照目标角度调整自身位置即可。

但是我们发现，该方法在无人机的半径基本正确而角度偏差较大的情况下并不稳定，所以仅作为预处理手段使用。



### 6.2.4 调整方案流程

我们的调整方案流程如下：

1. 首先选择“半径标定机”，其和原点（及 FY00）在之后的调整中不会改变位置；
2. 重复进行迭代矫正。每次重复时，执行一次“角度矫正轮”，除 FY00 和“半径标定点”外的所有无人机进行依次“角度矫正步”。如果是第一次执行，还会在“角度矫正轮”前，执行一次“半径矫正轮”，除 FY00 和“半径标定点”外的所有无人机进行依次“半径矫正步”。

### 6.3 调整方案的模拟

我们在题目给出的信息的基础上，对调整方案进行了模拟。为了衡量无人机组整体位置相对于理想位置的偏差，定义了如下的损失函数：

$$L = \sum_{i=0}^9 d_i^2 \quad (6.15)$$

其中， $d_i$  是序数为  $i$  的无人机，当前位置与理想位置的欧几里得距离。

我们还模拟了不采取“半径矫正轮”进行预处理的情况，作为对比，来说明“半径矫正轮”的重要性。

无人机在各个阶段的位置如图6.5所示。损失函数随迭代次数的变化如图6.6所示。损失函数的具体数值见表6.1，损失函数 1 代表不进行半径矫正时的损失函数，损失函数 2 代表进行半径矫正时的损失函数。

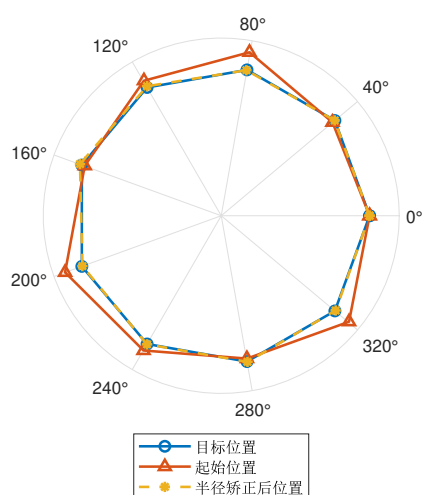


图 6.5: 无人机位置对比

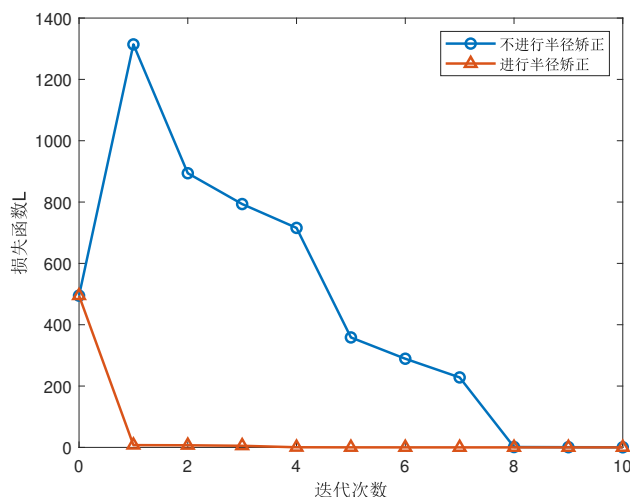


图 6.6: 损失函数变化情况

我们从图中可以看到，“半径矫正轮”可以非常有效地加速模型的收敛。且无论是否增加“半径矫正轮”，无人机都可以通过我们的算法最后逼近正确的位置。

表 6.1: 损失函数变化情况

迭代次数	损失函数 1	损失函数 2	迭代次数	损失函数 1	损失函数 2
0	$4.95 \times 10^2$	$4.95 \times 10^2$	6	$2.89 \times 10^2$	$4.04 \times 10^{-2}$
1	$1.31 \times 10^3$	$7.41 \times 10^0$	7	$2.28 \times 10^2$	$4.26 \times 10^{-5}$
2	$8.94 \times 10^2$	$6.88 \times 10^0$	8	$1.17 \times 10^0$	$3.32 \times 10^{-8}$
3	$7.93 \times 10^2$	$5.28 \times 10^0$	9	$6.18 \times 10^{-1}$	$1.98 \times 10^{-8}$
4	$7.16 \times 10^2$	$4.01 \times 10^{-1}$	10	$3.51 \times 10^{-1}$	$1.23 \times 10^{-8}$
5	$3.59 \times 10^2$	$8.99 \times 10^{-2}$			

当然，仅凭借 1 组数据不能说明模型的准确性，我们进行了更多的模拟，这些模拟中的数据为随机产生。我们首先将 10 个无人机的位置设置在正确位置，然后给除了 FY00 和 FY01 外所有的无人机的极坐标角度上加上一个符合  $N(0, 1)$  的偏移，极坐标半径上加上一个符合  $N(0, 10)$  的偏移，这个偏移量远远大于题目中给出的数据的偏移。重新进行了多次模拟，结果如图 6.7 所示，我们可以看到，我们的策略工作得是非常稳定的。

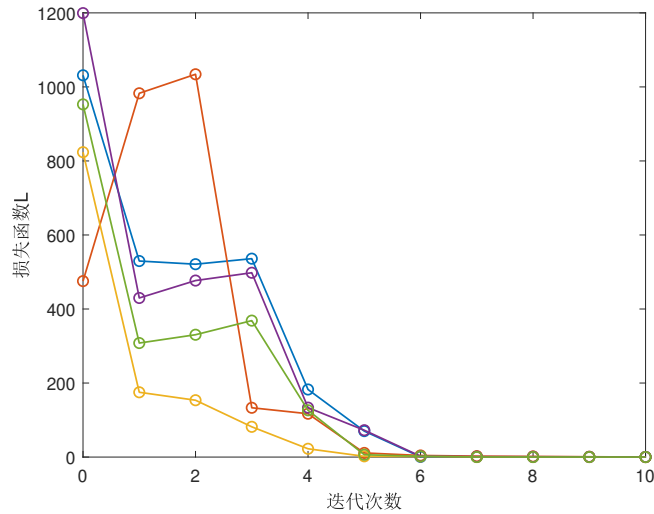


图 6.7: 圆形队列损失函数变化情况

## 6.4 收敛更快的调整策略

通过上面的讨论，我们已经已经知道了，三个不共线的位置精确的点，通过两个角度信息，可以确定两个点。若给定任意起始点，其可以沿着梯度方向，到达两个点中的一个。我们知道，对于任何队列来说，其中两个点的位置确定了，整个队列的正确位置就确定了。也就是我们需要使用两个无人机标定这个无人机队列。在这个问题中，我们选择了 FY00 和 FY01。所以我们只需要再确定一个位置精确的点，队列中所有的其他

点都可以直接通过角度确定。

在这个例子中，我们注意到，FY01、FY04 和 FY07 三等分了圆周，我们可以通过类似的迭代方法，使得 FY04 和 FY07 的位置逼近正确位置，然后其他无人机的位置就可以直接通过角度确定。这种方法有以下优点：

- 每次调整都有位置正确的 FY01 参与，可以加速收敛；
- 每次迭代都只有 2 次调整，可以加快迭代速度。

确定 FY01、FY04 和 FY07 位置后，其余无人机可以直接通过与 FY04 和 FY00 还有 FY07 和 FY00 之间的夹角，直接确定自身位置。我们记一架无人机与 FY00 和 FY04 形成的夹角为  $\alpha_4$ ，与 FY00 和 FY07 形成的夹角为  $\alpha_7$ 。则其应当调整至的标准角度如表6.2所示。

表 6.2: 圆形队列标准角度

无人机编号	$\alpha_4$	$\alpha_7$	无人机编号	$\alpha_4$	$\alpha_7$
FY02	$\frac{5\pi}{18}$	$\frac{\pi}{18}$	FY06	$\frac{5\pi}{18}$	$\frac{7\pi}{18}$
FY03	$\frac{7\pi}{18}$	$\frac{\pi}{18}$	FY08	$\frac{\pi}{18}$	$\frac{7\pi}{18}$
FY05	$\frac{7\pi}{18}$	$\frac{5\pi}{18}$	FY09	$\frac{\pi}{18}$	$\frac{5\pi}{18}$

#### 6.4.1 4 无人机圆周队列调整

圆周上有 3 个无人机与圆周上有 9 个无人机调整方案类似，但有些许不同。我们将 FY01 重新编号为 1，FY04 重新编号为 2，FY07 重新编号为 3。

则角度矫正步中理想角度应该改为：

$$\alpha'_{0NN-1} = \alpha'_{0NN+1} = \frac{2\pi - \frac{2}{3}\pi}{2} = \frac{1}{6}\pi \quad (6.16)$$

半径矫正步中，只会出现情况 2，且：

$$\angle LOD = \angle ROD = \frac{2}{3}\pi \quad (6.17)$$

其余不变。

#### 6.4.2 调整策略模拟

我们使用了与图6.7中相同的起始状态进行模拟。

由于我们的方案是在调整好 FY04 和 FY07 位置后，其他无人机立刻移动到正确的位置，并不好展现出迭代中逐渐逼近正确位置的过程。我们修改了我们的模拟方案，在 FY04 和 FY07 每次迭代调整结束之后，我们都会立即更新所有无人机的位置，然后计算损失函数。最终我们可以得到图6.8。

从中我们可以看出，这种优化后的方法可以非常好得加快收敛速度，可以使得无人机更快地逼近正确位置。

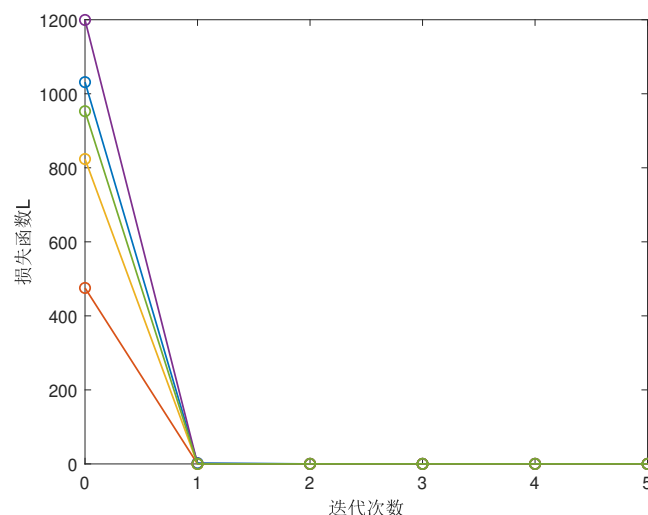


图 6.8: 优化后圆形队列损失函数变化情况

## 7 问题二：锥形队列调整方案

### 7.1 调整方案

我们沿用上面的策略，考虑对称性，我们选择 FY01 和 FY05 的位置来标定无人队列。接下来，我们还需要确定一个位置固定的点。观察图7.1，我们可以发现，在位置正确的情况下，FY01、FY07 和 FY10 在以 FY05 为圆心的圆周上，且将圆周三等分。

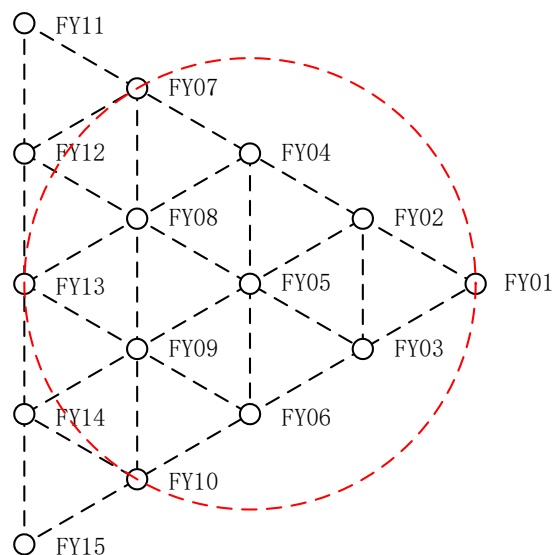


图 7.1: 问题 2 示意图

这样，我们就可以给出我们的调整策略：

1. 按照问题一中的策略，将 FY01、FY05、FY07 和 FY10 看作一个圆形无人机编队。

将 FY05 作为圆心，FY01 作为“半径标定机”。调整 FY05 和 FY07 的位置，使其逼近正确位置；

2. FY05 和 FY07 的位置正确后，其他无人机直接通过与 FY07 和 FY05 还有 FY10 和 FY05 之间的夹角，直接确定自身位置。

我们记一架无人机与 FY05 和 FY07 形成的夹角为  $\alpha_7$ ，与 FY05 和 FY10 形成的夹角为  $\alpha_{10}$ 。则其应当调整至的标准角度如表7.1所示。

表 7.1: 15 架无人机队列标准角度

无人机编号	$\alpha_7$	$\alpha_{10}$	无人机编号	$\alpha_7$	$\alpha_{10}$
FY02	$\frac{\pi}{3}$	$\arctan \frac{\sqrt{3}}{5}$	FY11	$\arctan \frac{\sqrt{3}}{5}$	$\arctan \frac{5\sqrt{3}}{17}$
FY03	$\arctan \frac{\sqrt{3}}{5}$	$\frac{\pi}{3}$	FY12	$\frac{\pi}{3}$	$\arctan \frac{\sqrt{3}}{2}$
FY04	$\frac{2\pi}{3}$	$\arctan \frac{\sqrt{3}}{5}$	FY13	$\frac{\pi}{3}$	$\frac{\pi}{3}$
FY06	$\arctan \frac{\sqrt{3}}{5}$	$\frac{2\pi}{3}$	FY14	$\arctan \frac{\sqrt{3}}{2}$	$\frac{\pi}{3}$
FY08	$\frac{2\pi}{3}$	$\frac{\pi}{3}$	FY15	$\arctan \frac{5\sqrt{3}}{17}$	$\arctan \frac{\sqrt{3}}{5}$
FY09	$\frac{\pi}{3}$	$\frac{2\pi}{3}$			

## 7.2 调整方案的模拟

我们首先将 15 个无人机的位置设置在正确位置，然后给除了 FY01 和 FY05 外所有的无人机加上一个长度随机，方向随机的偏移。这个偏移的长度符合  $U(0, \frac{S}{2})$ ，其中  $S$  为等边三角形的边长。然后我们使用我们的调整方案，将无人机移动到正确的位置。

同样的，我们修改了我们的模拟方案，在 FY07 和 FY10 每次迭代调整结束之后，我们都会立即更新所有无人机的位置，然后计算损失函数：

$$L = \sum_{i=1}^{15} d_i^2 \quad (7.1)$$

最终，调整前后的无人机位置对比如图7.2所示，损失函数随迭代次数的变化如图7.3所示。损失函数的具体数值见表7.2。

表 7.2: 锥形队列损失函数变化情况

迭代次数	损失函数	迭代次数	损失函数
0	$2.41 \times 10^3$	3	$3.70 \times 10^{-10}$
1	$2.91 \times 10^1$	4	$3.70 \times 10^{-10}$
2	$3.70 \times 10^{-10}$	5	$3.70 \times 10^{-10}$

我们从中可以看出，我们的对于锥形队列无人机的调整方案也非常有效，可以很好地将无人机固定在正确的位置。

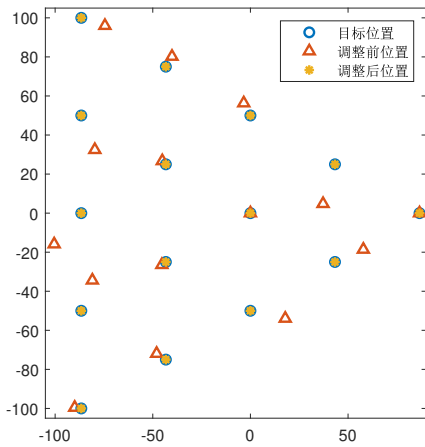


图 7.2: 锥形队列无人机位置对比

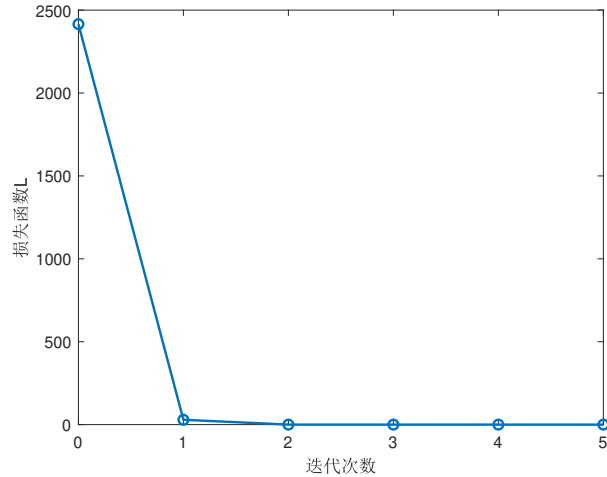


图 7.3: 锥形队列损失函数变化情况

## 8 模型评价

我们的模型具有以下优点：

- 在第一问第 1 小问中，我们的基于两圆相交的定位模型，可以给出待定位无人机位置的解析解；
- 在第一问第 2 小问中，我们基于严格的数学证明，证明了在待定无人机的位置偏差满足式5.4的情况下，通过额外两架无人机发射的信号，就可以确定待定无人机的位置；
- 在第一问第 2 小问中，我们还讨论了一架额外无人机就可以确定待定无人机位置的情况；
- 在第一问第 3 小问中，我们提出了一种基于迭代的调整方案，可以使得无人机队列快速逼近正确位置，并且给出了收敛更快的优化后的调整策略；
- 在第二问中，我们将其转化为了与第一问相似的问题，并进行了解决。

但是我们的模型还具有以下不足：

- 我们的模型对于误差范围更加极端的情况，适应性较差；
- 对于第二问，没有讨论待定无人机位置偏差不能满足式5.4的情况。

## 9 模型改进与推广

我们可以考虑如下的改进思路：

- 可以考虑无人机之间可以互相通信的情况；

- 可以考虑无人机按非中心对称的方式排列的情况；
- 可以考虑三维情况下的无人机定位问题；

而且我们的模型实际上并没有使用无人机所特有的性质，其可以推广到所有需要进行二维平面无源定位的情况。

## 参考文献

- [1] 司守奎, 孙玺菁. 数学建模算法与应用[M]. 国防工业出版社, 2022.



# 附录

## A 使用代码

### A.1 calculateAngle.m

```
function angle = calculateAngle(N1,N2,O)
%CALCULATEANGLE 计算角 N1ON2
% 其中 O 为角的顶点, N1、N2 和 O 用二维行向量表示笛卡尔坐标系下的坐标
% 返回角度的单位为弧度
ON1 = N1 - O;
ON2 = N2 - O;
angle = atan2( ...
    norm(cross([ON1,O],[ON2,O])), ...
    dot(ON1,ON2) ...
);
end
```

### A.2 moveToCertainAngle.m

```
function newR = moveToCertainAngle(A,B,O,R,ARO,BRO)
%MOVETOCERTAINANGLE 将 R 移动到 newR 位置, 使得角 ARO 与角 BRO 为特定值
% A,B,O,R 为用二维行向量表示笛卡尔坐标系下的坐标
% ARO 与 BRO 为指定角度, 弧度制
fun = @(x)[calculateAngle(A,O,x)-ARO,calculateAngle(B,O,x)-BRO];
opt = optimset('Display','off','FunValCheck','on','TolFun',1e-9);
newR = fsolve(fun,R,opt);
end
```

### A.3 radiusAdjust.m

```
function points = radiusAdjust(points,Rid,Did,O)
%RADIUSADJUST 对 points 中 Rid 的无人机以 Did 为标定进行半径调整

interval = mod(Rid - Did, length(points));
Lid = Did - interval;
while Lid < 1 || Lid > length(points)
    if(Lid < 1)
```

```

        Lid = Lid + length(points);
    else
        Lid = Lid - length(points);
    end
end

ROD = interval*2*pi/length(points);
if ROD > pi
    ROD = 2*pi-ROD;
end

ROL = ROD*2;
if ROL > pi
    ROL = 2*pi-ROL;
end

OR1D = (pi - ROD) ./ 2;
ORD = calculateAngle(points(Did,:),0,points(Rid,:));
ORL = calculateAngle(points(Lid,:),0,points(Rid,:));

k = sin(ORL) * sin(ROD + ORD) / (sin(ORD) * sin(ROL + ORL));
OR1L = atan(sin(ROL)./(1./k - cos(ROL)));

points(Rid,:) = moveToCertainAngle(points(Did,:), points(Lid,:), 0,
    ↪ points(Rid,:), OR1D, OR1L);

end

```

## A.4 angleAdjust.m

```

function points = angleAdjust(points,id,0)
%ANGLEADJUST 对 Points 中指定的 id 的点进行角度矫正

% 顺时针方向下一个点
if id == length(points)
    left = 1;
else

```

```

        left = id + 1;
end
% 顺时针方向上一个点
if id == 1
    right = length(points);
else
    right = id - 1;
end

target = (pi - 2*pi / length(points)) / 2;

points(id,:) = moveToCertainAngle(points(left,:), points(right,:), 0,
    ↪ points(id,:), target, target);

end

```

## A.5 question1\_3.m

```

clearvars

% 初始化
FY00pc = [0,0]; FY00xy = [0,0];
FYpc_g = [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];
FYpc_t = [100,0;100,40;100,80;100,120;100,160;100,200;
    100,240;100,280;100,320];
FYxy_t = zeros(9,2);FYxy_g = zeros(9,2);FYxy = zeros(9,2);
[FYxy_g(:,1),FYxy_g(:,2)] = pol2cart(deg2rad(FYpc_g(:,2)),FYpc_g(:,1));
[FYxy_t(:,1),FYxy_t(:,2)] = pol2cart(deg2rad(FYpc_t(:,2)),FYpc_t(:,1));

times = 10;

% 进行半径矫正组
[FYxy(:,1),FYxy(:,2)] = pol2cart(deg2rad(FYpc_g(:,2)),FYpc_g(:,1));
L1 = zeros(1,times + 1);
L1(1) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) - FYxy_t(:,2)).^2);
for k = 1:times

```

```

    if k == 1
        for i = 2:9
            FYxy = radiusAdjust(FYxy,i,1,FY00xy);
        end
        [tempTheta, tempRho] = cart2pol(FYxy(:,1),FYxy(:,2));
    end
    for i = 2:9
        FYxy = angleAdjust(FYxy,i,FY00xy);
    end
    L1(1 + k) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) -
        ↪ FYxy_t(:,2)).^2);
end

% 不进行半径矫正组
[FYxy(:,1),FYxy(:,2)] = pol2cart(deg2rad(FYpc_g(:,2)),FYpc_g(:,1));
L2 = zeros(1,times + 1);
L2(1) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) - FYxy_t(:,2)).^2);
for k = 1:times
    for i = 2:9
        FYxy = angleAdjust(FYxy,i,FY00xy);
    end
    L2(1 + k) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) -
        ↪ FYxy_t(:,2)).^2);
end

f1 = figure(1);
polarplot(deg2rad(FYpc_t([1:end,1],2)), FYpc_t([1:end,1],1), '-o',
    ↪ LineWidth=1.5)
hold on
polarplot(deg2rad(FYpc_g([1:end,1],2)), FYpc_g([1:end,1],1), '-^',
    ↪ LineWidth=1.5)
polarplot(tempTheta([1:end,1]), tempRho([1:end,1]),'--*', LineWidth=1.5)
legend(" 目标位置", " 起始位置", " 半径矫正后位置",
    ↪ Location="southoutside")
rticks([])
thetaticks([0 40 80 120 160 200 240 280 320])

```

```

f2 = figure(2);
plot(0:1:times,L2,'-o',LineWidth=1.5)
hold on
plot(0:1:times,L1,'-^',LineWidth=1.5)
xlabel(" 迭代次数")
ylabel(" 损失函数 L")
legend([" 不进行半径矫正"," 进行半径矫正"])

```

## A.6 question1\_\_3\_\_2.m

```
clearvars
```

```
% 初始化
```

```

FY00pc = [0,0]; FY00xy = [0,0];
FYpc_t = [100,0;100,40;100,80;100,120;100,160;100,200;
          100,240;100,280;100,320];
FYxy_t = zeros(9,2);
[FYxy_t(:,1),FYxy_t(:,2)] = pol2cart(deg2rad(FYpc_t(:,2)),FYpc_t(:,1));

```

```

times = 10;
num = 5;
L = zeros(num,times + 1);

```

```
rng(20240418)
```

```
% 进行半径矫正组
```

```

for j = 1:num
    FYpc_g = FYpc_t;
    FYpc_g(2:end,2) = FYpc_g(2:end,2) + normrnd(0,1,[8,1]);
    FYpc_g(2:end,1) = FYpc_g(2:end,1) + normrnd(0,10,[8,1]);
    FYxy = zeros(9,2);
    [FYxy(:,1),FYxy(:,2)] = pol2cart(deg2rad(FYpc_g(:,2)),FYpc_g(:,1));
    L(j,1) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) -
        ↪ FYxy_t(:,2)).^2);
    for k = 1:times
        if k == 1
            for i = 2:9

```

```

        FYxy = radiusAdjust(FYxy,i,1,FY00xy);
    end
    [tempTheta, tempRho] = cart2pol(FYxy(:,1),FYxy(:,2));
end
for i = 2:9
    FYxy = angleAdjust(FYxy,i,FY00xy);
end
L(j,1 + k) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) -
    ↪ FYxy_t(:,2)).^2);
end
end

f = figure(1);
for i = 1:num
    plot(0:1:times,L(i,1:end),'-o',LineWidth=1)
    hold on
end
xlabel(" 迭代次数")
ylabel(" 损失函数 L")

```

## A.7 question1\_3\_3.m

```

clearvars

% 初始化
FY00pc = [0,0]; FY00xy = [0,0];
FYpc_t = [100,0;100,40;100,80;100,120;100,160;100,200;
    100,240;100,280;100,320];
FYxy_t = zeros(9,2);
[FYxy_t(:,1),FYxy_t(:,2)] = pol2cart(deg2rad(FYpc_t(:,2)),FYpc_t(:,1));

times = 5;
num = 5;
L = zeros(num,times + 1);

rng(20240418)

```

```

for j = 1:num
    FYpc_g = FYpc_t;
    FYpc_g(2:end,2) = FYpc_g(2:end,2) + normrnd(0,1,[8,1]);
    FYpc_g(2:end,1) = FYpc_g(2:end,1) + normrnd(0,10,[8,1]);
    FYxy = zeros(9,2);
    [FYxy(:,1),FYxy(:,2)] = pol2cart(deg2rad(FYpc_g(:,2)),FYpc_g(:,1));
    L(j,1) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) -
        ↪ FYxy_t(:,2)).^2);
    for k = 1:times
        c147 = FYxy([1,4,7],:);
        if k == 1
            for i = 2:3
                c147 = radiusAdjust(c147,i,1,FY00xy);
            end
        end
        for i = 2:3
            c147 = angleAdjust(c147,i,FY00xy);
        end
        FYxy([1,4,7],:) = c147;
        % 调整无人机所用函数
        fmid = @(x) moveToCertainAngle( ...
            FYxy(4,:),FYxy(7,:),FY00xy,FYxy(x,:), ...
            calculateAngle(FYxy_t(4,:),FY00xy,FYxy_t(x,:)), ...
            calculateAngle(FYxy_t(7,:),FY00xy,FYxy_t(x,:)));
        % 依次根据角度调整剩下的无人机
        FYxy(2,:) = fmid(2);
        FYxy(3,:) = fmid(3);
        FYxy(5,:) = fmid(5);
        FYxy(6,:) = fmid(6);
        FYxy(8,:) = fmid(8);
        FYxy(9,:) = fmid(9);
        L(j,1 + k) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) -
            ↪ FYxy_t(:,2)).^2);
    end
end

f = figure(1);

```



```

for i = 1:num
    plot(0:1:times,L(i,1:end),'-o',LineWidth=1)
    hold on
end
xlabel(" 迭代次数")
ylabel(" 损失函数 L")

```

## A.8 question2.m

```

clearvars

sideLength = 50;    % 等边三角形边长
times = 5;
% 标准位置
FYxy_t = [sqrt(3),0;sqrt(3)/2,1/2;sqrt(3)/2,-1/2;0,1;0,0;
    0,-1;-sqrt(3)/2,3/2;-sqrt(3)/2,1/2;-sqrt(3)/2,-1/2;-sqrt(3)/2,-3/2;
    -sqrt(3),2;-sqrt(3),1;-sqrt(3),0;-sqrt(3),-1;-sqrt(3),-2] *
    ↪ sideLength;
% 初始位置, 在标准位置上添加偏差, FY01 和 FY05 作为基准, 保持不变
FYxy_g = FYxy_t;
rng(20240415)    % 指定随机数种子, 保证可重复性
for i = 1:15
    if i == 1 || i==5
        continue
    end
    eRho = unifrnd(0,sideLength/2);
    eTheta = unifrnd(0,2*pi);
    FYxy_g(i,1) = FYxy_g(i,1) + eRho * cos(eTheta);
    FYxy_g(i,2) = FYxy_g(i,2) + eRho * sin(eTheta);
end
FYxy = FYxy_g;

% 调整 FY01, FY07 和 FY10 使其在以 FY05 为圆心的圆上
c1710 = FYxy([1,7,10],:);
L = zeros(1,times + 1);
L(1) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) - FYxy_t(:,2)).^2);
for k = 1:times

```

```

if k == 1
    for i = 2:3
        c1710 = radiusAdjust(c1710,i,1,FYxy(5,:));
    end
end
for i = 2:3
    c1710 = angleAdjust(c1710,i,FYxy(5,:));
end
FYxy([1,7,10],:) = c1710;
% 调整无人机所用函数
fmid = @(x) moveToCertainAngle( ...
    FYxy(7,:),FYxy(10,:),FYxy(5,:),FYxy(x,:), ...
    calculateAngle(FYxy_t(7,:),FYxy_t(5,:),FYxy_t(x,:)), ...
    calculateAngle(FYxy_t(10,:),FYxy_t(5,:),FYxy_t(x,:)));
% 依次根据角度调整剩下的无人机
FYxy(2,:) = fmid(2);
FYxy(3,:) = fmid(3);
FYxy(4,:) = fmid(4);
FYxy(6,:) = fmid(6);
FYxy(8,:) = fmid(8);
FYxy(9,:) = fmid(9);
FYxy(11,:) = fmid(11);
FYxy(12,:) = fmid(12);
FYxy(13,:) = fmid(13);
FYxy(14,:) = fmid(14);
FYxy(15,:) = fmid(15);
L(1 + k) = sum((FYxy(:,1) - FYxy_t(:,1)).^2 + (FYxy(:,2) -
    FYxy_t(:,2)).^2);
end

f1 = figure(1);
plot(FYxy_t(:,1),FYxy_t(:,2),'o',LineWidth=1.5)
hold on
plot(FYxy_g(:,1),FYxy_g(:,2),'^',LineWidth=1.5)
plot(FYxy(:,1),FYxy(:,2),'*',LineWidth=1.5)
axis equal
axis([-105,90,-105,105])

```

```
legend([" 目标位置"," 调整前位置"," 调整后位置"])
```

```
f2 = figure(2);
```

```
plot(0:1:times,L,'-o',LineWidth=1.5)
```

```
xlabel(" 迭代次数")
```

```
ylabel(" 损失函数 L")
```